Diagram from earlier



This section focuses on the sample part of the above diagram. We are going to learn about how to properly summarize data in the sample. This will lead us to a general idea of what is happening in the population.

Note that there will be an emphasis on R code in this chapter. While the code may seem a little daunting at first, please think of my code as a template for your own. Often, you will be able to complete assigned work through copying and pasting my code into your own program and then change a few items. After a while, you will get better at writing code yourself without having to depend on my own.

Measures of centrality

Mean – The average value

Population mean – Population average value denoted by the Greek letter μ.

Suppose a sample of size n is taken from the population producing the observations y1, …, yn.

Sample mean – Sample average value denoted by the symbol .



Usually, μ will NOT be known! Therefore, we use  to estimate μ.

Median – Center of the data values when they are arranged from smallest to largest

Sample median:

* If n is odd, take the (n+1)/2 ordered value
* If n is even, average the n/2 and n/2+1 ordered value.

Example: Simple GPA problem (simple\_GPA.R)

Suppose we are interested in the mean college GPA for a set of individuals. The population consists of the GPAs 3.6, 2.7, 2.4, 2.8, 3.9, 3.2, 2.9, 3.4, 1.2, and 4.0. Note that .

Suppose a sample of size n = 4 is taken from the population and the sample consists of 2.9, 3.4, 3.6, and 2.8. Then .

The sample mean is off by |3.175-3.01| = 0.165.

What if we took another sample of n = 4? Suppose the sample consists of 3.6, 4.0, 2.7, and 2.8. Then 

The sample mean is off by |3.275-3.01| = 0.265.

Many more samples could be taken, and most (if not all) will not have a sample mean equal to the population mean! How can we attach then some level of accuracy to the sample mean? We will see in a few weeks!

To find the median, first order the values from smallest to largest:



Since n is even, the median is the average of the 4/2 = 2nd value and 4/2 + 1 = 3rd value. In this case, the average of 2.9 and 3.4 is 3.15.

If an additional observation greater than 3.6 was added to the data set, then the median would be 3.4.

Calculations in R:

> y <- c(2.9, 3.4, 3.6, 2.8)

> mean(y)

[1] 3.175

> median(y)

[1] 3.15

> #Another way

> summary(y)

 Min. 1st Qu. Median Mean 3rd Qu. Max.

 2.800 2.875 3.150 3.175 3.450 3.600

When extreme data values or “outliers” are present, the median is a preferred measure of the center of the data set over the mean. The reason why is that the median is not affected by extreme values (a.k.a., outliers). Why?

Although the median is better to use when extreme values are present, the mean is used more often later in this course because it has some nicer mathematical properties about it.

Measures of variation

The purpose here is to determine if observations are clustered closely about the center or are they widely dispersed.

Range = maximum data value – minimum data value

Example: Suppose the data consists of 100, 10, 11, and 9. Range = 100 - 9 = 91

Problem with range: Very sensitive to outliers.

Variance - Numerical measurement of how the data tend to vary around the mean.

Sample variance – “Average” squared deviation of the observations from their sample mean:



Population variance – “average” squared deviation of all population values (if they are obtainable) from their population mean:



where there are N different values in the population denoted by y1, …, yN. Note that y1 in the sample may not be y1 in the population.

Questions/comments:

* s2 estimates σ2
* Why is n – 1 in the denominator for s2 rather than n? This results in a better estimate of σ2, and further reasons will be given later in the course.
* What characteristics of the sample would lead to a very small s2? What characteristics of the sample would lead to a very large s2?
* Why is the square in ?
* Suppose you had two samples consisting of the following numerical values:
1. 8, 9, 10
2. 7, 8, 10

Without going through the calculations, which sample would have a larger s2? Suppose the “10” in sample 1 was changed to 11. Which sample would have the larger s2?

* The units that the variance is measured in are “squared” units. This is because of  or in the numerator. It is often easier to work in the original units that the observations are measured in.

Standard deviation – The positive square root of the variance.

