Lab01 (Mat 565)

KR

2023-01-25

1. (15 pts) Let's replicate the example in class with dataset faithful. This dataset is part of the base distribution of R so you don't need to load any library.

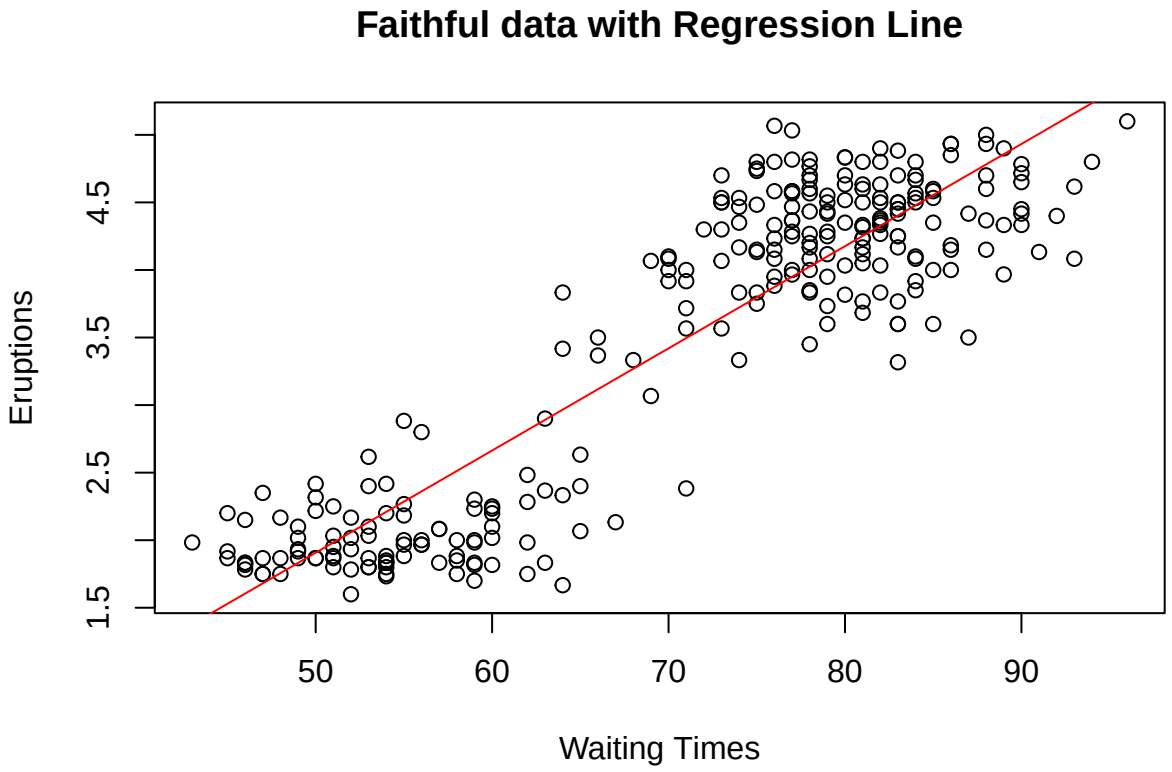
a. Run the regression model of eruptions on waiting times. (replace NULL by the appropriate code)

```
model <- lm(eruptions ~ waiting, data = faithful)
summary(model)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

b. Make a scatterplot of eruptions vs waiting times and overlay the regression line in red color. Put a title to your graph.

```
plot(faithful$waiting, faithful$eruptions,
     xlab = "Waiting Times",
     ylab = "Eruptions",
     main = "Faithful data with Regression Line")
abline(model, col = "red")
```



c. Run a test of hypothesis for the slope. State the null and alternative hypothesis. State the value of the test statistic with the number of degrees of freedom. State the p-value and state the conclusion of the test.

```
# Test of Hypothesis

coefficients <- coef(model)

summary_output <- summary(model)

t_statistic <- summary_output$coefficients["waiting", "t value"]


df <- summary_output$df[2]

p_value <- summary_output$coefficients["waiting", "Pr(>|t|)"]

summary(model)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
```

```
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

Whenever we perform simple linear regression, we end up with the following estimated regression equation

To determine if β_1 is statistically significant, we should perform a t-test with the following test statistic:

where se(β_1) represents the standard error of b1.

from the model output, we can see that the estimated regression equation is:

eruptions = -1.874016 + 0.075628(waiting).

Let us check if the slope coefficient is statistically significant,

t = (β_1 - 0)/se(β_1) t = 0.075628 / 0.002219 t = 34.082

The p-value that corresponds to this t-test statistic is shown in the column called Pr(>|t|) in the output and it turns out to be 2e-16.

