**Graphical summaries**

Frequency distribution – A summarization of the observations into classes (groups) where the number of observations per group is given.

Relative frequency distribution – The proportion of observations per class in the frequency distribution is given.

Example: Wind speed in Lincoln (wind\_speed.R, Lincoln\_Feb\_wind.csv)

|  |  |  |
| --- | --- | --- |
| **Class** | **Frequency** | **Relative  Frequency** |
| >2 and ≤ 4 | 6 | 0.04 |
| >4 and ≤ 6 | 22 | 0.15 |
| >6 and ≤ 8 | 23 | 0.16 |
| >8 and ≤ 10 | 29 | 0.20 |
| >10 and ≤ 12 | 20 | 0.14 |
| >12 and ≤ 14 | 11 | 0.08 |
| >14 and ≤ 16 | 14 | 0.10 |
| >16 and ≤ 18 | 9 | 0.06 |
| >18 and ≤ 20 | 3 | 0.02 |
| >20 and ≤ 22 | 4 | 0.03 |
| >22 and ≤ 24 | 1 | 0.01 |

This provides a nice summary, but a picture would be much better!

Histogram – A plot where the height of bars represents the frequency or relative frequency for each class.

Example: Wind speed in Lincoln (wind\_speed.R, Lincoln\_Feb\_wind.csv)

R code and output:

> hist(x = wind$y, col = NA)



> #Nicer version of plot

> win.graph(width = 7, height = 7, pointsize = 10)

> # Works for all operating systems

> # dev.new(width = 7, height = 7, pointsize = 10)

> hist(x = wind$y, main = "Daily wind speed for

Lincoln in February", xlab = "Wind

speed")



What would a histogram for the relative frequency distribution look like?

This cannot be done directly with hist() unless class sizes are equal to 1.

Remember from earlier that we had  = (1.25, 19.15) and  = (-3.22, 23.63). Does the rule of thumb hold here based on what you see in the plot?

How could you represent this on the plot using R?

Obtaining the frequency distribution itself is not as easy. Below is the code that I used.

> save.hist <- hist(x = wind$y, main = "Daily wind speed

for Lincoln", xlab = "Wind speed")

> names(save.hist)

[1] "breaks" "counts" "intensities" "density"

"mids" "xname" "equidist"

> #Information on the classes and the frequencies per

class

> save.hist$breaks #Notice there is one more "break"

than there are counts

[1] 2 4 6 8 10 12 14 16 18 20 22 24

> save.hist$counts

[1] 6 22 23 29 20 11 14 9 3 4 1

> #Frequency distribution

> Rel.Frequency <- round(save.hist$counts/

sum(save.hist$counts), digits = 2)

> data.frame(class = save.hist$breaks[-1], Frequency =

save.hist$counts, Rel.Frequency = Rel.Frequency)

class Frequency Rel.Frequency

1 4 6 0.04

2 6 22 0.15

3 8 23 0.16

4 10 29 0.20

5 12 20 0.14

6 14 11 0.08

7 16 14 0.10

8 18 9 0.06

9 20 3 0.02

10 22 4 0.03

11 24 1 0.01

Questions:

1. Why were these classes used?

There are a number of ways to choose classes. I chose to use the default given by hist(). Other classes could have been used. The key is to avoid too few or too many classes. For example, two classes would not be too informative here:



Also, 71 classes would not be too good because there are only 142 observations which leads to a “bumpy” plot with few observations in most classes:



The default way that hist() chooses the number of classes is not important to us. However, the method is not perfect. If needed, you should change the classes to make the histogram more descriptive. This can be done by using the breaks and the nclass arguments in hist(). Please see the examples in the program for the two ways this argument can be used.

1. Is there an easier way to find JUST the frequency distribution?

Typically, you will want to get the hist() no matter what. However, as a way to show how to write your own function in R, here is how I found just the frequency and relative frequency distribution with no histogram.

