Multiple Regression

This chapter deals with the **regression problem** where the goal is to understand the relationship between a specific variable called the *response* variable and several other related variables called *explanatory* varibles. Here are some natural examples.

1. Prospective buyers and sellers might want to understand how the price of a house depends on various characteristics of the house such as the total above ground living space, total basement square footage, lot area, number of cars that can be parked in the garage, construction year and presence or absence of a fireplace. This is an instance of a regression problem where the response variable is the house price and the other characteristics of the house listed above are the explanatory variables.

2. A bike rental company wants to understand how the number of bike rentals in a given hour depends on environmental and seasonal variables (such as temperature, humidity, presence of rain etc.) and various other factors such as weekend or weekday, holiday etc. This is also an instance of a regression problems where the response variable is the number of bike rentals and all other variables mentioned are explanatory variables.

3. We might want to understand how the retention rates of colleges depend on various aspects such as tuition fees, faculty salaries, number of faculty members that are full time, number of undergraduates enrolled, number of students on federal loans etc. This is again a regression problem with the response variable being the retention rate and other variables being the explanatory variables.

4. I might be interested in understanding the proportion of my body weight that is fat (body fat percentage). Directly measuring this quantity is probably hard but I can easily obtain various body measurements such as height, weight, age, chest circumeference, abdomen circumference, hip circumference and thigh circumference. Can I predict my body fat percentage based on these measurements? This is again a regression problem with the response variable being body fat percentage and all the measurements are explanatory variables.

How does one solve such problems? The first step is to obtain data on the relevant variables. Below we shall look at datasets which will enable us to provide some reasonable answers to each of the above questions.

1 Datasets

1.1 The Ames Housing Dataset

This dataset contains information on sales of houses in Ames, Iowa from 2006 to 2010. The full dataset can be obtained by following links given in the paper: `https://ww2.amstat.org/publications/jse/v19n3/decock.pdf`). I have shortened the dataset slightly to make life easier for us.

```
dataDir <- "../../finalDataSets"
dd = read.csv(file.path(dataDir, "Ames_Short.csv"),
    header = T)
dim(dd)
```

```
## [1] 1314    7
```

```
names(dd)
```

```
## [1] "Lot.Area"      "Total.Bsmt.SF" "Gr.Liv.Area"   "Garage.Cars"
## [5] "Fireplaces.YN" "Year.Built"    "SalePrice"
```

```
head(dd)
```

```
##   Lot.Area Total.Bsmt.SF Gr.Liv.Area Garage.Cars Fireplaces.YN Year.Built
## 1    11622           882         896           1             N       1961
## 2    14267          1329        1329           1             N       1958
## 3     4920          1338        1338           2             N       2001
## 4     5005          1280        1280           2             N       1992
## 5     7980          1168        1187           2             N       1992
## 6     8402           789        1465           2             Y       1998
##   SalePrice
## 1    105000
## 2    172000
## 3    213500
## 4    191500
## 5    185000
## 6    180400
```

```
summary(dd)
```

```
##     Lot.Area     Total.Bsmt.SF     Gr.Liv.Area    Garage.Cars
##  Min.   : 1300   Min.   : 105.0   Min.   : 438   Min.   :0.00
##  1st Qu.: 6346   1st Qu.: 732.0   1st Qu.: 984   1st Qu.:1.00
##  Median : 8472   Median : 923.0   Median :1151   Median :2.00
##  Mean   : 8515   Mean   : 938.1   Mean   :1149   Mean   :1.49
##  3rd Qu.:10200   3rd Qu.:1138.0   3rd Qu.:1344   3rd Qu.:2.00
##  Max.   :41600   Max.   :1645.0   Max.   :1500   Max.   :5.00
##  Fireplaces.YN   Year.Built     SalePrice
##  N:830         Min.   :1875   Min.   : 35000
##  Y:484         1st Qu.:1950   1st Qu.:120000
##                Median :1966   Median :138750
##                Mean   :1964   Mean   :141447
##                3rd Qu.:1981   3rd Qu.:160500
##                Max.   :2010   Max.   :290000
```

1.2 Bike Sharing Dataset

This dataset (from the UCI machine learning repository) contains information on bike rentals for two years (2011 and 2012) from Capital Bikeshare System, Washington D.C. The data is collected to address the problem of predicting the number of bike rentals in a given hour given the environmental and seasonal conditions for that hour.

```
bike <- read.csv(file.path(dataDir, "BikeSharingDataset.csv"))
dim(bike)
```

```
## [1] 17379    17
```

```
names(bike)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"
##  [6] "hr"         "holiday"    "weekday"    "workingday" "weathersit"
## [11] "temp"       "atemp"      "hum"        "windspeed"  "casual"
## [16] "registered" "cnt"
```

```
head(bike)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday
```

```
## 1       1 2011-01-01      1  0    1  0       0       6          0
## 2       2 2011-01-01      1  0    1  1       0       6          0
## 3       3 2011-01-01      1  0    1  2       0       6          0
## 4       4 2011-01-01      1  0    1  3       0       6          0
## 5       5 2011-01-01      1  0    1  4       0       6          0
## 6       6 2011-01-01      1  0    1  5       0       6          0
##   weathersit temp  atemp  hum windspeed casual registered cnt
## 1          1 0.24 0.2879 0.81    0.0000      3         13  16
## 2          1 0.22 0.2727 0.80    0.0000      8         32  40
## 3          1 0.22 0.2727 0.80    0.0000      5         27  32
## 4          1 0.24 0.2879 0.75    0.0000      3         10  13
## 5          1 0.24 0.2879 0.75    0.0000      0          1   1
## 6          2 0.24 0.2576 0.75    0.0896      0          1   1
```

```
summary(bike)
```

```
##     instant              dteday          season           yr
##  Min.   :    1   2011-01-01:   24   Min.   :1.000   Min.   :0.0000
##  1st Qu.: 4346   2011-01-08:   24   1st Qu.:2.000   1st Qu.:0.0000
##  Median : 8690   2011-01-09:   24   Median :3.000   Median :1.0000
##  Mean   : 8690   2011-01-10:   24   Mean   :2.502   Mean   :0.5026
##  3rd Qu.:13034   2011-01-13:   24   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :17379   2011-01-15:   24   Max.   :4.000   Max.   :1.0000
##                  (Other)   :17235
##       mnth              hr            holiday           weekday
##  Min.   : 1.000   Min.   : 0.00   Min.   :0.00000   Min.   :0.000
##  1st Qu.: 4.000   1st Qu.: 6.00   1st Qu.:0.00000   1st Qu.:1.000
##  Median : 7.000   Median :12.00   Median :0.00000   Median :3.000
##  Mean   : 6.538   Mean   :11.55   Mean   :0.02877   Mean   :3.004
##  3rd Qu.:10.000   3rd Qu.:18.00   3rd Qu.:0.00000   3rd Qu.:5.000
##  Max.   :12.000   Max.   :23.00   Max.   :1.00000   Max.   :6.000
##
##    workingday       weathersit         temp           atemp
##  Min.   :0.0000   Min.   :1.000   Min.   :0.020   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.340   1st Qu.:0.3333
##  Median :1.0000   Median :1.000   Median :0.500   Median :0.4848
##  Mean   :0.6827   Mean   :1.425   Mean   :0.497   Mean   :0.4758
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.660   3rd Qu.:0.6212
##  Max.   :1.0000   Max.   :4.000   Max.   :1.000   Max.   :1.0000
##
##       hum           windspeed          casual         registered
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.:  4.00   1st Qu.: 34.0
```

```
##  Median :0.6300   Median :0.1940   Median : 17.00   Median :115.0
##  Mean   :0.6272   Mean   :0.1901   Mean   : 35.68   Mean   :153.8
##  3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.: 48.00   3rd Qu.:220.0
##  Max.   :1.0000   Max.   :0.8507   Max.   :367.00   Max.   :886.0
##
##       cnt
##  Min.   :  1.0
##  1st Qu.: 40.0
##  Median :142.0
##  Mean   :189.5
##  3rd Qu.:281.0
##  Max.   :977.0
##
```

Here is a description of the variables in this dataset: The dataset contains 17379 observations with each observation corresponding to one particular hour. The dataset contains the following 17 variables:

1. **instant** : Serial number.
2. **dteday**: This is date.
3. **season**: Categorical variable (1: Spring, 2: Summer, 3: Fall, 4: Winter).
4. **yr**: Stands for year. Binary variable (0 stands for 2011 and 1 stands for 2012).
5. **mnth**: Stands for month. Takes the values $1, 2, \ldots, 12$.
6. **hr**: Indicates the hour of the day (takes values $0, \ldots, 23$).
7. **holiday**: Indicates whether the day is a holiday or not
8. **weekday**: Self explanatory
9. **workingday**: Takes the value 1 if the day is neither weekend nor holiday and takes the value 0 otherwise.
10. **weathersit**: Takes four values:
    (a) 1 if the weather is Clear, Few clouds, Partly cloudy, Partly cloudy.
    (b) 2 if the weather is Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist.
    (c) 3 if the weather is Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds.
    (d) 4 if the weather is Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog.

11. **temp**: Normalized temperature in Celsius. The values are divided by 41 (the maximum temperature).
12. **atemp**: Normalized feeling temperature in Celsius. The values are divided by 50 (the maximum temperature).
13. **hum**: Normalized humidity. The values are divided by 100 (the maximum humidity).
14. **windspeed**: Normalized wind speed. The values are divided by 67 (maximum wind speed).
15. **casual**: The number of bikes rented by casual (unregistered) users for that hour.
16. **registered**: The number of bikes rented by registered users for that hour.
17. **cnt**: The number of bikes rented by both casual and registered users for that hour (this is the sum of **casual** and **registered**).

1.3 College Dataset

You have already seen this dataset in this class (when discussing linear regression as part of curve fitting). This dataset contains information on colleges and universities for the year 2014 such as data on tuition fees (both in state and out of state), enrollments, demographics of student population, faculty salaries, student retention rates etc. I have reduced the size of this dataset by omitting some of the variables.

Here are the variable descriptions:

1. SATAVGALL: SAT average score.
2. AVGFACSAL: average faculty monthly salary.
3. TUITIONFEEIN: annual in-state students tuition.
4. TUITIONFEEOUT: annual out-of-state students tuition.
5. UGDS: number of undergraduate students.
6. RETFT4: full time student retention rate (students who return to the institution after the first year)
7. PCTFLOAN: percentage of undergraduates receiving federal loans.
8. PFTFAC: proportion of faculty that is full time.

9. TYPE: public(1), private nonprofit (2) or private for profit (3) (previously this variable was called CONTROL; I changed the name because TYPE seems a better name for this variable compared to CONTROL).

```
scorecard = read.csv(file.path(dataDir, "college_short.csv"))
dim(scorecard)
```

```
## [1] 1241    9
```

```
names(scorecard)
```

```
## [1] "SAT_AVG_ALL"    "AVGFACSAL"      "TUITIONFEE_IN"  "TUITIONFEE_OUT"
## [5] "UGDS"           "RET_FT4"        "PCTFLOAN"       "PFTFAC"
## [9] "TYPE"
```

```
head(scorecard)
```

```
##   SAT_AVG_ALL AVGFACSAL TUITIONFEE_IN TUITIONFEE_OUT  UGDS RET_FT4
## 1         823      7079          7182          12774  4051  0.6314
## 2        1146     10170          7206          16398 11200  0.8016
## 3        1180      9341          9192          21506  5525  0.8098
## 4         830      6557          8720          15656  5354  0.6219
## 5        1171      9605          9450          23950 28692  0.8700
## 6        1215      9429          9852          26364 19761  0.8946
##   PCTFLOAN PFTFAC TYPE
## 1   0.8204 0.8856    1
## 2   0.5397 0.9106    1
## 3   0.4728 0.6555    1
## 4   0.8735 0.6641    1
## 5   0.4148 0.7109    1
## 6   0.3610 0.8780    1
```

```
summary(scorecard)
```

```
##   SAT_AVG_ALL     AVGFACSAL     TUITIONFEE_IN   TUITIONFEE_OUT
##  Min.   : 666   Min.   : 1476   Min.   : 2082   Min.   : 3850
##  1st Qu.: 980   1st Qu.: 6152   1st Qu.: 9356   1st Qu.:18395
##  Median :1050   Median : 7280   Median :22290   Median :24566
```

```
##  Mean   :1067   Mean   : 7594   Mean   :21708   Mean   :25651
##  3rd Qu.:1137   3rd Qu.: 8626   3rd Qu.:30990   3rd Qu.:31500
##  Max.   :1534   Max.   :19862   Max.   :49138   Max.   :49138
##       UGDS          RET_FT4          PCTFLOAN          PFTFAC
##  Min.   :   82   Min.   :0.0000   Min.   :0.0000   Min.   :0.0403
##  1st Qu.: 1313   1st Qu.:0.6852   1st Qu.:0.4922   1st Qu.:0.5281
##  Median : 2466   Median :0.7640   Median :0.6133   Median :0.7111
##  Mean   : 5545   Mean   :0.7609   Mean   :0.5990   Mean   :0.7037
##  3rd Qu.: 6308   3rd Qu.:0.8457   3rd Qu.:0.7209   3rd Qu.:0.9363
##  Max.   :50919   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##       TYPE
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.646
##  3rd Qu.:2.000
##  Max.   :3.000
```

1.4 Bodyfat Dataset

A dataset that is often used in classrooms to demonstrate regression techniques is the bodyfat dataset. Body fat percentage (computed by a complicated underwater weighing technique) along with various body measurements are given for 252 adult men.

```
body = read.csv(file.path(dataDir, "bodyfat_short.csv"),
    header = T)
dim(body)
```

```
## [1] 252   8
```

```
names(body)
```

```
## [1] "BODYFAT" "AGE"     "WEIGHT"  "HEIGHT"  "CHEST"   "ABDOMEN" "HIP"
## [8] "THIGH"
```

```
head(body)
```

```
##   BODYFAT AGE WEIGHT HEIGHT CHEST ABDOMEN   HIP THIGH
## 1    12.3  23 154.25  67.75  93.1    85.2  94.5  59.0
## 2     6.1  22 173.25  72.25  93.6    83.0  98.7  58.7
## 3    25.3  22 154.00  66.25  95.8    87.9  99.2  59.6
## 4    10.4  26 184.75  72.25 101.8    86.4 101.2  60.1
## 5    28.7  24 184.25  71.25  97.3   100.0 101.9  63.2
## 6    20.9  24 210.25  74.75 104.5    94.4 107.8  66.0
```

```
summary(body)
```

```
##     BODYFAT           AGE            WEIGHT          HEIGHT
##  Min.   : 0.00   Min.   :22.00   Min.   :118.5   Min.   :29.50
##  1st Qu.:12.47   1st Qu.:35.75   1st Qu.:159.0   1st Qu.:68.25
##  Median :19.20   Median :43.00   Median :176.5   Median :70.00
##  Mean   :19.15   Mean   :44.88   Mean   :178.9   Mean   :70.15
##  3rd Qu.:25.30   3rd Qu.:54.00   3rd Qu.:197.0   3rd Qu.:72.25
##  Max.   :47.50   Max.   :81.00   Max.   :363.1   Max.   :77.75
##      CHEST           ABDOMEN            HIP            THIGH
##  Min.   : 79.30   Min.   : 69.40   Min.   : 85.0   Min.   :47.20
##  1st Qu.: 94.35   1st Qu.: 84.58   1st Qu.: 95.5   1st Qu.:56.00
##  Median : 99.65   Median : 90.95   Median : 99.3   Median :59.00
##  Mean   :100.82   Mean   : 92.56   Mean   : 99.9   Mean   :59.41
##  3rd Qu.:105.38   3rd Qu.: 99.33   3rd Qu.:103.5   3rd Qu.:62.35
##  Max.   :136.20   Max.   :148.10   Max.   :147.7   Max.   :87.30
```

These datasets allow us to provide reasonable answers to the four questions that we asked at the beginning of the lecture. The goal of a regression problem is always to understand the relationship between a response variable and a bunch of explanatory variables. This will, in turn, allow one to predict the value of the response variable given the explanatory variable values of future observations.

Multiple linear regression is one of the most widely used techniques for solving the regression problem. This is the subject of this chapter. Before goint to multiple linear regression however, let us first do a brief recap of simple linear regression.

2 Brief review of Simple Linear Regression

In simple regression, there are two variables: one response variable and one explanatory variable. The goal is to understand the relation between the response and the

explanatory variables.

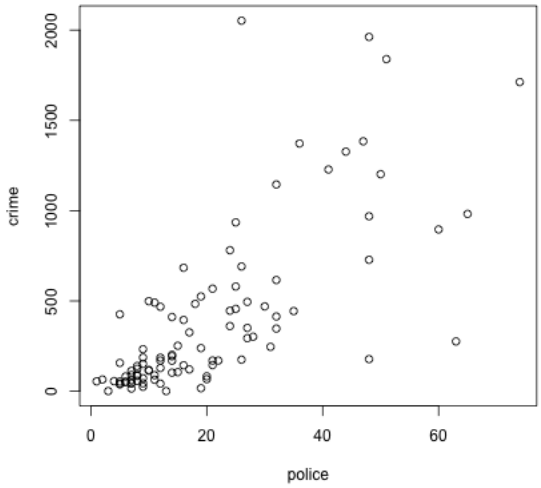
Let us take a simple example.

```
load(file.path(dataDir, "campus.Rdata"))
campus = data
campus.desc = desc
summary(campus)
```

```
##      enroll           priv            police          crime
##  Min.   : 1799   Min.   :0.0000   Min.   : 1.00   Min.   :   1.0
##  1st Qu.: 6485   1st Qu.:0.0000   1st Qu.: 9.00   1st Qu.:  85.0
##  Median :11990   Median :0.0000   Median :16.00   Median : 187.0
##  Mean   :16076   Mean   :0.1237   Mean   :20.49   Mean   : 394.5
##  3rd Qu.:21836   3rd Qu.:0.0000   3rd Qu.:27.00   3rd Qu.: 491.0
##  Max.   :56350   Max.   :1.0000   Max.   :74.00   Max.   :2052.0
##      lcrime         lenroll          lpolice
##  Min.   :0.000   Min.   : 7.495   Min.   :0.000
##  1st Qu.:4.443   1st Qu.: 8.777   1st Qu.:2.197
##  Median :5.231   Median : 9.392   Median :2.773
##  Mean   :5.277   Mean   : 9.379   Mean   :2.731
##  3rd Qu.:6.196   3rd Qu.: 9.991   3rd Qu.:3.296
##  Max.   :7.627   Max.   :10.939   Max.   :4.304
```

Suppose now that we are interested in the relationship between *crime* (response variable) and *police* (explanatory variable). The scatter plot between these two variables is:

```
plot(crime ~ police, data = campus)
```

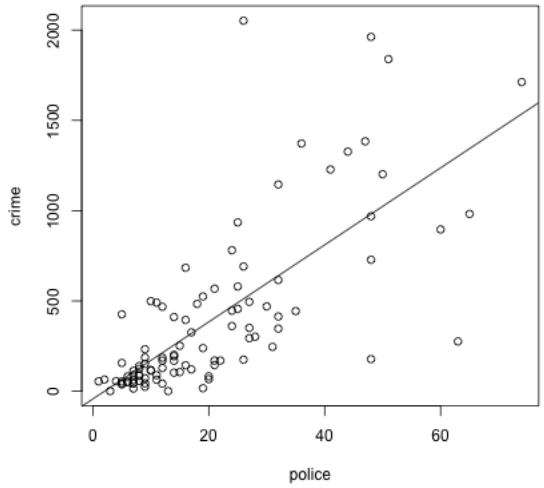


Simple linear regression will now fit a line to this dataset. This line can be obtained in R via the following line of code.

```
m1 = lm(crime ~ police, data = campus)
summary(m1)
```

```
##
## Call:
## lm(formula = crime ~ police, data = campus)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1024.79  -152.96   -35.38    89.48  1540.16
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -42.557     53.730  -0.792     0.43
## police        21.323      2.089  10.210   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 319.9 on 95 degrees of freedom
## Multiple R-squared:  0.5232,Adjusted R-squared:  0.5182
## F-statistic: 104.2 on 1 and 95 DF,  p-value: < 2.2e-16
```

```
plot(crime ~ police, data = campus)
abline(m1)
```



The fitted line here has the equation

$$y = -42.557 + 21.323x \tag{1}$$

where y represents the *crime* variable and x represents the *police* variable. Some FAQs and answers:

1. **How does R determine this equation?**: This line is determined by minimizing sum of squares.