H_0: $\beta_1 = 0$,
H_1: β_1 not equal to 0.

p-value: 2e-16 = 0

Conclusion: Since the p-value < 0.05, We reject the null hypothesis, The variable waiting is significant in the model.

d. Construct, by hand, a 98% confidence interval for the slope.

98% confidence interval for β_1: t^* = qt(.995,270) = 2.340238

Upper bound = 0.075628 + 2.340238*0.00219 = 0.08075*
*lower bound = 0.075628 - 2.340238*0.3148 = 0.07050

Verify that your computation coincides with the confidence interval that you can get using the command confint()

```
conf_interval <- confint(model, level = 0.98)
print(conf_interval)
```

```
##                      1 %        99 %
## (Intercept) -2.24878944 -1.49924253
## waiting      0.07043603  0.08081986
```

e. Find the estimate for the model's standard deviation. Verify this computation by computing the standard deviation by hand using the residuals of the model.

```
RSE <- summary_output$sigma
print(RSE)
```

```
## [1] 0.4965129
```

The standard deviation of the residuals is RSE = $\sqrt{\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n-p-1}}$.

1. Extract the residuals from the model.

```r
# Compute the standard deviation by hand using the residuals
residuals <- resid(model)
print(residuals[1:100])
```

```
##           1           2           3           4           5           6
## -0.500591902 -0.409893203 -0.389452162 -0.531916787 -0.021359589  0.597478849
##           7           8           9          10          11          12
## -0.081243433 -0.954359589 -0.033009359 -0.204359589 -0.376893203 -0.561731642
##          13          14          15          16          17          18
##  0.175036046  0.069502433  0.296896306  0.108362693 -1.064916787  0.321268358
##          19          20          21          22          23          24
## -0.458637307  0.149408098 -0.183009359  0.069502433 -0.574963954 -0.277312422
##          25          26          27          28          29          30
##  0.810547838 -0.803103694 -0.318521151  0.209291942 -0.174963954  0.332408098
##          31          32          33          34          35          36
##  0.653175786  0.517663994  0.249571422 -0.143219850  0.110547838 -0.041637307
##          37          38          39          40          41          42
##  0.110874485  0.656780150 -0.755032943 -0.149499329  0.173780150 -0.629404995
##          43          44          45          46          47          48
##  0.088268358 -0.762404995  0.886175786 -1.086103694  0.866827317 -0.034265255
##          49          50          51          52          53          54
##  0.305524254 -0.588032943  1.001919890 -0.216499329 -0.376893203  0.656780150
##          55          56          57          58          59          60
## -0.476893203  0.479896306  0.221431682 -1.299172683  0.617663994  0.065152202
##          61          62          63          64          65          66
## -0.355032943  0.021268358 -0.006125515  0.472524254 -0.846660891 -0.683755225
##          67          68          69          70          71          72
##  0.142036046  0.675036046 -0.974800630  1.053175786 -0.294475746 -0.394149099
##          73          74          75          76          77          78
##  0.399408098  0.504431682 -0.831916787  1.193291942 -0.646660891  0.542036046
##          79          80          81          82          83          84
##  0.009291942 -0.803103694  0.334919890  0.005524254  0.680059630 -0.408800630
##          85          86          87          88          89          90
##  0.420175786  0.151756567  0.076291942  0.340780150  0.410874485 -0.629987537
##          91          92          93          94          95          96
## -0.463660891 -0.599499329 -0.040381411  0.792036046 -1.057544735  0.728803734
##          97          98          99         100
##  0.188268358 -0.048080110 -0.116009359  0.572524254
```

2. calculate the mean squared residuals

```r
mean_squared_residuals <-mean(residuals^2)
print(mean_squared_residuals)
```

```
## [1] 0.2447124
```

The above value we obtained is the mean of the squared residuals.

