DAT-4253 LM 8 - Prediction; Multiple Regression

AUTHOR
Aaron Younger

PUBLISHED
October 19, 2024

Business Understanding

This analysis examines the relationship between performance factors of a quarterback compared to the quarterbacks salary. The specific performance based factors used in this analysis are pass completion rate, touchdowns scored, and the quarterbacks age. The final part of the analysis identifies which players are potentially overpaid or underpaid based on the model's predictions and provides recommendations for how the model can be applied and improved for future use.

Data Understanding

Libraries

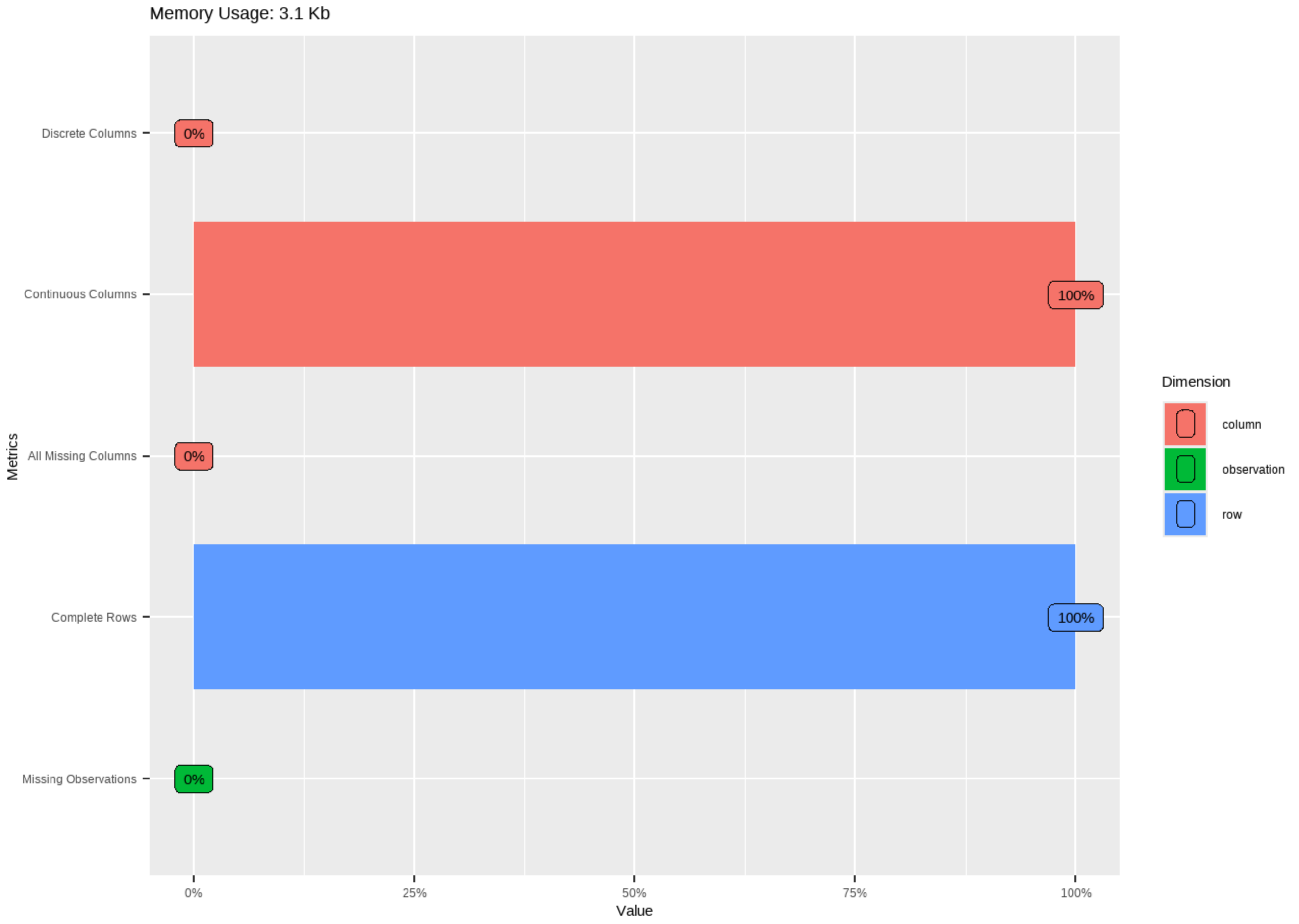
```
library(readxl)
library(tidyverse)
library(dplyr)
library(DataExplorer)
library(ggplot2)
library(caret)
library(dlookr)
library(e1071)
library(psych)
library(car)
library(stargazer)
library(tidyr)
library(purrr)
library(performance)
library(Metrics)
library(auditor)
```

Load Dataset

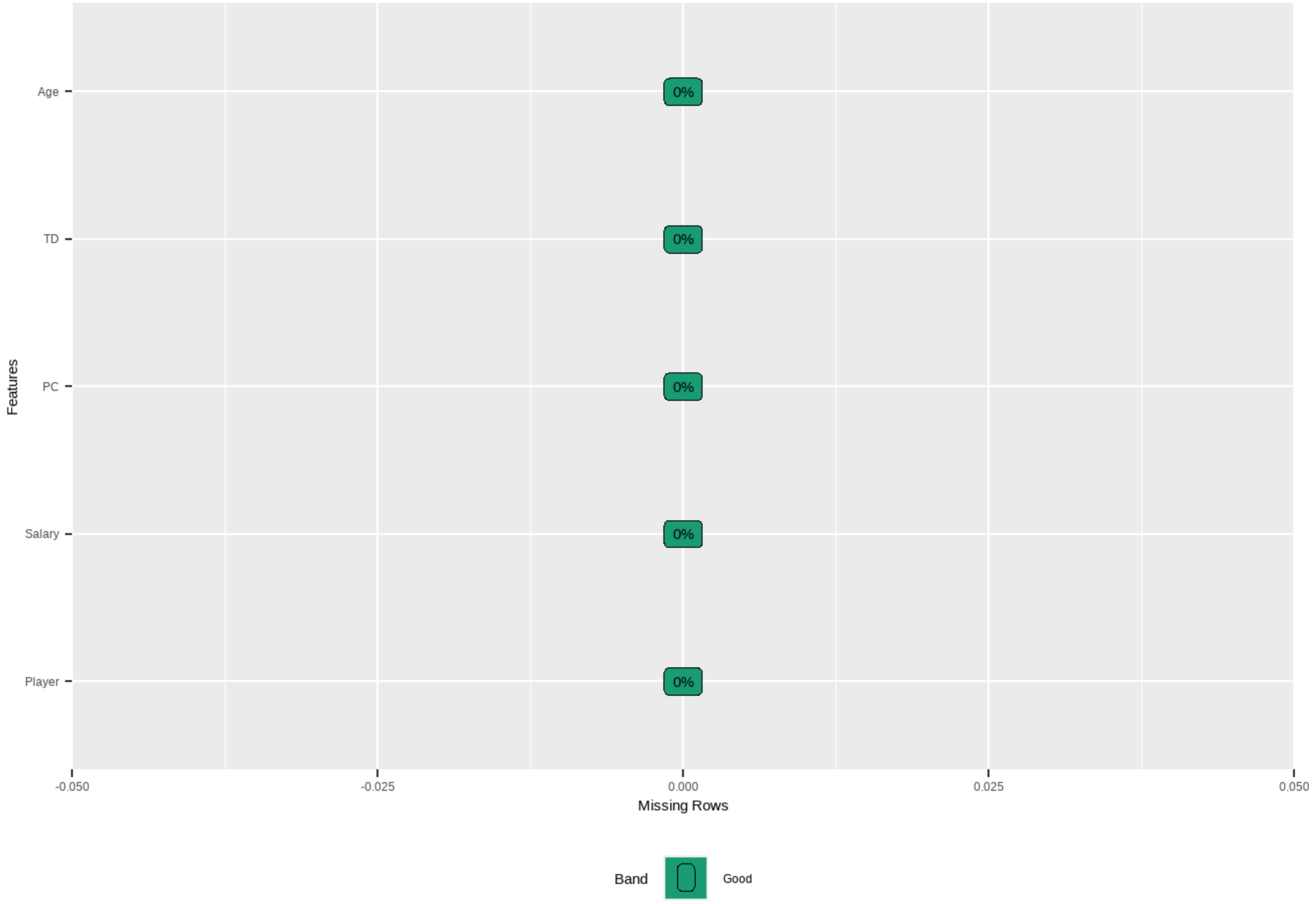
```
library(readxl)
quarterback_data <- read_excel("jaggia_ba_2e_ch07_data.xlsx",
    sheet = "Quarterbacks")
View(quarterback_data)

quarterback_data %>% str()
```

```
tibble [32 × 5] (S3: tbl_df/tbl/data.frame)
 $ Player: num [1:32] 1 2 3 4 5 6 7 8 9 10 ...
 $ Salary: num [1:32] 25.6 22 20.5 19 17 ...
 $ PC    : num [1:32] 65.2 60.5 62.3 66.1 67.9 55 68.8 70.6 60.3 68.4 ...
 $ TD    : num [1:32] 28 27 27 26 29 16 33 34 22 33 ...
 $ Age   : num [1:32] 27 26 28 38 28 27 33 30 33 40 ...
```

```
quarterback_data %>% plot_intro()
```



```
quarterback_data %>% plot_missing()
```



```
quarterback_data %>% head()
```

```
# A tibble: 6 × 5
  Player Salary    PC    TD   Age
   <dbl>  <dbl> <dbl> <dbl> <dbl>
1      1   25.6  65.2    28    27
2      2   22.0  60.5    27    26
3      3   20.5  62.3    27    28
4      4   19.0  66.1    26    38
5      5   17    67.9    29    28
6      6   15.0  55      16    27
```

```
quarterback_data %>% tail()
```

```
# A tibble: 6 × 5
  Player Salary    PC    TD   Age
   <dbl>  <dbl> <dbl> <dbl> <dbl>
1     27   2.17  58.7    10    26
2     28   2.01  53.1     8    25
3     29   1.38  54.5    10    21
4     30   1.10  62.1    21    27
```

```
5      31  0.950  60.8    12    24
6      32  0.626  63.1    26    29
```

Comments on Dataset:

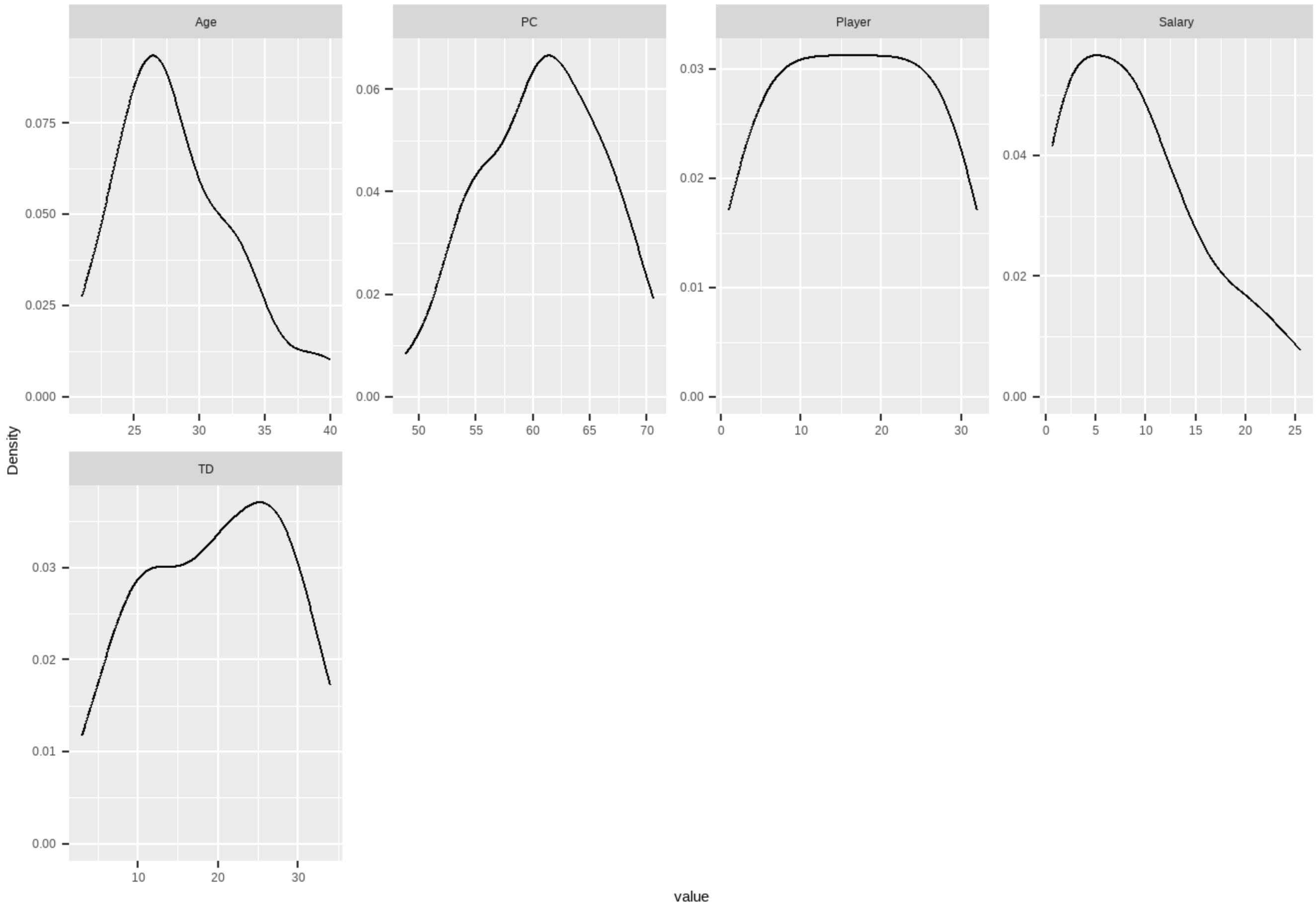
The Dataset has five variables, all of which are numeric. This dataset contains no missing values.

Dataset Variables include:

- Player: An ID identifier given for quarterbacks from all 32 teams in the NFL.
- Salary: Salary of associated quarterbacks, Salary is in millions (Dependent Variable).
- PC: Pass completion rate.
- TD: Amount of Touchdowns completed.
- Age: The age of the quarterback.

EDA

```
quarterback_data %>% plot_density()
```



```
## Look into skewness
skewness(quarterback_data$Salary) # Right Skewed
```

```
[1] 0.7295659
```

```
skewness(quarterback_data$Age) # Right Skewed
```

```
[1] 0.6968379
```

```
skewness(quarterback_data$TD) # Barely left skewed
```

```
[1] -0.1440229
```

```
skewness(quarterback_data$PC) # Barely left skewed
```

```
[1] -0.1796121
```

```
## Look for outliers
diagnose_outlier(quarterback_data)
```

```
# A tibble: 5 × 6
  variables outliers_cnt outliers_ratio outliers_mean with_mean without_mean
  <chr>            <int>          <dbl>         <dbl>     <dbl>        <dbl>
1 Player               0              0           NaN     16.5         16.5 
2 Salary               0              0           NaN      8.86         8.86
3 PC                   0              0           NaN     60.8         60.8 
4 TD                   0              0           NaN     19.7         19.7 
5 Age                  0              0           NaN     28.2         28.2 
```

```
range(quarterback_data$Salary)
```

```
[1]  0.62598 25.55663
```

```
range(quarterback_data$TD)
```

```
[1]  3 34
```

```
range(quarterback_data$Age)
```

```
[1] 21 40
```

```
range(quarterback_data$PC)
```

```
[1] 48.8 70.6
```

```
## Data deep dive
psych::describe(quarterback_data) ## Low Kurtosis values support outliers unlikely
```

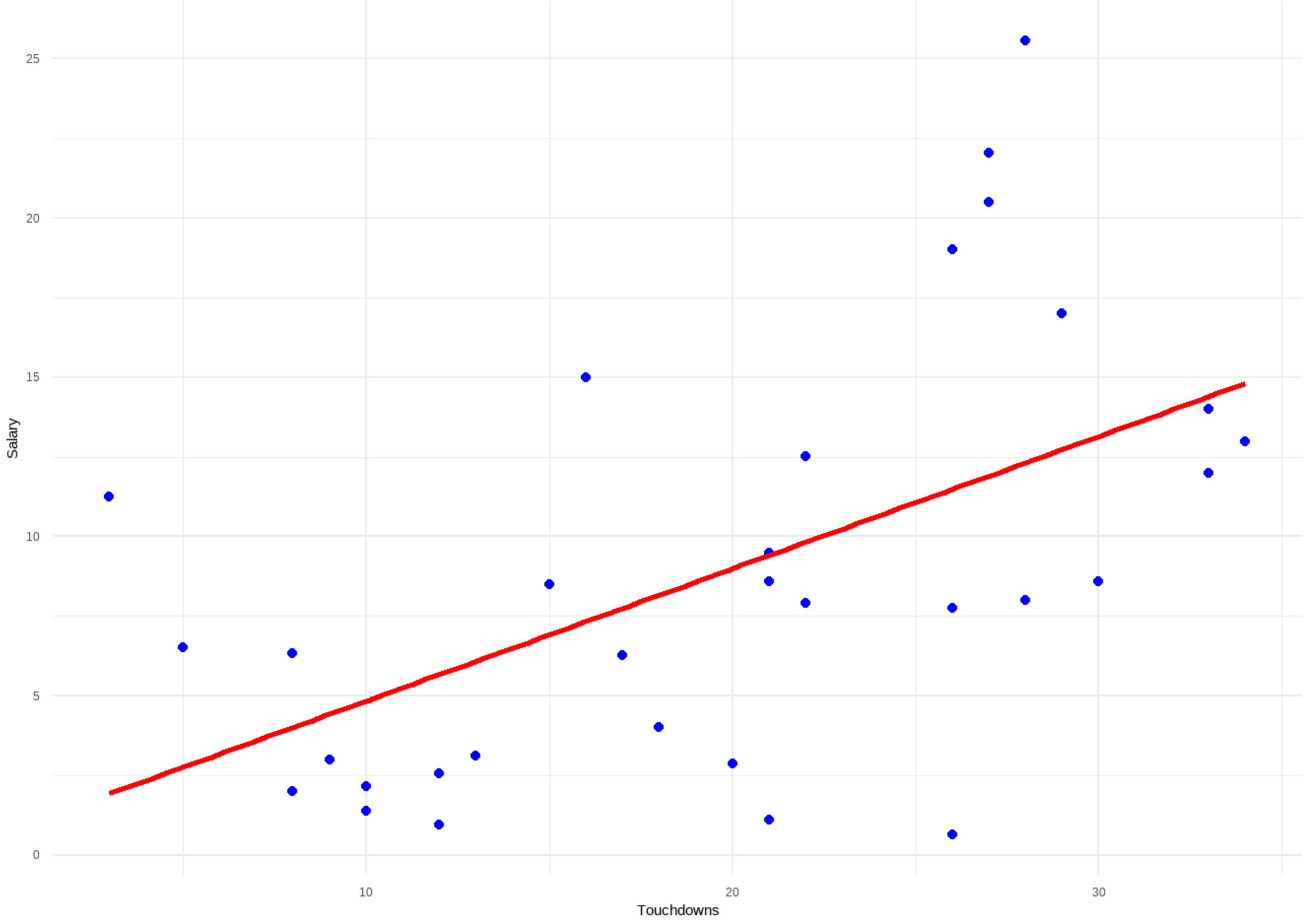
```
       vars  n  mean   sd median trimmed   mad  min   max range  skew kurtosis
Player    1 32 16.50 9.38  16.50   16.50 11.86 1.00 32.00 31.00  0.00    -1.31
```

```
Salary    2 32  8.86 6.71   7.96    8.18  7.41  0.63 25.56 24.93  0.73    -0.39
PC        3 32 60.82 5.33  60.65   60.89  6.38 48.80 70.60 21.80 -0.18    -0.79
TD        4 32 19.69 8.82  21.00   19.77 10.38  3.00 34.00 31.00 -0.14    -1.21
Age       5 32 28.19 4.53  27.00   27.88  4.45 21.00 40.00 19.00  0.70    -0.01
         se
Player 1.66
Salary 1.19
PC     0.94
TD     1.56
Age    0.80
```

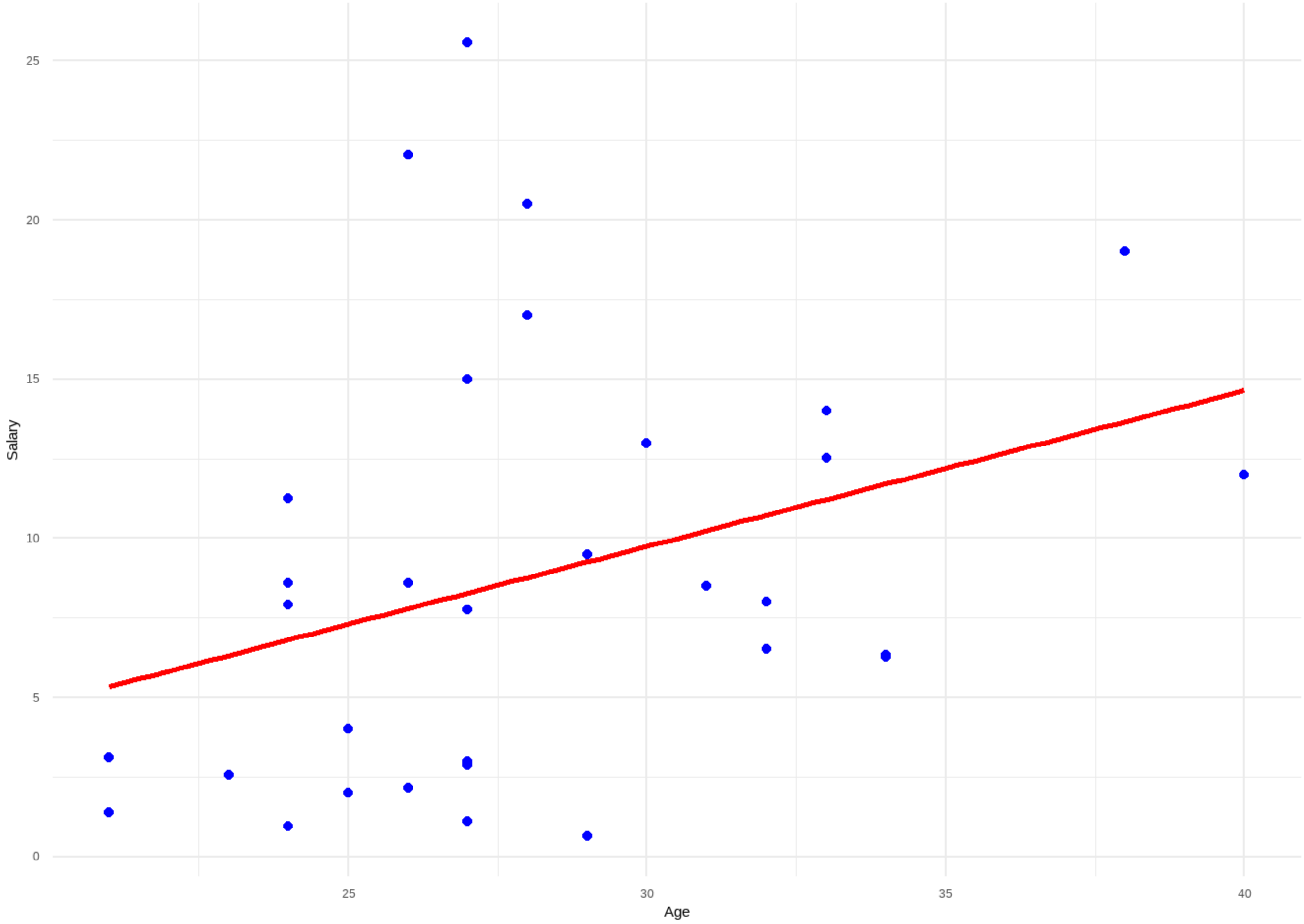
Comments on EDA:
Age and Salary are right skewed, Touchdown and Pass completion are slightly negatively skewed but are fine as is. Taking log of Age and/or Salary should be considered for modeling. There are no outliers in this dataset which is supported by the variables low kurtosis values.