2. **What is the interpretation of this equation?**: The equation (1) clearly implies that y increases by 21.323 for every unit increase in x. Note here that y represents the variable *crime* and x represents the variable *police*.

3. **Is the equation** (1) **reasonable?** I would argue that it does not make much sense in this context. One way of interpreting (1) is to say that if the number of police officers increases by 1, then there will be 21 more incidents of crime on average (**this statement should not be interpreted causally**). Now if there are 100 police officers in a particular campus, increasing the number of police officers to 101 should not really change anything all that much. On the other hand, there might be bigger differences when the number changes from 3 to 4. But the equation (1), in both of these scenarios, says that the incidents of crime increase in number by 21.

Another issue here is *heteroscedasticity*. There seems to be much more **variability** in the values of y for large values of x compared to the small values of x. As we shall see later, heteroscedasticity causes some problems for the least squares line.

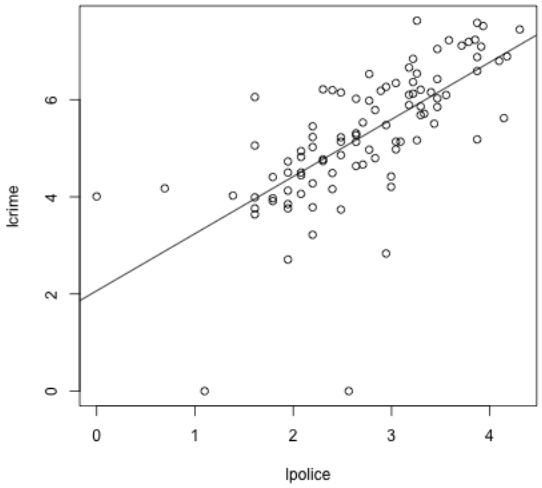
Log-transformation Suppose now that instead of fitting a regression line to *crime* in terms of *police*, we now fit a line to log(*crime*) in terms of log(*police*) (these two logged variables are saved in the *campus* dataset as *lcrime* and *lpolice* respectively).

The scatter plot between *lcrime* and *lpolice* along with the least squares line are given below.

```
m2 = lm(lcrime ~ lpolice, data = campus)
summary(m2)
```

```
##
## Call:
## lm(formula = lcrime ~ lpolice, data = campus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0819 -0.3726  0.0857  0.6388  2.0967
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0643     0.3641   5.670 1.53e-07 ***
## lpolice       1.1765     0.1279   9.197 8.62e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.01 on 95 degrees of freedom
## Multiple R-squared:  0.471,Adjusted R-squared:  0.4654
## F-statistic: 84.58 on 1 and 95 DF,  p-value: 8.621e-15
```

```
plot(lcrime ~ lpolice, data = campus)
abline(m2)
```



The fitted regression now is

$$\log(crime) = 2.0643 + 1.1765 \log(police). \tag{2}$$

What is the interpretation of this regression line? This equation says that when $\log(police)$ increases by 1, the variable $\log(crime)$ increases by 1.1765. The equation (2) also has the following *percentage* interpretation: When the number of police officers goes up by 1%, then the number of crime incidences also goes up by 1.1765%. To understand why this is true, suppose that the number of police officers increases by 1% from x_{old} to x_{new}. Then

$$\frac{x_{new}}{x_{old}} - 1 = 0.01$$

Now it is a fact that $\log(x) \approx x - 1$ when x is close to one. We therefore have

$$0.01 = \frac{x_{new}}{x_{old}} - 1 \approx \log\left(\frac{x_{new}}{x_{old}}\right) = \log x_{new} - \log x_{old}.$$

Therefore when x (which is the number of police officers) increases by 1%, then $\log x$ increases approximately by 0.01. This means, using (2), that $\log y$ (where y is the number of incidents of crime) increases by 0.011765 i.e.,

$$\log\left(\frac{y_{new}}{y_{old}}\right) = 0.011765.$$

Again using $\log x \approx x - 1$, we deduce that

$$\frac{y_{new} - y_{old}}{y_{old}} \times 100 \approx 1.1765.$$

This leads to the *percentage* interpretation for (2): When the number of police officers goes up by 1%, then the number of crime incidences also goes up by 1.1765%. This

equation should not be interpreted causally; it merely reflects the fact that campuses with more police officers tend to me generally affected by more crime.

Two important things to remember about simple linear regression are:

1. Always look at the scatterplot of the data before fitting a linear regression.
2. In R, linear regression is fit using the *lm* function which has a very simple syntax.

Further questions: What do the standard errors in the *lm()* output mean? Very loosely speaking, they are supposed to provide us with some idea of the variability of the slope and intercept estimates of the least squares regression line. There are two main ways of obtaining a quantification of this variability: (i) Bootstrap, and (ii) Using Normal distribution theory.

To review the bootstrap method for obtaining standard errors, consider the following which is very similar to the bootstrap functions that Prof. Purdom described in Chapter 3. the following bootstrap function that Prof. Purdom introduced:

```
y = campus$lcrime
x = campus$lpolice
nrep = 100
bslope = rep(-999, nrep)
bint = rep(-999, nrep)
for (i in 1:nrep) {
    sampled = sample(1:length(y), size = length(y),
        replace = T)
    bslope[i] = coef(lm(y[sampled] ~ x[sampled]))[2]
    bint[i] = coef(lm(y[sampled] ~ x[sampled]))[1]
}
c(sd(bint), sd(bslope))
```

```
## [1] 0.4612715 0.1503481
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = lcrime ~ lpolice, data = campus)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0819 -0.3726  0.0857  0.6388  2.0967
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0643     0.3641   5.670 1.53e-07 ***
## lpolice       1.1765     0.1279   9.197 8.62e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.01 on 95 degrees of freedom
## Multiple R-squared:  0.471,Adjusted R-squared:  0.4654
## F-statistic: 84.58 on 1 and 95 DF,  p-value: 8.621e-15
```

3 Multiple Linear Regression

3.1 Exploratory Data Analysis of the Bodyfat Dataset

We now start our formal discussion of multiple linear regression. Let us first look at the bodyfat dataset.

```
body <- read.csv(file.path(dataDir, "bodyfat_short.csv"),
    header = T)
dim(body)
```

```
## [1] 252   8
```

```
names(body)
```

```
## [1] "BODYFAT" "AGE"     "WEIGHT"  "HEIGHT"  "CHEST"   "ABDOMEN" "HIP"
## [8] "THIGH"
```

```
head(body)
```

```
##   BODYFAT AGE WEIGHT HEIGHT CHEST ABDOMEN   HIP THIGH
## 1    12.3  23 154.25  67.75  93.1    85.2  94.5  59.0
## 2     6.1  22 173.25  72.25  93.6    83.0  98.7  58.7
## 3    25.3  22 154.00  66.25  95.8    87.9  99.2  59.6
## 4    10.4  26 184.75  72.25 101.8    86.4 101.2  60.1
## 5    28.7  24 184.25  71.25  97.3   100.0 101.9  63.2
## 6    20.9  24 210.25  74.75 104.5    94.4 107.8  66.0
```

The goal here is to understand the relationship between body fat percentage and the explanatory variables: age, height, weight, chest circumference, abdomen circumference, hip circumference and thigh circumference.

Before fitting any model to this data, let us first look at the data more carefully

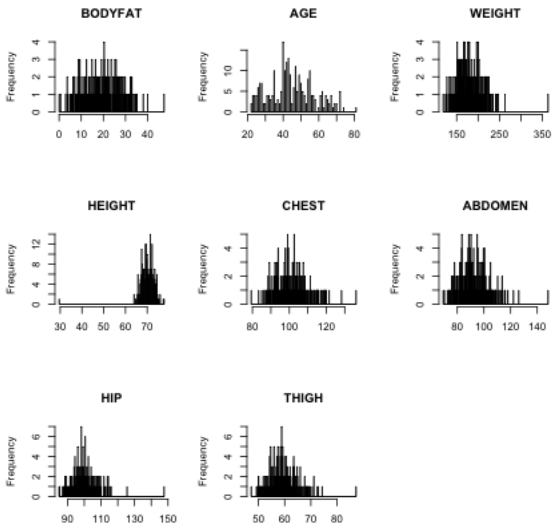
```
summary(body)
```

```
##     BODYFAT           AGE            WEIGHT          HEIGHT
##  Min.   : 0.00   Min.   :22.00   Min.   :118.5   Min.   :29.50
##  1st Qu.:12.47   1st Qu.:35.75   1st Qu.:159.0   1st Qu.:68.25
##  Median :19.20   Median :43.00   Median :176.5   Median :70.00
##  Mean   :19.15   Mean   :44.88   Mean   :178.9   Mean   :70.15
##  3rd Qu.:25.30   3rd Qu.:54.00   3rd Qu.:197.0   3rd Qu.:72.25
##  Max.   :47.50   Max.   :81.00   Max.   :363.1   Max.   :77.75
##      CHEST           ABDOMEN            HIP            THIGH
##  Min.   : 79.30   Min.   : 69.40   Min.   : 85.0   Min.   :47.20
##  1st Qu.: 94.35   1st Qu.: 84.58   1st Qu.: 95.5   1st Qu.:56.00
##  Median : 99.65   Median : 90.95   Median : 99.3   Median :59.00
##  Mean   :100.82   Mean   : 92.56   Mean   : 99.9   Mean   :59.41
##  3rd Qu.:105.38   3rd Qu.: 99.33   3rd Qu.:103.5   3rd Qu.:62.35
##  Max.   :136.20   Max.   :148.10   Max.   :147.7   Max.   :87.30
```

```
body[body$HEIGHT < 30, ]
```

```
##    BODYFAT AGE WEIGHT HEIGHT CHEST ABDOMEN   HIP THIGH
## 42    32.9  44    205   29.5   106   104.3 115.5  70.6
```

```
par(mfrow = c(3, 3))
for (i in 1:8) {
    hist(body[, i], xlab = "", main = names(body)[i],
        breaks = 500)
}
par(mfrow = c(1, 1))
```

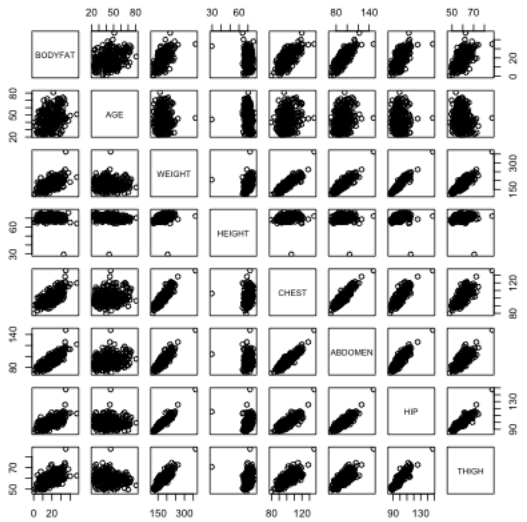



```
body[body$HIP > 140, ]
```

```
##    BODYFAT AGE WEIGHT HEIGHT CHEST ABDOMEN   HIP THIGH
## 39    35.2  46 363.15  72.25 136.2   148.1 147.7  87.3
```

We can see the pairwise relationships between the variables using the pairs plot.

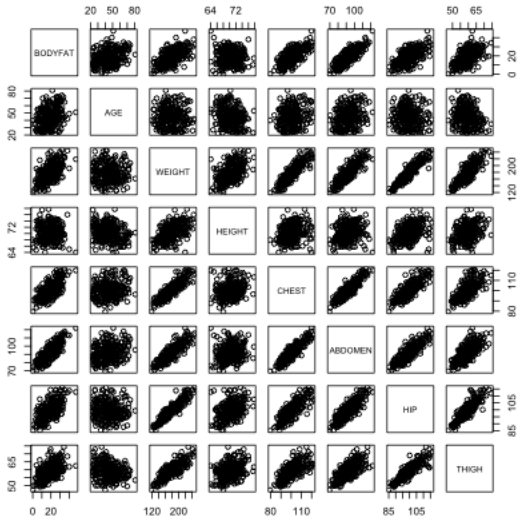
```
pairs(body)
```



There are outliers in the data and they make it hard to look at the relationships

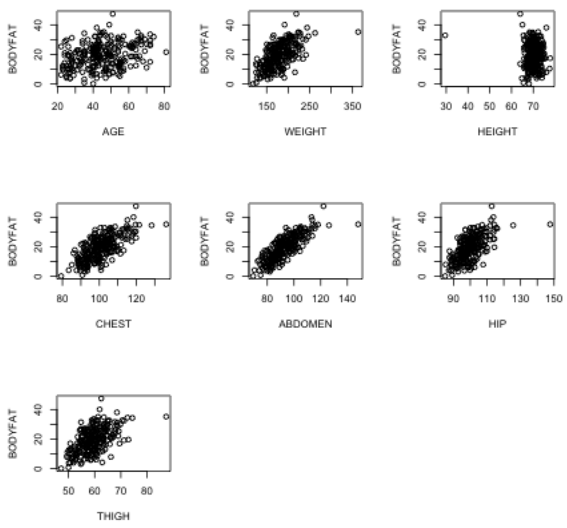
between the variables. We can try to look at the pairs plots after deleting some outlying observations.

```
ou1 = which(body$HEIGHT < 30)
ou2 = which(body$WEIGHT > 300)
ou3 = which(body$HIP > 120)
ou = c(ou1, ou2, ou3)
pairs(body[-ou, ])
```



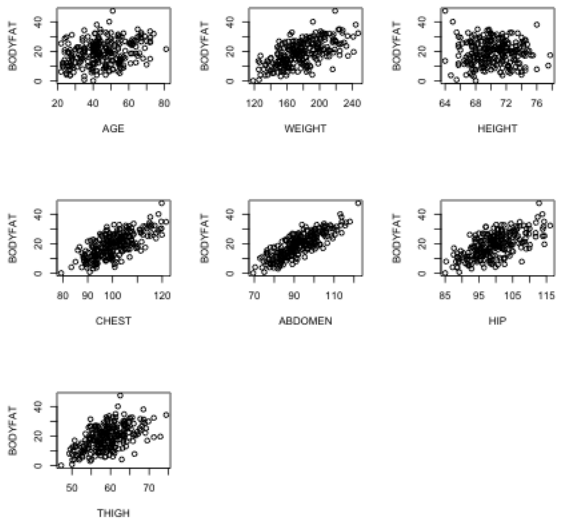
Many of the explanatory variables seem correlated with each other which is expected. Let us look at the plots between the response variable (bodyfat) and all the explanatory variables.

```
par(mfrow = c(3, 3))
for (i in 2:8) {
    plot(body[, i], body[, 1], xlab = names(body)[i],
        ylab = "BODYFAT")
}
par(mfrow = c(1, 1))
```



The same plot after removing the outliers reveals more information.

```
par(mfrow = c(3, 3))
for (i in 2:8) {
    plot(body[-ou, i], body[-ou, 1], xlab = names(body)[i],
        ylab = "BODYFAT")
}
par(mfrow = c(1, 1))
```



Most pairwise relationships seem to be linear. The clearest relationship is between bodyfat and abdomen. The next clearest is between bodyfat and chest.