3. Taking the root over the mean_squared_residuals

$\sqrt{0.2447124} = 0.4956$

This will be the RSE value.

f. Find the model's R^2. What is the interpretation of the R^2?

```
R_squared <- summary_output$r.squared
print(paste("R^2 value:",R_squared))
```

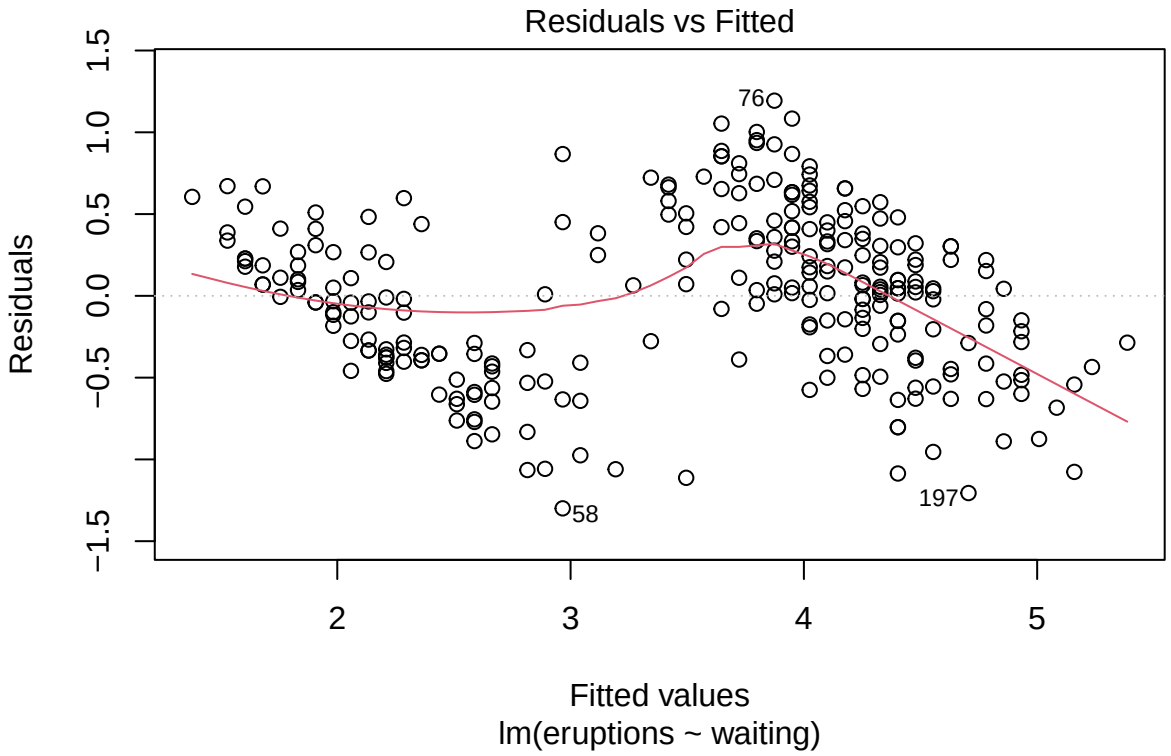
```
## [1] "R^2 value: 0.811460760973309"
```

Interpretation of the R^2: The model provides a reasonably good fit to the data, capturing a significant portion of the variability observed in the eruptions.

