Data Distributions

Aditya Guntuboyina & Elizabeth Purdom

This document has last been compiled on Feb 04, 2020.

Contents

We're going to review some basic ideas about distributions you should have learned in Data 8 or STAT 20. In addition to review, we introduce some new ideas and emphases to pay attention to:

- Continuous distributions and density curves
- New tools for visualizing and estimating distributions: boxplots and kernel density estimators
- Types of samples and how they effect estimation

1 Basic Exporatory analysis

Let's look at a dataset that contains the salaries of San Francisco employees.[1] We've streamlined this to the year 2014 (and removed some strange entires with negative pay). Let's explore this data.

```
dataDir <- "../../finalDataSets"
nameOfFile <- file.path(dataDir, "SFSalaries2014.csv")
salaries2014 <- read.csv(nameOfFile, na.strings = "Not Provided")
dim(salaries2014)
```

```
## [1] 38117    10
```

```
names(salaries2014)
```

```
##  [1] "X"                "Id"               "JobTitle"         "BasePay"
##  [5] "OvertimePay"      "OtherPay"         "Benefits"         "TotalPay"
##  [9] "TotalPayBenefits" "Status"
```

```
salaries2014[1:10, c("JobTitle", "Benefits", "TotalPay",
    "Status")]
```

```
##                          JobTitle Benefits TotalPay Status
## 1                 Deputy Chief 3 38780.04 471952.6     PT
## 2              Asst Med Examiner 89540.23 390112.0     FT
```

[1] https://www.kaggle.com/kaggle/sf-salaries/

```
## 3        Chief Investment Officer 96570.66 339653.7     PT
## 4                Chief of Police 91302.46 326716.8     FT
## 5          Chief, Fire Department 91201.66 326233.4     FT
## 6               Asst Med Examiner 71580.48 344187.5     FT
## 7                    Dept Head V 89772.32 311298.5     FT
## 8     Executive Contract Employee 88823.51 310161.0     FT
## 9  Battalion Chief, Fire Suppress 59876.90 335485.0     FT
## 10   Asst Chf of Dept (Fire Dept) 64599.59 329390.5     FT
```

Let's look at the column 'TotalPay' which gives the total pay, not including benefits.

How might we want to explore this data? What single number summaries would make sense? What visualizations could we do?

```
summary(salaries2014$TotalPay)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0   33482   72368   75476  107980  471953
```

Notice we entries with zero pay! Let's investigate why we have zero pay by subsetting to just those entries.

```
zeroPay <- subset(salaries2014, TotalPay == 0)
nrow(zeroPay)
```

```
## [1] 48
```

```
head(zeroPay)
```

```
##            X     Id                       JobTitle BasePay OvertimePay OtherPay
## 34997 145529 145529          Special Assistant 15       0           0        0
## 35403 145935 145935 Community Police Services Aide       0           0        0
## 35404 145936 145936     BdComm Mbr, Grp3,M=$50/Mtg       0           0        0
## 35405 145937 145937     BdComm Mbr, Grp3,M=$50/Mtg       0           0        0
## 35406 145938 145938                       Gardener       0           0        0
## 35407 145939 145939                       Engineer       0           0        0
```

```
##       Benefits TotalPay TotalPayBenefits Status
## 34997  5650.86        0          5650.86     PT
## 35403  4659.36        0          4659.36     PT
## 35404  4659.36        0          4659.36     PT
## 35405  4659.36        0          4659.36     PT
## 35406  4659.36        0          4659.36     PT
## 35407  4659.36        0          4659.36     PT
```

```
summary(zeroPay)
```

```
##        X                Id                                      JobTitle
##  Min.   :145529   Min.   :145529   General Laborer                  : 4
##  1st Qu.:145948   1st Qu.:145948   Custodian                        : 3
##  Median :145960   Median :145960   Transit Operator                 : 3
##  Mean   :147228   Mean   :147228   Arborist Technician              : 2
##  3rd Qu.:148637   3rd Qu.:148637   BdComm Mbr, Grp3,M=$50/Mtg       : 2
##  Max.   :148650   Max.   :148650   Community Police Services Aide: 2
##                                    (Other)                         :32
##     BasePay   OvertimePay    OtherPay    Benefits       TotalPay
##  Min.   :0   Min.   :0    Min.   :0   Min.   :   0   Min.   :0
##  1st Qu.:0   1st Qu.:0    1st Qu.:0   1st Qu.:   0   1st Qu.:0
##  Median :0   Median :0    Median :0   Median :4646   Median :0
##  Mean   :0   Mean   :0    Mean   :0   Mean   :2444   Mean   :0
##  3rd Qu.:0   3rd Qu.:0    3rd Qu.:0   3rd Qu.:4649   3rd Qu.:0
##  Max.   :0   Max.   :0    Max.   :0   Max.   :5651   Max.   :0
##
##  TotalPayBenefits Status
##  Min.   :   0     FT: 0
##  1st Qu.:   0     PT:48
##  Median :4646
##  Mean   :2444
##  3rd Qu.:4649
##  Max.   :5651
##
```

It's not clear why these people received zero pay. We might want to remove them, thinking that zero pay are some kind of weird problem with the data we aren't interested in. But let's so a quick summary of what the data would look like if we did remove them:

```
summary(subset(salaries2014, TotalPay > 0))
```

```
##        X                Id                                    JobTitle
##  Min.   :110532   Min.   :110532   Transit Operator                : 2476
##  1st Qu.:120049   1st Qu.:120049   Special Nurse                   : 1476
##  Median :129566   Median :129566   Registered Nurse                : 1234
##  Mean   :129568   Mean   :129568   Public Svc Aide-Public Works:  915
##  3rd Qu.:139083   3rd Qu.:139083   Firefighter                     :  813
##  Max.   :148626   Max.   :148626   Custodian                       :  801
##                                    (Other)                         :30354
##     BasePay        OvertimePay       OtherPay         Benefits
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :    0
##  1st Qu.: 30439   1st Qu.:     0   1st Qu.:     0   1st Qu.:10417
##  Median : 65055   Median :     0   Median :   700   Median :28443
##  Mean   : 66652   Mean   :  5409   Mean   :  3510   Mean   :24819
##  3rd Qu.: 94865   3rd Qu.:  5132   3rd Qu.:  4105   3rd Qu.:35445
##  Max.   :318836   Max.   :173548   Max.   :342803   Max.   :96571
##
##     TotalPay        TotalPayBenefits   Status
##  Min.   :     1.8   Min.   :     7.2   FT:22334
##  1st Qu.: 33688.3   1st Qu.: 44561.8   PT:15735
##  Median : 72414.3   Median :101234.9
##  Mean   : 75570.7   Mean   :100389.8
##  3rd Qu.:108066.1   3rd Qu.:142814.2
##  Max.   :471952.6   Max.   :510732.7
##
```

We can see that in fact we still have some weird pay entires (e.g. total payment of \$1.8). This points to the slippery slope you can get into in "cleaning" your data – where do you stop?

A better observation is to notice that all the zero-entries have "Status" value of PT, meaning they are part-time workers.

```
summary(subset(salaries2014, Status == "FT"))
```

```
##        X                Id                       JobTitle        BasePay
##  Min.   :110533   Min.   :110533   Transit Operator: 1524   Min.   : 26364
##  1st Qu.:116598   1st Qu.:116598   Firefighter     :  738   1st Qu.: 65055
##  Median :122928   Median :122928   Police Officer 3:  642   Median : 84084
##  Mean   :123068   Mean   :123068   Custodian       :  565   Mean   : 91174
##  3rd Qu.:129309   3rd Qu.:129309   Deputy Sheriff  :  552   3rd Qu.:112171
```

```
##  Max.   :140326   Max.   :140326   Police Officer  :  399   Max.   :318836
##                                    (Other)         :17914
##   OvertimePay        OtherPay         Benefits        TotalPay
##  Min.   :     0   Min.   :     0   Min.   :    0   Min.   : 26364
##  1st Qu.:     0   1st Qu.:     0   1st Qu.:29122   1st Qu.: 72356
##  Median :  1621   Median :  1398   Median :33862   Median : 94272
##  Mean   :  8241   Mean   :  4091   Mean   :35023   Mean   :103506
##  3rd Qu.: 10459   3rd Qu.:  5506   3rd Qu.:38639   3rd Qu.:127856
##  Max.   :173548   Max.   :112776   Max.   :91302   Max.   :390112
##
##  TotalPayBenefits Status
##  Min.   : 31973   FT:22334
##  1st Qu.:102031   PT:    0
##  Median :127850
##  Mean   :138528
##  3rd Qu.:167464
##  Max.   :479652
##
```

```
summary(subset(salaries2014, Status == "PT"))
```

```
##        X                Id                                    JobTitle
##  Min.   :110532   Min.   :110532   Special Nurse                  : 1473
##  1st Qu.:136520   1st Qu.:136520   Transit Operator               :  955
##  Median :140757   Median :140757   Public Svc Aide-Public Works:  897
##  Mean   :138820   Mean   :138820   Registered Nurse               :  855
##  3rd Qu.:144704   3rd Qu.:144704   Recreation Leader              :  690
##  Max.   :148650   Max.   :148650   Public Service Trainee         :  479
##                                    (Other)                       :10434
##     BasePay        OvertimePay         OtherPay           Benefits
##  Min.   :     0   Min.   :    0.0   Min.   :     0.0   Min.   :    0.0
##  1st Qu.:  6600   1st Qu.:    0.0   1st Qu.:     0.0   1st Qu.:  115.7
##  Median : 20557   Median :    0.0   Median :   191.7   Median : 4659.4
##  Mean   : 31749   Mean   : 1385.6   Mean   :  2676.7   Mean   :10312.3
##  3rd Qu.: 47896   3rd Qu.:  681.2   3rd Qu.:  1624.7   3rd Qu.:19246.2
##  Max.   :257340   Max.   :74936.0   Max.   :342802.6   Max.   :96570.7
##
##     TotalPay      TotalPayBenefits Status
##  Min.   :     0   Min.   :     0   FT:    0
##  1st Qu.:  7359   1st Qu.:  8256   PT:15783
##  Median : 22410   Median : 27834
##  Mean   : 35811   Mean   : 46123
##  3rd Qu.: 52998   3rd Qu.: 72569
```

```
##  Max.    :471953   Max.    :510733
##
```

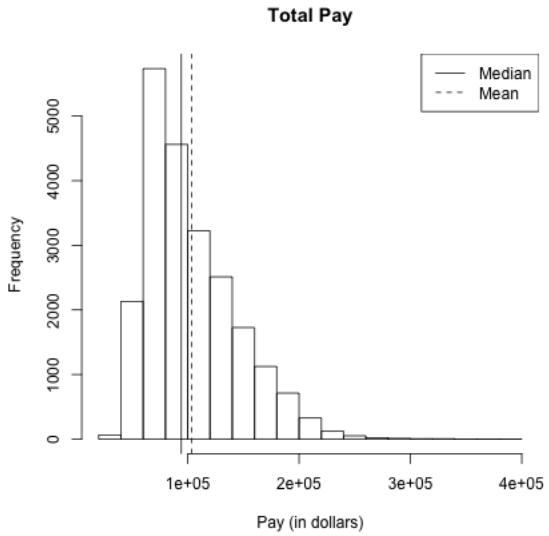
So it is clear that analyzing data from part-time workers will be tricky (and we have no information here as to whether they worked a week or eleven months). To simplify things, we will make a new data set with only full-time workers:

```
salaries2014_FT <- subset(salaries2014, Status == "FT")
```

1.1 Histograms

Let's draw a histogram of the total salary for full-time workers only.

```
hist(salaries2014_FT$TotalPay, main = "Total Pay",
    xlab = "Pay (in dollars)")
abline(v = mean(salaries2014_FT$TotalPay), lty = "dashed")
abline(v = median(salaries2014_FT$TotalPay))
legend("topright", legend = c("Median", "Mean"), lty = c("solid",
    "dashed"))
```



What do you notice about the histogram? What does it tell you about the data?

How good of a summary is the mean or median here?

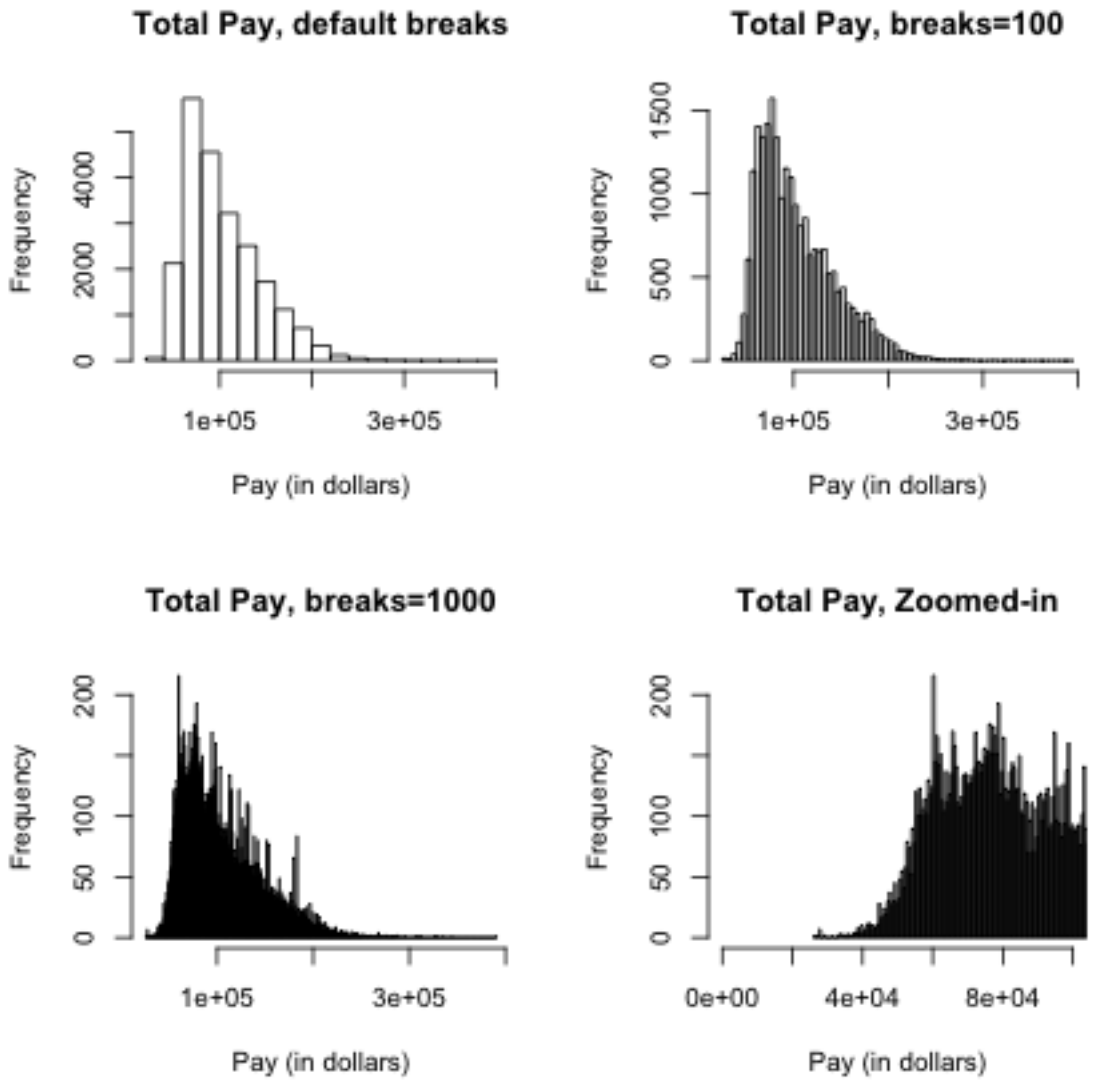
1.1.1 Constructing Frequency Histograms

How do you construct a histogram? Practically, most histograms are created by taking an evenly spaced set of K breaks that span the range of the data, call them $b_1 \leq b_2 \leq ... \leq b_K$, and counting the number of observations in each bin.[2] Then the histogram consists of a series of bars, where the x-coordinates of the rectangles correspond to the range of the bin, and the height corresponds to the number of observations in that bin.

Breaks of Histograms Here's two more histogram of the same data that differ only by the number of breakpoints in making the histograms.

```
par(mfrow = c(2, 2))
hist(salaries2014_FT$TotalPay, main = "Total Pay, default breaks",
    xlab = "Pay (in dollars)")
hist(salaries2014_FT$TotalPay, main = "Total Pay, breaks=100",
    xlab = "Pay (in dollars)", breaks = 100)
hist(salaries2014_FT$TotalPay, main = "Total Pay, breaks=1000",
    xlab = "Pay (in dollars)", breaks = 1000)
hist(salaries2014_FT$TotalPay, main = "Total Pay, Zoomed-in",
    xlab = "Pay (in dollars)", xlim = c(0, 1e+05),
    breaks = 1000)
```

[2] You might have been taught that you *can* make a histogram with uneven break points, which is true, but in practice is rather exotic thing to do. If you do, then you have to calculate the height of the bar differently based on the width of the bin because it is the *area* of the bin that should be proportional to the number of entries in a bin, not the height of the bin.



