Lab11

Jonathan Stiefel

10/27/2020

Learning Objectives

- Describe how to merge two dataframes in R with a common variable.
- Interpret correlation given bivariate data
- Create scatter plots to visualize the relationship between bivariate data variables–x,y.
- Use linear regression in R to compute the slope and intercept given a bivariate data sample.
- Explain the meaning of the R^2 value and describe how it related to the residuals for a linear regression model.

Some data wrangline in R Merging Two Data Frames

Two data sets, "Batting.csv" and "Master.csv", are taken from a website called Sean Lahman's Baseball Database where this individual (and his friends, I assume) have compiled complete batting and pitching statistics for U.S. Major League Baseball from 1871 to 2018.

Let's import the batting data

```
batting<-read.csv("Batting.csv")
master<-read.csv("Master.csv")
```

If you look at the codebook/data dictionary for these files (which is posted on Moodle) or examine the contents of these files with the `str` function:

```
str(batting)
```

```
## 'data.frame':    97889 obs. of  24 variables:
##  $ playerID : chr  "aardsda01" "aardsda01" "aardsda01" "aardsda01" ...
##  $ yearID   : int  2004 2006 2007 2008 2009 2010 2012 1954 1955 1956 ...
##  $ stint    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ teamID   : chr  "SFN" "CHN" "CHA" "BOS" ...
##  $ lgID     : chr  "NL" "NL" "AL" "AL" ...
##  $ G        : int  11 45 25 47 73 53 1 122 153 153 ...
##  $ G_batting: int  11 43 2 5 3 4 NA 122 153 153 ...
##  $ AB       : int  0 2 0 1 0 0 NA 468 602 609 ...
##  $ R        : int  0 0 0 0 0 0 NA 58 105 106 ...
##  $ H        : int  0 0 0 0 0 0 NA 131 189 200 ...
##  $ X2B      : int  0 0 0 0 0 0 NA 27 37 34 ...
##  $ X3B      : int  0 0 0 0 0 0 NA 6 9 14 ...
##  $ HR       : int  0 0 0 0 0 0 NA 13 27 26 ...
##  $ RBI      : int  0 0 0 0 0 0 NA 69 106 92 ...
##  $ SB       : int  0 0 0 0 0 0 NA 2 3 2 ...
##  $ CS       : int  0 0 0 0 0 0 NA 2 1 4 ...
##  $ BB       : int  0 0 0 0 0 0 NA 28 49 37 ...
##  $ SO       : int  0 0 0 1 0 0 NA 39 61 54 ...
##  $ IBB      : int  0 0 0 0 0 0 NA NA 5 6 ...
```

```
##  $ HBP      : int  0 0 0 0 0 0 NA 3 3 2 ...
##  $ SH       : int  0 1 0 0 0 0 NA 6 7 5 ...
##  $ SF       : int  0 0 0 0 0 0 NA 4 4 7 ...
##  $ GIDP     : int  0 0 0 0 0 0 NA 13 20 21 ...
##  $ G_old    : int  11 45 2 5 NA NA NA 122 153 153 ...
```

```
str(master)
```

```
## 'data.frame':    18354 obs. of  24 variables:
##  $ playerID    : chr  "aardsda01" "aaronha01" "aaronto01" "aasedo01" ...
##  $ birthYear   : int  1981 1934 1939 1954 1972 1985 1854 1877 1869 1866 ...
##  $ birthMonth  : int  12 2 8 9 8 12 11 4 11 10 ...
##  $ birthDay    : int  27 5 5 8 25 17 4 15 11 14 ...
##  $ birthCountry: chr  "USA" "USA" "USA" "USA" ...
##  $ birthState  : chr  "CO" "AL" "AL" "CA" ...
##  $ birthCity   : chr  "Denver" "Mobile" "Mobile" "Orange" ...
##  $ deathYear   : int  NA NA 1984 NA NA NA 1905 1957 1962 1926 ...
##  $ deathMonth  : int  NA NA 8 NA NA NA 5 1 6 4 ...
##  $ deathDay    : int  NA NA 16 NA NA NA 17 6 11 27 ...
##  $ deathCountry: chr  "" "" "USA" "" ...
##  $ deathState  : chr  "" "" "GA" "" ...
##  $ deathCity   : chr  "" "" "Atlanta" "" ...
##  $ nameFirst   : chr  "David" "Hank" "Tommie" "Don" ...
##  $ nameLast    : chr  "Aardsma" "Aaron" "Aaron" "Aase" ...
##  $ nameGiven   : chr  "David Allan" "Henry Louis" "Tommie Lee" "Donald William" ...
##  $ weight      : int  205 180 190 190 184 220 192 170 175 169 ...
##  $ height      : int  75 72 75 75 73 73 72 71 71 68 ...
##  $ bats        : chr  "R" "R" "R" "R" ...
##  $ throws      : chr  "R" "R" "R" "R" ...
##  $ debut       : chr  "2004-04-06" "1954-04-13" "1962-04-10" "1977-07-26" ...
##  $ finalGame   : chr  "2013-09-28" "1976-10-03" "1971-09-26" "1990-10-03" ...
##  $ retroID     : chr  "aardd001" "aaroh101" "aarot101" "aased001" ...
##  $ bbrefID     : chr  "aardsda01" "aaronha01" "aaronto01" "aasedo01" ...
```

you see that the "Batting" data set contains detailed records on players' batting records for every year they played. (In other words, each row corresponds to 1 player in 1 particular year.) The "Master" file contains details on the players themselves such as full name, place of birth, handedness, height/weight. (In other words, each row corresponds to 1 player.) In both files there is a variable called "playerID" that uniquely identifies the player and can be used to "link" information from the two files together.

Imagine that I am interested in data from a particular year (2012), and I want to know if left-handed batters were more likely to get a higher number of hits than right-handed batters. The "Batters" file will tell me how many hits a player got in a particular year, but it doesn't contain the information on whether they are left-handed or right-handed!To get that, I need the other "Master" file. How am I going to put these together? There are many ways to merge two data frames or tables (and frankly, R isn't the software that's best-suited for this, but it can be done fairly simply for examples like this). First, we will subset the "batting" file so that we have only the records from the year 2012.

```
batting_2012 <- batting[which(batting$yearID==2012), ]
```

You have seen this kind of statement before. Note the extra "comma" before the last bracket – is needed to create a whole new data frame without getting errors.

Next, we want to "merge" the two files, using the linking variable "playerID". For every row of the "Batting" file, we want to add extra columns that contain all of the information from the "Master" file that correspond to that particular player. We do that with this syntax:

```
batting_final<-merge(batting_2012, master, by="playerID")
```

The merge( ) function has 3 arguments: the first two are the names of the data frames that you are merging, and the last argument by=" " gives R the name of the linking variable that is shared between them. This is a very simple example, and you should be aware that there is a whole programming language (SQL) and many, many books that are dedicated solely to better ways to merge multiple data files together, but this merge( ) command will be good enough for our purposes.

