MTH 3270 Notes 6

7 Statistical Foundations (7, E.1, E.2, E.5)

7.1 Samples and Populations

- **Data** are often collected by taking a ***random sample*** from a ***population***.

  Random sample data are needed if the goal is to use the data to draw ***statistical inferences*** (conclusions) about the larger population.

7.2 Statistics, Standard Errors, and Sampling Distributions

- A ***statistic*** is a numerical value computed from a set of data.

  Usually, the statistic **summarizes** some feature of the data (e.g. its central value or its spread).

  Statistics are often used to ***estimate*** the corresponding feature of the **population**.

  For example, the ***sample mean*** $\bar{X}$ is used to *estimate* the ***population mean*** μ, the ***sample median*** $\tilde{X}$ is used to *estimate* the ***population median*** $\tilde{\mu}$, and the ***sample standard deviation*** S is used to *estimate* the ***population standard deviation*** σ.

- Two **different random samples** from the **same population** will produce **different values** of a given **statistic**.

  This **chance variation** in the value of a statistic from one random sample to the next is called ***sampling variation***.

- The ***standard error*** of a statistic is a value that measures the magnitude of its **sampling variation**.

  The ***standard error*** is interpreted as the size of a **typical error** made by the **statistic** as an *estimate* of the corresponding **population feature**.

  A **smaller** standard error indicates a **more reliable** estimate.

- The ***sampling distribution*** of a statistic describes the **pattern** of the **sampling variation**.

More formally, it's the *statistic's* **probability distribution**, and indicates the values that the statistic might take and their probabilities.

The **sampling distribution** can be interpreted as describing the values the statistic would take if samples of a given size were drawn a *large number* of times from the population.

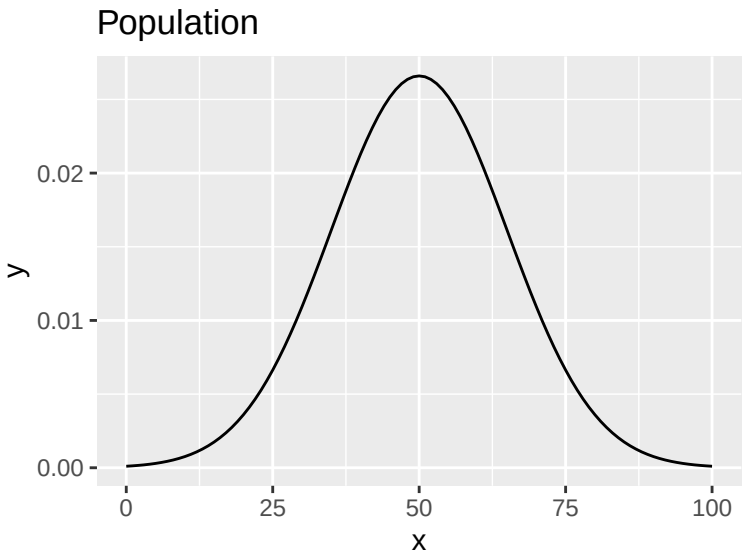
- The **standard error** of a statistic is just the **standard deviation** of the statistic's **sampling distribution**.
- **Two ways** of determining a statistic's **sampling distribution**:
  - Using ***mathematical theory***.
  - Using ***computer simulations***.

(The second method is used when the first one can't be.)

7.3 Simulation

- Consider a **population** represented by a ***normal distribution*** with **mean $\mu = 50$** and **standard deviation $\sigma = 15$**, i.e. a **N(50, 15)** distribution:

```
ggplot(data.frame(x = c(0, 100)), aes(x = x)) +
  stat_function(fun = dnorm, args = list(mean = 50, sd = 15)) +
  labs(title = "Population")
```



To generate a random **sample** of size $n = 10$ from the population, type:

```
my.sample <- rnorm(n = 10, mean = 50, sd = 15)
my.sample

##  [1] 56.76978 51.14709 53.33778 64.04903 46.32967 45.88862
##  [7] 52.51233 58.09736 29.23560 32.23515
```

The **sample mean** $\bar{X}$ is

```
mean(my.sample)

## [1] 48.96024
```

(In this case the **error** in the *estimate* $\bar{X}$ of the population mean $\mu = 50$ is only -1.03976.)

The **sample median** $\tilde{X}$ is

```
median(my.sample)

## [1] 51.82971
```

- To investigate the **standard error** and **sampling distribution** of a statistic like $\bar{X}$, we can ***simulate***, say, 1,000 samples, each of size $n = 10$, from the population and store their statistic values in a vector:

```
# Create an empty vector:
my.sample_means <- rep(NA, 1000)

# Generate a new sample each iteration, save it's mean:
for(i in 1:1000) {
  x <- rnorm(n = 10, mean = 50, sd = 15)
  my.sample_means[i] <- mean(x)
}
```

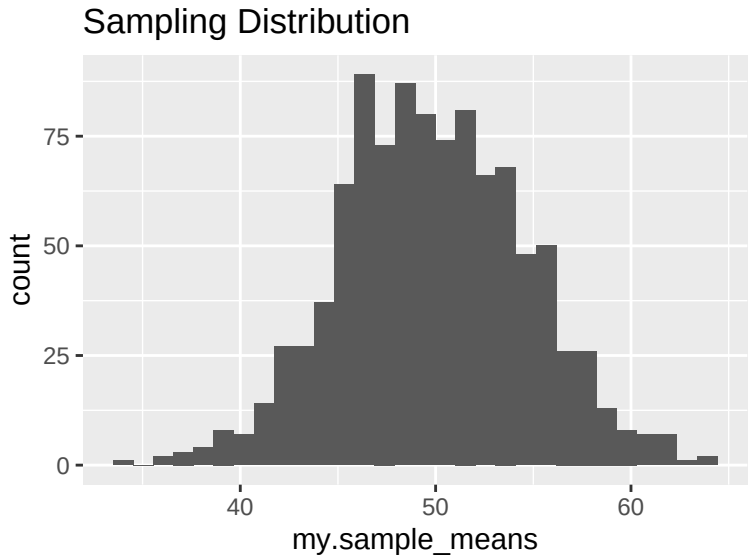
The **standard error** of the statistic (e.g. $\bar{X}$) is the **standard deviation** of the 1,000 values:

```
my.se <- sd(my.sample_means)
my.se

## [1] 4.76193
```

The **sampling distribution** of the statistic (e.g. $\bar{X}$) is depicted by a **histogram** of the 1,000 values:

```
ggplot(data = data.frame(my.sample_means)) +
 geom_histogram(mapping = aes(x = my.sample_means)) +
  labs(title = "Sampling Distribution")
```



Section 7.3 Exercises

Exercise 1 Simulate 1,000 random samples of size $n = 10$ from a N(50, 15) population (i.e. $\mu = 50$ and $\sigma = 15$), and compute the sample mean $\bar{X}$ of each sample.

a) Now compute the mean and standard error of the 1,000 $\bar{X}$ values. Report these two values.

b) You learned in your introductory statistics class that if a random sample of size n is drawn from a N(μ, σ) population, the *sampling distribution* of $\bar{X}$ is N(μ, $\sigma/\sqrt{n}$).

Compare the two values of Part *a* to the theoretical mean and standard error, μ and $\sigma/\sqrt{n}$, of the sampling distribution of $\bar{X}$.

c) Make a histogram of the 1,000 simulated $\bar{X}$ values. Compare the shape, center, and spread of the histogram to the theoretical N(μ, $\sigma/\sqrt{n}$) sampling distribution of $\bar{X}$.

Exercise 2 Simulate 1,000 random samples of size $n = 10$ from a N(50, 15) population (i.e. $\mu = 50$ and $\sigma = 15$), and compute the four statistics below for each sample.

In each case: 1) Report the **mean** and **standard error** of the simulated statistic values, and 2) Plot the simulated values in a histogram and describe the shape, center, and spread of this **sampling distribution**.

a) The **sample median** $\tilde{\boldsymbol{X}}$.

b) The **sample standard deviation** $\boldsymbol{S}$.

c) The **sample minimum** $\boldsymbol{X_{(1)}}$.

d) The **sample maximum** $\boldsymbol{X_{(n)}}$.

7.4 The Bootstrap

- In the last section, we simulated samples from N(μ, σ) populations.

  When the population **isn't** known to be a particular distribution (normal or otherwise), we won't know which distribution to simulate samples from to investigate the **standard error** and **sampling distribution** of a statistic.

