

Lab 1

Y. Samuel Wang

Katherine Pulham

01/30/23

Lab 1

Today we will be reviewing various data sets having to do with the Buffalo Bills and US presidential elections. Our goals for this week are

- Review concepts about regression and correlation
- Introduce the `lm` function
- Examine the effect of outliers

Best Fitting Line: Buffalo Bills

The Buffalo Bills are a team in the National Football League based out of Buffalo, NY. To review a few points about regression, we'll consider the weight and height of the Buffalo Bills roster. First, let's read in the data and plot what it looks like.

```
file <- "https://raw.githubusercontent.com/ysamwang/btry6020_sp22/main/lab1/buffaloBills.csv"
buffaloBills <- read.csv(file)
```

Let's take a look at what's in the data. We can use the `head` function to view the first few lines of our data.

```
# The buffaloBills variable stores a table which contains information about each player.
#The 'head' command shows the first few lines of the table
head(buffaloBills)
```

```
##           Player Number Position Height Weight Experience      College
## 1 Mario Addison    97         DE    75    260         11          Troy
## 2   Josh Allen    17         QB    77    237          4        Wyoming
## 3 Boogie Basham   96         DE    75    274          0      Wake Forest
## 4   Tyler Bass     2         K    70    183          2 Georgia Southern
## 5   Ryan Bates    71         G    76    302          3        Penn State
## 6  Cole Beasley   11        WR    68    174         10           SMU
```

```
# dim gets the size of the table
dim(buffaloBills)
```

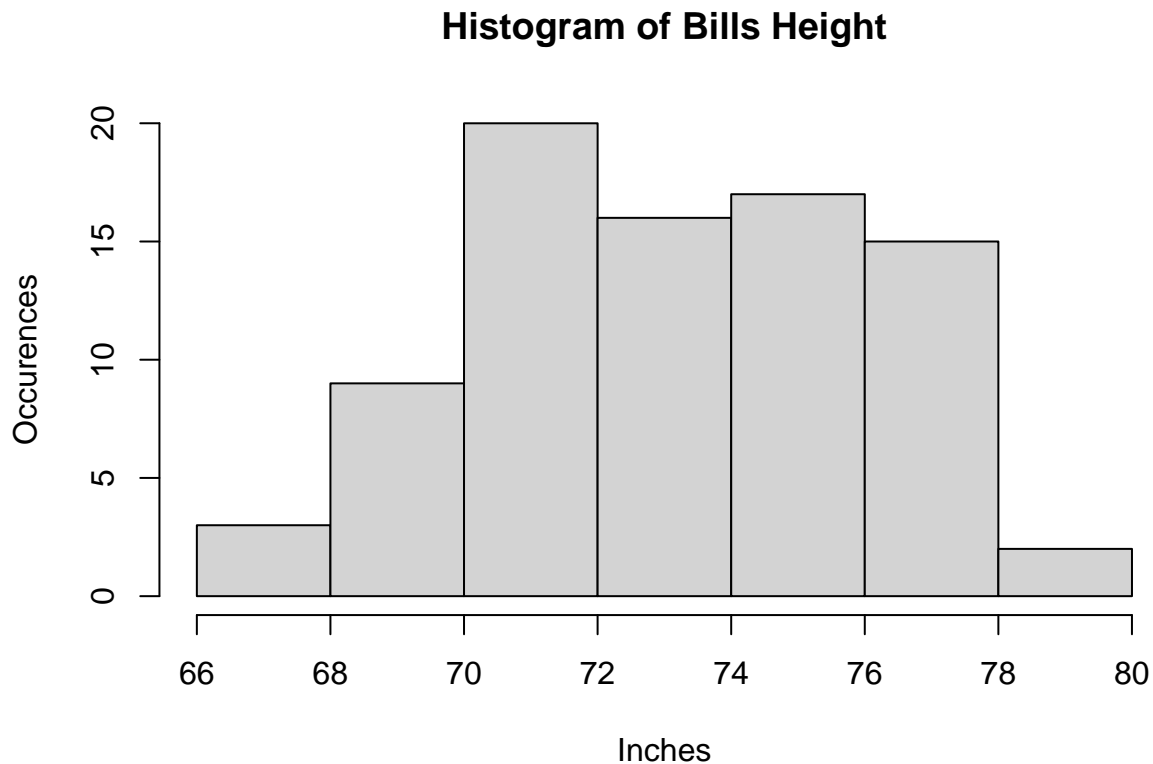
```
## [1] 82  7
```

Given a table with multiple columns, we can use the `$` operator to pull out specific columns. For example `buffaloBills$Height` will return the `Height` column from `buffaloBills`. Notice for the `hist` command, we include the following arguments to label the plot (`main` is the main title, `yaxis` is the label for the y-axis and `xaxis` is the label for the x-axis). We can first view a histogram of the "Height" column which tells us how many times a specific number occurred in our data set.

```
# To reference a specific column in the table, you can use
# the dollar sign and then use the column name. Note that it is case sensitive
buffaloBills$Height
```

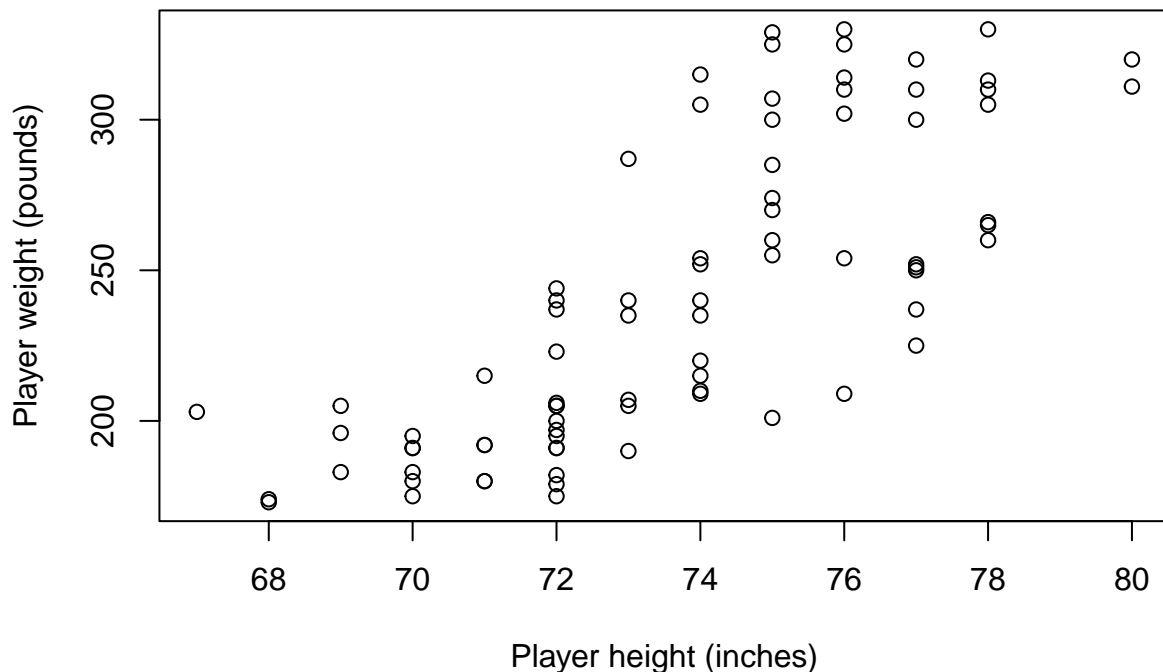
```
## [1] 75 77 75 70 76 68 70 80 76 74 77 72 72 80 77 78 76 74 75 72 72 72 77 74 72
## [26] 71 70 71 72 73 76 76 69 74 73 68 72 78 69 72 78 73 75 72 78 71 67 72 70 77
## [51] 74 72 78 70 75 75 74 77 74 74 73 72 73 75 78 75 73 74 72 77 71 69 78 75 71
## [76] 75 76 76 74 72 77 70
```

```
hist(buffaloBills$Height, main = "Histogram of Bills Height",
      ylab = "Occurences", xlab = "Inches")
```