g. Look at the first 2 assessment plots plots

Residual plot

```
plot(model,1)
```

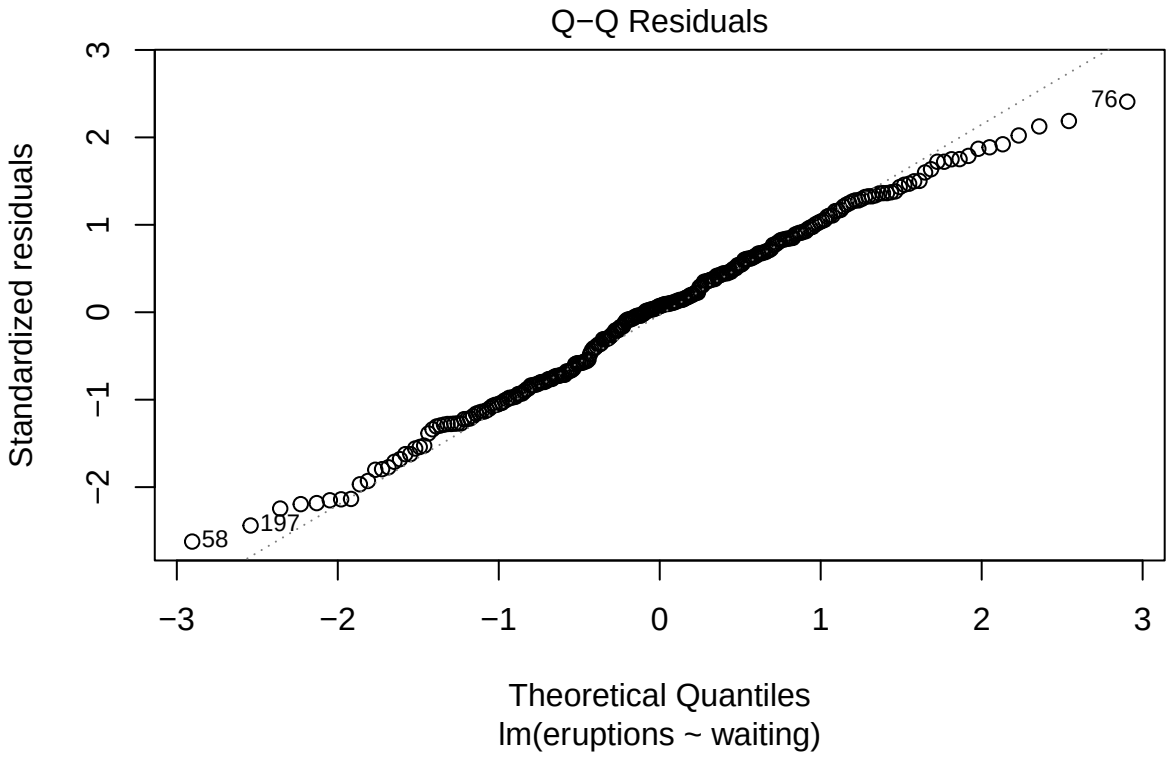


Write a sentence about what you observe

The data points have been scattered around the horizontal axis indicating the assumptions of constant variance of residuals and independence of errors may hold. It doesn't seem to have a clear pattern or trend.

Normal probability plot:

```
plot(model,2)
```



Write a sentence about what you observe and what this plot says about the residuals.

The observed residuals closely follow a diagonal line, indicating that the residuals are normally distributed and since the residuals are normally distributed i can conclude that the plot shows a linear pattern.

2. (10pts) Interpretation of coefficients Let's use again the dataset faithful.

a. Run the regression model of eruptions on waiting times.

```
model <- lm(eruptions ~ waiting, data = faithful)
summary(model)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

Interpret the meaning of the coefficient β_1.

For every each increase in the value of waiting time in the data faithful, the eruption duration between waiting times is said to increase by 0.075628 times according to the regression model.

b. Let's re-scale the waiting times by subtracting its mean and dividing by its standard deviation.

```
wait_time_scaled <- (faithful$waiting - mean(faithful$waiting))/sd(faithful$waiting)
print(wait_time_scaled[1:100])
```

```
##   [1]  0.596024774 -1.242890136  0.228241792 -0.654437365  1.037364352
##   [6] -1.169333540  1.258034141  1.037364352 -1.463559925  1.037364352
##  [11] -1.242890136  0.963807756  0.522468177 -1.757786311  0.890251159
##  [16] -1.390003329 -0.654437365  0.963807756 -1.390003329  0.596024774
##  [21] -1.463559925 -1.757786311  0.522468177 -0.139541190  0.228241792
##  [26]  0.890251159 -1.169333540  0.375354985  0.522468177  0.596024774
##  [31]  0.154685195  0.448911581 -0.360210979  0.669581370  0.228241792
##  [36] -1.390003329 -1.684229714  0.669581370 -0.875107154  1.405147334
##  [41]  0.669581370 -0.948663750  0.963807756 -0.948663750  0.154685195
##  [46]  0.890251159 -0.507324172 -1.316446732  0.816694563 -0.875107154
##  [51]  0.301798388  1.405147334 -1.242890136  0.669581370 -1.242890136
##  [56]  0.890251159  0.007572003 -0.507324172  0.448911581  0.743137966
##  [61] -0.875107154  0.963807756 -1.684229714  0.816694563 -0.801550558
##  [66]  1.552260527  0.522468177  0.522468177 -0.433767576  0.154685195
##  [71]  0.816694563 -1.095776943  0.596024774  0.007572003 -0.654437365
##  [76]  0.375354985 -0.801550558  0.522468177  0.375354985  0.890251159
##  [81]  0.301798388  0.816694563 -0.065984594 -0.433767576  0.154685195
##  [86]  1.258034141  0.375354985  0.669581370 -1.684229714  1.110920948
##  [91] -0.801550558  1.405147334 -1.537116522  0.522468177 -0.580880769
##  [96]  0.081128599  0.963807756  0.301798388 -1.463559925  0.816694563
```

c. Run the regression model of eruptions on wait_time_scaled and interpret the meaning of the coefficient β_1.

```
model_scaled <- lm(eruptions ~ wait_time_scaled, data = faithful)
summary(model_scaled)
```

```
##
## Call:
## lm(formula = eruptions ~ wait_time_scaled, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.48778    0.03011  115.85   <2e-16 ***
## wait_time_scaled  1.02816    0.03016   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

The value of β_1 gives a value of 1.02816 of the coefficient wait_time_scaled. For every increase in the value of wait_time_scaled, the value of eruptions increase by β_1 times.

