3 – Univariate Analysis

Quantitative variables

Some basic functions for univariate analysis on a quantitative variable.

```
# Lengths superheroes
lengths <- c(141, 198, 143, 201, 184)
```

Measures of central tendency

```
mean(lengths)                       # mean or average
```

```
## [1] 173.4
```

```
median(lengths)
```

```
## [1] 184
```

Measures of dispersion

```
range(lengths)                      # minimum & maximum
```

```
## [1] 141 201
```

```
abs(max(lengths) - min(lengths))    # range
```

```
## [1] 60
```

```
summary(lengths)                    # Quartiles, etc.
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   141.0   143.0   184.0   173.4   198.0   201.0
```

```
IQR(lengths)                        # Interquartile range
```

```
## [1] 55
```

```
sd(lengths)                         # standard deviation of a *sample*
```

```
## [1] 29.38197
```

Formula breakdown

Mathematical formulae can often be translated to R quite straightforwardly. Take e.g. the formula for the mean:

$\mu = \sum_{x=1}^{n} x_i n \mu = \sum x=1nxin$
In R, this becomes:

```
sum(lengths) / length(lengths)
```

```
## [1] 173.4
```

The same goes for the variance and standard deviation. In the example below, we use the definition of population variance (denominator n).
$\sigma^2 = \sum_{i=1}^{n}(x_i - \mu)^2 / n$
In the code below, we break down the formula in parts:

```
res_1 <- lengths - mean(lengths)    # difference of data points with mean
res_2 <- res_1^2                    # squared differences
res_3 <- sum(res_2)                 # take the sum
variance <- res_3 / length(lengths) # calculate average
stdev <- sqrt(variance)             # take the square root
```

Or, the entire calculation of variance in one formula. Can you recognize the mathematical formula in the R-code?

```
variance <- sum((lengths - mean(lengths))^2) / length(lengths)
```

Calculations by group

Often, you want to get calculations of mean, standard deviation, etc. grouped by some factor. This can be done using the `aggregate` function, and the "group by" `~` operator.

We show some examples from the `mtcars` dataset.

```
View(mtcars)
# Show the average mileage per gallon for cars with and
# without an automatic transmission.
aggregate(mpg ~ am, data = mtcars, FUN = mean)
##   am      mpg
## 1  0 17.14737
## 2  1 24.39231
# Standard deviation
aggregate(mpg ~ am, data = mtcars, FUN = sd)
##   am      mpg
## 1  0 3.833966
## 2  1 6.166504
# Apply the summary function
aggregate(mpg ~ am, data = mtcars, FUN = summary)
##   am mpg.Min. mpg.1st Qu. mpg.Median mpg.Mean mpg.3rd Qu. mpg.Max.
## 1  0 10.40000    14.95000   17.30000 17.14737    19.20000 24.40000
## 2  1 15.00000    21.00000   22.80000 24.39231    30.40000 33.90000
```

Remark that the `data` parameter allows you to reference the column names directly, instead of having to use the notation `mtcars$mpg ~ mtcars$am`.

Qualitative variables

In R, qualitative variables are called *factors*. As an example, we'll use the `esoph` dataset from the `datasets` package, available in R.

```
?esoph

View(esoph)
```

Measures of central tendency

About the only measure of central tendency for a factor/qualitative variable is the mode. There is no actual mode function in R, but you can find it in several ways. The first is to print a frequency table and read the maximum from there:

```
freq_tab <- table(esoph$agegp) # Calculate a frequency table

freq_tab
```

```
##
## 25-34 35-44 45-54 55-64 65-74   75+
##    15    15    16    16    15    11
```

```
summary(esoph$agegp)            # The summary function applied to a factor
```

```
## 25-34 35-44 45-54 55-64 65-74   75+
##    15    15    16    16    15    11
```

The output of both `table` and `summary` (applied to a factor) is identical.

From there, we could look for the value that occurs the most:

```
which.max(table(esoph$agegp)) # Only works for the case with a single mode
```

```
## 45-54
##     3
```

```
names(freq_tab)[freq_tab == max(freq_tab)] # Also works for multimodal vari
ables
```

```
## [1] "45-54" "55-64"
```

It's a bit convoluted, but in R, this is the only way to calculate the mode.

Charts in R

For univariate statistics, the most common chart types are:

- boxplot, that shows the spread of a (quantitative)
- bar chart, which can be used for
    - showing the values of a quantitative variable

    - showing the frequencies of a qualitative variable
- histogram, a variant of the bar chart for frequencies, where ranges of x-values are taken together in “buckets”

```
# Chart types
barplot(lengths)  # bar chart (without any fancy layout add-ons)
```

```
boxplot(lengths)  # boxplot
```

```
# Example "Active Duty Personnel, 1998"
active_duty_personnel <- c(492, 363, 381, 176)

barplot(active_duty_personnel)
```

```
pie(active_duty_personnel)        # Shown for reference, avoid pie charts!
```

```
# Plot of the frequencies of a qualitative variable
barplot(table(esoph$agegp))
```

As an example of a histogram, we take the `cars` dataset as an example.

```
# For reference, a boxplot of the data
boxplot(cars$dist, horizontal = TRUE)
```

```
# A simple histogram, the hist function decides on the number
# of buckets
hist(cars$dist)
```

```
# Only 4 "breaks" between buckets => 5 buckets
hist(cars$dist, breaks = 4)
```

```
# Specify boundaries between buckets explicitly
```

```
hist(cars$dist, breaks = c(0,30,60,90,120))
```

More elaborate charts

Charts can be extended with titles, legends, colours, etc. A few examples are given here, you can find a lot more on the Internet. Remark that there is a separate chart library in R, called `ggplot`, that is not discussed in this guide.

```
heroes <- c("Spiderman", "Batman", "Superman",
            "Deadpool", "Catwoman")
barplot(lengths,
        main = "Lengths of super heroes",
        names = heroes,
        xlab = "Hero",
        ylab = "Length (cm)",
        col = "light blue"
        )
```

```
military_categories <- c("Army", "Air Force", "Navy", "Marine Corps")
barplot(active_duty_personnel,
        names = military_categories)
```