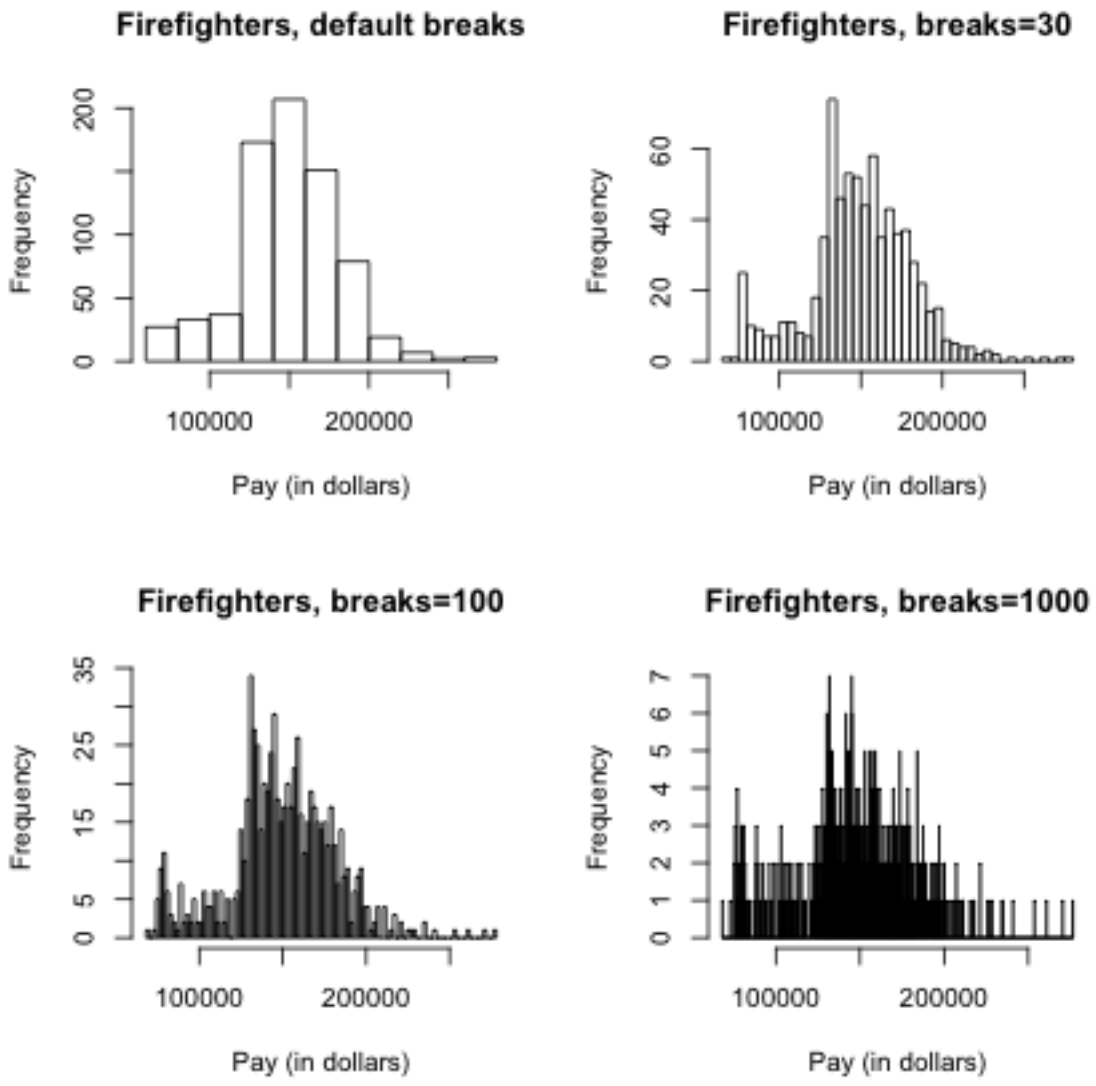
What seems better here? Is there a right number of breaks?

What if we used a subset, say only full-time firefighters? Now there are only 738 data points.

```
salaries2014_FT_FF <- subset(salaries2014_FT, JobTitle ==
    "Firefighter" & Status == "FT")
dim(salaries2014_FT_FF)
```

```
## [1] 738  10
```

```
par(mfrow = c(2, 2))
hist(salaries2014_FT_FF$TotalPay, main = "Firefighters, default breaks",
    xlab = "Pay (in dollars)")
hist(salaries2014_FT_FF$TotalPay, main = "Firefighters, breaks=30",
    xlab = "Pay (in dollars)", breaks = 30)
hist(salaries2014_FT_FF$TotalPay, main = "Firefighters, breaks=100",
    xlab = "Pay (in dollars)", breaks = 100)
hist(salaries2014_FT_FF$TotalPay, main = "Firefighters, breaks=1000",
    xlab = "Pay (in dollars)", breaks = 1000)
```



1.1.2 Density Histograms

The above are called **frequency histograms**, because we plot on the y-axis (the height of the rectangles) the count of the number of observations in each bin. **Density histograms** plot the height of rectangles so that the *area* of each rectangle is equal to the proportion of observations in the bin. If each rectangle has equal width, say w, and there are n total observations, this means for a bin k, it's height is given by

$$w * h_k = \frac{\text{\# observations in bin } k}{n}$$

So that the height of a rectangle for bin k is given by

$$h_k = \frac{\text{\# observations in bin } k}{w \times n}$$

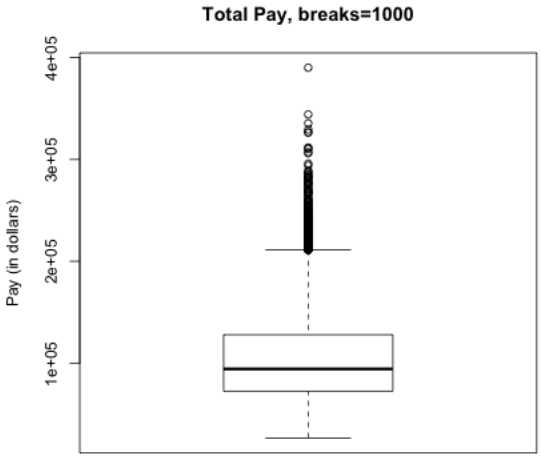
In other words, the *density* histogram with equal-width bins will look like the frequency histogram, only the heights of all the rectangles will be divided by wn.

We will return to the importance of density histograms more when we discuss continuous distributions.

1.2 Boxplots

Another very useful visualization can be a boxplot. A boxplot is like a histogram, in that it gives you a visualization of how the data are distributed. However, it is a much greater simplification of the distribution. It plots only a box for the bulk of the data, where the limits of the box are the 0.25 and 0.75 quantiles of the data (or 25th and 75th percentiles). A dark line across the middle is the median of the data. In addition, a boxplot gives additional information to evaluate the extremities of the distribution. It draws "whiskers" out from the box to indicate how far out is the data beyond the 25th and 75th percentiles. Specifically it calculates the interquartile range (IQR), which is just the difference between the 25th and 75th percentiles. It then draws the whiskers out 1.5 IQR distance from the boxes OR to the smallest/largest data point (whichever is closest to the box). Any data points outside of this range of the whiskers are ploted individually.

```
par(mfrow = c(1, 1))
boxplot(salaries2014_FT$TotalPay, main = "Total Pay, breaks=1000",
    ylab = "Pay (in dollars)")
```



These points are often called "outliers" based the 1.5 IQR rule of thumb. The term **outlier** is usually used for unusual or extreme points. However, we can see a lot of data points fall outside this definition of "outlier" for our data; this is common for data that is skewed, and doesn't really mean that these points are "wrong", or "unusual" or anything else that we might think about for an outlier.[3]

You might think, why would I want such a limited display of the distribution, compared to the wealth of information in the histogram? I can't tell at all that the data is bimodal from a boxplot, for example.

First of all, the boxplot emphasizes different things about the distribution. It shows the main parts of the bulk of the data very quickly and simply, and emphasizes more fine grained information about the extremes ("tails") of the distribution.

Furthermore, because of their simplicity, it is far easier to plot many boxplots and compare them than histograms. For example, I have information of the job title of the employees, and I might be interested in comparing the distribution of salaries with different job titles (firefighters, teachers, nurses, etc). Here I will isolate only those samples that correspond to the top 10 most numerous full-time job titles and do side-by-side boxplots of the distribution within each job title for all 10 jobs.

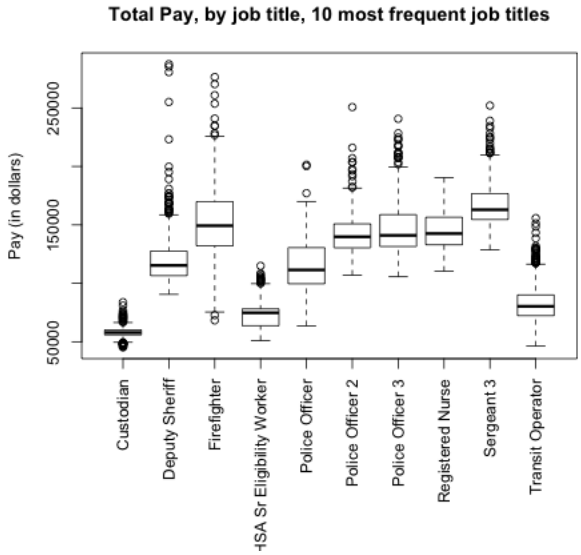
```
tabJobType <- table(subset(salaries2014_FT, Status ==
    "FT")$JobTitle)
tabJobType <- sort(tabJobType, decreasing = TRUE)
topJobs <- head(names(tabJobType), 10)
salaries2014_top <- subset(salaries2014_FT, JobTitle %in%
```

[3]If our data had a nice symmetric distribution around the median, like the normal distribution, the rule of thumb would be more appropriate, and this wouldn't happen to the same degree

```
    topJobs & Status == "FT")
salaries2014_top <- droplevels(salaries2014_top)
dim(salaries2014_top)
```

```
## [1] 5816   10
```

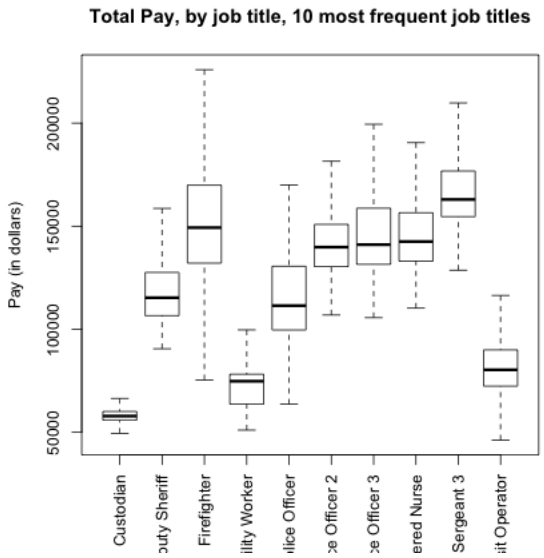
```
par(mar = c(10, 4.1, 4.1, 0.1))
boxplot(salaries2014_top$TotalPay ~ salaries2014_top$JobTitle,
    main = "Total Pay, by job title, 10 most frequent job titles",
    xlab = "", ylab = "Pay (in dollars)", las = 3)
```



This would be hard to do with histograms – we'd either have 10 separate plots, or the histograms would all lie on top of each other. Later on, we will discuss "violin plots" which combine some of the strengths of both boxplots and histograms.

Notice that the outliers draw a lot of attention, since there are so many of them; this is common in large data sets especially when the data are skewed. I might want to mask all of the "outlier" points as distracting for this comparison,

```
boxplot(TotalPay ~ JobTitle, data = salaries2014_top,
    main = "Total Pay, by job title, 10 most frequent job titles",
    xlab = "", ylab = "Pay (in dollars)", las = 3,
    outline = FALSE)
```



1.3 Descriptive Vocabulary

Here are some useful things to consider in describing distributions of data or comparing two different distributions.

Symmetric refers to equal amounts of data on either side of the 'middle' of the data, i.e. the distribution of the data on one side is the mirror image of the distribution on the other side. This means that the median of the data is roughly equal to the mean.

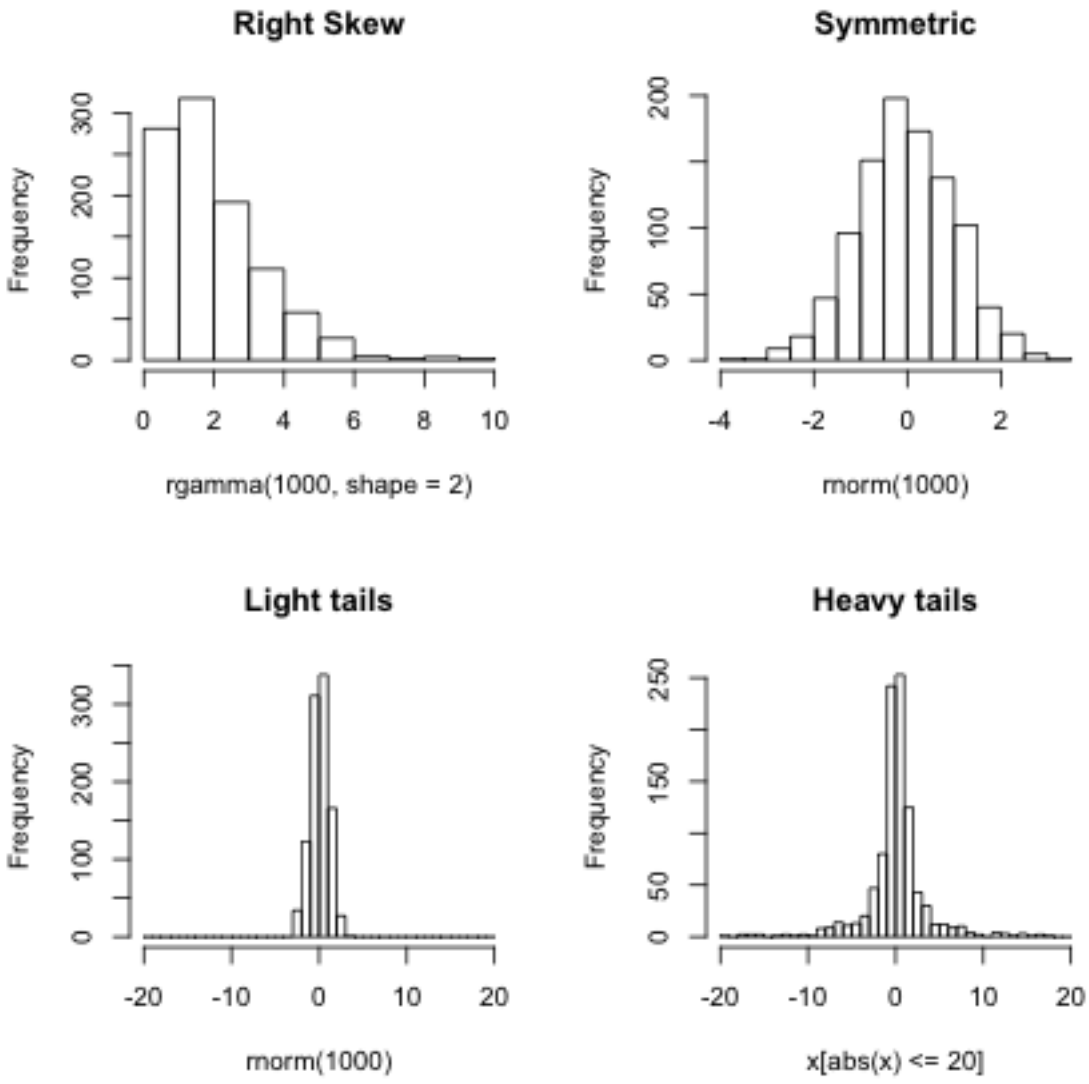
Skewed refers to when one 'side' of the data spreads out to take on larger values than the other side. More precisely, it refers to where the mean is relative to the median. If the mean is much bigger than the median, then there must be large values on the right-hand side of the distribution, compared to the left hand side (**right skewed**), and if the mean is much smaller than the median then it is the reverse.

Spread refers to how spread out the data is from the middle (e.g. mean or median).