3. OLS (10pts) In the following two cases, find the OLS estimate of the beta parameter and the unbiased estimate for the variance. How many degrees of freedom does it have. (Hint: how many betas are being estimated?)

a) $Y_i = \beta_0 + \epsilon_i$

In simple linear regression, the linear regression model without the predictors - only the intercepts, the OLS estimate for β_0 is the sample mean of Y. Therefore, the OLS estimate is: $\beta_0 = \overline{Y}$ for the error term ϵ_i, the unbiased estimate for the variance (σ^2) is given by $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n-1}$ In this case, there is only one parameter being estimated (β_0), so the degrees of freedom for the estimate is n-1

b) $Y_i = \beta_1 x_i + \epsilon_i$

The OLS estimate for β_1 is $\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$ and the unbiased estimate for the variance is $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n-2}$. In this case, there are two parameters that are being estimated. Even though β_0 is not present in the equation explicitly, it is still estimated by the model. Hence the β_0 will be the intercept and β_1 will be the coefficient of x_i. It has n-2 degrees of freedom because of two beta parameters are being estimated.

4. (10pts) Summary Statistics Can Hide Important Relationships. Call the dataset in R called Anscombe (1973). The purpose of this exercise is to demonstrate how plotting data can reveal important information that is not evident in numerical summary statistics.

a. Compute the averages and standard deviations of each column of data. Check that the averages and standard deviations of each of the x columns are the same, within two decimal places, and similarly for each of the y columns.

```
data("anscombe")
head(anscombe)
```

```
##   x1 x2 x3 x4   y1   y2    y3   y4
## 1 10 10 10  8 8.04 9.14  7.46 6.58
## 2  8  8  8  8 6.95 8.14  6.77 5.76
## 3 13 13 13  8 7.58 8.74 12.74 7.71
## 4  9  9  9  8 8.81 8.77  7.11 8.84
## 5 11 11 11  8 8.33 9.26  7.81 8.47
## 6 14 14 14  8 9.96 8.10  8.84 7.04
```

```
print(mean(anscombe$x1))
```

```
## [1] 9
```

```
print(mean(anscombe$x2))
```

```
## [1] 9
```

```
print(mean(anscombe$x3))
```

```
## [1] 9
```

```
print(mean(anscombe$x4))
```

```
## [1] 9
```

```
print(mean(anscombe$y1))
```

```
## [1] 7.500909
```

```
print(mean(anscombe$y2))
```

```
## [1] 7.500909
```

```r
print(mean(anscombe$y3))
```

```
## [1] 7.5
```

```r
print(mean(anscombe$y4))
```

```
## [1] 7.500909
```

```r
print(sd(anscombe$x1))
```

```
## [1] 3.316625
```

```r
print(sd(anscombe$x2))
```

```
## [1] 3.316625
```

```r
print(sd(anscombe$x3))
```

```
## [1] 3.316625
```

```r
print(sd(anscombe$x4))
```

```
## [1] 3.316625
```

```r
print(sd(anscombe$y1))
```

```
## [1] 2.031568
```

```r
print(sd(anscombe$y2))
```

```
## [1] 2.031657
```

```r
print(sd(anscombe$y3))
```

```
## [1] 2.030424
```

```r
print(sd(anscombe$y4))
```

```
## [1] 2.030579
```

b. Run four regressions, (i) y_1 on x_1, (ii) y_2 on x_2, (iii) y_3 on x_3, and (iv) y_4 on x_4.

```r
model_y1_x1 <- lm(y1 ~ x1, data = anscombe)
model_y2_x2 <- lm(y2 ~ x2, data = anscombe)
model_y3_x3 <- lm(y3 ~ x3, data = anscombe)
model_y4_x4 <- lm(y4 ~ x4, data = anscombe)
summary(model_y1_x1)
```

```
##
## Call:
## lm(formula = y1 ~ x1, data = anscombe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## x1            0.5001     0.1179   4.241  0.00217 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```
summary(model_y2_x2)
```

```
##
## Call:
## lm(formula = y2 ~ x2, data = anscombe)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x2             0.500      0.118   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```
summary(model_y3_x3)
```

```
##
## Call:
## lm(formula = y3 ~ x3, data = anscombe)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x3            0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

