Project 4

Seamus O'Malley

This data set is about player NBA performances over the span of a season. The *Y* variable I chose was the average points per game (PTS), as I figured it was a useful variable to predict. For the X variables, I decided to do a linear combination of *X1:* 3 points made per game (X3P), *X2:* 2 points made per game (X2P), and *X3:* free throws made per game(FT). I did this because it made sense to me that your points per game encompasses those three possible scoring methods, so using all three I believe I can create a useful and accurate model.

```
##           PTS       X3P       X2P        FT
## PTS 1.0000000 0.4448207 0.9305196 0.9028465
## X3P 0.4448207 1.0000000 0.1085941 0.2482929
## X2P 0.9305196 0.1085941 1.0000000 0.8464801
## FT  0.9028465 0.2482929 0.8464801 1.0000000
```

Not a perfect correlation matrix, but decided to continue with data and see how the model does.

Splitting the data into training/testing groups (80% data in train, 20% in test).

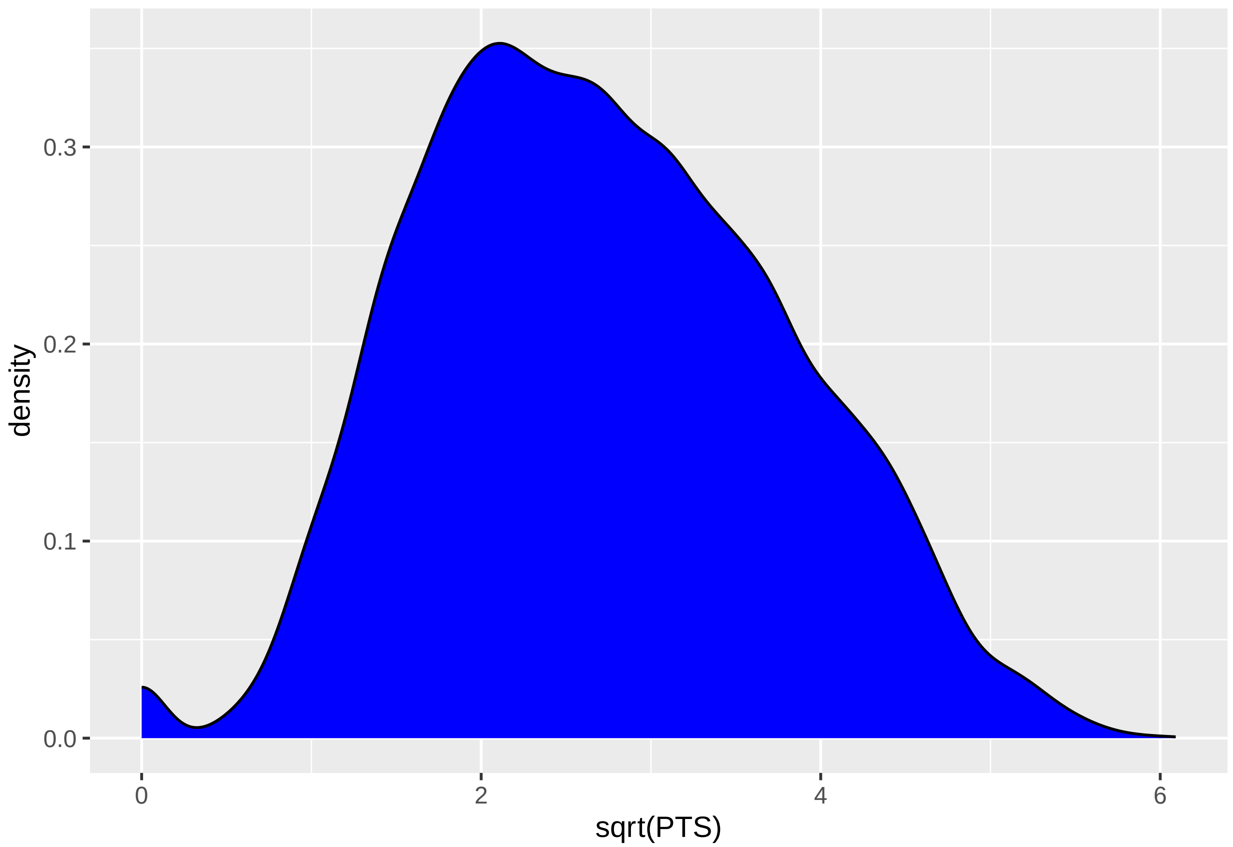
```
train = d1[row.number,]; dim(train)
```

```
## [1] 14380     4
```

```
test = d1[-row.number,]; dim(test)
```

```
## [1] 3596    4
```

Transformed the Y variable to sqrt(Y) as it brings it closer to normal. Still not a perfect normal distribution, but much better than normal Y distribution (highly skewed).



