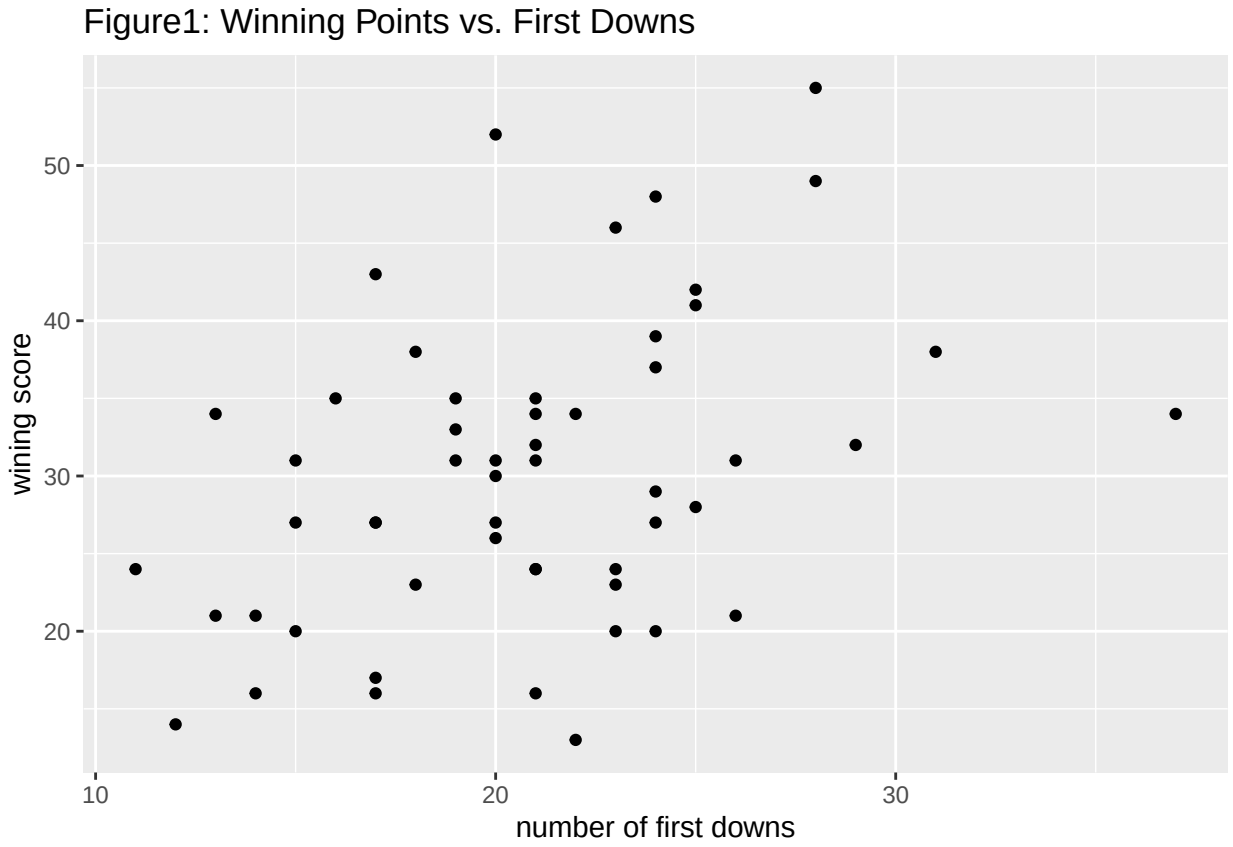
STA363Lab3SecA - Rickey Huang & Phoebe Yan