Now if we examine the new data frame we created (batting_final) we see that it contains information on 2012 batting records and the basic information on the players, too. To see the variable names or column headers in the merged dataframe: `batting_final`, use the `names()` command.

```
names(batting_final)
```

```
##  [1] "playerID"     "yearID"       "stint"        "teamID"       "lgID"
##  [6] "G"            "G_batting"    "AB"           "R"            "H"
## [11] "X2B"          "X3B"          "HR"           "RBI"          "SB"
## [16] "CS"           "BB"           "SO"           "IBB"          "HBP"
## [21] "SH"           "SF"           "GIDP"         "G_old"        "birthYear"
## [26] "birthMonth"   "birthDay"     "birthCountry" "birthState"   "birthCity"
## [31] "deathYear"    "deathMonth"   "deathDay"     "deathCountry" "deathState"
## [36] "deathCity"    "nameFirst"    "nameLast"     "nameGiven"    "weight"
## [41] "height"       "bats"         "throws"       "debut"        "finalGame"
## [46] "retroID"      "bbrefID"
```

Hits vs. Batting Hand

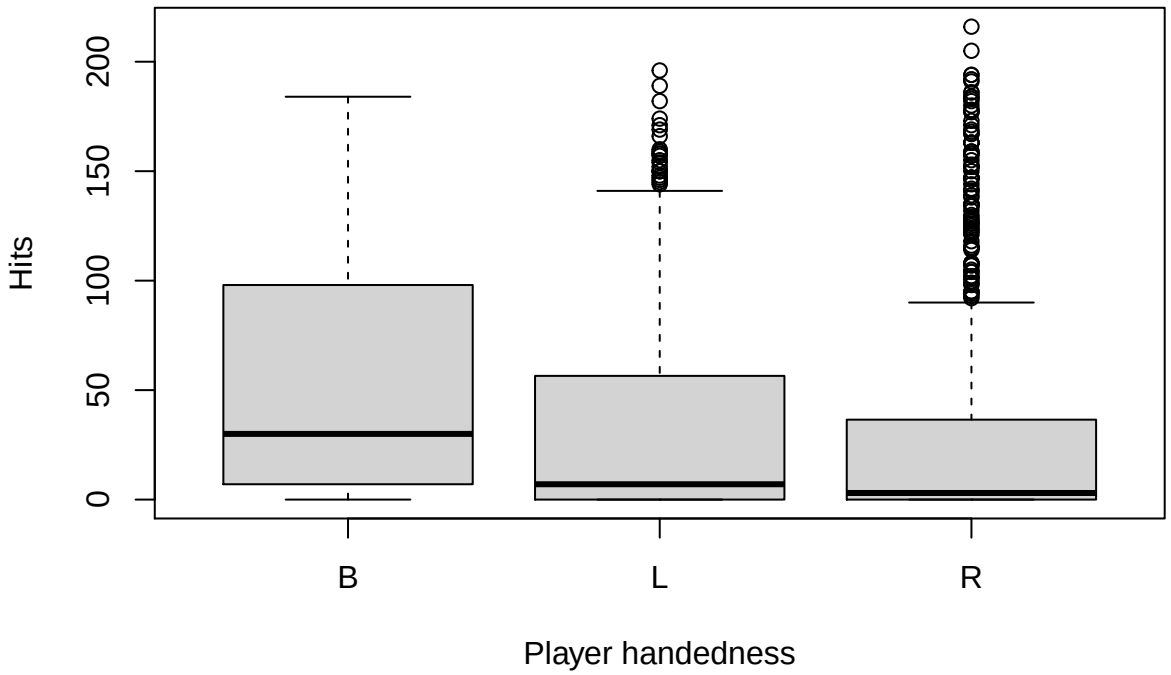
If we want to make a quick table to see how many players bat with each hand ("bats"), we could say:

```
table(batting_final$bats)
```

```
##
##   B   L   R
## 111 432 863
```

You can see there were 863 right-handed players, 432 left-handed players, and 111 "both" ("switch hitters"). If we want to make a quick boxplot comparing the number of hits ("H") per season for players who bat with different hands, we can write:

```
boxplot(batting_final$H ~ batting_final$bats, ylab="Hits", xlab = "Player handedness")
```

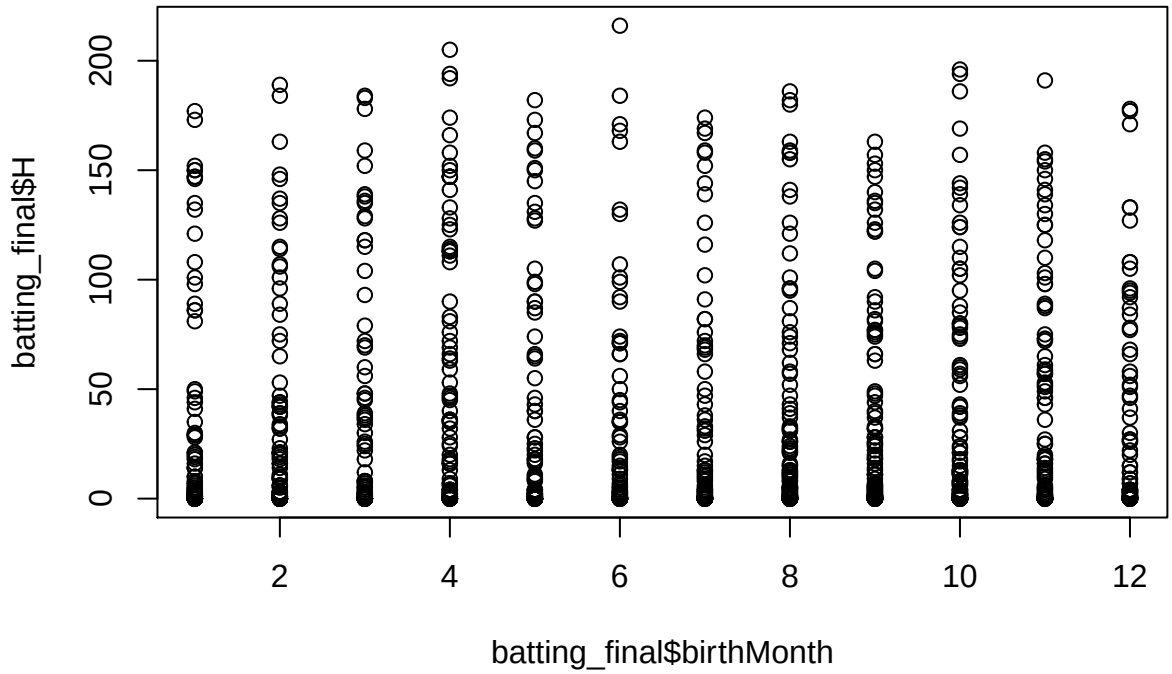


Note that the syntax used for creating this boxplot is different than what you've used before. The column `batting_final$bats` indicates whether a player bats right, left, or both. And the column `batting_final$H` shows the number of hits per player.