Suppose I am interested in the line which best describes the relationship between height (x variable) and weight (y variable) for the current Buffalo Bills roster. In such cases, it is helpful to first create a scatterplot of the variables of interest:

```
plot(
  buffaloBills$Height, buffaloBills$Weight, #x and y variables respectively
  type = "p", #p for points, we want a scatterplot,
  xlab = "Player height (inches)",
  ylab = "Player weight (pounds)"
)
```



Note here that my **population** of interest is the current Bills roster. Thus, in this case, I can actually calculate my **parameters** of interest, the b_0 and b_1 which minimize the sum of squared residuals, because I have access to the entire population (note this is typically not the case).

We can use the `cov`, `var`, and `mean` functions to calculate the relevant sample quantities.

```
# Using the formulas from class
b1 <- cov(buffaloBills$Weight, buffaloBills$Height) / var(buffaloBills$Height)
b0 <- mean(buffaloBills$Weight) - b1 * mean(buffaloBills$Height)
```

```
# Population parameters
b0
```

```
## [1] -730.2878
```

```
b1
```

```
## [1] 13.17275
```

So our estimated regression model would be

$$\text{Weight}_i = -730.2878 + 13.1725 \times \text{Height} + \epsilon_i$$

Questions

- How should we interpret these parameters?

We can also use the `lm` function (`lm` stands for linear model) to do all the work for us. Let's take the output of `lm` and assign it to the variable `reression.model`. Inside the `lm` function, we've specified the formula we

want the function to fit. The response variable (y) is on the left side of the \sim (it should be located next to the number 1 on your keyboard). On the right hand side of the tilde, we put the explanatory variable. We also specify the data frame which contains the data of interest.

```
# Using the lm function, we put the dependent variable on the left
# and we put the covariate on the right
buffalo.regression <- lm(Weight ~ Height, data = buffaloBills)

# fitted coefficients can be retrieved by
buffalo.regression$coefficients

## (Intercept)      Height
## -730.28782    13.17275
```

Using these values, we can create predictions for each player's weight based on their height. We can also calculate the residual and check that the sum of the residuals is 0 as we claimed in class.

```
y.hat <- b0 + b1 * buffaloBills$Height
residual <- buffaloBills$Weight - y.hat
sum(residual)

## [1] 3.637979e-12

# Check to see that the residual calculated by hand is the same as getting it
# from the lm function. We just check the first ten values
(buffalo.regression$residual - residual)[1:10]

##          1          2          3          4          5
## 2.827960e-12 -9.947598e-14 -6.394885e-14 -8.881784e-15 -1.065814e-13
##          6          7          8          9         10
## -5.506706e-14 -7.993606e-15 -1.243450e-14 -1.065814e-13 -3.552714e-14

# We can also get the predicted/fitted values using $fitted.values
# we just look at the first ten values below
buffalo.regression$fitted.values[1:10]

##          1          2          3          4          5          6          7          8
## 257.6682 284.0137 257.6682 191.8045 270.8410 165.4590 191.8045 323.5319
##          9         10
## 270.8410 244.4955
```

Now let's check to see that these values of b_0 and b_1 actually minimize the sum of squared errors

$$\text{RSS} = \sum_i (y_i - \hat{y}_i)^2$$

To do this, let's first calculate the RSS for our current estimates of b_0 and b_1

```
sum(residual^2)

## [1] 83857.67
```

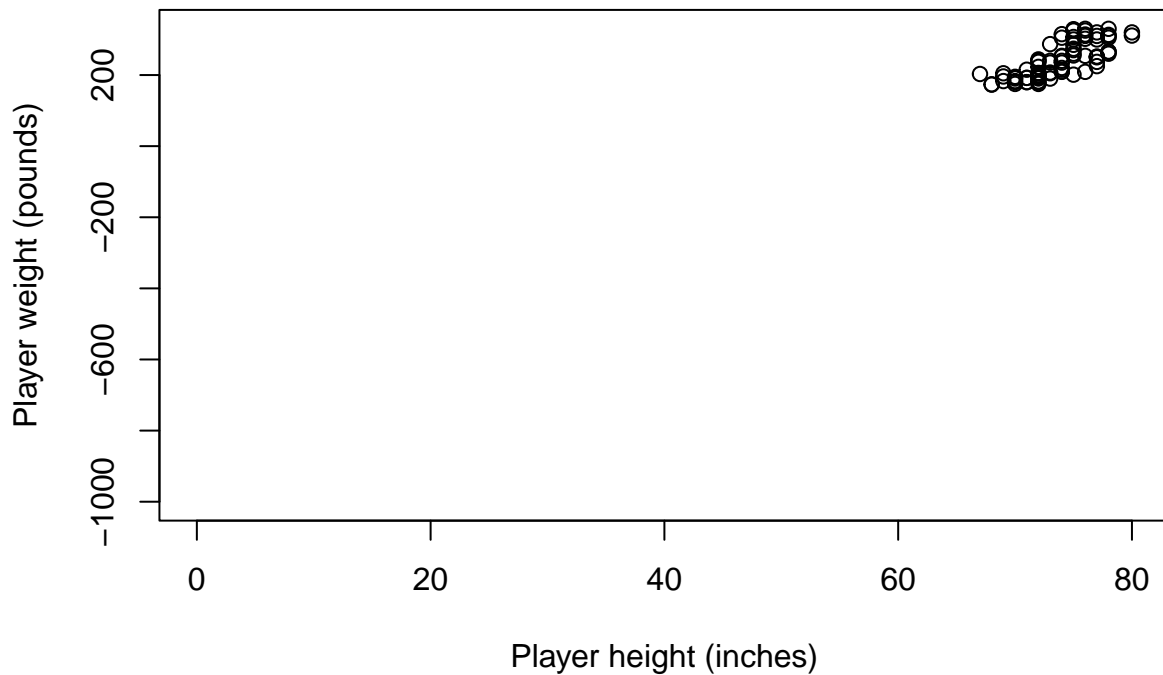
Now let's take a quick eyeball at the plot, and select a value for b_0 and b_1 (pretend you don't know the actual values we just calculated). To make these guesses, it is helpful to modify the scatterplot from earlier:

```
plot(
  buffaloBills$Height, buffaloBills$Weight, #x and y variables respectively
  type = "p", #p for points, we want a scatterplot,
  xlab = "Player height (inches)",
```

```

ylab = "Player weight (pounds)",
xlim = c(0,max(buffaloBills$Height)), #include y=0 and x=0 on graph
ylim = c(-1000,max(buffaloBills$Weight))
)