Phoebe Yan & Rickey Huang

2/11/2021

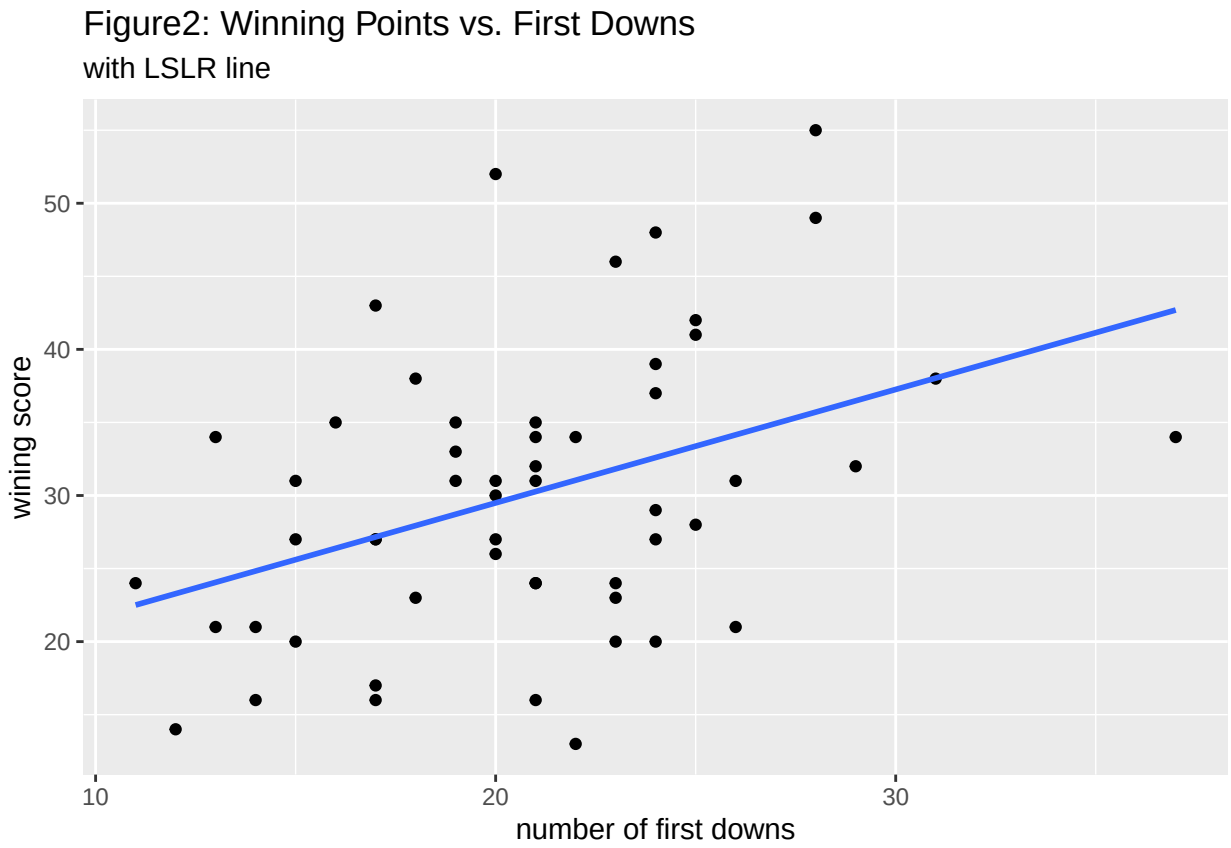
Question 1

Using ggplot2, make a plot to explore the relationship between these two variables of interest. Note: For every plot you do from now on in this course, whenever I tell you to make a plot, that means a plot with labelled axes, and a title like "Figure 1: Winning Points vs. First Downs".



Question 2

Add a fitted LSLR line to your graph from Question 1. Hint: There is code on how to do this in Lab 2.