Heavy/light tails refers to how much of the data is concentrated in values far away from the middle, versus close to the middle.

```
set.seed(1)
par(mfrow = c(2, 2))
hist(rgamma(1000, shape = 2), main = "Right Skew")
hist(rnorm(1000), main = "Symmetric")
breaks = seq(-20, 20, 1)
```

```
hist(rnorm(1000), main = "Light tails", xlim = c(-20,
    20), breaks = breaks, freq = TRUE)
x <- rcauchy(1000)
hist(x[abs(x) <= 20], main = "Heavy tails", xlim = c(-20,
    20), breaks = breaks, freq = TRUE)
```



As you can see, several of these terms are mainly relevant for comparing two distributions.[4]

[4]But they are often used without providing an explicit comparison distribution; in this case, the comparison distribution is always the normal distribution, which is a standard benchmark in statistics

1.4 Transformations

When we have skewed data, it can be difficult to compare the distributions because so much of the data is bunched up on one end, but our axes stretch to cover the large values that make up a relatively small proportion of the data. This is also means that our eye focuses on those values too.

This is a mild problem with this data, particularly if we focus on the full-time workers, but let's look quickly at another dataset that really shows this problem.

1.4.1 Flight Data from SFO

This data consists of all flights out of San Francisco Airport in 2016 in January (we will look at this data more in the next module).

```
flightSF <- read.table(file.path(dataDir, "SFO.txt"),
    sep = "\t", header = TRUE)
dim(flightSF)
```

```
## [1] 13207    64
```

```
names(flightSF)
```

```
##  [1] "Year"                "Quarter"             "Month"
##  [4] "DayofMonth"          "DayOfWeek"           "FlightDate"
##  [7] "UniqueCarrier"       "AirlineID"           "Carrier"
## [10] "TailNum"             "FlightNum"           "OriginAirportID"
## [13] "OriginAirportSeqID"  "OriginCityMarketID"  "Origin"
## [16] "OriginCityName"      "OriginState"         "OriginStateFips"
## [19] "OriginStateName"     "OriginWac"           "DestAirportID"
## [22] "DestAirportSeqID"    "DestCityMarketID"    "Dest"
## [25] "DestCityName"        "DestState"           "DestStateFips"
## [28] "DestStateName"       "DestWac"             "CRSDepTime"
## [31] "DepTime"             "DepDelay"            "DepDelayMinutes"
## [34] "DepDel15"            "DepartureDelayGroups" "DepTimeBlk"
## [37] "TaxiOut"             "WheelsOff"           "WheelsOn"
## [40] "TaxiIn"              "CRSArrTime"          "ArrTime"
## [43] "ArrDelay"            "ArrDelayMinutes"     "ArrDel15"
## [46] "ArrivalDelayGroups"  "ArrTimeBlk"          "Cancelled"
## [49] "CancellationCode"    "Diverted"            "CRSElapsedTime"
```

```
## [52] "ActualElapsedTime"    "AirTime"              "Flights"
## [55] "Distance"             "DistanceGroup"        "CarrierDelay"
## [58] "WeatherDelay"         "NASDelay"             "SecurityDelay"
## [61] "LateAircraftDelay"    "FirstDepTime"         "TotalAddGTime"
## [64] "LongestAddGTime"
```

This dataset contains a lot of information about the flights departing from SFO. For starters, let's just try to understand how often flights are delayed (or canceled), and by how long. Let's look at the column 'DepDelay' which represents departure delays.

```
summary(flightSF$DepDelay)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   -25.0    -5.0    -1.0    13.8    12.0   861.0     413
```

Notice the NA's. Let's look at just the subset of some variables for those observations with NA values for departure time (I chose a few variables so it's easier to look at)

```
naDepDf <- subset(flightSF, is.na(DepDelay))
head(naDepDf[, c("FlightDate", "Carrier", "FlightNum",
    "DepDelay", "Cancelled")])
```

```
##     FlightDate Carrier FlightNum DepDelay Cancelled
## 44  2016-01-14      AA       209       NA         1
## 75  2016-01-14      AA       218       NA         1
## 112 2016-01-24      AA        12       NA         1
## 138 2016-01-22      AA        16       NA         1
## 139 2016-01-23      AA        16       NA         1
## 140 2016-01-24      AA        16       NA         1
```

```
summary(naDepDf[, c("FlightDate", "Carrier", "FlightNum",
    "DepDelay", "Cancelled")])
```

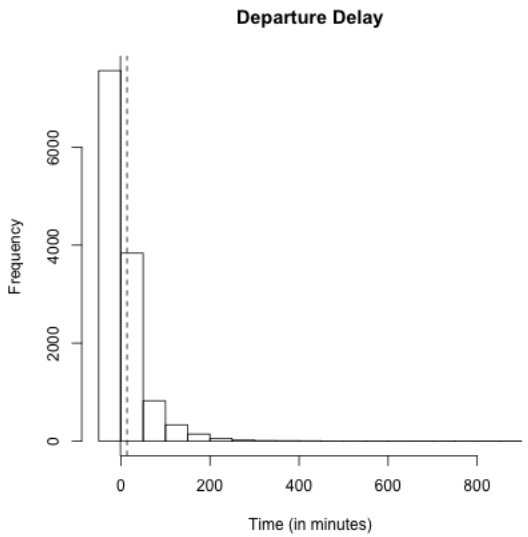
```
##       FlightDate     Carrier      FlightNum       DepDelay     Cancelled
##  2016-01-22: 92   OO     :176   Min.   :   1   Min.   : NA   Min.   :1
##  2016-01-06: 49   UA     : 76   1st Qu.: 616   1st Qu.: NA   1st Qu.:1
##  2016-01-23: 41   WN     : 55   Median :2080   Median : NA   Median :1
```

```
##  2016-01-24: 40   AA     : 35   Mean   :3059   Mean   :NaN   Mean   :1
##  2016-01-19: 31   VX     : 33   3rd Qu.:5555   3rd Qu.: NA   3rd Qu.:1
##  2016-01-03: 27   DL     : 17   Max.   :6503   Max.   : NA   Max.   :1
##  (Other)   :133   (Other): 21                  NA's   :413
```

So, the NAs correspond to flights that were cancelled (Cancelled=1).

Histogram of flight delays Let's draw a histogram of the departure delay.

```
par(mfrow = c(1, 1))
hist(flightSF$DepDelay, main = "Departure Delay", xlab = "Time (in minutes)")
abline(v = c(mean(flightSF$DepDelay, na.rm = TRUE),
    median(flightSF$DepDelay, na.rm = TRUE)), lty = c("dashed",
    "solid"))
```



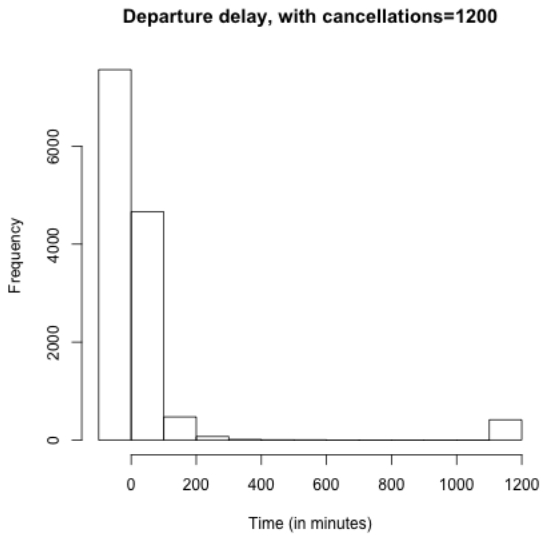
What do you notice about the histogram? What does it tell you about the data?

How good of a summary is the mean or median here? Why are they so different?

Effect of removing data What happened to the NA's? They are just silently not plotted. What does that mean for interpreting the histogram?

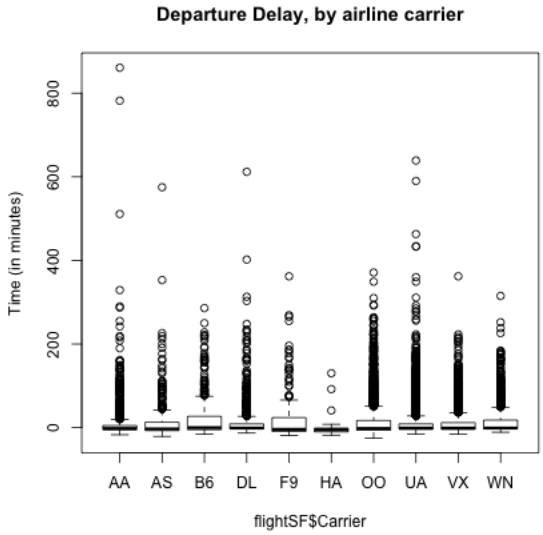
We could give the cancelled data a 'fake' value so that it plots.

```
flightSF$DepDelayWithCancel <- flightSF$DepDelay
flightSF$DepDelayWithCancel[is.na(flightSF$DepDelay)] <- 1200
hist(flightSF$DepDelayWithCancel, xlab = "Time (in minutes)",
    main = "Departure delay, with cancellations=1200")
```



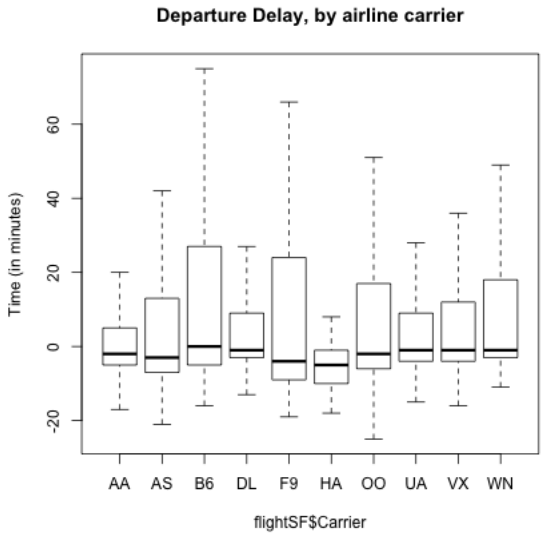
Boxplots If we do boxplots separated by carrier, we can see the problem with the "outlier" points

```
boxplot(flightSF$DepDelay ~ flightSF$Carrier, main = "Departure Delay, by airline car
    ylab = "Time (in minutes)")
```



Here is the same plot suppressing the outlying points:

```
boxplot(flightSF$DepDelay ~ flightSF$Carrier, main = "Departure Delay, by airline car
    ylab = "Time (in minutes)", outline = FALSE)
```



1.4.2 Log and Sqrt Transformations

In data like the flight data, we can remove these outliers for the boxplots to better see the median, etc, but it's a lot of data we are removing – what if the different carriers are actually quite different in the distribution of these outer points? This is

a problem with visualizations of skewed data: either the outlier points dominate the visualization or they get removed from the visualization.

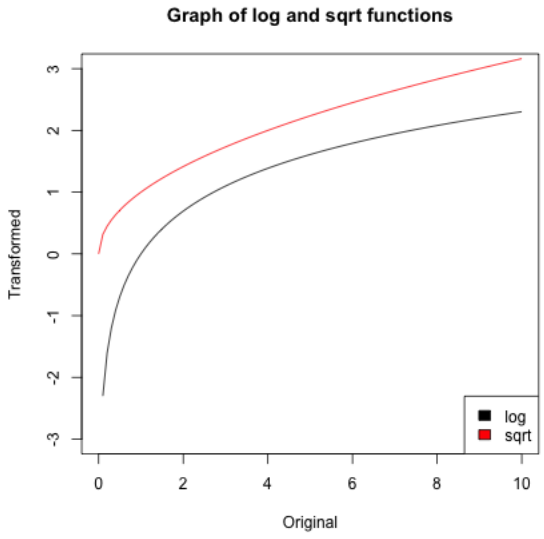
A common way to get around this is to transform our data, which simply means we pick a function f and turn every data point x into $f(x)$. For example, a log-transformation of data point x means that we define new data point y so that

$$y = \log(x).$$

A common example of when we want a transformation is for data that are all positive, yet take on values close to zero. In this case, there are often many data points bunched up by zero (because they can't go lower) with a definite right skew.

Such data is often nicely spread out for visualization purposes by either the log or square-root transformations.

```
ylim <- c(-3, 3)
curve(log, from = 0, to = 10, ylim = ylim, ylab = "Transformed",
    xlab = "Original")
curve(sqrt, from = 0, to = 10, add = TRUE, col = "red")
legend("bottomright", legend = c("log", "sqrt"), fill = c("black",
    "red"))
title(main = "Graph of log and sqrt functions")
```

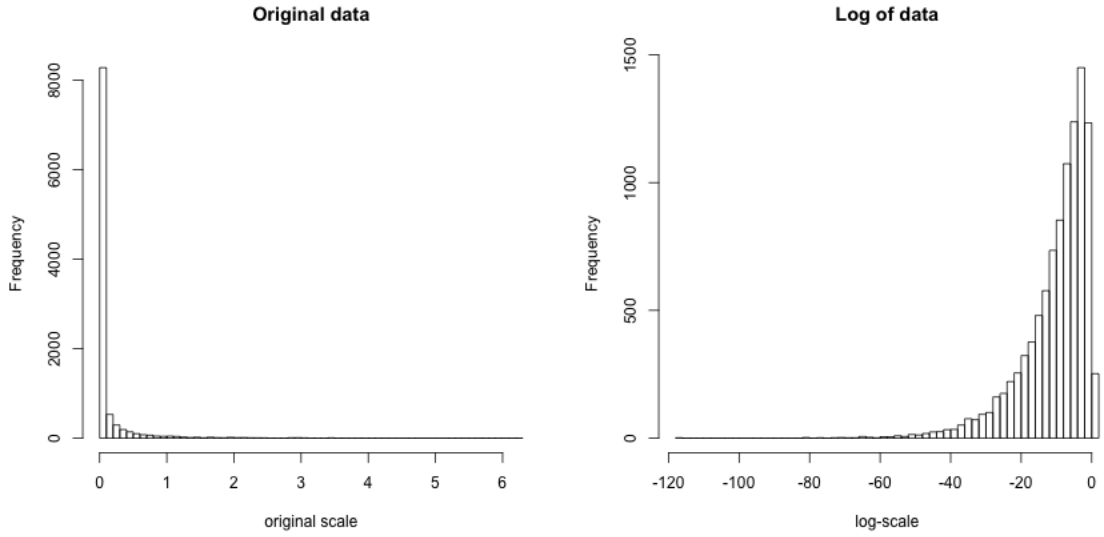


These functions are similar in two important ways. First, they are both *monotone increasing*, meaning that the slope is always positive. As a result, the rankings of the data points are always preserved: if $x_1 > x_2$ then $f(x_1) > f(x_2)$, so the largest data point in the original data set is still the largest in the transformed data set.

The second important property is that both functions are *concave*, meaning that the slope of $f(x)$ gets smaller as f increases. As a result, the largest data points are pushed together while the smallest data points get spread apart. For example, in the case of the log transform, the distance between two data points depends only on their ratio: $\log(x_1) - \log(x_2) = \log(x_1/x_2)$. Before transforming, 100 and 200 were far apart but 1 and 2 were close together, but after transforming, these two pairs of points are equally far from each other. The log scale can make a lot of sense in situations where the ratio is a better match for our "perceptual distance," for example when comparing incomes, the difference between making \$500,000 and \$550,000 salary feels a lot less important than the difference between \$20,000 and \$70,000.

Let's look at how this works with simulated data from a fairly skewed distribution (the Gamma distribution with shape parameter 1/10):

```
y <- rgamma(10000, scale = 1, shape = 0.1)
par(mfrow = c(1, 2))
hist(y, main = "Original data", xlab = "original scale",
    breaks = 50)
hist(log(y), main = "Log of data", xlab = "log-scale",
    breaks = 50)
```



Note that in this case, after transforming the data they are even a bit *left*-skewed because the tiny data points are getting pulled very far apart: $\log(x) = -80$ corresponds to $x = e^{-80} = 1.8 \times 10^{-35}$, and $\log(x) = -40$ to $x = 4.2 \times 10^{-18}$. Still, it is much less skewed than before.

Does it make sense to use transformations? Doesn't this mess-up our data?