This boxplot shows that the median number of hits was higher for left-handers, as was the 75th percentile. In neither group does the number of hits look normally distributed (they're very skewed), so a t-test would not be the right thing to do here.

In the videos this week we introduced scatter plots. To create a scatter plot of hits versus birth month, I would use the following code

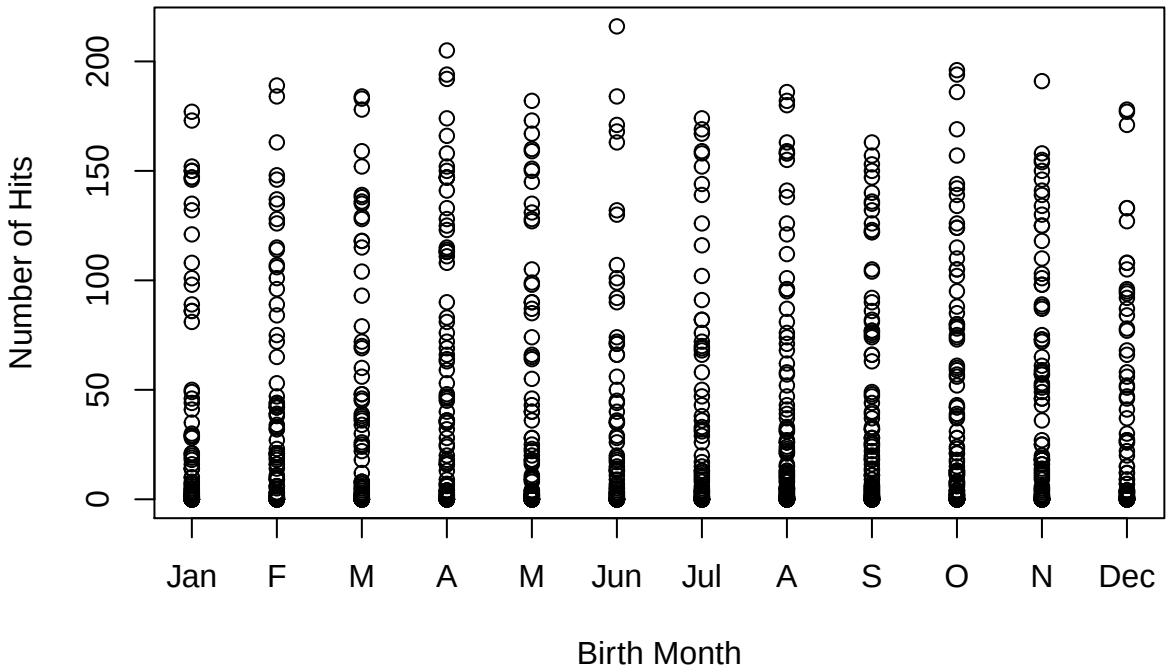
```
plot(batting_final$H ~ batting_final$birthMonth)
```



```
#Notice the syntax here is y~x instead of x,y.  Both work.
```

To customize this plot, I would use the following code:

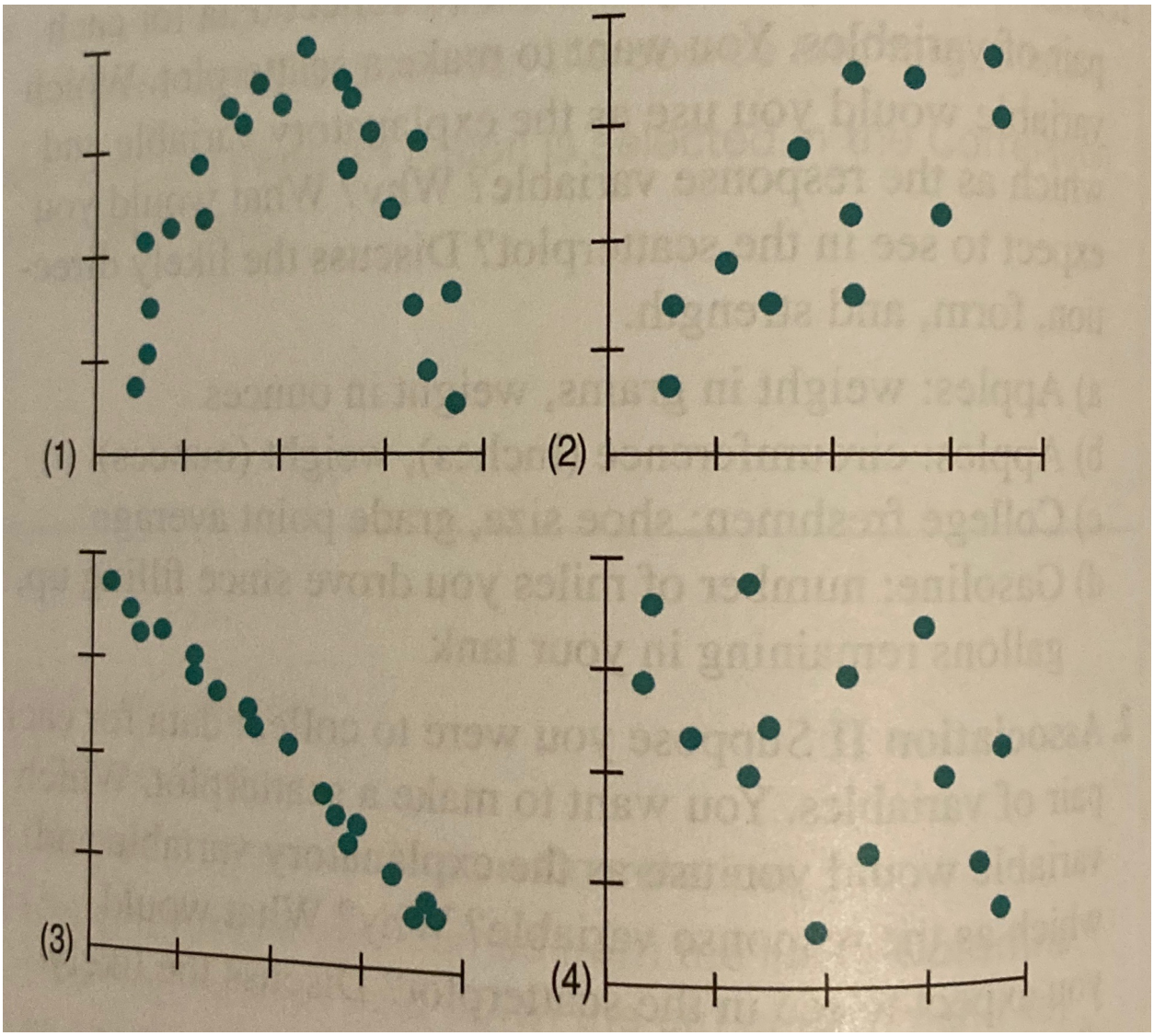
```
plot(batting_final$H ~ batting_final$birthMonth,  xlab = "Birth Month", ylab = "Number of Hits", xaxt =
axis(1, at = 1:12, labels=c("Jan", "F", "M", "A", "M", "Jun", "Jul", "A", "S", "O", "N", "Dec"))
```



As expected we don't see a relationship between birth month and hits, but you will use this data in Problem 4.