> freq.dist <- function(data, numb.breaks = "Sturges") {

save.hist <- hist(x = data, plot = FALSE, breaks =

numb.breaks)

Rel.Frequency <- round(save.hist$counts /

sum(save.hist$counts), 2)

data.frame(class = save.hist$breaks[-1], Frequency =

save.hist$counts, Rel.Frequency = Rel.Frequency)

}

> freq.dist(data = wind$y)

class Frequency Rel.Frequency

1 4 6 0.04

2 6 22 0.15

3 8 23 0.16

4 10 29 0.20

5 12 20 0.14

6 14 11 0.08

7 16 14 0.10

8 18 9 0.06

9 20 3 0.02

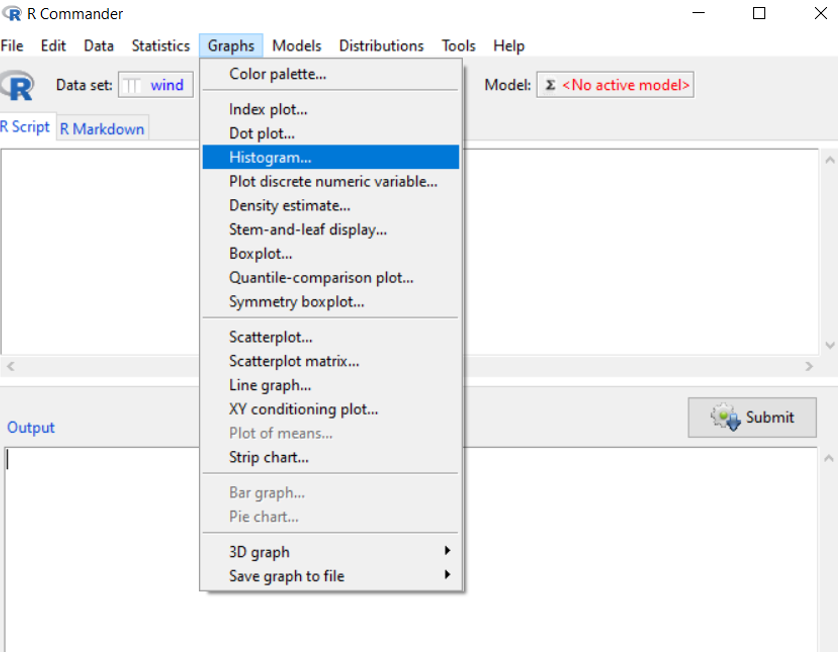
10 22 4 0.03

11 24 1 0.01

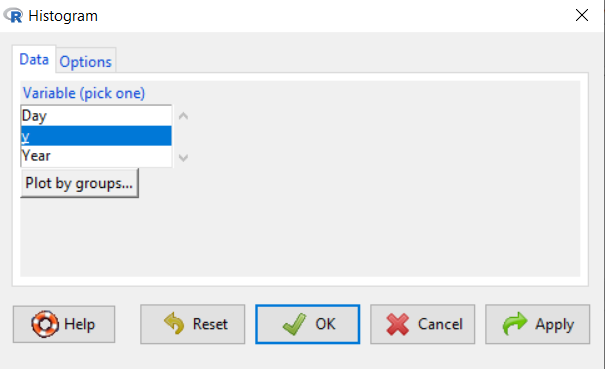
1. How about R Commander?

Yes, it can be used to find the frequency distribution and histogram. Note that the code produced is a little different than my own. This is because the author of R commander wrote his own functions (like my freq.dist() function) to better fit his needs.

To use R commander, I opened it up by using library(package = Rcmdr) in the R Console. I chose my data set by selecting the “No data set” area and then I selected the wind data set. Next, I selected GRAPH > HISTOGRAM.



Finally, I selected the variable of interest



to produce the plot below



What have we learned about the wind speed?

Leads into discussion about probability distributions and more mathematical ways to represent one. Discuss wind power.

Skewness – A term used to describe the symmetry of a histogram (or some other similar type of plot displaying the observations)

|  |  |  |
| --- | --- | --- |
| **Skewness** | **Histogram bars** | **Mean & median** |
| None | Symmetric about middle | mean = median |
| Left | Longer tail on the left than right | mean < median |
| Right | Longer tail on the right than left | mean > median |

Note: A “tail” of a histogram is a term used to describe how the bars go down on the left and right sides of it.

Questions:

* Why would the mean > median if there was a longer tail of the right than on the left?
* Is the wind histogram right or left skewed?

Dot plot – A dot is plotted for each observation with respect to the y-axis. Often, these dots are jittered (moved a little in the x-axis direction) to prevent overlapping.