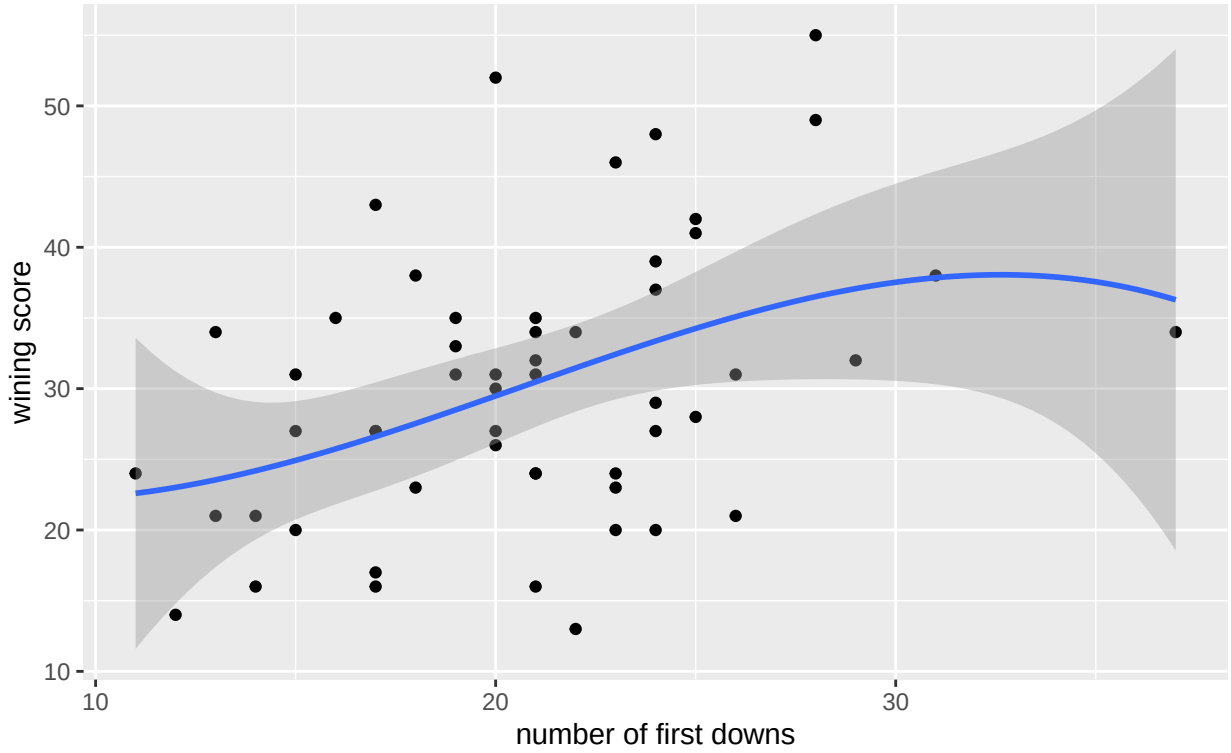


Question 3

Add a fitted third order polynomial to your graph from Question 1. Hint: There is code on how to do this in Lab 2.

Figure3: Winning Points vs. First Downs

with third order polynomial



Question 4

Which of the two model choices (LSLR regression or third order polynomial regression) is a more flexible model choice?

The third order polynomial regression is a more flexible model choice as it follows some small trend in the data.

Question 5

If we were to choose LSLR for our model, do you think it is more likely that we would under-fit the data or over-fit the data? Explain.

If we were to choose LSLR for the model, I think it is more likely to under-fit the data since it does not capture little trends in the data

Question 6

Write down the form of f(X) for both the LSLR model and the polynomial regression model. Hint: Writing out f(X) involves symbols, not numbers.

for the LSLR model: $f(X) = \beta_0 + \beta_1 X$, where $f(X)$=WiningScore, and X=number of first downs. for the polynomial regression model: $f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$, where $f(X)$=WiningScore, and X=number of first downs.

Question 7

Write down $f(\hat{X})$ for the LSLR model. Hint: Writing out $f(\hat{X})$ involves numeric estimates for the parameters, not symbols.

$\widehat{f(X)} = 13.9609 + 0.7766X$, where $f(X)$=WiningScore, X=number of first downs.

Question 8

Write down $\hat{f}(X)$ for the polynomial regression model. Hint: When you fit a polynomial model in R, you need to use code like lm(Y ~ X + I(X^2) + I(X^3), data =).

$\widehat{f(X)} = 29.789465 - 1.940728X + 0.142239X^2 - 0.002298X^3$, where $\widehat{f(X)}$=WiningScore, X=number of first downs.

Question 9

Using Model 1 (the LSLR model), make a prediction for the winning score for the 2021 SuperBowl. Show your steps (and don't use the predict function.) State the prediction, and the value of the residual for the 2021 SuperBowl.

$Y_{m_1} = 13.9609 + 0.7766 \times 26 = 34.1525$ $res_2 1 = 31 - Y_{m_1} = -3.1525$ The prediction for the winning score for the 2021 SuperBowl by model 2 is 34.1525. The residual of model 1 is -3.1525.

Question 10

Repeat the same steps, but for Model 2 (the polynomial regression model).

$Y_{m_2} = 29.789465 - 1.940728 \times 26 + 0.142239 \times (26^2) - 0.002298 \times (26^3)$

$res_{m_2} = 31 - Y_{m_2}$

The prediction for the winning score for the 2021 SuperBowl by model 2 is 35.094453. The residual is -4.094453

Question 11

Based on what we have computed so far, which of the two models more accurately predicted the winning score of SuperBowl 2021?

