Lab 2: Intro to Data Summaries and Graphics in R

STAT 310, Spring 2023

NHANES Data Set

Here we consider a data set from the National Health and Nutrition Examination Survey (NHANES), which is a program designed to assess the health-status of people in the United States. The survey is unique since it combines interviews and physical examinations. The NHANES interview includes demographic, socioeconomic, and health-related questions. The examination consists of medical and physiological measurements, as well as laboratory tests administered by trained medical personal. The NHANES website contains a complete description of the data: https://www.cdc.gov/nchs/nhanes/about_nhanes.htm

For this lab we focus on a random sample of 1500 people from this survey. Run the following command to load the data set into R:

```
nhanes = readRDS(url("https://ericwfox.github.io/data/nhanes.rds"))
```

To view the variable (column) names and dimension of the `nhanes` data frame type the following commands:

```
names(nhanes)
```

```
##  [1] "Gender"     "Age"        "Education"  "HHIncome"   "Weight"
##  [6] "Height"     "BPSysAve"   "BPDiaAve"   "HealthGen"  "PhysActive"
## [11] "Smoke100"
```

```
dim(nhanes)
```

```
## [1] 1500   11
```

We can see clearly now that the data frame contains 1500 entries (rows) on 11 variables. Each of the variables corresponds to a questions from the survey, or a measurement taken during the physical examination. Descriptions of the variables are provided below:

- `Gender`: gender of participant, coded as `male` or `female`
- `Age`: age in years
- `Education`: education level
- `HHIncome`: household income in thousands of US dollars
- `Weight`: weight in kg

- `Height`: height in cm
- `BPSysAve`: combined systolic blood pressure reading
- `BPDiaAve`: combined diastolic blood pressure reading
- `HealthGen`: self reported rating of participant's health in general
- `PhysActive`: participant does moderate or vigorous-intensity sports, fitness or recreational activities (`Yes` or `No`)
- `Smoke100`: participant has smoked at least 100 cigarettes in their entire life (`Yes` or `No`)

We can have a look at the first several rows of the data with the command

```
head(nhanes)
```

```
##   Gender Age    Education HHIncome Weight Height BPSysAve BPDiaAve HealthGen
## 1 female  23  High School     87.5   72.3  160.0      106       66      Good
## 2   male  64 College Grad    100.0   79.4  175.7      127       90     Vgood
## 3   male  45 College Grad    100.0   86.1  171.8      106       72      Good
## 4 female  51 College Grad    100.0   53.1  163.9      123       77     Vgood
## 5 female  60 College Grad    100.0   53.0  158.3      146       73      Good
## 6   male  42 College Grad    100.0   89.0  185.2      107       62     Vgood
##   PhysActive Smoke100
## 1        Yes      Yes
## 2         No      Yes
## 3        Yes       No
## 4        Yes      Yes
## 5        Yes       No
## 6        Yes       No
```

You could also look at all of the data at once by typing its name into the console, but that might be unwise here. We know `nhanes` has 1500 rows, so viewing the entire data set would mean flooding your screen. It's better to take small peeks at the data with `head()`

Alternatively, to open a scrollable, spreadsheet display of the data run the command

```
View(nhanes)
```

In-Class Exercise: Which variables in the `nhanes` data frame are numerical? Which variables are categorical?

Numerical Summaries

A good place to start in any data analysis is to compute some descriptive statistics on the variables, and to make some graphics.

For numerical variables, the `summary()` function computes the min, first quartile, median, mean, third quartile, and max.

```
summary(nhanes$Height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   139.9   162.2   169.2   169.2   175.9   199.4
```

We can also compute summary statistics one-at-a-time. For example:

```
min(nhanes$Height)
```

```
## [1] 139.9
```

```
median(nhanes$Height)
```

```
## [1] 169.15
```

```
mean(nhanes$Height)
```

```
## [1] 169.1584
```

```
max(nhanes$Height)
```

```
## [1] 199.4
```

```
sd(nhanes$Height)
```

```
## [1] 9.760015
```

While it makes sense to describe a numerical variable like height in terms of these statistics, what about categorical data? We could instead consider the frequency or relative frequency distribution.