Relationship of Numeric Values

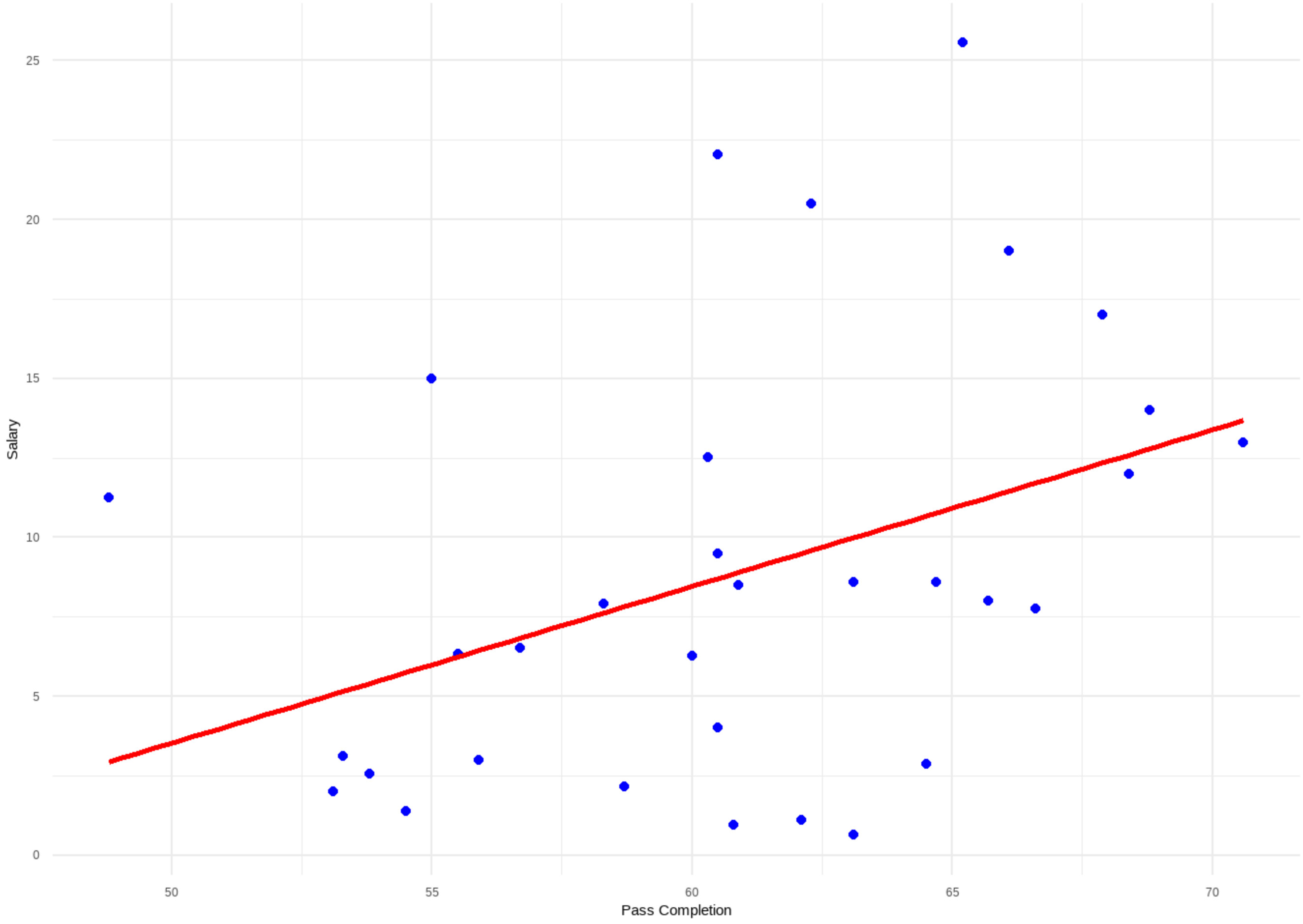
```
# Look at relationship between variables and dependent variable.
## Touchdown and Salary

ggplot(data=quarterback_data, aes(x= TD, y = Salary)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se=FALSE)+
  labs(Title = "Relationship between Touchdowns and Salary",
       subitile = "Expected relationship: Positive",
       x = "Touchdowns",
       y = "Salary")+
  theme_minimal() ## Moderately Positively Correlated
```



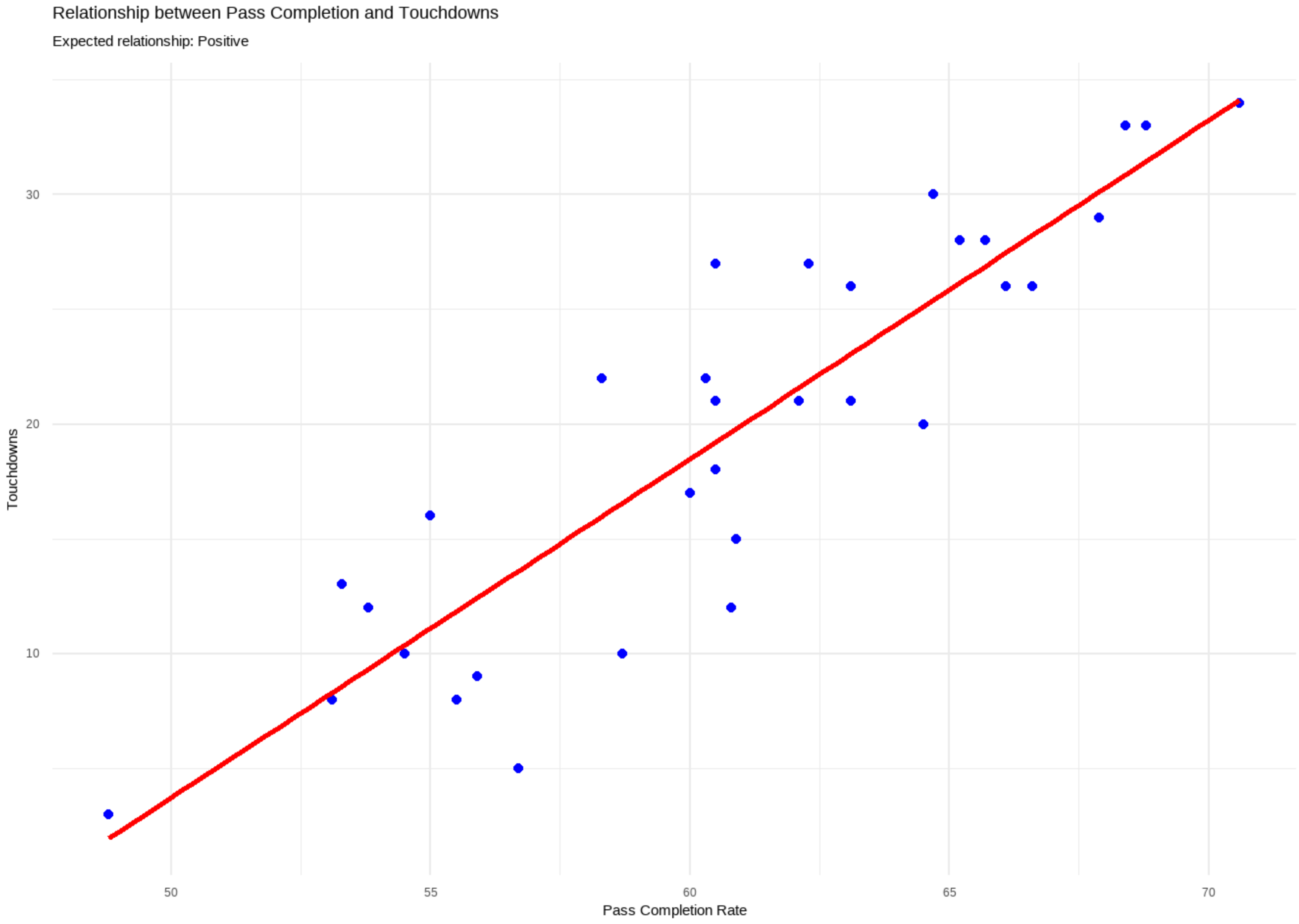
```
## Age and Salary

ggplot(data=quarterback_data, aes(x= Age, y = Salary)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se=FALSE)+
  labs(Title = "Relationship between Age and Salary",
       subitile = "Expected relationship: Positive",
       x = "Age",
       y = "Salary")+
  theme_minimal() ## Weakly Positively Correlated
```



```
## Pass Completion and Salary

ggplot(data=quarterback_data, aes(x= PC, y = Salary)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se=FALSE)+
  labs(Title = "Relationship between Touchdowns and Salary",
       subitile = "Expected relationship: Positive",
       x = "Pass Completion",
       y = "Salary")+
  theme_minimal() ## Weakly Positively Correlated
```