```
summary(model_y4_x4)
```

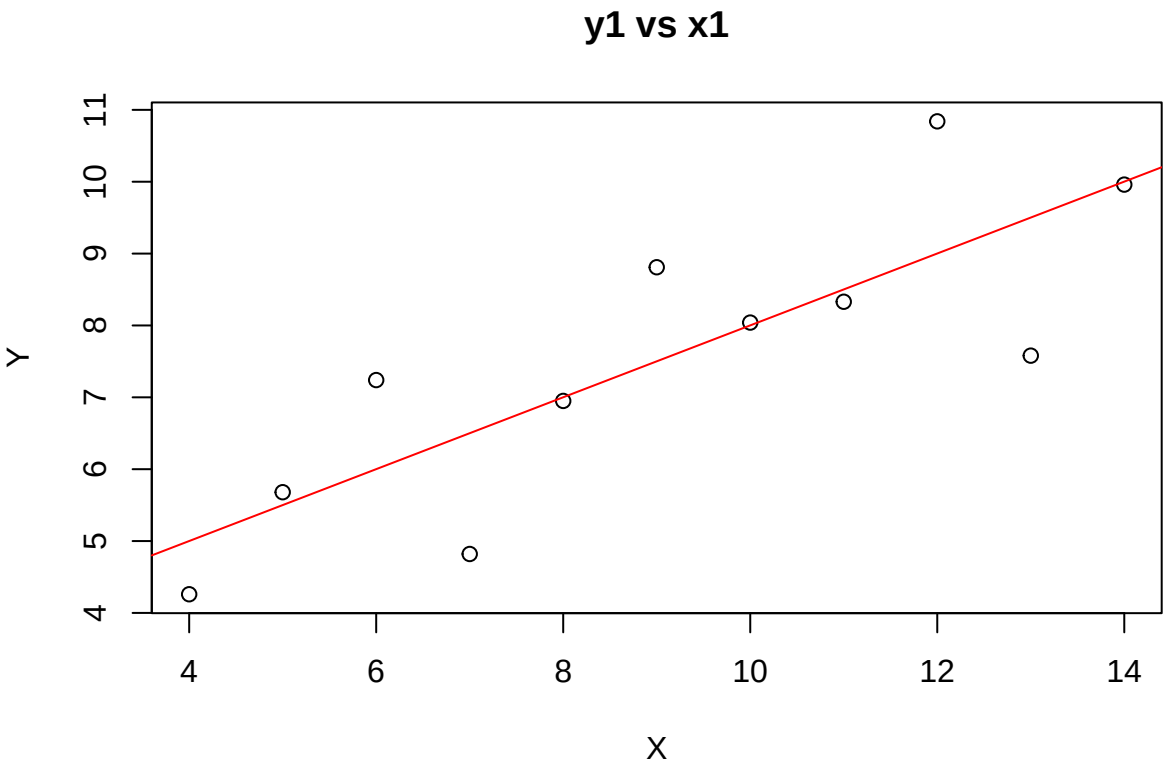
```
##
## Call:
## lm(formula = y4 ~ x4, data = anscombe)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## x4            0.4999     0.1178   4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