  The ***bootstrap*** (B. Efron) refers to using a data set **sampled** from the population as a **proxy** for the **population**, then ***resampling*** from that **data set** to *mimic simulating samples from the population.*

  The **resamples** are drawn ***with replacement*** from the **original data set** using the **same sample size *n*** as the original data set:

```
Bootstrap:

1. Given a set of original data of size n, resample n observations with
   replacement from the data.

2. Compute the statistic of interest from resample.

3. Repeat steps 1 and 2 many, B, times (e.g. B = 1,000).

4. Use the distribution of the B values of the statistic as an approxima-
   tion of the sampling distribution that the statistic would follow if
   the samples had been simulated from the population.
```

- The following function (from the `"dplyr"` package) is useful for generating the **resamples**.

```
sample_n()      # Generate a random sample of n rows from a data frame
```

Data Set: `iris`

This famous (Fisher and Anderson's) `iris` data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of three species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

The five variables are:

`Sepal.Length`	The sepal length.
`Sepal.Width`	The sepal width.
`Petal.Length`	The petal length.
`Petal.Width`	The petal width.
`Species`	The species (Iris setosa, versicolor, or virginica).

- For example, consider the famous built-in `iris` data set:

```
head(iris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

The **sample mean** petal width is $\bar{X} = \mathbf{1.199}$:

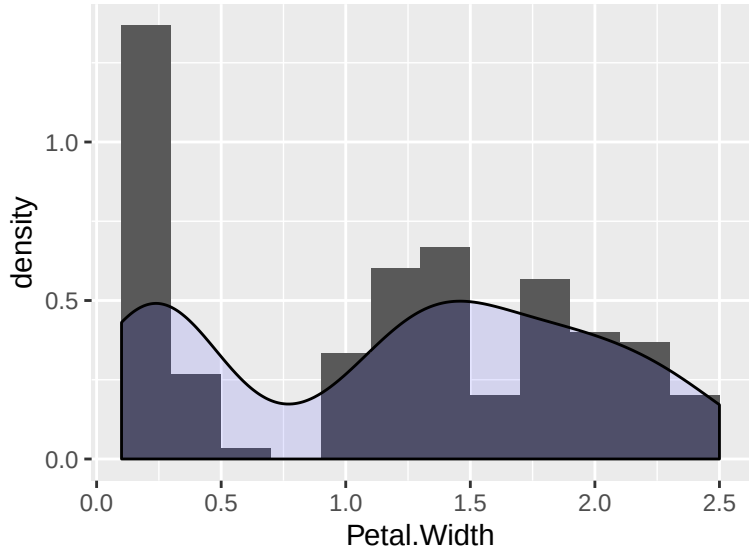
```
mean(iris$Petal.Width)

## [1] 1.199333
```

We want to know how **reliable** this value (1.199) is as an **estimate** of the (unknown) **population mean** petal width μ.

The sample looks like it came from a **non-normal** (i.e. non-bell-shaped) population:

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = stat(density))) +
  geom_histogram(binwidth = 0.2) +
  geom_density(fill = "blue", alpha = 0.1)
```



We'll determine the **standard error** of our *estimate* (1.199) of μ using the **bootstrap** method.

For illustrative purposes, consider first a ***single* bootstrap resample** (i.e. $B = 1$):

```
n <- nrow(iris)       # The original sample size, 150
# Set the seed for regenerating the resample later:
set.seed(3)
# Generate a single resample (for now):
resamp <- sample_n(tbl = iris,
                   size = n,
                   replace = TRUE)
head(resamp)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1          5.0         3.0          1.6         0.2     setosa
## 2          5.6         2.8          4.9         2.0  virginica
## 3          4.9         2.4          3.3         1.0 versicolor
## 4          5.0         3.3          1.4         0.2     setosa
## 5          5.5         2.6          4.4         1.2 versicolor
## 6          5.5         2.6          4.4         1.2 versicolor
```

Setting `replace = TRUE` in `sample_n()` indicates sampling **with replacement**. This allows the resample to include **duplicates** of rows from the original (`iris`) data set. Above, the row `91` appears twice (labeled 5 and 6) in the resample.

Now we'll take $\boldsymbol{B = 1,000}$ bootstrap **resamples**, compute the **sample mean** from each resample, then use these 1,000 sample means to approximate the **standard error** and the **sampling distribution** of $\bar{X}$.

```
n <- nrow(iris)       # The original sample size, 150
B <- 1000             # Number of bootstrap resamples
boot.sample_means <- rep(NA, B)
for(i in 1:B) {
  resamp <- sample_n(iris,
                     size = n,
                     replace = TRUE)
  boot.sample_means[i] <- mean(resamp$Petal.Width)
}
```

```
# boot.sample_means is a vector of 1,000 simulated sample means:
length(boot.sample_means)

## [1] 1000

boot.sample_means[1:5]

## [1] 1.178667 1.178667 1.259333 1.216000 1.132000
```

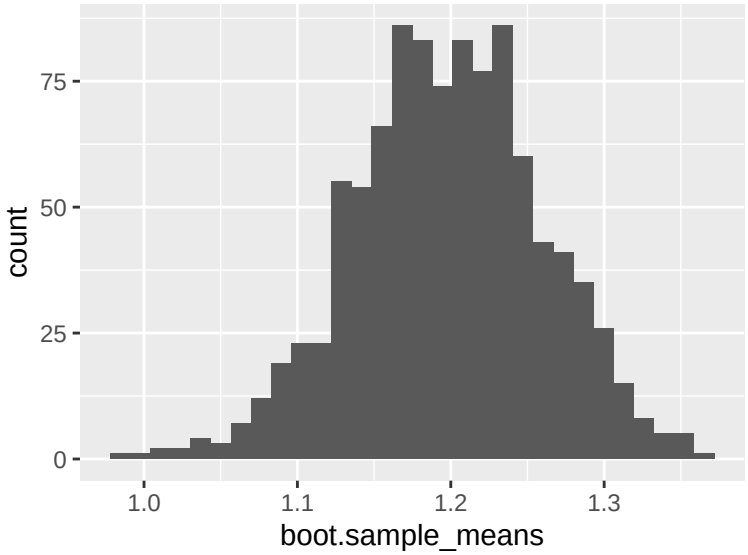
The **standard error** of the statistic (i.e. of the **sample mean**) is the **standard deviation** of the 1,000 values:

```
boot.se <- sd(boot.sample_means)
boot.se

## [1] 0.06103439
```

The (approximate) **sampling distribution** of the statistic (i.e. of $\bar{\boldsymbol{X}}$) is depicted by a **histogram** of the 1,000 values:

```
ggplot(data = data.frame(boot.sample_means)) +
 geom_histogram(mapping = aes(x = boot.sample_means))
```



(The fact that the **sampling distribution** of $\bar{\boldsymbol{X}}$ is approximately **normal**, despite the fact that the **population** was very **non-normal**, is a consequence of the ***Central Limit Theorem***.)

Section 7.4 Exercises

Exercise 3 Use the **bootstrap** method to simulate 1,000 ***resamples*** of size $n = 150$ from the `iris` data set, and using the `Petal.Width` variable, compute the four statistics below for each sample.

In each case: 1) Report the **mean** and **standard error** of the simulated statistic values, and 2) Plot the simulated values in a histogram and describe the shape, center, and spread of this **sampling distribution**.

a) The **sample median** $\tilde{\boldsymbol{X}}$.

b) The **sample standard deviation** $\boldsymbol{S}$.

c) The **sample minimum** $\boldsymbol{X_{(1)}}$.

d) The **sample maximum** $\boldsymbol{X_{(n)}}$.

7.5 Outliers

- An **outlier** is an observation that falls outside the overall pattern of values in a data set.
- For example, the following data represent **numbers of deaths by lightning strikes** in the U.S. for each of the years 1959 - 2005 (in time order), as compiled by the *National Climatic Data Center* from reports by the *National Weather Service*.