```

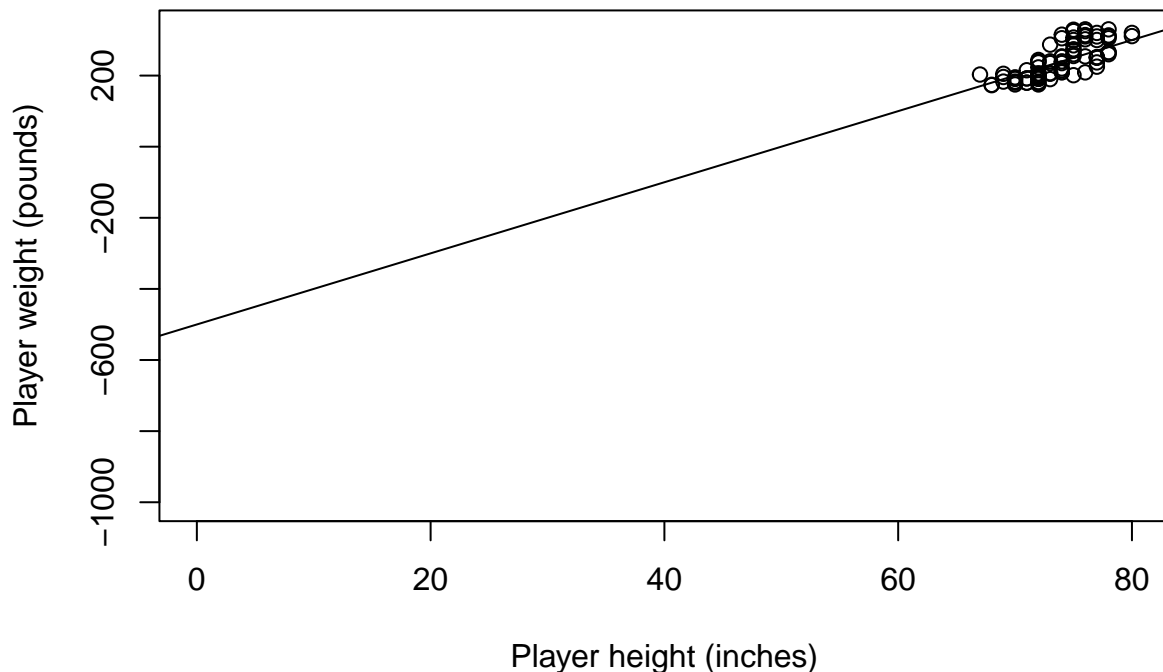


I've filled in a guess based on this scatterplot, but you should change the code to your own values for `b0.guess` and `b1.guess`. Here is the scatterplot, with my guessed values added in:

```

plot(
  buffaloBills$Height, buffaloBills$Weight, #x and y variables respectively
  type = "p", #p for points, we want a scatterplot,
  xlab = "Player height (inches)",
  ylab = "Player weight (pounds)",
  xlim = c(0,max(buffaloBills$Height)), #include y=0 and x=0 on graph
  ylim = c(-1000,max(buffaloBills$Weight))
)
abline( #abline adds a line to the current plot
  a = -500, #y-intercept
  b=10      #slope
)

```



Lets' check the RSS based on these guesses:

```
b0.guess <- -500
b1.guess <- 10
y.hat.guess <- b0.guess + b1.guess * buffaloBills$Height
residual.guess <- buffaloBills$Weight - y.hat.guess
sum(residual.guess^2)
```

```
## [1] 91932
```

Questions

- What is the RSS for your “guessed” values of b_0 and b_1 ?
– 91932
- Is it less than the for the least squares values of b_0 and b_1 ?
– This is larger than the RSS calculated from the `lm` function earlier, which was 83857.67.

However, let's suppose I didn't have data for the full roster, but instead I needed to gather it myself. I ask Sean McDermott, the Bills Coach, and he says I can get the data from the players. However, since they're in the middle of the season and he doesn't want to distract the players, he says I can only ask 10 of the players, not the entire team. So I randomly select 10 players out of the 82 listed on the roster and get the following data.

To simulate this hypothetical situation happen, we first use the `sample` function to pick 10 random numbers between 1 and 82 (the number of players on the roster). Note that `c(1:82)` is shorthand for a vector containing all whole numbers between 1 and 82.

```
players <- sample(x = c(1:82), size = 10)
```

```
# Set of players we selected. This is will be our sample
players
```

```
## [1] 82 15 41 8 35 24 60 49 46 75
```

```
buffaloBills[players, ]
```

```
##           Player Number Position Height Weight Experience      College
## 82 Rachad Wildgoose    32       CB      70     191         R   Wisconsin
## 15 Tremaine Edmunds   49      MLB      77     250         4 Virginia Tech
## 41      Efe Obada      93       DE      78     265         4
## 8      Spencer Brown   79       OT      80     311         0 Northern Iowa
## 35 Tyler Matakevich   44      ILB      73     235         6         Temple
## 24      Jerry Hughes   55       DE      74     254        12          TCU
## 60 Joe Giles-Harris   42       LB      74     240         2          Duke
## 49 Marquez Stevenson   5       WR      70     180         R    Houston
## 46 Emmanuel Sanders    1       WR      71     180        12          SMU
## 75 Tre'Davious White  27       CB      71     192         5          LSU
```

We then fit a regression to the data from the 10 players selected. The 10 players that we would select is our **sample**, and the \hat{a} and \hat{b} we would get from only measuring 10 players are **statistics** which describe our sample.

```
buffalo.regression.subset <- lm(buffaloBills$Weight[players]~ buffaloBills$Height[players])
```

```
# The statistics we calculate from our sample
```

```
buffalo.regression.subset$coefficients
```

```
##           (Intercept) buffaloBills$Height[players]
##           -630.27419                11.65412
```

Questions

- Try this out yourself by running the code. You will get a different answer because your sample will probably be different from mine.
- How do these values differ from our parameters calculated above?
- Should I use the ‘population values’ from the Buffalo Bills roster to make predictions about the average American adult? Would you expect the ‘population values’ for the American adult population be different?