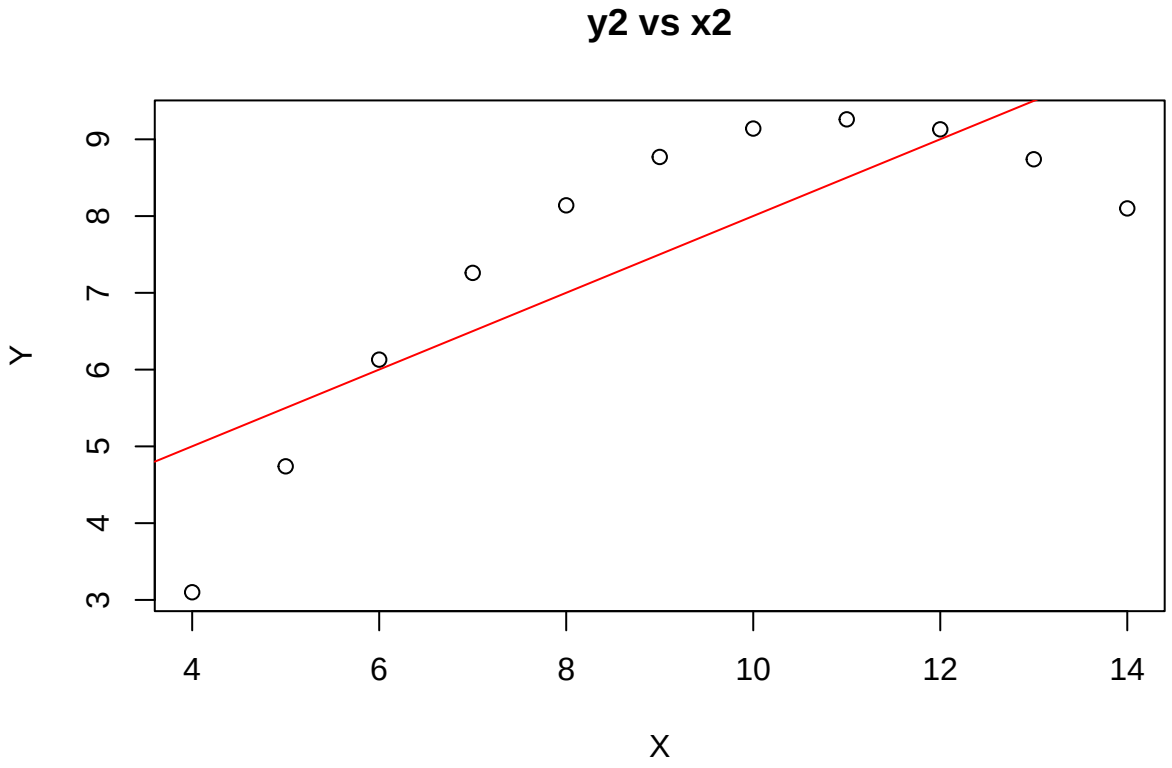
Verify, for each of the four regressions fits, that $b_0 \sim 3.0$, $b_1 \sim 0.5$, $s \sim 1.237$, and $R^2 \sim 0.666$, within two decimal places.

c. Produce scatter plots for each of the four regression models that you fit in part (b) and add the regression line in red.

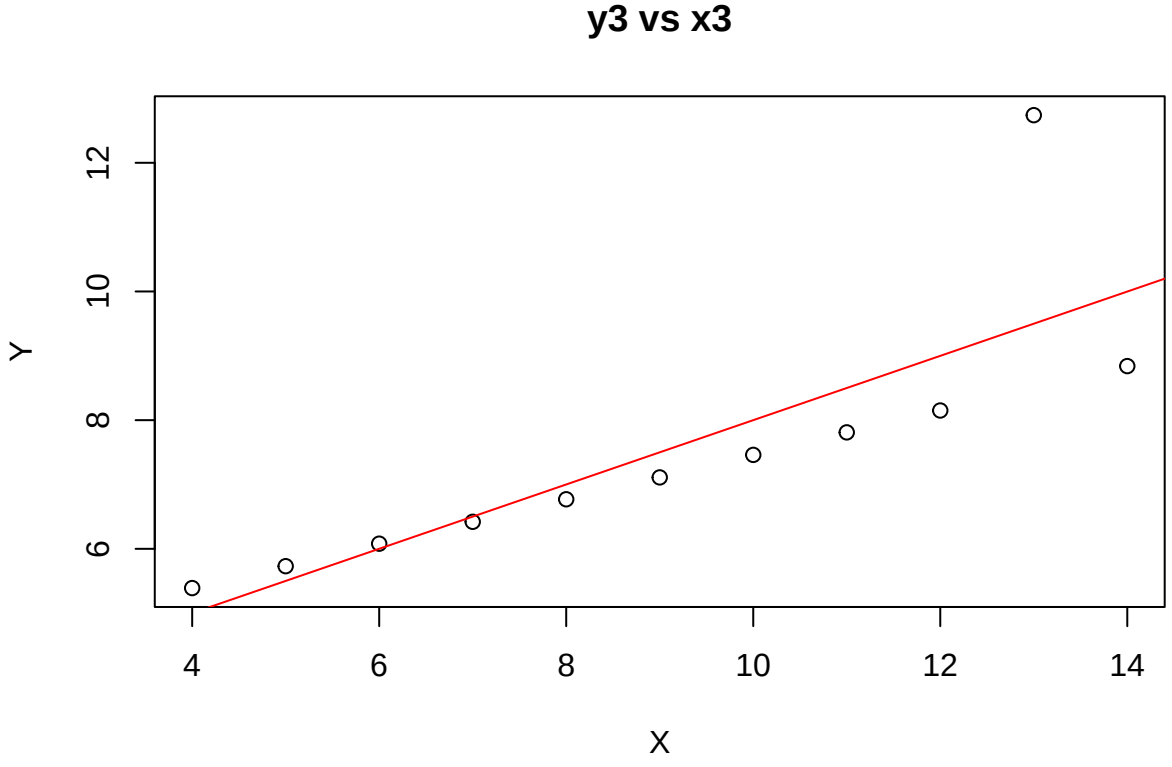
```
plot_with_regression_line <- function(x, y, model, title) {
  plot(x, y, main = title, xlab = "X", ylab = "Y")
  abline(model, col = "red")
}
plot_with_regression_line(anscombe$x1, anscombe$y1, model_y1_x1, "y1 vs x1")
```