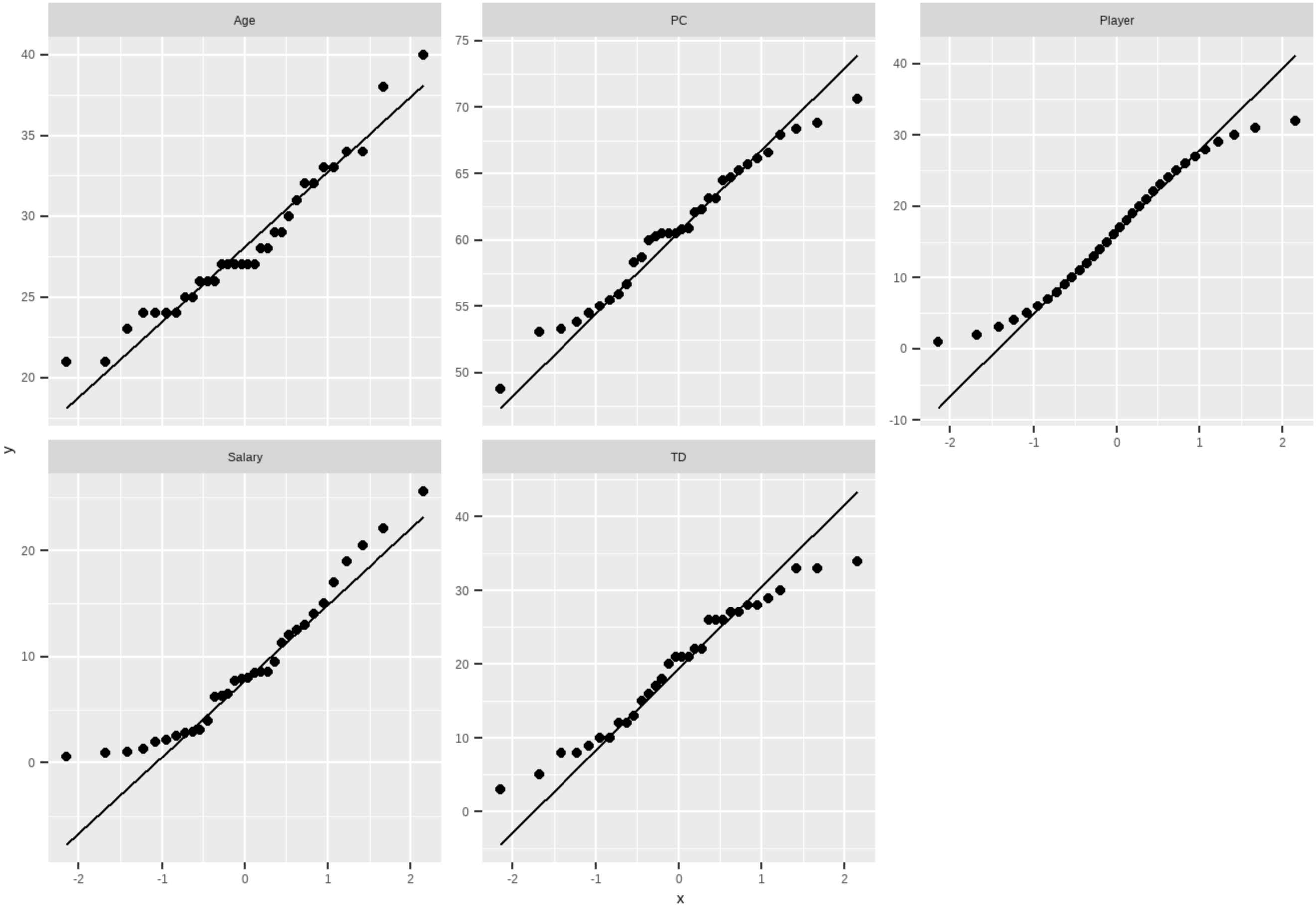


```
## Look at relationship between touchdown and pass completion because these seem like the most
         correlated variables from the predictors

ggplot(data = quarterback_data, aes(x = PC, y = TD)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Relationship between Pass Completion and Touchdowns",
       subtitle = "Expected relationship: Positive",
       x = "Pass Completion Rate",
       y = "Touchdowns") +
  theme_minimal()
```



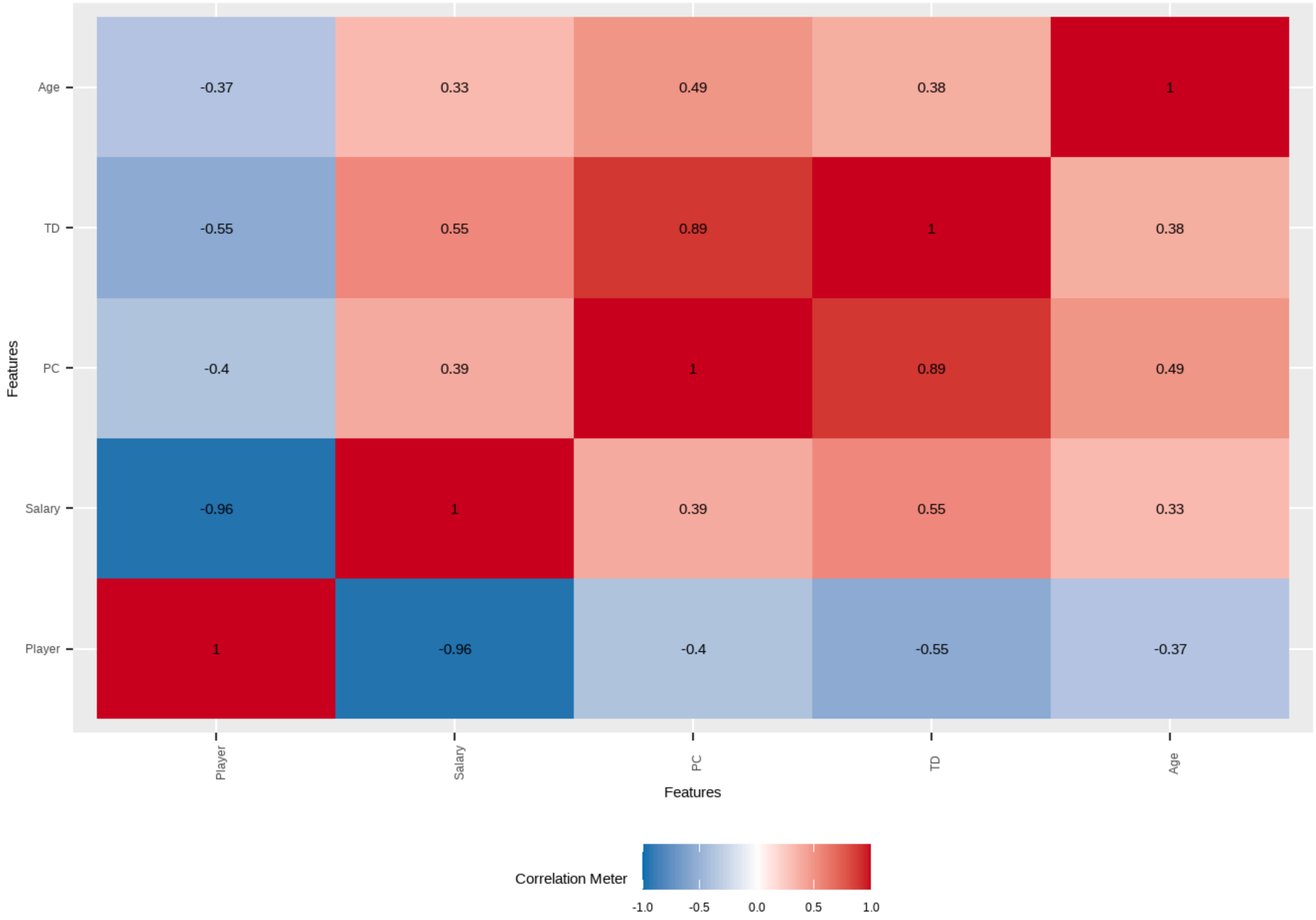
Comments on Variable Relationships:
All variables have a positive linear relationship to the dependent variable, salary, with touchdown seeming to have the strongest positive linear relationship with salary. It is also important to note that Pass completion and touchdown have a very strong positive linear relationship. This could lead to multicollinearity due to strong correlation.

```
quarterback_data %>% plot_qq()
```



Comments on QQ Plot:
The Q-Q plots indicate that most variables are approximately normally distributed with the exception of Age and Salary. Age and Salary as seen in earlier EDA are right skewed which is supported by the Q-Q plots. It is supported since most data points cluster near the bottom and middle of the Q-Q plot.

```
DataExplorer::plot_correlation(quarterback_data)
```



Comments on Correlation:
The Correlation matrix supports what was seen in earlier EDA. PC, TD, and Age all have a positive correlation with Salary. Again important to note that TD and PC have a very strong correlation which is something to remember when checking for multicollinearity.

Data Preparation

Partition Data

```
set.seed(1)
my_index <- createDataPartition(quarterback_data$Salary, p=0.75, list = FALSE)
trainset <- quarterback_data[my_index,]
testset <- quarterback_data[-my_index,]

mean(quarterback_data$Salary)
```

```
[1] 8.861379
```

```
mean(trainset$Salary)
```

```
[1] 8.929338
```

```
mean(testset$Salary)
```

```
[1] 8.657502
```

Comments on Data Partition:
This dataset was partitioned into a 75/25 split, with 75% of the data being used for the training dataset and 25% of the data being used for the test dataset. Since salary is the dependent variable, the mean of salary was taken to see if the split maintained a consistent distribution of the dependent variable. Since the means of salary was close across the datasets, it is acceptable to proceed into modeling. Set.seed was used to maintain reproducible results.

Modeling

Create Models

```
# Outputs of models purposefully hidden, summary of models show in next code chunk.
## Log Salary and Age
set.seed(1)

myctrl <- trainControl(method = "CV", number = 10)

# No Variable Transformations
raw_model <- train(Salary ~ PC + TD + Age,
                   data = trainset,
                   method = "lm",
                   trControl = myctrl
)
summary(raw_model)

## Log Salary
logs_model <- train(log(Salary) ~ PC + TD + Age,
                    data = trainset,
                    method = "lm",
                    trControl = myctrl)
summary(logs_model)

## Log Age

loga_model <- train(Salary ~ PC + TD + log(Age),
                    data = trainset,
                    method = "lm",
                    trControl = myctrl)
summary(loga_model)

## Log Both
```

```
logb_model <- train(log(Salary) ~ PC + TD + log(Age),
                    data = trainset,
                    method = "lm",
                    trControl = myctrl)
summary(logb_model)
```

Comments on Creating Models:
Four regression models were created. The first regression model used original, untransformed variables. The second regression model took the log of salary. The third regression model used log of age. Finally, the fourth regression model used both log of salary and log of age. Creating multiple models allowed for evaluation of which specification provided the best statistical fit and interpretability.

Compare Models

```
model1 <- raw_model$finalModel
model2 <- logs_model$finalModel
model3 <- loga_model$finalModel
model4 <- logb_model$finalModel

stargazer(model1, model2, model3, model4,
          type = "text",
          title = "Comparison of Regression Models",
          column.labels = c("Raw", "Log Salary", "Log Age", "Log Both"),
          dep.var.labels = "Salary (Dependent Variable)",
          covariate.labels = c("Pass Completion", "Touchdowns", "Age"),
          no.space = TRUE)
```

```
Comparison of Regression Models
===================================================================
                                       Dependent variable:
                              -------------------------------------
                                   Salary (Dependent Variable)
                                Raw    Log Salary Log Age  Log Both
                                (1)       (2)       (3)      (4)
-------------------------------------------------------------------
Pass Completion                -0.823    -0.092    -0.839   -0.094
                              (0.545)   (0.084)   (0.550)  (0.085)
Touchdowns                    0.769**    0.084    0.773**   0.085
                              (0.319)   (0.049)   (0.319)  (0.049)
Age                            0.112     0.037
                              (0.349)   (0.054)
`log(Age)`                                         4.025    1.127
                                                  (10.553) (1.636)
Constant                       40.276    4.687     30.928   2.079
                              (26.182) (4.054)   (36.149) (5.605)
-------------------------------------------------------------------
Observations                     24        24        24       24
```

```
R2                             0.298     0.201     0.299    0.201
Adjusted R2                    0.192     0.081     0.194    0.081
Residual Std. Error (df = 20)  5.920     0.917     5.914    0.917
F Statistic (df = 3; 20)       2.824*    1.678     2.844*   1.675
=================================================================
Note:                                  *p<0.1; **p<0.05; ***p<0.01
```

Comments on Comparison of Models:
Four regression models were tested, each varying in whether the logarithm of Salary and/or Age was applied. Among these, the model using only the log of Age achieved the best statistical fit based on adjusted R^2 and F-statistic values. However, the model using only the log of Salary was selected. The log-salary transformation effectively addresses the right-skewed distribution of salary while allowing coefficients to be interpreted as percentage changes in salary, making the results more meaningful in practical interpretations. The log-salary model is also statistically signficant.