Since the residual of model 1 is smaller, the LSLR model more accurately predicted the winning score of SuperBowl 2021.

Question 12

Using an 80/20 split, determine which rows of data are going to be used for the CV training data set. Print out the row numbers that you have chosen. Show the code you used, and annotate it. Note: Annotate means using a line in your chunk with a # in the front to add a brief comment about what each line of your code does. For instance, # Set a random seed.

```
#set a random seed
set.seed(300)
#sample the rows
rowsCVtraining <- sample(1:54, 54*0.8, replace=FALSE)
rowsCVtraining
```

```
##  [1] 14 42  2 25 28 21 41 19 48 16 13 12 39 43 47 46  5  1 22 38 17 32 11  8 50
## [26]  7 30  9 33 51 34 52 26 15  3 29 31  6 44 24 53 27 35
```

Question 13

Explain why it is important to use random sampling to determine which rows in the original training data end up in the CV training set.

Because we want to use our training data set to fit our model which will be applied to the whole data set, both the training and test data set, we want the training data to simulate the trend of the complete data to predict. Thus, we want our training data to be randomly chosen from the data set, in order to avoid some problem that there might be some clusters of data with some special characteristics.

Question 14

Now, actually create the CV test and CV training data sets based on the rows you selected in the Question 12. State the dimensions of the two data sets. Also, show the code you used to create the data sets, and annotate it.

```
#create the cv training data and return the dimension of the training set
CVtraining <- SuperBowl[rowsCVtraining, ]
dim(CVtraining)
```

```
## [1] 43 38
```

```
#create the cv test data and return the dimension of the test set
CVtest <- SuperBowl[-rowsCVtraining, ]
dim(CVtest)
```

```
## [1] 11 38
```

The CV training data set has 43 rows with 38 columns, The CV test data set has 11 rows with 38 columns.

Question 15

Train both a LSLR model (Model 1) and a third order polynomial model (Model 2). Show the code you used to do so, and annotate it. Write out both trained models (regression lines). Hint: (1) Remember this means using the numeric estimates for the parameters. (2) Training a model does NOT mean drawing a graph. Your answer should be an equation.

```
#train a LSLR model
m1_train <- lm(Winner_Pts ~ Winner_FirstDowns, data = CVtraining)
summary(m1_train)
```

```
##
## Call:
## lm(formula = Winner_Pts ~ Winner_FirstDowns, data = CVtraining)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.793  -6.071  -1.349   5.096  23.318
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.5716     7.0202   1.079  0.28710
## Winner_FirstDowns   1.0555     0.3322   3.177  0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.284 on 41 degrees of freedom
## Multiple R-squared:  0.1976, Adjusted R-squared:  0.178
## F-statistic:  10.1 on 1 and 41 DF,  p-value: 0.002824
```

```
#train a third order polynomial model
m2_train <- lm(Winner_Pts ~ Winner_FirstDowns + I(Winner_FirstDowns^2) + I(Winner_FirstDowns^3), data =
summary(m2_train)
```

```
##
## Call:
## lm(formula = Winner_Pts ~ Winner_FirstDowns + I(Winner_FirstDowns^2) +
##     I(Winner_FirstDowns^3), data = CVtraining)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.514  -5.210  -1.465   6.289  23.535
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -81.57722   97.01834  -0.841    0.406
## Winner_FirstDowns       16.04965   15.31987   1.048    0.301
## I(Winner_FirstDowns^2)  -0.80112    0.78046  -1.026    0.311
## I(Winner_FirstDowns^3)   0.01369    0.01287   1.064    0.294
##
## Residual standard error: 9.366 on 39 degrees of freedom
## Multiple R-squared:  0.2233, Adjusted R-squared:  0.1636
## F-statistic: 3.738 on 3 and 39 DF,  p-value: 0.01874
```

LSLR: $\widehat{Y} = 3.9496 + 1.2280X$, where Y=WiningScore, X=number of first downs.

Third order polynomial: $\widehat{Y} = -105.625476 + 18.94921X - 0.91915X^2 + 0.01535X^3$, where Y=WiningScore, X=number of first downs.

