Lab 2: Descriptive statistics

STAT218

By now you have seen basic descriptive and graphical summaries. But what determines when some descriptive statistics are "better" to use than others?

This lab has two objectives:

1. Teach you to compute descriptive statistics and prepare graphical summaries for a single variable in R
2. Learn when and why to use certain descriptive statistics in place of others

We will focus on how outliers, or observations far from center, affect different descriptive measures. In statistics, the term for this is **robustness**: a robust statistic is less affected by the presence of outliers.

Accessing data

For this lab we'll use the FAMuSS dataset, as in lecture. This dataset is stored in the `oibiostat` package, which is an R package companion to the Vu and Harrington textbook.

```
# load openintro biostat package
library(oibiostat)

# load famuss data
data(famuss)

# open data documentation
?famuss
```

The command above makes the dataset available in your environment as a data frame named `famuss` in which the columns represent variables and the rows represent observations. You can see the first few rows using `head()`:

```
head(famuss)
```

```
  ndrm.ch drm.ch    sex age      race height weight actn3.r577x    bmi
1      40     40 Female  27 Caucasian   65.0    199          CC 33.112
2      25      0   Male  36 Caucasian   71.7    189          CT 25.845
3      40      0 Female  24 Caucasian   65.0    134          CT 22.296
4     125      0 Female  40 Caucasian   68.0    171          CT 25.998
5      40     20 Female  32 Caucasian   61.0    118          CC 22.293
6      75      0 Female  24  Hispanic   62.2    120          CT 21.805
```

You can extract a vector of the observations for any particular variable from the dataframe as follows: `famuss$[variable name]`. We will be performing calculations one column at a time, so you'll need to be able to extract and store a column as a new R object.

```
# extract the age variable
famuss$age

# extract the bmi variable
famuss$bmi

# store the age column as a vector
age <- famuss$age
```

i Your turn

```
# try it yourself: pick a variable to extract and store
```

Graphical and tabular summaries

Our most basic summaries are **frequency distributions**. For categorical variables, these are simply observation counts by category; for numeric variables, values must be *binned* and then tabulated.

For **categorical** variables (*i.e.,* character vectors or factors):

- `table()` will tabulate the number of occurrences of unique values in a vector
- *for a categorical variable,* `plot()` will create a barplot of the frequency distribution

For **numeric** variables:

- `hist()` will create a histogram

```r
## categorical summaries
race <- famuss$race

# frequency distribution
table(race)

# barplot of frequency distribution
plot(race)
```

```r
## quantitative summaries
age <- famuss$age

# histogram; 'breaks = ...' controls the binning
hist(age, breaks = 20)
```

A quick word about functions. The inputs to functions are called *arguments*. The histogram function `hist()` requires a data argument, in this case `age`, to make the plot. However, the function also has several optional arguments that control things like labels, binning, axis limits, and so on. The `breaks = 20` part of the last command is an example of an optional argument. Most functions in R have optional arguments that control their behavior.

> **Your turn**
>
> Try making a table, barplot, and histogram on your own with some of the other variables. When generating the histogram, adjust the number of bins using the `breaks = ...` argument. Experiment to see how the shape of the distribution appears at various binning resolutions; then pick a number of breaks that you feel reflects the data best.
>
> ```r
> # make a table of the frequency distribution of genotypes
>
> # make a barplot of the frequency distribution of genotypes
>
> # make a histogram of dominant arm change; play with the binning!
> ```

Numerical summaries

In class we discussed several **descriptive statistics** for numeric variables.

- measures of center: mean, median
- percentiles: quartiles, min, max

These statistics are so commonly used that they have their own functions in R.

```
# the median gets its own function
median(age)

# ditto mean
mean(age)

# ... and minimum and maximum
min(age)
max(age)

# 30th percentile of age ("quantile" is another term for percentile)
quantile(age, probs = 0.3)

# 30th *and* 60th percentile of age
quantile(age, probs = c(0.3, 0.6))
```

The five-number summary is the collection of the percentiles that give the minimum, quartiles, and maximum.