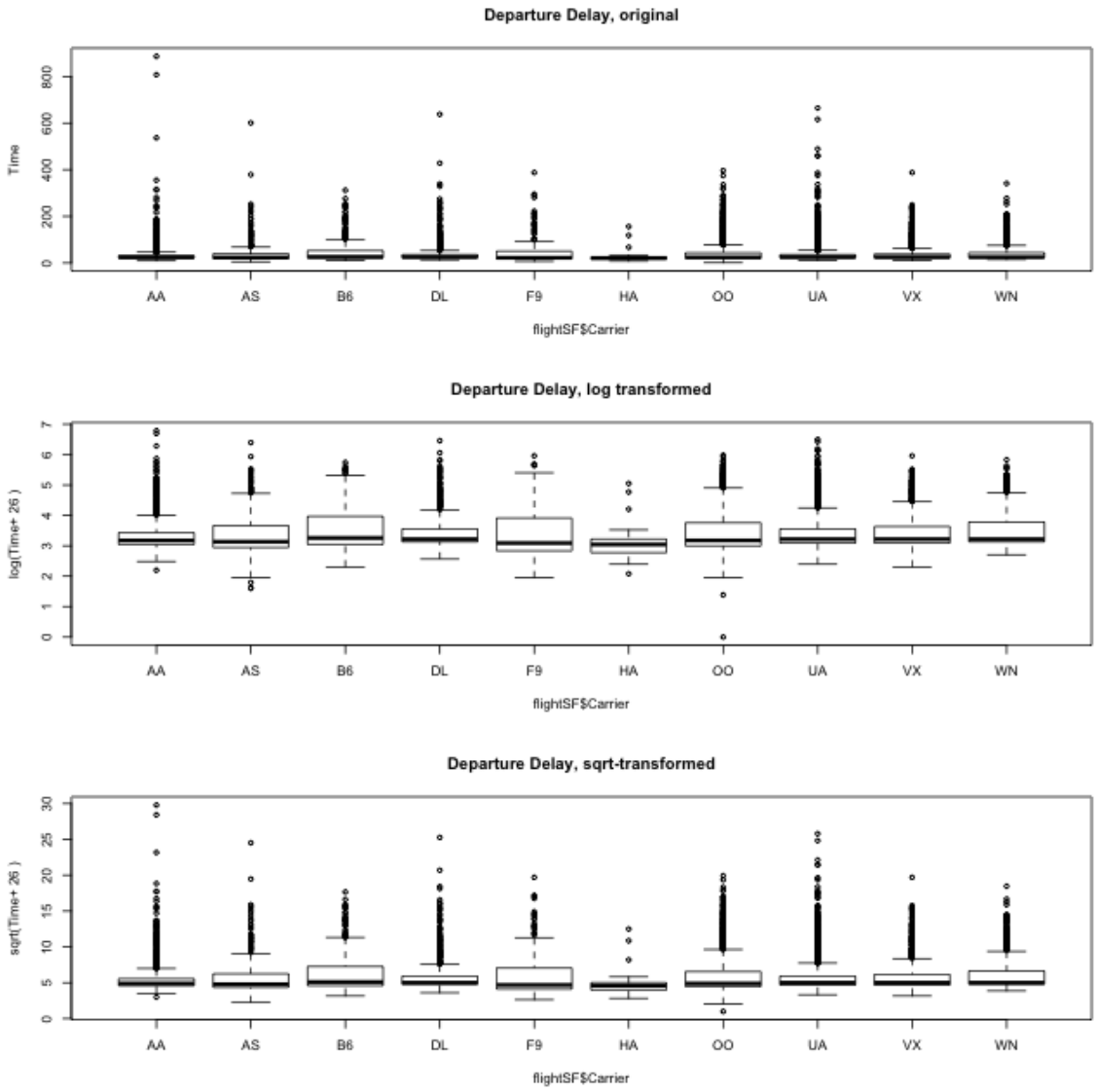
Notice an important property is that these are **monotone** functions, meaning we are preserving the rank of our data – we are not suddenly inverting the relative order of the data. But it does certainly change the meaning when you move to the log-scale. A distance on the log-scale of '2' can imply different distances on the original scale, depending on where the original data was located.[5]

1.4.3 Transforming our data sets

Our flight delay data is not so obliging as the simulated data, since it also has negative numbers. But we could, for visualization purposes, shift the data before taking the log or square-root. Here I compare the boxplots of the original data, as well as that of the data after the log and the square-root.

```
addValue <- abs(min(flightSF$DepDelay, na.rm = TRUE)) +
    1
par(mfrow = c(3, 1))
boxplot(flightSF$DepDelay + addValue ~ flightSF$Carrier,
    main = "Departure Delay, original", ylab = "Time")
boxplot(log(flightSF$DepDelay + addValue) ~ flightSF$Carrier,
    main = "Departure Delay, log transformed", ylab = paste("log(Time+",
        addValue, ")"))
boxplot(sqrt(flightSF$DepDelay + addValue) ~ flightSF$Carrier,
    main = "Departure Delay, sqrt-transformed", ylab = paste("sqrt(Time+",
        addValue, ")"))
```

[5] Of course the distance of '2' on the log-scale *does* have a very specific meaning: a distance of '2' on the (base 10) log scale is equivalent to being 100 times greater



Notice that there are fewer 'outliers' and I can see the differences in the bulk of the data better. Did the data become symmetrically distributed or is it still skewed?

2 Probability Distributions

Let's review some basic ideas of sampling and probability distributions that you should have learned in Data 8/STAT 20.

In the salary data we have *all* salaries of the employees of SF in 2014. This a

census, i.e. a complete enumeration of the entire population of SF employees.

We have data from the US Census that tells us the median household income in 2014 in all of San Fransisco was around \$72K. [6] We could want to use this data to ask, what was the probability an employee in SF makes less than the regional median household number?

We really need to be more careful, however, because this question doesn't really make sense because we haven't defined any notion of randomness. If I pick employee John Doe and ask what is the probability he makes less than \$72K, this is not a reasonable question, because either he did or didn't make less than that.

So we don't actually want to ask about a particular person if we are interested in probabilities – we need to have some notion of asking about a randomly selected employee. Commonly, the randomness we will assume is that a employee is randomly selected from the full population of full-time employees, with all employees having an equal probability of being selected. This is called a **simple random sample**.

Now we can ask, what is the probability of such a randomly selected employee making less than \$72K? Notice that we have exactly defined the randomness mechanism, and so now can calculate probabilities. How would you calculate the following probabilities based on this probability mechanism?

1. $P(\text{income} = \$72K)$
2. $P(\text{income} \leq \$72\text{K})$
3. $P(\text{income} > \$200\text{K})$

This kind of sampling is called a **simple random sample** and is what most people mean when they say "at random" if they stop to think about it. However, there are many other kinds of samples where data are chosen randomly, but not every data point is equally likely to be picked. There are, of course, also many samples that are not random at all.

Notation We call the salary value of a randomly selected employee a **random variable**. We can simplify our notation for probabilities by letting the variable X be short hand for the value of that random variable, and make statements like $P(X > 2)$. We call the complete set of probabilities the **probability distribution** of X.

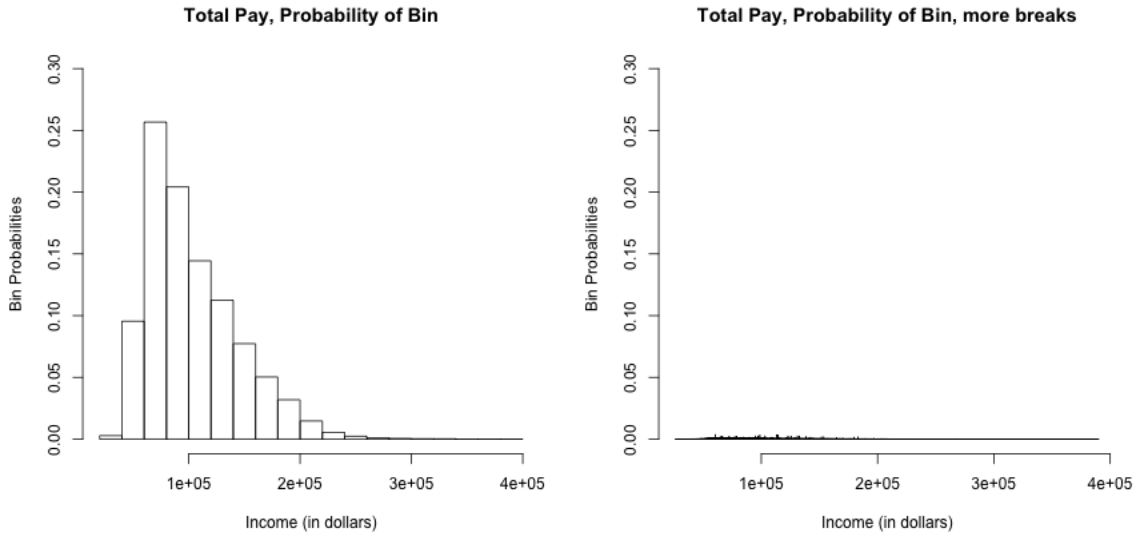
[6] `http://www.hcd.ca.gov/grants-funding/income-limits/state-and-federal-income-limits/docs/inc2k14.pdf`

2.1 Probabilities and Histograms

The **frequency histograms** we plotted of the entire population above give us information about the probabilities of discrete distributions, since they give the count of the numbers of observations in an interval. We can divide that count by the total number of observations, and this gives us the probability of observations lying in each bin.

How would you use the notation above to write this probability, say for the first bin of $(b_1, b_2]$?

I'm going to plot these probabilities for each bin of our histogram, for both large and small size bins. [7]



Be careful, this plot is not the same thing as the density histograms that you have learned – the density value involves the *area* of a bin. For this reason, plotting the bin probabilities as the height of each bar is NOT what is meant by a density histogram.

What happens as I decrease the size of the bins?

[7] Plotting these probabilities is not done automatically by R, so we have to mainpulate the histogram command in R to do this (and I don't normally recommend that you make this plot – I'm just making it for teaching purposes here).

2.2 Considering a subpopulation (Conditioning)

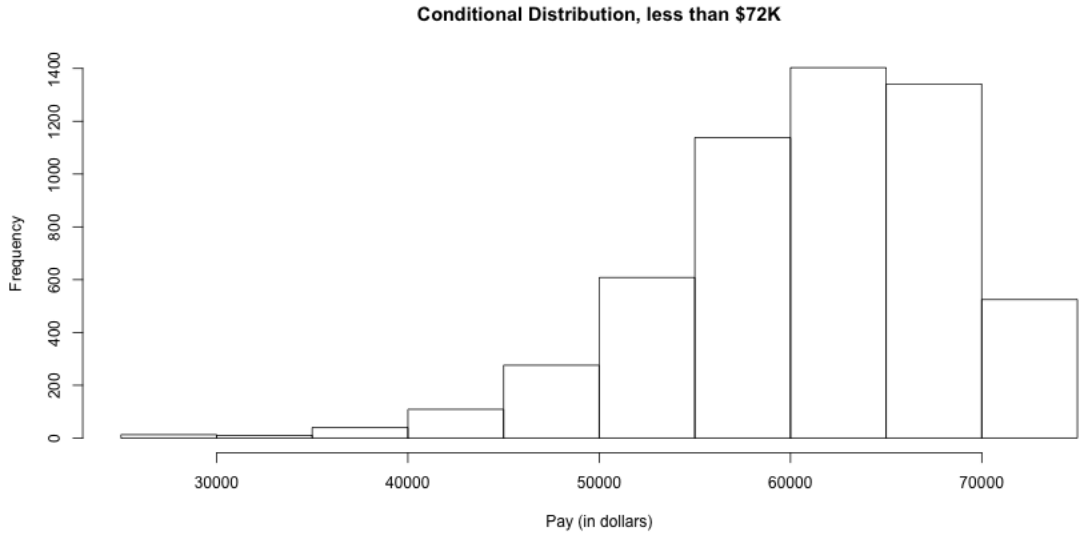
Previously we asked about the population of all FT employees, so that X is the random variable corresponding to income of a randomly selected employee from *that population*. We might want to consider asking questions about the population of employees making less than \$72K. For example, low-income in 2014 for an individual in San Francisco was defined by the same source as \$64K – what is the probability of a random employee making less than \$72K to be considered low income?

We can write this as $P(X \leq 64 \mid X < 72)$, which we say as the probability a employee is low-income *given that* or *conditional on* the employee makes less than the median income. How would we compute a probability like this?.

Note that this is a different probability than $P(X \leq 64)$. How?

Once we condition on a portion of the population, we've actually defined a new random variable. We could call this new random variable Y, but we usually notated it as $X \mid X > 72K$. Since it is a random variable, it has a new probability distribution, which is called the **conditional distribution**. We can plot the histogram of this conditional distribution:

```
condPop <- subset(salaries2014_FT, TotalPay < 72000)
par(mfrow = c(1, 1))
hist(condPop$TotalPay, main = "Conditional Distribution, less than $72K",
    xlab = "Pay (in dollars)")
```



We can think of the probabilities of a conditional distribution as the probabilities we would get if we repeatedly drew X from its marginal distribution but only "keeping" it when we get one with $X < 72K$.

Consider the flight data we looked at briefly above. Let X for this data be the flight delay, in minutes, where if you recall NA values were given if the flight was cancelled.

How would you state the following probability statements in words?

$$P(X > 60 | X \neq \text{NA})$$

$$P(X > 60 | X \neq \text{NA} \& X > 0)$$

3 Distributions of samples of data

Usually the data we work with is a sample, not the complete population. This needs to changes our interpretation of what the plot is doing.

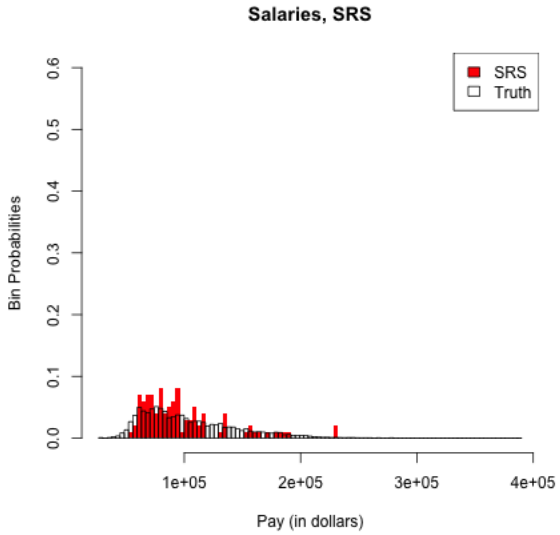
Consider what happens if you take a simple random sample of 100 employees from our complete set of full-time employees and calculate a histogram.

```
salariesSRS <- sample(x = salaries2014_FT$TotalPay,
    size = 100, replace = FALSE)
sample(1:5)
```

```
## [1] 5 2 1 4 3
```

Let's draw a plot giving the proportions of the total sample in each bin (i.e. not a histogram). I'm going to also draw the true population probabilities of being in each bin as well, and put it on the same histogram as the sample proportions. To make sure they are using the same breakpoints, I'm going to define the break points manually. (Otherwise the specific breakpoints will depend on the range of each dataset and so be different)

Pretty good. Suppose I had smaller width breakpoints (next figure), what conclusions would you make?



3.1 Histograms as Estimates and Types of Samples

So when we are working with a sample of data, we should always think of probabilities obtained from a sample as an *estimate* of the probabilities of the full population distribution. This means histograms, boxplots, quantiles, and *any* estimate of a probability calculated from a sample of the full population have variability, like any other estimate.

This means we need to be careful about the dual use of histograms as both visualization tools and estimates. As visualization tools, they are always appropriate for understanding *the data you have*: whether it is skewed, whether there are outlying or strange points, what are the range of values you observe, etc.

To draw broader conclusions from histograms or boxplots performed on a sample, however, is by definition to view them as estimates of the entire population. In this case you need to think carefully about how the data was collected.

3.1.1 Different Types of Samples

For example, let's consider that I want to compare the salaries of fire-fighters and teachers in all of California. To say this more precisely for data analysis, I want to see how similar are the distribution of salaries for fire-fighters to that of teachers in 2014 in California. Consider the following *samples* of data

- All salaries in San Franscisco (the data we have)

- A simple random sample drawn from a list of all employees in all localities in California.
- A separate simple random samples drawn from every locality, combined together into a single dataset

Why do I now consider all salaries in San Franscisco as a sample, when before I said it was a census?

All three of these are samples from the population of interest and for simplicity let's assume that we make them so that they are all same total sample size.

One is *not* a *random sample* (which one?). Only one is a *simple random sample* . The last sampling scheme, created by doing a SRS of each month and combining the results, is also a random sampling scheme. We know it's random because if we did it again, we wouldn't get exactly the same set of data (unlike our SF data). But it is not a SRS – it is called a **Stratified random sample**.

If we draw histograms of these different samples, they will all describe the observed distribution of *the sample we have*, but they will not all be good estimates of the underlying population distribution.