Model Interpretation Assuming a 10% level of significance

Goodness of Fit

```r
logs_model_summary <- summary(logs_model$finalModel)
logs_model_summary
```

```

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.62580 -0.24334 -0.02242  0.63341  1.18324 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.68726    4.05447   1.156    0.261
PC          -0.09205    0.08444  -1.090    0.289
TD           0.08429    0.04936   1.708    0.103
Age          0.03748    0.05401   0.694    0.496

Residual standard error: 0.9167 on 20 degrees of freedom
Multiple R-squared:  0.2011,    Adjusted R-squared:  0.08124 
F-statistic: 1.678 on 3 and 20 DF,  p-value: 0.2037
```

Comments on Goodness of Fit:

Goodness of fit metric interpretation:
The p-value on the F-statistic is <.1 showing the model is significant.
The model explains 20.1% of the variation in Spend.
The standard error of the estimate is $0.917. This means that on average, the predicted log-salary values differ from the actual log-salary values by about 0.837 log units. However converting back into real salary

terms, e^-0.837 and e^0.837, actual salaries are typically between 0.43 and 2.31 times the predicted salary. Further converting this into a percentage, 2.31 - 1, actual salaries are roughly + or - 130% from predicted values on average. This shows moderate level of uncertainty when predicting but can be expected due to the spread of salary.

Coefficient Significance Interpretation:
One Variable is statistically significant, and that is the TD (touchdown) variable.

Coefficient Value Interpretation:
- PC: For every one unit increase there is on average a -0.092% decrease in salary ceterus paribus.
- TD: For every one unit increase in touchdown, there is on average a 0.084% increase in salary, ceterus paribus.
- Age: For every one unit increase in Age, there is on average a 0.037% increase in salary, ceterus paribus.

Multicollinearity

```
check_collinearity(logs_model$finalModel)
```

```
# Check for Multicollinearity

Low Correlation

 Term  VIF    VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
  Age 1.30 [1.06,  2.48]     1.14      0.77     [0.40, 0.94]

Moderate Correlation

 Term  VIF    VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
   PC 6.09 [3.64, 10.81]     2.47      0.16     [0.09, 0.27]
   TD 5.60 [3.37,  9.93]     2.37      0.18     [0.10, 0.30]
```

Comments on Multicollinearity:
- Age: Age shows low correlation with a VIF value of 1.33. The Tolerance value is 0.75 meaning age has low overlap with other variables in the dataset.
- TD: TD has moderate correlation with a VIF value of 4.76. The Tolerance value is 0.21 showing high overlap with other variables in the dataset. This is most likely with PC, as seen in EDA both showed strong correlation between each other.
- PC: PC has moderate correlation with a VIF value of 5.39. The Tolerance value is 0.19 showing high overlap with other variables in the dataset. This is most likely with TD, as seen in EDA both showed strong correlation between each other.

Although PC and TD have a moderately high VIF value both values should be kept. They are both different quarterback performance metrics and both metrics involve passing in them which explains some of the overlap.

Residual Analysis

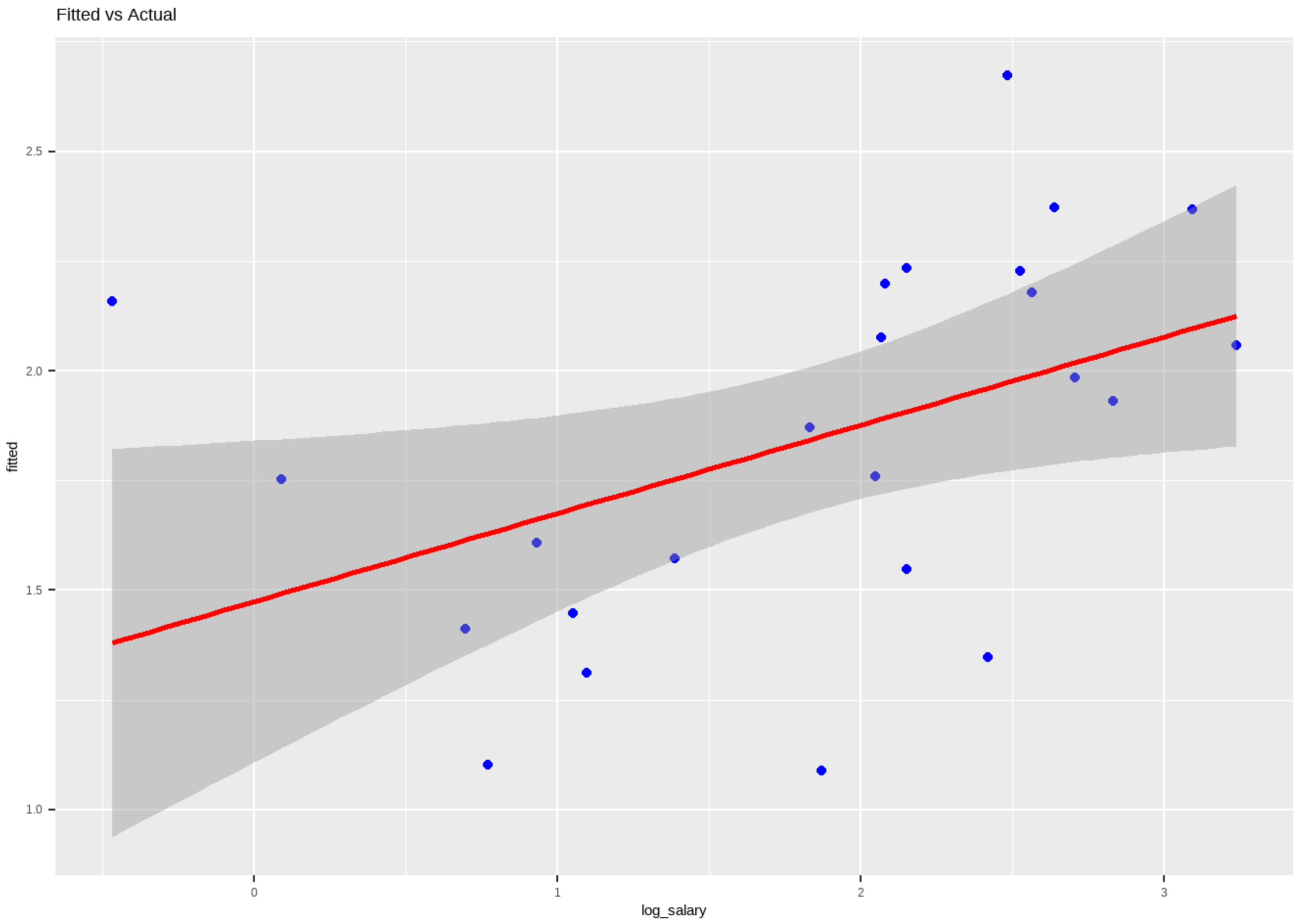
Create Dataset for Residual analysis

```
residual_data <- trainset %>%
  mutate(log_salary = log(Salary)) %>%
  cbind(fitted = logs_model$finalModel$fitted.values, residuals =
         logs_model$finalModel$residuals)
View(residual_data)
```

Comments on dataset creation for residual analysis:
To perform residual analysis, both the fitted values and residuals from the model were stored in a new dataset. Since the regression model was trained on the logarithm of Salary, the log of Salary variable was also added to the dataset to ensure consistency between the model scale and the residual diagnostics.

Fitted vs Actual (looking for linear relationship)

```
ggplot(residual_data, aes(x = log_salary, y = fitted)) +
  geom_point(color = "blue")+
  geom_smooth(method = "lm", color = "red", se = TRUE)+
  labs(title = "Fitted vs Actual")
```

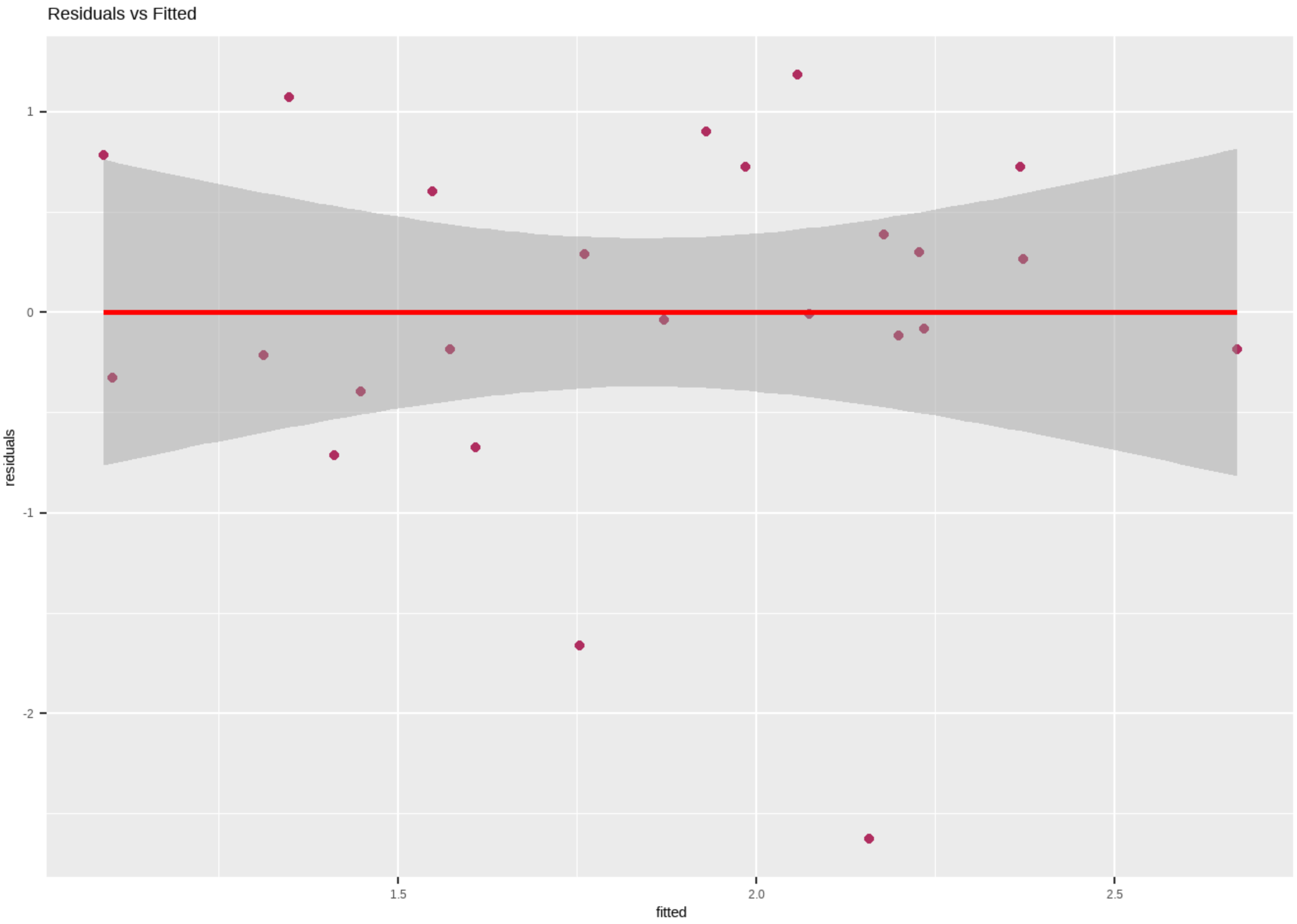


Comments on fitted vs actual plot:
This graph compares predicted log-salary values to actual log-salary values to see how closely the model

can replicate the observed data. The points follow an upward linear trend with some scatter around the line. This shows the model captures the main relationship between predictors and salary reasonbly well.