Your turn

Try computing the five-number summary using a variable of your choice. Be sure you pick a numeric and not a categorical variable.
Pay attention to how the `probs = ...` argument to the `quantile()` function can be used to calculate multiple percentiles at once. Use this feature to calculate all five numbers in the five-number summary with one command.

```
# choose a *quantiative* variable from the dataset

# compute the five-number summary using quantile()

# compare the mean and the median. are they close?
```

Alternatively, the `summary()` command will do the work for you.

```
summary(age)
```

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
17.0    20.0    22.0    24.4    27.0    40.0
```

Exploring robustness

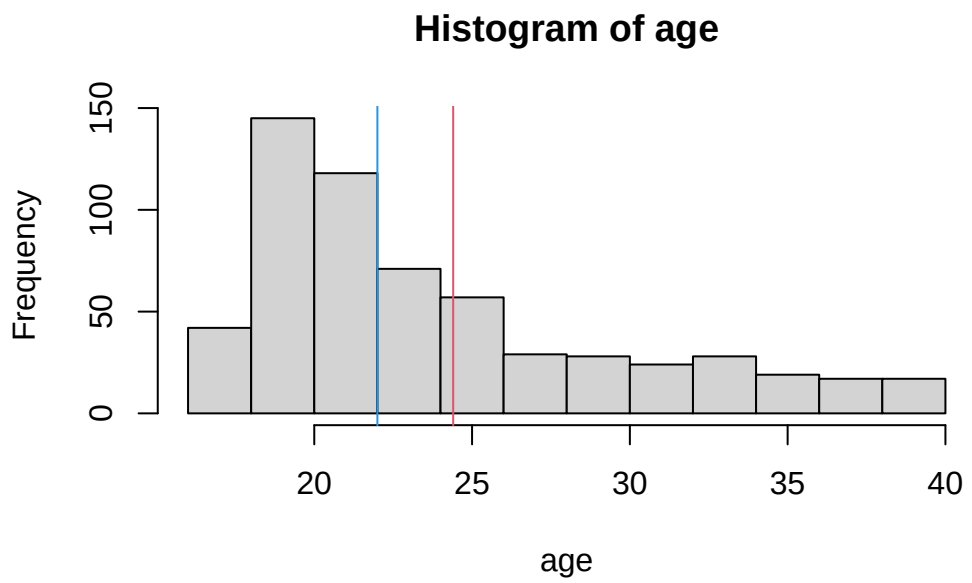
Statistics based on percentiles are in general insensitive to outliers, unless there's a large group of outlying observations. In this sense they are robust statistics.

An easy way to see this is to consider the median (middle value or 50th percentile). The maximum observation could be arbitrarily large without changing the middle value, so the median will be the same whether the largest value is 10 or 10,000. The mean, by contrast, does not share this property.

```
# make up some observations between 1 and 100
x <- sample(1:100, size = 20)
x

# median of made up observations, plus 101
median(c(x, 101))

# median of made up observations, plus 1M
median(c(x, 1000000))

# same comparison, but with mean
mean(c(x, 101))
mean(c(x, 1000000))
```

So in the presence of outliers, the median will capture the center of the distribution of values more accurately. In fact, even if the distribution is simply skewed, the mean will shift away from center.

```
# distribution of ages is right-skewed
hist(age)

# plot the mean and median on top of the histogram
hist(age)
abline(v = mean(age), col = 2)
abline(v = median(age), col = 4)
```



This difference is useful — the comparison of median and mean can indicate the direction and amount of skewness present.

```
# mean > median ---> right-skewed
summary(age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   17.0    20.0    22.0    24.4    27.0    40.0
```

Your turn

Calculate the numeric summary for percent change in dominant arm strength and see if you can determine the direction and magnitude of skewness *based on this summary only.* The amount of skewness is a bit subjective — so just discuss and make a determination about whether the difference between median and mean seems sizeable.

```
# check the numeric summary for percent change in dominant arm strength
# can you tell the direction of skewness?? does it seem very skewed??
```