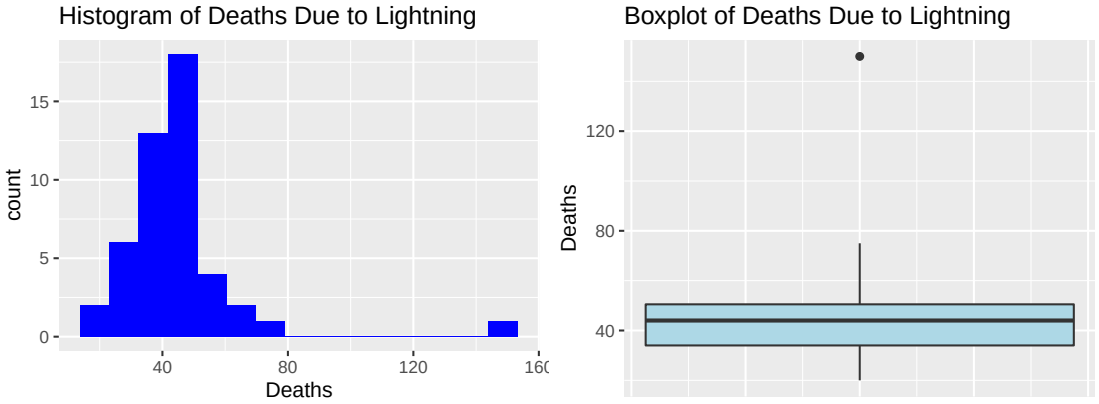


Figure 1

```
Year <- 1959:2005
Deaths <- c(75, 48, 61, 48, 150, 49, 57, 39, 27, 51, 46, 50, 62, 51, 50,
            58, 38, 34, 59, 44, 24, 39, 40, 33, 49, 33, 34, 32, 35, 30,
            23, 39, 36, 25, 20, 32, 43, 52, 42, 44, 46, 51, 47, 51, 44,
            33, 38)
LightningData <- data.frame(Year, Deaths)
```

```
## Histogram
ggplot(data = LightningData) +
  geom_histogram(mapping = aes(x = Deaths), fill = "blue", bins = 15) +
  ggtitle("Histogram of Deaths Due to Lightning")
```

```
## Boxplot
ggplot(data = LightningData) +
  geom_boxplot(mapping = aes(y = Deaths), fill = "lightblue") +
  ggtitle("Boxplot of Deaths Due to Lightning") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```

Regarding the **outlier** (150 deaths in 1963), a *National Weather Service* report states:

> "On December 8, 1963 the crash of a jetliner killing 81 people near Elkin, Maryland, was attributed to lightning by the Civil Aeronautics Board investigators."

- **Comments**:
  - Outlying data values should be **checked for accuracy** (e.g. typos). Inaccuracies should be corrected.
  - Outliers may reveal **important insights**. Outliers **shouldn't** be dropped unless there's a clear rationale.
  - ***Robust*** statistical procedures are ones that aren't unduly affected by outliers (or other data irregularities).

– ***Multivariate outliers*** might not show up in graphs. Instead, ***multivariate outlier detection*** procedures are needed to identify them.

Section 7.5 Exercises

Exercise 4 Multivariate outliers may not show up in graphs that don't show all the variables simultaneously.

Here are lengths (cm) and weights (g) of $\boldsymbol{n = 10}$ female snakes, one of which is an **outlier**.

SnakeID	Length	Weight
1	60	136
2	69	198
3	66	194
4	64	140
5	54	93
6	67	172
7	55	195
8	59	116
9	65	174
10	63	145

```
SnakeID <- 1:10
Ln <- c(60, 69, 66, 64, 54, 67, 55, 59, 65, 63)
Wt <- c(136, 198, 194, 140, 93, 172, 195, 116, 174, 145)
Snakes <- data.frame(SnakeID, Ln, Wt)
```

a) Can you identify the **outlier** in either of these **univariate** graphs (histograms)?

```
ggplot(data = Snakes) +
  geom_histogram(mapping = aes(x = Ln),
                 fill = "blue",
                 color = "white",
                 bins = 5) +
  ggtitle("Histogram of Snakes Lengths")
```

```
ggplot(data = Snakes) +
  geom_histogram(mapping = aes(x = Wt),
                 fill = "blue",
                 color = "white",
                 bins = 5) +
  ggtitle("Histogram of Snakes Weights")
```

b) Can you identify the **outlier** in this **bivariate** graph (scatterplot)? If so, which snake (`SnakeID`) is the **outlier**?

```
ggplot(data = Snakes) +
  geom_point(mapping = aes(x = Ln, y = Wt)) +
  ggtitle("Scatterplot of Weights vs Lengths")
```

7.6 Regression Models: Explaining Variation (7, E.1, E.2)

7.6.1 Simple Linear Regression: One Explanatory Variable

- ***Regression models*** describe **variation** in a ***response*** variable $\boldsymbol{Y}$ as a function of an ***explanatory*** variable $\boldsymbol{X}$.
- A ***simple linear regression analysis*** involves obtaining the **equation** of the **line** that best fits the a scatterplot. The equation is useful for:
  1. **Predicting** the value of $\boldsymbol{Y}$ from a given value $\boldsymbol{X}$ (by plugging the X into the equation of the line).
  2. **Quantifying** a typical **change** in $\boldsymbol{Y}$ associated with a given **change** in $\boldsymbol{X}$ (using the slope of the line).
- We can carry out a **regression analysis** using the "linear model" function:

```
lm()        # Carry out a linear regression analysis by fitting a
            # linear model to a data set.
summary()   # Summarize the results of the regression analysis.
```

  The resulting ***fitted regression line*** has the form

$$\hat{Y} \;=\; \hat{\beta}_0 + \hat{\beta}_1 X\,.$$

  The **intercept** $\hat{\boldsymbol{\beta}}_{\mathbf{0}}$ and **slope** $\hat{\boldsymbol{\beta}}_{\mathbf{1}}$ are referred to as the ***coefficients*** of the **fitted model**.
- One way to obtain **predicted** values of Y is to use the following function.

```
predict()   # Returns the predicted response (Y) values from a fit-
            # ted regression model and a data frame of explanatory
            # (X) values.
```

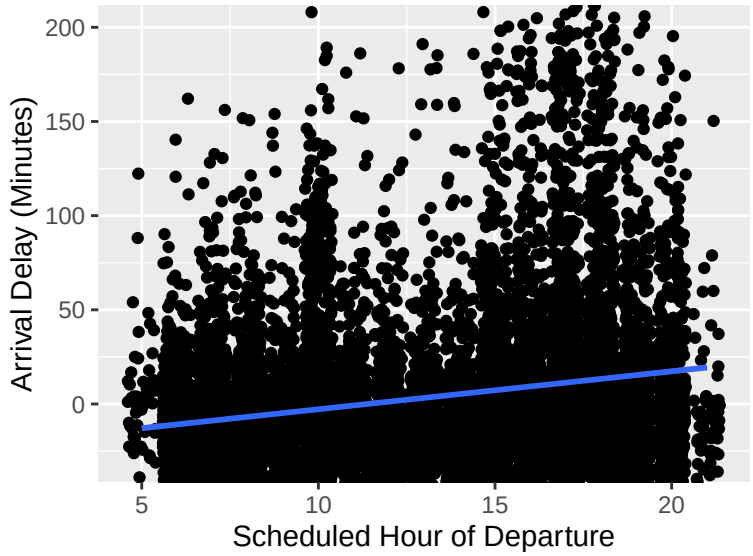
- Consider, as an example, the `flights` data (from the `"nycflights13"` package.

  We'll investigate how the **arrival delay** (`arr_delay`) depends on the **scheduled departure hour** (`hour`) for flights to San Francisco. A plot of these variables is below.

```
library(nycflights13)

SF <- filter(.data = flights, dest == "SFO", !is.na(arr_delay))

ggplot(data = SF, mapping = aes(x = hour, y = arr_delay)) +
  geom_point(position = "jitter") +
  geom_smooth(method = "lm") +
  xlab("Scheduled Hour of Departure") + ylab("Arrival Delay (Minutes)") +
  coord_cartesian(ylim = c(-30, 200))
```



We carry out the **regression analysis** by typing:

```
my.reg <- lm(arr_delay ~ hour, data = SF)
summary(my.reg)

##
## Call:
## lm(formula = arr_delay ~ hour, data = SF)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -97.32 -25.22  -9.17   9.83 993.66
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -22.93267    1.23275  -18.60   <2e-16
## hour          2.01487    0.09154   22.01   <2e-16
##
## Residual standard error: 46.82 on 13171 degrees of freedom
## Multiple R-squared:  0.03548,Adjusted R-squared:  0.03541
```

```
## F-statistic: 484.5 on 1 and 13171 DF,  p-value: < 2.2e-16
```

From the output, the **equation** of the fitted line, which is shown in the plot above, is:

$$\hat{Y} = -22.93 + 2.01X$$

Based on the equation, we'd **predict** the **delay** for a flight whose scheduled departure is at **hour 15** to be:

$$\hat{Y} = -22.93 + 2.01(15) = \mathbf{7.22}$$

minutes.

Each **additional hour** in the **scheduled departure** time typically *increases* the **arrival delay** by **2.01 minutes**.

- Another way to get **predicted values** from a **fitted regression model** is to use `predict()`.

To use `predict()`, we store X value(s) for which we want predicted Y in a data frame, for example:

```
newHour <- data.frame(hour = 15)
newHour

##   hour
## 1   15
```

The variable name needs to be the same as in the data frame used to build the model, (`hour` in this case.)

After creating `my.reg` and the `newHour` data frame, we get the **predicted arrival delay** by typing:

```
predict(my.reg, newdata = newHour)

##        1
## 7.290446
```

Thus the **predicted delay** is **7.29** minutes (which differs from before due to round-off error).

Section 7.6 Exercises

Exercise 5 Here are the `Snakes` data (minus the outlier) from Exercise 4 above:

```
SnakeID <- 1:9
Ln <- c(60, 69, 66, 64, 54, 67, 59, 65, 63)
Wt <- c(136, 198, 194, 140, 93, 172, 116, 174, 145)
Snakes <- data.frame(SnakeID, Ln, Wt)
```

a) After looking at this plot of the data with the fitted regression line, describe the relationship between lengths and weights:

```
ggplot(data = Snakes, mapping = aes(x = Ln, y = Wt)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Scatterplot of Weights vs Lengths")
```

b) Obtain the equation of the line by fitting the linear regression model, with `Wt` as the response (Y) and `Ln` as the explanatory variable (X), by typing:

```
my.reg <- lm(Wt ~ Ln, data = Snakes)
summary(my.reg)
```

c) From the output of Part *a*, the **equation** of the fitted line is:

$$\hat{Y} = -301.09 + 7.19X$$

Based on the equation, what weight would you **predict** for a snake whose length is **62** cm?

d) What's a typical **change** in weight for each **1** cm **elongation**? What about for a **5** cm **elongation**?

Exercise 6 This exercise uses the `flights` data (from the `"nycflights13"` package).

We'll investigate how the **departure delay** (`dep_delay`) depends on the **scheduled departure hour** (`hour`) for flights to San Francisco.

Create the `SF` data set:

```
library(nycflights13)

SF <- filter(.data = flights, dest == "SFO", !is.na(arr_delay))
```

a) After looking at this plot of the data with the fitted regression line, describe the relationship between **departure delay** and **scheduled hour of departure**:

```
ggplot(data = SF, mapping = aes(x = hour, y = dep_delay)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlab("Scheduled Hour of Departure") + ylab("Departure Delay (Minutes)") +
  coord_cartesian(ylim = c(-30, 200))
```

b) Use `lm()` to carry out the **regression analysis**, with `dep_delay` as the response and `hour` as the explanatory variable. Then use `summary()` to view the results.

Report the **equation** of the fitted line.

c) Based on the equation from Part *a*, what departure delay would you **predict** for a flight whose departure hour is **15**?

d) Now, after creating the following data frame,

```
newHour <- data.frame(hour = 15)
newHour
```

use `predict()` to obtain the **predicted delay** for a flight whose departure hour is **15**.

e) By how many minutes does the **departure delay** increase for each **additional hour** in the **scheduled departure**?

- The **vertical deviations** of the points away from the line are called ***residuals***.

Consider the `Snakes` data frame from Exercise 5.

```
Snakes
```

```
##   SnakeID Ln  Wt
## 1       1 60 136
## 2       2 69 198
## 3       3 66 194
## 4       4 64 140
## 5       5 54  93
## 6       6 67 172
## 7       7 59 116
## 8       8 65 174
## 9       9 63 145
```

After fitting the **regression line** to the data:

```
my.reg <- lm(Wt ~ Ln, data = Snakes)
```

the **residuals** are the line segments shown in Fig. 2.

We can obtain the **residuals** via `my.reg$residuals`:

```
library(dplyr)
```

```
Snakes <- mutate(Snakes, Residuals = my.reg$residuals)
```

- A line fits the data "well" if the **residuals** are **small**. The "best" line (according to the ***least squares criterion***) is the one that minimizes the **sum of squared residuals**.
- The ***fitted values*** are defined as

$$\hat{Y}_i \;=\; \hat{\beta}_0 + \hat{\beta}_1 X_i \,,$$

where the X_i's are the observed values of the predictor (used to fit the model). In Fig. 2, the **fitted values** are points **on** the **fitted regression line** where the line segments meet it.

We can obtain the **fitted values** via `my.reg$fitted.values`:

```
Snakes <- mutate(Snakes, FittedVals = my.reg$fitted.values)
```

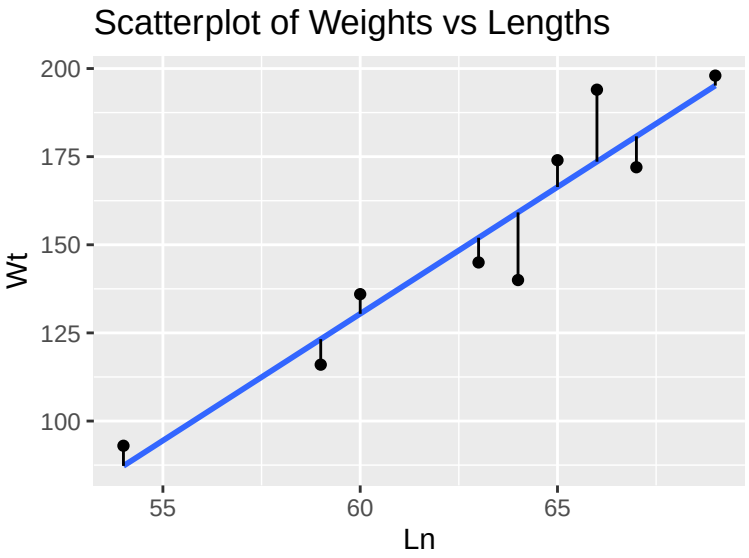


Figure 2

Section 7.6 Exercises

Exercise 7 Here are the `Snakes` data (minus the outlier) from Exercise 4 above:

```
SnakeID <- 1:9
Ln <- c(60, 69, 66, 64, 54, 67, 59, 65, 63)
Wt <- c(136, 198, 194, 140, 93, 172, 116, 174, 145)
Snakes <- data.frame(SnakeID, Ln, Wt)
```

Fit the linear regression model:

```
my.reg <- lm(Wt ~ Ln, data = Snakes)
```

a) What class of object is returned by `lm()`? Find out by typing:

```
class(my.reg)
```

b) The `"lm"` class of objects is a special case of the `"list"` class. What is the result of the following?

```
is.list(my.reg)
```

c) How many objects are contained in the `my.reg` *list*? Find out by looking at their names:

```
names(my.reg)
```

d) Recall that the **equation** of the **fitted regression line** is

$$\hat{Y} = -301.09 + 7.19X$$

so the **nine** *fitted values* are defined as

$$\hat{Y}_i = -301.09 + 7.19X_i$$

(where the X_i's are the **lengths** of the **nine** snakes in the data set).

What would a plot of the **fitted values** versus the **lengths** ($\boldsymbol{X_i}$'s) look like? Try it:

```
library(dplyr)
```

```
Snakes <- mutate(Snakes, FittedVals = my.reg$fitted.values)
```

```
ggplot(data = Snakes, mapping = aes(x = Ln, y = FittedVals)) +
  geom_point() +
  ggtitle("Scatterplot of Fitted Values vs Lengths")
```

e) What would a plot of the **residuals** versus the **lengths** ($\boldsymbol{X}$) look like? Try it:

```
Snakes <- mutate(Snakes, Residuals = my.reg$residuals)
```

```
ggplot(data = Snakes, mapping = aes(x = Ln, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  ggtitle("Scatterplot of Residuals vs Lengths")
```

f) Show that the **residuals** sum to zero:

```
sum(Snakes$Residuals)
```

7.6.2 Measuring the Fit of a Simple Linear Regression Model: SSE, MSE, and R^2

- A model fits the data "well" if the **residuals** are **small**.
- One measure of how well the model fits is the ***residual sum of squares*** (also called ***error sum of squares***), denoted **SSE**:

$$\text{SSE} = \sum_{i=1}^{n} e_i^2 ,$$

where $\boldsymbol{e_i}$ is the ith **residual**, that is,

$$e_i = Y_i - \hat{Y}_i .$$

A **smaller** SSE indicates a **better fit**.

- SSE depends on the number of observations n, so it's better to measure the fit by the ***mean squared residual*** (also called ***mean squared error***), denoted **MSE** and defined as:

$$\text{MSE} = \frac{\text{SSE}}{n-2} .$$

- MSE is measured in the **squared** units of Y (e.g. if Y is measured in dollars, MSE is measured in dollars *squared*). So it's better to measure the fit by its square root, $\sqrt{\textbf{MSE}}$ which is called the ***root mean squared residual*** (or ***root mean squared error*** or ***residual standard error***).

For example, here (again) are the results of the regression of **arrival delay** on **scheduled departure hour** for flights to San Francisco:

```
my.reg <- lm(arr_delay ~ hour, data = SF)
summary(my.reg)

##
## Call:
## lm(formula = arr_delay ~ hour, data = SF)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -97.32 -25.22  -9.17   9.83 993.66
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -22.93267    1.23275  -18.60   <2e-16
## hour          2.01487    0.09154   22.01   <2e-16
##
## Residual standard error: 46.82 on 13171 degrees of freedom
## Multiple R-squared:  0.03548,Adjusted R-squared:  0.03541
## F-statistic: 484.5 on 1 and 13171 DF,  p-value: < 2.2e-16
```

From the output, the **root mean squared residual** (labeled `Residual standard error`) is $\sqrt{\text{MSE}} = \mathbf{46.82}$.

- The root mean squared residual depends on the units of Y (e.g. inches vs cm), so it's sometimes desirable instead to measure the fit by the $\boldsymbol{R^2}$, defined as:

$$R^2 = 1 - \frac{\text{SSE}}{(n-1)\text{Var}(Y_i's)} = 1 - \frac{\sum_{i=1}^n e_i^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2},$$

where Var(Y_i's) is the ***variance*** (*squared* **standard deviation**) of the Y_i's.

- $\boldsymbol{R^2}$ lies between **0** and **1**. A **larger** R^2 indicates a **better fit**.

For example, from the output of `summary()` (above), the $\boldsymbol{R^2}$ (labeled `Multiple R-squared`) is $\boldsymbol{R^2 = 0.0355}$, indicating a **poor fit**.

Section 7.6 Exercises

Exercise 8 This exercise uses the `flights` data (from the `"nycflights13"` package).

From Exercise 6, here are the plot and linear regression of **departure delay** on **scheduled departure hour** for flights to San Francisco:

```
ggplot(data = SF, mapping = aes(x = hour, y = dep_delay)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlab("Scheduled Hour of Departure") + ylab("Departure Delay (Minutes)") +
  coord_cartesian(ylim = c(-30, 200))
```

```
my.reg <- lm(dep_delay ~ hour, data = SF)
summary(my.reg)
```

a) From the output of `summary()`, what's the value of the **root mean squared resid-**

ual (labeled `Residual standard error`)?

b) From the output of `summary()`, what's the value of the $\boldsymbol{R^2}$ (labeled `Multiple R-squared`)?

7.6.3 Multiple Regression: Multiple Explanatory Variables

- ***Multiple regression models*** describe **variation** in a **response** variable $\boldsymbol{Y}$ as a function of ***several*** **explanatory variables** $\boldsymbol{X_1, X_2, \ldots, X_p}$.
- A ***multiple regression analysis*** involves obtaining the **equation** of the **plane** or **"hyperplane"** that best fits the data.

7.6.4 Multiple Regression with Two Explanatory Variables

- When $\boldsymbol{p = 2}$ (i.e. two explanatory variables), a multiple regression analysis involves obtaining the **equation** of the **plane** that best fits the data in a three-dimensional scatterplot.
- We carry out the analysis and view the results using `lm()` and `summary()`, as before.

For example, consider the following data set.

Data Set: `waterUsage`

The `waterUsage` data contain, for $\boldsymbol{n = 28}$ U.S. cities, the **four** variables:

`City`	Name of the city.
`Water`	The city's water consumption (log of millions of liters/day).
`Population`	The population of the city (in millions, year 2000)
`Wealth`	A measure of the city's wealth (z-score of the city's median income).

Water Usage for U.S. Metropolitan Areas

City	Water Usage (Y)	Wealth (X_1)	Population (X_2)
New York	9.2	2.8	21.3
Los Angeles	9.1	0.1	16.4
Chicago	8.4	-0.2	9.2
DC/Baltimore	8.1	1.8	6.5
San Francisco	8.0	1.9	6.3
⋮	⋮	⋮	⋮
Stockton	5.7	-0.9	0.6
Mobile	6.4	-1.5	0.5

```
waterUsage <- data.frame(City = c("New_York", "Los_Angeles", "Chicago", "DC_Baltimore",
                                  "San_Francisco", "Detroit_Ann_Arbor", "Dallas",
                                  "Atlanta", "Seattle", "Miami", "Phoenix",
                                  "Minneapolis", "Denver", "Pittsburgh", "St_Louis",
                                  "Portland_Salem", "San_Antonio", "Salt_Lake_City",
                                  "Las_Vegas", "Providence", "Jacksonville",
                                  "Dayton_Springfield", "Albany_Schenectady",
                                  "Albuquerque", "Omaha", "Little_Rock", "Stockton",
                                  "Mobile"),
                         Water = c(9.2, 9.1, 8.4, 8.1, 8.0, 7.9, 7.9, 7.6, 7.6, 7.6,
                                   7.6, 6.8, 7.2, 6.7, 6.7, 7.1, 6.7, 7.0, 7.1, 6.2,
                                   6.0, 6.1, 6.1, 6.3, 6.0, 5.7, 5.7, 6.41),
                         Wealth = c(2.8, 0.1, -0.2, 1.8, 1.9, 0.3, 0.4, 0.9, -0.3,
                                    -0.5, 0.1, 0.9, 0.7, -1.3, 0.0, -0.4, -1.3, -1.2,
                                    0.1, 0.0, -0.3, -0.3, 0.1, -0.5, -0.4, -0.9, -0.9,
                                    -1.5),
                         Population = c(21.3, 16.4, 9.2, 6.5, 6.3, 5.5, 5.2, 4.3, 3.6,
                                    3.9, 3.3, 3.0, 2.6, 2.5, 2.2, 2.3, 1.6, 1.3, 1.6,
                                    1.5, 1.1, 1.0, 0.9, 0.7, 0.7, 0.6, 0.6, 0.5))
```

We can obtain the ***fitted regression model***, with response variable **water consumption** ($\boldsymbol{Y}$) and *both* **wealth** ($\boldsymbol{X_1}$) *and* **population size** ($\boldsymbol{X_2}$) as explanatory variables, by typing:

```
my.reg <- lm(Water ~ Wealth + Population, data = waterUsage)
summary(my.reg)

##
## Call:
## lm(formula = Water ~ Wealth + Population, data = waterUsage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95881 -0.50504  0.06659  0.40889  0.58966
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.47674    0.14084  45.987  < 2e-16
## Wealth       0.11042    0.12603   0.876    0.389
## Population   0.15835    0.02633   6.014 2.78e-06
##
## Residual standard error: 0.5063 on 25 degrees of freedom
## Multiple R-squared:  0.7441,Adjusted R-squared:  0.7237
## F-statistic: 36.35 on 2 and 25 DF,  p-value: 3.985e-08
```

The resulting **fitted model** is:

$$\hat{Y} \;=\; 6.48 \,+\, 0.11\, X_1 \,+\, 0.16\, X_2\,.$$

This is the **equation** of the **plane** in a three-dimensional coordinate system, as shown below.

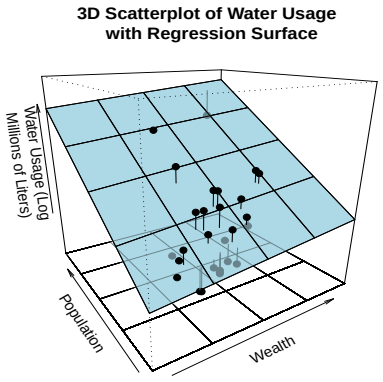


Figure 3

Based on the equation, we'd **predict** the **water usage** for a city whose **wealth** is **1.5** *and* whose **population** is **3.0** to be:

$$\hat{Y} \;=\; 6.48 \,+\, 0.11\,(1.5) \,+\, 0.16\,(3.0) \;=\; \mathbf{7.1}\,.$$

The **coefficient $\hat{\boldsymbol{\beta}}_{\mathbf{1}} = \mathbf{0.11}$** says that for each one-unit increase in a city's **wealth**, its water usage *increases* by **0.11** units when **population** is **fixed** (held **constant**).

- In general, an equation of the form

$$\hat{Y} \;=\; \hat{\beta}_0 \,+\, \hat{\beta}_1\, X_1 \,+\, \hat{\beta}_2\, X_2$$

describes a **plane** relating a response variable $\boldsymbol{Y}$ to *two* explanatory variables $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$:

1. The ***intercept*** $\hat{\boldsymbol{\beta}}_{\mathbf{0}}$ is the value of $\boldsymbol{Y}$ when $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are **both zero**.
2. The ***coefficient*** $\hat{\boldsymbol{\beta}}_{\mathbf{1}}$ quantifies the **change** in $\boldsymbol{Y}$ for each **one-unit change** in $\boldsymbol{X_1}$, **while $\boldsymbol{X_2}$ is fixed** (held **constant**).
3. The ***coefficient*** $\hat{\boldsymbol{\beta}}_{\mathbf{2}}$ quantifies the **change** in $\boldsymbol{Y}$ for each **one-unit change** in $\boldsymbol{X_2}$, **while $\boldsymbol{X_1}$ is fixed** (held **constant**).

Data Set: `portraitSales`

The `portraitSales` data set (below) was provided by Dwaine Studios, Inc., a portrait studio that specializes in portraits of children and operates in 21 cities. The data set includes information for each city about sales, number of persons under age 16, and per capita income.

`Sales`	Portrait sales (in thousands of dollars).
`Under16`	Number of persons under age 16 (in thousands of persons)
`Income`	Per capita disposable personal income (in thousands of dollars).

