STAT 551 Linear Regression Assignment

Yuchi Hu

February 3, 2019

Part 1

Part 1.1

We make a scatterplot of Y vs. *numvar1*.

```
# Load LinRegData.csv
mydata <- read.csv('C:/Users/George/Desktop/PredictiveAnalyticsI/LinRegData.csv', header=T)

theme_update(plot.title=element_text(hjust=0.5))
# Y vs. numvar1
ggplot(data=mydata, aes(y=Y, x=numvar1)) + geom_point() + labs(title='Y vs. numvar1')
```

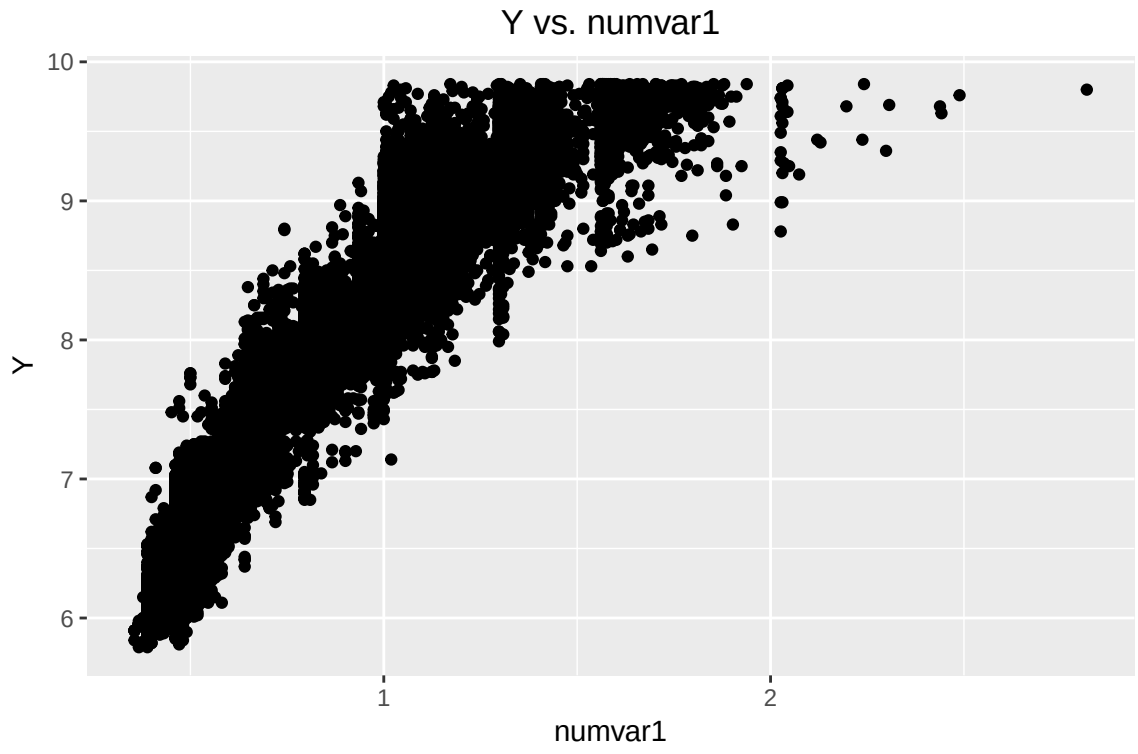

Figure 1: Y vs. numvar1

From **Figure 1**, the variables seem to be associated. The association appears to be linear.

The model $Y = \beta_0 + \beta_1 * numvar1 + \epsilon$ is a simple linear regression model. It is appropriate in this case since we have only one predictor (*numvar1*), which appears to be linearly associated with the response (Y).

Part 1.2

We fit a simple linear regression model (model1) with Y as the response and *numvar1* as the predictor.

```
# Simple linear regression model
model1 <- lm(Y ~ numvar1, data=mydata)
```

The summary of model1 is below:

```
# Model summary
summary(model1)
```

```
##
## Call:
## lm(formula = Y ~ numvar1, data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0135 -0.1934  0.0149  0.2016  1.4574
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.259101   0.004573  1149.9   <2e-16 ***
## numvar1     3.035931   0.005130   591.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3249 on 39998 degrees of freedom
## Multiple R-squared:  0.8975, Adjusted R-squared:  0.8975
## F-statistic: 3.503e+05 on 1 and 39998 DF,  p-value: < 2.2e-16
```

Two parameters are estimated in model1. The parameter for the intercept estimates an average value of 5.259 units for Y when *numvar1* $= 0$. The parameter for *numvar1* estimates an average increase of 3.036 units in Y for each unit increase in *numvar1*.