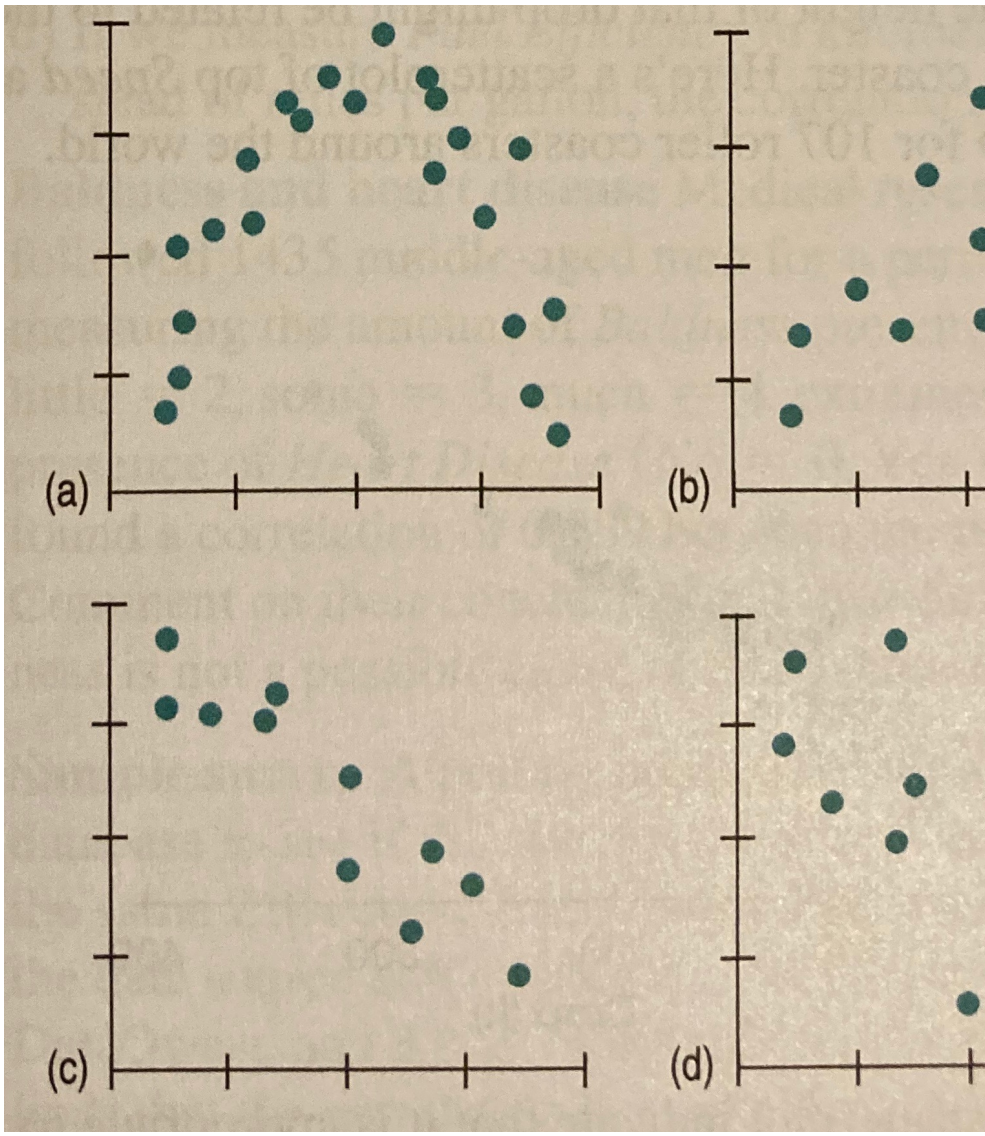
Problems Problem 1

Which of the following scatter plots (numbered 1,2,3,4) show: a) little or no association 4 b) a negative association 3 c) a linear association 3 d) a moderately strong association? 2 e)a very strong association? 3



Problem 2

Here are four scatter plots. The calculated correlations are -0.923, -0.487, 0.006, 0.777. Which is which?



a = 0.006 b = -0.487 c = 0.777 d = -0.923

Problem 3 Baseball

(a) Each row (record) of this baseball data set represents one player's batting statistics in a particular year. The variable "AB" tells how many "at-bats" the player had in that particular year; i.e., how many times they had an opportunity to bat. Make a smaller dataset (a subset, or filtered dataset) that contains only the records from the original file where the player had at least 100 at-bats (i.e., the value of AB is 100 or greater). [Hint: this should result in a new data frame that has 37,085 rows. You can use the `nrow()` command to check.

```
B100 <- batting[which(batting$AB>=100),]
nrow(B100)
```