Example: Cereal data (cereal.R, cereal.csv)

Cereals marketed to kids often have high sugar content.

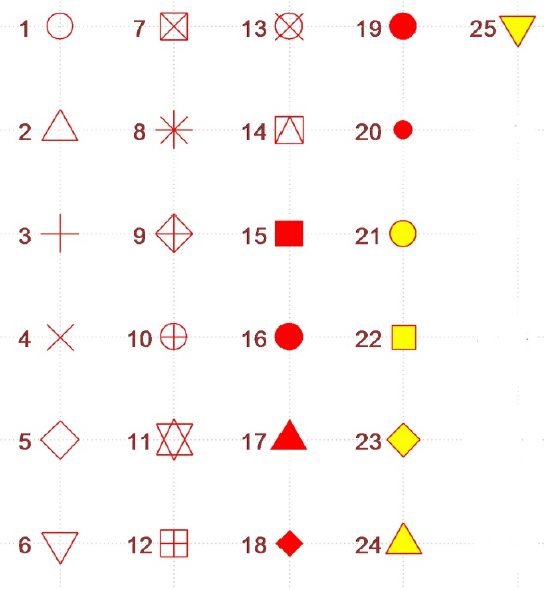
> stripchart(x = cereal$sugar ~ cereal$Shelf, method =

"jitter", vertical = TRUE, pch = 1, main = "Dot plot",

ylab = "Sugar", xlab = "Shelf")



Note that the pch argument corresponds to the plotting character for each point. Below are the available plotting characters:



Questions:

1. What have we learned about the sugar content of cereals?
2. Why is an “open” circle better to use for plotting that a “filled-in” circle?

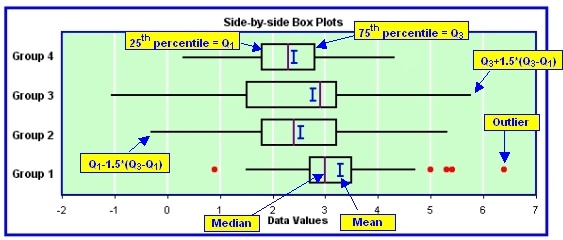
Please see the corresponding program for how to draw these plots with the help of using the plot() function.

It would be nice to have additional numerical summary measures, like the mean, quantiles, and/or the rule of thumb, represented on the plot. The next plot shows how to include some of these items.

Box plot – It is easiest to describe this plot in steps:

1. Calculate 25th, 50th, and 75th percentiles (Q1, Q2, and Q3, respectively).
2. Plot a rectangle or “box” relative to the y-axis where the bottom of the rectangle is at Q1 and the top of the rectangle is at the Q3.
3. Plot a horizontal line within the box at Q2.
4. Extend a vertical line out from the bottom of the rectangle to Q1 – 1.5(Q3 – Q1). Put a short horizontal line at the bottom of the vertical line. This line is called a whisker.
5. Extend a vertical line out from the top of the rectangle to Q3 + 1.5(Q3 – Q1). Put a short horizontal line at the top of the vertical line. This line is called a whisker.
6. Represent any observations outside of the Q1 – 1.5(Q3 – Q1) and Q3 + 1.5(Q3 – Q1) range as open dots. These observations are referred to as outliers because they are somewhat unusual in value.

Below is an example plot, but drawn horizontally instead of vertically as described above.



Note: There are many different ways to draw box plots. R deviates from the steps given here with respect to its whiskers. It will draw its whiskers out to particular observations only, not to Q3 + 1.5(Q3 – Q1) and Q1 – 1.5(Q3 – Q1). For example, if there are no observations above Q3 + 1.5(Q3 – Q1), the whisker is extended out to the largest observation value. If there are one or more observations larger than Q3 + 1.5(Q3 – Q1), the whisker is extended out to the largest observation value LESS THAN Q3 + 1.5(Q3 – Q1). Similar adjustments are made with respect to the other whisker.