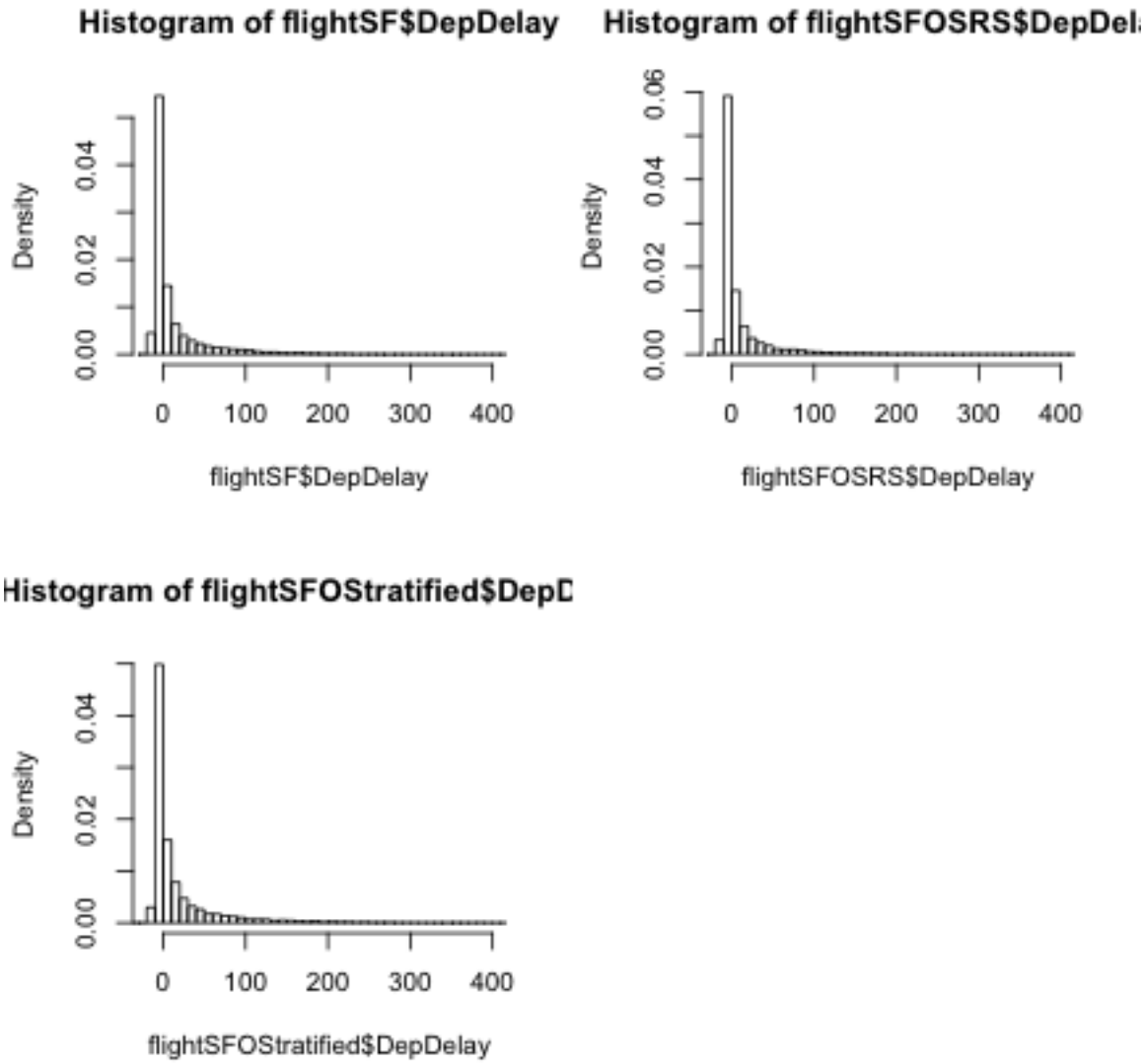
3.1.2 Example on Data

We don't have this data, but we do have the full year of flight data in 2015/2016 academice year (previously we imported only the month of January). Consider the following ways of sampling from the full set of flight data and consider how they correspond to the above:

- 12 separate simple random samples drawn from every month in the 2015/2016 academic year, combined together into a single dataset
- All flights in January
- A simple random sample drawn from all flights in the 2015/2016 academic year.

We can actually make all of these samples and compare them to the truth (I've made these samples previously and I'm going to just read them, because the entire year is a big dataset to work with in class).

```
flightSFOSRS <- read.table(file.path(dataDir, "SFO_SRS.txt"),
    sep = "\t", header = TRUE, stringsAsFactors = FALSE)
flightSFOStratified <- read.table(file.path(dataDir,
    "SFO_Stratified.txt"), sep = "\t", header = TRUE,
    stringsAsFactors = FALSE)
par(mfrow = c(2, 2))
xlim <- c(-20, 400)
hist(flightSF$DepDelay, breaks = 100, xlim = xlim,
    freq = FALSE)
hist(flightSFOSRS$DepDelay, breaks = 100, xlim = xlim,
    freq = FALSE)
hist(flightSFOStratified$DepDelay, breaks = 100, xlim = xlim,
    freq = FALSE)
```



How do these histograms compare?

In particular, drawing histograms or estimating probabilities from data as we have done here only give good estimates of the population distribution *if the data is a SRS.* Otherwise they can vary quite dramatically from the actual population.

So are only SRS good random samples? NO! The stratified random sample described above can actually be a much better way to get a random sample and give you *better* estimates – but you must correctly create your estimates.

For the case of the histogram, you have to estimate the histogram in such a way that it correctly estimates the distribution of population, rather than the distribution of the sample. How? The key thing is that because it is a random sample, drawn according to a *known probability mechanism*, it is possible to make a correct estimate of the population.

How to make these kind of estimates for random samples that are not SRS is beyond the scope of this class, but there are standard ways to do so for stratified samples and many other sampling designs (this field of statistics is called *survey sampling*). Indeed most national surveys, particularly any that require face-to-face interviewing, are not SRS but much more complicated sampling schemes that can give equally accurate estimates, but often with less cost.

4 Continuous Distributions

Data 8 and Stat 20 primarily relied on probability from **discrete distributions**, meaning that the complete set of possible values that can be observed is a finite set of values. For example, if we draw a random sample from our salary data we know that only the 35711 unique values of the salaries in that year can be observed – not all numeric values are possible. We saw this when we asked what was the probability that we drew a random employee with salary exactly equal to \$72K.

However, it can be useful to think about probability distributions that allow for all numeric values (i.e. continuous values), *even when we know the actual population is finite.* These are **continuous distributions**.

For example, suppose we wanted to use this set of data to make decisions about policy to improve salaries for a certain class of employees. It's more reasonable to think that there is an (unknown) probability distribution that defines what we expect

to see for that data that is defined on a continuous range of values, not the specific ones we see in 2014.

Of course some features of the data are "naturally" discrete, like the set of job titles, and there no rational way to think of them being continuous.

4.1 Probability with Continuous distributions

Some probability ideas become more complicated/nuanced for continuous distributions. In particular, for a discrete distribution, it makes sense to say $P(X = 72K)$ (the probability of a salary exactly equal to $72K$). For continuous distributions, such an innocent statement is actually fraught with problems.

To see why, remember what you know about discrete probability distributions. In particular,

$$0 \leq P(X = 72,000) \leq 1$$

Furthermore, any probability statment has to have this property, not just ones involving '=': e.g. $P(X \leq 10)$ or $P(X \geq 0)$. This is a fundamental rule of probability, and thus also holds true for continuous distributions.

Okay so far. Now another thing you learned is if I give all possible values that my random variable X can take (the **sample space**) and call them $v_1, \ldots, v_K$, then if I sum up all these probabilities they must sum exactly to 1,

$$\sum_{i=1}^{K} P(X = v_i) = 1$$

Furthermore, $P(X \in \{v_1, \ldots, v_K\}) = 1$, i.e. the probability X is in the sample space must of course be 1.

Well this becomes more complicated for continuous values – this leads us to an infinite sum since we have an infinite number of possible values. Moreover, if we give *any* positive probability (i.e. $\neq 0$) to each point in the sample space, then we won't 'sum' to one [8] These kinds of concepts from discrete probability just don't translate over exactly to continuous random variables.

To deal with this, *continuous distributions do not allow any positive probability for a single value*: if X has a continuous distribution, then $P(X = x) = 0$ for any value of x.

[8] For those with more math: convergent infinite series can of course sum to 1. But we are working with the continuous real line (or an interval of the real line), and there is not a bijection between the integers and the continuous line.

Notation: Notice the notation here. We generally use a capital letter, like X for random variables, and lower case value for a particulary possible value that they can take on (a value that it takes on is also called a **realization**). We often use the same letter, with one lower-case and one upper case, as is done here. Why? Otherwise we start to run out of letters and symbols once we have multiple random variables – we don't want statements like $P(W = v, X = y, Z = u)$ because it's hard to remember which value goes with which random variable.

Instead, continuous distributions only allow for positive probability of an interval: $P(x_1 \leq X \leq x_2)$ *can* be greater than 0.

Note that this also means that for continuous distributions $P(X \leq x) = P(X < x)$, why?

Giving zero probability for a single value isn't so strange if you think about it. Think about our flight data. What is your intuitive sense of the probability of a flight delay of exactly 10 minutes – and not 10 minutes 10 sec or 9 minutes 45 sec? You see that once you allow for infinite precision, it is actually reasonable to say that *exactly* 10 minutes has no real probability that you need worry about.

For our salary data, of course we don't have infinite precision, but we still see that it's useful to think of ranges of salary – there is no one that makes exactly \$72K, but there are 1 within \$1 dollar of that amount, and 6 employees within \$10 dollars of that amount, all equivalent salaries in any practical discussion of salaries.

What if you want the chance of getting a 10 minute flight delay? Well, you really mean a small interval around 10 minutes, since there's a limit to our measurement ability anyway. This is what we also do with continuous distributions: we discuss the probability in terms of increasingly small intervals around 10 minutes.

The mathemtatics of calculus give us the tools to do this via integration. In practice, the functions we want to integrate are not tractable anyway, so we will use the computer. We are going to focus on understanding how to think about continuous distributions so we can understand the statistical question of how to *estimate* distributions and probabilities (rather than the more in-depth probability treatment you would get in a probability class).

4.2 Cummulative Distribution Function (cdfs)

For discrete distributions, we can *completely* describe the distribution of a random variable by describing the probability of each of the discrete values it takes on. In

other words, knowing $P(X = v_i)$ for all possible values of v_i in the sample space completely defines the probability distribution.

If we can't talk about $P(X = x)$, then how do we define a continous distribution? We basically define the probably of every single possible *interval*. Obviously, there are an infinite number of intervals, but we can use the simple fact that

$$P(x_1 < X \leq x_2) = P(X \leq x_2) - P(X \leq x_1)$$

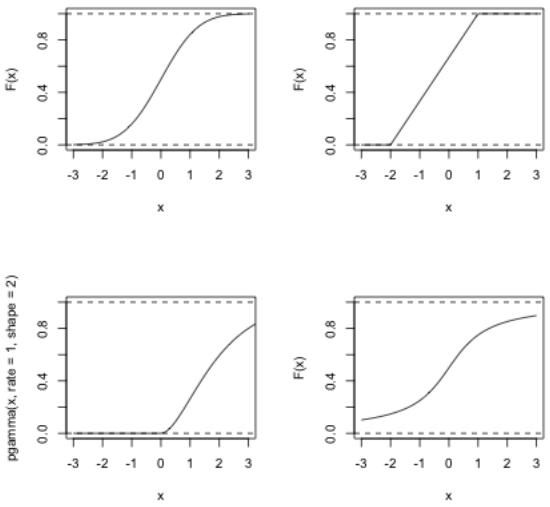
Why? (Use the case of discrete distribution to reason it out)

Thus rather than define the probably of every single possible interval, we can tackle the simpler task to define $P(X \leq x)$ for every single x on the real line. That's just a function of x

$$F(x) = P(X \leq x)$$

F is called a **cumulative distribution function (cdf)**. And while we will focus on continuous distributions, discrete distributions can also be defined in the same way by their cumulative distribution function.

Here are some illustrations of different F functions for x between -3 and 3:



Which of these distributions is likely to have values of X less than -3?

Which equally likely to be positive or negative?

What is the $P(X > 3)$ – how would you calculate that? Which distributions are likely to have values greater than 3?

What is $\lim_{x\to\infty} F(x)$ for all cdfs? What is $\lim_{x\to-\infty} F(x)$ for all cdfs? Why?

Key properties of continuous distributions

1. Probabilities are always between 0 and 1, inclusive.
2. Probabilities are only calculated for intervals, not individual points

4.3 Probability Density Functions (pdfs)

You see from these questions, that you can make all of the assessments we have discussed (like symmetry, or compare if a distribution has heavier tails than another) from the cdf. But it is not the most common way to think about the distribution. More frequently the **probability density function (pdf)** is more intuitive, and is similar to a histogram in the information it gives about the distribution.

Formally, the pdf $p(x)$ is derivative of $F(x)$, if $F(x)$ is differentiable

$$p(x) = \frac{d}{dx}F(x)$$

If F isn't differentiable, the distribution doesn't have a density, which in practice you will rarely run into for continuous variables.[9]

Conversely, $p(x)$ is the function such that if you take the area under its curve for an interval, i.e. the integral, it gives you probability of that interval:

$$\int_a^b p(x) = P(a \leq X \leq b) = F(b) - F(a)$$

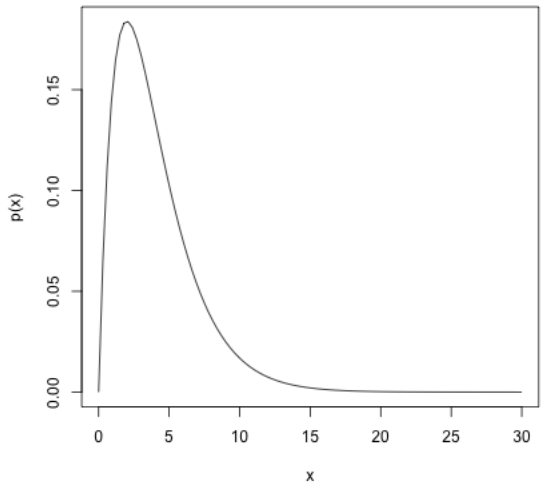
More formally, you can derive $P(X \leq v) = F(v)$ from $p(x)$ as

$$F(v) = \int_{-\infty}^{v} p(x)dx.$$

[9]Discrete distributions have cdfs where $F(x)$ is not differentiable, so they do not have densities. But even some continuous distributions can have cdfs that are non-differentiable

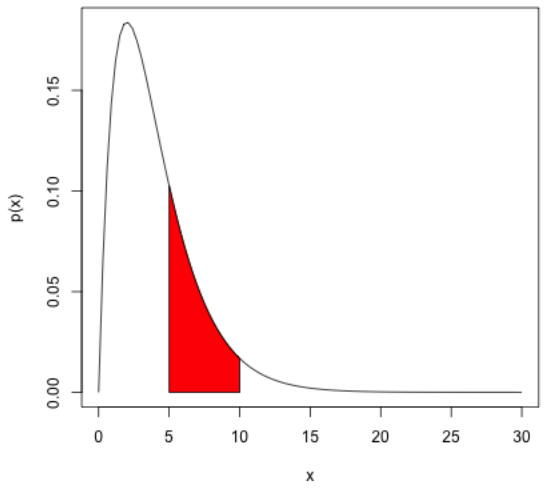
Let's look at an example with the following pdf, which is perhaps vaguely similar to our flight or salary data, though on a different scale of values for X,

$$p(x) = \frac{1}{4} x e^{-x/2}$$



Suppose that X is a random variable from a distribution with this pdf. Then to find $P(5 \leq X \leq 10)$, I find the area under the curve of $p(x)$ between 5 and 10, by taking the integral of $p(x)$ over the range of $(5, 10)$:

$$\int_5^{10} \frac{1}{4} x e^{-x/2}$$



In this case, we can actually solve the integral through integration by parts (which you may or may not have covered),

$$\int_5^{10} \frac{1}{4} x e^{-x/2} = (-\frac{1}{2} x e^{-x/2} - e^{-x/2}) \Big|_5^{10} =$$

Evaluating this gives us $P(5 \leq X \leq 10) = 0.247$. Most of the time, however, the integrals of common pdfs that are used as models for data (like the normal), cannot be done by hand, and we rely on the computer to evaluate the integral for us.

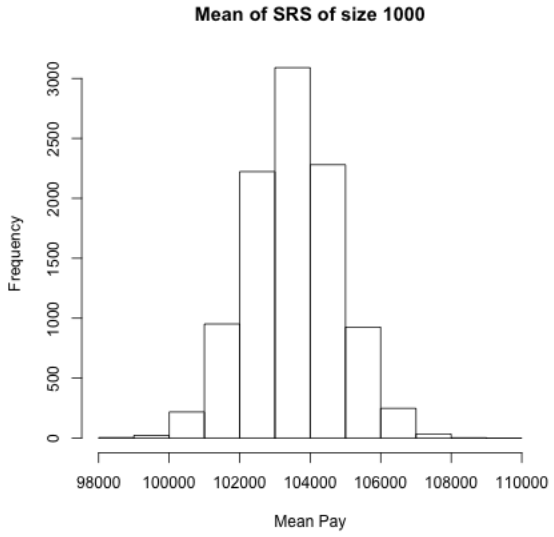
Total Probability Our same rule from discrete distribution applies, namely that the probability of X being in the entire sample space must be 1. Here the sample space is the whole real line. What does this mean in terms of the cumulative area under the curve of $p(x)$?