The sample standard deviation is . The population standard deviation is .

Example: Simple GPA problem (simple\_GPA.R)

The sample variance is



The sample standard deviation is .

The population variance is



The population standard deviation, σ, is σ = 0.7816.

Remember that s2 estimates σ2.

R code:

> var(y)

[1] 0.1491667

> sd(y)

[1] 0.386221

Rule of thumb for the number of standard deviations all data lies from its mean: All or most observed values should be 2 to 3 standard deviations from the mean:

 or 

Why is this useful? Discuss the time it takes to get to a class.

Where does this result come from?

“Chebyshev’s Rule” says that 75% of all observations are within  and 89% are within . Also, the “Empirical Rule” says that approximately 95% of the observations are within  and approximately 99.7% of the observations are within  when additional conditions are given (mound shaped “histogram” – to be discussed later).

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

This is a sample of average daily wind speeds from over 5 different years for the month of February.

Questions:

1. What are possible reasons why someone would want to examine wind speed?
2. What is the population?
3. Is this a random sample?

This is referred to as an observational study because we are not controlling other items that may be related to wind speed. This is in contrast to a scientific study where these other items are controlled. More will be said about these types of studies as we proceed through the course.

There are a total of 142 observations in the sample. Just looking at the observation values alone, it is difficult to understand (or least get a general idea) of what they are telling us about the population. Therefore, we will examine some summary measures of the sample.

> # May need to set folder location of file

> # This will not be needed if you open the program from

 the folder where the data is located

> # setwd(dir = "C:\\Chris\\data") #Set this to your own

 folder

> wind <- read.csv(file = "Lincoln\_Feb\_wind.csv")

> head(wind) #Shows first 6 observations

 Year Day y

1 1 1 9.4

2 1 2 12.7

3 1 3 3.9

4 1 4 9.8

5 1 5 9.5

6 1 6 15.0

> tail(wind) #Shows last 6 observations

 Year Day y

137 5 24 7.8

138 5 25 21.6

139 5 26 14.9

140 5 27 5.0

141 5 28 18.5

142 5 29 10.3

> #Summary statistics

> mean(wind$y)

[1] 10.2

> sum(wind$y)/length(wind$y) #Alternative way

[1] 10.2

> median(wind$y)

[1] 9.7

> sd(wind$y)

[1] 4.476305

> sqrt(sum((wind$y - mean(wind$y))^2)/(length(wind$y) - 1))

 #Alternative way

[1] 4.476305

> data.frame(lower = mean(wind$y) - 2\*sd(wind$y),

 upper = mean(wind$y) + 2\*sd(wind$y))

 lower upper

1 1.247390 19.15261

> data.frame(lower = mean(wind$y) - 3\*sd(wind$y),

 upper = mean(wind$y) + 3\*sd(wind$y))

 lower upper

1 -3.228915 23.62891

> save.interval <- data.frame(lower = mean(wind$y) –

 2\*sd(wind$y), upper = mean(wind$y) + 2\*sd(wind$y))

> save.interval$upper

[1] 19.15261

> sum(wind$y > save.interval$upper)

[1] 6

> sum(wind$y < save.interval$lower)

[1] 0

> sum(wind$y > save.interval$upper)/length(wind$y)

[1] 0.04225352

How could this information be used?

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

I collected the following data on cereal from a grocery store. This data was collected as a stratified random sample where shelf of the cereal was the stratum. This is called “stratified” since I randomly selected 10 cereals within a shelf rather than across all shelves which could have led to > or < 10 cereals per shelf. Shelf #1 represents the bottom shelf and shelf #4 represents the top shelf.