Example: Cereal data (cereal.R, cereal.csv)

> par(mfrow = c(1,2)) # One row and two columns of plots

> stripchart(x = cereal$sugar ~ cereal$Shelf,

method = "jitter", vertical = TRUE, pch = 1, main

= "Dot plot", ylab = "Sugar", xlab = "Shelf")

> boxplot(formula = sugar ~ Shelf, data = cereal, col

= "lightblue", main = "Box plot", ylab = "Sugar",

xlab = "Shelf")



Comments:

* Just to verify the location of some lines, below are 25th, 50th, and 75th percentiles.

> aggregate(x = sugar ~ Shelf, data = cereal, FUN

= quantile, probs = c(0.25, 0.5, 0.75))

Shelf sugar.25% sugar.50% sugar.75%

1 1 0.07142857 0.34408602 0.37376847

2 2 0.34166667 0.42037037 0.46363147

3 3 0.09838710 0.25690236 0.33294099

4 4 0.16935484 0.28181818 0.34545455

Added after video recording: R has changed the syntax for aggregate(). In the video, I show formula = sugar ~ Shelf. Now, the proper syntax is x = sugar ~ Shelf. I made the correction here and in the program.

Notice there are no outliers.

* The box plots allow one to see the central part of the observations for each shelf.
* If the two plots did not have the same y-axis limits, you could specify the limits through a ylim = c(0, 0.55) type of argument.

What have we learned about the sugar content of cereals?

One could also overlap the dot and box plots:



This works well when there are not too many observations. Please see the program code for how I did this plot (see the add = TRUE argument value).

Example: Wind speed in Lincoln (wind\_speed.R, Lincoln\_Feb\_wind.csv)

Are there differences across years?

> aggregate(x = y ~ Year, data = wind, FUN = mean)

Year y

1 1 10.303448

2 2 8.992857

3 3 11.232143

4 4 10.057143

5 5 10.403448

> aggregate(x = y ~ Year, data = wind, FUN = sd)

Year y

1 1 4.098561

2 2 5.022905

3 3 5.080978

4 4 3.421872

5 5 4.581834

> aggregate(x = y ~ Year, data = wind, FUN =

quantile, probs = c(0.25, 0.5, 0.75))

Year y.25% y.50% y.75%

1 1 7.800 9.500 12.900

2 2 5.175 7.750 11.000

3 3 7.425 10.150 14.350

4 4 7.650 10.000 12.525

5 5 6.900 9.800 13.900



Notice the outlier for year #2. One needs to be careful then with overlaying the dot plot in these settings so that you do not think there are TWO observations above the whisker. To prevent this from happening, you can add the argument pars = list(outpch=NA) to the boxplot() code.

Suppose a similar plot was done for temperature for every decade from 1900 to now. What would you expect the plot to look like? What if every year was used instead of decade?

Example: Dividend yield (div\_yield.R, div\_yield.csv)

I took a random sample of 30 companies that were listed on the New York Stock Exchange (NYSE) and 30 companies that were listed on the NASDAQ. I recorded their dividend yields. Below is part of the data.

> div <- read.csv(file = "div\_yield.csv")