4.4 Normal Distribution and Central Limit Theorem

You've seen a continuous distribution when you learned about the central limit theorem.

Recall, if I take a SRS of a population and calculate it's mean, call it $\bar{X}$, this is itself a random variable that has a distribution. It's randomness is due to the randomness in the SRS. If I do this process many times I can look at the distribution of $\bar{X}$

```
sampleSize <- 1000
sampleMean <- replicate(n = 10000, expr = mean(sample(salaries2014_FT$TotalPay,
    size = sampleSize, replace = TRUE)))
hist(sampleMean, xlab = "Mean Pay", main = paste("Mean of SRS of size",
    sampleSize))
```



If the size of the sample is large enough, the distribution (i.e. histogram) of $\bar{X}$ will look like a bell-shaped curve. The central limit theorem tells us that for large sample sizes, this always happens, *regardless of the original distribution of the data*. This bell-shaped curve is called the *normal distribution*. Because of the CLT – and because many natural estimates are means of one form or another – the normal is a key distribution for statistics.

A normal distribution has two **parameters** that define the distribution: its mean μ and variance σ^2 (recall the variance is the standard deviation squared). It's pdf is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

It's a mouthful, but easy for a computer to evaluate.[10]

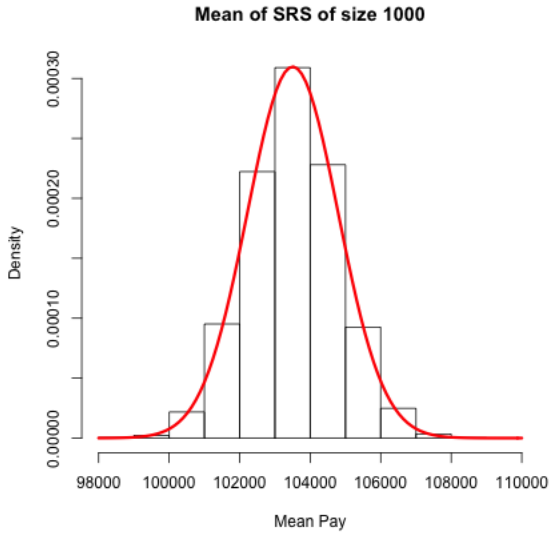
Then the central limit theorem says that if the original distribution has mean μ_{true} and variance τ^2_{true}, then the distribution of $\bar{X}$ for a sample of size n will be approximately

$$N(\mu_{true}, \frac{\tau^2_{true}}{n})$$

Back to the Salary data We can overlay the normal distribution on our histogram, if we draw a density histogram (i.e. scale the frequencies so that the area in the rectangles sums to 1). Notice we also have to pick the right mean and standard deviation for our normal distribution for these to align. How?

[10] It's cdf – the integral of this equation – is intractable, but again easy for a computer to approximate to arbitrarily good precision.

For most actual datasets, of course, we don't know the true mean of the population, but since we sampled from a known population we do.



Probabilities of a normal distribution Recall that for a normal distribution, the probability of being within 1 standard deviation of μ is roughly 0.68 and the probability of being within 2 standard deviations of μ is roughly 0.95.

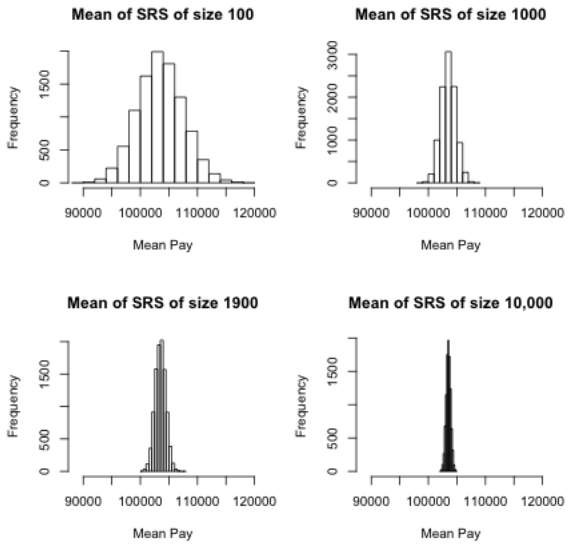
What is the probability that a observed random variable from a $N(\mu, \sigma^2)$ distribution is *less* than μ by more than 2σ?

For $\bar{X}$, which is approximately normal, if the original population had mean μ and standard deviation τ, the standard deviation of that normal is $\tau/\sqrt{n}$. What does this mean for the chance of a single mean calculated from your data being far from the true mean (relate your answer to the above information about probabilities in a normal)?

Improvement with larger n We generally want to increase the sample size to be more accurate. What does this mean and why does this work? The mean $\bar{X}$ we observe in our data will be a random, single observation. If we could collect our data over and over again, we know that $\bar{X}$ will fluctuates around the truth for different samples. If we're lucky, τ is small, so that variability will be small, so any particular sample (like the one we get!) will be close to the mean. But we can't control τ. We

can (perhaps) control the sample size, however – we can gather more data. The CLT tells us that if we have more observations, n, the fluctations of the mean $\bar{X}$ from the truth will be smaller and smaller for larger n – meaning the particular mean we observe in our data will be closer and closer to the true mean. So means with large sample size should be more accurate.

However, there's a catch, in the sense that the amount of improvement you get with larger n gets less and less for larger n. If you go from n observations to $2n$ observations, the standard deviation goes from $\frac{\tau_{true}}{\sqrt{n}}$ to $\frac{\tau_{true}}{\sqrt{2n}}$ – a decrease of $1\sqrt{2}$. In other words, the standard deviation decreases as n decreases like $1/\sqrt{n}$.



4.5 More on density curves

"Not much good to me" you might think – you can't evaluate $p(x)$ and get any probabilities out. It just requires the new task of finding an area. However, finding areas under curves is a routine integration task, and even if there is not a analytical solution, the computer can calculate the area. So pdfs are actually quite useful.

Moreover, $p(x)$ *is* interpretable, just not as a direct tool for probability calculations. For smaller and smaller intervals you are getting close to the idea of the "probability" of $X = 72K$. For this reason, where discrete distributions use $P(X = 72K)$, the closest corresponding idea for continuous distributions is $p(72,000)$: though $p(72,000)$ is not a probability like $P(X = 72,000)$ the value of $p(x)$ gives you an idea of more likely regions of data.

More intuitively, the curve $p(x)$ corresponds to the idea of of a histogram of data.

It's shape tells you about where the data are likely to be found, just like the bins of the histogram. We see for our example of $\bar{X}$ that the histogram of $\bar{X}$ (when properly plotted on a density scale) approaches the smooth curve of a normal distribution. So the same intuition we have from the discrete histograms carry over to pdfs.

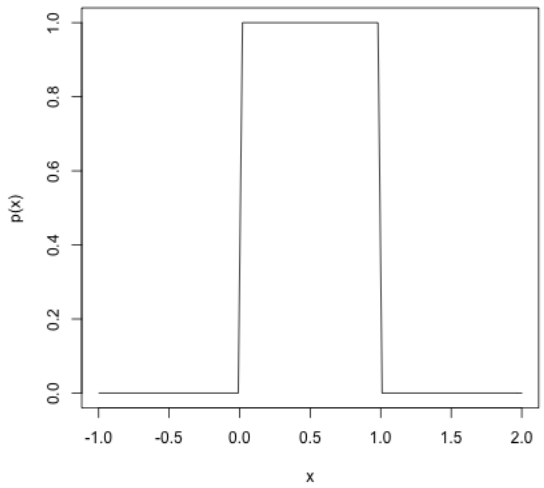
Properties of pdfs

1. A probability density function gives the probability of any interval by taking the area under the curve
2. The total area under the curve $p(x)$ must be exactly equal to 1
3. Unlike probabilities, the value of $p(x)$ can be ≥ 1 (!).

This last one is surprising to people, but $p(x)$ is not a probability – only the area under it's curve is a probability.

To understand this, consider this very simple density function:

$$p(x) = \begin{cases} 1 & x \in [0,1] \\ 0 & x > 1, x < 0 \end{cases}$$

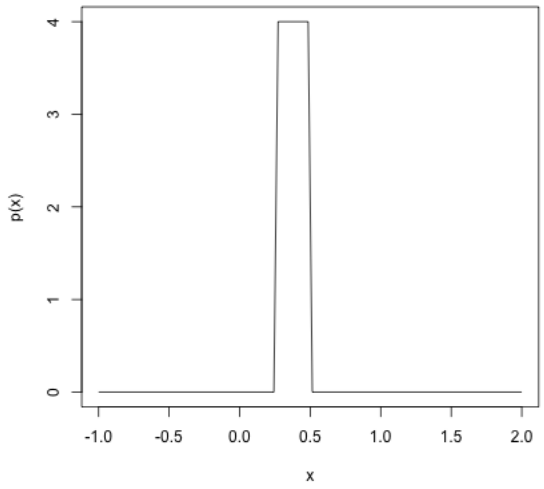


This is a density function that corresponds to being equally likely for any value between 0 and 1; why?

What is the area under this curve? Well it's just a rectangle, so...

This distribution is called a *uniform distribution* on [0,1], some times abbreviated $U(0,1)$.

Suppose instead, I want density function that corresponds to being equally likely for any value between 1/4 and 1/2 (i.e. $U(1/4, 1/2)$).



Then again, we can easily calculate this area . If $p(x)$ was required to be less than one, you couldn't get the total area to be 1.

So you see that the scale of values that X takes on matters to the value of $p(x)$. If X is concentrated on a small interval, then the density function will be quite large, while if it is diffuse over a large area the value of the density function will be small.

Example: Changing the scale of measurements : Suppose my random variable X are measurements in centimeters, with a normal distribution, $N(\mu = 100\text{cm}, \sigma^2 = 100\text{cm}^2)$. What is the standard deviation?

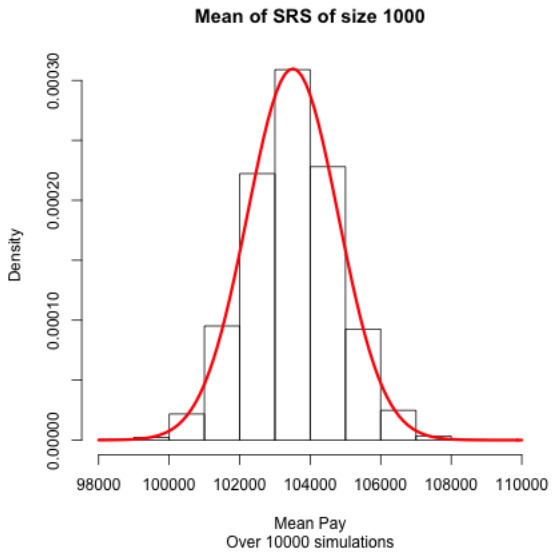
Then I decide to convert all the measurements to meters (FYI: 100 centimeters=1 meter). What is now the mean? And standard deviation?

4.5.1 Density Histograms Revisited

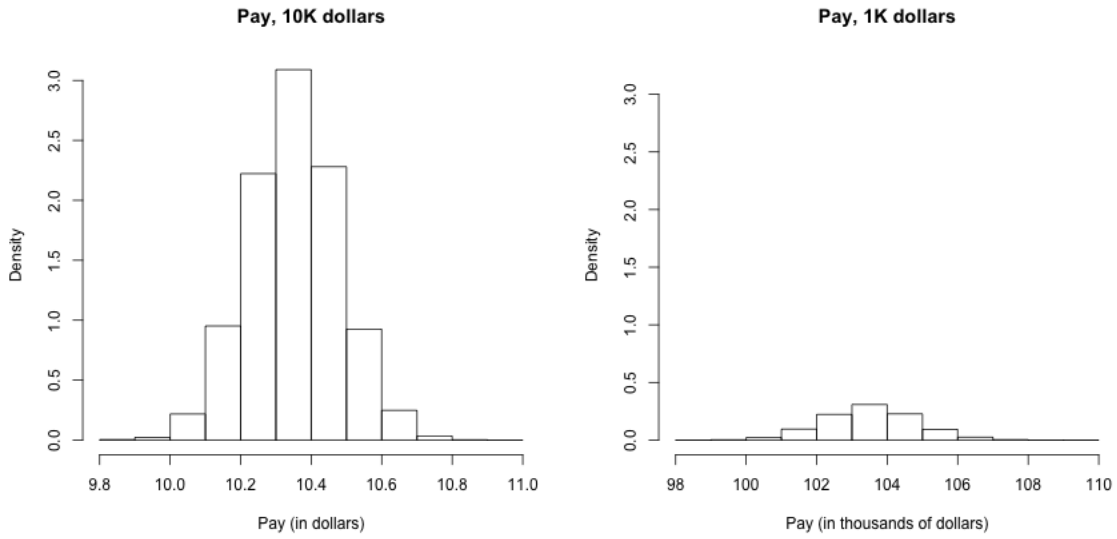
We've been showing histograms with the frequency of counts in each bin on the y-axis. But, histograms are actually meant to represent the distribution of continuous measurements, i.e. to approximate density functions. Specifically, histograms are properly drawn on the density scale, meaning that you want the total area in all of the rectangles of the histogram to have area 1.

Notice how when I overlay the normal curve for discussing the central limit theorem, I had to set my `hist` function to `freq=FALSE` to get proper density histograms. Otherwise the histogram is on the wrong scale.

```
hist(sampleMean, xlab = "Mean Pay", main = paste("Mean of SRS of size",
    sampleSize), freq = FALSE, sub = paste("Over",
    length(sampleMean), "simulations"))
m <- mean(salaries2014_FT$TotalPay)
s <- sqrt(var(salaries2014_FT$TotalPay)/sampleSize)
p <- function(x) {
    dnorm(x, mean = m, sd = s)
}
curve(p, add = TRUE, col = "red", lwd = 3)
```

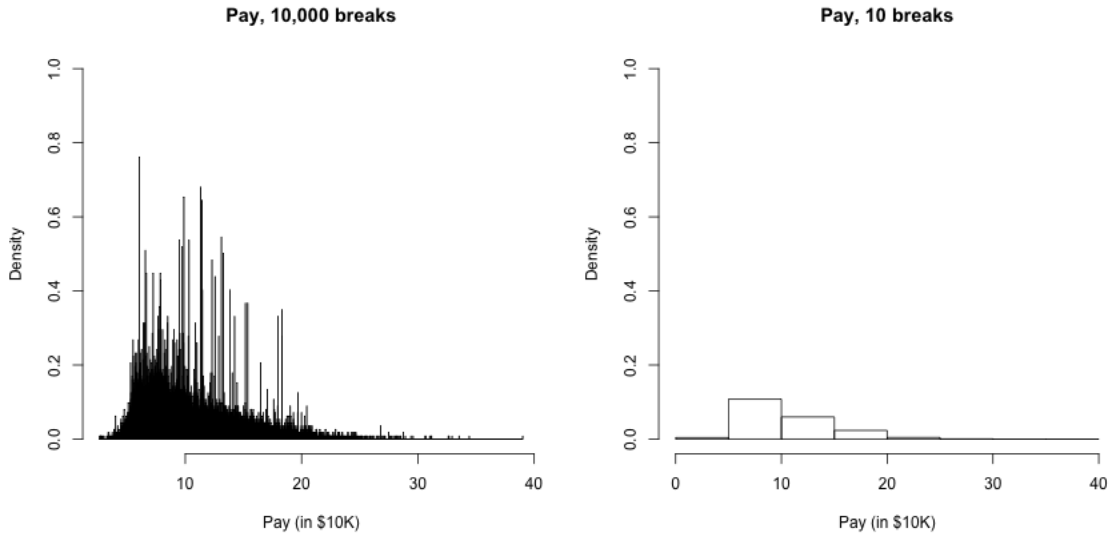


We can demonstrate the effect of the scale of the data on this density histogram by changing the scale that we measure to be in units of 10K rather than say 1K.