Let’s see how these values differ as we take many random samples. To do this, we will use a for loop which repeats a block of code. Each time it repeats the block, it sets an index variable (in this case i) to the next value in the specified vector. We will repeat this procedure 500 times. We also create two vectors (record.b0 and record.b1) to record the estimates values of \hat{b}_0 and \hat{b}_1 for each sample

```
sample.size <- 500
```

```
record.b0 <- rep(0, sample.size)
```

```
record.b1 <- rep(0, sample.size)
```

```
### Test out to see how a for loop works
```

```
# for(i in 1:5){
```

```
#   print(i^2)
```

```
# }
```

```
for(i in c(1:sample.size)){
```

```
  # Set of players we selected. This is will be our sample
```

```

players <- sample(c(1:dim(buffaloBills)[1]), size = 10)

# calculate the statistics
buffalo.regression.subset <- lm(buffaloBills$Weight[players]~ buffaloBills$Height[players])

# record the statistics we calculate from our sample
#NOTE: I HAD TO SWITCH THE INDICES HERE TO GET THE NEXT SECTION TO WORK
record.b1[i] <- buffalo.regression.subset$coefficients[2]
record.b0[i] <- buffalo.regression.subset$coefficients[1]
}

```

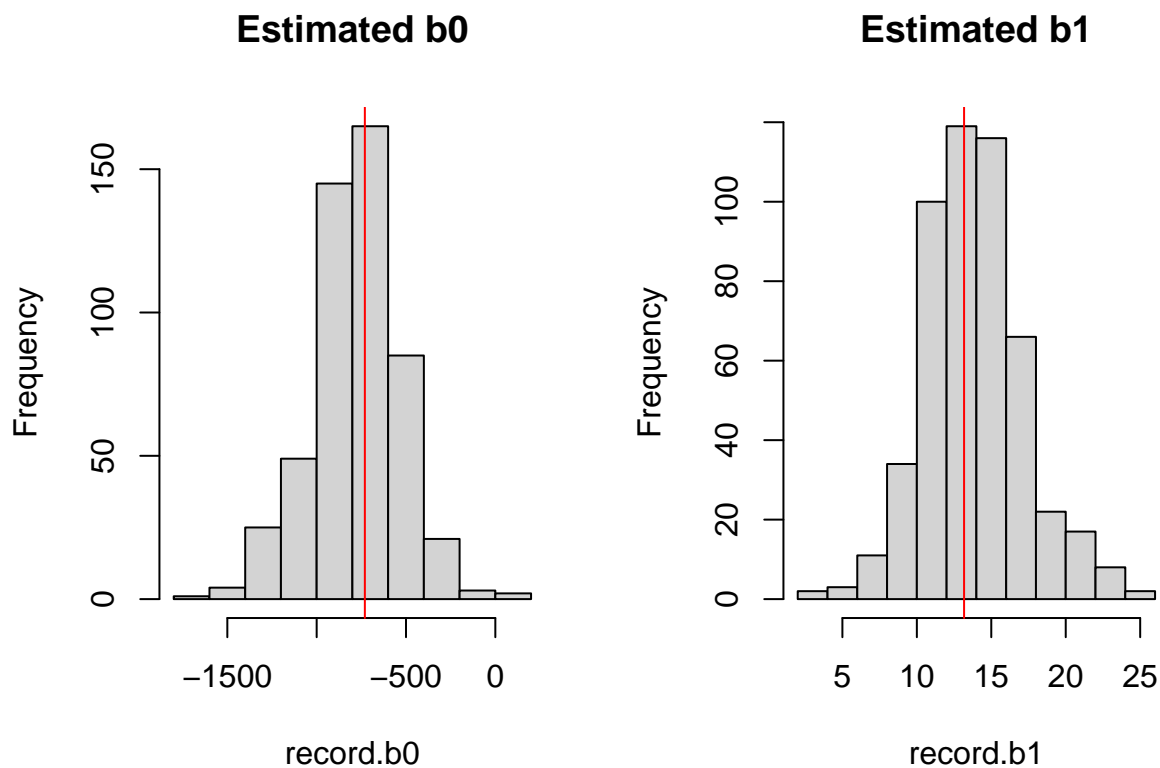
We can plot the distribution of the estimated \hat{b}_1 and \hat{b}_0 values and see that they vary with each sample around the true value of b_1 and b_0 we calculated above. The parameter values are indicated with the red vertical lines in the plots below.

```

# this arranges the plots together so there is 1 row and 2 columns
par(mfrow = c(1,2))

hist(record.b0, main = "Estimated b0")
abline(v = b0, col = "red")
hist(record.b1, main = "Estimated b1")
abline(v = b1, col = "red")

```



We can see that each random sample we take gives us a good estimate of the true values of b_0 and b_1 , but \hat{b}_0 and \hat{b}_1 are different each time.

Linear Models with US Presidential Elections

In the 2000 US Presidential election with George Bush vs Al Gore, the entire election was decided by the state of Florida which itself was decided by less than 600 votes (a margin of .009%). In particular, Palm Beach county used a butterfly ballot which was widely criticized for its confusing design. Many speculated that this may have caused a large number of voters who intended to vote for Al Gore to vote for Pat Buchanan (Reform Party) instead.