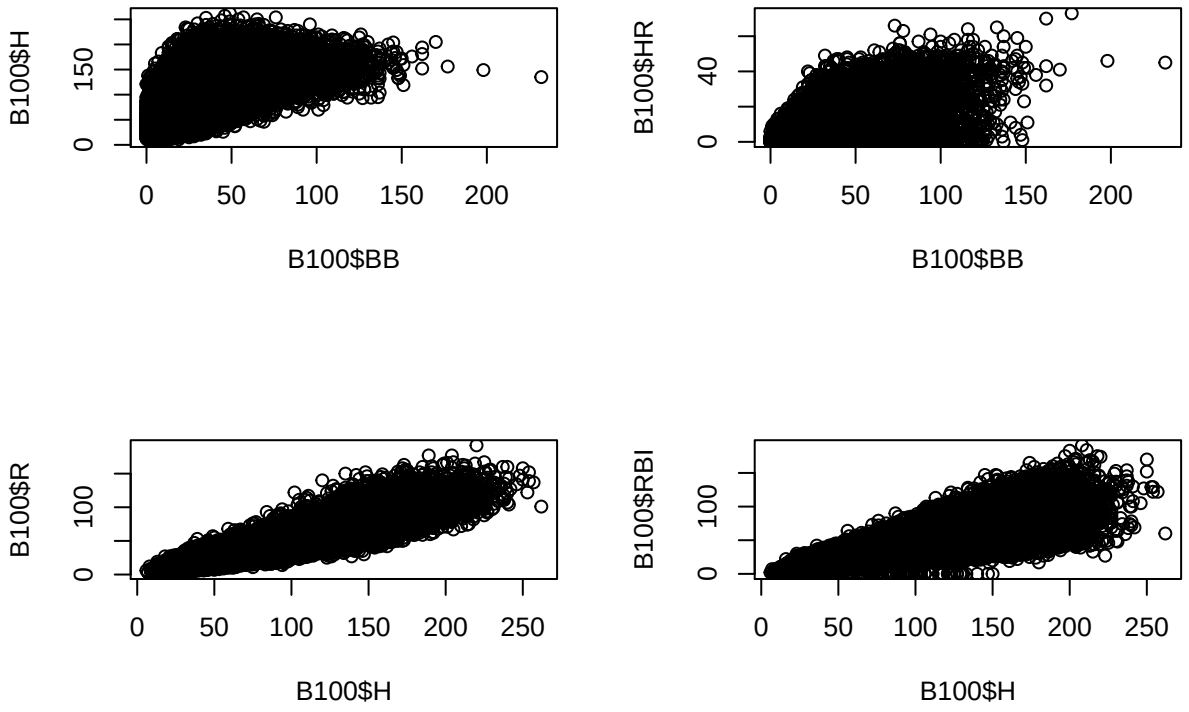
```
## [1] 37085
```

(b) The variables "H", "BB", "R", RBI" and "HR" represent the number of "hits", "base on balls" (also known as "walks"), "runs," "runs batted in," and "home runs," respectively. Use the `par(mfrow = c(2,2))` and `plot( )` commands to create a panel of 4 scatter plots for the following variables: H

vs. BB, HR vs. BB, R vs. H, and RBI vs. H. Use descriptive labels for all x and y axes. You do not need to include plots titles. Comment every line of code.

```
par(mfrow = c(2,2))

plot(B100$H~B100$BB)
plot(B100$HR~B100$BB)
plot(B100$R~B100$H)
plot(B100$RBI~B100$H)
```



(c) Explain why "R vs. H" is the most appropriate among these four data sets for a linear regression model. Think about the shape of the residuals around a regression line for each of these data sets.

This linear regression model has the strongest correlation and the lowest residuals, so a linear regression model would most accurately model the R vs H plot.

(d) Use the the `cor(x, y)` command to calculate the correlation coefficient for the following three (out of four) data sets you plotted in part (b): H vs. BB, HR vs. BB, R vs. H. The RBI data column has 139 NAs (or missing values), so you have to do some workaround to caclulate the correlation coefficient. I provided it below.

```
cor(B100$H,B100$BB)
```

```
## [1] 0.7008802
```

```
cor(B100$HR,B100$BB)
```

```
## [1] 0.5896373
```

```
cor(B100$R,B100$H)
```

```
## [1] 0.9210048
```

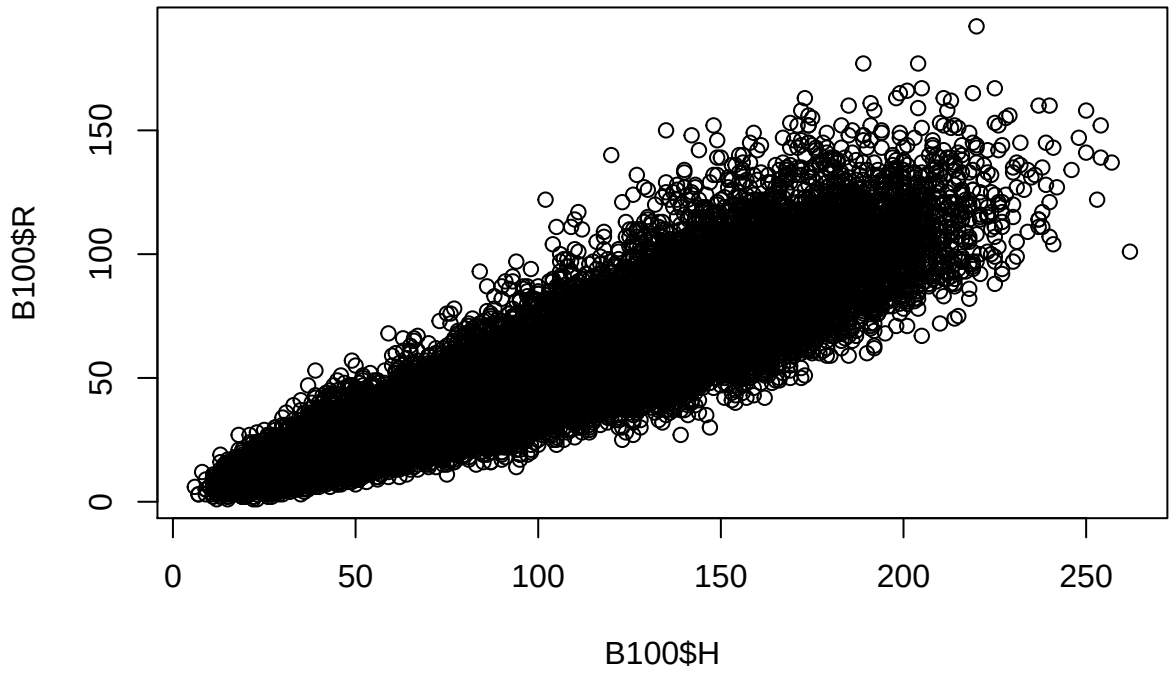
```
#calcualate R for H, RBI
# the correlation coefficient for H, RBI is 0.85
```

(e) Did your answer to part c change? Refer to the end of video 3, and describe what underlying assumtion is required for linear regression.

The linear regression model assumes that for any fixed x value e has a normal distribution. My answer from part c has not been changed. The R vs H plot has the strongest correlation value.

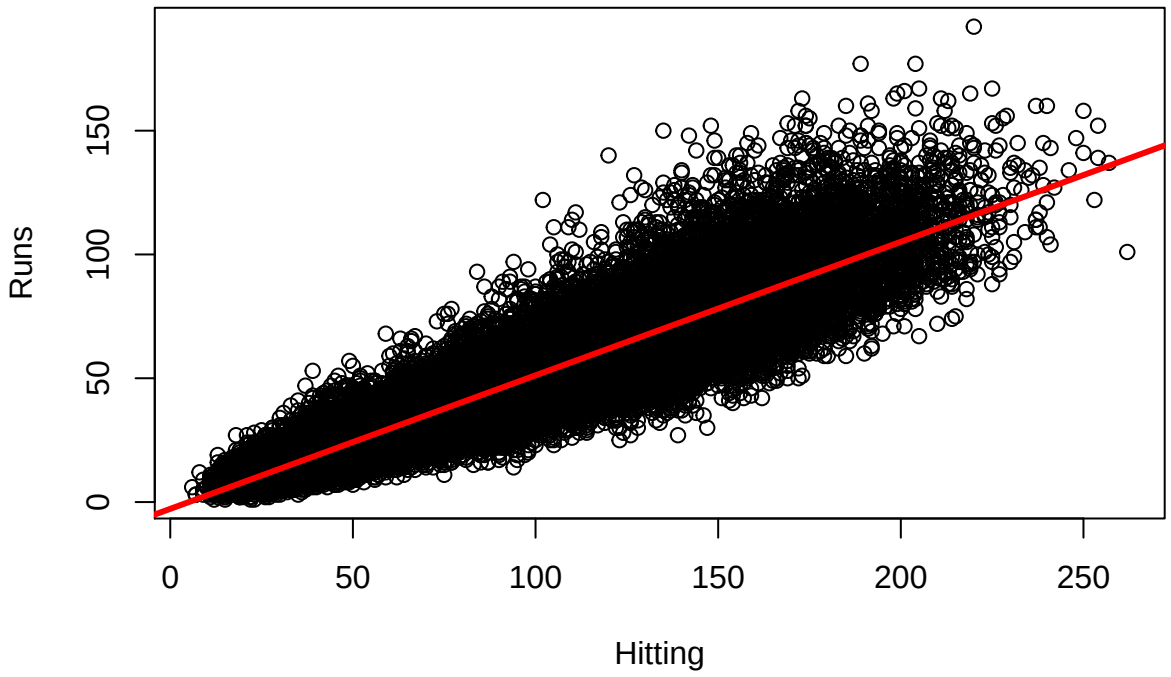
(f) Use the `lm( )` function to fit a linear regression model through the R (runs) versus H (hits) data.

```
lreg <- lm(B100$R~ B100$H)
plot(B100$R~ B100$H)
```



