MATH5743M: Statistical Learning: Assessed Practical 1 - Predicting the Olympic Games

Kalyani Sanjay Bhase

25/03/2022

Task 1:

library(tidyverse) library(ggplot2) library(Metrics) library(psych) library(caret)

Let's start with importing the dataset and building a data frame before we do any analysis on it.

```
df <- read.csv("medal_pop_gdp_data_statlearn.csv")
```

The data file includes the following information for 71 countries that have won at least one gold medal in each of the last three Olympic Games: country, population, GDP, and medals earned in Beijing 2008, London 2012, and Rio 2016.Our data looks as below:

```
head(df)
```

```
##      Country     GDP Population Medal2008 Medal2012 Medal2016
## 1    Algeria  188.68   37100000         2         1         2
## 2  Argentina  445.99   40117096         6         4         4
## 3    Armenia   10.25    3268500         6         3         4
## 4  Australia 1371.76   22880619        46        35        29
## 5 Azerbaijan   63.40    9111100         7        10        18
## 6    Bahamas    7.79     353658         2         1         2
```

The following is a summary of our data:

```
summary(df)
```

```
##    Country               GDP             Population          Medal2008
##  Length:71          Min.   :    6.52   Min.   :3.537e+05   Min.   :  1.00
##  Class :character   1st Qu.:   51.52   1st Qu.:5.513e+06   1st Qu.:  2.00
##  Mode  :character   Median :  229.53   Median :1.673e+07   Median :  6.00
##                     Mean   :  903.25   Mean   :7.384e+07   Mean   : 13.11
##                     3rd Qu.:  704.37   3rd Qu.:4.958e+07   3rd Qu.: 13.50
##                     Max.   :15094.00   Max.   :1.347e+09   Max.   :110.00
##    Medal2012       Medal2016
##  Min.   :  1.0   Min.   :  1.00
##  1st Qu.:  3.0   1st Qu.:  3.00
##  Median :  6.0   Median :  7.00
##  Mean   : 13.3   Mean   : 13.44
##  3rd Qu.: 13.0   3rd Qu.: 15.00
##  Max.   :104.0   Max.   :121.00
```

A linear model is one in which the input variables and the single output variable are assumed to have a linear relationship. Linear regression is a basic and commonly used type of predictive analysis, which determines how a target variable is affected by the predicted variables. The multiple linear regression for Y as a function of X is:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon,\ \epsilon \sim N(0, \sigma^2)$$

where,

- A scalar constant – β_0
- β_1, β_2 are regression coefficients
- x_1 and x_2 are input variables
- A residual ϵ which is unknown,but assumed to come from a Normal distribution with zero mean and unknown variance:

  $$\epsilon \sim (0, \sigma^2)$$

- Y is the target variable

The glm() method in R is used to fit generalised linear models, which are defined by a symbolic description of the linear predictor and a description of the error distribution.

```
LinearReg_model = glm(Medal2012 ~ Population + GDP , data= df)
summary(LinearReg_model)
```

```
##
## Call:
## glm(formula = Medal2012 ~ Population + GDP, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -20.568   -5.961   -2.462    3.932   60.121
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.076e+00  1.500e+00   4.051 0.000133 ***
## Population  5.247e-09  7.193e-09   0.729 0.468225
## GDP         7.564e-03  7.325e-04  10.326 1.45e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 132.1562)
##
##     Null deviance: 28402.8  on 70  degrees of freedom
## Residual deviance:  8986.6  on 68  degrees of freedom
## AIC: 553.19
##
## Number of Fisher Scoring iterations: 2
```

A linear regression model was created for medal count in 2012 from Population and GDP as input variable. Some of the observations were made from the model and they can stated as below:

- The intercept is 6.076

- The regression coefficient value for Population is very low, which is $5.247e-09$ and has P-value of 0.468 which is very high. This demonstrates that population has a minor impact on a country's medal count and is hence statistically unimportant.
- In contrast to Population, the GDP has a regression coefficient value of $7.564e-03$ and has a p-value of $1.45e-15$, which is less than 0.5 and hence is statistically significant. This means that one unit change in GDP produces approximately a $7.564e-03$ increase in the target variable.
- The AIC score for the model is 553.19

To make predictions for 2016, let us first create a new dataframe with 2016 data.