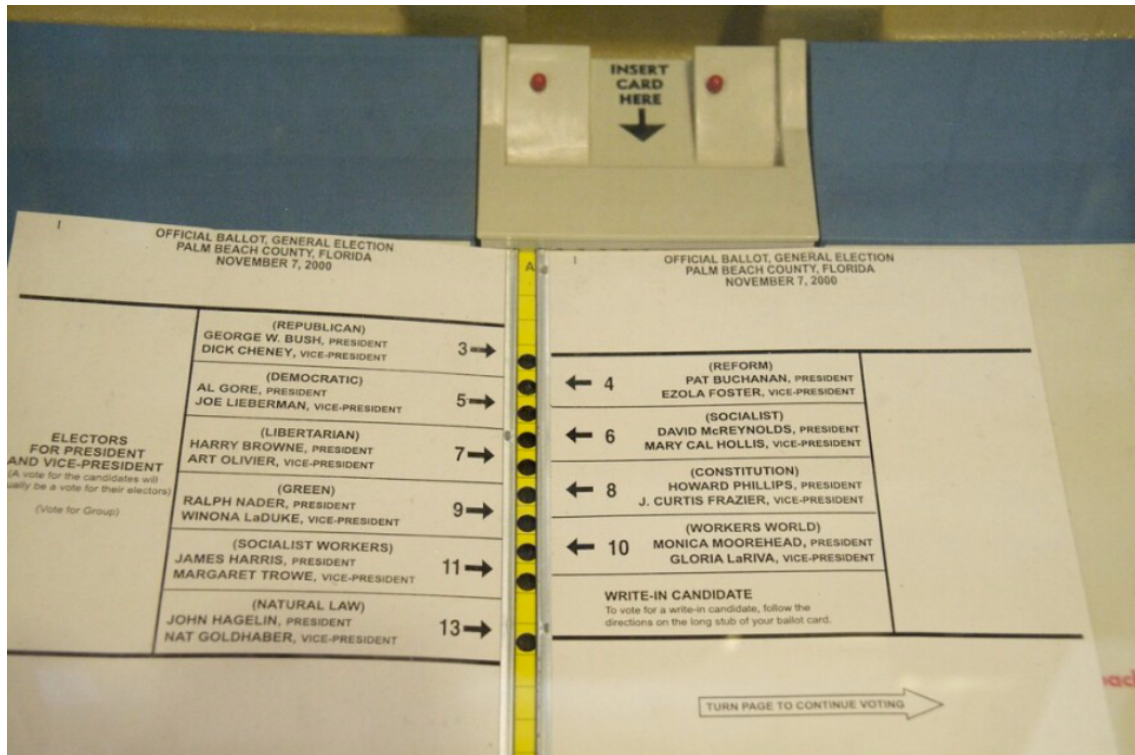


Figure 1: Confusing butterfly ballot

We would expect that the number of registered voters in 2000 who belonged to the Reform party should be a pretty good predictor of how many people ended up voting for Pat Buchanan. For each county in Florida, we have combined vote data from Wikipedia with data from the Florida Division of Elections on the party affiliation of the registered voters in 2000. The variable `Buch.Votes` is the number of votes cast for Pat Buchanan and `Reg.Reform` is the number of registered reform party voters. `Total.Reg` is the total number of registered voters in that county.

```
florida <- read.csv("https://raw.githubusercontent.com/ysamwang/btry6020_sp22/main/lab1/FL.csv")
head(florida)
```

##	County	Reg.Dem	Reg.Rep	Reg.Reform	Total.Reg	Buch.Votes
## 1	Alachua	64135	34319	91	120867	263
## 2	Baker	10261	1684	4	12352	73
## 3	Bay	44209	34286	55	92749	268
## 4	Bradford	9639	2832	3	13547	45
## 5	Brevard	107840	131427	148	283680	570
## 6	Broward	456789	266829	332	887764	795

First, let's take a look at the distributions of registered reform party voters and votes for the reform party candidate Pat Buchanan. The red line in the plots below indicate the values for Palm County.

```

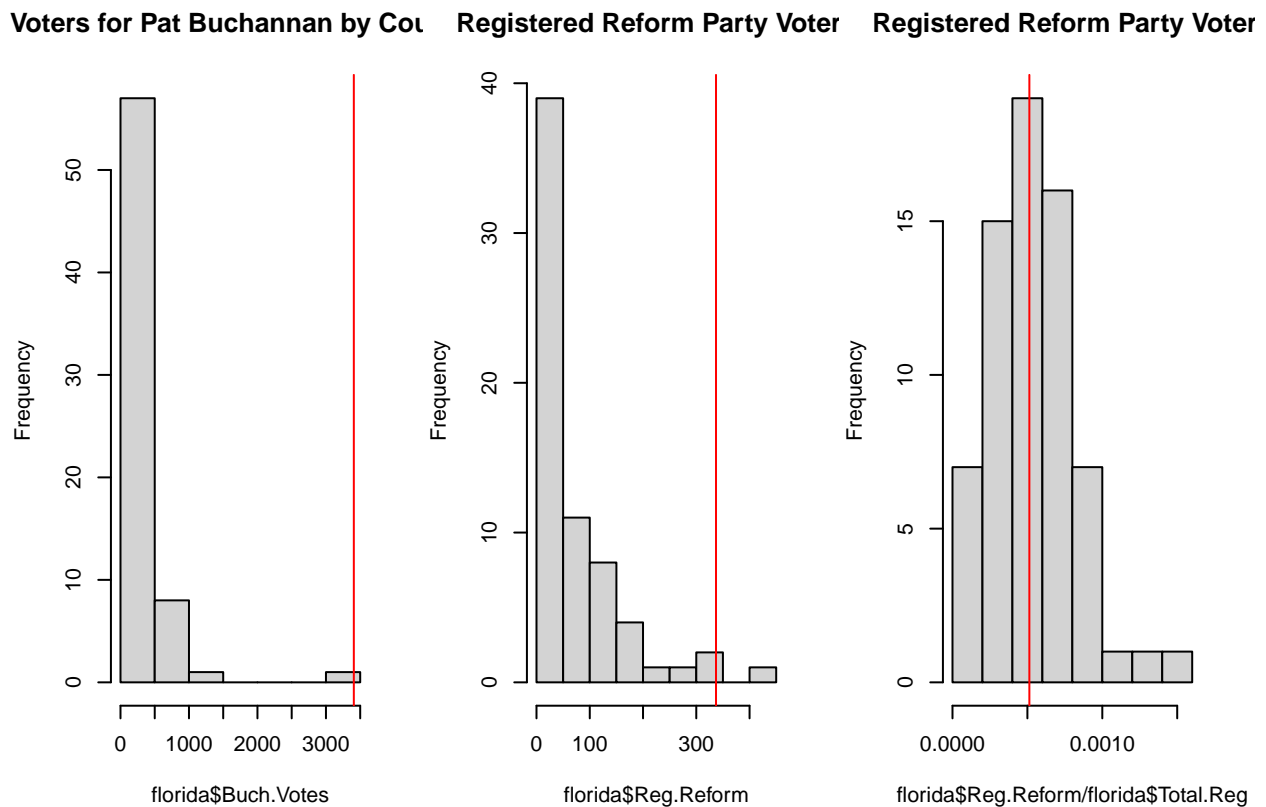
par(mfrow = c(1,3))

# Histogram of number of votes for Pat Buchanan
hist(floriga$Buch.Votes, main = "Voters for Pat Buchanan by County")
abline(v = floriga$Buch.Votes[50], col = "red")

# Histogram of total registered reform party voters
hist(floriga$Reg.Reform, main = "Registered Reform Party Voters")
abline(v = floriga$Reg.Reform[50], col = "red")

# Normalize for the number of total registered voters
hist(floriga$Reg.Reform/floriga$Total.Reg,
     main = "Registered Reform Party Voters")
abline(v = floriga$Reg.Reform[50]/floriga$Total.Reg[50], col = "red")