The R^2 of 0.8975 means that model1 explains 89.75% of the variability in Y.

Part 2

Part 2.1

We make a new variable *numvar1new* by raising *numvar1* to the $(1/2.25)^{th}$ power. Then, we fit a simple linear regression model (model2) with Y as the response and *numvar1new* as the predictor. The summary of model2 is below:

```
# Raise numvar1 to the (1/2.25)th power
mydata$numvar1new <- mydata$numvar1^(1/2.25)

# New simple linear regression model
model2 <- lm(Y ~ numvar1new, data=mydata)
# Model summary
summary(model2)
```

```
##
## Call:
## lm(formula = Y ~ numvar1new, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.33834 -0.16978  0.00022  0.17189  1.36788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.979233   0.008311   238.1   <2e-16 ***
## numvar1new  6.410648   0.009043   708.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2755 on 39998 degrees of freedom
## Multiple R-squared:  0.9263, Adjusted R-squared:  0.9263
## F-statistic: 5.025e+05 on 1 and 39998 DF,  p-value: < 2.2e-16
```

The parameter for the intercept estimates an average value of 1.979 units for Y when $numvar1new = 0$. The parameter for *numvar1new* estimates an average increase of 6.411 units in Y for each unit increase in *numvar1new*.

The R^2 of 0.9263 means that model2 explains 92.63% of the variability in Y. Compared to the R^2 of model1 (0.8975), the R^2 of model2 is larger, so model2 is a better fit to the data.

Part 3

We make three scatterplots of Y vs. *numvar1new*, each colored by *catvar1*, *catvar2*, or *catvar3*.

```
# Y vs. numvar1 (Colored by catvar1)
p1 <- ggplot(data=mydata, aes(y=Y, x=numvar1new, color=catvar1)) + geom_point() +
  labs(title='Y vs. numvar1new (Colored by catvar1)')
# Y vs. numvar1 (Colored by catvar2)
p2 <- ggplot(data=mydata, aes(y=Y, x=numvar1new, color=catvar2)) + geom_point() +
  labs(title='Y vs. numvar1new (Colored by catvar2)')
# Y vs. numvar1 (Colored by catvar3)
p3 <- ggplot(data=mydata, aes(y=Y, x=numvar1new, color=catvar3)) + geom_point() +
  labs(title='Y vs. numvar1new (Colored by catvar3)')

# Combine the plots into one graph
grid.arrange(p1, p2, p3, ncol=2)
```