(g) Create a new scatter plot for "R vs. H". Use the `abline( )` function to draw the regression line onto the plot in red. I would use `lwd = 3` to make a thick enough line so it shows up nicely. `lwd` stands for line width.

```
lreg <- lm(B100$R~ B100$H)
plot(B100$R~ B100$H, xlab = "Hitting", ylab = "Runs")
abline(lreg, col = "red", lwd = 3)
```



(h) State what the intercept and slope are for this model, and explain what each of these 2 quantities represent.

The y intercept is -2.75 and the slope is 0.54

```
summary(lreg)
```

```
##
## Call:
## lm(formula = B100$R ~ B100$H)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.520  -6.551  -0.516   5.484  79.951
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.751076   0.124385  -22.12   <2e-16 ***
## B100$H       0.539260   0.001184  455.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.49 on 37083 degrees of freedom
## Multiple R-squared:  0.8482, Adjusted R-squared:  0.8482
## F-statistic: 2.073e+05 on 1 and 37083 DF,  p-value: < 2.2e-16
```

(i) Can you conclude that having more hits causes more runs? Explain. You can refer to p. 115 in the textbook to read about Correlation and Causation.

Yes a fair conclusion is that more hits causes more runs. This is shown by the positive sloping linear regression model that I fit onto the R vs H data.

(j) Use the merged data set for 2012 (the name of that dataframe is `batting_final`) that we created at the beginning of lab and find out which player had the most runs in 2012? You can refer to the height demo on Lab 7. Have you ever heard of this person?

troutmi01, who I've never heard of, has the most runs in 2012.

```
batting_final_sort <- batting_final[order(batting_final$R, decreasing = T),]
head(batting_final_sort)
```

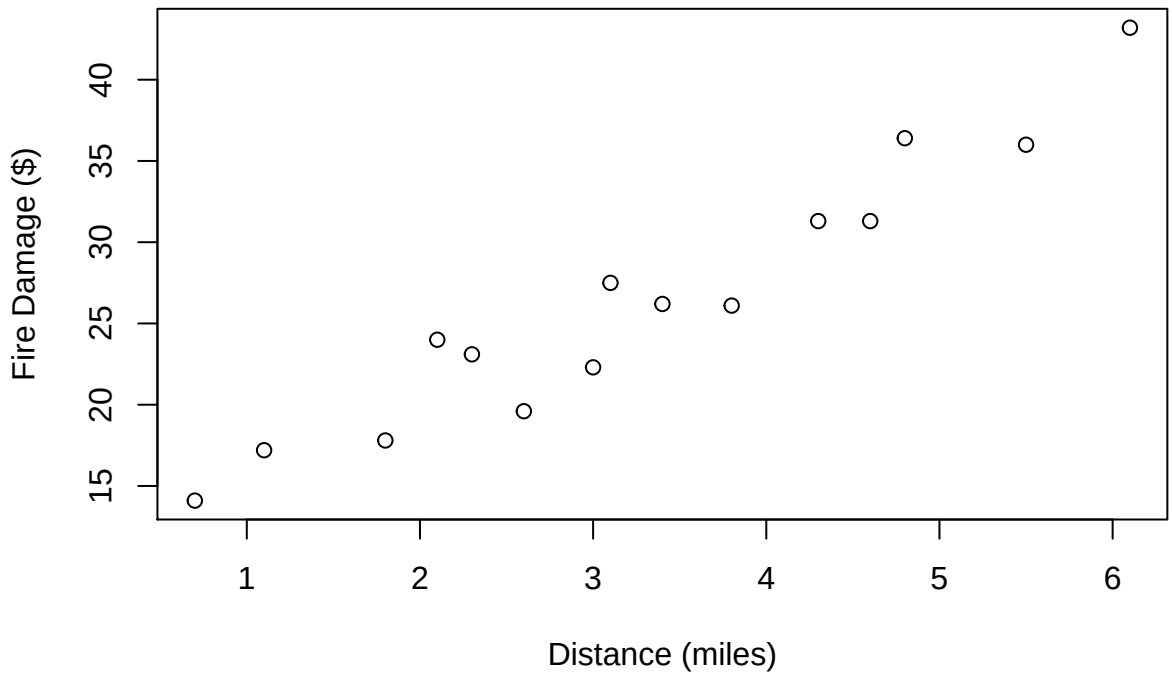
```
##       playerID yearID stint teamID lgID   G G_batting  AB   R   H X2B X3B HR
## 1293 troutmi01   2012     1    LAA   AL 139        NA 559 129 182  27   8 30
## 174  cabremi01   2012     1    DET   AL 161        NA 622 109 205  40   0 44
## 134  braunry02   2012     1    MIL   NL 154        NA 598 108 191  36   3 41
## 831  mccutan01   2012     1    PIT   NL 157        NA 593 107 194  29   6 31
## 1302 uptonju01   2012     1    ARI   NL 150        NA 554 107 155  24   4 17
## 183   canoro01   2012     1    NYA   AL 161        NA 627 105 196  48   1 33
##      RBI SB CS BB  SO IBB HBP SH SF GIDP G_old birthYear birthMonth birthDay
## 1293  83 49  5 67 139   4   6  0  7    7    NA      1991          8        7
## 174  139  4  1 66  98  17   3  0  6   28    NA      1983          4       18
## 134  112 30  7 63 128  15  11  0  5   12    NA      1983         11       17
## 831   96 20 12 70 132  13   5  0  5    9    NA      1986         10       10
## 1302  67 18  8 63 121   5   5  0  6    7    NA      1987          8       25
## 183   94  3  2 61  96  10   7  0  2   22    NA      1982         10       22
##      birthCountry           birthState            birthCity deathYear
## 1293          USA                   NJ             Vineland        NA
## 174     Venezuela               Aragua              Maracay        NA
## 134           USA                   CA        Mission Hills        NA
## 831           USA                   FL           Fort Meade        NA
## 1302          USA                   VA              Norfolk        NA
## 183          D.R. San Pedro de Macoris San Pedro de Macoris        NA
##      deathMonth deathDay deathCountry deathState deathCity nameFirst  nameLast
## 1293         NA       NA                                        Mike     Trout
## 174          NA       NA                                      Miguel   Cabrera
## 134          NA       NA                                        Ryan     Braun
## 831          NA       NA                                      Andrew McCutchen
## 1302         NA       NA                                      Justin     Upton
## 183          NA       NA                                    Robinson      Cano
##           nameGiven weight height bats throws      debut  finalGame  retroID
## 1293 Michael Nelson    230     74    R      R 2011-07-08 2013-09-29 troum001
## 174     Jose Miguel    240     76    R      R 2003-06-20 2013-09-28 cabrm001
## 134     Ryan Joseph    205     74    R      R 2007-05-25 2013-07-21 braur002
## 831   Andrew Stefan    190     70    R      R 2009-06-04 2013-09-28 mccua001
## 1302   Justin Irvin    205     74    R      R 2007-08-02 2013-09-29 uptoj001
## 183   Robinson Jose    210     72    L      R 2005-05-03 2013-09-28 canor001
##        bbrefID
## 1293 troutmi01
## 174  cabremi01
## 134  braunry02
## 831  mccutan01
## 1302 uptonju01
## 183   canoro01
```

