Lab 2: Descriptive statistics

STAT218

This lab has two objectives:

1. Teach you to compute descriptive statistics and prepare graphical summaries for a single variable in R
2. Allow you to explore the concept of robustness and learn when and why to use certain descriptive statistics in place of others

In statistics, the term **robustness** refers to sensitivity to outliers; outliers are observations far from center. As you will see in this lab, certain descriptive summaries are more sensitive to outliers than others.

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)


## quantitative summaries
age <- famuss$age

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

> **ℹ 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!
> ```

Creating a table of the frequency distribution of a numeric variable takes an extra step: you'll need to bin the values into intervals first, and then make the table just as you would with a categorical variable.

```r
# notice what happens by default
table(age)

# bin age into 6 brackets
age_binned <- cut(age, breaks = 6)

# then make a table
table(age_binned)
```

```
# manually specify the cutoffs
age_binned <- cut(age, breaks = c(15, 20, 30, 40))
table(age_binned)

# show *relative* frequency distribution
table(age_binned)/length(age_binned)
```

> **ℹ Your turn**
>
> Try doing the 'manual' binning illustrated above on your own with the `weight` variable.
> Pay attention to how the `breaks = ...` argument to the `cut()` function is used to
> specify bin limits.
>
> ```
> # bin body weight into 10 intervals
>
>
> # tabulate frequency distribution
>
>
> # repeat with bin limits 80-130, 130-180, 180-220, 220-320
> ```

Usually, you can skip this step and just make a histogram with the appropriate number of
bins, but it's helpful to know how to tabulate in case the need ever arises.

Numerical summaries

In class we discussed several **descriptive statistics** for numeric variables, including:

- percentiles
- mean
- standard deviation

These statistics together formed what we said was the standard numerical summary usually
reported: the "five number summary" (minimum, quartiles, and maximum) together with the
mean and standard deviation.

```
# 30th percentile of age
quantile(age, probs = 0.3)

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

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

# ditto mean
mean(age)

# ... and standard deviation
sd(age)

# ... aaand interquartile range
IQR(age)

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

> **ℹ Your turn**
>
> Try computing the numeric summaries above 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 most of 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*.

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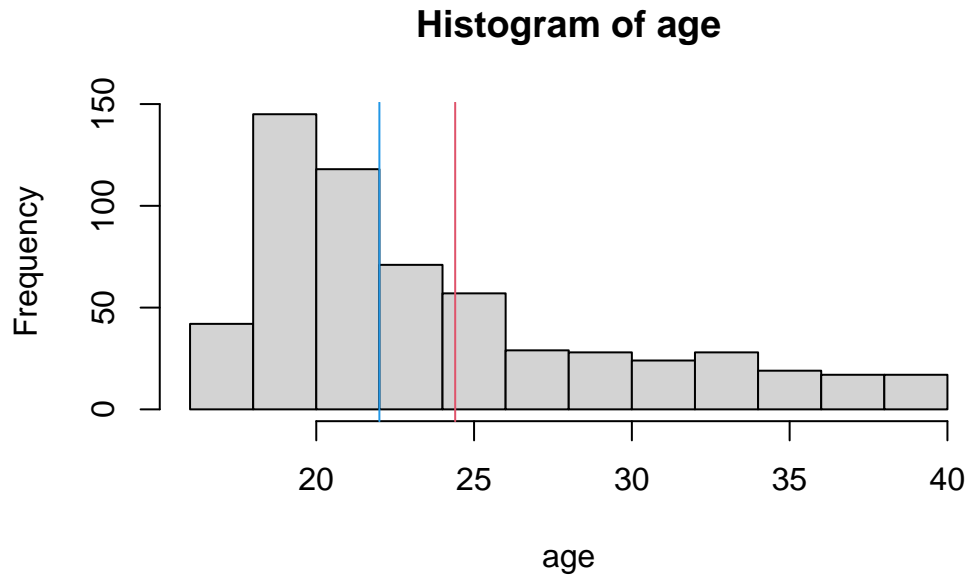
So in the presence of outliers, the median will capture the center of the distribution of values more accurately. In fact, even if the distribution is simply skewed, the mean will shift away from center.

```
# 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)
```

Histogram of age

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
```

> **ℹ Your turn**
>
> See if you can put this property into practice: 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??
> ```

Let's explore how some of the other descriptive statistics we've discussed behave in response to outliers. Specifically, measures of spread: standard deviation and IQR.

The variable you just looked at — dominant arm percent change in strength — has a group of observations at 100%. If these are removed, the standard deviation increases by 10%, but the IQR only increases by 5%.

```
# extract dominant arm percent change in strength
drm <- famuss$drm.ch

# drop the observations over 80%
drm.drop <- drm[drm < 100]

# compute the numeric summary with and without outliers
summary(drm)
summary(drm.drop)

# compare standard deviations
sd(drm)/sd(drm.drop)

# compare interquartile ranges
IQR(drm)/IQR(drm.drop)
```

This may not seem very notable, so let's make up an example that's a bit more extreme: let's add a very large positive observation, say, 1000. Then, the IQR does not change at all, but the standard deviation more than doubles!

```
# add a large observation
drm.add <- c(drm, 1000)

# compare IQR with and without
IQR(drm.add)/IQR(drm)

# compare SD with and without
sd(drm.add)/sd(drm)
```

The differences in robustness between IQR and standard deviation, and between mean and median, are largely why *both* the five-number summary *and* the mean and standard deviation are reported. When these statistics differ dramatically, it is most likely due to the presence of outliers!

> **ⓘ Your turn**
>
> Compute the numeric summary for a variable from a different dataset and *based on this alone* attempt a guess at whether there are outliers. If so, are they more likely outliers

to the left or right?

```r
# load a new dataset (census)
data(census.2010)

# number of doctors per state (thousands)
doctors <- census.2010$doctors

# compute numeric summary -- guess whether there are outliers?

# make a histogram or boxplot to confirm your guess

# bonus: which state??
```