```
Sales <- c(174.4, 164.4, 244.2, 154.6, 181.6, 207.5, 152.8, 163.2,
           145.4, 137.2, 241.9, 191.1, 232.0, 145.3, 161.1, 209.7,
           146.4, 144.0, 232.6, 224.1, 166.5)

Under16 <- c(68.5, 45.2, 91.3, 47.8, 46.9, 66.1, 49.5, 52.0, 48.9,
             38.4, 87.9, 72.8, 88.4, 42.9, 52.5, 85.7, 41.3, 51.7,
             89.6, 82.7, 52.3)

Income <- c(16.7, 16.8, 18.2, 16.3, 17.3, 18.2, 15.9, 17.2, 16.6, 16.0,
            18.3, 17.1, 17.4, 15.8, 17.8, 18.4, 16.5, 16.3, 18.1, 19.1,
            16.0)

portraitSales <- data.frame(Sales, Under16, Income)
```

Section 7.6 Exercises

Exercise 9 Using the `portraitSales` data from above, we want to **predict** portrait sales from the number of persons under age 16 and per capita income.

a) Use `lm()` to fit the multiple regression model and `summary()` obtain the results. Write out the **equation** of the fitted plane.

b) What is the **predicted** sales for a city with **45.0** thousand people under 16 and per capita disposable income of **17.0** thousand dollars?

c) By how much does **sales** increase for each increase of **1.0** thousand people under 16 (holding income constant)?

d) By how much does **sales** increase for each increase of **1.0** thousand dollars in disposable income (holding the number of people under 16 constant)?

7.6.5 Multiple Regression with More than Two Explanatory Variables

- When $p > 2$ (i.e. more than two explanatory variables), a ***multiple regression model*** is no longer a plane to a 3D scatterplot, but it shares many features of a plane.
- As an example, we'll use the `waste` data described below.

Data Set: `waste`

The `waste` data below were used for the design of an efficient waste incinerator. On each of $n = 30$ waste specimens, **six variables** were recorded:

`Specimen`	ID number identifying the waste specimen (1-30)
`EnergyContent`	The energy content of the specimen (kcal/kg), a measure of burnability
`Plastics`	Plastics in the specimen (percent, by weight)
`Paper`	Paper in the specimen (percent, by weight)
`Garbage`	Garbage in the specimen (percent, by weight)
`Water`	Moisture in the specimen (percent, by weight)

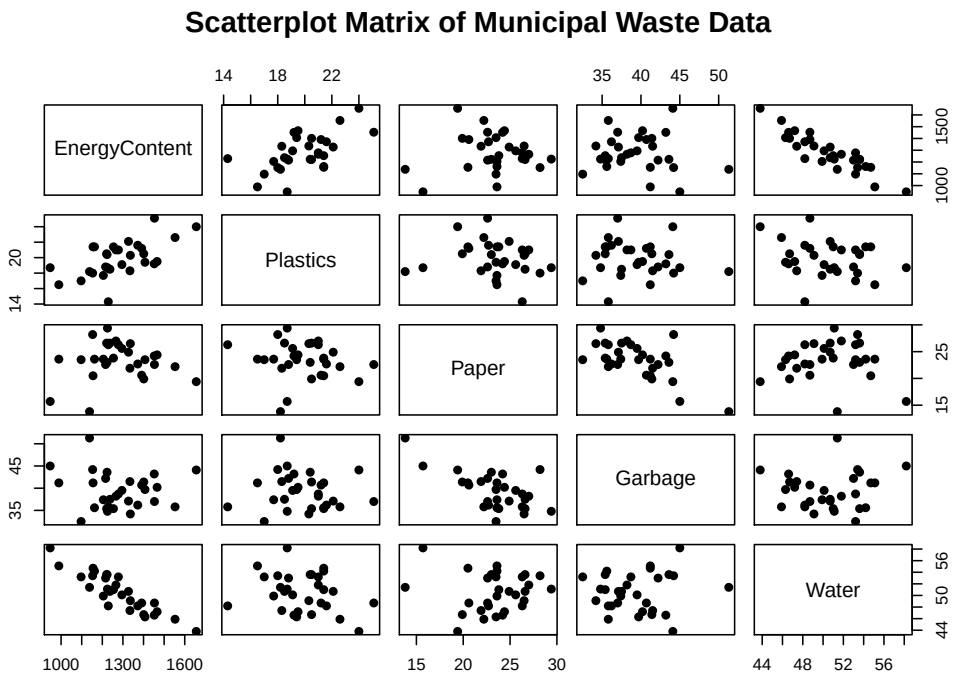
Municipal Waste Composition

Waste Specimen	Energy Content	Plastics	Paper	Garbage	Water
1	947	18.69	15.65	45.01	58.21
2	1407	19.43	23.51	39.69	46.31
3	1452	19.24	24.23	43.16	46.63
4	1553	22.64	22.20	35.76	45.85
5	989	16.54	23.56	41.20	55.14
⋮	⋮	⋮	⋮	⋮	⋮
29	1391	21.25	20.63	40.72	48.67
30	1372	21.62	22.71	36.22	48.19

```
waste <- data.frame(Specimen = 1:30,
                    EnergyContent = c(947, 1407, 1452, 1553, 989, 1162, 1466,
                        1656, 1254, 1336, 1097, 1266, 1401, 1223, 1216, 1334,
                        1155, 1453, 1278, 1153, 1225, 1237, 1327, 1229, 1205,
                        1221, 1138, 1295, 1391, 1372),
                    Plastics = c(18.7, 19.4, 19.2, 22.6, 16.5, 21.4, 19.5, 24.0,
                        21.4, 20.3, 17.0, 21.0, 20.5, 20.4, 18.8, 18.3, 21.4,
                        25.1, 21.0, 18.0, 18.7, 18.5, 22.1, 14.3, 17.7, 20.5,
                        18.2, 19.1, 21.2, 21.6),
                    Paper = c(15.7, 23.5, 24.2, 22.2, 23.6, 23.6, 24.4, 19.4,
                        23.8, 26.5, 23.5, 27.0, 19.9, 23.0, 22.6, 21.9, 20.5,
                        22.6, 26.3, 28.2, 29.4, 26.6, 24.9, 26.3, 23.6, 26.6,
                        13.8, 25.6, 20.6, 22.7),
                    Garbage = c(45.0, 39.7, 43.2, 35.8, 41.2, 35.6, 40.2, 44.1,
                        35.4, 34.2, 32.5, 38.2, 41.4, 43.6, 42.2, 41.5, 41.2,
                        37.0, 38.7, 44.2, 34.8, 37.5, 37.1, 35.8, 37.4, 35.4,
                        51.3, 39.5, 40.7, 36.2),
                    Water = c(58.2, 46.3, 46.6, 45.9, 55.1, 54.2, 47.2, 43.8,
                        51.0, 49.1, 53.2, 51.8, 46.7, 53.6, 53.0, 47.4, 54.7,
                        48.7, 53.2, 53.4, 51.1, 50.7, 50.7, 48.2, 49.9, 53.6,
                        51.4, 50.1, 48.7, 48.2))
```

Here's a ***scatterplot matrix*** of the data:

```
pairs(select(waste, -Specimen),
      main = "Scatterplot Matrix of Municipal Waste Data",
      pch = 19)
```



Here's the ***correlation matrix***:

```
cor_mat <- cor(select(waste, -Specimen))
round(cor_mat, 2)

##               EnergyContent Plastics Paper Garbage Water
## EnergyContent          1.00     0.59  0.04   -0.09 -0.90
## Plastics               0.59     1.00 -0.15   -0.09 -0.26
## Paper                  0.04    -0.15  1.00   -0.63  0.00
## Garbage               -0.09    -0.09 -0.63    1.00  0.07
## Water                 -0.90    -0.26  0.00    0.07  1.00
```

The **correlation matrix** is just a table showing, in rows and columns, the ***correlations*** corresponding to the plots in the **scatterplot matrix**.

We'll develop a **model** for ***predicting*** **energy content** from the $\boldsymbol{p = 4}$ other variables.

We obtain the ***fitted regression model***, with response variable **energy content** ($\boldsymbol{Y}$) and explanatory variables **Plastics** ($\boldsymbol{X_1}$), **Paper** ($\boldsymbol{X_2}$), **Garbage** ($\boldsymbol{X_3}$), and **Water** ($\boldsymbol{X_4}$), by typing:

```
my.reg <- lm(EnergyContent ~ Plastics + Paper + Garbage + Water, data = waste)
summary(my.reg)