| **ID** | **Shelf** | **Cereal** | **Serving Size (g)** | **Sugar (g)** | **Fat (g)** | **Sodium (mg)** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | Kellogg’s Razzle Dazzle Rice Crispies | 28 | 10 | 0 | 170 |
| 2 | 1 | Post Toasties Corn Flakes | 28 | 2 | 0 | 270 |
| 3 | 1 | Kellogg’s Corn Flakes | 28 | 2 | 0 | 300 |
| 4 | 1 | Food Club Toasted Oats | 32 | 2 | 2 | 280 |
| 5 | 1 | Frosted Cheerios | 30 | 13 | 1 | 210 |
| 6 | 1 | Food Club Frosted Flakes | 31 | 11 | 0 | 180 |
| 7 | 1 | Capn Crunch | 27 | 12 | 1.5 | 200 |
| 8 | 1 | Capn Crunch's Peanut Butter Crunch | 27 | 9 | 2.5 | 200 |
| 9 | 1 | Post Honeycomb | 29 | 11 | 0.5 | 220 |
| 10 | 1 | Food Club Crispy Rice | 33 | 2 | 0 | 330 |
| 11 | 2 | Rice Crispies Treats | 30 | 9 | 1.5 | 190 |
| 12 | 2 | Kellogg's Smacks | 27 | 15 | 0.5 | 50 |
| 13 | 2 | Kellogg's Froot Loops | 32 | 15 | 1 | 150 |
| 14 | 2 | Capn Crunch's Peanut Butter Crunch | 27 | 9 | 2.5 | 200 |
| 15 | 2 | Cinnamon Grahams | 30 | 11 | 1 | 230 |
| 16 | 2 | Marshmallow Blasted Froot Loops | 30 | 16 | 0.5 | 105 |
| 17 | 2 | Koala Coco Krunch | 30 | 13 | 1 | 170 |
| 18 | 2 | Food Club Toasted Oats | 33 | 10 | 1.5 | 150 |
| 19 | 2 | Cocoa Pebbles | 29 | 13 | 1 | 160 |
| 20 | 2 | Oreo O's | 27 | 11 | 2.5 | 150 |
| 21 | 3 | Food Club Raisin Bran | 54 | 17 | 1 | 280 |
| 22 | 3 | Post Honey Bunches of Oats | 30 | 6 | 1.5 | 190 |
| 23 | 3 | Rice Chex | 31 | 2 | 0 | 290 |
| 24 | 3 | Kellogg's Corn Pops | 31 | 14 | 0 | 120 |
| 25 | 3 | Post Morning Traditions - Raisin, Date, Pecan | 54 | 14 | 5 | 160 |
| 26 | 3 | Post Shredded Wheat Spoon Size | 49 | 0 | 0.5 | 0 |
| 27 | 3 | Basic 4 | 55 | 14 | 3 | 320 |
| 28 | 3 | French Toast Crunch | 30 | 12 | 1 | 180 |
| 29 | 3 | Post Raisin Bran | 59 | 20 | 1 | 300 |
| 30 | 3 | Food Club Frosted Shredded Wheat | 50 | 1 | 1 | 0 |
| 31 | 4 | Total Raisin Bran | 55 | 19 | 1 | 240 |
| 32 | 4 | Food Club Wheat Crunch | 60 | 6 | 0 | 300 |
| 33 | 4 | Oatmeal Crisp Raisin | 55 | 19 | 2 | 220 |
| 34 | 4 | Food Club Bran Flakes | 31 | 5 | 0.5 | 220 |
| 35 | 4 | Cookie Crisp | 30 | 12 | 1 | 180 |
| 36 | 4 | Kellogg's All Bran Original | 31 | 6 | 1 | 65 |
| 37 | 4 | Food Club Low Fat Granola | 55 | 14 | 3 | 100 |
| 38 | 4 | Oatmeal Crisp Apple Cinnamon | 55 | 19 | 2 | 260 |
| 39 | 4 | Post Fruit and Fibre - Dates, Raisons, Walnuts | 55 | 17 | 3 | 280 |
| 40 | 4 | Total Corn Flakes | 30 | 3 | 0 | 200 |

The reason for the data collection was to determine if the amount of sugar (or fat or sodium) differed based on the shelf. What are possible reasons why it may differ?

Questions:

* What is the population?
* Could all of the population values be obtained?