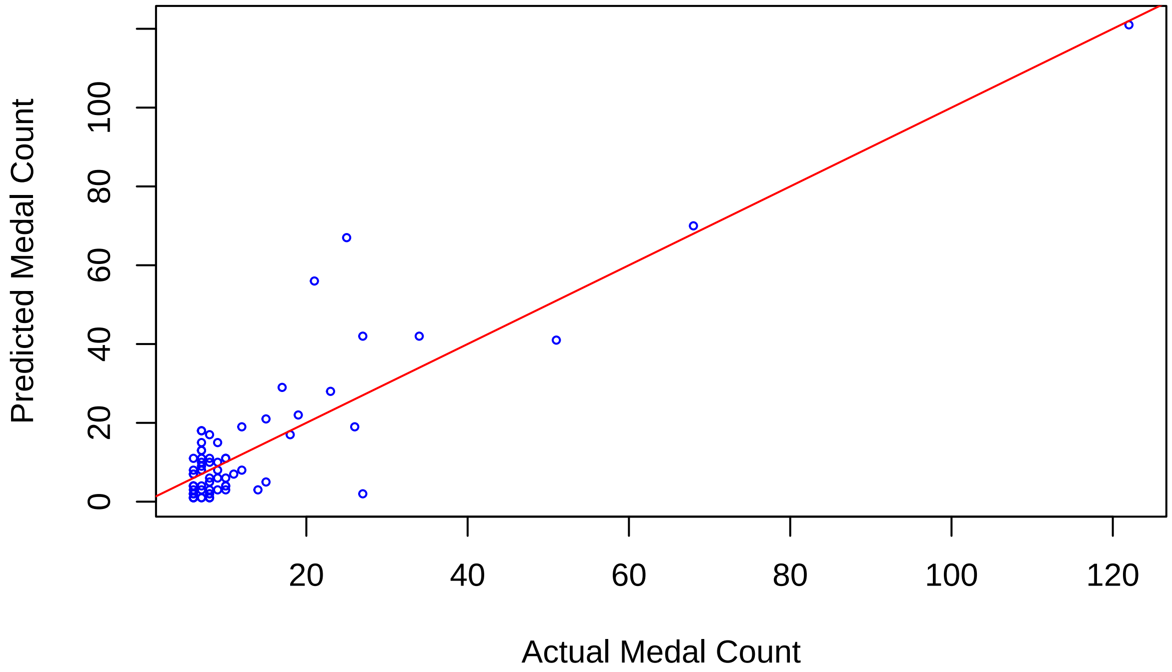
```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.1.3
```

```
#Creating a New Dataframe
new_df = data.frame(df[,c(2,3)])
#Creating a Duplicate dataframe to store predicted values
Games_df = df
#Predicting the model count for 2016 and
predictions= predict(LinearReg_model, newdata = new_df)
Games_df$predictions = round(predictions,digits = 0)
print(Games_df$predictions)
```

```
##  [1]   8  10   6  17   7   6   6   7  10  26   7  19  68   9   7   7   8   9   7
## [20]   8   6   7   8  27   6  34  25   8   7  27  14   7   8  23   6  51   8   7
## [39]   6   8  15   6   6   7  12   7   6  10  10   8   8  21   6   8   7   6   9
## [58]  15  18  10  11  10   6   9   6   6  12   8 122   7   9
```

```
#Plotting the predicted VS actual value for medal count in 2016
plot(Games_df$predictions, Games_df$Medal2016, col="blue",
xlab = "Actual Medal Count", ylab = "Predicted Medal Count",cex = 0.5)
abline(a=0,b=1, col = "red")
```

