# *Packet 2*

*Todd CadwalladerOlsker*

*2022-01-29*

*A .pdf copy of this packet is available at <Packet2.pdf>.*

## *Descriptive Statistics*

In order to describe a data set, we need to *summarize* it. The phrase “Exploratory Data Analysis” is used in *Introduction to Modern Statistics*. We can summarize data by visualizing it, describing it numerically, or (even better) doing some of each.

I’ll try to keep these packets brief, they won’t contain everything there is to say about the material. You should reference the textbooks and keep your own notes as well.

### *Medians, Quartiles, and the Five-Number Summary*

When we look at **numeric** variables, we can look at *median*-based statistics or *mean*-based statistics. The *median* is the value of the middle data point (or when there are an even number of data points, the value halfway between the two middle data points). In R, we can find the median of a data set with:

```
median(county$poverty,na.rm = TRUE)
```

Here, the `na.rm = TRUE` tells R to ignore values that have missing data. (Try the command without the `na.rm = TRUE` and see what happens!) You can also just use `median(county$pop2017,TRUE)` for the same result.

In this example, we have the median of the *population*, since we have information about every county. (well, except for the NAs, but it’s not like we didn’t ask for that information.) This is technically a *parameter*, not a *statistic*. If we only take a small sample of the population, then we are calculating the sample median statistic.

Now, a *measure of central tendency* like the median is not worth very much without a *measure of spread* to go along with it. For the median, one way to get a handle on the spread is to also report the maximum, minimum, and 1st and 3rd quartile values. Together, these are called a *five-number summary*.

To be a bit more technical, the measures of spread are actually the *range*, which is the difference of the maximum and minimum, and the *interquartile range*, or IQR, which is the difference of the 3rd and 1st quartiles. The five number summary contains all this information and more.

Recall that the 1st quartile is the “median” of the minimum value and the median, while the 3rd quartile is the “median” of the maximum value and the median… sort of. In fact, it’s actually silghtly more complicated than that. For example, consider the simple example of the data

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$$

The median is 6, but what should the first quartile be? Should we include 6, and get a median of 3.5, or exclude it, at get a median of 3? Try `fivenum(c(1:11))` or `summary(c(1:11))` and see what R does!

Now try both of the following:

```
fivenum(c(1:10))
summary(c(1:10))
```

Why do these give different results? If you want to get *very* deep into the weeds, check out `help("quantile")` and read about the nine different “types” it uses to calculate quantiles. You can choose a type in `summary` with `summary(data,quantile.type=X)` where X is the type described in the help file. The R documentation is not very easy to read, but the Mathworld page for Quantile is a bit better, see `https://mathworld.wolfram.com/Quantile.html`.

For example, try
`summary(c(1:10),quantile.type=2)`

```
fivenum(county$poverty)
  # The fivenum() function defaults to na.rm = TRUE,
  # so we don't need to add it in.

summary(county$poverty)
  # This is a little friendlier to the eyes,
  # and also includes the mean.
```

We can visualize the five number summary as a plot:

```
county %>% na.omit(poverty) %>% ggplot(aes(x = poverty)) +
  stat_boxplot(coef = 10, geom = "errorbar") +
  geom_boxplot(coef = 10)
```

Traditionally, the boxplot includes vertical lines for each of the five numbers in the summary.

A couple of things are going on here: First, notice that we needed to add “error bars” to visualize the whiskers, gglot does not include them as standard.

Second, notice that we set `coef = 10`. What does that mean?

Third, notice that we have values on a vertical axis. These are meaningless, we can suppress those with the following code:

```
county %>% na.omit(poverty) %>% ggplot(aes(x = poverty)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot() +
  labs(title = "Poverty Rates of U.S. Counties",
       y = NULL,
       x = "Poverty rate",
       subtitle = "Percentage of residents living in poverty",
       caption = "Source: county dataset") +
  scale_y_continuous(breaks = NULL, labels = NULL)
```

Also notice I left off `coef = 10` this time. Most of the highest-poverty counties are considered “outliers”, as they are more than 1.5 times the IQR away from the median. This is only one possible definition of “outliers”, and it may or may not be a good definition, depending on the data. The `coef` variable overrides this by changing the number of multiples of the IQR we need to be from the median before we call something an outlier. Use `help(geom_boxplot)` for more.

## *Mean, Variance, and Standard Deviation*

In many cases, the *mean* is a more useful measurement than the median. In many other cases the median is more useful – it depends on the situation. The mean is the sum of the values of our data points, divided by the number of data points – exactly how you learned to calculate the average since forever. The calculation is the same whether we want to calculate a population mean or sample mean.

The mean is more useful in the sense that we can do a lot more with it, but it is not as *robust* as the median. It's much more sensitive to outliers and skewed data.

The natural measure of spread to pair with the mean is the *variance*, or the *standard deviation*. The definition of variance starts with a natural idea: let's measure the distance between the value of each data point and the mean (this is the *deviation from the mean* or just *deviation* of each data point). Then, we could find the average deviation.

Let’s try this out: suppose we have sample data of

$$2, 2, 3, 7, 8, 9$$

In R, we can save this as a vector, then make it a data frame that we can manipulate with dplyr:

```
sample_data <- data.frame(c(2,2,3,7,8,9))
colnames(sample_data) <- "Value"
sample_data <- sample_data %>%
  mutate(Deviation = Value - mean(sample_data$Value))
mean(sample_data$Deviation) %>% round(15)
```

You should find that the mean deviation is 0! The positive deviations are cancelled out by the negative deviations. This will always happen in any data set, since the average value of the data points is 0 units away from the average. Therefore, this is not a very useful measurement of spread. So what can we do?

Note the use of `round` in the last command; here we are rounding to 15 digits. If you just use `mean()` in the last step, you will get a very small number in scientific notation. The actual value of the mean is in fact 0, but you are seeing a very small number due to a *floating point error* in R. These are very common in any programming language, and dealing with them is a necessary evil.