3.2 Fitted regression equation via *lm()*

Let us now start the topic of multiple linear regression. Multiple linear regression is performed in R using the same function *lm()* that you have previously used for simple linear regression. Only the syntax needs to be changed slightly.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
summary(ft)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
##     HIP + THIGH, data = body)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.748e+01  1.449e+01  -2.585  0.01031 *
## AGE          1.202e-02  2.934e-02   0.410  0.68246
## WEIGHT      -1.392e-01  4.509e-02  -3.087  0.00225 **
## HEIGHT      -1.028e-01  9.787e-02  -1.051  0.29438
## CHEST       -8.312e-04  9.989e-02  -0.008  0.99337
## ABDOMEN      9.685e-01  8.531e-02  11.352  < 2e-16 ***
## HIP         -1.834e-01  1.448e-01  -1.267  0.20648
## THIGH        2.857e-01  1.362e-01   2.098  0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  0.7266,Adjusted R-squared:  0.7187
## F-statistic: 92.62 on 7 and 244 DF,  p-value: < 2.2e-16
```

Notice how similar the output to the function above is to the case of simple linear regression. The meaning of this regression output is the following. R has fit a linear equation for the variable BODYFAT in terms of the variables AGE, WEIGHT, HEIGHT, CHEST, ABDOMEN, HIP and THIGH. The specific equation that it has fit is (the following numbers are copied from the *Estimate* column in the table given

by *summary(ft)*):

$$BODYFAT = -37.48 + 0.012 * AGE - 0.139 * WEIGHT - 0.102 * HEIGHT \\ - 0.0008 * CHEST + 0.968 * ABDOMEN - 0.183 * HIP + 0.286 * THIGH \tag{3}$$

Why does R produce this equation with these particular numbers and not some other set of numbers? For example, why did R not produce an equation like:

$$BODYFAT = -30 + 0.05 * AGE - 0.013 * WEIGHT - 0.2 * HEIGHT \\ - 0.8 * CHEST + 0.0009 * ABDOMEN - 0.03 * HIP - 0.6 * THIGH$$

The reason is that the *sum of squares* will be smaller for the equation that R produced compared to my arbitrary equation. We can check this fact in R in the following way:

```
eqn = -37.48 + 0.012 * body$AGE - 0.139 * body$WEIGHT -
    0.102 * body$HEIGHT - 8e-04 * body$CHEST + 0.968 *
    body$ABDOMEN - 0.183 * body$HIP + 0.286 * body$THIGH
sum.sq = sum((body$BODYFAT - eqn)^2)
sum.sq
```

```
## [1] 4809.504
```

Now let us compute the sum of squares of my arbitrary equation:

```
my.eqn = -30 + 0.05 * body$AGE - 0.013 * body$WEIGHT -
    0.2 * body$HEIGHT - 0.8 * body$CHEST + 9e-04 *
    body$ABDOMEN - 0.03 * body$HIP + 0.6 * body$THIGH
my.sum.sq = sum((body$BODYFAT - my.eqn)^2)
my.sum.sq
```

```
## [1] 3153233
```

Clearly the sum of squares for R's regression equation is almost 650 times smaller than my arbitrary equation.

The main fact here is that **The sum of squares of R's regression equation will be smaller than any equation that you can come up with**. In other words, **R outputs the linear equation which has the smallest possible sum of squares**. This is the reason why R's equation is also called the **least squares equation**.

You will need some more math to understand how exactly the *lm* function is able to find the least squares equation. We will skip this part in this course. It is enough for our purposes to simply know that the linear equation outputted by R is indeed the least squares equation.

3.3 Interpretation of the fitted regression equation

Consider a fitted regression equation:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p. \tag{4}$$

Here the coefficient b_1 is interpreted as the average increase in y for unit increase in x_1 provided all other explanatory variables $x_2, \ldots, x_p$ are kept constant. More generally for $j \geq 1$, the coefficient b_j is interpreted as the average increase in y for unit increase in x_j provided all other explanatory variables x_k for $k \neq j$ are kept constant. The intercept b_0 is interpreted as the average value of y when all the explanatory variables are equal to zero.

In the body fat example, the fitted regression equation as we have seen is:

$$\begin{aligned} BODYFAT = -37.48 + 0.012 * AGE - 0.139 * WEIGHT - 0.102 * HEIGHT \\ - 0.0008 * CHEST + 0.968 * ABDOMEN - 0.183 * HIP + 0.286 * THIGH \end{aligned} \tag{5}$$

The coefficient of 0.968 can be interpreted as the average percentage increase in bodyfat percentage per unit (i.e., 1 cm) increase in Abdomen circumference provided all the other explanatory variables age, weight, height, chest circumference, hip circumference and thigh circumference are kept unchanged.

Do the signs of the fitted regression coefficients in (7) make sense?

The interpretation of the coefficient b_j in (4) depends crucially on the other explanatory variables $x_k, k \neq j$ that are present in the equation (this is because of the phrase "all other explanatory variables kept constant").

For the bodyfat data, we have seen that the variables chest circumference and abdomen circumference are highly correlated:

```
cor(body$CHEST, body$ABDOMEN)
```

```
## [1] 0.9158277
```

This effectively means that these two variables are measuring essentially the same thing and, therefore, it might make more sense to just have one of these two variables in the regression equation. Similarly the variables hip circumference and thigh circumference are also highly correlated. Let us therefore fit a linear model for the body fat percentage based on age, weight, height, chest and thigh:

```
ft1 = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST +
    THIGH, data = body)
summary(ft1)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + THIGH,
##     data = body)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4106  -3.8409  -0.1898   3.6800  15.0222
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -45.22628   13.79574  -3.278 0.001195 **
## AGE           0.15899    0.03271   4.860 2.09e-06 ***
## WEIGHT       -0.02991    0.04384  -0.682 0.495714
## HEIGHT       -0.31266    0.11466  -2.727 0.006854 **
## CHEST         0.52668    0.10763   4.893 1.79e-06 ***
## THIGH         0.52895    0.15701   3.369 0.000876 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.503 on 246 degrees of freedom
## Multiple R-squared:  0.5762,Adjusted R-squared:  0.5676
## F-statistic: 66.89 on 5 and 246 DF,  p-value: < 2.2e-16
```

The regression equation now is

$$BODYFAT = -45.226 + 0.159 * AGE - 0.03 * WEIGHT - 0.313 * HEIGHT + 0.527 * CHEST + 0.529 * THIGH \quad (6)$$

See now that the regression equation is quite different from the previous one. The coefficients are different now (and they have different interpretations as well).

3.4 Regression Line vs Regression Plane

In simple linear regression (when there is only one explanatory variable), the fitted regression equation describes a line. In multiple linear regression, there are multiple explanatory variables, and in this case, the fitted regression equation describes a plane called the fitted regression plane.

This plane can be plotted in a 3D plot when there are two explanatory variables. When the number of explanatory variables is 3 or more, we cannot plot this plane.

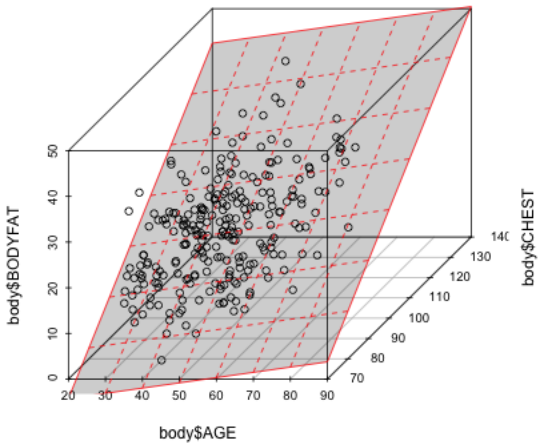
To illustrate this, let us fit a regression equation to bodyfat percentage in terms of age and chest circumference:

```
ft2 = lm(BODYFAT ~ AGE + CHEST, data = body)
summary(ft2)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + CHEST, data = body)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4657  -4.3271   0.1406   3.9607  14.9866
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53.27135    4.43061 -12.023  < 2e-16 ***
## AGE           0.11479    0.02954   3.886 0.000131 ***
## CHEST         0.66720    0.04416  15.109  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.805 on 249 degrees of freedom
## Multiple R-squared:  0.5226,Adjusted R-squared:  0.5188
## F-statistic: 136.3 on 2 and 249 DF,  p-value: < 2.2e-16
```

The fitted regression plane can be plotted in a 3D plot as follows:

```
library(scatterplot3d)
sp = scatterplot3d(body$AGE, body$CHEST, body$BODYFAT)
sp$plane3d(ft2, lty.box = "solid", draw_lines = TRUE,
    draw_polygon = TRUE, col = "red")
```



3.5 Fitted Values and Multiple R^2

Any regression equation can be used to predict the value of the response variable given values of the explanatory variables. For example, consider the fitted regression equation obtained by applying $lm()$ with bodyfat percentage as the response and age, weight, height, chest circumference, abdomen circumference, hip circumference and thigh circumference as the explanatory variables:

$$BODYFAT = -37.48 + 0.01202 * AGE - 0.1392 * WEIGHT - 0.1028 * HEIGHT \quad (7)$$
$$- 0.0008312 * CHEST + 0.9685 * ABDOMEN - 0.1834 * HIP + 0.2857 * THIGH$$

Suppose a person X (who is of 30 years of age, weighs 180 pounds and is 70 inches tall) wants to find out his bodyfat percentage. Let us say that he is able to measure his chest circumference as 90 cm, abdomen circumference as 86 cm, hip circumference as 97 cm and thigh circumference as 60 cm. Then he can simply use the regression equation to predict his bodyfat percentage as:

```
bf.pred = -37.48 + 0.01202 * 30 - 0.1392 * 180 - 0.1028 *
    70 - 0.0008312 * 90 + 0.9685 * 86 - 0.1834 * 97 +
    0.2857 * 60
bf.pred
```

```
## [1] 13.19699
```

The predictions given by the fitted regression equation for each of the observations are known as *fitted values*. For example, in the bodyfat dataset, the first observation (first row) is given by:

```
obs1 = body[1, ]
obs1
```

```
##   BODYFAT AGE WEIGHT HEIGHT CHEST ABDOMEN  HIP THIGH
## 1    12.3  23 154.25  67.75  93.1    85.2 94.5    59
```

The observed value of the response (bodyfat percentage) for this individual is 12.3 %. The prediction for this person's response given by the regression equation (7) is

```
-37.48 + 0.01202 * body[1, "AGE"] - 0.1392 * body[1,
    "WEIGHT"] - 0.1028 * body[1, "HEIGHT"] - 0.0008312 *
    body[1, "CHEST"] + 0.9685 * body[1, "ABDOMEN"] -
    0.1834 * body[1, "HIP"] + 0.2857 * body[1, "THIGH"]
```

```
## [1] 16.32398
```

Therefore the *fitted value* for the first observation is 16.424%. R directly calculates all fitted values and they are stored in the $lm()$ object. You can obtain these via:

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
names(ft)
```

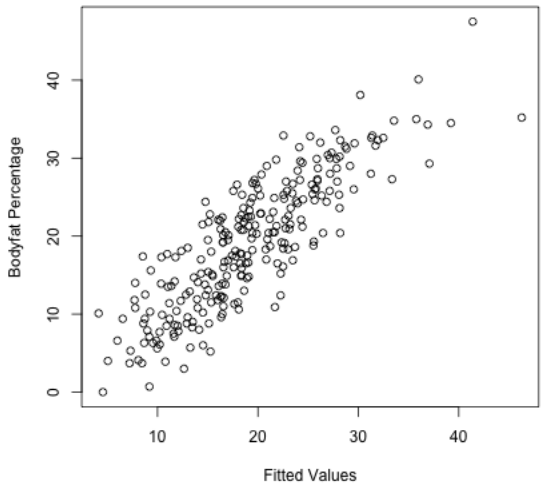
```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
head(ft$fitted.values)
```

```
##        1        2        3        4        5        6
## 16.32670 10.22019 18.42600 11.89502 25.97564 16.28529
```

If the regression equation fits the data well, we would expect the fitted values to be close to the observed responses. We can check this by just plotting the fitted values against the observed response values.

```
plot(ft$fitted.values, body$BODYFAT, xlab = "Fitted Values",
    ylab = "Bodyfat Percentage")
```



The square of the correlation between the obseved response values and the fitted values obtained by the regression equation is an important and widely used measure of the effectiveness of the regression equation. This squared correlation is known as the **Coefficient of Determination** or **Multiple** R^2 or simply R^2:

$$R^2 = (\text{correlation(response, fitted values)})^2 .$$

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
Rsquared = (cor(body$BODYFAT, ft$fitted.values))^2
Rsquared
```

```
## [1] 0.7265596
```

The value of R^2 is so standard that it is included in the summary for the $lm()$ function.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
summary(ft)
```

```
##
```

```
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
##     HIP + THIGH, data = body)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.748e+01  1.449e+01  -2.585  0.01031 *
## AGE          1.202e-02  2.934e-02   0.410  0.68246
## WEIGHT      -1.392e-01  4.509e-02  -3.087  0.00225 **
## HEIGHT      -1.028e-01  9.787e-02  -1.051  0.29438
## CHEST       -8.312e-04  9.989e-02  -0.008  0.99337
## ABDOMEN      9.685e-01  8.531e-02  11.352  < 2e-16 ***
## HIP         -1.834e-01  1.448e-01  -1.267  0.20648
## THIGH        2.857e-01  1.362e-01   2.098  0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  0.7266,Adjusted R-squared:  0.7187
## F-statistic: 92.62 on 7 and 244 DF,  p-value: < 2.2e-16
```

A high value of R^2 means that the fitted values (given by the fitted regression equation) are close to the observed values and hence indicates that the regression equation fits the data well. A low value, on the other hand, means that the fitted values are far from the observed values and hence the regression line does not fit the data well.

Note that R^2 has no units. In other words, it is scale-free.

3.6 Residuals and Residual Sum of Squares (RSS)

For every point in the scatter plot, its distance to the *corresponding* point on the regression plane is called the *residual*. It can also be defined as

$$\text{residual} = \text{response} - \text{fitted value}$$

For example, for the first observation (row) in the bodyfat dataset, the residual can be calculated as:

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
res.1 = body[1, "BODYFAT"] - ft$fitted.values[1]
res.1
```

```
##         1
## -4.026695
```

Residuals are again so important that $lm()$ automatically calculates them for us.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
names(ft)
```

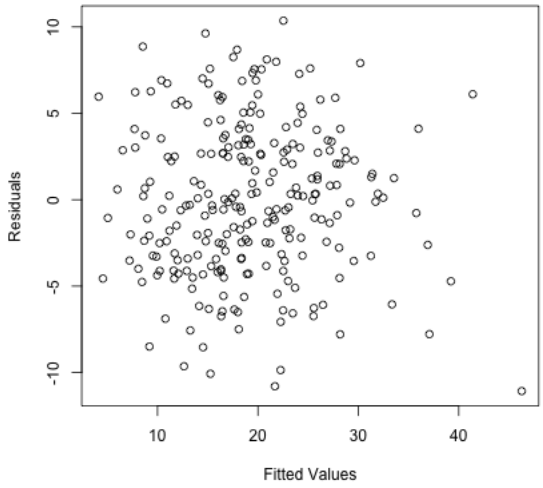
```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
head(ft$residuals)
```

```
##         1         2         3         4         5         6
## -4.026695 -4.120189  6.874004 -1.495017  2.724355  4.614712
```

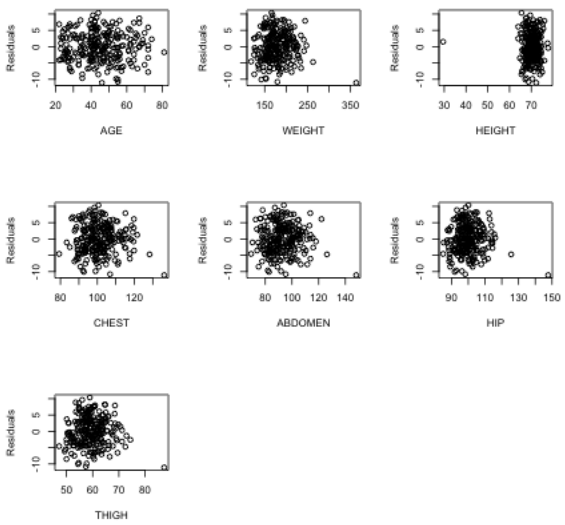
A common way of looking at the residuals is to plot them against the fitted values.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
plot(ft$fitted.values, ft$residuals, xlab = "Fitted Values",
    ylab = "Residuals")
```



One can also plot the residuals against each of the explanatory variables.

```
par(mfrow = c(3, 3))
for (i in 2:8) {
    plot(body[, i], ft$residuals, xlab = names(body)[i],
        ylab = "Residuals")
}
par(mfrow = c(1, 1))
```



The residuals represent what is left in the response (y) after all the linear effects of the explanatory variables are taken out. One consequence of this is that the residuals

are **uncorrelated with every explanatory variable**. We can check this in easily in the body fat example.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
cor(ft$residuals, body$AGE)
```

```
## [1] -1.754044e-17
```

```
cor(ft$residuals, body$WEIGHT)
```

```
## [1] 4.71057e-17
```

```
cor(ft$residuals, body$HEIGHT)
```

```
## [1] -1.720483e-15
```

```
cor(ft$residuals, body$CHEST)
```

```
## [1] -4.672628e-16
```

```
cor(ft$residuals, body$ABDOMEN)
```

```
## [1] -7.012368e-16
```

```
cor(ft$residuals, body$HIP)
```

```
## [1] -8.493675e-16
```

```
cor(ft$residuals, body$THIGH)
```

```
## [1] -5.509094e-16
```

Moreover, the residuals always have mean zero:

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
mean(ft$residuals)
```

```
## [1] 2.467747e-16
```

Also, if one were to fit a regression equation to the residuals in terms of the same explanatory variables, then the fitted regression equation will have all coefficients exactly equal to zero:

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
m.res = lm(ft$residuals ~ body$AGE + body$WEIGHT +
    body$HEIGHT + body$CHEST + body$ABDOMEN + body$HIP +
    body$THIGH)
summary(m.res)
```

```
##
## Call:
## lm(formula = ft$residuals ~ body$AGE + body$WEIGHT + body$HEIGHT +
##     body$CHEST + body$ABDOMEN + body$HIP + body$THIGH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.154e-14  1.449e+01       0        1
## body$AGE      1.282e-17  2.934e-02       0        1
## body$WEIGHT   1.057e-16  4.509e-02       0        1
## body$HEIGHT  -1.509e-16  9.787e-02       0        1
## body$CHEST    1.180e-16  9.989e-02       0        1
## body$ABDOMEN -2.452e-16  8.531e-02       0        1
## body$HIP     -1.284e-16  1.448e-01       0        1
## body$THIGH   -1.090e-16  1.362e-01       0        1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  6.384e-32,Adjusted R-squared:  -0.02869
## F-statistic: 2.225e-30 on 7 and 244 DF,  p-value: 1
```

If the regression equation fits the data well, the residuals are supposed to be small. One popular way of assessing the size of the residuals is to compute their sum of squares. This quantity is called the **Residual Sum of Squares (RSS)**.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
rss.ft = sum((ft$residuals)^2)
rss.ft
```

```
## [1] 4806.806
```

Note that RSS depends on the units in which the response variable is measured.

There is a very simple relationship between RSS and R^2 (recall from the last lecture that R^2 is the square of the correlation between the response values and the fitted values):

$$R^2 = 1 - \frac{RSS}{TSS}$$

where TSS stands for Total Sum of Squares and is defined as

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2 .$$

It is easy to verify this formula in R.

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
rss.ft = sum((ft$residuals)^2)
rss.ft
```

```
## [1] 4806.806
```

```
tss = sum(((body$BODYFAT) - mean(body$BODYFAT))^2)
1 - (rss.ft/tss)
```

```
## [1] 0.7265596
```

```
summary(ft)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
##     HIP + THIGH, data = body)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.748e+01  1.449e+01  -2.585  0.01031 *
## AGE          1.202e-02  2.934e-02   0.410  0.68246
## WEIGHT      -1.392e-01  4.509e-02  -3.087  0.00225 **
## HEIGHT      -1.028e-01  9.787e-02  -1.051  0.29438
## CHEST       -8.312e-04  9.989e-02  -0.008  0.99337
## ABDOMEN      9.685e-01  8.531e-02  11.352  < 2e-16 ***
## HIP         -1.834e-01  1.448e-01  -1.267  0.20648
## THIGH        2.857e-01  1.362e-01   2.098  0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  0.7266,Adjusted R-squared:  0.7187
## F-statistic: 92.62 on 7 and 244 DF,  p-value: < 2.2e-16
```

If we did not have any explanatory variables, then we would predict the value of bodyfat percentage for any individual by simply the mean of the bodyfat values in our sample. The total squared error for this prediction is given by TSS. On the other hand, the total squared error for the prediction using linear regression based on the explanatory variables is given by RSS. Therefore $1 - R^2$ represents the reduction in the squared error because of the explanatory variables.

3.7 Behaviour of RSS (and R^2) when variables are added or removed from the regression equation

The value of RSS always increases when one or more explanatory variables are removed from the regression equation. For example, suppose that we remove the variable abdomen circumeference from the regression equation. The new RSS will then be:

```
ft.1 = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST +
    HIP + THIGH, data = body)
rss.ft1 = sum((ft.1$residuals)^2)
rss.ft1
```

```
## [1] 7345.724
```

```
rss.ft
```

```
## [1] 4806.806
```

Notice that there is a quite a lot of increase in the RSS. What if we had kept ABDOMEN in the model but dropped the variable CHEST?

```
ft.2 = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + ABDOMEN +
    HIP + THIGH, data = body)
rss.ft2 = sum((ft.2$residuals)^2)
rss.ft2
```

```
## [1] 4806.808
```

```
rss.ft
```

```
## [1] 4806.806
```

The RSS again increases but by a very very small amount. This therefore suggests that Abdomen circumference is a more important variable in this regression compared to Chest circumference.