Below is how I read in the data and made some small adjustments to it that account for different serving sizes.

> cereal <- read.csv(file = "cereal.csv")

> head(cereal) #Shows first 6 observations

 ID Shelf Cereal size\_g sugar\_g fat\_g sodium\_mg

1 1 1 Kellog's Razzle Dazzle Rice Crispies 28 10 0 170

2 2 1 Post Toasties Corn Flakes 28 2 0 270

3 3 1 Kellog's Corn Flakes 28 2 0 300

4 4 1 Food Club Toasted Oats 32 2 2 280

5 5 1 Frosted Cheerios 30 13 1 210

6 6 1 Food Club Frosted Flakes 31 11 0 180

> tail(cereal) #Shows last 6 observations

 ID Shelf Cereal size\_g sugar\_g fat\_g

35 35 4 Cookie Crisp 30 12 1

36 36 4 Kellogg's All Bran Original 31 6 1

37 37 4 Food Club Low Fat Granola 55 14 3

38 38 4 Oatmeal Crisp Apple Cinnamon 55 19 2

39 39 4 Post Fruit and Fibre - Dates, Raisons, Walnuts 55 17 3

40 40 4 Total Corn Flakes 30 3 0

 sodium\_mg

35 180

36 65

37 100

38 260

39 280

40 200

> #Adjust data to take into account the different serving

 sizes

> cereal$sugar <- cereal$sugar\_g/cereal$size\_g

> cereal$fat <- cereal$fat\_g/cereal$size\_g

> cereal$sodium <- cereal$sodium\_mg/cereal$size\_g

> head(cereal) #Shows first 6 observations

 ID Shelf Cereal size\_g sugar\_g fat\_g sodium\_mg

1 1 1 Kellog's Razzle Dazzle Rice Crispies 28 10 0 170

2 2 1 Post Toasties Corn Flakes 28 2 0 270

3 3 1 Kellog's Corn Flakes 28 2 0 300

4 4 1 Food Club Toasted Oats 32 2 2 280

5 5 1 Frosted Cheerios 30 13 1 210

6 6 1 Food Club Frosted Flakes 31 11 0 180

 sugar fat sodium

1 0.35714286 0.00000000 6.071429

2 0.07142857 0.00000000 9.642857

3 0.07142857 0.00000000 10.714286

4 0.06250000 0.06250000 8.750000

5 0.43333333 0.03333333 7.000000

6 0.35483871 0.00000000 5.806452

Because we are interested in the sugar content by shelf, a simple application of the same code as before will not work as well. Instead, we need to do the calculations “by” shelf. Below is my code and output:

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

 mean)

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.

 Shelf sugar

1 1 0.2568366

2 2 0.4149686

3 3 0.2303732

4 4 0.2554839

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

 Shelf sugar

1 1 0.16729566

2 2 0.09001019

3 3 0.15770057

4 4 0.11010226

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

 Shelf sugar.Min. sugar.1st Qu. sugar.Median sugar.Mean sugar.3rd Qu.

1 1 0.06061 0.07143 0.34410 0.25680 0.37380

2 2 0.30000 0.34170 0.42040 0.41500 0.46360

3 3 0.00000 0.09839 0.25690 0.23040 0.33290

4 4 0.10000 0.16940 0.28180 0.25550 0.34550

 sugar.Max.

1 0.44440

2 0.55560

3 0.45160

4 0.40000

What have we learned about the sugar content of cereals and shelf placement?

Measures of position

pth percentile – Numerical value where at least p percent of the items are less than or equal to this value and (100-p)% of the items are greater than or equal to this value. Note that 0 < p < 100.

There are a number of ways to calculate percentiles. In fact, R has 9 different ways! Below is one way:

1. Order the data values from smallest to largest
2. The jth ordered observation corresponds to the 100(j – 0.5)/n percentile, where n is the sample size.