Tables and Bar Plots

For categorical variables, the function `table()` counts the number of observations falling in each category. For example, to see the counts for the general-health of the participants, type

```
table(nhanes$HealthGen)
```

```
##
## Excellent     Vgood      Good      Fair      Poor
##       172       470       610       213        35
```

or instead to look at the proportions, type

```
table(nhanes$HealthGen) / 1500
```

```
##
##  Excellent      Vgood       Good       Fair       Poor
## 0.11466667 0.31333333 0.40666667 0.14200000 0.02333333
```

Next, to make a bar plot of the entries in the table, put the table inside the `barplot()` command.

```
barplot(table(nhanes$HealthGen), xlab = "General Health", ylab = "Count")
```

Notice what we have done here: We computed the table of `nhanes$HealthGen` and then applied the graphical function, `barplot()`. This is an important concept: R commands can be nested. You could also break this into two steps by typing the following:

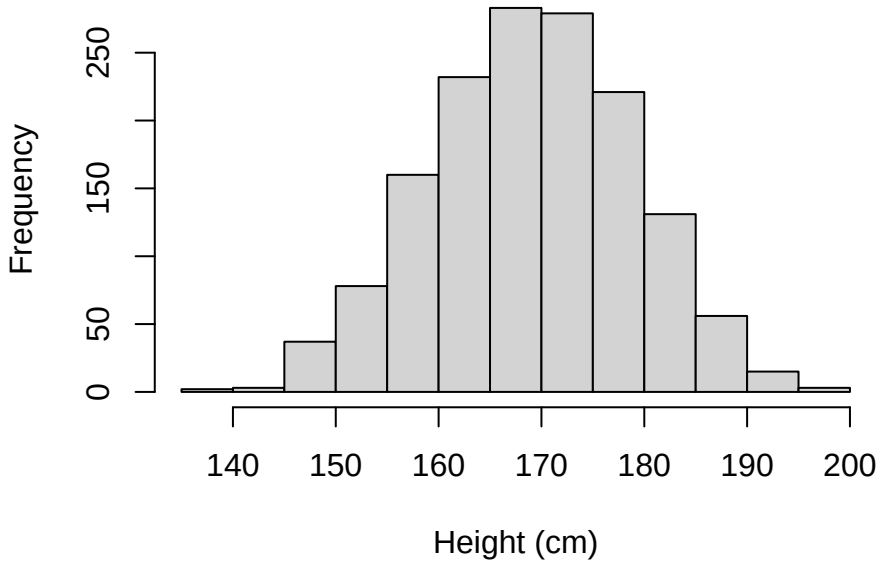
```
health_tb = table(nhanes$HealthGen)
barplot(health_tb)
```

Histogram

Histograms are a useful way to visualize the distribution of a single numerical variable. To construct a histogram, the range of the data is divided into bins of equal width. Then the number of observations falling in each bin are counted. The counts are plotted as rectangles over each bin.