Residual vs Fitted (looking for no change in variability)

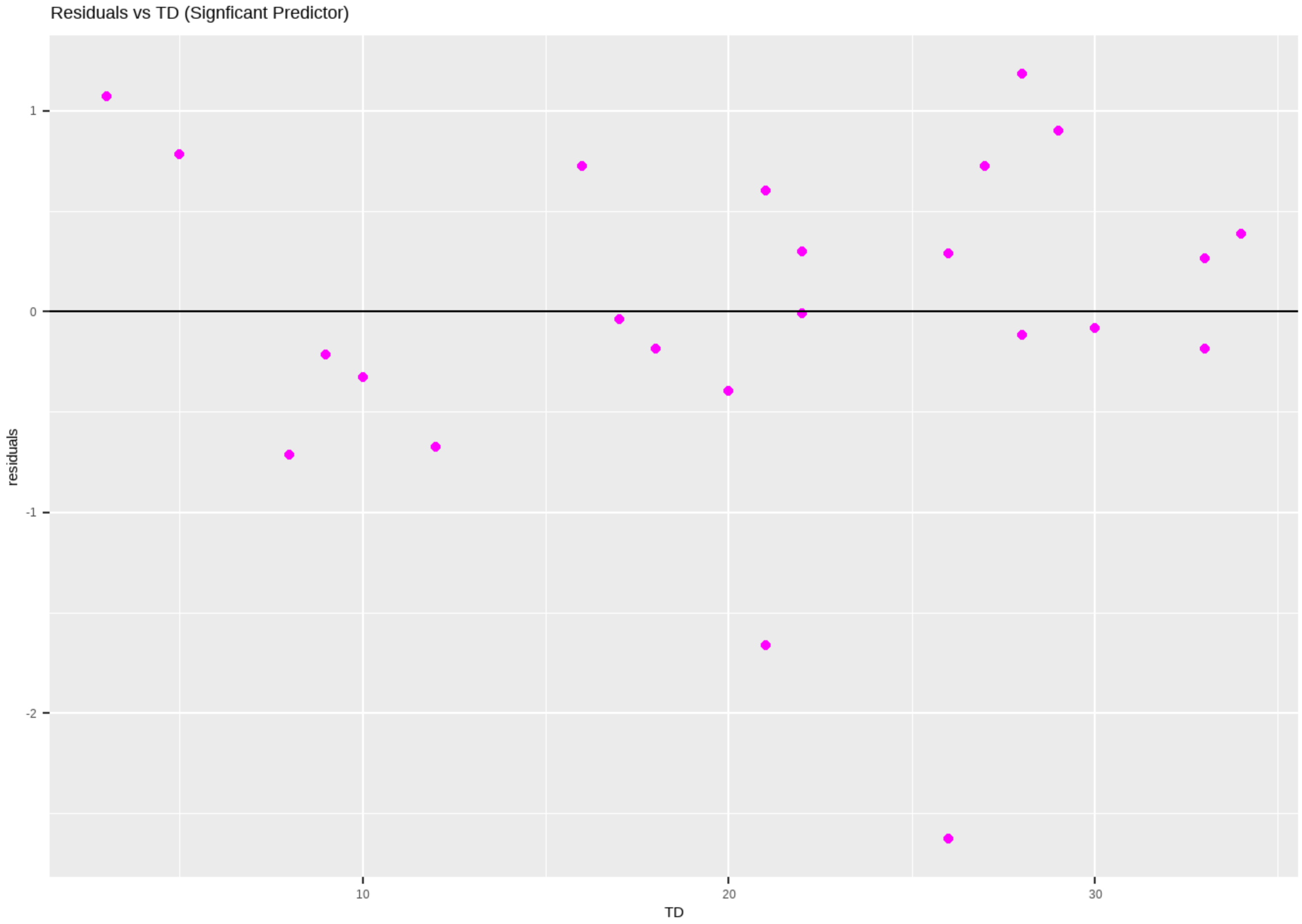
```
ggplot(residual_data, aes(x = fitted, y = residuals)) +
  geom_point(color = "maroon")+
  geom_smooth(method = "lm", color = "red", se=TRUE)+
  labs(title = "Residuals vs Fitted")
```



Comments on Residual vs Fitted:
This graph shows residuals plotted against fitted values to test for linearity. The residuals in this graph are randomly scattered around the zero line without strong patterns, showing the model satisfies linearity assumption.

Residuals vs Predictor (want linear pattern)

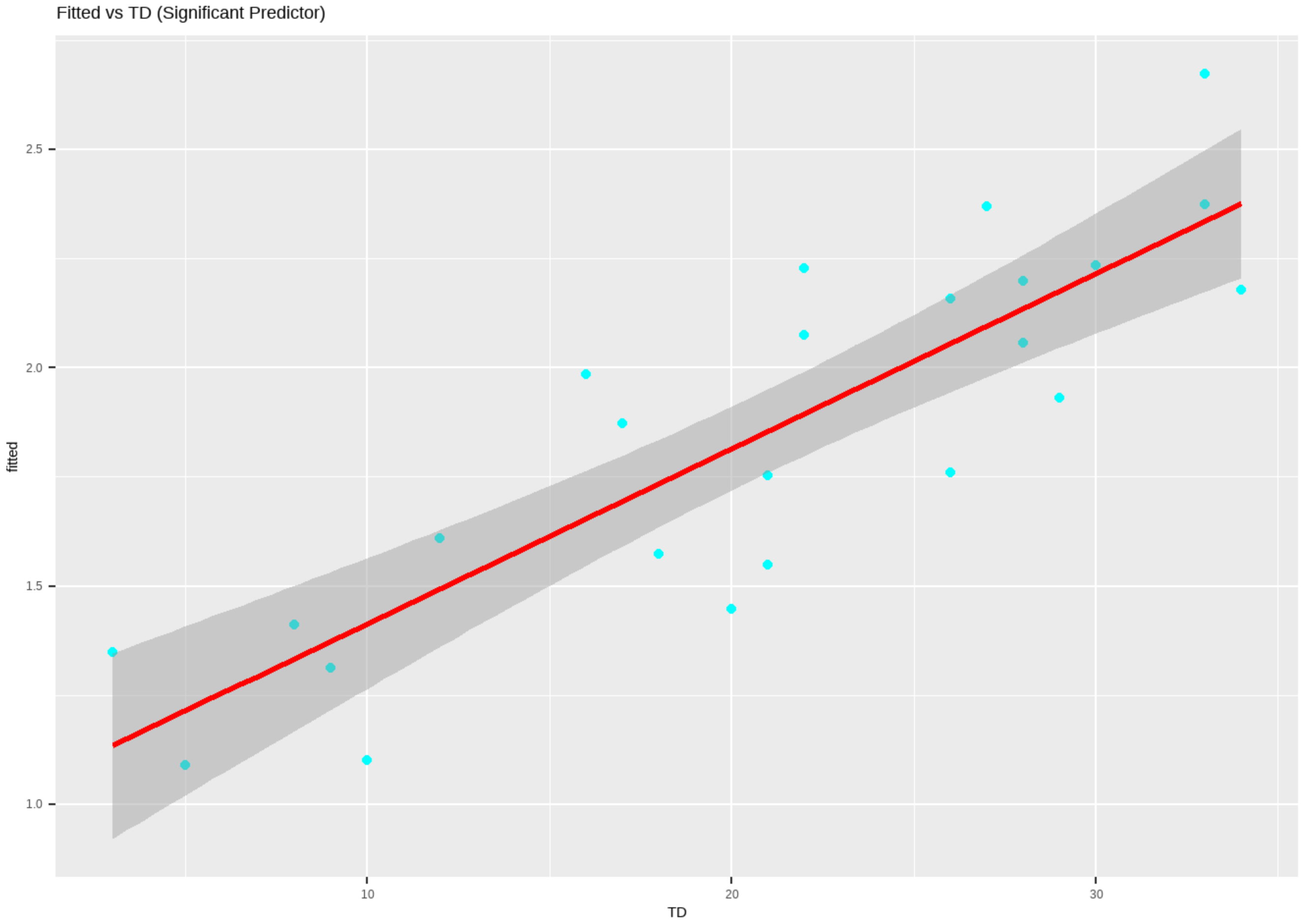
```
ggplot(residual_data, aes(x = TD, y = residuals))+
  geom_point(color="magenta") +
  geom_hline(yintercept = 0)+
  labs(title = "Residuals vs TD (Signficant Predictor)")
```



Comments on Residuals vs Predictor:
Since there was only one significant predictor variable in my model, TD, there was only one residual vs predictor graph made. This graph checks to see whether residuals have any visible trend when plotted against the TD variable. The residuals are randomly dispersed around around the zero line, showing that the relationship between TD and log-salary is captured well.

Fitted vs Predictor (want a linear pattern)

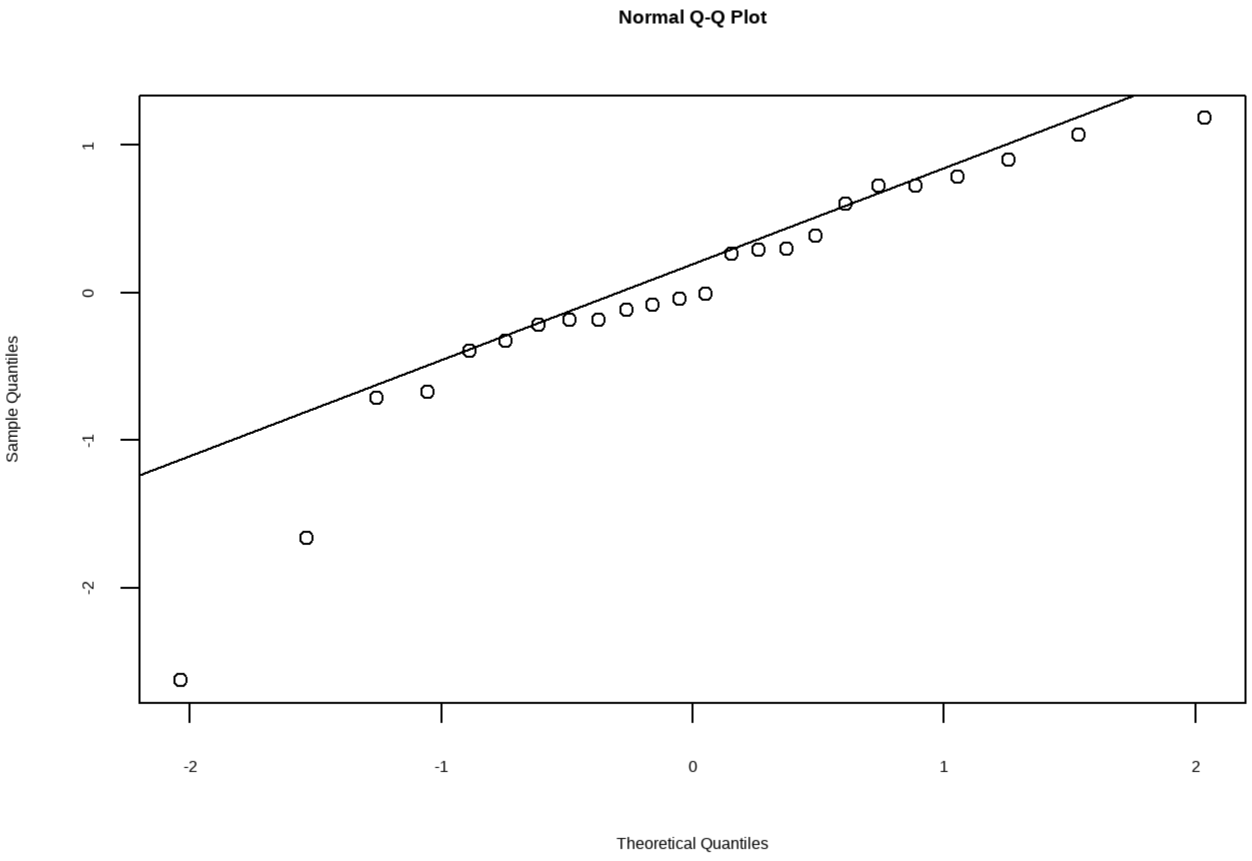
```
ggplot(residual_data, aes(x = TD, y = fitted)) +
  geom_point(color="cyan")+
  geom_smooth(method = "lm", color = "red", se = TRUE)+
  labs(title = "Fitted vs TD (Significant Predictor)")
```



Comments on Fitted vs Predictor:
This plot displays the fitted log-salary values against the number of touchdowns variable to show the relationship between the two. The graph shows an upward linear trend which supports a positive relationship between the two variables. Confirming that a higher touchdown count leads to a higher predicted salary.