The moral of this exercise is the following. The RSS always increases when variables are dropped from the regression equation. However the amount of increase varies for different variables. We can understand the importance of variables in a multiple regression equation by noting the amount by which the RSS increases when the individual variables are dropped. We will come back to this point while studying inference in the multiple regression model.

Because RSS has a direct relation to R^2 via $R^2 = 1 - (RSS/TSS)$, one can see R^2 decreases when variables are removed from the model. However the amount of

decrease will be different for different variables. For example, in the body fat dataset, after removing the abdomen circumference variable, R^2 changes to:

```
ft.1 = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST +
    HIP + THIGH, data = body)
R2.ft1 = (cor(body$BODYFAT, ft.1$fitted.values))^2
R2.ft1
```

```
## [1] 0.5821305
```

```
R2.ft = (cor(body$BODYFAT, ft$fitted.values))^2
R2.ft
```

```
## [1] 0.7265596
```

Notice that there is a lot of decrease in R^2. What happens if the variable Chest circumference is dropped.

```
ft.2 = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + ABDOMEN +
    HIP + THIGH, data = body)
R2.ft2 = (cor(body$BODYFAT, ft.2$fitted.values))^2
R2.ft2
```

```
## [1] 0.7265595
```

```
R2.ft
```

```
## [1] 0.7265596
```

There is now a very very small decrease.

3.8 Residual Degrees of Freedom and Residual Standard Error

In a regression with p explanatory variables, the residual degrees of freedom is given by $n-p-1$ (recall that n is the number of observations). This can be thought of as the

effective number of residuals. Even though there are n residuals, they are supposed to satisfy $p+1$ exact equations (they sum to zero and they have zero correlation with each of the p explanatory variables).

The Residual Standard Error is defined as:

$$\sqrt{\frac{\text{Residual Sum of Squares}}{\text{Residual Degrees of Freedom}}}$$

This can be interpreted as the average magnitude of an individual residual and can be used to assess the sizes of residuals (in particular, to find identify large residual values).

For illustration,

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
n = nrow(body)
p = 7
rs.df = n - p - 1
rs.df
```

```
## [1] 244
```

```
ft = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
rss = sum((ft$residuals)^2)
rse = sqrt(rss/rs.df)
rse
```

```
## [1] 4.438471
```

Both of these are printed in the summary function in R:

```
summary(ft)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
##     HIP + THIGH, data = body)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.748e+01  1.449e+01  -2.585  0.01031 *
## AGE          1.202e-02  2.934e-02   0.410  0.68246
## WEIGHT      -1.392e-01  4.509e-02  -3.087  0.00225 **
## HEIGHT      -1.028e-01  9.787e-02  -1.051  0.29438
## CHEST       -8.312e-04  9.989e-02  -0.008  0.99337
## ABDOMEN      9.685e-01  8.531e-02  11.352  < 2e-16 ***
## HIP         -1.834e-01  1.448e-01  -1.267  0.20648
## THIGH        2.857e-01  1.362e-01   2.098  0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  0.7266,Adjusted R-squared:  0.7187
## F-statistic: 92.62 on 7 and 244 DF,  p-value: < 2.2e-16
```

4 Multiple Regression when some explanatory variables are categorical

In many instances of regression, some of the explanatory variables are categorical (note that the response variable is always continuous). For example, consider the (short version of the) *college* dataset that you have already encountered.

```
scorecard <- read.csv(file.path(dataDir, "college_short.csv"))
dim(scorecard)
```

```
## [1] 1241    9
```

```
names(scorecard)
```

```
## [1] "SAT_AVG_ALL"    "AVGFACSAL"      "TUITIONFEE_IN"  "TUITIONFEE_OUT"
## [5] "UGDS"           "RET_FT4"        "PCTFLOAN"       "PFTFAC"
## [9] "TYPE"
```

We can do a regression here with the retention rate (variable name `RET-FT4`) as the response and all other variables as the explanatory variables. Note that one of the explanatory variables (variable name `TYPE`) is categorical. This variable represents whether the college is public (1), private non-profit (2) or private for profit (3). Dealing with such categorical variables is a little tricky. To illustrate the ideas here, let us focus on a regression for the retention rate based on just two explanatory variables: the out-of-state tuition and the categorical variable `TYPE`.

The important thing to note about the variable `TYPE` is that its *levels* 1, 2 and 3 are completely arbitrary and have no particular meaning. For example, we could have called its levels A, B, C or Pu, $Pr-np$, $Pr-fp$ as well. If we use the $lm()$ function in the usual way with `TUITIONFEE` and `TYPE` as the explanatory variables, then R will treat `TYPE` as a continuous variable which does not make sense:

```
req.bad = lm(RET_FT4 ~ TUITIONFEE_OUT + TYPE, data = scorecard)
summary(req.bad)
```

```
##
## Call:
## lm(formula = RET_FT4 ~ TUITIONFEE_OUT + TYPE, data = scorecard)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69041 -0.04915  0.00516  0.05554  0.33165
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.661e-01  9.265e-03   71.90   <2e-16 ***
## TUITIONFEE_OUT  9.405e-06  3.022e-07   31.12   <2e-16 ***
## TYPE           -8.898e-02  5.741e-03  -15.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08741 on 1238 degrees of freedom
## Multiple R-squared:  0.4391,Adjusted R-squared:  0.4382
## F-statistic: 484.5 on 2 and 1238 DF,  p-value: < 2.2e-16
```

The regression coefficient for `TYPE` has the usual interpretation (if `TYPE` increases by one unit, ...) which does not make much sense because `TYPE` is categorical and so increasing it by one unit is nonsensical. You can check that R is treating `TYPE` as a numeric variable by:

```
is.numeric(scorecard$TYPE)
```

```
## [1] TRUE
```

The correct way to deal with categorical variables in R is to treat them as factors:

```
req = lm(RET_FT4 ~ TUITIONFEE_OUT + as.factor(TYPE),
    data = scorecard)
summary(req)
```

```
##
## Call:
## lm(formula = RET_FT4 ~ TUITIONFEE_OUT + as.factor(TYPE), data = scorecard)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68856 -0.04910  0.00505  0.05568  0.33150
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.765e-01  7.257e-03  79.434  < 2e-16 ***
## TUITIONFEE_OUT    9.494e-06  3.054e-07  31.090  < 2e-16 ***
## as.factor(TYPE)2 -9.204e-02  5.948e-03 -15.474  < 2e-16 ***
## as.factor(TYPE)3 -1.218e-01  3.116e-02  -3.909 9.75e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08732 on 1237 degrees of freedom
## Multiple R-squared:  0.4408,Adjusted R-squared:  0.4394
## F-statistic:   325 on 3 and 1237 DF,  p-value: < 2.2e-16
```

Why is the coefficient of TUITIONFEE so small?

This equation can be written in full as:

$$RET = 0.5765 + 9.4 \times 10^{-6} * TUITIONFEE - 0.0092 * I\left(TYPE = 2\right) - 0.1218 * I\left(TYPE = 3\right). \tag{8}$$

The variable $I\left(TYPE = 2\right)$ takes the value 1 if the college has `TYPE` equal to 2 (i.e., if the college is private non-profit) and 0 otherwise. Similarly the variable $I\left(TYPE = 3\right)$ takes the value 1 if the college has `TYPE` equal to 3 (i.e., if the college

is private for profit) and 0 otherwise. Variables which take only the two values 0 and 1 are called indicator variables.

Note that the variable $I\,(TYPE = 1)$ does not appear in the regression equation (8). This means that the level 1 (i.e., the college is public) is the baseline level here and the effects of -0.0092 and 0.1218 for private for-profit and private non-profit colleges respectively should be interpreted relative to public colleges.

The regression equation (8) can effectively be broken down into three equations. For public colleges, the two indicator variables in (8) are zero and the equation becomes:

$$RET = 0.5765 + 9.4 \times 10^{-6} * TUITIONFEE. \tag{9}$$

For private non-profit colleges, the equation becomes

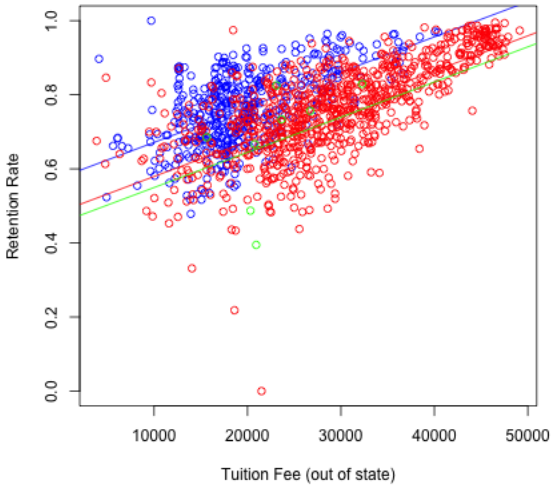
$$RET = 0.5673 + 9.4 \times 10^{-6} * TUITIONFEE. \tag{10}$$

and for private for-profit colleges,

$$RET = 0.4547 + 9.4 \times 10^{-6} * TUITIONFEE. \tag{11}$$

Note that the coefficient of TUITIONFEE is the same in each of these equations (only the intercept changes). We can plot a scatterplot together with all these lines.

```
plot(RET_FT4 ~ TUITIONFEE_OUT, data = scorecard, xlab = "Tuition Fee (out of state)",
    ylab = "Retention Rate", type = "n")
w1 = (scorecard$TYPE == 1)
points(RET_FT4[w1] ~ TUITIONFEE_OUT[w1], data = scorecard,
    col = "blue")
abline(c(req$coefficients[1], req$coefficients[2]),
    col = "blue")
w2 = (scorecard$TYPE == 2)
points(RET_FT4[w2] ~ TUITIONFEE_OUT[w2], data = scorecard,
    col = "red")
abline(c(req$coefficients[1] + req$coefficients[3],
    req$coefficients[2]), col = "red")
w3 = (scorecard$TYPE == 3)
points(RET_FT4[w3] ~ TUITIONFEE_OUT[w3], data = scorecard,
    col = "green")
abline(c(req$coefficients[1] + req$coefficients[4],
    req$coefficients[2]), col = "green")
```



What if we want these regression equations to have different slopes? We can do separate regressions for each of the three groups given by the `TYPE` variable. Alternatively, we can do this in multiple regression by adding an interaction variable between `TYPE` and `TUITIONFEE` as follows:

```
req.1 = lm(RET_FT4 ~ TUITIONFEE_OUT + as.factor(TYPE) +
    TUITIONFEE_OUT:as.factor(TYPE), data = scorecard)
summary(req.1)
```

```
##
## Call:
## lm(formula = RET_FT4 ~ TUITIONFEE_OUT + as.factor(TYPE) + TUITIONFEE_OUT:as.factor
##     data = scorecard)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68822 -0.04982  0.00491  0.05555  0.32900
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      5.814e-01  1.405e-02  41.372  < 2e-16 ***
## TUITIONFEE_OUT                   9.240e-06  6.874e-07  13.441  < 2e-16 ***
## as.factor(TYPE)2                -9.830e-02  1.750e-02  -5.617  2.4e-08 ***
## as.factor(TYPE)3                -2.863e-01  1.568e-01  -1.826   0.0681 .
## TUITIONFEE_OUT:as.factor(TYPE)2  2.988e-07  7.676e-07   0.389   0.6971
## TUITIONFEE_OUT:as.factor(TYPE)3  7.215e-06  6.716e-06   1.074   0.2829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.08734 on 1235 degrees of freedom
## Multiple R-squared:  0.4413,Adjusted R-squared:  0.4391
## F-statistic: 195.1 on 5 and 1235 DF,  p-value: < 2.2e-16
```

Note that this regression equation has two more variables compared to the previous regression (which did not have the interaction term). The two additional variables are the product terms $TUITIONFEE * I(TYPE = 2)$ and $TUITIONFEE * I(TYPE = 3)$. The presence of these product terms means that three separate regressions are essentially being fit here (why?).

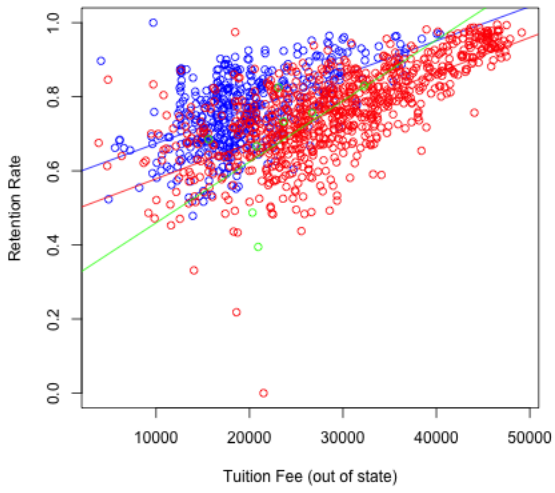
Alternatively, this regression with interaction can also be done in R via:

```
req.2 = lm(RET_FT4 ~ TUITIONFEE_OUT * as.factor(TYPE),
    data = scorecard)
summary(req.2)
```

```
##
## Call:
## lm(formula = RET_FT4 ~ TUITIONFEE_OUT * as.factor(TYPE), data = scorecard)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68822 -0.04982  0.00491  0.05555  0.32900
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       5.814e-01  1.405e-02  41.372  < 2e-16 ***
## TUITIONFEE_OUT                    9.240e-06  6.874e-07  13.441  < 2e-16 ***
## as.factor(TYPE)2                 -9.830e-02  1.750e-02  -5.617  2.4e-08 ***
## as.factor(TYPE)3                 -2.863e-01  1.568e-01  -1.826   0.0681 .
## TUITIONFEE_OUT:as.factor(TYPE)2   2.988e-07  7.676e-07   0.389   0.6971
## TUITIONFEE_OUT:as.factor(TYPE)3   7.215e-06  6.716e-06   1.074   0.2829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08734 on 1235 degrees of freedom
## Multiple R-squared:  0.4413,Adjusted R-squared:  0.4391
## F-statistic: 195.1 on 5 and 1235 DF,  p-value: < 2.2e-16
```

The three separate regressions can be plotted in one plot as before.

```
plot(RET_FT4 ~ TUITIONFEE_OUT, data = scorecard, xlab = "Tuition Fee (out of state)",
    ylab = "Retention Rate", type = "n")
w1 = (scorecard$TYPE == 1)
points(RET_FT4[w1] ~ TUITIONFEE_OUT[w1], data = scorecard,
    col = "blue")
abline(c(req.1$coefficients[1], req.1$coefficients[2]),
    col = "blue")
w2 = (scorecard$TYPE == 2)
points(RET_FT4[w2] ~ TUITIONFEE_OUT[w2], data = scorecard,
    col = "red")
abline(c(req.1$coefficients[1] + req.1$coefficients[3],
    req.1$coefficients[2] + req.1$coefficients[5]),
    col = "red")
w3 = (scorecard$TYPE == 3)
points(RET_FT4[w3] ~ TUITIONFEE_OUT[w3], data = scorecard,
    col = "green")
abline(c(req.1$coefficients[1] + req.1$coefficients[4],
    req.1$coefficients[2] + req.1$coefficients[6]),
    col = "green")
```



Interaction terms make regression equations complicated (have more variables) and also slightly harder to interpret although, in some situations, they really improve predictive power. In this particular example, note that the multiple R^2 only increased from 0.4408 to 0.4413 after adding the interaction terms. This small increase means that the interaction terms are not really adding much to the regression equation so we are better off using the previous model with no interaction terms.

To get more practice with regressions having categorical variables, let us consider the bike sharing dataset. This dataset contains information on bike rentals for two years (2011-2012) from Capital Bikeshare System in Washington, D. C:

```
bike = read.csv(file.path(dataDir, "BikeSharingDataset.csv"))
dim(bike)
```

```
## [1] 17379    17
```

```
names(bike)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"
##  [6] "hr"         "holiday"    "weekday"    "workingday" "weathersit"
## [11] "temp"       "atemp"      "hum"        "windspeed"  "casual"
## [16] "registered" "cnt"
```

```
head(bike)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday
## 1       1 2011-01-01      1  0    1  0       0       6          0
## 2       2 2011-01-01      1  0    1  1       0       6          0
## 3       3 2011-01-01      1  0    1  2       0       6          0
## 4       4 2011-01-01      1  0    1  3       0       6          0
## 5       5 2011-01-01      1  0    1  4       0       6          0
## 6       6 2011-01-01      1  0    1  5       0       6          0
##   weathersit temp  atemp  hum windspeed casual registered cnt
## 1          1 0.24 0.2879 0.81    0.0000      3         13  16
## 2          1 0.22 0.2727 0.80    0.0000      8         32  40
## 3          1 0.22 0.2727 0.80    0.0000      5         27  32
## 4          1 0.24 0.2879 0.75    0.0000      3         10  13
## 5          1 0.24 0.2879 0.75    0.0000      0          1   1
## 6          2 0.24 0.2576 0.75    0.0896      0          1   1
```

Let us fit a basic regression equation with `casual` (number of bikes rented by casual users hourly) as the response variable and the explanatory variables being `atemp` (normalized feeling temperature), `workingday` (takes the value 1 if the day is neither weekend or a holiday and 0 otherwise) and `weathersit`. The `weathersit` variable takes four values:

1. 1 if the weather is Clear, Few clouds, Partly cloudy, Partly cloudy.

2. 2 if the weather is Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist.
3. 3 if the weather is Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds.
4. 4 if the weather is Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog.