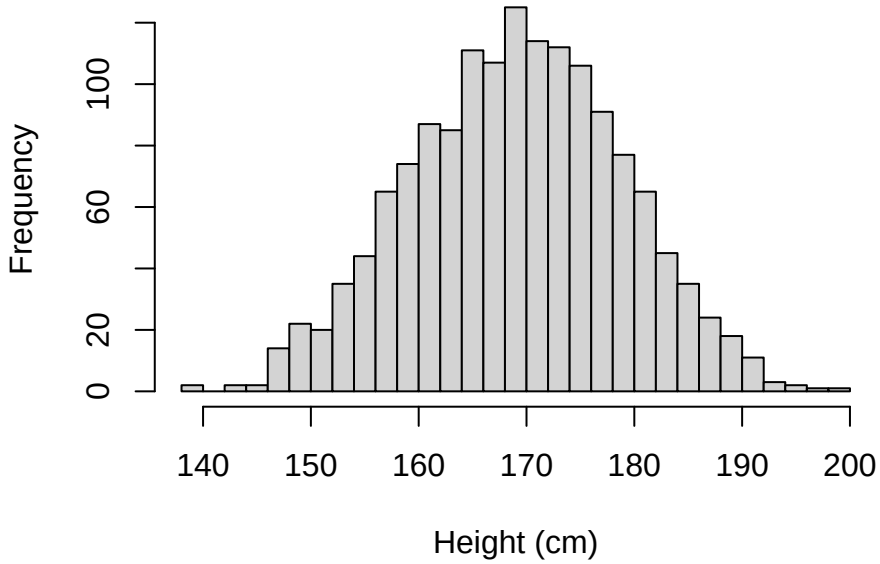
The `hist()` function creates a histogram in R. For example:

```
hist(nhanes$Height, xlab = "Height (cm)", main = "")
```



The number of bins can be changed by specifying the `breaks` argument (see code below). Although, the default number of bins is usually adequate.

```
hist(nhanes$Height, breaks = 30, xlab = "Height (cm)", main = "")
```



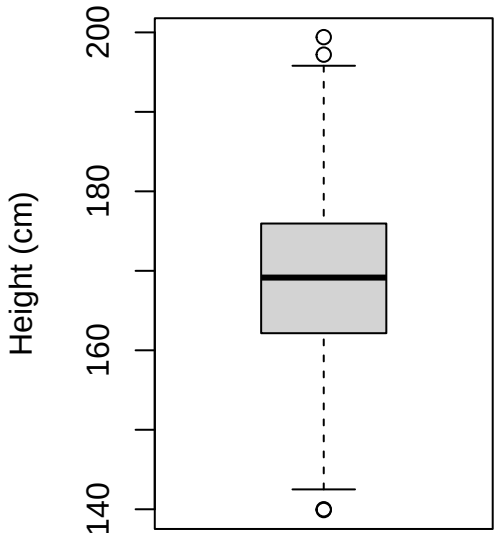
Box Plot

A box plot is another useful way to display the distribution of a numerical variable, and identify potential outliers.

```
summary(nhanes$Height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   139.9   162.2   169.2   169.2   175.9   199.4
```

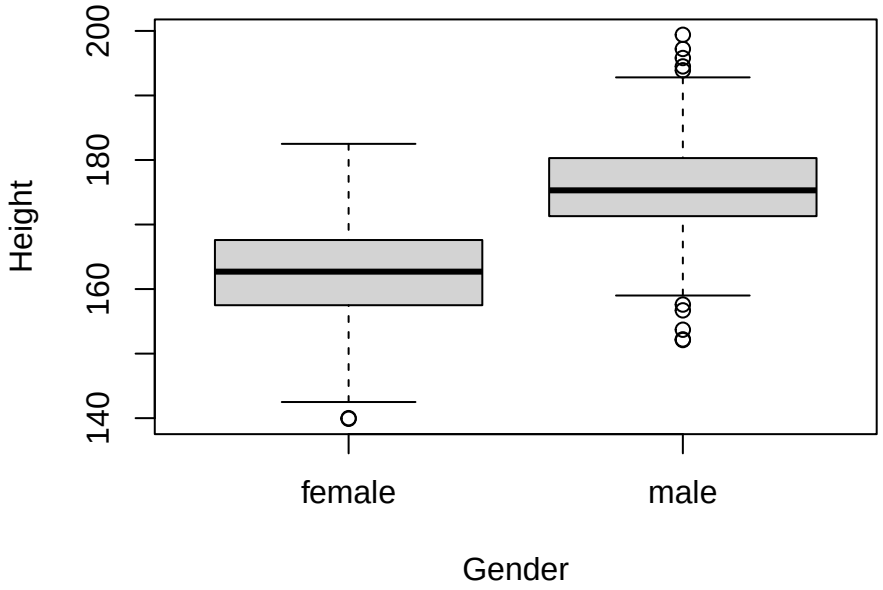
```
boxplot(nhanes$Height, ylab = "Height (cm)")
```



The "box" displays the first quartile (Q_1), median, and third quartile (Q_3), respectively. Recall that the first quartile is the value such that 25% of the data falls below, the median is the value such that 50% of the data falls below, and the third quartile is the value such that 75% of the data falls below. Any points that are outside the whiskers are considered outliers.