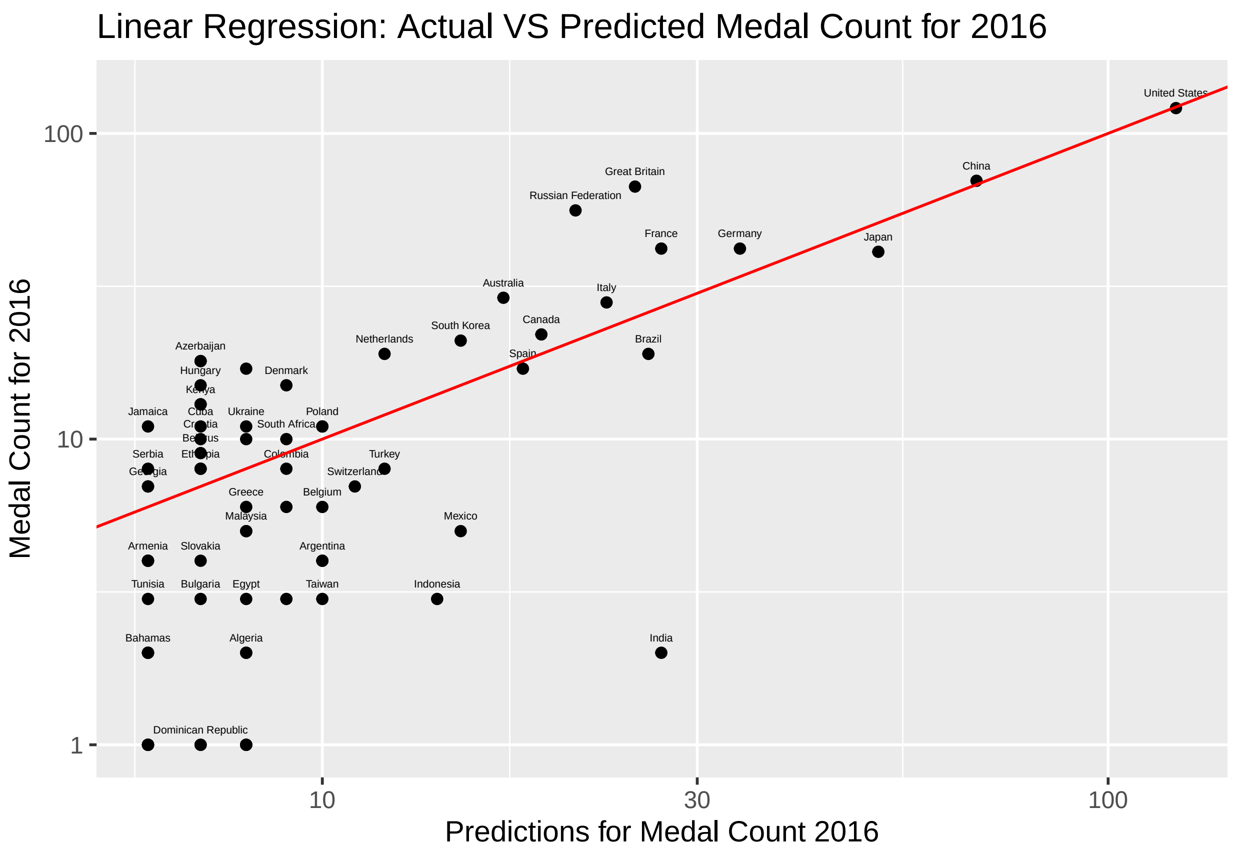


From the graph plotted above it is evident that most of the countries have earned medals less than 20. Let us transform the axes for further enhanced visibility.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
ggplot(data=Games_df, aes(x=predictions, y=Medal2016,label = Country)) +
  geom_point() +
  scale_y_continuous(trans='log10')+
  scale_x_continuous(trans='log10')+
  xlab("Predictions for Medal Count 2016")+
  ylab("Medal Count for 2016")+
  ggtitle("Linear Regression: Actual VS Predicted Medal Count for 2016")+
  geom_text(size=1.5,nudge_y = 0.05,  check_overlap = TRUE)+
  geom_abline(slope = 1, intercept = 0, col = 'red')
```



We have now transformed the graph for better understanding of the number of medals each country has won using logarithmic function. From the transformed graph, it is evident that majority of the countries have won the medals in the range 0 to 10. The line passing through the data points is the imaginary line on which all data points would have lied if the predicted values were same as the actual values. Countries such as Spain, China, and the United States are quite near to the plotted line with slope 1 and intercept 0, indicating that their actual and predicted medal counts for 2016 are close.