Problem 4 Back to infrastructure and civil engineering Suppose a fire insurance company wants to relate the amount of fire damage in major residential fires to the distance between the residence and the nearest fire station.This study is to be conducted in a large suburb of a major city; a sample of 15 recent fires in this suburb is selected. The amount of damage y (in dollars) and the distance x (in miles) between the fire and the nearest fire station are recorded for each fire. The results in tabulated in the csv file names "fire_damage.csv."

(a) Read the data into R, and create a scatter plot of Fire Damage versus Distance. Label the axis with descriptive titles (including units) and comment every line of code.

```
fire<- read.csv("fire_damage.csv")  #reads in file
Damage <- fire$Damage  #creates Damage variable
Distance <- fire$Distance    #creates Distance variable

plot(Damage~Distance, ylab = "Fire Damage ($)",xlab = "Distance (miles)")  #creates scatter plot of Dam
```

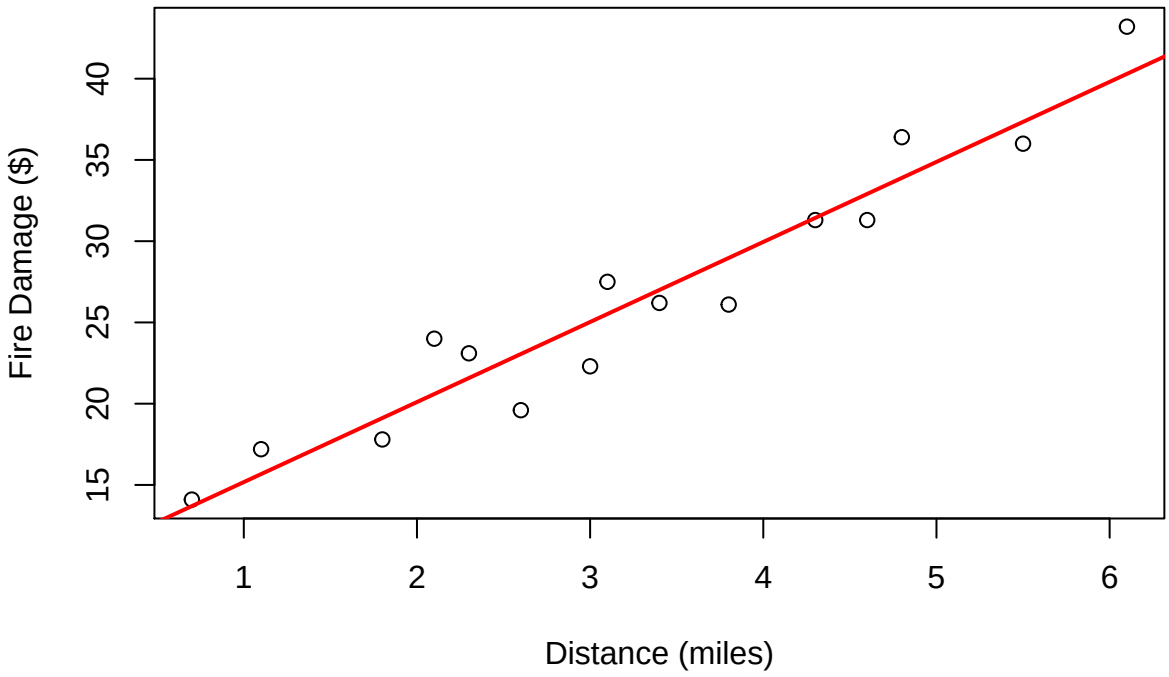


(b) Does the data seem appropriate for a linear regression model? Explain.

This does seems like an appropriate linear regression model because the data point seem to follow a strong linear trend.

(c) Use the `lm( )` function to fit a linear regression model through the damage versus distance data.

```
reg <- lm(Damage~Distance)
plot(Damage~Distance, ylab = "Fire Damage ($)",xlab = "Distance (miles)")  #creates scatter plot of Dam
abline(reg, col = "red", lwd = 2)
```