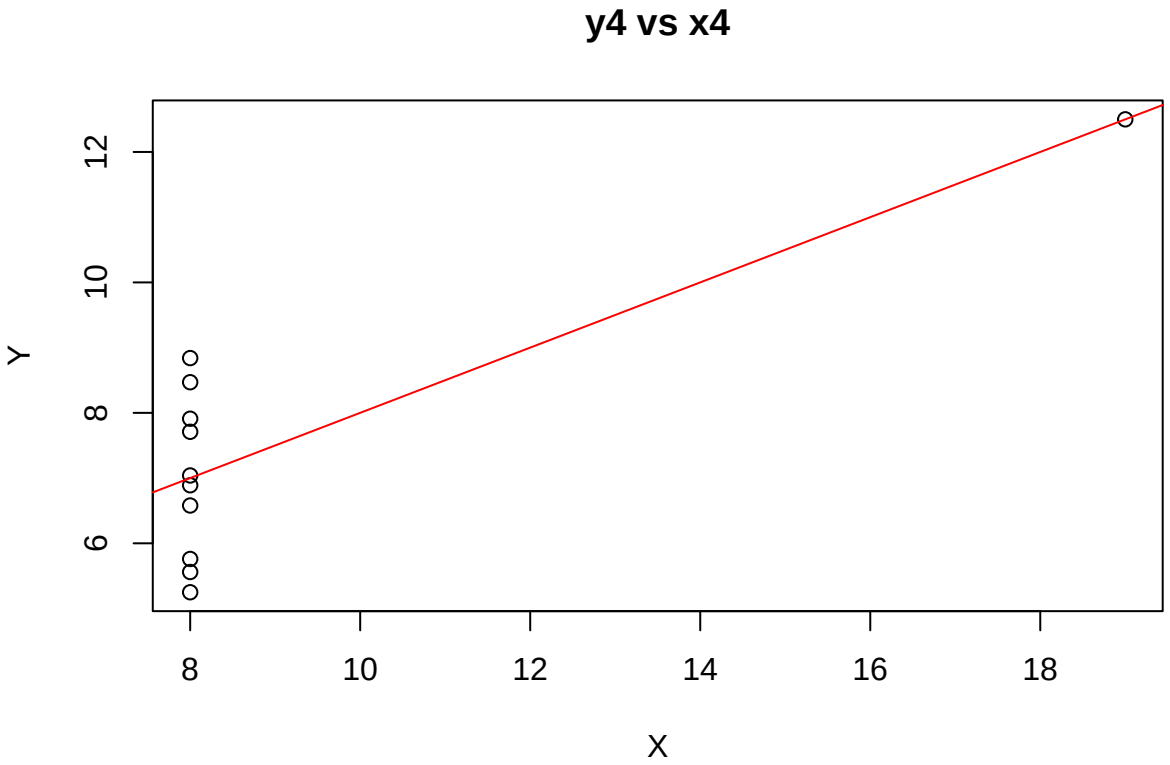

```
plot_with_regression_line(anscombe$x2, anscombe$y2, model_y2_x2, "y2 vs x2")
```




```
plot_with_regression_line(anscombe$x3, anscombe$y3, model_y3_x3, "y3 vs x3")
```




```
plot_with_regression_line(anscombe$x4, anscombe$y4, model_y4_x4, "y4 vs x4")
```



d. Discuss the fact that the fitted regression models produced in part
   (b) imply that the four datasets are similar, though the four scatter plots produced in part (c) yield a dramatically different story.

5. (5 pts) Effects of an Unusual Point. You are analyzing a data set of size $n = 100$. You have just performed a regression analysis using one predictor variable and notice that the residual for the 10th observation is unusually large.

a. Suppose that, in fact, it turns out that $e_{10} = 8s$. What percentage of the sum of squares errors, SSE, is due to the 10th observation?

b. Suppose that $e_{10} = 4s$. What percentage of the sum of squares errors, SSE, is due to the 10th observation?

c. Suppose that you reduce the dataset to size $n = 20$. After running the regression, it turns out that we still have $e_{10} = 4s$. What percentage of the error sum of squares, Error SS, is due to the 10th observation?