```
Games_df$DifferenceInCount <- (Games_df$Medal2016 - Games_df$predictions)
#sorting the dataframe to obtain outliers
Games_df = Games_df[with(Games_df, order(DifferenceInCount)),]
head(x = Games_df[, c(1,6,7,8)])
```

```
##      Country Medal2016 predictions DifferenceInCount
## 30     India         2          27               -25
## 31 Indonesia         3          14               -11
## 36     Japan        41          51               -10
## 41    Mexico         5          15               -10
## 10    Brazil        19          26                -7
## 23   Finland         1           8                -7
```

```
tail(x = Games_df[, c(1,6,7,8)])
```

```
##               Country Medal2016 predictions DifferenceInCount
## 5          Azerbaijan        18           7                11
## 46        New Zealand        18           7                11
## 4           Australia        29          17                12
## 24            France        42           27                  15
## 52 Russian Federation        56           21                  35
## 27      Great Britain        67           25                  42
```

The difference between the actual and expected medal count for 2016 is generated and saved in the ‘DifferenceInCount’ column, which can aid in the detection of outliers. A positive difference suggests that the country did significantly better than expected at the Olympics, whilst a negative difference indicates that the country did poorly. As can be seen from the graph and the difference between actual and predicted results, India, Indonesia, Japan, Mexico, Brazil, and Finland performed poorly at the 2016 Olympics, while countries such as the Great Britain, Russian Federation, France, Australia, New Zealand, and Azerbaijan performed admirably.

```
#postResample(pre = Games_df$predictions,obs = Games_df$Medal2016)
mae(Games_df$predictions,Games_df$Medal2016)
```

```
## [1] 6.084507
```

```
rmse(Games_df$predictions,Games_df$Medal2016)
```

```
## [1] 9.086439
```

The mean absolute error (MAE) is 6.08.This indicates that the average absolute difference between the observed values and the predicted values is 6.08. The RMSE for the model is 9.08 and indicates that the model is a decent.

Task 2:

```
lg=log(df$Medal2012)
Log_RegModel = glm(lg ~ Population + GDP , data= df)
summary(Log_RegModel)
```

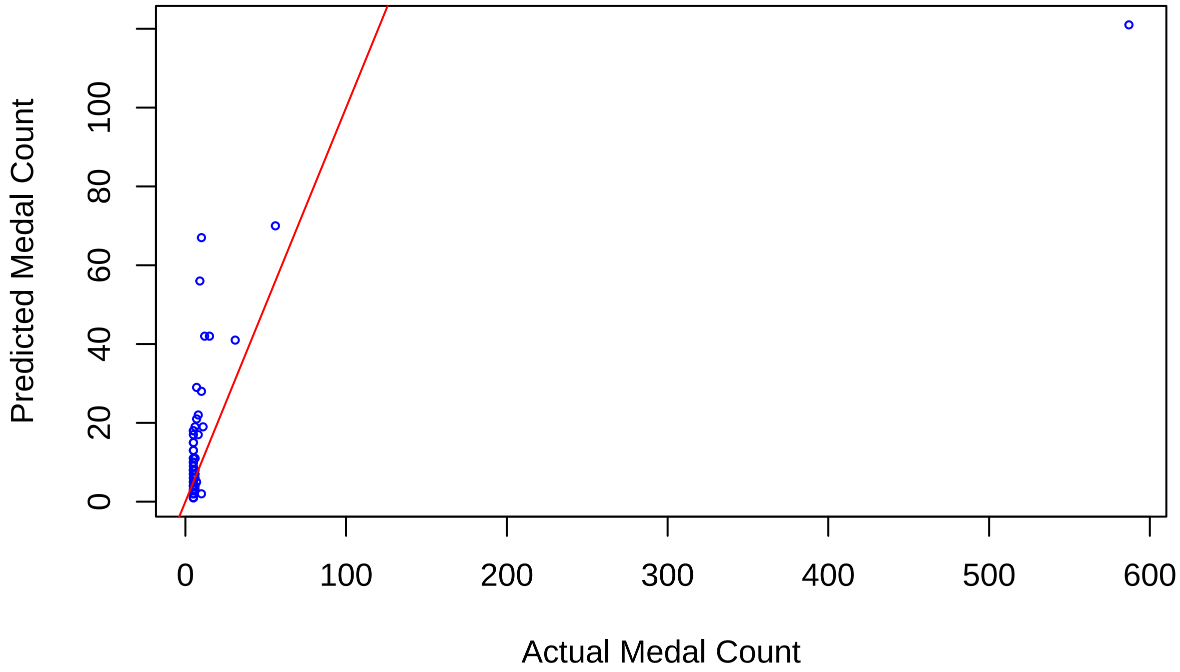
```
##
## Call:
## glm(formula = lg ~ Population + GDP, data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.73090  -0.75630   0.02616   0.77789   2.22198
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.569e+00  1.263e-01  12.422  < 2e-16 ***
## Population  1.105e-10  6.058e-10   0.182    0.856
## GDP         3.161e-04  6.170e-05   5.123 2.68e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for gaussian family taken to be 0.9376449)
##
##     Null deviance: 96.505  on 70  degrees of freedom
## Residual deviance: 63.760  on 68  degrees of freedom
## AIC: 201.85
##
## Number of Fisher Scoring iterations: 2
```

The log of Medal Count in 2012 was used as target variable to devlop a linear regression model with Population and GDP as input variables. Some of the observations were made from the model and they can stated as below:

- The AIC for the model is 201.85
- The intercept value is 1.569
- The regression coefficient value for Population is very low, which is $1.105e - 10$ and has P-value of 0.856 which is very high. This demonstrates that population has a minor impact on a country's medal count and is hence statistically unimportant.
- In contrast to Population, the GDP has a regression coefficient value of $3.161e - 04$ and has a p-value of $2.68e - 06$, which is less than 0.5 and hence is statistically significant. This means that one unit change in GDP produces approximately a $3.161e - 04$ increase in the target variable.

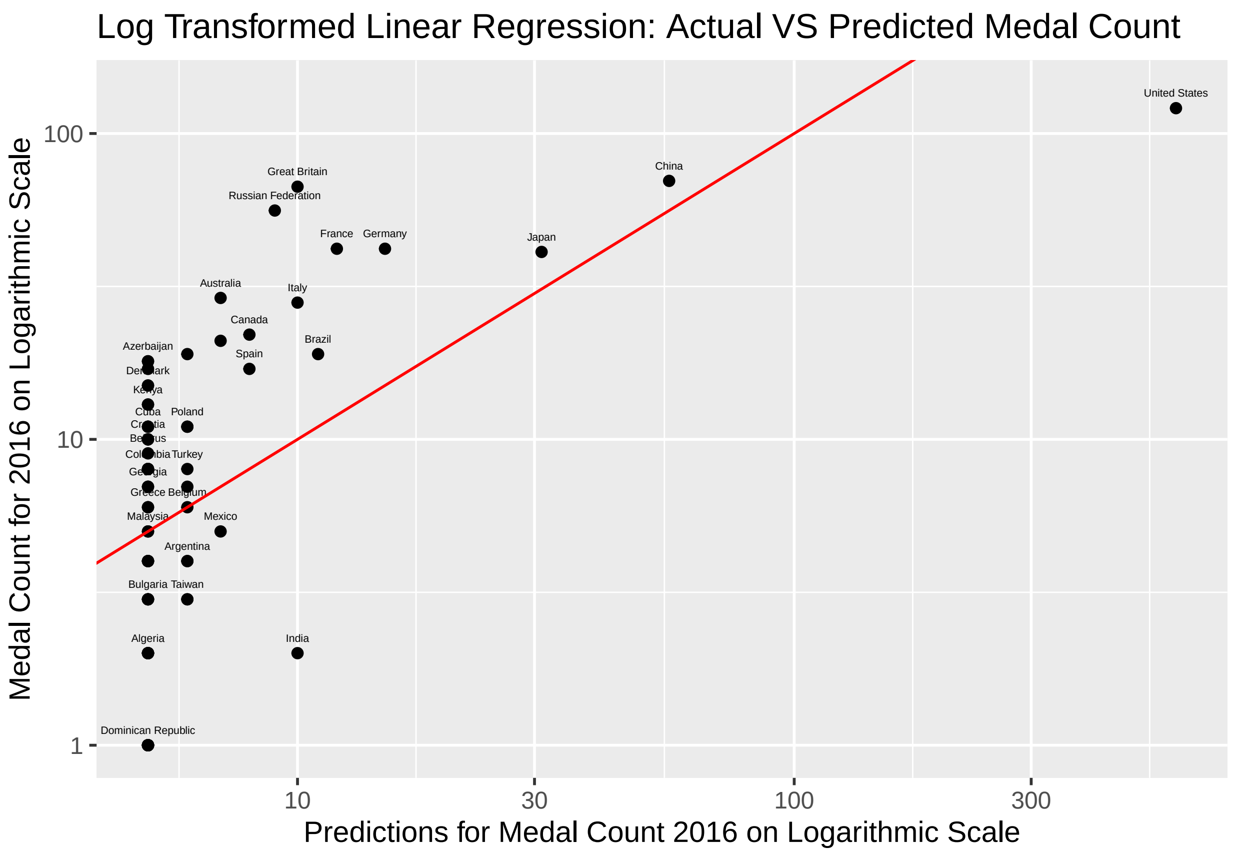
```r
#Creating a duplicate of original dataframe to store results of logarithmic model
LinearReg_model_Log = df