```



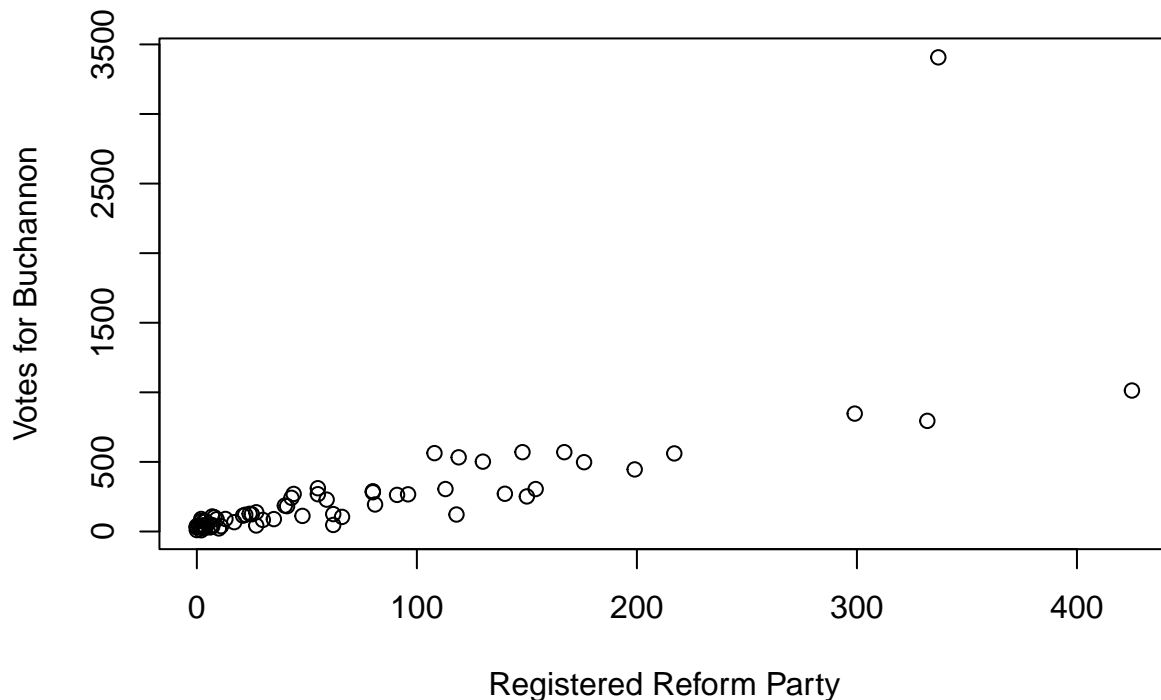
We can also plot the scatter plot, and use the `cor` function to calculate the sample correlation

```

plot(floriga$Reg.Reform, floriga$Buch.Votes,
     xlab = "Registered Reform Party", ylab = "Votes for Buchannon",
     main = "Florida 2000 Presidential Election")

```

Florida 2000 Presidential Election



```
round(cor(florida$Reg.Reform, florida$Buch.Votes), 3)
```

```
## [1] 0.741
```

Below, we calculate coefficients for the least squares line:

$$\text{estimated votes for Buchanan}_i = \hat{b}_0 + \hat{b}_1 \text{number of registered reform party voters}_i \quad (1)$$

```
florida.regression <- lm(Buch.Votes ~ Reg.Reform, data = florida)
```

We can get the fitted coefficients (\hat{b}_0 and \hat{b}_1) from the `florida.regression` object by using `$coefficients`. The first value is the y-intercept, and the second value is the coefficient on our explanatory variable (year.2004), which is denoted by \hat{b} in the equation above. We can see that the values returned by `lm` are the same as the values we calculated above

```
florida.regression$coefficients
```

```
## (Intercept)  Reg.Reform
##   -0.246390    3.652078
```

```
b0.hat <- florida.regression$coefficients[1]
b1.hat <- florida.regression$coefficients[2]
```

We can calculate predicted values using the estimated coefficients. Alternatively, we can get the predicted (or fitted values) from the `lm` object 'florida.regression'.

```
y.hat <- b0.hat + b1.hat * florida$Reg.Reform
```

```
# check to see that the predicted values we formed are the same as the
```

```
# lm object's fitted values (at least up to 10 digits)
round(florigida.regression$fitted.values - y.hat, 10)
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

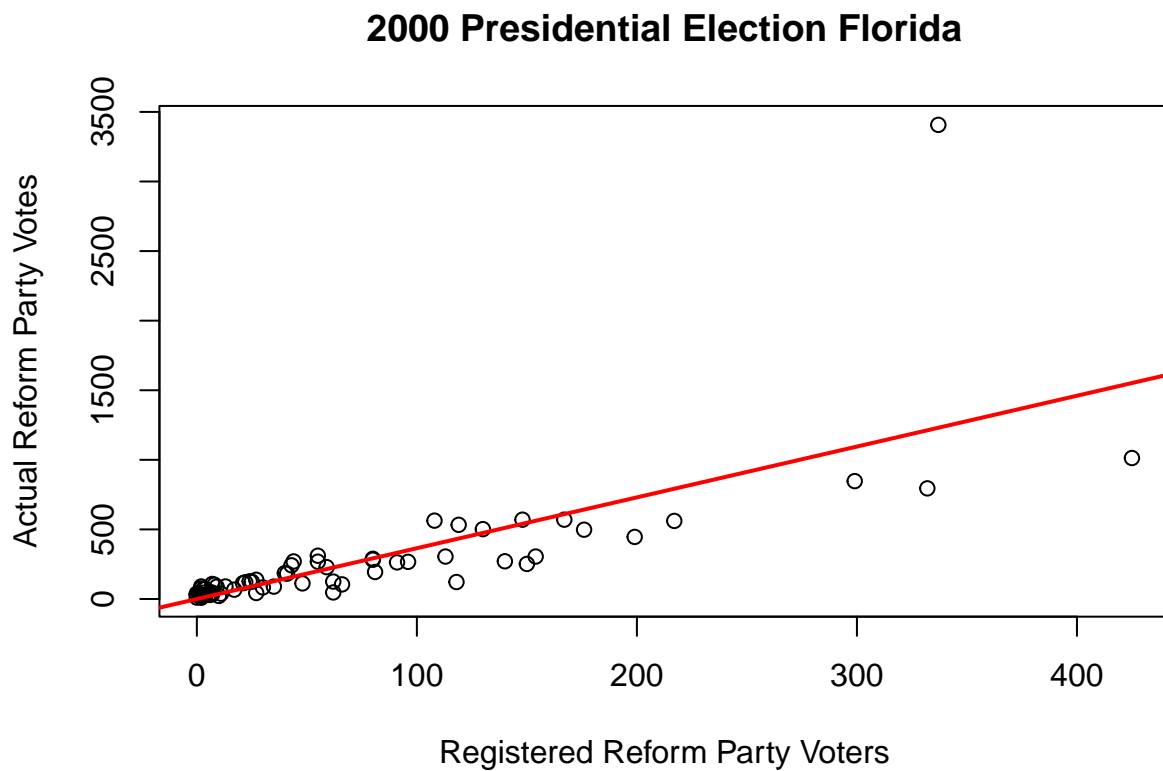
In fact, the `lm` object has lots of information stored which we can access. To see, use the `names` function:

```
names(florigida.regression)
```

```
## [1] "coefficients" "residuals" "effects" "rank"
## [5] "fitted.values" "assign" "qr" "df.residual"
## [9] "xlevels" "call" "terms" "model"
```

Let's take a look at the observed values and the predicted values. To plot the line, we use the `abline` command which plots a line given the y-intercept (specified by the argument `a`) and the slope (specified by the argument `b`). It looks like the model fits relatively well.