```
summary(bike$atemp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3333  0.4848  0.4758  0.6212  1.0000
```

```
summary(as.factor(bike$workingday))
```

```
##     0     1
##  5514 11865
```

```
summary(as.factor(bike$weathersit))
```

```
##     1     2     3     4
## 11413  4544  1419     3
```

Note that there are only 3 observations where the `weathersit` variable takes the value 4. Because `workingday` and `weathersit` are categorical variables, we fit the regression equation as

```
md1 = lm(casual ~ atemp + as.factor(workingday) + as.factor(weathersit),
    data = bike)
summary(md1)
```

```
##
## Call:
## lm(formula = casual ~ atemp + as.factor(workingday) + as.factor(weathersit),
##     data = bike)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -100.104  -24.209   -3.655   14.379  292.104
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.4416     1.0366  -1.391    0.164
## atemp                    132.7107     1.8076  73.418  < 2e-16 ***
## as.factor(workingday)1 -34.1359     0.6642 -51.393  < 2e-16 ***
## as.factor(weathersit)2  -5.5858     0.7156  -7.805 6.27e-15 ***
## as.factor(weathersit)3 -15.3972     1.1488 -13.403  < 2e-16 ***
## as.factor(weathersit)4   2.0619    23.4727   0.088    0.930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.64 on 17373 degrees of freedom
## Multiple R-squared:  0.3208,Adjusted R-squared:  0.3206
## F-statistic:  1641 on 5 and 17373 DF,  p-value: < 2.2e-16
```

How are the coefficients in the above regression interpreted?

Now let me add an interaction between the two categorical variables above. This can be done as:

```
md2 = lm(casual ~ atemp + as.factor(workingday) + as.factor(weathersit) +
    as.factor(workingday):as.factor(weathersit), data = bike)
summary(md2)
```

```
##
## Call:
## lm(formula = casual ~ atemp + as.factor(workingday) + as.factor(weathersit) +
##     as.factor(workingday):as.factor(weathersit), data = bike)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -102.67  -24.25   -3.42   14.37  289.49
##
## Coefficients:
##                                                Estimate Std. Error t value
## (Intercept)                                      1.2457     1.0820   1.151
## atemp                                          132.5508     1.8039  73.480
## as.factor(workingday)1                         -38.0378     0.8073 -47.119
## as.factor(weathersit)2                         -12.8922     1.2866 -10.021
## as.factor(weathersit)3                         -27.2829     2.1876 -12.471
```

```
## as.factor(weathersit)4                          -18.3256    40.5639  -0.452
## as.factor(workingday)1:as.factor(weathersit)2    10.5834     1.5432   6.858
## as.factor(workingday)1:as.factor(weathersit)3    16.5474     2.5635   6.455
## as.factor(workingday)1:as.factor(weathersit)4    30.4970    49.6749   0.614
##                                                 Pr(>|t|)
## (Intercept)                                        0.250
## atemp                                            < 2e-16 ***
## as.factor(workingday)1                           < 2e-16 ***
## as.factor(weathersit)2                           < 2e-16 ***
## as.factor(weathersit)3                           < 2e-16 ***
## as.factor(weathersit)4                             0.651
## as.factor(workingday)1:as.factor(weathersit)2 7.22e-12 ***
## as.factor(workingday)1:as.factor(weathersit)3 1.11e-10 ***
## as.factor(workingday)1:as.factor(weathersit)4    0.539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.55 on 17370 degrees of freedom
## Multiple R-squared:  0.3238,Adjusted R-squared:  0.3235
## F-statistic:  1040 on 8 and 17370 DF,  p-value: < 2.2e-16
```

How are the coefficients interpreted now? Note that the multiple R^2 has not increased by that much. There are other interactons that one can add here too. For example, I can add an interaction between `workingday` and `atemp`:

```
md3 = lm(casual ~ atemp + as.factor(workingday) + as.factor(workingday):atemp +
    as.factor(weathersit), data = bike)
summary(md3)
```

```
##
## Call:
## lm(formula = casual ~ atemp + as.factor(workingday) + as.factor(workingday):atemp
##     as.factor(weathersit), data = bike)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -126.311  -19.432   -3.966   12.335  290.198
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -40.4566     1.5124 -26.751  < 2e-16 ***
## atemp                          217.1180     3.0094  72.147  < 2e-16 ***
```

```
## as.factor(workingday)1         25.3616     1.8420  13.768  < 2e-16 ***
## as.factor(weathersit)2         -5.4654     0.6924  -7.894 3.11e-15 ***
## as.factor(weathersit)3        -15.5520     1.1115 -13.992  < 2e-16 ***
## as.factor(weathersit)4          3.5877    22.7098   0.158    0.874
## atemp:as.factor(workingday)1 -126.9260     3.6827 -34.465  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.32 on 17372 degrees of freedom
## Multiple R-squared:  0.3643,Adjusted R-squared:  0.3641
## F-statistic:  1659 on 6 and 17372 DF,  p-value: < 2.2e-16
```

What is the interpretation of the coefficients now? Note that this increases the R^2 more.

By the way, for the bike sharing dataset, will the above models work well for prediction? They do not use hourly information all that much. Throwing in hourly information such as rush hours etc. should improve prediction. Plus there are other unused variables.

5 Inference in Multiple Regression

So far, we have learned how to fit multiple regression equations to observed data. We have not made any modeling assumptions. Inference is necessary for answering questions such as: "Is the observed relationship between the response and the explanatory variables real or is it merely caused by sampling variability?"

In order to perform inference, we need to assume a model. The inference will then be valid only if the assumptions of the model hold true for the particular dataset. The most standard multiple linear regression model (and the only one that we deal with in this class) makes the following assumptions. It is very very similar to the simple linear regression model that you learned in data 8.

5.1 The Multiple Linear Regression Model

There is a response variable y and p explanatory variables $x_1, \ldots, x_p$. The model specifies that the observations are generated at random as follows:

1. The relation between y and $x_1, \ldots, x_p$ is perfectly linear and given by

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p. \tag{12}$$

   The numbers $\beta_0, \ldots, \beta_p$ are called parameters of the model. They are unknown and they form a "true regression plane" that is unknown.

2. Suppose the explanatory variable values for the i^{th} observation are denoted by $x_{i1}, \ldots, x_{ip}$. Then first calculate the value of the response for these x-values from the equation (12) and then add to the obtain value a random error that is normal with mean zero and some variance σ^2. This generates the response value y_i for the i^{th} observation. In other words, the i^{th} response y_i is generated according to the equation:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

   where ϵ_i represents a random error that is normally distributed with mean zero and variance σ^2.

3. The errors $\epsilon_1, \ldots, \epsilon_n$ corresponding to the different observations are assumed to be independent.

The numbers $\beta_0, \ldots, \beta_p$ capture the true relationship between y and $x_1, \ldots, x_p$ and are unknown. Also unknown is the quantity σ^2 which is the variance of the unknown random errors $\epsilon_1, \ldots, \epsilon_n$. We call $\beta_0, \ldots, \beta_p$ and σ^2 the unknown parameters in the multiple linear regression model.

When we fit a regression equation to a dataset via $lm()$ in R, we obtain an equation of the form:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p.$$

The coefficients $b_0, \ldots, b_p$ will be given to us by R which computes them so as to minimize the sum of squares. When we assume the linear regression model, these coefficients $b_0, \ldots, b_p$ can be thought naturally as estimates of the uknown parameters $\beta_0, \ldots, \beta_p$. More specifically, b_0 is an estimate of β_0, b_1 is an estimate of β_1 etc.

The residuals serve as natural proxies for the unknown random errors $\epsilon_1, \ldots, \epsilon_n$. Therefore a natural estimate for the error standard deviation σ is the Residual Standard Error.

5.2 Hypothesis Testing in Multiple Linear Regression

Inference in the linear regression model involves getting standard errors of estimates of $\beta_0, \ldots, \beta_p$, confidence intervals for $\beta_0, \ldots, \beta_p$ and testing various hypotheses involving $\beta_0, \ldots, \beta_p$. We shall first study the problem of testing hypotheses.

Examples of hypothesis testing problems include:

1. $H_0 : \beta_1 = \cdots = \beta_p$.
2. $H_0 : \beta_1 = 0$.
3. $H_0 : \beta_1 = 1$.
4. $H_0 : \beta_1 = \beta_2$.

Let us now see some data examples where each of the hypothesis testing questions above make sense. These examples are taken from the Wooldridge econometrics book.

5.2.1 Example One

```
load(file.path(dataDir, "wage1.Rdata"))
wages = data
wages.desc = desc
head(wages)
```

```
##   wage educ exper tenure nonwhite female married numdep smsa northcen
## 1 3.10   11     2      0        0      1       0      2    1         0
## 2 3.24   12    22      2        0      1       1      3    1         0
## 3 3.00   11     2      0        0      0       0      2    0         0
## 4 6.00    8    44     28        0      0       1      0    1         0
## 5 5.30   12     7      2        0      0       1      1    0         0
## 6 8.75   16     9      8        0      0       1      0    1         0
##   south west construc ndurman trcommpu trade services profserv profocc
## 1     0    1        0       0        0     0        0        0       0
## 2     0    1        0       0        0     0        1        0       0
## 3     0    1        0       0        0     1        0        0       0
## 4     0    1        0       0        0     0        0        0       0
## 5     0    1        0       0        0     0        0        0       0
## 6     0    1        0       0        0     0        0        1       1
##   clerocc servocc    lwage expersq tenursq
## 1       0       0 1.131402       4       0
## 2       0       1 1.175573     484       4
## 3       0       0 1.098612       4       0
## 4       1       0 1.791759    1936     784
## 5       0       0 1.667707      49       4
## 6       0       0 2.169054      81      64
```

```
dim(wages)
```

```
## [1] 526  24
```

```
wages.desc
```

```
##    variable                          label
## 1      wage         average hourly earnings
## 2      educ              years of education
## 3     exper      years potential experience
## 4    tenure     years with current employer
## 5  nonwhite                  =1 if nonwhite
## 6    female                    =1 if female
## 7   married                   =1 if married
## 8    numdep            number of dependents
## 9      smsa               =1 if live in SMSA
## 10 northcen =1 if live in north central U.S
## 11    south   =1 if live in southern region
## 12     west    =1 if live in western region
## 13 construc  =1 if work in construc. indus.
## 14  ndurman  =1 if in nondur. manuf. indus.
## 15 trcommpu  =1 if in trans, commun, pub ut
## 16    trade    =1 if in wholesale or retail
## 17 services        =1 if in services indus.
## 18 profserv     =1 if in prof. serv. indus.
## 19  profocc    =1 if in profess. occupation
## 20  clerocc    =1 if in clerical occupation
## 21  servocc     =1 if in service occupation
## 22    lwage                      log(wage)
## 23  expersq                        exper^2
## 24  tenursq                       tenure^2
```

Suppose we fit a linear regression equation to $\log(wage)$ (why $\log(wage)$ as opposed to $wage$) based on *educ* (years of education), *exper* (years of potential experience) and *tenure* (years with current employer). Suppose we want to test the hypothesis that the return on experience, controlling for education and tenure, is zero in the population against the alternative that it is positive. This corresponds to testing the hypothesis:

$$H_0 : \beta_2 = 0 \quad \text{against} \quad H_1 : \beta_2 > 0$$

in the linear model:

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 tenure.$$

5.2.2 Example Two

```
load(file.path(dataDir, "campus.Rdata"))
campus = data
campus.desc = desc
summary(campus)
```

```
##      enroll           priv           police          crime
##  Min.   : 1799   Min.   :0.0000   Min.   : 1.00   Min.   :   1.0
##  1st Qu.: 6485   1st Qu.:0.0000   1st Qu.: 9.00   1st Qu.:  85.0
##  Median :11990   Median :0.0000   Median :16.00   Median : 187.0
##  Mean   :16076   Mean   :0.1237   Mean   :20.49   Mean   : 394.5
##  3rd Qu.:21836   3rd Qu.:0.0000   3rd Qu.:27.00   3rd Qu.: 491.0
##  Max.   :56350   Max.   :1.0000   Max.   :74.00   Max.   :2052.0
##      lcrime          lenroll          lpolice
##  Min.   :0.000   Min.   : 7.495   Min.   :0.000
##  1st Qu.:4.443   1st Qu.: 8.777   1st Qu.:2.197
##  Median :5.231   Median : 9.392   Median :2.773
##  Mean   :5.277   Mean   : 9.379   Mean   :2.731
##  3rd Qu.:6.196   3rd Qu.: 9.991   3rd Qu.:3.296
##  Max.   :7.627   Max.   :10.939   Max.   :4.304
```

Suppose we fit a linear regression equation with *lcrime* as the response and *lenroll* and *lpolice* as the explanatory variables. It is of interest here to test the hypothesis $H_0 : \beta_1 = 1$ against the alternative $H_1 : \beta_1 > 1$. β_1 has the interpretation as the percentage change in *crime* for a 1 percentage increase in *enrollment* provide *lpolice* remains unchanged. If $\beta_1 > 1$, then, in a relative sense (not just an absolute sense), crime is more of a problem on larger campuses.

5.2.3 Example Three

```
load(file.path(dataDir, "twoyear.Rdata"))
ty = data
ty.desc = desc
head(ty)
```

```
##   female phsrank BA AA black hispanic  id exper        jc     univ
## 1      1      65  0  0     0        0  19   161 0.0000000 0.000000
## 2      1      97  0  0     0        0  93   119 0.0000000 7.033333
## 3      1      44  0  0     0        0  96    81 0.0000000 0.000000
## 4      1      34  0  0     0        1 119    39 0.2666667 0.000000
## 5      1      80  0  0     0        0 132   141 0.0000000 0.000000
## 6      0      59  0  0     0        0 156   165 0.0000000 0.000000
##      lwage     stotal smcity medcity submed lgcity sublg vlgcity subvlg ne
## 1 1.925291 -0.4417497      0       0      0      0     1       0      0  1
## 2 2.796494  0.0000000      1       0      0      0     0       0      0  0
## 3 1.625600 -1.3570027      0       0      0      0     1       0      0  1
## 4 2.223312 -0.1900551      1       0      0      0     0       0      0  0
## 5 1.642083  0.0000000      0       0      0      0     0       0      0  0
## 6 2.079442  1.3887565      1       0      0      0     0       0      0  0
##   nc south   totcoll
## 1  0     0 0.0000000
## 2  1     0 7.0333333
## 3  0     0 0.0000000
## 4  0     0 0.2666667
## 5  0     1 0.0000000
## 6  0     1 0.0000000
```

It is natural to fit a regression equation here with $\log(wage)$ as the response and *jc* (number of years in junior college), *univ* (number of years in university) and *exper* (number of years in the workforce) as the explanatory variables. It is meaningful here to test the null hypothesis $H_0 : \beta_1 = \beta_2$.

5.2.4 Example Four

```
load(file.path(dataDir, "mlb1.Rdata"))
bb = data
bb.desc = desc
bb.desc
```

```
##    variable                      label
## 1    salary         1993 season salary
## 2   teamsal               team payroll
## 3        nl      =1 if national league
## 4     years     years in major leagues
## 5     games        career games played
## 6    atbats             career at bats
```

```
## 7      runs         career runs scored
## 8      hits                career hits
## 9   doubles             career doubles
## 10  triples             career triples
## 11    hruns           career home runs
## 12     rbis      career runs batted in
## 13     bavg     career batting average
## 14       bb               career walks
## 15       so         career strike outs
## 16   sbases        career stolen bases
## 17  fldperc       career fielding perc
## 18 frstbase          = 1 if first base
## 19 scndbase          =1 if second base
## 20 shrtstop            =1 if shortstop
## 21 thrdbase           =1 if third base
## 22 outfield             =1 if outfield
## 23  catcher              =1 if catcher
## 24 yrsallst          years as all-star
## 25   hispan             =1 if hispanic
## 26    black                =1 if black
## 27 whitepop         white pop. in city
## 28 blackpop         black pop. in city
## 29  hisppop      hispanic pop. in city
## 30    pcinc     city per capita income
## 31  gamesyr   games per year in league
## 32  hrunsyr         home runs per year
## 33 atbatsyr           at bats per year
## 34  allstar perc. of years an all-star
## 35  slugavg    career slugging average
## 36   rbisyr              rbis per year
## 37 sbasesyr      stolen bases per year
## 38   runsyr       runs scored per year
## 39 percwhte      percent white in city
## 40 percblck      percent black in city
## 41 perchisp   percent hispanic in city
## 42   blckpb             black*percblck
## 43   hispph            hispan*perchisp
## 44   whtepw             white*percwhte
## 45   blckph             black*perchisp
## 46   hisppb            hispan*percblck
## 47  lsalary              log(salary)
```

```
head(bb)
```

```
##    salary  teamsal nl years games atbats runs hits doubles triples hruns
## 1 6329213 38407380  1    12  1705   6705 1076 1939     320      67   231
## 2 3375000 38407380  1     8   918   3333  407  863     156      38    73
## 3 3100000 38407380  1     5   751   2807  370  840     148      18    46
## 4 2900000 38407380  1     8  1056   3337  405  816     143      18   107
## 5 1650000 38407380  1    12  1196   3603  437  928      19      16   124
## 6  700000 38407380  1    17  2032   7489 1136 2145     270     142    40
##   rbis bavg  bb   so sbases fldperc frstbase scndbase shrtstop thrdbase
## 1  836  289 619  948    314     989        0        1        0        0
## 2  342  259 137  582    133     968        0        0        1        0
## 3  355  299 341  228     41     994        1        0        0        0
## 4  421  245 306  653     15     971        0        0        0        1
## 5  541  258 316  725     32     977        0        0        0        0
## 6  574  286 416 1098    660     987        0        0        0        0
##   outfield catcher yrsallst hispan black whitepop blackpop hisppop pcinc
## 1        0       0        9      0     0  5772110  1547725  893422 18840
## 2        0       0        2      0     1  5772110  1547725  893422 18840
## 3        0       0        0      0     0  5772110  1547725  893422 18840
## 4        0       0        0      0     0  5772110  1547725  893422 18840
## 5        1       0        0      0     1  5772110  1547725  893422 18840
## 6        1       0        2      0     1  5772110  1547725  893422 18840
##     gamesyr   hrunsyr atbatsyr  allstar  slugavg   rbisyr  sbasesyr
## 1 142.08333 19.250000 558.7500 75.00000 46.02535 69.66666 26.166666
## 2 114.75000  9.125000 416.6250 25.00000 39.42394 42.75000 16.625000
## 3 150.20000  9.200000 561.4000  0.00000 41.39651 71.00000  8.200000
## 4 132.00000 13.375000 417.1250  0.00000 39.43662 52.62500  1.875000
## 5  99.66666 10.333333 300.2500  0.00000 37.49653 45.08333  2.666667
## 6 119.52941  2.352941 440.5294 11.76471 37.64188 33.76471 38.823528
##     runsyr percwhte percblck perchisp   blckpb hispph   whtepw  blckph
## 1 89.66666 70.27797 18.84423  10.8778  0.00000      0 70.27797  0.0000
## 2 50.87500 70.27797 18.84423  10.8778 18.84423      0  0.00000 10.8778
## 3 74.00000 70.27797 18.84423  10.8778  0.00000      0 70.27797  0.0000
## 4 50.62500 70.27797 18.84423  10.8778  0.00000      0 70.27797  0.0000
## 5 36.41667 70.27797 18.84423  10.8778 18.84423      0  0.00000 10.8778
## 6 66.82353 70.27797 18.84423  10.8778 18.84423      0  0.00000 10.8778
##   hisppb  lsalary
## 1      0 15.66069
## 2      0 15.03191
## 3      0 14.94691
## 4      0 14.88022
## 5      0 14.31629
## 6      0 13.45884
```

It is reasonable here to fit a regression equation for $\log(salary)$ based on *years*

(number of years in the league), *gamesyr* (average number of games played per year), *bavg* (career batting average), *hrunsyr* (home runs per year) and *rbisyr* (runs batted in per year).

In this regression, what does it mean to test the hypothesis that the betas corresponding to *bavg*, *hrunsyr* and *rbisyr* are all simultaneously zero.

5.3 Every null hypothesis leads to a submodel

Hypothesis testing in the linear model is based on the observation that every null hypothesis leads to a submodel. This is easily seen in the examples.

5.3.1 Example One