We can also create a *side-by-side box plot*:

```
boxplot(Height ~ Gender, data = nhanes)
```

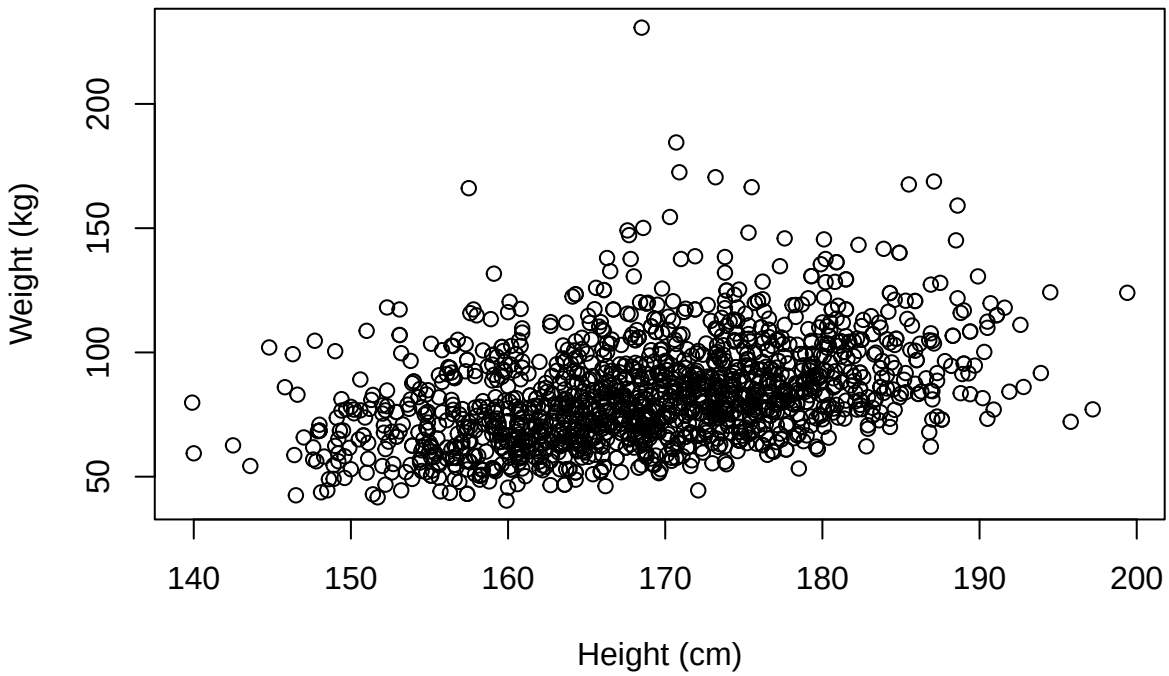


This shows the box plots of `Height` for males and females separately. The function uses the formula notation `y ~ x` where `y` is a numerical variable, and `x` is a categorical variable.

Scatter Plot

Last, scatter plots are used to display the relationship between two numerical variables. To make a scatter plot use the `plot()` function.

```
plot(Weight ~ Height, data = nhanes, xlab = "Height (cm)", ylab = "Weight (kg)")
```



The function again uses the formula notation `y ~ x` where both `y` and `x` are numerical variables.

In-Class Exercise: Run the command below. What do you think `cex` is doing? Does this improve the visualization?

```
plot(Weight ~ Height, data = nhanes, cex = 0.5)
```