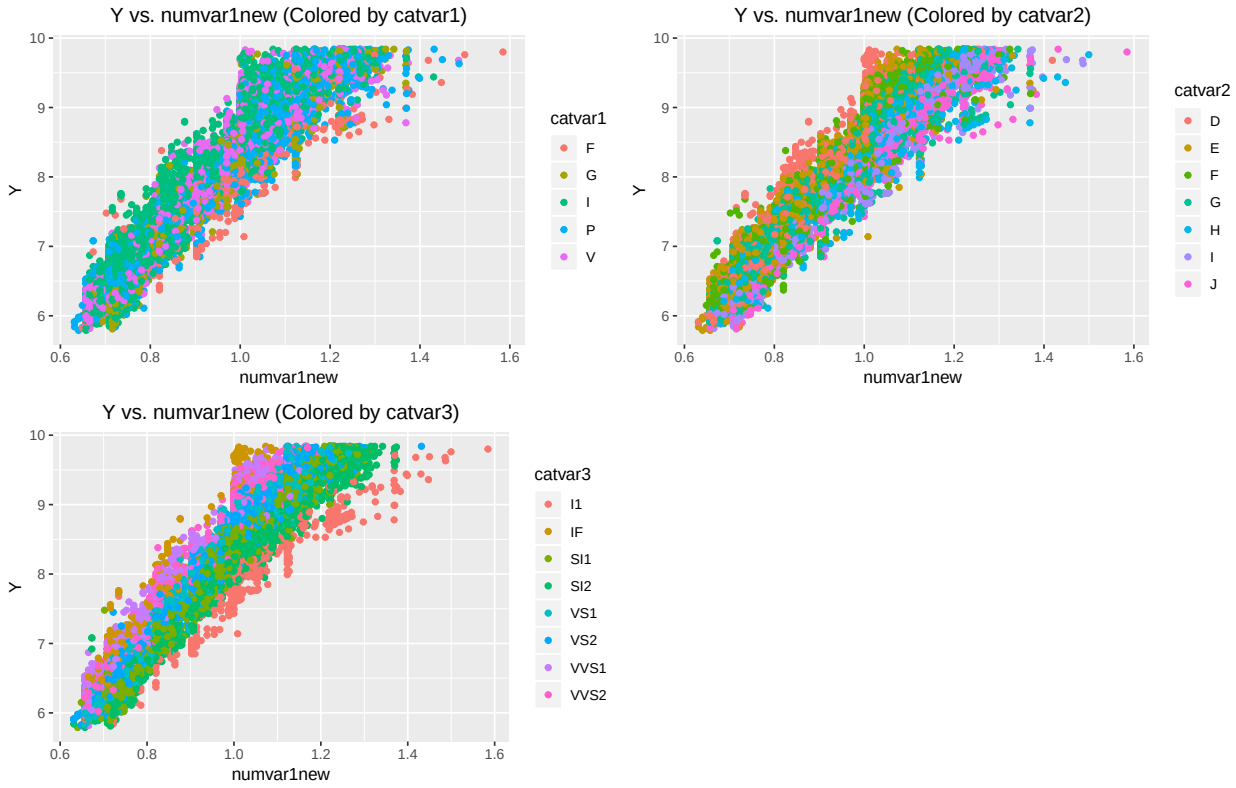

Figure 2: Y vs. numvar1new (Colored by catvar1, catvar2, or catvar3)

From **Figure 2**, in the top left plot (colored by *catvar1*), only the points with "F" values for *catvar1* appear to be clearly separated from the rest of the points. In the top right plot (colored by *catvar2*), the color distinction is more defined, and in the bottom left plot (colored by *catvar3*), the color distinction is even more defined with clear separation of points with different values of *catvar3*. Thus, all of the categorical variables should be included in the model (perhaps as interaction terms with *numvar1new*) since the distribution of the levels in each factor does not appear to be random.

Part 4

Similar to the transformation in part 2.1, we make a new variable *numvar2new* by raising *numvar2* to the $(1/3.75)^{th}$ power. Then, we make three new models.

```
# Raise numvar2 to the (1/3.75)th power
mydata$numvar2new <- mydata$numvar2^(1/3.75)

# Three new models
model3 <- lm(Y ~ numvar1new + catvar1 + catvar2 + catvar3, data=mydata)
model4 <- lm(Y ~ numvar1new*catvar1 + numvar1new*catvar2 + numvar1new*catvar3, data=mydata)
model5 <- lm(Y ~ numvar2new +numvar1new*catvar1 + numvar1new*catvar2 + numvar1new*catvar3, data=mydata)
```

We compare the adjusted R^2 of the five models (see table below) since not all of the models have the same number of predictors. We also compare their residual standard error (RSE).

```
# Adjusted R^2 of the models
rsq1 <- summary(model1)$adj.r.squared
rsq2 <- summary(model2)$adj.r.squared
rsq3 <- summary(model3)$adj.r.squared
rsq4 <- summary(model4)$adj.r.squared
rsq5 <- summary(model5)$adj.r.squared
# RSE of the models
rse1 <- summary(model1)$sigma
rse2 <- summary(model2)$sigma
rse3 <- summary(model3)$sigma
rse4 <- summary(model4)$sigma
rse5 <- summary(model5)$sigma

# Columns for table
models <- c('model1', 'model2', 'model3', 'model4', 'model5')
rsq <- round(c(rsq1, rsq2, rsq3, rsq4, rsq5), 5)
rse <- round(c(rse1, rse2, rse3, rse4, rse5), 5)

# Table of adjusted R^2
dt <- cbind('Model'=models, 'Adjusted R-Squared'=rsq, 'RSE'=rse)
kable(dt, align='c', caption='Adjusted R-Squared and RSE Comparison')
```

Table 1: Adjusted R-Squared and RSE Comparison

Model	Adjusted R-Squared	RSE
model1	0.89751	0.32487
model2	0.92627	0.27554
model3	0.97638	0.15597
model4	0.9803	0.14244
model5	0.98038	0.14213

From **Table 1**, we can see that model5 has the highest adjusted R^2 and lowest RSE; thus, model5 is the best fit to the data.