Part 2

For this model, the adjusted R-squared value is .9504, meaning the linear model accounts for about 95 percent of the variation of sqrt(PTS), and the p-value is less than 2.2e-16, showing that the overall model is statistically significant. In regards to individual components, for PTS, X3P, and X2P, the p value is less than 2e-16 for all 3, showing each variable is statistically significant to the model.

```
model1 <- lm(sqrt(PTS) ~ X2P + X3P +FT, data = train)
summary(model1)
```

```
##
## Call:
## lm(formula = sqrt(PTS) ~ X2P + X3P + FT, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26435 -0.08957  0.07331  0.15309  0.47477
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.264354   0.003423  369.36   <2e-16 ***
## X2P         0.377318   0.001777  212.30   <2e-16 ***
## X3P         0.559860   0.003276  170.89   <2e-16 ***
## FT          0.095361   0.002712   35.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2362 on 14376 degrees of freedom
## Multiple R-squared:  0.9504, Adjusted R-squared:  0.9504
## F-statistic: 9.189e+04 on 3 and 14376 DF,  p-value: < 2.2e-16
```

The following is the equation for the multiple linear regression model.

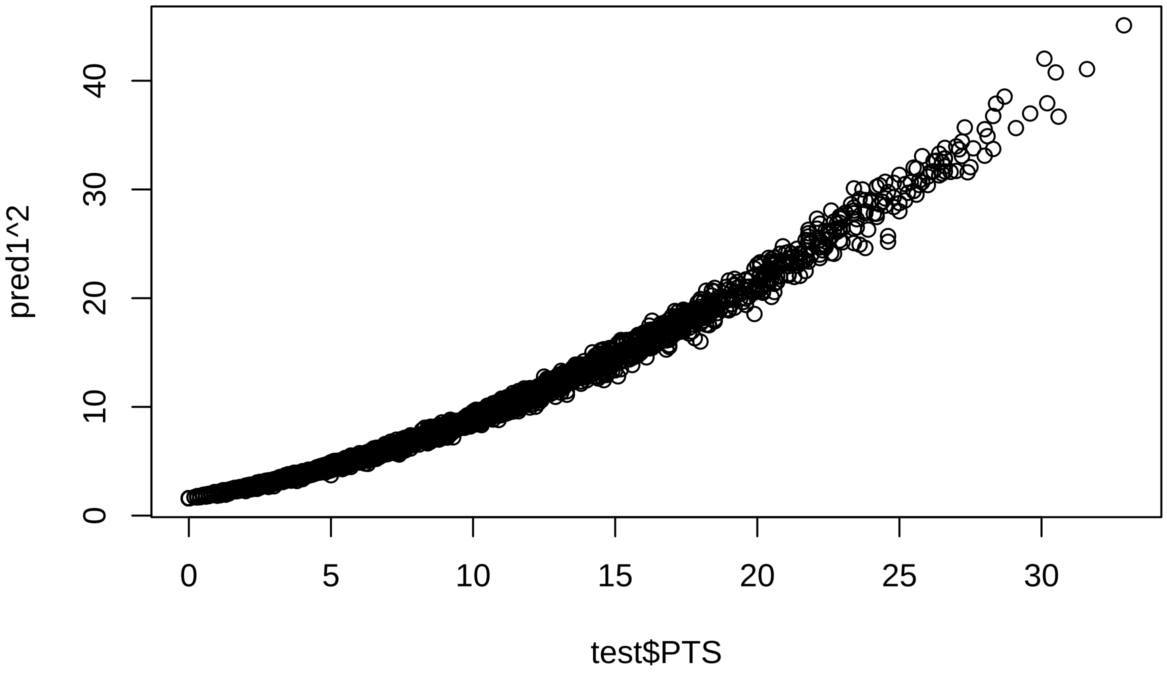
```
## [1] "yhat_i= 1.26 + 0.38 * X2P_i + 0.56 * X3P_i + 0.1 * FT_i"
```

Testing our prediction model on the testing dataset using model1. This procedure will take the x1, x2, and x3 values in the test data and get a predicted value of yhat_i using Model 1. Then, it will subtract yhat_i from the actual y_i that is given with the x1, x2, and x3 values in the test dataset. Note, I do have to square the predicted values as the linear model predicts the value of sqrt(Y), so squaring will give Y.

Based on RMSE = 1.320892, I can conclude that on an average the predicted value will be off by 1.320892 points from the actual value. I also found the mean absolute percentage error (MAPE) to measure the accuracy of the model. A lower MAPE means less error and more accuracy. In my case, MAPE=16.78%, so my training model has about 17% error in the testing dataset.

```
##     RMSE
## 1.320892
```

```
##     MAPE
## "16.78%"
```


pred1^2
test$PTS