QQ-Plot (want dots along the line - indicates normal distribution)

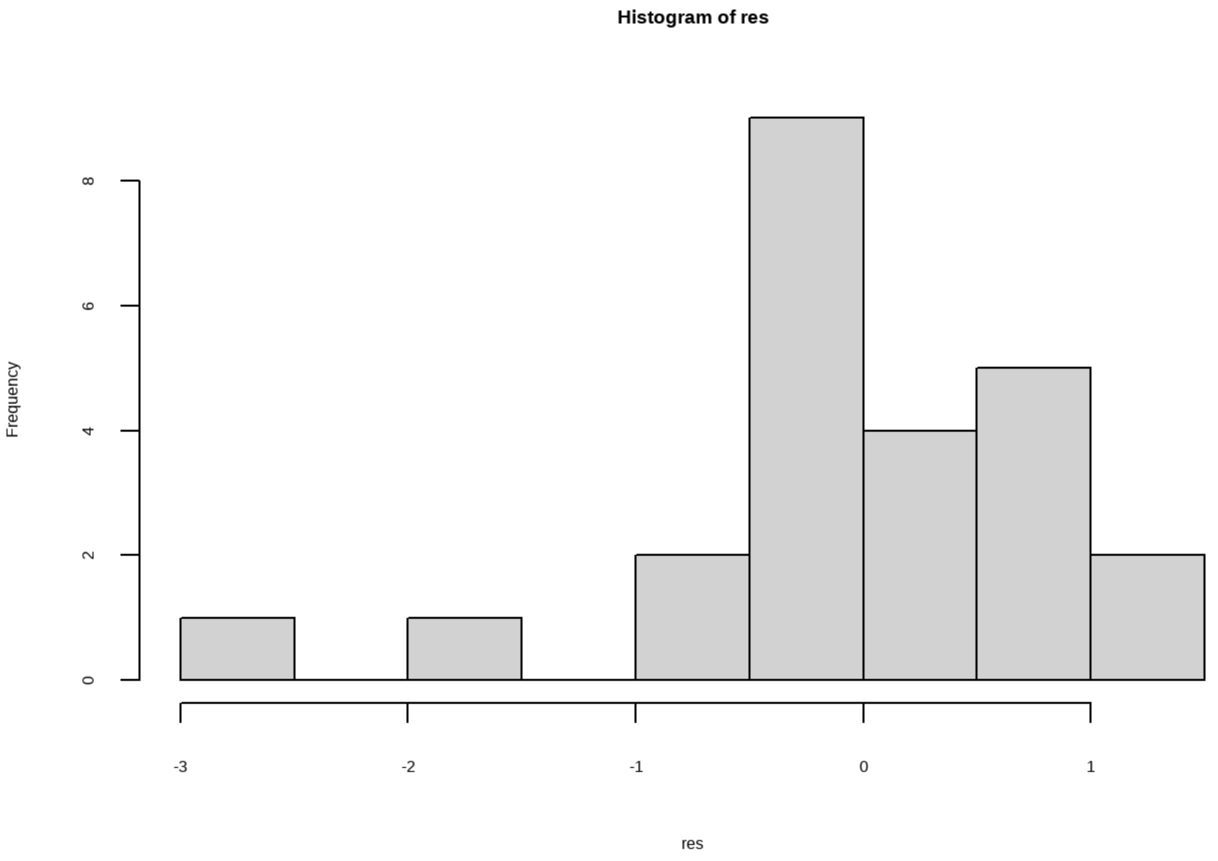
```
res <- resid(logs_model$finalModel)
qqnorm(res)
qqline(res)
```



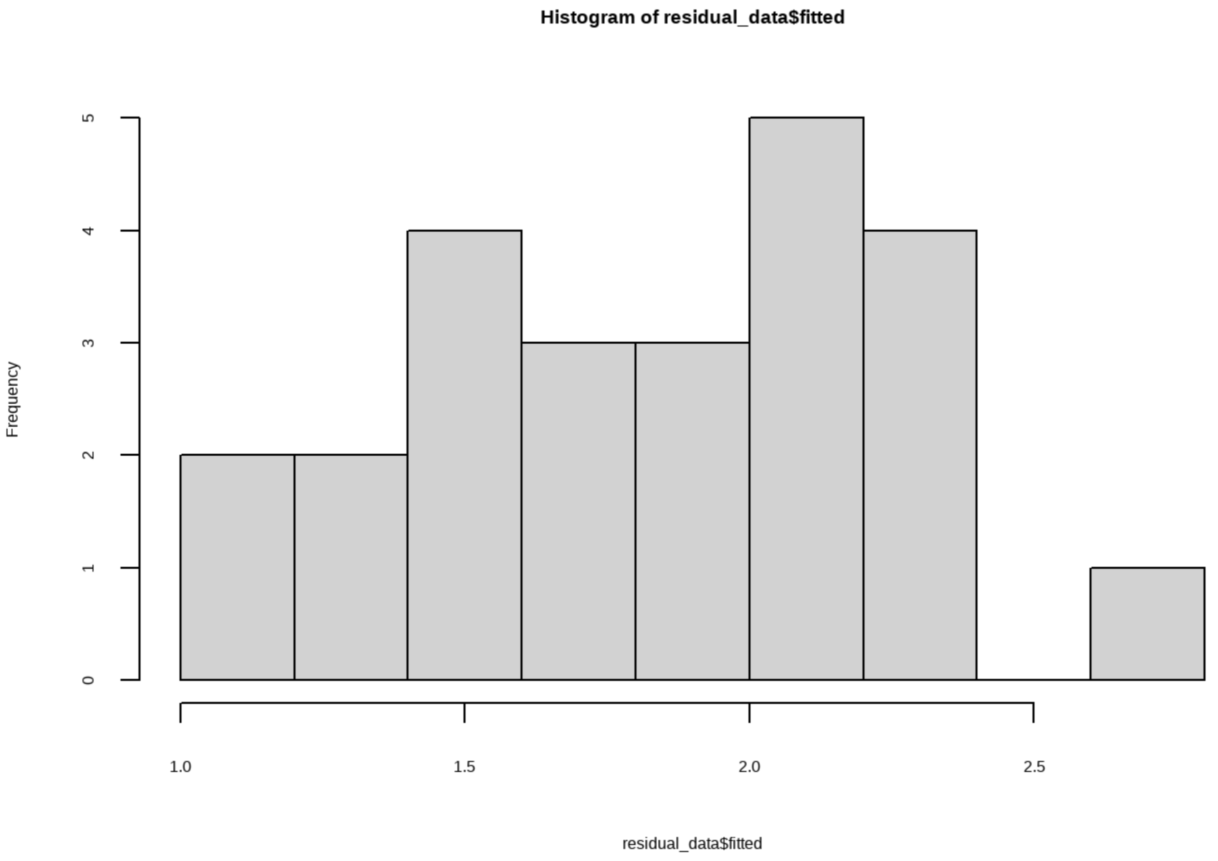
Comments on QQ-Plot:
The Q-Q plot compares distribution of standardized residuals to a normal distribution line. The points closely follow the normal distribution line showing an approximate normal distributions, supporting normality.

Histogram of Residuals (want normal distribution)

```
# Distribution of residuals
hist(res)
```



```
# Distribution of fitted values
hist(residual_data$fitted)
```



```
skewness(res)
```

```
[1] -1.218944
```

```
skewness(residual_data$fitted)
```

```
[1] -0.1141671
```

Comments on Histograms (residuals and fitted Values):
The histograms display the distributions of residuals and fitted values to show any potential spread. Although both histograms are not normally distributed, there is no severe skewness.

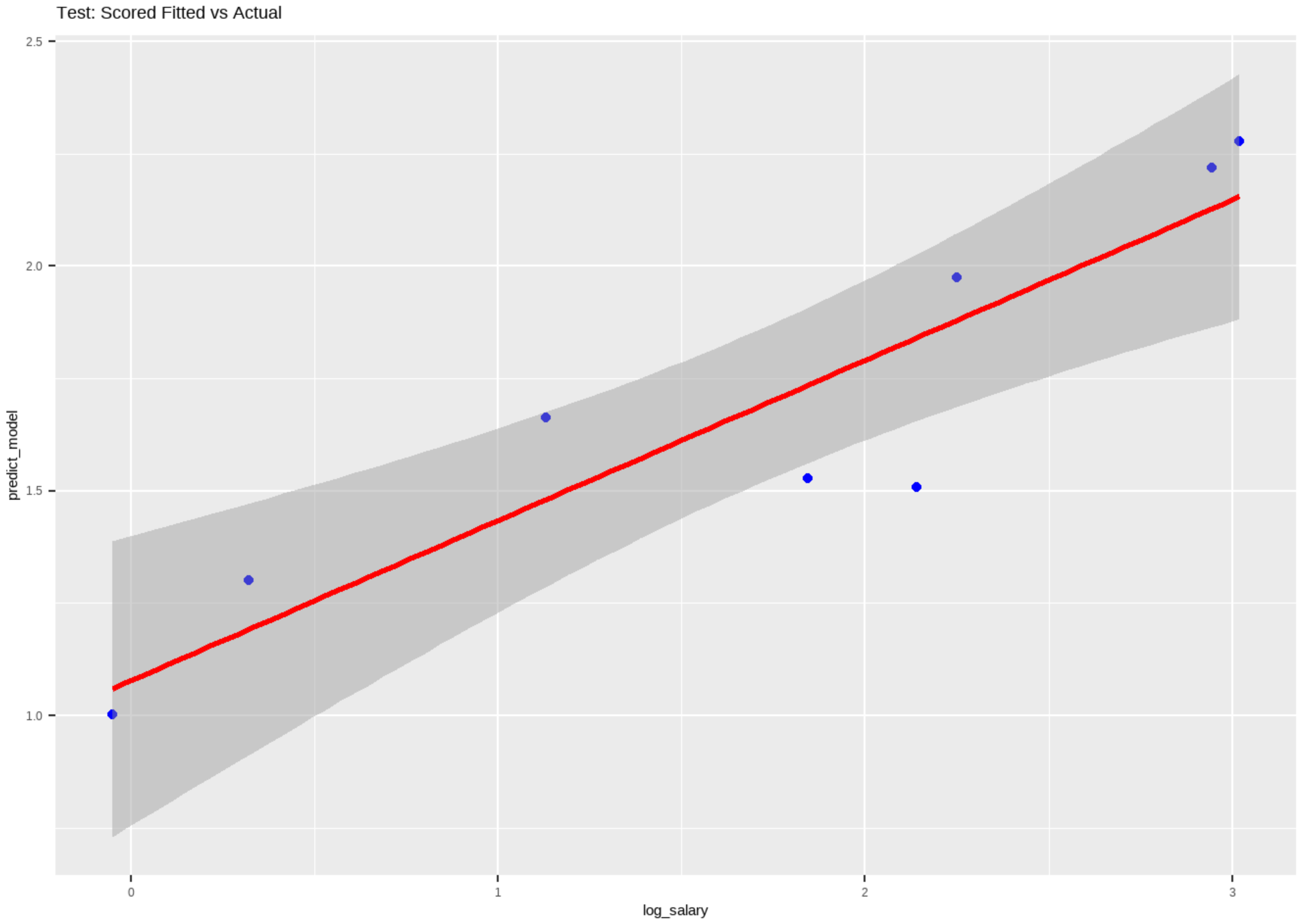
Does Model Meet OLS Assumptions

Comments on Model meeting OLS Assumptions:
Based on the residual plots and model statistics all major OLS assumptions are satisfied.
1. Linearity - The Model is linear because the coefficients enter the model linearly. Taking log of salary does not violage this assumption.
2. Random Sampling - This dataset takes the entire population of quarterbacks so sampling was not necessarily needed.
3. No perfect Multicollinearity - Predictors are not exact linear combinations of each other, only moderate