```
load(file.path(dataDir, "wage1.Rdata"))
wages = data
wages.desc = desc
```

In this case, we are fitting a linear regression equation to log(*wage*) based on *educ* (years of education), *exper* (years of potential experience) and *tenure* (years with current employer). Let us call this linear model M:

```
M = lm(lwage ~ educ + exper + tenure, data = wages)
summary(M)
```

```
##
## Call:
## lm(formula = lwage ~ educ + exper + tenure, data = wages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05802 -0.29645 -0.03265  0.28788  1.42809
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.284360   0.104190   2.729  0.00656 **
## educ        0.092029   0.007330  12.555  < 2e-16 ***
## exper       0.004121   0.001723   2.391  0.01714 *
## tenure      0.022067   0.003094   7.133 3.29e-12 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4409 on 522 degrees of freedom
## Multiple R-squared:  0.316,Adjusted R-squared:  0.3121
## F-statistic: 80.39 on 3 and 522 DF,  p-value: < 2.2e-16
```

Now suppose that we want to test the hypothesis that $H_0 : \beta_2 = 0$. If this hypothesis is true, it means that we can drop the variable *exper*. Therefore, the null hypothesis here corresponds to the model:

```
m = lm(lwage ~ educ + tenure, data = wages)
summary(m)
```

```
##
## Call:
## lm(formula = lwage ~ educ + tenure, data = wages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10350 -0.29287 -0.04081  0.28672  1.44967
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.404474   0.091696   4.411 1.25e-05 ***
## educ        0.086528   0.006991  12.377  < 2e-16 ***
## tenure      0.025814   0.002680   9.634  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4428 on 523 degrees of freedom
## Multiple R-squared:  0.3085,Adjusted R-squared:  0.3059
## F-statistic: 116.7 on 2 and 523 DF,  p-value: < 2.2e-16
```

5.3.2 Example Two

```
load(file.path(dataDir, "campus.Rdata"))
campus = data
campus.desc = desc
```

Suppose we fit a linear regression equation with *lcrime* as the response and *lenroll* and *lpolice* as the explanatory variables. This model will be denoted by M:

```
M = lm(lcrime ~ lenroll + lpolice, data = campus)
summary(M)
```

```
##
## Call:
## lm(formula = lcrime ~ lenroll + lpolice, data = campus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6362 -0.2223  0.1047  0.4480  1.7519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.7938     1.1120  -4.311 4.01e-05 ***
## lenroll       0.9235     0.1440   6.414 5.67e-09 ***
## lpolice       0.5164     0.1487   3.473 0.000779 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8467 on 94 degrees of freedom
## Multiple R-squared:  0.632,Adjusted R-squared:  0.6242
## F-statistic: 80.72 on 2 and 94 DF,  p-value: < 2.2e-16
```

Now the hypothesis $H_0 : \beta_1 = 1$ corresponds to the submodel m given by

```
m = lm(lcrime ~ offset(1 * lenroll) + lpolice, data = campus)
summary(m)
```

```
##
## Call:
## lm(formula = lcrime ~ offset(1 * lenroll) + lpolice, data = campus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5992 -0.2191  0.0912  0.4321  1.7575
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.3621     0.3041  -17.63  < 2e-16 ***
```

```
## lpolice       0.4617     0.1069    4.32 3.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8435 on 95 degrees of freedom
## Multiple R-squared:  0.6413,Adjusted R-squared:  0.6375
## F-statistic: 169.8 on 1 and 95 DF,  p-value: < 2.2e-16
```

5.3.3 Example Three

```
load(file.path(dataDir, "twoyear.Rdata"))
ty = data
ty.desc = desc
```

It is natural to fit a regression equation here with $\log(wage)$ (which is same as *lwage*) as the response and *jc* (number of years in junior college), *univ* (number of years in university) and *exper* (number of years in the workforce) as the explanatory variables. Let us denote this model by M:

```
M = lm(lwage ~ jc + univ + exper, data = ty)
summary(M)
```

```
##
## Call:
## lm(formula = lwage ~ jc + univ + exper, data = ty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10362 -0.28132  0.00551  0.28518  1.78167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.4723256  0.0210602  69.910   <2e-16 ***
## jc          0.0666967  0.0068288   9.767   <2e-16 ***
## univ        0.0768762  0.0023087  33.298   <2e-16 ***
## exper       0.0049442  0.0001575  31.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4301 on 6759 degrees of freedom
```

```
## Multiple R-squared:  0.2224,Adjusted R-squared:  0.2221
## F-statistic: 644.5 on 3 and 6759 DF,  p-value: < 2.2e-16
```

Now the hypothesis $H_0 : \beta_1 = \beta_2$ corresponds to the submodel m given by

```
m = lm(lwage ~ I(jc + univ) + exper, data = ty)
summary(m)
```

```
##
## Call:
## lm(formula = lwage ~ I(jc + univ) + exper, data = ty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.09708 -0.28069  0.00532  0.28324  1.78332
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.4719702  0.0210606   69.89   <2e-16 ***
## I(jc + univ) 0.0761563  0.0022562   33.75   <2e-16 ***
## exper        0.0049323  0.0001573   31.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4302 on 6760 degrees of freedom
## Multiple R-squared:  0.2222,Adjusted R-squared:  0.222
## F-statistic: 965.6 on 2 and 6760 DF,  p-value: < 2.2e-16
```

5.3.4 Example Four

```
load(file.path(dataDir, "mlb1.Rdata"))
bb = data
bb.desc = desc
```

It is reasonable here to fit a regression equation for $\log(salary)$ based on *years* (number of years in the league), *gamesyr* (average number of games played per year), *bavg* (career batting average), *hrunsyr* (home runs per year) and *rbisyr* (runs batted in per year). Let us call this model M.

```
M = lm(lsalary ~ years + gamesyr + bavg + hrunsyr +
    rbisyr, data = bb)
summary(M)
```

```
##
## Call:
## lm(formula = lsalary ~ years + gamesyr + bavg + hrunsyr + rbisyr,
##     data = bb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02508 -0.45034 -0.04013  0.47014  2.68924
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.119e+01  2.888e-01  38.752  < 2e-16 ***
## years       6.886e-02  1.211e-02   5.684 2.79e-08 ***
## gamesyr     1.255e-02  2.647e-03   4.742 3.09e-06 ***
## bavg        9.786e-04  1.104e-03   0.887    0.376
## hrunsyr     1.443e-02  1.606e-02   0.899    0.369
## rbisyr      1.077e-02  7.175e-03   1.500    0.134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7266 on 347 degrees of freedom
## Multiple R-squared:  0.6278,Adjusted R-squared:  0.6224
## F-statistic: 117.1 on 5 and 347 DF,  p-value: < 2.2e-16
```

Suppose now that we want to test the hypothesis that the betas corresponding to *bavg*, *hrunsyr* and *rbisyr* are all simultaneously zero. This corresponds to the submodel:

```
m = lm(lsalary ~ years + gamesyr, data = bb)
summary(m)
```

```
##
## Call:
## lm(formula = lsalary ~ years + gamesyr, data = bb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.66858 -0.46412 -0.01176  0.49219  2.68829
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.223804   0.108312 103.625  < 2e-16 ***
## years        0.071318   0.012505   5.703  2.5e-08 ***
## gamesyr      0.020174   0.001343  15.023  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7527 on 350 degrees of freedom
## Multiple R-squared:  0.5971,Adjusted R-squared:  0.5948
## F-statistic: 259.3 on 2 and 350 DF,  p-value: < 2.2e-16
```

In all these examples, we are first fitting a regression equation to the response involving some explanatory variables. We are calling this regression model M. In this model, we want to test some null hypothesis involving the parameters. Incorporating this null hypothesis in the model M results in a submodel m. The problem of testing the null hypothesis then effectively boils down to comparison between the model m and the larger model M. This is done by comparing the RSS values of the two regressions which results in the F-statistic.

5.4 F-statistic and F-test

Let $RSS(m)$ and $RSS(M)$ denote the values of the residual sum of squares for the two models m and M respectively. Note that m is a smaller model (contains fewer variables) compared to M and hence $RSS(m)$ will always be atleast as large as $RSS(M)$. The difference

$$RSS(m) - RSS(M)$$

is a natural comparison between m and M. But this quantity depends on scale so one divides by $RSS(M)$ to obtain

$$\frac{RSS(m) - RSS(M)}{RSS(M)}. \tag{13}$$

The above quantity is scale-free. It is also customary to divide the numerator and denominator in the above fraction by the corresponding *degrees of freedom*. The degrees of freedom corresponding to $RSS(M)$ is the residual degrees of freedom in the model M which is $n - p - 1$ where p is the number of explanatory variables in the model M. The degrees of freedom corresponding to $RSS(m)$ is $n - q - 1$ where q is the number of explanatory variables in the model m. Therefore the degrees of freedom corresponding to $RSS(m) - RSS(M)$ (the numerator in (13)) is $(n - q - 1) -$

$(n-p-1) = p-q$. And the degrees of freedom corresponding to the denominator is $n-p-1$. Dividing by the degrees of freedom, we obtain the quantity:

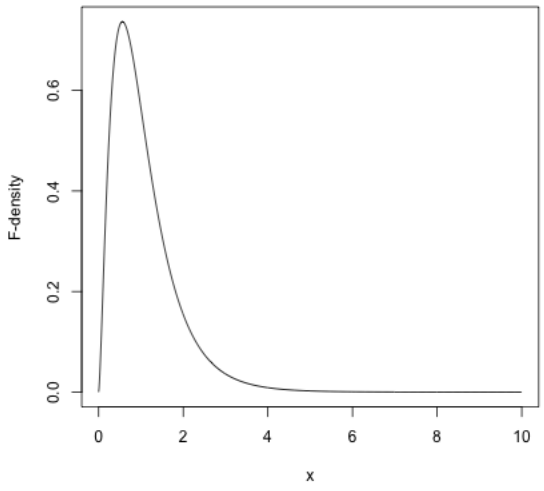
$$F := \frac{(RSS(m) - RSS(M))/(p-q)}{RSS(M)/(n-p-1)}. \tag{14}$$

The above quantity is called the F-statistic and we will denote it by F.

We will reject the null hypothesis (i.e., reject the model m in favor of the larger model M) if the F-statistic is large. The important question though is: **how large is large**?

To answer the question of **how large is large**, we need to know how the value of F looks like if, indeed, the null hypothesis is true (i.e., the data are coming from the smaller model m). Under the assumptions of the multiple linear regression model, it can be shown (this is done in upper division classes on linear models) that, when the null hypothesis is true, the F-statistic has a known distribution. This distribution is called the F-distribution with degrees of freedom given by $p-q$ and $n-p-1$. The F-distribution is a distribution (just like the t-distribution and normal distribution that you have already studied) with two parameters called the degrees of freedom. You can plot its density curve in R easily for any specified choices of the degrees of freedom.

```
d1 = 5
d2 = 30
xx = seq(0, 10, 0.01)
plot(xx, df(xx, d1, d2), type = "l", xlab = "x", ylab = "F-density")
```



We can now easily test any hypotheses in the linear model by simply comparing the observed value of the F-statistic with the corresponding density of the F-distribution.

Let us revisit our four examples.

5.4.1 Example One

```
load(file.path(dataDir, "wage1.Rdata"))
wages = data
wages.desc = desc
```

In this case, we are fitting a linear regression equation to $\log(wage)$ based on *educ* (years of education), *exper* (years of potential experience) and *tenure* (years with current employer). Let us call this linear model M:

```
M = lm(lwage ~ educ + exper + tenure, data = wages)
summary(M)
```

```
##
## Call:
## lm(formula = lwage ~ educ + exper + tenure, data = wages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05802 -0.29645 -0.03265  0.28788  1.42809
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.284360   0.104190   2.729  0.00656 **
## educ        0.092029   0.007330  12.555  < 2e-16 ***
## exper       0.004121   0.001723   2.391  0.01714 *
## tenure      0.022067   0.003094   7.133 3.29e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4409 on 522 degrees of freedom
## Multiple R-squared:  0.316,Adjusted R-squared:  0.3121
## F-statistic: 80.39 on 3 and 522 DF,  p-value: < 2.2e-16
```

Now suppose that we want to test the hypothesis that $H_0 : \beta_2 = 0$. If this hypothesis is true, it means that we can drop the variable *exper*. Therefore, the null hypothesis here corresponds to the model:

```
m = lm(lwage ~ educ + tenure, data = wages)
summary(m)
```

```
##
## Call:
## lm(formula = lwage ~ educ + tenure, data = wages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10350 -0.29287 -0.04081  0.28672  1.44967
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.404474   0.091696   4.411 1.25e-05 ***
## educ        0.086528   0.006991  12.377  < 2e-16 ***
## tenure      0.025814   0.002680   9.634  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4428 on 523 degrees of freedom
## Multiple R-squared:  0.3085,Adjusted R-squared:  0.3059
## F-statistic: 116.7 on 2 and 523 DF,  p-value: < 2.2e-16
```

We can calculate the F-statistic easily via:

```
rss.M = sum((M$residuals)^2)
rss.M
```

```
## [1] 101.4556
```

```
p = 3
rss.m = sum((m$residuals)^2)
rss.m
```

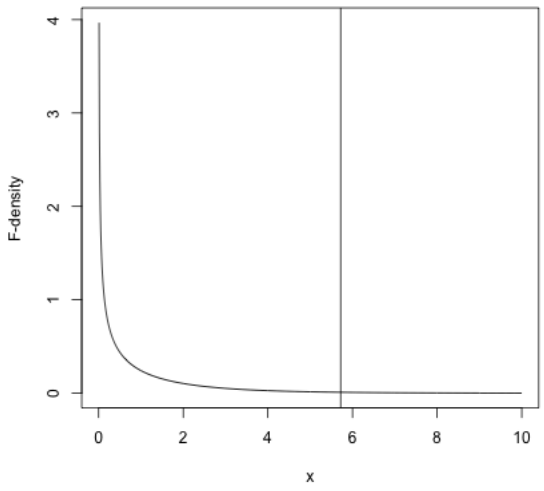
```
## [1] 102.5671
```

```
q = 2
n = nrow(wages)
F = ((rss.m - rss.M)/(p - q))/(rss.M/(n - p - 1))
F
```

```
## [1] 5.718971
```

The distribution of F-statistic under the null hypothesis is given by the F-distribution with degrees of freedom equal to $p - q$ and $n - p - 1$. This density looks like

```
d1 = p - q
d2 = n - p - 1
xx = seq(0, 10, 0.01)
plot(xx, df(xx, d1, d2), type = "l", xlab = "x", ylab = "F-density")
abline(v = F)
```



It seems that the observed F-statistic is quite extreme compared to the null density. We can get a numerical quantification of this extremeness by computing the probability that the null distribution is larger than the observed F-statistic. This gives us the p-value for testing the null hypothesis.

```
1 - pf(F, d1, d2)
```

```
## [1] 0.01713562
```

We get a p-value of 0.0171 which is small (smaller than the usual cutoff of 0.05). We would therefore reject the null hypothesis of $\beta_2 = 0$. This means that even after controlling for *educ* and *tenure*, the explanatory variable *exper* still has an effect on the response.

The p-value for testing the null hypothesis $H_0 : \beta_2 = 0$ can also be obtained from the $lm()$ summary for the model M in R.

```
summary(M)
```

```
##
## Call:
## lm(formula = lwage ~ educ + exper + tenure, data = wages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05802 -0.29645 -0.03265  0.28788  1.42809
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.284360   0.104190   2.729  0.00656 **
## educ        0.092029   0.007330  12.555  < 2e-16 ***
## exper       0.004121   0.001723   2.391  0.01714 *
## tenure      0.022067   0.003094   7.133 3.29e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4409 on 522 degrees of freedom
## Multiple R-squared:  0.316,Adjusted R-squared:  0.3121
## F-statistic: 80.39 on 3 and 522 DF,  p-value: < 2.2e-16
```

Note the value 0.01714 appearing as the p-value corresponding to the variable *exper*. This p-value can also be calculated using a t-distribution. The idea is that the estimate of β_2 divided by its standard error follows the t-distribution with degrees of freedom equal to $n - p - 1$.

5.5 The *anova* function in R

Hypothesis testing is so common that R has an inbuilt function for doing this (you do not have to manually compute the F-statistic and the p-value everytime). This is the *anova* function and it works in the following very simple way. You simply type in $anova(m, M)$ in R and it gives all the necessary information for carrying out the test.

```
anova(m, M)
```

```
## Analysis of Variance Table
##
## Model 1: lwage ~ educ + tenure
## Model 2: lwage ~ educ + exper + tenure
##   Res.Df    RSS Df Sum of Sq     F  Pr(>F)
## 1    523 102.57
## 2    522 101.46  1    1.1115 5.719 0.01714 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let us now revisit our remaining three examples and test the associated null hypothesis in each of them via the *anova*() function.

5.5.1 Example Two

In the campus dataset, we fit a linear regression equation to *lcrime* (response) using the explanatory variables *lenroll* and *lpolice*. This is the model M.

```
M = lm(lcrime ~ lenroll + lpolice, data = campus)
```

In this model, we want to test the null hypothesis $H_0 : \beta_1 = 1$. This null hypothesis corresponds to the model m given by

```
m = lm(lcrime ~ offset(1 * lenroll) + lpolice, data = campus)
```

The null hypothesis H_0 can then be tested via

```
anova(m, M)
```

```
## Analysis of Variance Table
##
## Model 1: lcrime ~ offset(1 * lenroll) + lpolice
## Model 2: lcrime ~ lenroll + lpolice
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     95 67.589
## 2     94 67.387  1   0.20249 0.2825 0.5963
```

This p-value is quite large (0.5963) so we **cannot** reject the null hypothesis that $\beta_1 = 1$.

5.5.2 Example Three

For the "twoyear" data, we fit a regression equation M to $\log(wage)$ in terms of jc (number of years in junior college), $univ$ (number of years in university) and $exper$ (number of years in the workforce). This the model M.

```
M = lm(lwage ~ jc + univ + exper, data = ty)
```

We want to test the null hypothesis $H_0 : \beta_1 = \beta_2$. This null hypothesis corresponds to the submodel:

```
m = lm(lwage ~ I(jc + univ) + exper, data = ty)
```

The test can be done by

```
anova(m, M)
```

```
## Analysis of Variance Table
##
## Model 1: lwage ~ I(jc + univ) + exper
## Model 2: lwage ~ jc + univ + exper
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1   6760 1250.9
## 2   6759 1250.5  1   0.39853 2.154 0.1422
```

The p-value is large so we cannot reject the null hypothesis which means that there is not enough evidence to believe that the effects of junior college and university are different in terms of wages.