Part 5

In each of 1000 simulations, we create a dataset of 100 random X and Y points, and we create a linear model on these points. We save the values of β_1 and create a histogram of these values.

```
# Number of simulations
B <- 1000
# Initialize vector for b1
b1 <- rep(NA, B)

for(i in 1:B){
  set.seed(i)
  X <- rnorm(100, 20, 10)  # X values
  Y <- rnorm(100, 70, 5)  # Y values
  model <- lm(Y ~ X)  # Model
  b1[i] <- model$coefficients[2] # Add slope to vector
}

# Histogram of b1
ggplot(data=NULL, aes(x=b1)) + geom_histogram() +
  labs(title=expression(paste('Histogram of ', beta[1])), x=expression(beta[1]))
```

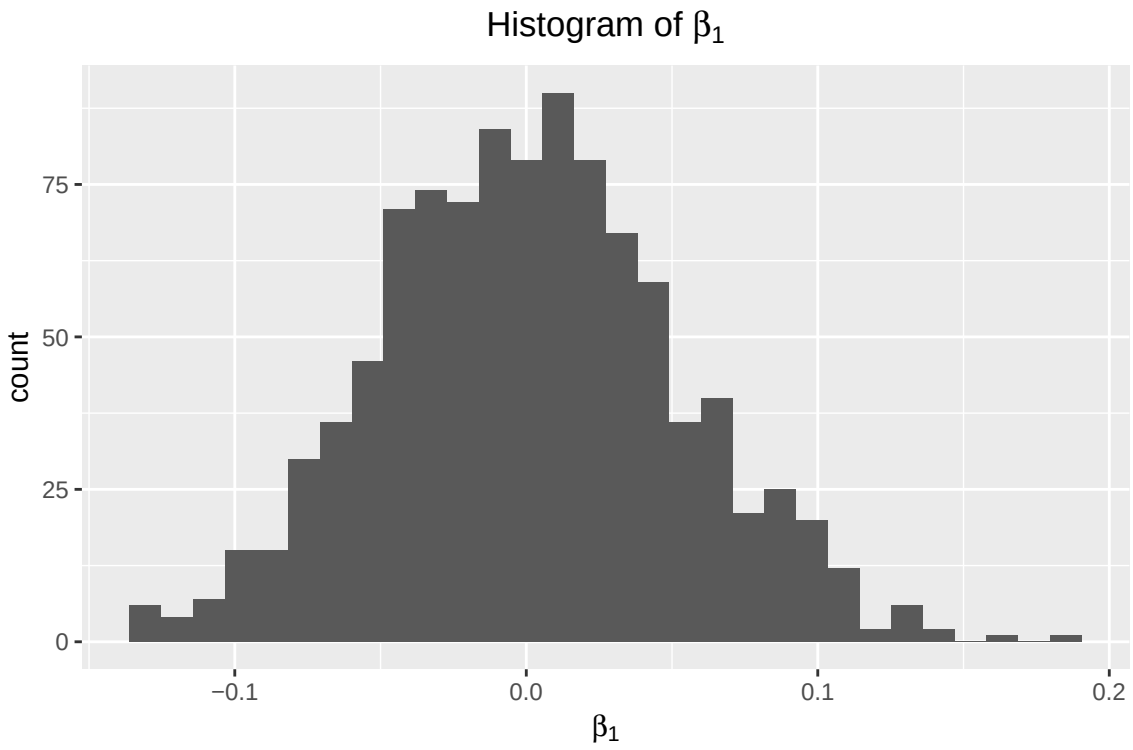

Figure 3: Histogram of β_1

From **Figure 3**, we can see that the distribution of β_1's appears to be normal and centered at 0; thus, the estimate for β_1 is 0. This value ($\beta_1 = 0$) can be interpreted as the slope being 0; that is, there is not a significant association between X and Y.