```
plot(florigida$Reg.Reform, florigida$Buch.Votes, main = "2000 Presidential Election Florida",
     xlab = "Registered Reform Party Voters", ylab = "Actual Reform Party Votes")
abline(a = b0.hat, b = b1.hat,
      col = "red", lwd = 2)
```

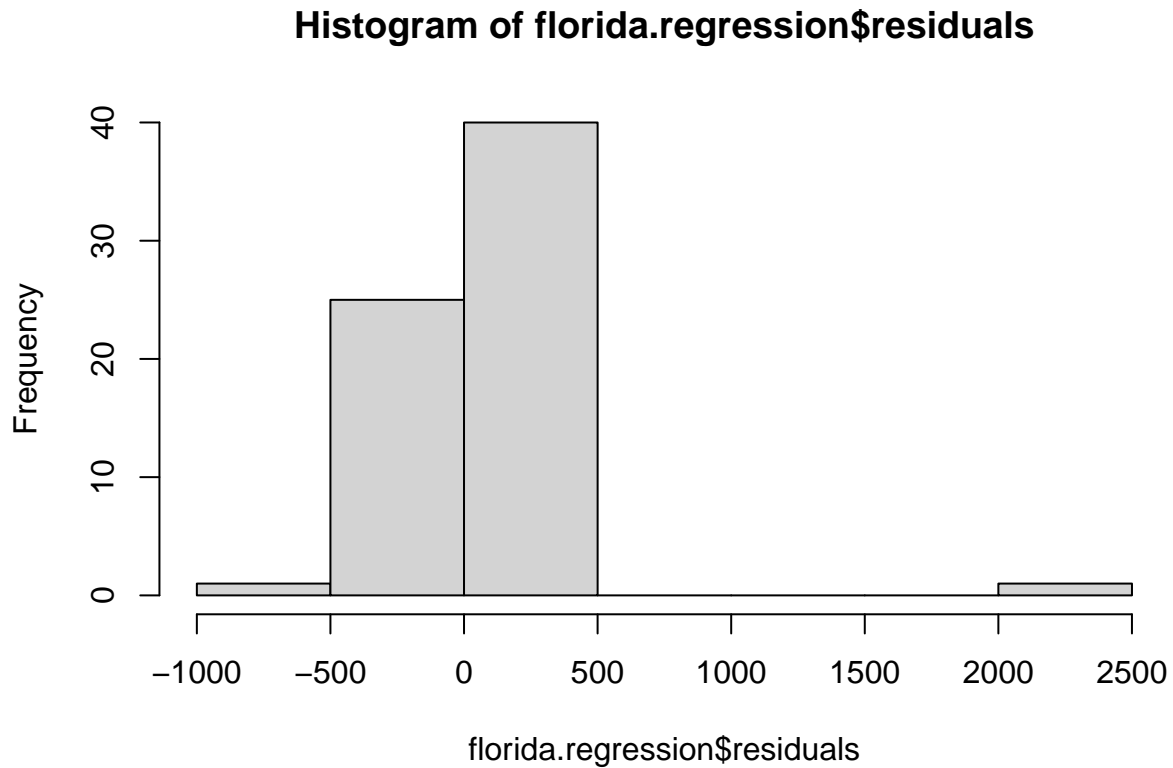


Questions

- Does the line fit well? Does the relationship look mostly linear?
- Are there any outliers?

We can also take a look at the distribution of the residuals.

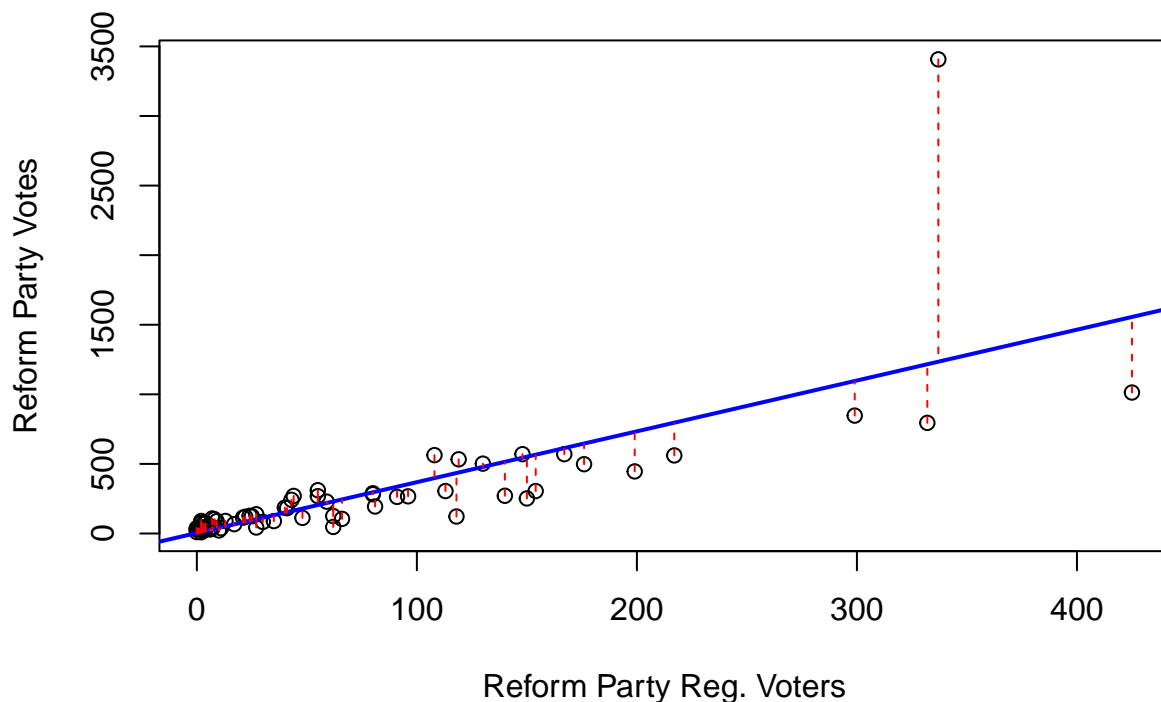
```
hist(florigida.regression$residuals)
```



It looks like there is one county with a large residual.