For example, suppose there is a sample size n = 20. The 1st ordered observation is the 100(1 – 0.5)/20 = 2.5th percentile. The 2nd ordered observation is the 100(2 – 0.5)/20 = 7.5th percentile.

The reason for using 100(j – 0.5)/n instead of 100j/n is to avoid the largest observation being the 100th percentile.

Question: How do you find the 5th percentile when n = 20?

Linear interpolation can be used. Because 5 is half way between 2.5 and 7.5, you can use the average of 2.5 and 7.5 percentiles.

qth quantile – Numerical value where at least 100q percent of the items are less than or equal to this value and (100-100q)% of the items are greater than or equal to this value. Note that 0 < q < 1.

A quantile is the same as a percentile, but it is just said a different way. For example, the 0.95 quantile is the 95th percentile.

Some of the important percentiles and quantiles:

* Median = 50th percentile = 0.5 quantile = Q2
* 25th percentile = 0.25 quantile = Q1 = 1st quartile
* 75th percentile = 0.75 quantile = Q3 = 3rd quartile

Example: Cholesterol (cholesterol.R)

Note that the observations are already ordered.

> y <- c(133, 137, 148, 149, 152, 167, 174, 179, 189, 192,

 201, 209, 210, 211, 218, 238, 245, 248, 253, 257)

> # While not needed, put the data into a data frame

> set1 <- data.frame(y = y)

> head(set1)

 y

1 133

2 137

3 148

4 149

5 152

6 167

> type1 <- quantile(x = set1$y, probs = seq(from = 0.025,

 to = 0.975, by = 0.05), type = 1)

> type5 <- quantile(x = set1$y, probs = seq(from = 0.025,

 to = 0.975, by = 0.05), type = 5)

> type7 <- quantile(x = set1$y, probs = seq(from = 0.025,

 to = 0.975, by = 0.05)) #type = 7 is the default

> data.frame(y.sort = sort(set1$y), type5, type1, type7)

 y.sort type5 type1 type7

2.5% 133 133 133 134.900

7.5% 137 137 137 141.675

12.5% 148 148 148 148.375

17.5% 149 149 149 149.975

22.5% 152 152 152 156.125

27.5% 167 167 167 168.575

32.5% 174 174 174 174.875

37.5% 179 179 179 180.250

42.5% 189 189 189 189.225

47.5% 192 192 192 192.225

52.5% 201 201 201 200.775

57.5% 209 209 209 208.400

62.5% 210 210 210 209.875

67.5% 211 211 211 210.825

72.5% 218 218 218 216.425

77.5% 238 238 238 232.500

82.5% 245 245 245 242.725

87.5% 248 248 248 246.875

92.5% 253 253 253 250.875

97.5% 257 257 257 255.100

> quantile(x = set1$y, probs = seq(from = 0.025, to =

 0.075, by = 0.01), type = 5)

 2.5% 3.5% 4.5% 5.5% 6.5% 7.5%

133.0 133.8 134.6 135.4 136.2 137.0

> quantile(x = set1$y, probs = seq(from = 0.025, to =

 0.075, by = 0.01), type = 1)

2.5% 3.5% 4.5% 5.5% 6.5% 7.5%

 133 133 133 137 137 137

> quantile(x = set1$y, probs = seq(from = 0.025, to =

 0.075, by = 0.01), type = 7)

 2.5% 3.5% 4.5% 5.5% 6.5% 7.5%

134.900 135.660 136.420 137.495 139.585 141.675

Comments:

* The type = 5 argument value leads to the calculation in the same way as described previously.
* I gave type = 1 and type = 7 just as a way to show you that there are different definitions and they could lead to different answers. You are not responsible for knowing how these other types are done.
* Notice how the linear interpolation is done with type = 5.

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

Perhaps a power company needs to know whether the wind speed is greater than 5 MPH at least 80% of the time. If it is, a wind turbine could be profitable at a particular location.

> quantile(x = wind$y, probs = 0.2, type = 5)

 20%

6.18

> quantile(x = wind$y, probs = 0.8, type = 5)

 80%

14.5