> head(div) #Shows first 6 observations

ID Company Exchange Closing\_Price Dividend Dividend\_Yield

1 1 AMF Bowlng NYSE 21.75 0.00 0.000000

2 2 Alr TOPRS NYSE 25.25 1.90 0.075248

3 3 AmerHess NYSE 62.19 0.60 0.009648

4 4 AmStratll NYSE 11.81 0.99 0.083810

5 5 ArdenRlty NYSE 30.75 1.60 0.052033

6 6 Aviall NYSE 13.31 0.00 0.000000

> tail(div) #Shows last 6 observations

ID Company Exchange Closing\_Price Dividend Dividend\_Yield

55 25 EqityOil NASDAQ 3.188 0.00 0.00000

56 26 FFY Fnl NASDAQ 29.000 0.80 0.02759

57 27 FstAlbny NASDAQ 15.000 0.20 0.01333

58 28 FstSavSLA NASDAQ 42.500 0.48 0.01129

59 29 FtWaynNtl NASDAQ 36.875 0.80 0.02169

60 30 GalileoCp NASDAQ 11.375 0.00 0.00000

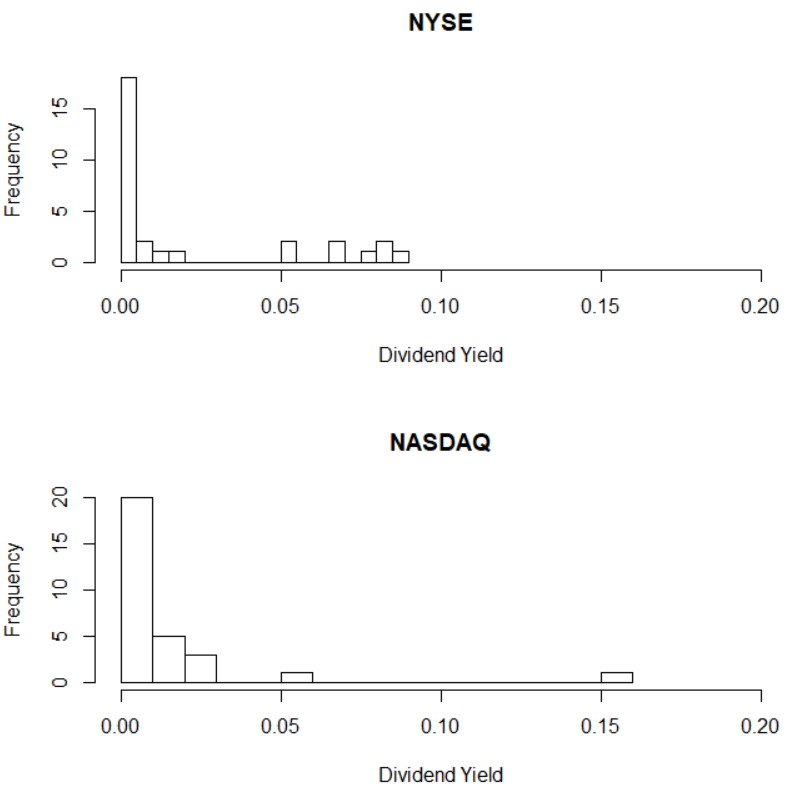
Question: Is there a difference in dividend yield among stocks traded on the two exchanges? How do you measure “difference”? Mean? Median? Distribution?

Below are some plot and summary statistics:



The default gray background for the box was removed using col = NA in boxplot().

Notice the lower parts of the box plots are a little odd. Why does this happen?



The default gray background for the bars was removed using col = NA in hist().

> aggregate(x = Dividend\_Yield ~ Exchange, data = div,

FUN = mean)

Exchange Dividend\_Yield

1 NASDAQ 0.01180

2 NYSE 0.02060

> aggregate(x = Dividend\_Yield ~ Exchange, data = div,

FUN = sd)

Exchange Dividend\_Yield

1 NASDAQ 0.02894

2 NYSE 0.03191

> aggregate(x = Dividend\_Yield ~ Exchange, data = div,

FUN = quantile, probs = c(0.25, 0.5, 0.75))

Exchange Dividend\_Yield.25% Dividend\_Yield.50% Dividend\_Yield.75%

1 NASDAQ 0.00000 0.00000 0.01282

2 NYSE 0.00000 0.00000 0.04308

Pay special attention to the code corresponding to the histograms.

Note that the NYSE traditionally has contained more of the larger companies (an exception is technology stocks) than the NASDAQ. How can we relate this to our results here?

Parallel Coordinates plot – This plot is best explained through looking at the example plot below.



Comments on plot construction:

1. Each variable of interest is rescaled ((value - minimum) / (maximum - minimum)) so that minimum observation value is at the bottom and the maximum observation value is at the top. This scale is now used as the vertical access for each variable.
2. Similar to a dot plot, the rescaled observations are plotted for each variable.
3. Lines connect observations corresponding to the same cereal in the data set.

One needs to watch out for the overlapping of lines. This is especially important when a variable has few values (e.g., 0 or 1).

Example: Cereal data (cereal.R, cereal.csv)

The function parcoord() is in the MASS package. This package is in the default installation of R, so it does not need to be downloaded; however, the library() function does need to be run to let R know that you want to use a function within it.

> library(package = MASS)

> cereal2 <- data.frame(cereal$Cereal.ID, cereal$sugar,

cereal$fat, cereal$sodium)

> color.by.shelf <- rep(x = c("black", "red", "blue",

"green"), each = 10)

> parcoord(x = cereal2, col = color.by.shelf, main = "Parallel

coordinate plot for cereal data")

> legend(locator(1), legend = c("1", "2", "3", "4"), lty =

"solid", col = c("black", "red", "blue", "green"), bty =

"n")

What can we learn from this plot?