```
plot(florigida$Reg.Reform, florigida$Buch.Votes,  
     main = "200 Presidential Election Florida Votes",  
     xlab = "Reform Party Reg. Voters", ylab = "Reform Party Votes")  
abline(a = b1.hat, b = b1.hat,  
       col = "blue", lwd = 2)  
  
segments(x0 = florigida$Reg.Reform, y0 = florigida$Buch.Votes,  
         x1 = florigida$Reg.Reform, y1 = y.hat, col = "red", lty = 2)
```

200 Presidential Election Florida Votes



```
# get the name of the county with a large residual
# which.max/which.min returns the index of the max/min value in the vector
florida$County[which.max(abs(florida.regression$residual))]
```

```
## [1] "Palm"
```

Questions

- Does Palm County appear to be an outlier in the joint distribution?
- Based on the number of registered voters belonging to the reform party in Palm County, what is the fitted number of actual votes for Pat Buchanan to be?
- What is the residual for Palm County? (hint: Palm County is the 50th row in our data.frame)

There's a very useful function in R called `summary`, which we've already seen from last lab. We can also use "summary" to our `regression.model` which gives us more information than just the raw output. Notice that it gives estimates for the coefficients, as well as standard errors for the coefficients. Recall in class that we said the estimated \hat{a} and \hat{b} are just estimates (statistics) of a parameter. The standard errors are rough estimates of how much our estimates might change if we took another sample. Recall the exercise above where we took samples of 10 Buffalo Bills players, and each sample gave a different result. The standard error is an estimate of the standard deviation of the histograms we were able to plot.

```
summary(florida.regression)
```

```
##
## Call:
## lm(formula = Buch.Votes ~ Reg.Reform, data = florida)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -538.89 -66.07   15.64   39.77 2176.50
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2464    46.7415  -0.005   0.996
## Reg.Reform    3.6521     0.4099   8.909 7.16e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 302.7 on 65 degrees of freedom
## Multiple R-squared:  0.5498, Adjusted R-squared:  0.5429
## F-statistic: 79.37 on 1 and 65 DF,  p-value: 7.159e-13
```

Let's view the effect of Palm county on the regression and fit another model to the new data.

```
no.palm.county <- florida[-50, ]
florida.regression.no.palm <- lm(Buch.Votes~Reg.Reform, data = no.palm.county)
summary(florida.regression.no.palm)
```

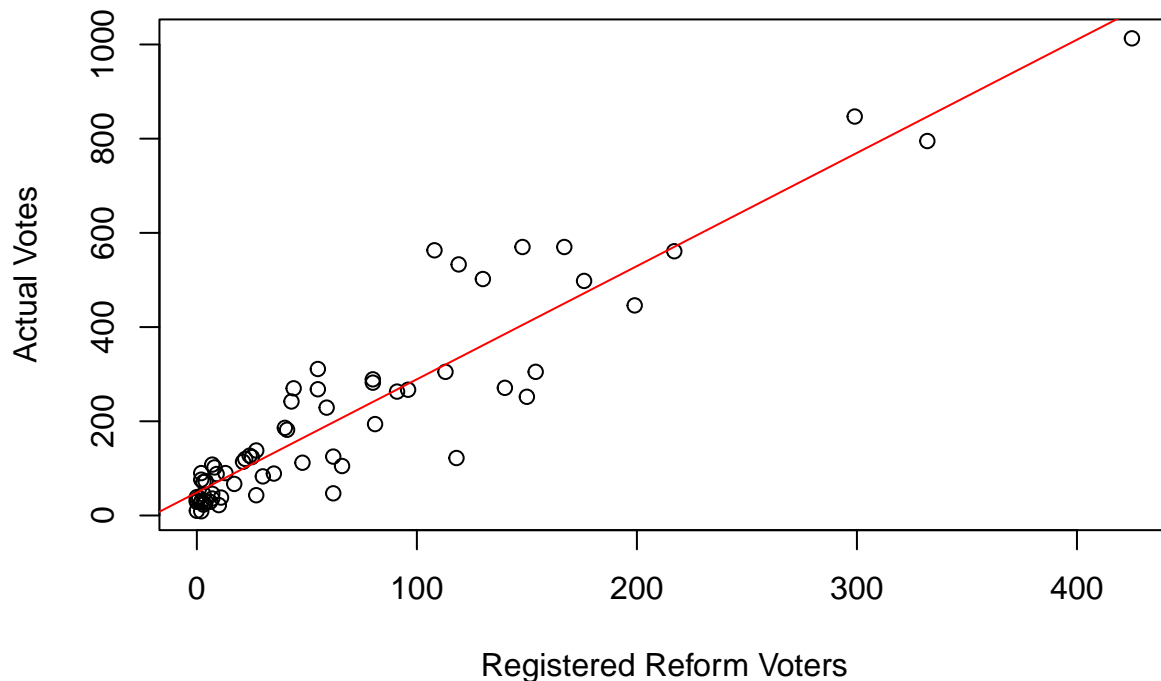
```
##
## Call:
## lm(formula = Buch.Votes ~ Reg.Reform, data = no.palm.county)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -210.38 -38.58 -11.76   34.49  254.65
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.8089    12.4691   3.914 0.000222 ***
## Reg.Reform    2.4031     0.1164  20.648 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.03 on 64 degrees of freedom
## Multiple R-squared:  0.8695, Adjusted R-squared:  0.8674
## F-statistic: 426.3 on 1 and 64 DF,  p-value: < 2.2e-16
```

Let's view the predicted vs observed values without Palm county. Here we can see that the line seems to fit the data much better than before.

```
plot(x = no.palm.county$Reg.Reform,
     y = no.palm.county$Buch.Votes,
     main = "Registered Voters vs Actual Votes (No Palm County)",
     xlab = "Registered Reform Voters", ylab = "Actual Votes")

abline(a = florida.regression.no.palm$coefficients[1],
       b = florida.regression.no.palm$coefficients[2], col = "red")
```

Registered Voters vs Actual Votes (No Palm County)



Questions

- How would we interpret the estimated coefficients from the regression output?
- Using this model, what is the predicted number of votes for Buchanan in Palm County? What is the prediction error? (Note this is similar, but not a residual because we did not use Palm County to fit our model)
- Compare the estimated values for this model with the estimated values of the previous model
- So which model is “correct”? The answer depends on how we define “correct,” but if you had to predict the number of votes in each Florida county for the reform party candidate in this upcoming 2024 election, which model would you use? Why?

As an alternative, we can fit a line which minimizes the absolute deviation (this is like using the median instead of the mean). As you can see, even when we include Palm County, the estimated coefficients are relatively similar to the estimates when we removed Palm County. As we see, quantile regression can be more robust to outliers.

```
# if you don't have the quantreg package, run the following line of code once  
#install.packages("quantreg")
```

```
# load the quantreg package  
library(quantreg)
```

```
## Loading required package: SparseM
```

```
##
```

```
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
```



```
##
##      backsolve
# This is a general function for quantile regression
# the tau parameter specifies the quantile we are intersted in, so using tau = .5
# fits the median
florida.quantile.regression <- rq(Buch.Votes ~ Reg.Reform, data = florida, tau = .5)
summary(florida.quantile.regression)

##
## Call: rq(formula = Buch.Votes ~ Reg.Reform, tau = 0.5, data = florida)
##
## tau: [1] 0.5
##
## Coefficients:
##              coefficients lower bd upper bd
## (Intercept) 34.57407      23.95715 64.08950
## Reg.Reform   2.42593       2.26156  2.74116
```