##
## Call:
## lm(formula = EnergyContent ~ Plastics + Paper + Garbage + Water,
##     data = waste)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39.47 -24.05 -11.72  21.45  60.56
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2234.508    178.576  12.513 2.91e-12
## Plastics      29.090      2.836  10.256 1.92e-10
## Paper          7.738      2.329   3.323  0.00274
## Garbage        4.353      1.926   2.260  0.03282
## Water        -37.291      1.840 -20.270  < 2e-16
##
## Residual standard error: 31.58 on 25 degrees of freedom
## Multiple R-squared:  0.9639,Adjusted R-squared:  0.9581
## F-statistic: 166.6 on 4 and 25 DF,  p-value: < 2.2e-16
```

The ***fitted multiple regression model*** is

$$\hat{Y} \;=\; 2234.5 \,+\, 29.1\, X_1 \,+\, 7.7\, X_2 \,+\, 4.4\, X_3 \,-\, 37.3\, X_4.$$

- In general, the **coefficients** of a ***fitted regression model*** of the form

$$\hat{Y} \;=\; \hat{\beta}_0 \,+\, \hat{\beta}_1\, X_1 \,+\, \hat{\beta}_2\, X_2 \cdots +\, \hat{\beta}_p\, X_p$$

have the same interpretation as they did for the *plane* when $p = 2$:

1. The ***intercept*** $\boldsymbol{\hat{\beta}_0}$ is the value of $\boldsymbol{Y}$ when $\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{\ldots}, \boldsymbol{X_p}$ are **all zero**.
2. For each $k = 1, 2, \ldots, p$, the ***coefficient*** $\boldsymbol{\hat{\beta}_k}$ quantifies the **change** in $\boldsymbol{Y}$ for each **one-unit change** in $\boldsymbol{X_k}$, ***while the other*** $\boldsymbol{p-1}$ $\boldsymbol{X_i}$***'s*** are all **fixed** (held **constant**).

Data Set: `cdi`

The `cdi` data set (**CDI.txt** on the course website) provides selective county demographic information (CDI) for 440 of the most populous counties in the U.S. Each line of the data set has an identification number with a county name and state abbreviation and provides information on 14 variables for a single county. Counties with missing data were deleted from the data set. The information generally pertains to the years 1990 and 1992. The **17 variables** are:

`ID`	Identification number (1-440)
`County`	County name
`State`	Two-letter state abbreviation
`LandArea`	Land area (square miles)
`TotPop`	Estimated 1990 population
`PctPop18_34`	Percent of 1990 CDI population aged 18-34
`PctPop65`	Percent of 1990 CDI population aged 65 or older
`nActPhys`	Number of professionally active nonfederal physicians during 1990
`nHospBeds`	Total number of beds, cribs, and bassinets during 1990
`nCrimes`	Total number of serious crimes in 1990, including murder, rape, robbery, aggravated assault, burglary, larceny-theft, and motor vehicle theft, as reported by law enforcement agencies
`PctHSGrad`	Percent of adult population (aged 25 or older) who completed 12 or more years of school
`PctBach`	Percent of adult population (aged 25 or older) with a bachelor's degree
`PctBelPov`	Percent of 1990 CDI population with income below poverty level
`PctUnemp`	Percent of 1990 CDI labor force that was unemployed
`PerCapInc`	Per capita income of 1990 CDI population (dollars)
`TotInc`	Total personal income of 1990 CDI population (in millions of dollars)
`Region`	Geographic region classification that is used by the U.S. Bureau of the Census, where 1=NE, 2=NC, 3=S, 4=W

Section 7.6 Exercises

Exercise 10 Using the `cdi` data set (**CDI.txt** on the course website), we want to **predict** the **number of active physicians** ($\boldsymbol{Y}$) from the **total population** ($\boldsymbol{X_1}$), **land area** ($\boldsymbol{X_2}$), and **total personal income** ($\boldsymbol{X_3}$).

a) Use `pairs()` to make a **scatterplot matrix** of these variables.

b) Use `cor()` to make the **correlation matrix** of the data.

c) Use `lm()` to carry out a **multiple regression analysis** and `summary()` obtain the results. Write out the **equation** of the **fitted model**.

d) What is the **predicted** number of active physicians for a county with a total population of **400,000** people, a land area of **1,000** square miles, and total personal income of **8,000** million dollars?

e) By how much does **number of active physicians** increase for each increase in total population of **1** person (holding land area and total personal income constant)?

f) By how much does **number of active physicians** increase for each increase in total personal income of **1.0** million dollars (holding land area and total population constant)?

Exercise 11 Now suppose we want to **predict** the **number of active physicians** ($\boldsymbol{Y}$) from the **population density** ($\boldsymbol{X_1}$, total population divided by $\boldsymbol{X_2}$, land area), **percent**

of population 65 or older ($\boldsymbol{X_2}$), and **per capita income** ($\boldsymbol{X_3}$) (using the `cdi` data set again).

a) Use `mutate()` (from the `"dplyr"` package) to create a new variable in the `cdi` data frame `PopDens` containing the **population density** of each county. Report your R command(s).

b) Use `lm()` to carry out the **multiple regression analysis** and `summary()` obtain the results. Write out the **equation** of the **fitted model**.

c) What is the **predicted** number of active physicians for a county with a population density of **900** people per square mile, **15** percent of its population over 65, and total personal income of **20,000** dollars?

- The **deviations** of the observations away from the fitted model are called ***residuals***.

  Consider (again) the `waterUsage` data frame (from above).

```
head(waterUsage)
```

```
##                City Water Wealth Population
## 1          New_York   9.2    2.8       21.3
## 2       Los_Angeles   9.1    0.1       16.4
## 3           Chicago   8.4   -0.2        9.2
## 4      DC_Baltimore   8.1    1.8        6.5
## 5     San_Francisco   8.0    1.9        6.3
## 6 Detroit_Ann_Arbor   7.9    0.3        5.5
```

  After fitting the **regression model** to the data:

```
my.reg <- lm(Water ~ Wealth + Population, data = waterUsage)
```

  the **residuals** are the line segments shown in Fig. 3.

  We can obtain the **residuals** via `my.reg$residuals`:

```
waterUsage <- mutate(waterUsage, Residuals = my.reg$residuals)
```

- The ***fitted values*** are defined as

$$\hat{Y}_i \;=\; \hat{\beta}_0 \,+\, \hat{\beta}_1\, X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_p X_{pi}\,,$$

  where the $X_{1i}, X_{2i}, \ldots, X_{pi}$ are the observed values of the predictors (used to fit the model). In Fig. 3, the **fitted values** are points **on** the **fitted plane** where the line segments meet it.

  We can obtain the **fitted values** via `my.reg$fitted.values`:

```
waterUsage <- mutate(waterUsage, FittedVals = my.reg$fitted.values)
```

7.6.6 Measuring the Fit of a Multiple Regression Model: SSE, MSE, and R^2

- A model fits the data "well" if the **residuals** are **small**.
- One measure of how well the model fits is the ***residual sum of squares*** (also called ***error sum of squares***), denoted **SSE**:

$$\text{SSE} = \sum_{i=1}^{n} e_i^2 ,$$

  where $\boldsymbol{e_i}$ is the ith **residual**, that is,

$$e_i = Y_i - \hat{Y}_i .$$

- SSE depends on the number of observations n, so it's better to measure the fit by the ***mean squared residual*** (also called ***mean squared error***), denoted **MSE** and defined as:

$$\text{MSE} = \frac{\text{SSE}}{n-p-1} .$$

- MSE is measured in the **squared** units of Y (e.g. dollars *squared*). So it's better to measure the fit by its square root, $\sqrt{\textbf{MSE}}$ which is called the ***root mean squared residual*** (or ***root mean squared error*** or ***residual standard error***).

  For example, from the output of `summary()` for the `waterUsage` regression (above), $\boldsymbol{\sqrt{\text{MSE}} = 0.5063}$ (labeled `Residual Standard Error`).

- The root mean squared residual depends on the units of Y (e.g. inches vs cm), so it's sometimes desirable instead to measure the fit by the $\boldsymbol{R^2}$, defined as:

$$R^2 = 1 - \frac{\text{SSE}}{(n-1)\text{Var}(Y_i's)} = 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2},$$

  where Var(Y_i's) is the ***variance*** (*squared* **standard deviation**) of the Y_i's.

- $\boldsymbol{R^2}$ lies between **0** and **1**. A **larger** R^2 indicates a **better fit**.

  For example, from the output of `summary()` for the `waterUsage` regression (above), the $\boldsymbol{R^2}$ (labeled `Multiple R-squared`) is $\boldsymbol{R^2 = 0.7441}$, indicating **decent fit**.

Section 7.6 Exercises

Exercise 12 In this exercise, you'll compare the fits of the two models from Exercises 10 and 11.

a) Using the **root mean squared residual** (labeled `Residual standard error` in

the output from `summary()`), which model fits the data better? **Hint**: A *smaller* root mean squared residual indicates a *better* fitting model.

b) Using the $\boldsymbol{R^2}$ (labeled `Multiple R-squared` in the output from `summary()`), which model fits the data better? **Hint**: A *larger* R^2 (*closer to 1*) indicates a *better* fitting model.

c) Based on your answers to Parts *a* and *b*, which model would you expect to give **better predictions** of the number of active physicians in a county?

7.7 Logistic Regression: Dichotomous (0 or 1) Response Variable (E.5)

- ***Logistic regression*** is used when the response variable Y is ***dichotomous***, that is, only takes **two** values (e.g. Yes/No, Healthy/Unhealthy, etc.). which we code as **0** and **1**.

  For a **dichotomous** response variable, we (usually) want to estimate the **probability** that $\boldsymbol{Y}$ will equal **one** as a **function** of an explanatory variable $\boldsymbol{X}$.

Data Set: `dues`

The `dues` data set (**DUES.txt** on the course website) contains responses to a survey of **30** members conducted by the board of directors of a professional association to assess the effects of several possible amounts of dues increase. The **two variables** are:

`DuesIncr`	The amount of dues increase
`NotRenew`	Whether the interviewee would not renew their membership at that amount of dues increase (1 if they would not renew, 0 if they would renew)

- For example, consider the `dues` data set:

```
head(dues)
```

```
##   NotRenew DuesIncr
## 1        0       25
## 2        0       27
## 3        0       30
## 4        0       30
## 5        0       31
## 6        0       32
```

We might want to model the **probability** of a person **not renewing** their membership as a **function** of the dues increase $\boldsymbol{X}$.

A *linear regression model* is **not appropriate** (the line extends above and below the range of probability values 0 to 1 (see Fig. 4):

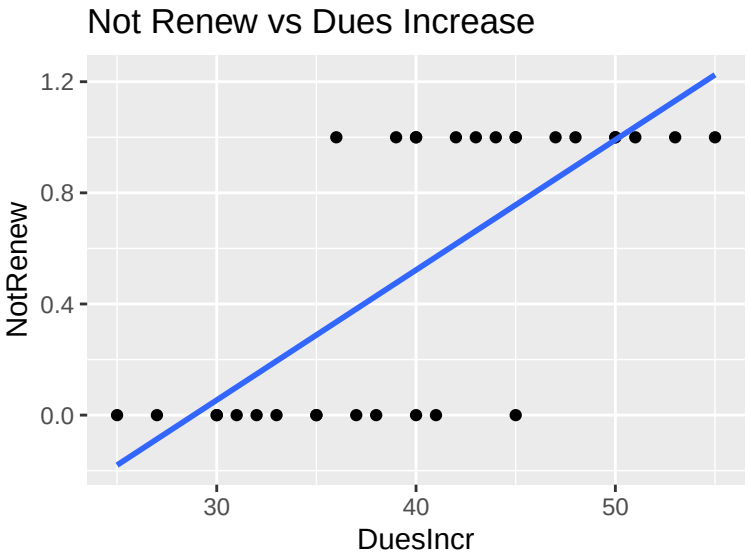


Figure 4

```
ggplot(data = dues, mapping = aes(x = DuesIncr, y = NotRenew)) +
  geom_point() +
  labs(title = "Not Renew vs Dues Increase") +
  geom_smooth(method = "lm", se = FALSE)
```

Here's a (more appropriate) **fitted *logistic regression model*** (Fig. 5):

```
ggplot(data = dues, mapping = aes(x = DuesIncr, y = NotRenew)) +
  geom_point() +
  labs(title = "Not Renew vs Dues Increase") +
  geom_smooth(method = "glm",
    method.args = list(family = "binomial"),
    se = FALSE)
```

The curve in Fig. 5 gives the (estimated) **probability** of **not renewing** for any given value of the **dues increase** $\boldsymbol{X}$.

- We fit a **logistic regression model** to data (and view the results) using the following functions.

```
glm()          # Fit logistic regression model by specifying
               # family = "binomial" (other so-called generalized
               # linear models use a different family).
summary()      # Look at a summary of the fitted model.
```

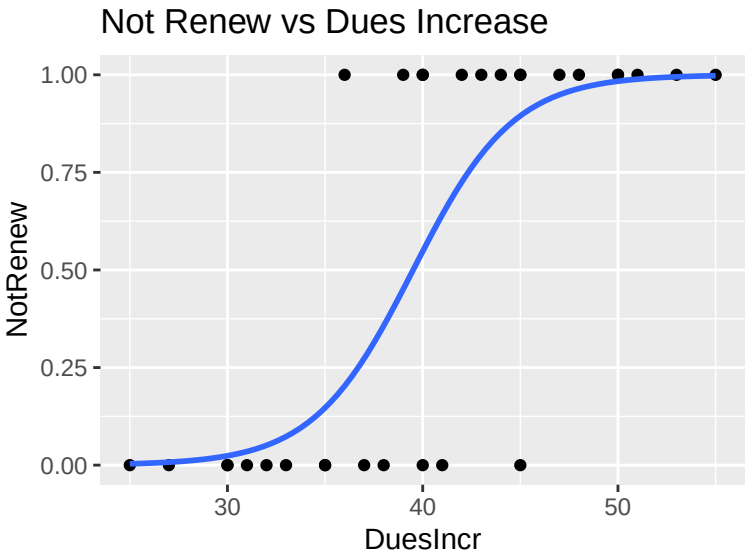


Figure 5

- For example, using the `dues` data set, with **not renewing** as the response (Y) and **dues increase** as the explanatory variable (X):

```
my.logreg <- glm(NotRenew ~ DuesIncr, data = dues, family = "binomial")
```

```
summary(my.logreg)

##
## Call:
## glm(formula = NotRenew ~ DuesIncr, family = "binomial", data = dues)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.12290  -0.37290   0.08522   0.47113   1.78651
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.4157     5.6780  -2.715  0.00663
## DuesIncr      0.3902     0.1421   2.745  0.00605
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.455  on 29  degrees of freedom
## Residual deviance: 20.083  on 28  degrees of freedom
## AIC: 24.083
##
## Number of Fisher Scoring iterations: 6
```

(We'll interpret some of the output later.)

- To understand the **fitted logistic regression model**, we'll define a function $\boldsymbol{p(X)}$ as

$$p(X) \;=\; P(Y = 1 \text{ when the value of the explanatory variable is } X)$$

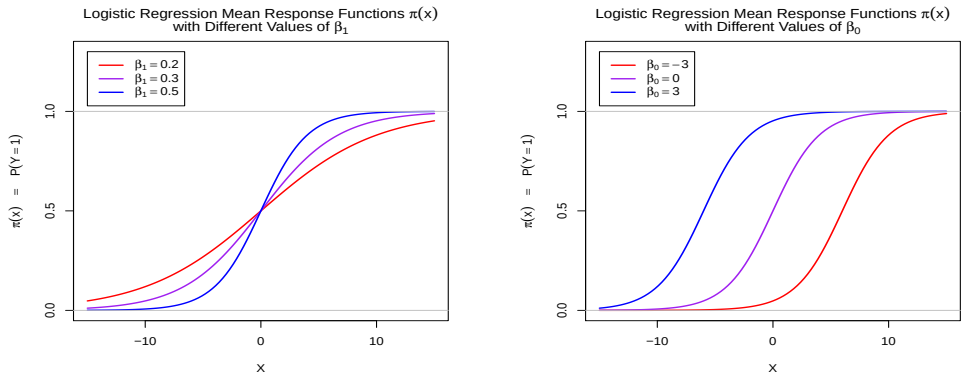
The ***fitted logistic regression model*** has the form

$$p(X) \;=\; \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} \tag{1}$$

(were $\boldsymbol{e}$ is the *exponential constant*, $e = 2.718282...$).

The **fitted logistic regression model** has the following properties:

1. $P(X)$ is constrained to lie between zero and one.
2. If $\hat{\beta}_1 > 0$, then $P(X)$ is an increasing function of X. Also, $p(X) \to 1$ as X increases and $p(X) \to 0$ as X decreases.
3. If $\hat{\beta}_1 < 0$, $P(X)$ is a decreasing function of X. Also, $p(X) \to 0$ as X increases and $p(X) \to 1$ as X decreases.
4. $\hat{\beta}_1$ determines the "steepness" of the "middle" part of the graph of $p(X)$. A larger $\hat{\beta}_1$ results in a "steeper" graph.
5. $\hat{\beta}_0$ shifts the graph left or right.



- Refer to the output from `summary(my.logreg)` above. For the `dues` data, the ***coefficients*** in the **fitted logistic regression model** are:

$$\hat{\beta}_0 \;=\; -15.42 \qquad \text{and} \qquad \hat{\beta}_1 \;=\; 0.39.$$

So the **fitted model** is

$$p(X) \;=\; \frac{e^{-15.42+0.39X}}{1+e^{-15.42+0.39X}}\,.$$

This is the curve graphed in Fig. 5.

If we plug any value in for X, we get the (estimated) **probability** that a person **won't renew** at that **dues increase** value, e.g.

$$p(42) = \frac{e^{-15.42+0.39\times 42}}{1+e^{-15.42+0.39\times 42}} = 0.72.$$

We could also get this value using `predict()`:

```
newDues <- data.frame(DuesIncr = 42)
predict(my.logreg, newDues, type = "response")

##         1
## 0.7254854
```

- Note that another (equivalent) way of writing the model (1) is

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \hat{\beta}_0 + \hat{\beta}_1 X\,.$$

Section 7.7 Exercises

Exercise 13 Fit the **logistic regression model** to the `dues` data set (**DUES.txt** on the course website), with `NotRenew` as the response variable (Y) and `DuesIncr` as the explanatory variable (X), using `glm()` with `family = "binomial"`:

```
my.logreg <- glm(NotRenew ~ DuesIncr, family = "binomial", data = dues)
summary(my.logreg)
```

Now use `predict()` to answer the following questions.

a) What's the (estimated) probability that a person **won't renew** their membership if the **dues increase** is **45** dollars?

b) What's the (estimated) probability if the **dues increase** is only **35** dollars?