5.5.3 Example Four

For the baseball data, we fit a regression equation for $\log(salary)$ based on $years$ (number of years in league), $gamesyr$ (average number of games played per year), $bavg$ (career batting average), $hrunsyr$ (home runs per year) and $rbisyr$ (runs batted in per year). This is the model M:

```
M = lm(lsalary ~ years + gamesyr + bavg + hrunsyr +
    rbisyr, data = bb)
```

We wanted to test the hypothesis that the betas corresponding to *bavg*, *hrunsyr* and *rbisyr* are all simultaneously zero. This corresponds to the submodel:

```
m = lm(lsalary ~ years + gamesyr, data = bb)
```

And the test is done via

```
anova(m, M)
```

```
## Analysis of Variance Table
##
## Model 1: lsalary ~ years + gamesyr
## Model 2: lsalary ~ years + gamesyr + bavg + hrunsyr + rbisyr
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    350 198.31
## 2    347 183.19  3    15.125 9.5503 4.474e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small which means that we would reject the null hypothesis. Therefore performance does count towards salary.

5.6 One sided vs Two sided p-values

When the null hypothesis is of the form $H_0 : \beta_1 = 0$, the alternative hypothesis H_1 is either $H_1 : \beta_1 \neq 0$ or $H_1 : \beta_1 > 0$. We use the second option here ($H_1 : \beta_1 > 0$) when we believe that the effect of the variable x_1 on the response cannot be negative. The p-value given by the *anova*() function applies to the two sided alternative $H_1 : \beta_1 \neq 0$ and **not** to the one sided alternative $H_1 : \beta_1 > 0$. If we want to test the one-sided alternative, then we have to divide the p-value given by the *anova*() function **by 2**.

5.7 The F-statistic given by R in *summary*()

In every summary of a linear model in R, there is an F-statistic value together with degrees of freedom and p-value in the last line. This F-statistic corresponds to the

problem of testing $H_0 : \beta_1 = \cdots = \beta_p = 0$ i.e., that all the explanatory variables can be thrown out of the regression equation.

For example in the baseball dataset,

```
M = lm(lsalary ~ years + gamesyr + bavg + hrunsyr +
    rbisyr, data = bb)
summary(M)
```

```
##
## Call:
## lm(formula = lsalary ~ years + gamesyr + bavg + hrunsyr + rbisyr,
##     data = bb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02508 -0.45034 -0.04013  0.47014  2.68924
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.119e+01  2.888e-01  38.752  < 2e-16 ***
## years       6.886e-02  1.211e-02   5.684 2.79e-08 ***
## gamesyr     1.255e-02  2.647e-03   4.742 3.09e-06 ***
## bavg        9.786e-04  1.104e-03   0.887    0.376
## hrunsyr     1.443e-02  1.606e-02   0.899    0.369
## rbisyr      1.077e-02  7.175e-03   1.500    0.134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7266 on 347 degrees of freedom
## Multiple R-squared:  0.6278,Adjusted R-squared:  0.6224
## F-statistic: 117.1 on 5 and 347 DF,  p-value: < 2.2e-16
```

The F-statistic reported in the last line of the output above is 117.1 with degrees of freedom 5 and 347 along with the p-value $< 2.2 \times 10^{-16}$. This corresponds to testing the null hypothesis $H_0 : \beta_1 = \cdots = \beta_p$. We can test this hypothesis using anova function via

```
m = lm(lsalary ~ 1, data = bb)
summary(m)
```

```
##
```

```
## Call:
## lm(formula = lsalary ~ 1, data = bb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89308 -1.04867 -0.06971  1.13426  2.16850
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.49218    0.06294   214.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.182 on 352 degrees of freedom
```

```
anova(m, M)
```

```
## Analysis of Variance Table
##
## Model 1: lsalary ~ 1
## Model 2: lsalary ~ years + gamesyr + bavg + hrunsyr + rbisyr
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    352 492.18
## 2    347 183.19  5    308.99 117.06 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It is quite rare to have a regression where this p-value is not very small.

5.8 Standard Errors, Confidence Intervals and Prediction Intervals

Recall that the coefficients $b_0, b_1, \ldots, b_p$ in the regression given by R provide estimates of the unknown parameters $\beta_0, \ldots, \beta_p$. The accuracies of these estimates can be gauged by their standard errors. These standard errors are reported in a column next to the estimates in the summary output in R. The smaller the standard error, the more accurate the estimate. The standard errors reported by R are computed under the assumptions of the multiple linear regression model (the details can be learned from more advanced classes).

```
body = read.csv(file.path(dataDir, "bodyfat_short.csv"),
    header = T)
md = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
summary(md)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
##     HIP + THIGH, data = body)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.748e+01  1.449e+01  -2.585  0.01031 *
## AGE          1.202e-02  2.934e-02   0.410  0.68246
## WEIGHT      -1.392e-01  4.509e-02  -3.087  0.00225 **
## HEIGHT      -1.028e-01  9.787e-02  -1.051  0.29438
## CHEST       -8.312e-04  9.989e-02  -0.008  0.99337
## ABDOMEN      9.685e-01  8.531e-02  11.352  < 2e-16 ***
## HIP         -1.834e-01  1.448e-01  -1.267  0.20648
## THIGH        2.857e-01  1.362e-01   2.098  0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  0.7266,Adjusted R-squared:  0.7187
## F-statistic: 92.62 on 7 and 244 DF,  p-value: < 2.2e-16
```

Note that the standard errors corresponding to the different variables (and the intercept) reported above are all different. This means that the accuracies of the different coefficient estimates are different.

It turns out that, under the assumptions of the multiple linear regression model,

$$\frac{b_j - \beta_j}{\text{std. error of} b_j} \sim t_{n-p-1}. \tag{15}$$

Here b_j is the regression coefficient estimate given by R (the denominator above is the standard error of b_j reported by R) and β_j is the unknown regression coefficient.

Also t_{n-p-1} is the t-distribution with $n-p-1$ degrees of freedom. The fact (15) can be used to test the hypothesis $H_0 : \beta_j = 0$ (or any hypothesis of the form $H_1 : \beta_j = c$ for some c). This presents an alternative to the F-test. The two tests are always exactly the same though.

The t-value reported by R is simply the ratio of b_j and its standard error. The p-value is computed by the probability that a t-distribution with $n-p-1$ degrees of freedom exceeds the observed t-statistic.

A 95% confidence interval for the unknown coefficient β_j is computed via:

$$[b_j - (\text{t.cut.off}) * \text{std. error of } b_j, b_j + (\text{t.cut.off}) * \text{std. error of } b_j]$$

where t.cut.off is the cut-off for the t-distribution with $n-p-1$ degrees of freedom which can be computed in R via $qt(0.975, n-p-1)$. As long as $n-p-1$ is not too small, the t.cut.off will be quite close to 2.

```
n = nrow(body)
p = 7
t.cut.off = qt(0.975, (n - p - 1))
t.cut.off
```

```
## [1] 1.969734
```

```
lci = 0.9685 - t.cut.off * (0.08531)
uci = 0.9685 + t.cut.off * (0.08531)
c(lci, uci)
```

```
## [1] 0.800462 1.136538
```

The confidence intervals for all the β-coefficients can be obtained in R by the command $confint()$:

```
confint(md)
```

```
##                    2.5 %      97.5 %
## (Intercept) -66.02663751 -8.92482947
## AGE          -0.04577114  0.06980505
## WEIGHT       -0.22800670 -0.05039445
## HEIGHT       -0.29563565  0.08993870
## CHEST        -0.19757912  0.19591678
```

```
## ABDOMEN       0.80042723  1.13649684
## HIP          -0.46849414  0.10177426
## THIGH         0.01747363  0.55397187
```

Now let us come to prediction. Suppose now that we are asked to predict the bodyfat percentage of an individual who is 30 years of age, 180 pounds in weight, 70 inches tall and whose chest circumference is 95 cm, abdomen circumference is 90 cm, hip circumference is 100 cm and thigh circumference is 60 cm. We can use our linear regression to predict this individual's bodyfat percentage as:

```
x0 = c(1, 30, 180, 70, 95, 90, 100, 60)
pred.bodyfat = sum(x0 * md$coefficients)
pred.bodyfat
```

```
## [1] 16.51927
```

There are two intervals associated with prediction:

1. Confidence intervals for the **average** response. This gives a confidence interval for the **average** bodyfat percentage for **all individuals** who are 30 years of age, 180 pounds in weight, 70 inches tall and whose chest circumference is 95 cm, abdomen circumference is 90 cm, hip circumference is 100 cm and thigh circumference is 60 cm.
2. Confidence intervals for a particular individual. This gives a confidence interval for the body fat percentage of a particular individual with those explanatory variable values. This type of interval is called a **prediction interval**.

These intervals are obtained in R via the *predict* function.

```
x0 = data.frame(AGE = 30, WEIGHT = 180, HEIGHT = 70,
    CHEST = 95, ABDOMEN = 90, HIP = 100, THIGH = 60)
predict(md, x0, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 16.51927 15.20692 17.83162
```

```
predict(md, x0, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 16.51927 7.678715 25.35983
```

Note that the prediction interval is much wider compared to the confidence interval for average response.

6 Variable Selection

Consider a regression problem with a response variable y and p explanatory variables $x_1, \ldots, x_p$. Should we just go ahead and fit a linear model to y with all the p explanatory variables or should we throw out some unnecessary explanatory variables and then fit a linear model for y based on the remaining variables? One often does the latter in practice. The process of selecting important explanatory variables to include in a regression model is called variable selection. The following are reasons for performing variable selection:

1. Removing unnecessary variables results in a simpler model. Simpler models are always preferred to complicated models.
2. Unnecessary explanatory variables will add noise to the estimation of quantities that we are interested in.
3. Collinearity is a problem with having too many variables trying to do the same job.
4. We can save time and/or money by not measuring redundant explanatory variables.

There are two broad ways of performing variable selection in linear models:

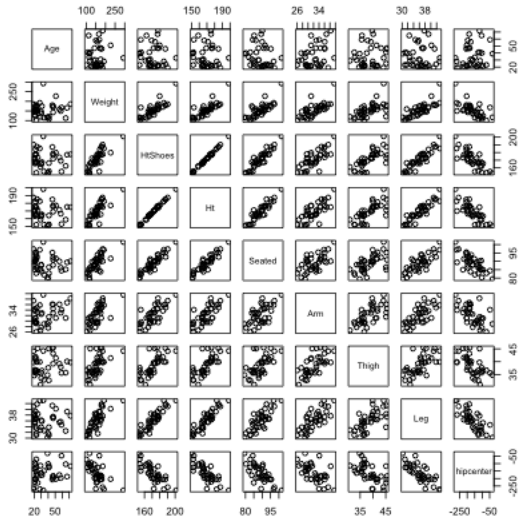
1. Stepwise regression based on p-values
2. Criteria based Variable Selection

We shall illustrate variable selection procedures using the following dataset (which is available in R from the "faraway" package). The material below is mostly taken from the book on linear models by Julian Faraway.

```
library(faraway)
data(seatpos)
names(seatpos)
```

```
## [1] "Age"       "Weight"    "HtShoes"   "Ht"        "Seated"    "Arm"
## [7] "Thigh"     "Leg"       "hipcenter"
```

```
pairs(seatpos)
```




```
g = lm(hipcenter ~ ., seatpos)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ ., data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.827 -22.833  -3.678  25.017  62.337
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.43213  166.57162   2.620   0.0138 *
## Age           0.77572    0.57033   1.360   0.1843
## Weight        0.02631    0.33097   0.080   0.9372
## HtShoes      -2.69241    9.75304  -0.276   0.7845
## Ht            0.60134   10.12987   0.059   0.9531
## Seated        0.53375    3.76189   0.142   0.8882
## Arm          -1.32807    3.90020  -0.341   0.7359
## Thigh        -1.14312    2.66002  -0.430   0.6706
## Leg          -6.43905    4.71386  -1.366   0.1824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 37.72 on 29 degrees of freedom
## Multiple R-squared:  0.6866,Adjusted R-squared:  0.6001
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

Note that the overall p-value reported for the F-statistic in the summary is almost zero but none of the p-values for the individual explanatory variables is small.

7 Stepwise Regression Methods based on p-values

The two main stepwise regression methods are backward elimination and forward selection.

7.1 Backward Elimination

1. Start with all the explanatory variables in the model.
2. Remove the explanatory variable with highest p-value larger than a critical value.
3. Refit the model and go to the previous step.
4. Stop when all the p-values are less than the critical value.

In the car seat position data, the highest p-value in the full regression equation corresponded to the variable Ht. So we can remove it first from the full regression. We will use the *update*() function for this purpose.

```
g <- update(g, . ~ . - Ht)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + Weight + HtShoes + Seated + Arm +
##     Thigh + Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -74.107 -22.467  -4.207  25.106  62.225
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.84207  163.64104   2.670   0.0121 *
## Age           0.76574    0.53590   1.429   0.1634
## Weight        0.02897    0.32244   0.090   0.9290
## HtShoes      -2.13409    2.53896  -0.841   0.4073
## Seated        0.54959    3.68958   0.149   0.8826
## Arm          -1.30087    3.80833  -0.342   0.7350
## Thigh        -1.09039    2.46534  -0.442   0.6615
## Leg          -6.40612    4.60272  -1.392   0.1742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.09 on 30 degrees of freedom
## Multiple R-squared:  0.6865,Adjusted R-squared:  0.6134
## F-statistic: 9.385 on 7 and 30 DF,  p-value: 4.014e-06
```

```
g <- update(g, . ~ . - Weight)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + HtShoes + Seated + Arm + Thigh +
##     Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -74.263 -22.571  -4.842  24.647  61.926
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 427.5073   124.3877   3.437   0.0017 **
## Age           0.7757     0.5158   1.504   0.1427
## HtShoes      -2.0823     2.4329  -0.856   0.3986
## Seated        0.5858     3.6083   0.162   0.8721
## Arm          -1.2826     3.7415  -0.343   0.7341
## Thigh        -1.1153     2.4101  -0.463   0.6468
## Leg          -6.3572     4.4966  -1.414   0.1674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 36.49 on 31 degrees of freedom
## Multiple R-squared:  0.6864,Adjusted R-squared:  0.6257
## F-statistic: 11.31 on 6 and 31 DF,  p-value: 1.122e-06
```

```
g <- update(g, . ~ . - Seated)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + HtShoes + Arm + Thigh + Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.966 -22.403  -4.725  24.989  60.834
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.5463   109.5266   3.986 0.000365 ***
## Age           0.7667     0.5049   1.518 0.138717
## HtShoes      -1.7716     1.4786  -1.198 0.239648
## Arm          -1.3390     3.6683  -0.365 0.717498
## Thigh        -1.1983     2.3193  -0.517 0.608955
## Leg          -6.4910     4.3527  -1.491 0.145686
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.93 on 32 degrees of freedom
## Multiple R-squared:  0.6862,Adjusted R-squared:  0.6371
## F-statistic: 13.99 on 5 and 32 DF,  p-value: 2.823e-07
```

```
g <- update(g, . ~ . - Arm)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + HtShoes + Thigh + Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.069 -24.643  -3.584  26.092  59.182
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 445.7977   105.1452   4.240  0.00017 ***
## Age           0.6525     0.3910   1.669  0.10462
## HtShoes      -1.9171     1.4050  -1.365  0.18164
## Thigh        -1.3732     2.2392  -0.613  0.54391
## Leg          -6.9502     4.1118  -1.690  0.10040
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.46 on 33 degrees of freedom
## Multiple R-squared:  0.6849,Adjusted R-squared:  0.6467
## F-statistic: 17.93 on 4 and 33 DF,  p-value: 6.535e-08
```

```
g <- update(g, . ~ . - Thigh)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + HtShoes + Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.269 -22.770  -4.342  21.853  60.907
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 456.2137   102.8078   4.438 9.09e-05 ***
## Age           0.5998     0.3779   1.587   0.1217
## HtShoes      -2.3023     1.2452  -1.849   0.0732 .
## Leg          -6.8297     4.0693  -1.678   0.1024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.13 on 34 degrees of freedom
## Multiple R-squared:  0.6813,Adjusted R-squared:  0.6531
## F-statistic: 24.22 on 3 and 34 DF,  p-value: 1.437e-08
```

```
g <- update(g, . ~ . - Age)
summary(g)
```

```
##
```

```
## Call:
## lm(formula = hipcenter ~ HtShoes + Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -89.713 -25.787   2.549  18.445  71.735
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  493.794    102.192   4.832 2.66e-05 ***
## HtShoes       -2.496      1.266  -1.971   0.0566 .
## Leg           -6.369      4.146  -1.536   0.1335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.88 on 35 degrees of freedom
## Multiple R-squared:  0.6577,Adjusted R-squared:  0.6381
## F-statistic: 33.62 on 2 and 35 DF,  p-value: 7.132e-09
```

```
g <- update(g, . ~ . - Leg)
summary(g)
```

```
##
## Call:
## lm(formula = hipcenter ~ HtShoes, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.981 -27.150   2.983  22.637  73.731
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 565.5927    92.5794   6.109 4.97e-07 ***
## HtShoes      -4.2621     0.5391  -7.907 2.21e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.55 on 36 degrees of freedom
## Multiple R-squared:  0.6346,Adjusted R-squared:  0.6244
## F-statistic: 62.51 on 1 and 36 DF,  p-value: 2.207e-09
```

The critical value is sometimes called the p-to-remove and does not always have to be 0.05. If prediction performance is the goal, then a 0.15-0.20 cut-off may work best, although methods designed more directly for optimal prediction (such as cross-validation discussed later) should be preferred.

What model will we end up with if we chose the critical value to be 0.15? 0.20?

7.2 Forward Selection

This just reverses the backward method:

1. Start with no variables in the model.
2. For all predictors not in the model, check their p-value if they are added to the model. Choose the one with lowest p-value less than the critical value.
3. Continue until no new predictors can be added.

This can be easily done in any regression. For example, in the car seat position data, we can do the following:

```
for (i in 1:8) {
    g1 <- lm(hipcenter ~ ., seatpos[, c(i, 9)])
    print((summary(g1))$coef)
}
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -192.964532 24.3015104 -7.940434 1.997902e-09
## Age            0.796289  0.6330851  1.257791 2.165650e-01
##              Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)  1.242164 34.0562959  0.03647384 9.711060e-01
## Weight      -1.067438  0.2134036 -5.00196673 1.493391e-05
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 565.592659 92.5794472  6.109268 4.966825e-07
## HtShoes      -4.262091  0.5390607 -7.906513 2.206673e-09
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 556.255344 90.6704339  6.134914 4.590529e-07
## Ht           -4.264977  0.5351079 -7.970312 1.830624e-09
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 621.823241 122.488378  5.076590 1.188644e-05
## Seated       -8.844124   1.374951 -6.432321 1.844619e-07
```

```
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 168.5938  77.445423  2.176936 0.0361229045
## Arm         -10.3514   2.391242 -4.328881 0.0001142407
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 186.890582  80.373022  2.325290 2.580801e-02
## Thigh        -9.100325   2.069128 -4.398145 9.290168e-05
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 335.35118  65.601153  5.111971 1.066600e-05
## Leg         -13.79461   1.801321 -7.658051 4.587375e-09
```