correlation exists between PC and TD.
4. Zero Conditional Mean of Erros - Residuals are randomly scattered around the zero line in residuals vs fitted plot, no obvious trends so no sign of systematic error.
5. Homescedasticity - The Variance of residuals is constant across all fitted values.
6. No Serial Correlation - Non-Applicable, this data is cross-sectional not time-series.
7. Exogeneity - The random residual pattern against TD and fitted values appear unrelated to errors.
8. Normality of Errors - The Q-Q plot shows slight deviation at th etails, and the histogram is not perfrectly symmetric. This shows slight non-normality, but is acceptable.

Evaluation

```
predict_model <- predict(logs_model, newdata=testset)
model_predict <- cbind(testset, predict_model) %>%
  mutate(log_salary = log(Salary))

View(model_predict)



# Fitted vs Actual
ggplot(model_predict, aes(x = log_salary, y = predict_model)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(title = "Test: Scored Fitted vs Actual")
```



Evaluation Metrics

```r
# Make sure the values are numeric in vectors not array (Metrics package is quirky that way)
actual <- as.numeric(model_predict$log_salary)
pred   <- as.numeric(model_predict$predict_model)

RMSE <- formattable::comma(round(Metrics::rmse(actual, pred), 2),
                                                 digits = 2, format = "f", big.mark = ",")

MAE <- formattable::comma(round(Metrics::mae(actual, pred), 2),
                                                digits = 2, format = "f", big.mark = ",")

MAD <- formattable::comma(round(mad(actual - pred), 2),
                                                digits = 2, format = "f", big.mark = ",")

MAPE <- formattable::percent(round(Metrics::mape(actual, pred), 5),
                                                digits = 2)

cat("RMSE:", RMSE,"\n")
```

```
RMSE: 0.71
```

```
cat("MAE :", MAE,"\n")
```

```
MAE : 0.66
```

```
cat("MAD :", MAD,"\n")
```

```
MAD : 0.65
```

```
cat("MAPE:", round(MAPE*100,2),"%\n") ## Refresh Memory on what these mean
```

```
MAPE: 315.7 %
```

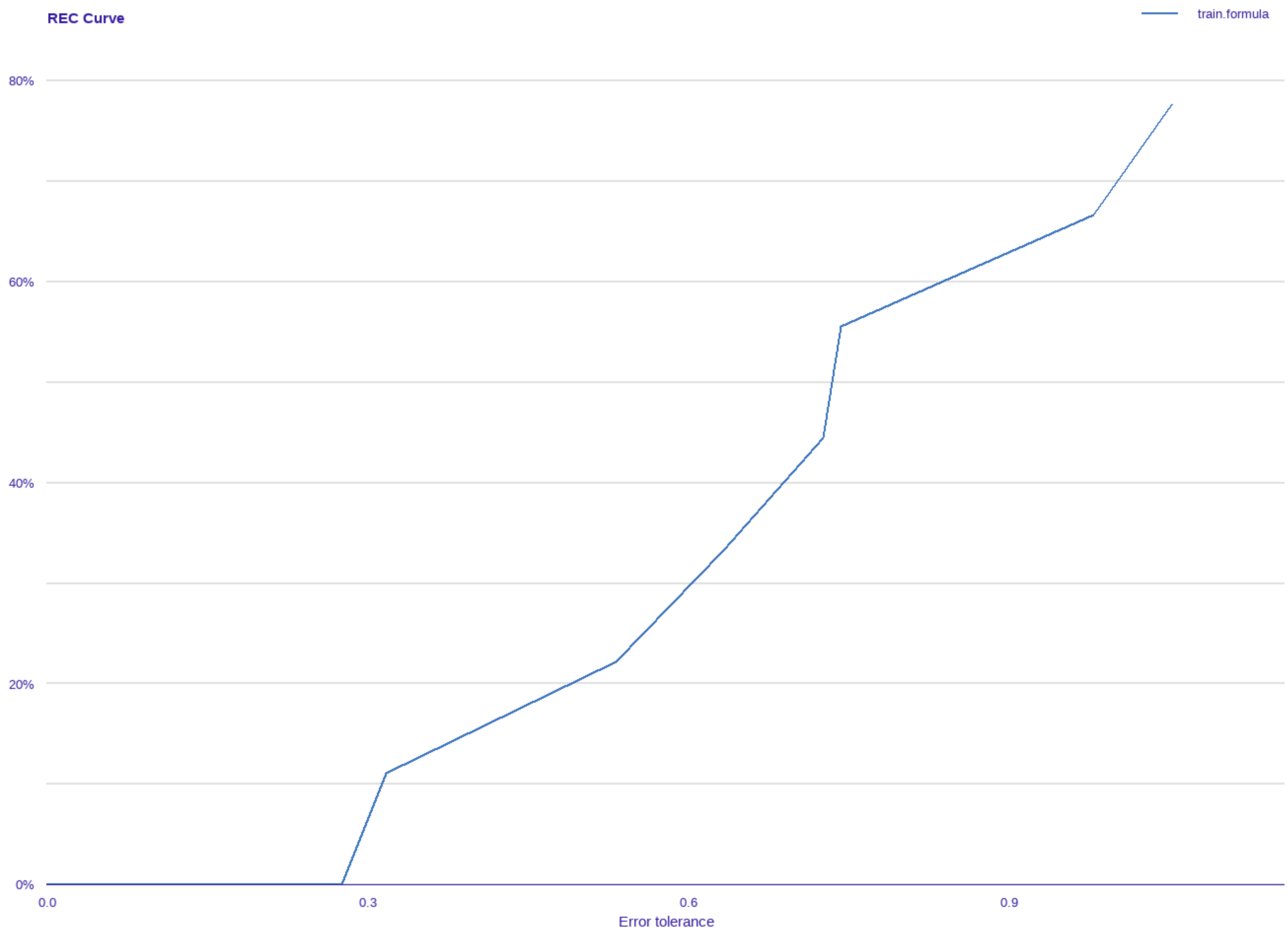
Comments on Evaluation Metrics:
Since the model predicts log of salary, the above error metrics represent proportional rather than absolute deviations.
- RMSE (0.33): The Model's predictions typically deviate from actual salaries by 0.33 log unites or +/- 39 % in real terms. e^0.33 = 1.39 - 1 = 0.39 or 39%.
- MAE (0.26): The Average absolute prediction error is 0.26 log units, meaning predicted salaries are about 30% off actual values on average.
- MAD (0.31): The average deviation of residuals from the fitted line is 0.31, showing a moderate level of prediction error.
- MAPE (17.95%): On average, the model's salary predctions are within about 18% of the actual values.

Evaluation REC

```
## Add log of salary to testset

testset <- testset %>%
  mutate(log_salary = log(Salary))
View(testset)


lm_audit <- audit(logs_model, data = testset, y = testset$log_salary)
```

```
Preparation of a new explainer is initiated
  -> model label       :  train.formula  (  default  )
  -> data              :  8  rows  6  cols
  -> data              :  tibble converted into a data.frame
  -> target variable   :  8  values
  -> predict function  :  yhat.train  will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package caret , ver. 6.0.94 , task regression (  default  )
  -> predicted values  :  numerical, min =  1.001657 , mean =  1.684076 , max =  2.277801
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -1.052592 , mean =  0.01636467 , max =  0.7426241
  A new explainer has been created!
```

```
mr_lm <- model_residual(lm_audit)
plot_rec(mr_lm)
```



```
score_rec(lm_audit)
```

```
rec: 0.5104055
```

Comments on REC Value:
The REC Area is relatively low at 0.20, the closer the value is to 1 the better. I think this number is acceptable however due to the smaller dataset and the range of quarterback salaries.

Deployment

Predict Players Based off Model

```
model_all <- train(log(Salary) ~ PC + TD + Age,
                   data = quarterback_data,
                   method = "lm",
                   trControl = myctrl)
summary(model_all)
```

```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.70738 -0.29993  0.03669  0.60311  1.21737 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  4.80081    3.15421   1.522   0.1392  
PC          -0.12010    0.06669  -1.801   0.0825 .
TD           0.10090    0.03793   2.660   0.0128 *
Age          0.08251    0.03853   2.141   0.0411 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8367 on 28 degrees of freedom
Multiple R-squared:  0.3523,	Adjusted R-squared:  0.2829 
F-statistic: 5.077 on 3 and 28 DF,  p-value: 0.006221
```

```r
test <- quarterback_data %>%
  mutate(resid = residuals(model_all),
         predict = fitted(model_all),
         log_salary = log(Salary))

test2_real <- test %>%
  mutate(actual_salary = exp(log_salary),
         predicted_salary = exp(predict),
         actual_resid = actual_salary - predicted_salary)


test2 <- dplyr::filter(test2_real, Player %in% c(8, 16))
test2 %>% dplyr::select(Player, actual_salary, actual_resid, predicted_salary)
```

```
# A tibble: 2 × 4
  Player actual_salary actual_resid predicted_salary
   <dbl>         <dbl>        <dbl>            <dbl>
1      8         13.0          3.71             9.28
2     16          8.01        -2.75            10.8 
```

Comments on Pay based off the Model:

As part of this analysis the general manager wants to know if based on the influences that affect salary if players 8 and 16 are being over or underpayed.

- Player 8: Based on the models predicted salary, player 8 is being overpayed by about 3.7 million dollars with his predicted salary being about 9.2 million dollars but his actual salary being about 12.9 million dollars.
- Player 16: Based on the models predicted salary, player 16 is being underpayed by about 2.7 million

dollars with his predicted salary being about 10.7 million dollars but his acutal salary being about 8 million dollars.

Recommendations to the General Manager

The linear regression model is a useful tool for evaluating performance metrics relating to salary. It has relatively strong accuracy and moderately low prediction error. However, this model should not be the only thing used in decision making. The model accounted for low variability in the dataset. The dataset this model uses also does not account for all factor’s of quarterback salary such as leadership qualities and team fit.