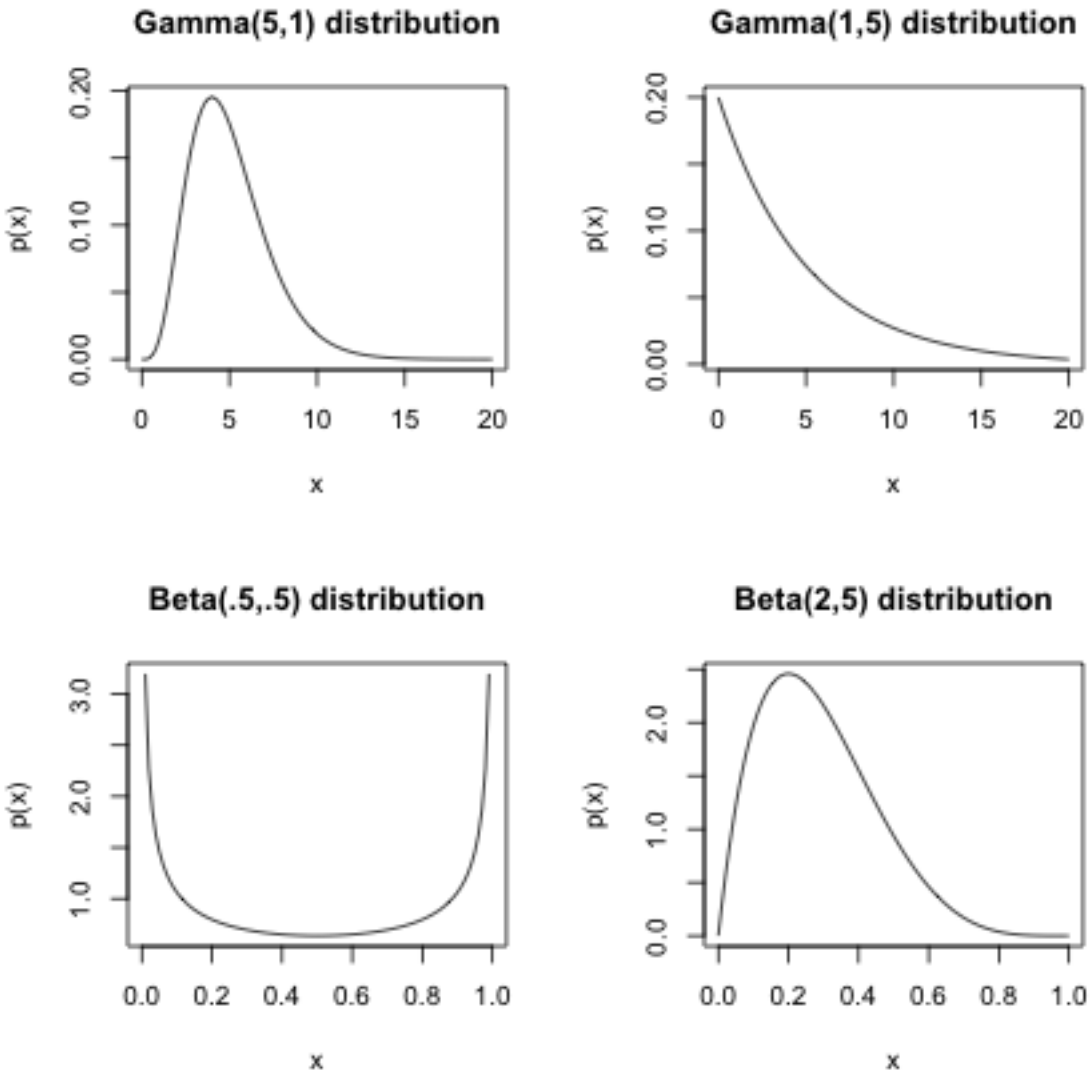
Just like density curves, if you plot histograms on the density scale, you can get values greater than 1.

Notice how density values vary (like counts) as you change the breaks. Why?



4.5.2 Examples of other distributions

Here are some examples of some pdfs from some two common continuous distributions other than the normal:



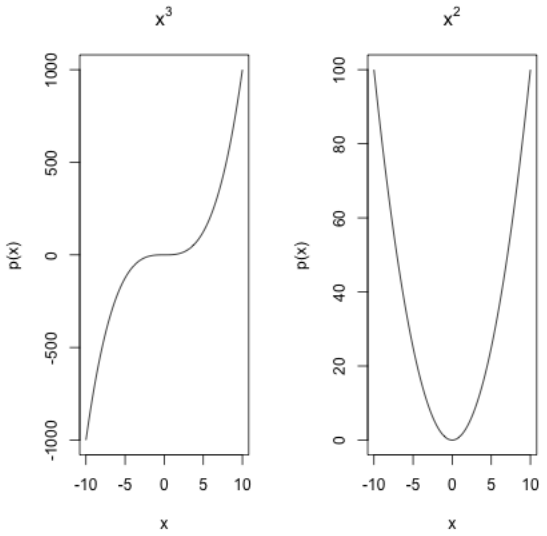
These are all called **parametric distributions**. Notice a few things illustrated by these examples:

- that 'a' parametric distribution is actually a family of distributions that differ by changing the **parameters** (e.g. Normal has a mean and a standard deviation that defines it)
- Unlike the normal, many distributions have very different shapes for different

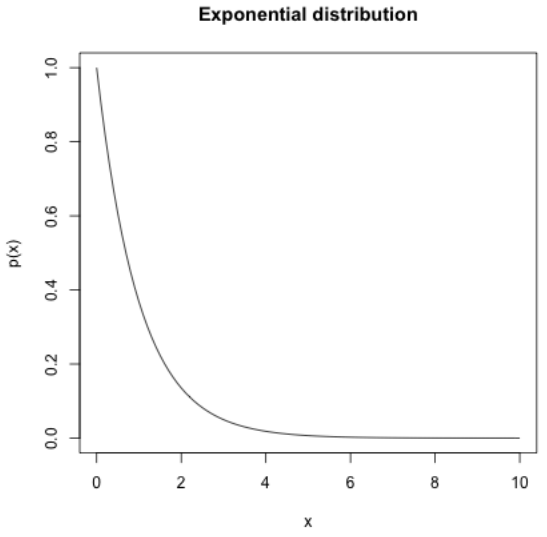
parameters

- Continuous distributions can be limited to an interval or region (i.e. not take on all values of the real line). They are still considered continuous distributions because the range of points with positive probability is still a continuous range.

The following cannot be pdfs, why?



But be careful. Just because a function $p(x)$ goes to infinity (i.e. is unbounded), doesn't mean that it can't be a probability density!



5 Density Curve Estimation

We've seen that histograms can approximate density curves (by making the area in the histogram sum to 1). If we have data from a continuous distribution, we are estimating a pdf, so we would want an estimate that is written as a function, say $\hat{p}(x)$.

5.1 Histogram as estimate of pdf

So if we don't know $p(x)$ but have a SRS from the distribution and we want to estimate $p(x)$.

Let's think of an easy situation. Suppose that we want to estimate $p(x)$ between the values b_1, b_2, and that in that region, $p(x)$ is constant, i.e. a flat line

Then what do we know about $P(b_1 \leq X \leq b_2)$?

We have a good idea of how to estimate from a SRS $P(b_1 \leq X \leq b_2)$, how?

So a good estimate of $p(x)$ if it is a flat function in that area is going to be

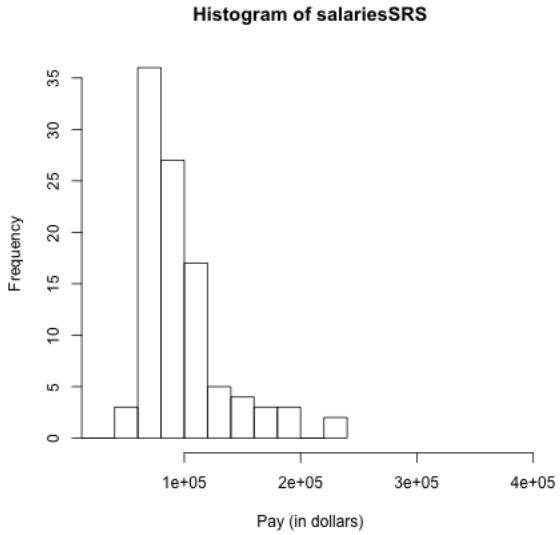
$$\hat{P}(b_1 \leq X \leq b_2)/(b_2 - b_1) = \frac{\#\text{ Points in } [b_1, b_2]}{w \times n}$$

This is exactly what we calculate for a density histogram.

More generally, if the pdf $p(x)$ is pretty smooth, then in a *small enough* windows around x, $p(x)$ is going to be not changing too much roughly the same value. So if the width of the interval is small, we can more generally say

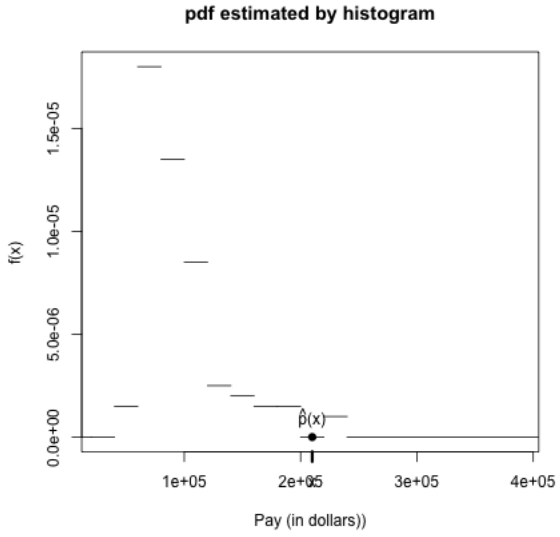
$$\hat{p}(x) = \frac{\hat{P}(\text{data in interval})}{w}$$

With this idea, we can view our histogram as a estimate of the pdf. For example, suppose we consider a histogram of our SRS of salaries,



Then the frequency counts in each bin can be convert to density scale by dividing by the width of the interval of the bins (this is what is meant by the density values in a histogram). Then by our argument above, this density histogram is an estimate of $p(x)$. Specifically, while we don't normally plot it this way, the histogram is a function that estimates $\hat{p}(x)$. We can call it $\hat{p}_{hist}(x)$, and it is a function that is what is called a step function. For every x, we can define an estimated value of $\hat{p}_{hist}(x)$ based on what bin x is in:

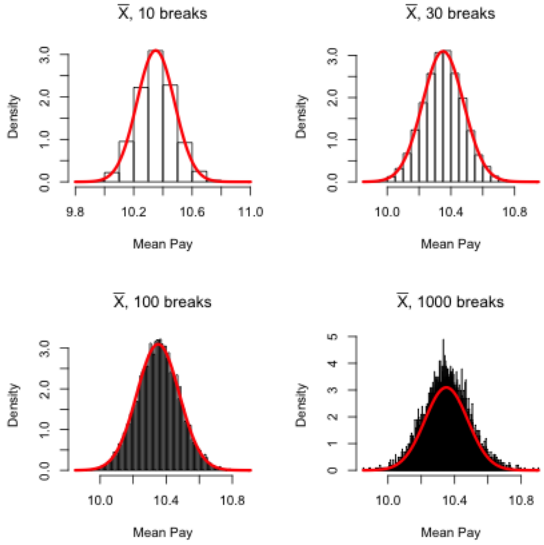
$$\hat{p}_{hist}(x) = \frac{\hat{P}(\text{data in bin of } x\ )}{w}$$



Suppose we want to calculate $\hat{p}_{hist}(60K)$, and we've set up our breaks of our

histogram so that $x = 60K$ is in the bin with interval $[50K, 70K)$. Then how do you calculate $\hat{p}_{hist}(60K)$ from a sample of size 100?

How we choose the breaks in a histogram can affect their ability to be a *good* estimate of the density. Consider our sample of $\bar{X}$ values, which we know approximates a normal,



5.2 Kernel density estimation

The histogram estimate is reasonable estimate if x is right in the middle of the bin, but if x is on the boundary of the bin, what happens?

It makes not only the *size* of the bins, but also the specific *centers* of the bins that matter. And clearly this doesn't make sense for a continuous function!

5.2.1 Moving Windows

As a motivation to kernel density estimation, which is what people use in practice, lets consider a simple version: a moving window or bin.

If you want to estimate $p(x)$ at a specific x, say $x = 72,000$. We would want

$72,000$ to be in the center of the bin. But strangely, when we make a histogram, we set a fix number of centers of the bins, and estimate $\hat{p}_{hist}(x)$, and then return $\hat{p}_{hist}(72,000)$.

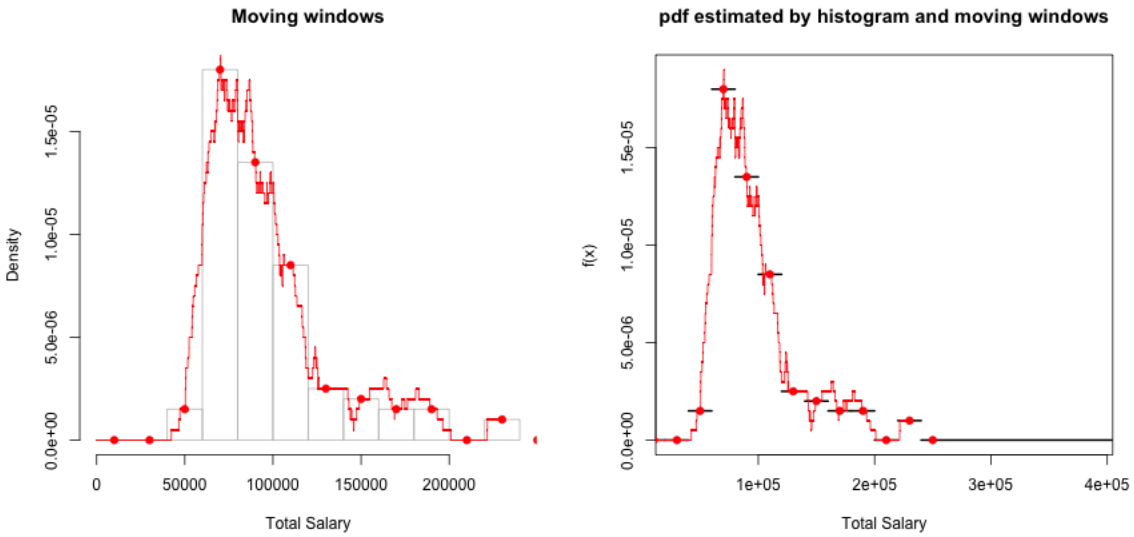
Clearly, then, if I wanted to estimate just $p(72,000)$, I should change my bin and not use $\hat{p}_{hist}(x)$. But estimating $p(x)$, the curve, is equivalent to estimate $p(x)$ *for every* x. So by the same analog, I should estimate a $\hat{p}(x)$ by making a bin centered at x, *for every* x.

For example, say we pick a bin width of $20K$, and want to estimate the density around $72,000$. Then for $x = 72,000$, we could make a interval of width $20K$, $[52,000, 92,000)$, and calculate

$$\frac{\#\text{ x } \in [52K, 92K)}{20K \times 100}$$

We can do this for $x = 80,000$, with an interval of $[60K, 100K)$ and so forth for each x.

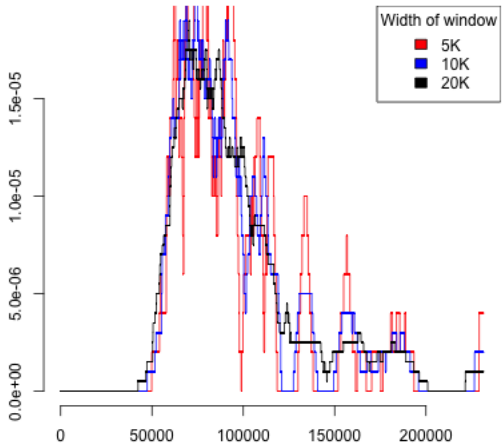
Doing this for *every single* x would give us a curve like this:



More formally, our estimate of $p(x)$, is

$$\hat{p}(x) = \frac{\#x_i \in [x - \frac{w}{2}, x + \frac{w}{2})}{w \times n}$$

We can consider using different size windows:



What is the effect of larger windows or bins?