```
for (i in c(1:3, 5:8)) {
    g2 <- lm(hipcenter ~ ., seatpos[, c(i, 4, 9)])
    print((summary(g2))$coef)
}
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 526.9588909 92.2478769  5.712423 1.848352e-06
## Age           0.5210614  0.3862472  1.349036 1.859889e-01
## Ht           -4.2003806  0.5312787 -7.906172 2.691691e-09
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 589.9005691 127.4824497  4.6273081 4.918677e-05
## Weight        0.1148334   0.3020236  0.3802135 7.060846e-01
## Ht           -4.5696588   0.9671936 -4.7246577 3.675055e-05
##               Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) 552.568840  95.754525  5.7706812 1.548571e-06
## HtShoes       1.230074   8.937649  0.1376284 8.913228e-01
## Ht           -5.490019   8.917605 -0.6156382 5.421163e-01
##               Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) 538.486199 112.509336  4.7861468 3.055654e-05
## Seated        0.903040   3.301534  0.2735214 7.860601e-01
## Ht           -4.634962   1.457264 -3.1805911 3.074214e-03
##                Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) 560.2221531 93.4055743  5.9977379 7.774137e-07
## Arm           0.6441307  2.7270290  0.2362024 8.146525e-01
## Ht           -4.4111641  0.8228605 -5.3607676 5.378845e-06
##                Estimate Std. Error     t value     Pr(>|t|)
## (Intercept) 555.6572893 92.5218032  6.00569023 7.588931e-07
## Thigh        -0.1346205  2.3075402 -0.05833939 9.538101e-01
## Ht           -4.2306633  0.8002714 -5.28653594 6.737776e-06
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 491.243757  99.543358  4.934973 1.952182e-05
## Leg          -6.135518   4.164051 -1.473449 1.495675e-01
## Ht           -2.564612   1.268480 -2.021799 5.089444e-02
```

7.3 Other Stepwise Regression Methods

There are several other stepwise regression methods. These are all combinations of backward elimination and forward selection. These might be better than backward elimination or forward selection by addressing the situation where variables are added or removed early in the process and we want to change our mind about them later. At each stage a variable may be added or removed and there are several variations on exactly how this is done.

7.4 Drawbacks of Stepwise Regression based on p-values

Stepwise procedures based on p-values are relatively cheap computationally but they do have the following drawbacks:

1. Because of the one-at-a-time nature of adding/dropping variables, it is possible to miss the optimal model.
2. It is difficult to justify the reliance on p-values for variable selection.
3. The procedures are not directly linked to final objectives of prediction or explanation and so may not really help solve the problem of interest.

7.5 Criteria based variable selection

With p explanatory variables, there are 2^p possible linear regression models that one can fit to the data. A natural idea is to select a performance criterion and compare all the 2^p models according to this criterion and choose the model with optimizes the criterion. What is a natural criterion to use?

A first choice for the criterion is the RSS (Residual Sum of Squares). The RSS is indeed a commonly used measure of the performance of a regression model. However it is not a good criterion for variable selection because it will always choose the full model. This is because RSS decreases as one increases the number of explanatory variables. However, RSS is a natural criterion to use when comparing models having the **same number** of explanatory variables.

A function in R that is useful for variable selection is *regsubsets* in the R package *leaps*. For each value of $k = 1, \ldots, p$, this function gives the best model with k variables according to the residual sum of squares.

```
library(leaps)
b = regsubsets(hipcenter ~ ., seatpos)
rs = summary(b)
rs$which
```

```
##   (Intercept)   Age Weight HtShoes    Ht Seated   Arm Thigh   Leg
## 1        TRUE FALSE  FALSE   FALSE  TRUE  FALSE FALSE FALSE FALSE
## 2        TRUE FALSE  FALSE   FALSE  TRUE  FALSE FALSE FALSE  TRUE
## 3        TRUE  TRUE  FALSE   FALSE  TRUE  FALSE FALSE FALSE  TRUE
## 4        TRUE  TRUE  FALSE    TRUE FALSE  FALSE FALSE  TRUE  TRUE
## 5        TRUE  TRUE  FALSE    TRUE FALSE  FALSE  TRUE  TRUE  TRUE
## 6        TRUE  TRUE  FALSE    TRUE FALSE   TRUE  TRUE  TRUE  TRUE
## 7        TRUE  TRUE   TRUE    TRUE FALSE   TRUE  TRUE  TRUE  TRUE
## 8        TRUE  TRUE   TRUE    TRUE  TRUE   TRUE  TRUE  TRUE  TRUE
```

This output should be interpreted in the following way. The best model with one explanatory variable (let us denote this by M_1) is the model with *Ht*. The best model with two explanatory variables (denoted by M_2) is the one involving Ht and Leg. The model with three explanatory variables (M_3) involves Age, Ht and Leg. And so on. Here best is in terms of RSS. This gives us 8 regression models: $M_1, M_2, \ldots, M_8$. The model M_8 is the full regression model involving all the explanatory variables.

The key question now is: how does not compare the eight models $M_1, \ldots, M_8$? Note that we can no longer use RSS because that would simply give M_8. Cross-validation is a natural tool here.

The idea behind cross validation is the following. It is sensible to pick, among the models $M_1, \ldots, M_8$, the model which has the **best predictive performance**. If we had access to future data, we can evaluate our models based on their predictiver performance on that future data. How can we do this based on existing data alone?

Let m denote one of the eight models (or any other regression model m). For example, suppose m is the regression model involving all the explanatory variables or only the variables *Age*, *Ht* and *Leg*. How do we measure the predictive performance of m? Here is a simple idea: For each $i = 1, \ldots, n$, fit the model m to the $(n-1)$ observations obtained by **excluding** the i^{th} observation. Predict the response for the i^{th} observation using this model m and the values of the explanatory variables for the i^{th} observation. Record the prediction error. Do this for each $i = 1, \ldots, n$ and then add the squares of the prediction errors. This gives the **Leave One Out Cross Validation Score** for the model m. Pick the model m for which this score is the smallest.

For the car seat position dataset, the Leave One Out Cross Validation Score for

the model using all the variables is:

```
n <- nrow(seatpos)
pred.y <- rep(NA, n)
for (i in 1:nrow(seatpos)) {
    g <- lm(hipcenter ~ ., seatpos, subset = (1:n)[-i])
    pred.y[i] <- predict(g, seatpos[i, -9])
}
cv.err.full <- sum((seatpos[, 9] - pred.y)^2)
cv.err.full
```

```
## [1] 75065.76
```

On the other hand, the Leave One Out Cross Validation Score for the model M_3 which uses only the variables *Age*, *Ht* and *Leg* is

```
for (i in 1:nrow(seatpos)) {
    g <- lm(hipcenter ~ Age + Ht + Leg, seatpos, subset = (1:n)[-i])
    pred.y[i] <- predict(g, seatpos[i, c("Age", "Ht",
        "Leg")])
}
cv.err.Age.Ht.Leg <- sum((seatpos[, 9] - pred.y)^2)
cv.err.Age.Ht.Leg
```

```
## [1] 53794.79
```

Clearly the cross validation score of the second model is much smaller. We can therefore compute the cross validation score of all the eight models $M_1, \ldots, M_8$ and then pick the model which has the smallest cross-validation score.

8 Regression Diagnostics

Our final topic in multiple regression is regression diagnostics. The inference procedures that we talked about work under the assumptions of the linear regression model. If these assumptions are violated, then our hypothesis tests, standard errors and confidence intervals will not be violated. Regression diagnostics enable us to diagnose if the model assumptions are violated or not.

The assumptions in the regression model are:

1. Linearity: the response is linearly related to the explanatory variables.
2. Homoscedasticity: the errors have the same variance.
3. Normality: the errors have the normal distribution.
4. All the observations obey the same model (i.e., there are no outliers or exceptional observations).

These assumptions can be checked by essentially looking at the residuals:

1. **Linearity**: The residuals represent what is left in the response variable after the linear effects of the explanatory variables are taken out. So if there is a non-linear relationship between the response and one or more of the explanatory variables, the residuals will be related non-linearly to the explanatory variables. This can be detected by plotting the residuals against the explanatory variables. It is also common to plot the residuals against the fitted values. Note that one can also detect non-linearity by simply plotting the response against each of the explanatory variables.
2. **Homoscedasticity**: Heteroscedasticity can be checked again by plotting the residuals against the explanatory variables and the fitted values. It is common here to plot the absolute values of the residuals or the square root of the absolute values of the residuals.
3. **Normality**: Detected by the normal Q-Q plot of the residuals.
4. **Outliers**: Detected by large (in absolute value) residuals. There is a notion called Cook's distance which detects by how much the regression coefficients change if a particular observation is removed. Outliers typically will have either large (in absolute value) residuals and/or large Cook's distance.

Consider the bodyfat dataset.

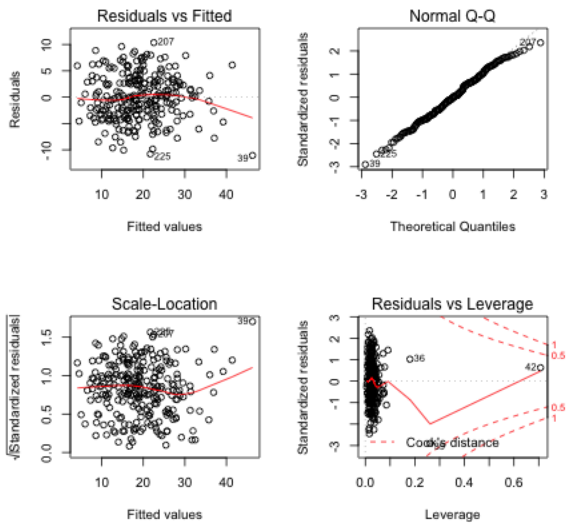
```
body = read.csv(file.path(dataDir, "bodyfat_short.csv"),
    header = T)
md = lm(BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
    HIP + THIGH, data = body)
summary(md)
```

```
##
## Call:
## lm(formula = BODYFAT ~ AGE + WEIGHT + HEIGHT + CHEST + ABDOMEN +
##     HIP + THIGH, data = body)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0729  -3.2387  -0.0782   3.0623  10.3611
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.748e+01  1.449e+01  -2.585  0.01031 *
## AGE          1.202e-02  2.934e-02   0.410  0.68246
## WEIGHT      -1.392e-01  4.509e-02  -3.087  0.00225 **
## HEIGHT      -1.028e-01  9.787e-02  -1.051  0.29438
## CHEST       -8.312e-04  9.989e-02  -0.008  0.99337
## ABDOMEN      9.685e-01  8.531e-02  11.352  < 2e-16 ***
## HIP         -1.834e-01  1.448e-01  -1.267  0.20648
## THIGH        2.857e-01  1.362e-01   2.098  0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.438 on 244 degrees of freedom
## Multiple R-squared:  0.7266,Adjusted R-squared:  0.7187
## F-statistic: 92.62 on 7 and 244 DF,  p-value: < 2.2e-16
```

A simple way for doing regression diagnostics is to use the *plot*(*md*) comment in R:

```
par(mfrow = c(2, 2))
plot(md)
```



```
par(mfrow = c(1, 1))
```

The first plot is the residuals plotted against the fitted values. The points should look like a random scatter with no discernible pattern. Non-linearity (if exists) will be visible in this plot.

The second plot is the normal Q-Q plot. If the normal assumption holds, then the points should be along the line here.

The third plot is called the Scale-Location plot. It plots the square root of the absolute value of the residuals (actually standardized residuals but these are similar to the residuals) against the fitted values. Any increasing or decreasing pattern in this plot indicates heteroscedasticity.

The final plot is used for detecting outliers and other exceptional observations. The x-axis is called leverage (we will not discuss this here); the y-axis is standardized residuals. This flags observations that are potential outliers. Three points flagged here are observations 39, 42 and 36. Let us look at these observations separately:

```
body[c(39, 42, 36), ]
```

```
##    BODYFAT AGE WEIGHT HEIGHT CHEST ABDOMEN   HIP THIGH
## 39    35.2  46 363.15  72.25 136.2   148.1 147.7  87.3
## 42    32.9  44 205.00  29.50 106.0   104.3 115.5  70.6
## 36    40.1  49 191.75  65.00 118.5   113.1 113.8  61.9
```

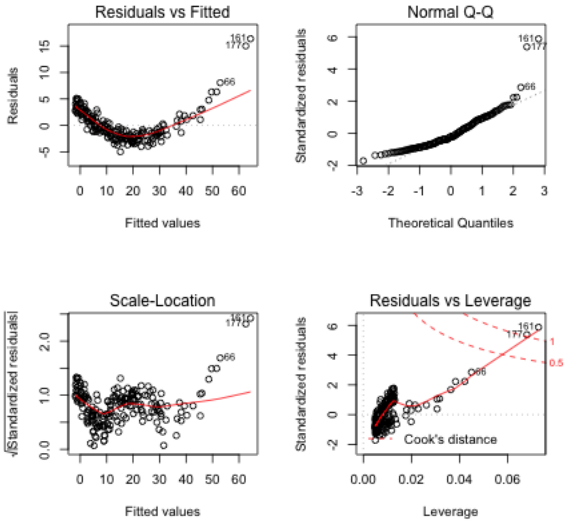
```
apply(body, 2, mean)
```

```
##   BODYFAT       AGE    WEIGHT    HEIGHT     CHEST   ABDOMEN       HIP
##  19.15079  44.88492 178.92440  70.14881 100.82421  92.55595  99.90476
##     THIGH
##  59.40595
```

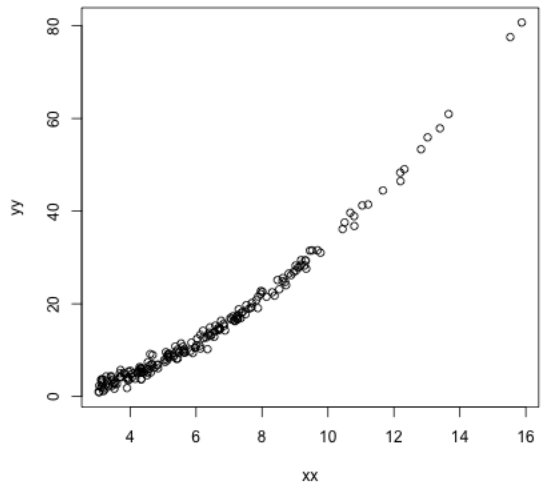
The observation 39 is certainly an outlier. Observation 42 seems to have an erroneous height recording. Observation 36 has high values for chest, abdomen and hip circumference values and also a high value for the response. When outliers are detected, one should perform the regression analysis after dropping the outlying observations. After this, one needs to decide whether to report the analysis with the outliers or without them.

Let us now look at some simulation examples. In the next example, the response is related non-linearly to x.

```
n = 200
xx = 3 + 4 * abs(rnorm(n))
yy = -2 + 0.5 * xx^(1.85) + rnorm(n)
m1 = lm(yy ~ xx)
par(mfrow = c(2, 2))
plot(m1)
```




```
par(mfrow = c(1, 1))
plot(yy ~ xx)
```

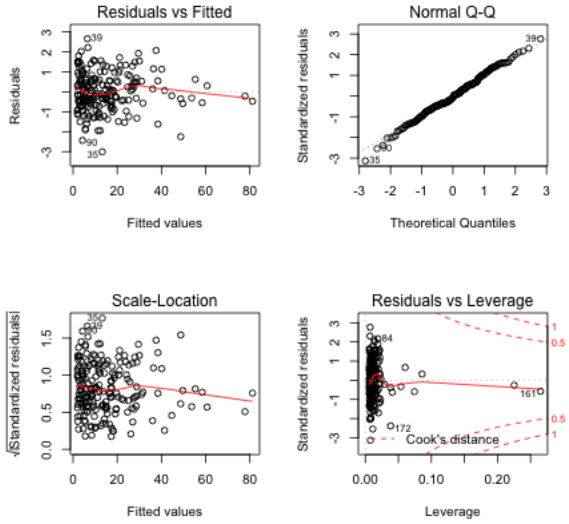


Non-linearity is fixed by adding non-linear functions of explanatory variables as additional explanatory variables. In this example, for instance, we can add x^2 as an additional explanatory variable.

```
m2 = lm(yy ~ xx + I(xx^2))
summary(m2)
```

```
##
## Call:
## lm(formula = yy ~ xx + I(xx^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.00815 -0.57585 -0.05844  0.60106  2.65706
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.840658   0.410884  -6.914 6.42e-11 ***
## xx           0.690140   0.115226   5.989 9.83e-09 ***
## I(xx^2)      0.289921   0.007278  39.835  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9643 on 197 degrees of freedom
## Multiple R-squared:  0.9951,Adjusted R-squared:  0.9951
## F-statistic: 2.004e+04 on 2 and 197 DF,  p-value: < 2.2e-16
```

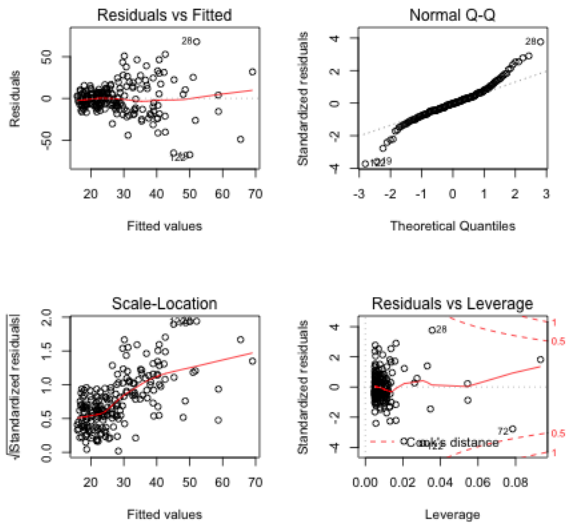
```
par(mfrow = c(2, 2))
plot(m2)
```



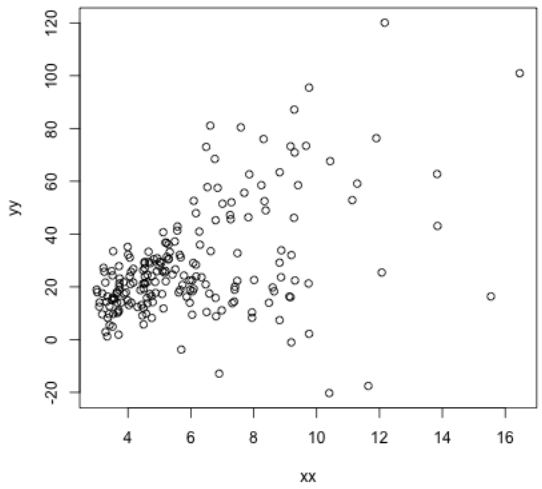

```
par(mfrow = c(1, 1))
```

Next let us consider an example involving heterscedasticity.

```
n = 200
xx = 3 + 4 * abs(rnorm(n))
yy = -2 + 5 * xx + (xx^(1.5)) * rnorm(n)
m1 = lm(yy ~ xx)
par(mfrow = c(2, 2))
plot(m1)
```



```
par(mfrow = c(1, 1))
plot(yy ~ xx)
```



Heteroscedasticity is a little tricky to handle in general. If all the response values are positive, heteroscedasiticity is often fixed by fitting a regression equation to the logarithm or square root of the response variable.