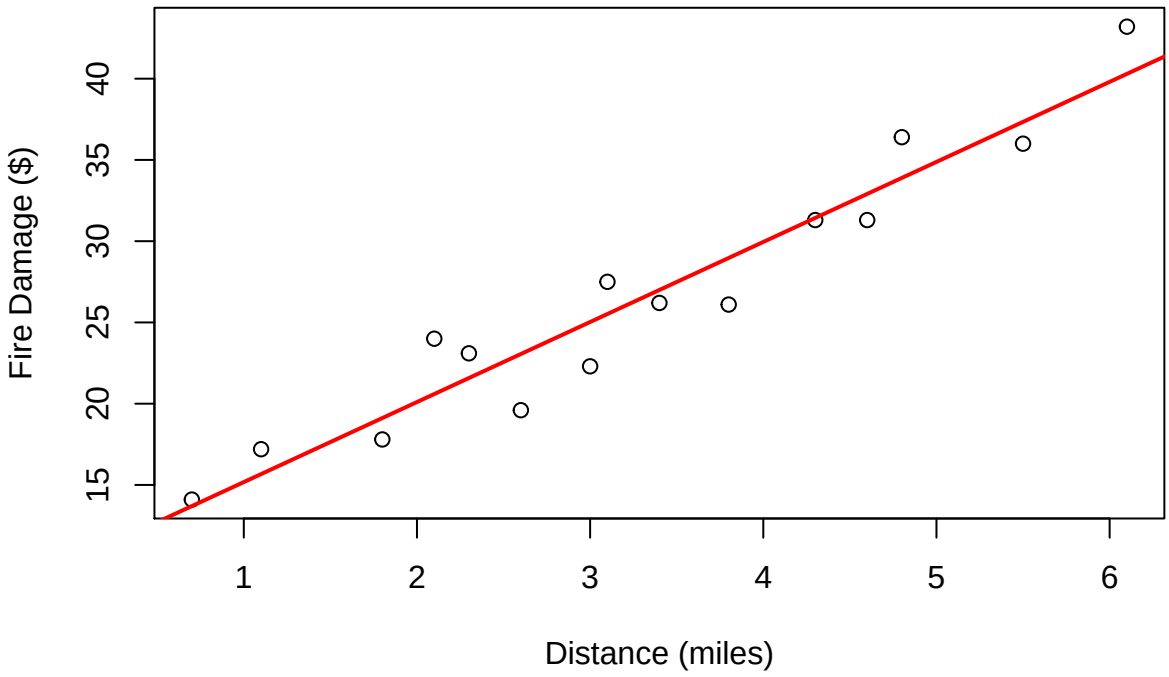

```
summary(reg)
```

```
##
## Call:
## lm(formula = Damage ~ Distance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4573 -1.4750 -0.1308  1.7555  3.4055
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2507     1.4171   7.234 6.61e-06 ***
## Distance      4.9256     0.3919  12.570 1.20e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.311 on 13 degrees of freedom
## Multiple R-squared:  0.924,  Adjusted R-squared:  0.9181
## F-statistic:   158 on 1 and 13 DF,  p-value: 1.196e-08
```

(d) Write the equation for the best fit line, using the slope and intercept above. y= 4.9x + 10.3

(e) Create a new scatter plot for Damage versus Distance, and use the `abline( )` function to draw the regression line onto the plot in a color of your choice.

```
plot(Damage~Distance, ylab = "Fire Damage ($)",xlab = "Distance (miles)")  #creates scatter plot of Dam
abline(reg, col = "red", lwd = 2)
```

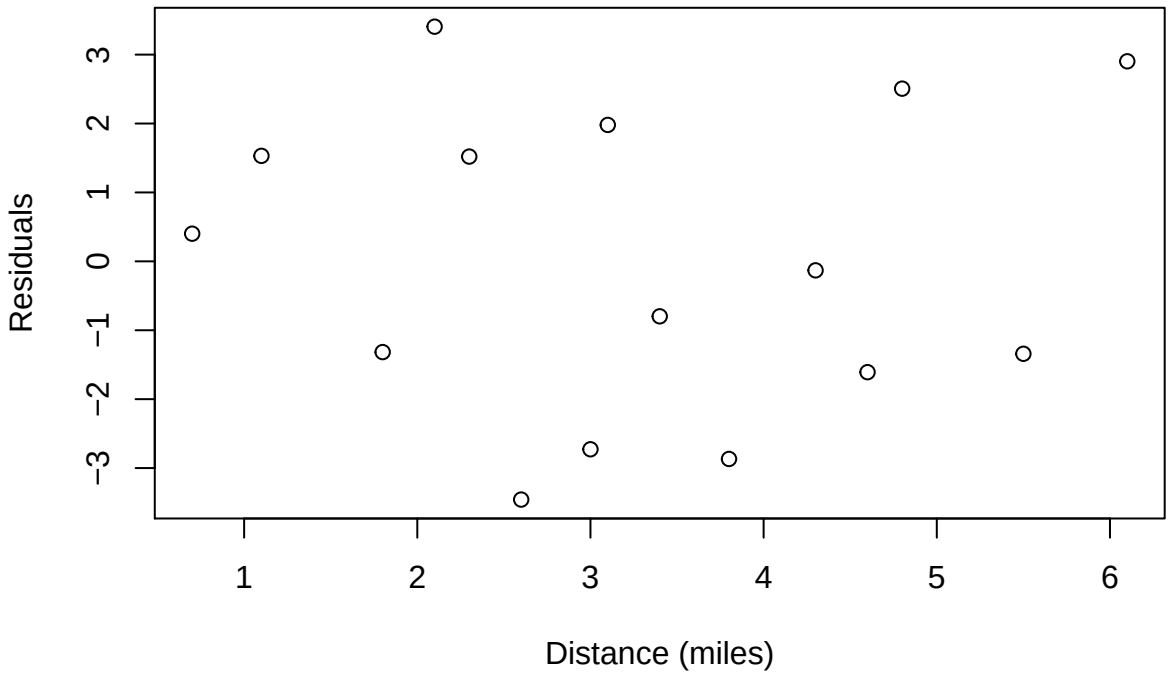


(f) Use the `resid(m)` command where m is the name of your model to calculate the residuals. Create a scatter plot of the residuals versus the independent variable. Label the axis with descriptive titles (including units) and comment every line of code.

```
resid(reg)  #calculates residuals
```

```
##          1          2          3          4          5          6          7
## -0.7977384 -1.3167817 -1.6084560  1.5204193  1.9799410 -1.3414942  0.4013761
##          8          9         10         11         12         13         14
## -2.7274992 -3.4572600 -0.1307766  3.4055389  1.5311369  2.9031470  2.5064244
##         15
## -2.8679776
```

```
plot(Distance,resid(reg), xlab = "Distance (miles)", ylab = "Residuals")  # plots distance versus resid
```



(g) Explain what R^2 ("Multiple R-squared" in the print out from the summary(model_name) command) represents, and how it relates to the residuals.

R squared is the coefficient of determination and it represents the amount of variance between the variables. R squared is a function of the residual and represents the distance from a data point to the linear regression. ****

(h) Based on the plot of residuals, what do you conclude about the appropriateness of the linear model you fit to your data. You can refer to p.126 of the textbook to read about what a desirable plot of residuals looks like.

This is a desirable plot the the (x, residual) pairs because it shows no particular pattern. The points are thoroughly and evenly scattered.

(i) In a couple of sentences describe what residuals are. It may be useful to create a drawing to illustrate your explanation.

Residuals are the vertical distance from the linear regression line to the data points. ****

(j) Next week we will apply more inferential statistics to the fire damage dataset. Until then do a little research to find out about which communities have more or less access to fire stations? Or the relationship of socioeconomic factors and incidence of fire.

Communities of lower income tend to have an increased risk of fires. Other factors tied to low income that can be attributed to increased fire risk are eduction, housing vacancy, housing crowding, and much more.

https://www.usfa.fema.gov/downloads/pdf/statistics/socio.pdf

To get started here is a link to a FEMA report:

https://www.usfa.fema.gov/downloads/pdf/statistics/socio.pdf

You can use this source or look for others, but please cite your source.