1. Notice how the shelf 2 cereals tend to be from the middle to the top for the sugar variable. This is giving preliminary indication that those cereals have some of the higher sugar content cereals in comparison with some of the other shelves.
2. There are a few outliers for the fat variable as indicated by their large values in comparison to the rest.
3. Examine what happens when you follow the cereal lines from one variable to another. For example, the highest in sugar content cereals do not necessarily have high fat content.

Scatter plot

See the introduction to R notes for the GPA example.

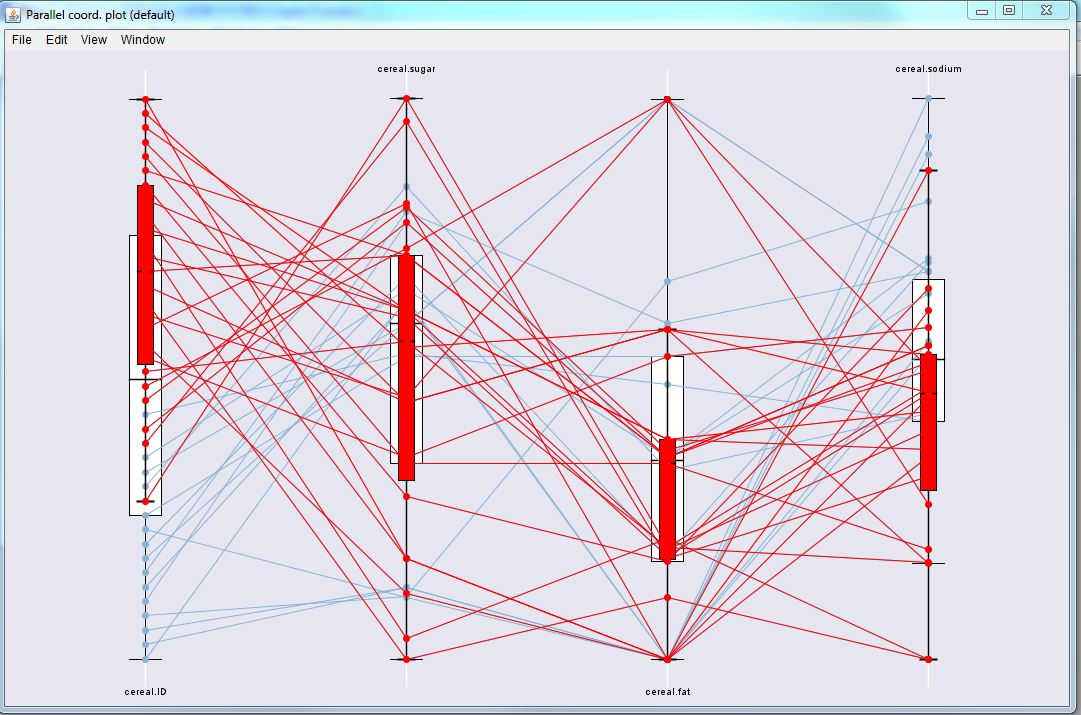
There are many more plots that can be done. Below are a few discussed in the book or available in R:

1. Stem-and-leaf plot – A form of a histogram where the actual numerical values of the observations help to form the bars. See stem() in R.
2. Time series plot – Observations are plotted over time. For example, the observations could be the stock price of a company, and this is plotted over a one-month period of time. I teach a course on time series analysis where this type of plot is used a lot – see [www.chrisbilder.com/ts](http://www.chrisbilder.com/ts).
3. Other R functions – There are a number of R packages that can create more sophisticated plots. For example, the iplots package can create interactive plots. Below is another parallel coordinate plot that allows for “brushing” of points along with a number of additions through using the GUI.

> library(package = iplots)

> ipcp(vars = cereal2)

ID:1 Name: "Parallel coord. plot (default)"



Note that Java needs to be enabled on your computer to use this package.

The ggplot2 and the lattice package are additional packages that offer a full set of plots similar to what we have been using in the graphics package.

A good book on graphics in R is “R Graphics” by Paul Murrell.