Question 16

Compare the trained models from Question 15 to the trained models (fitted models) you got when you used the entire training data set in Questions 7 and 8. Are the estimates of the parameters the same? Do we expect them to be?

The estimates of the parameters are not the same. We do not expect them to be the same since we are using different data set.

Question 17

Using the LSLR model trained on the CV training data, make a prediction for the first row in the CV test data set. Don't use predict; compute it mathematically.

$\widehat{Y_{train_1}} = 3.9496 + 1.2280 \times X_{train_1} = 49.3856$

Question 18

Using the LSLR model trained on the CV training data, make predictions for all the rows of the CV test data set. (Now you can use predict!). Store the predictions as an object called predsLSLR. Show the code you used to do so, and annotate it. Show that the prediction you obtained in the previous question is the first element in the predsLSLR vector. Note: It is okay if the values are little different due to rounding!

```
#predict for all the rows of the CV test data set with LSLR model
predsLSLR <- predict(m1_train, newdata = CVtest)
#return the first predict result
head(predsLSLR,1)
```

```
##        4
## 46.62618
```

Question 19

Using the polynomial regression model trained on the CV training set, make predictions on the CV test data set. Store the predictions as an object called predsPoly. Show the code you used to do so, and annotate it.

```
#predict for all the rows of the CV test data set with polynomial regression model
predsPoly <- predict(m2_train, newdata = CVtest)
```

Question 20

Using the CV test data, estimate the test RSS and test MSE for both Model 1 and Model 2. State the numeric values, and also show the code you used to compute your answer.

```
#LSLR model
#obtain the RSS
rss_1 = t(CVtest$Winner_Pts - predsLSLR) %*% (CVtest$Winner_Pts - predsLSLR)
#obtain the MSE
mse_1 <- mean((CVtest$Winner_Pts - predsLSLR)^2)

#polynomial regression model
#obtain the RSS
rss_2 = t(CVtest$Winner_Pts - predsPoly) %*% (CVtest$Winner_Pts - predsPoly)
#obtain the MSE
mse_2 <- mean((CVtest$Winner_Pts - predsPoly)^2)
```

The test RSS for model 1 is 835.5017, for model 2 is 6499.753 The test MSE for model 1 is 75.9547, for model 2 is 590.8866

Question 21

Based on your results, which model has the highest predictive accuracy? Explain.

Based on our result, the LSLR model has the highest predictive accuracy, since the MSE of model 1 is the smaller which is 75.9547, and the RSS of it is also smaller than the second model, which is 835.5017.

Question 22

In this chunk you have just created, change the random seed to 497, and hit play. Does your answer to Question 21 change?

```
#set a random seed
set.seed(497)
#sample the rows
rowsCVtraining <- sample(1:54, 54*0.8, replace=FALSE)
#create the cv training data
CVtraining <- SuperBowl[rowsCVtraining, ]
#create the cv test data
CVtest <- SuperBowl[-rowsCVtraining, ]
#train a LSLR model
m1_train <- lm(Winner_Pts ~ Winner_FirstDowns, data = CVtraining)
#train a third order polynomial model
m2_train <- lm(Winner_Pts ~ Winner_FirstDowns + I(Winner_FirstDowns^2) + I(Winner_FirstDowns^3), data =
#predict for all the rows of the CV test data set with LSLR model
predsLSLR <- predict(m1_train, newdata = CVtest)
#predict for all the rows of the CV test data set with polynomial regression model
predsPoly <- predict(m2_train, newdata = CVtest)
#LSLR model
```

```r
#obtain the RSS
rss_1 = t(CVtest$Winner_Pts - predsLSLR) %*% (CVtest$Winner_Pts - predsLSLR)
#obtain the MSE
mse_1 <- mean((CVtest$Winner_Pts - predsLSLR)^2)

#polynomial regression model
#obtain the RSS
rss_2 = t(CVtest$Winner_Pts - predsPoly) %*% (CVtest$Winner_Pts - predsPoly)
#obtain the MSE
mse_2 <- mean((CVtest$Winner_Pts - predsPoly)^2)
```

After changing the random seed to 497, the test RSS for model 1 is 865.0746, for model 2 is 935.9522; the test MSE for model 1 is 78.6431, for model 2 is 85.0866.

Our answer to the test RSS and test MSE for model 1 and 2 all changed from the question 21.