5.2.2 Weighted Kernel Function

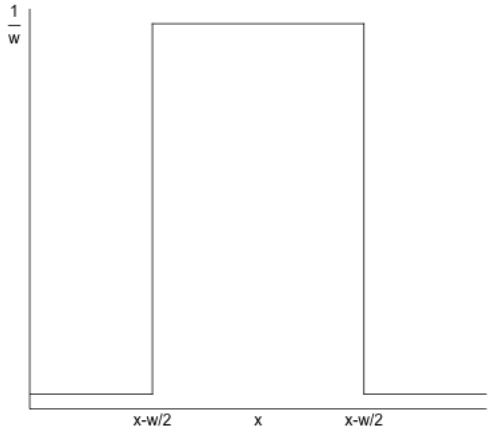
We said our estimate of $p(x)$, is

$$\hat{p}(x) = \frac{\#x_i \in [x - \frac{w}{2}, x + \frac{w}{2})}{w \times n}$$

So to estimate the density around x, we are using the individual data observations if and only if they are close to x. We could write this as a sum over all of our data in our SRS, where some of the data are not counted depending on whether it is close enough to x or not:

$$\hat{p}(x) = \frac{1}{n} \sum_{i: x_i \in [x - \frac{w}{2}, x + \frac{w}{2})} \frac{1}{w}$$

Here is a visualization of how we determine how much a point x_i counts toward estimating $p(x)$ – it either contributes $1/w$ or 0 depending on how far it is from x



We can think of this as a function f of x and x_i: for every x for which we want to estimate $p(x)$, we have a function that tells us how much each of our data points x_i should contribute.

$$f(x, x_i) = \begin{cases} \frac{1}{w} & x_i \in x_i \in [x - \frac{w}{2}, x + \frac{w}{2}) \\ 0 & otherwise \end{cases}$$

It's a function that is different for every x, but just like our moving windows, it's the same function and we just slide it across all of the x. So we can simply write our estimate at each x as an average of the values $f(x, x_i)$

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} f(x, x_i)$$

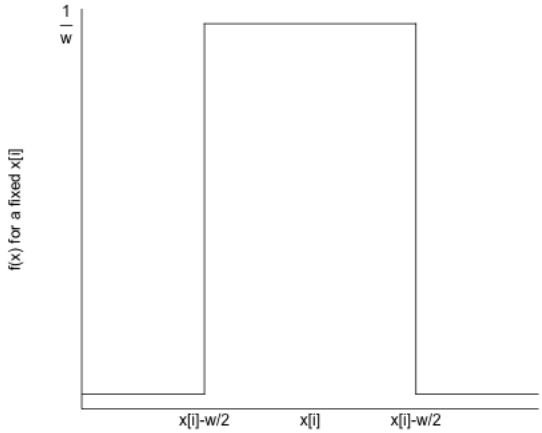
Is this a proper density? Does $\hat{p}(x)$ form a proper density, i.e. is the area under its curve equal 1? We can answer this question by integrating $\hat{p}(x)$

$$\begin{aligned} \int_{-\infty}^{\infty} \hat{p}(x)dx &= \int_{-\infty}^{\infty} \frac{1}{n} \sum_{i=1}^{n} f(x, x_i)dx \\ &= \frac{1}{n} \sum_{i=1}^{n} \int_{-\infty}^{\infty} f(x, x_i)dx \end{aligned}$$

So if $\int_{-\infty}^{\infty} f(x, x_i)dx = 1$ for any x_i, we will have,

$$\int_{-\infty}^{\infty} \hat{p}(x)dx = \frac{1}{n} \sum_{i=1}^{n} 1 = 1.$$

Is this the case? Well, considering $f(x, x_i)$ as a function of x with a fixed x_i value, it is equal to $1/w$ when x is within $w/2$ of x_i, and zero otherwise (i.e. the same function as before, but now centered at x_i):



This means $\int_{-\infty}^{\infty} f(x, x_i)dx = 1$ for any fixed x_i, and so it is a valid density function.

Writing in terms of a kernel function K For various reasons, we will often speak in terms of the distance between x and the x_i relative to our the width on one side of x h:

$$\frac{|x - x_i|}{h}$$

You can think of this as the number of h units x_i is from x. So if we are trying to estimate $p(72,000)$ and our bin width is $w = 5,000$, then $h = 2,500$ and $\frac{|x-x_i|}{h}$ is the number of $2.5K$ units a data point x_i is from $72,000$.

Doing this we can write

$$f_x(x_i) = \frac{1}{h}K(\frac{|x - x_i|}{h})$$

where

$$K(d) = \begin{cases} \frac{1}{2} & d \leq 1 \\ 0 & otherwise \end{cases}$$

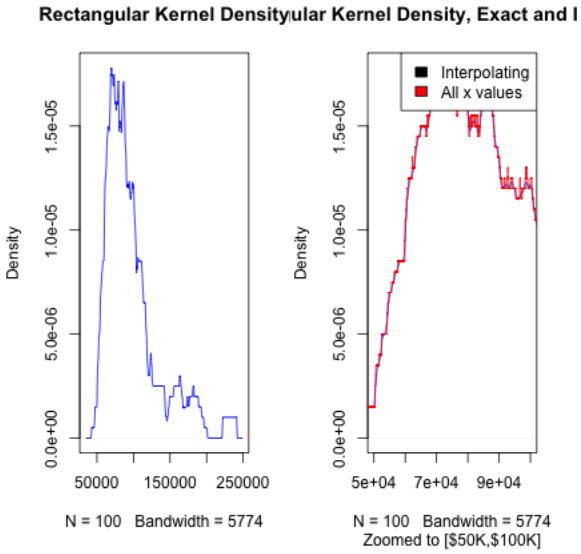
We call a function $K(d)$ that defines a weight for each data point at h-units distance d from x a **kernel function**.

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{|x - x_i|}{h})$$

All of this mucking about with the function K versus $f(x, x_i)$ is not really important – it gives us the same estimate! K is just slightly easier to write mathematically because we took away it's dependence on x, x_i and (somewhat) h.

The parameter h is called the **bandwidth** parameter.

Example of Salary data In R, the standard function to calculate the density is `density`. Our moving window is called the "rectangular" kernel, and so we can replicate what we did using the option `kernel=''rectangular''` in the `density` function [11]

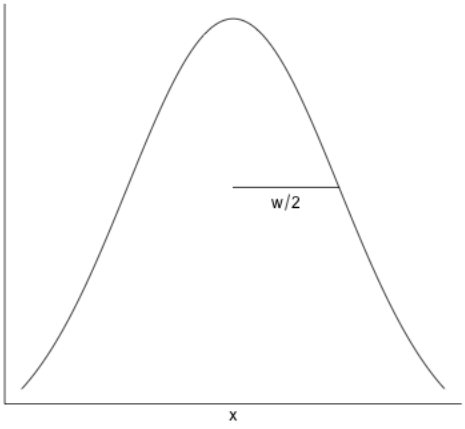


5.2.3 Other choices of kernel functions

Once we think about our estimate like that, we can think about not having such a sharp distinction for the interval around x. After all, what if you have a data point that is $5,100$ away from x rather than $5,000$? Similarly, if you have 50 data points within 100 of x shouldn't they be more informative about the density around x than 50 data points more than $4,500$ away from x?

This gives the idea of letting data points contribute to the estimate of $p(x)$ based on their distance from x, but in a smoother way. For example, consider this more 'gentle' visualization of the contribution of a data point x_i to the density of x

[11] It's actually hard to exactly replicate what I did above with the `density` function, because R is smarter. First of all, it picks a bandwidth from the data. Second, it doesn't evaluate at every possible x like I did. It picks a number, and interpolates between them. For the rectangular density, this makes much more sense, as you can see in the above plot.



This is also the form of a kernel function, called a normal (or gaussian) kernel and is very common for density estimation. It is a normal curve centered at x[12]; as you move away from x you start to decrease in your contribution to the estimate of $p(x)$ but more gradually than the rectangle kernel we started with.

If we want to formally write this in terms of a function K, like above then we would say that our $K(\cdot)$ function is the standard normal curve centered at zero with standard deviation 0.

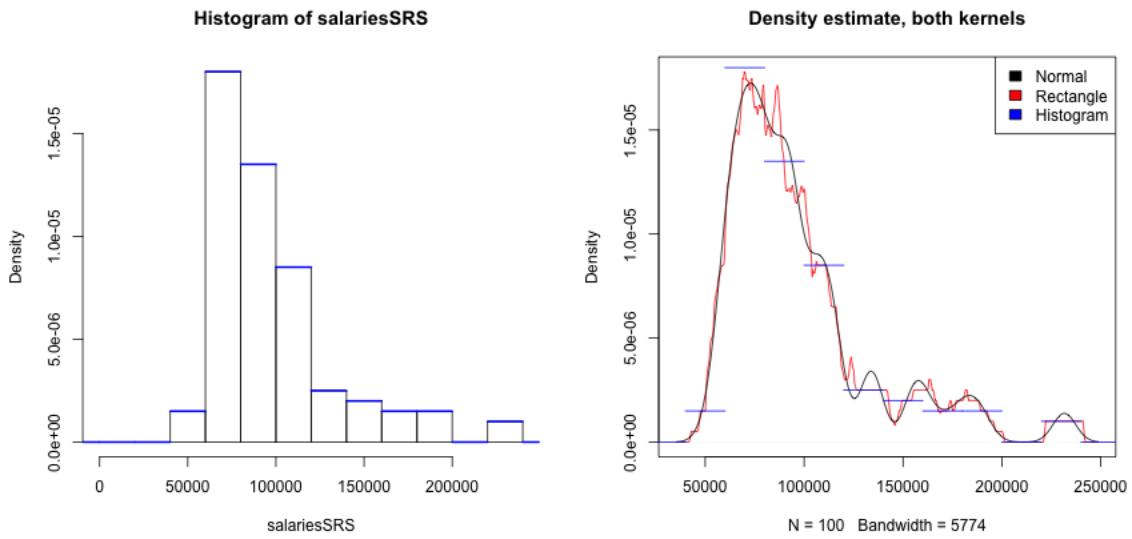
This would imply that

$$\frac{1}{h}K(\frac{|x-x_i|}{h})$$

will give you the normal curve with mean x and standard deviation h.

We can compare the kernel estimates: Here is the estimate of the density based on the rectangular kernel and the normal kernel (now using the defaults), along with our estimate from the histogram:

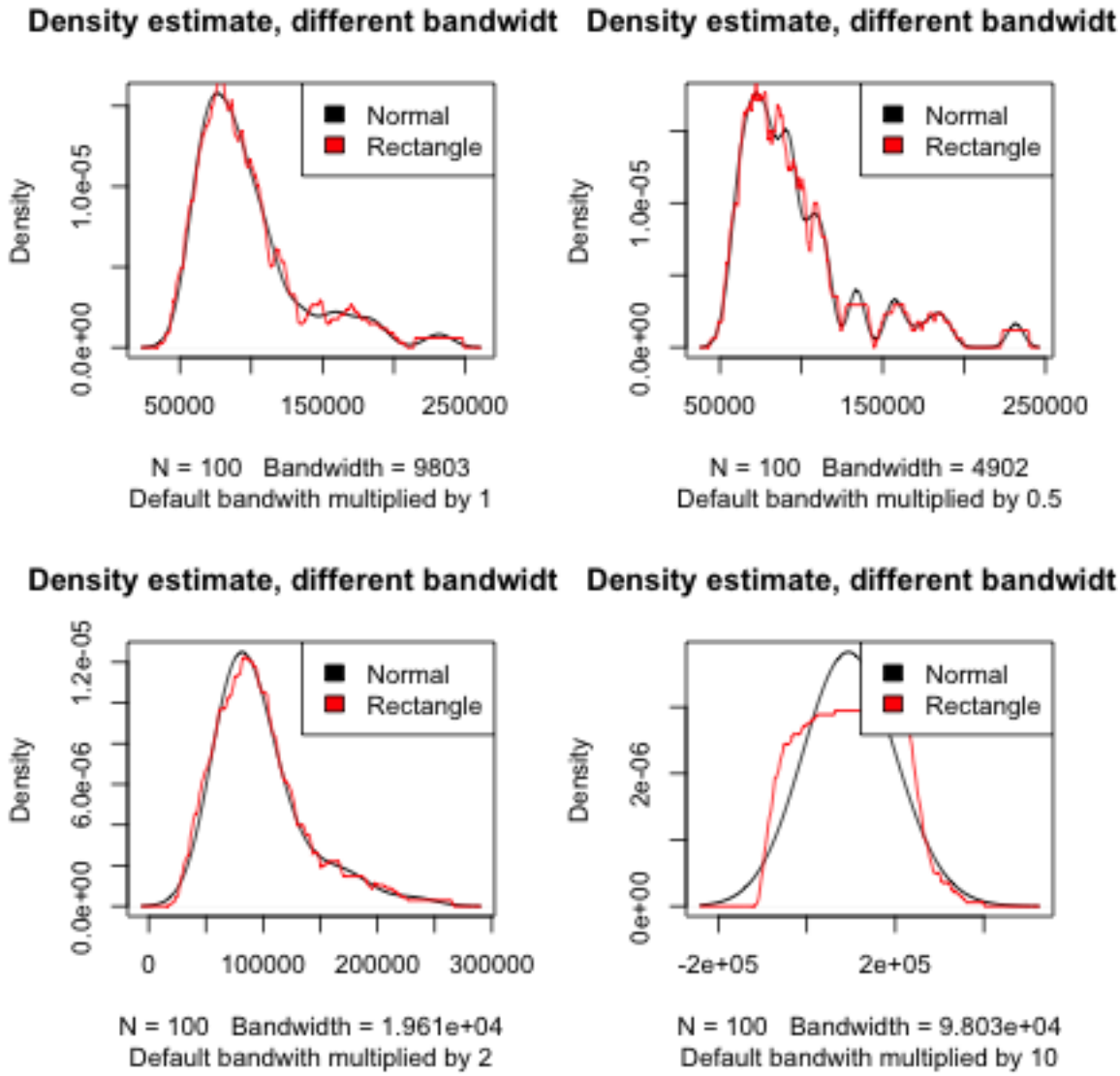
[12]You have to properly scale the height of the kernel function curve so that you get area under the final estimate $\hat{p}(x)$ curve equal to 1



What do you notice when comparing the estimates of the density from these two kernels?

Bandwidth Notice that I still have a problem of picking a width for the rectangular kernel, or the spread/standard deviation for the gaussian kernel. This w is called generically a **bandwidth** parameter. In the above plot I forced the functions to have the same bandwidth corresponding to the moving window of \$20K.

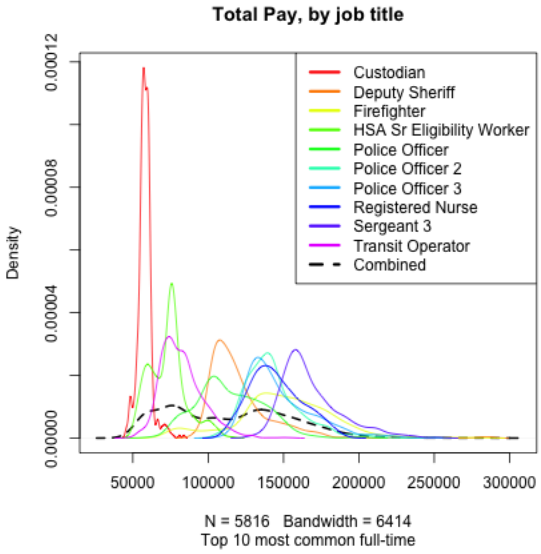
Here are different choices of the bandwidth:



The default parameter of the `density` function is usually pretty reasonable, particularly if used with the gaussian kernel (also the default). Indeed, while we discussed the rectangular kernel to motivate going from the histogram to the kernel density estimator, it's rarely used in practice. It is almost always the gaussian kernel.

5.3 Comparing multiple groups with density curves

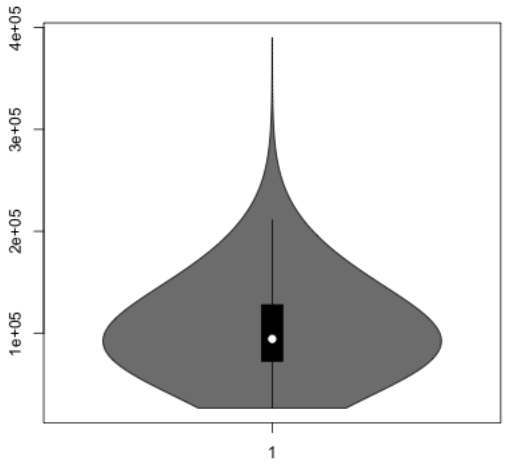
In addition to being a more satisfying estimation of a pdf, density curves are much easier to compare between groups than histograms because you can easily overlay them.



5.4 Violin Plots

We can combine the idea of density plots and boxplots to get something called a 'violin plot'.

```
library(vioplot)
vioplot(salaries2014_FT$TotalPay)
```



This is basically just turning the density estimate on its side and putting it next to the boxplot so that you can get finer-grain information about the distribution. Like

boxplots, this allows you to compare many groups (but unlike the standard `boxplot` command, the `vioplot` function is a bit akward for plotting multiple groups, so I've made my own little function 'vioplot2' available online which I will import here)

```
source("http://www.stat.berkeley.edu/~epurdom/RcodeForClasses/myvioplot.R")
par(mar = c(10, 4.1, 4.1, 0.1))
vioplot2(salaries2014_top$TotalPay, salaries2014_top$JobTitle,
    col = cols, las = 3, ylab = "Salary (in dollars)")
```