#Predicting the results and storing them in exponential form in the dataframe
log_predictions= exp(predict(Log_RegModel, newdata = new_df))
LinearReg_model_Log$log_predictions = round(log_predictions,digits = 0)
#Plotting the predicted VS actual value for medal count in 2016
plot(LinearReg_model_Log$log_predictions, LinearReg_model_Log$Medal2016, col="blue",
xlab = "Actual Medal Count", ylab = "Predicted Medal Count",cex = 0.5)
abline(a=0,b=1, col = "red")
```



The results observed here are similar to the one seen in Task 1. For better understanding of the plot, let us transform the axes by applying log.

```
ggplot(data=LinearReg_model_Log, aes(x=log_predictions, y=Medal2016,label = Country)) +
  geom_point() +
  scale_y_continuous(trans='log10')+
  scale_x_continuous(trans='log10')+
  xlab("Predictions for Medal Count 2016 on Logarithmic Scale")+
  ylab("Medal Count for 2016 on Logarithmic Scale")+
  ggtitle("Log Transformed Linear Regression: Actual VS Predicted Medal Count")+
  geom_text(size=1.5,nudge_y = 0.05,  check_overlap = TRUE)+
  geom_abline(slope = 1, intercept = 0, col = 'red')
```



From the above transformed graph, we can consider USA to be potential outlier as it lies far away from the imaginary line. With the application of log to the medal count it is seen that the count of majority of countries lies around 10.Let us have a look further at the outliers.

```
LinearReg_model_Log$DifferenceInCount <- (LinearReg_model_Log$Medal2016 - LinearReg_model_Log$log_predi
#sorting the dataframe to obtain outliers
LinearReg_model_Log = LinearReg_model_Log[with(LinearReg_model_Log, order(DifferenceInCount)),]
head(x = LinearReg_model_Log[, c(1,6,7,8)])
```

```
##                Country Medal2016 log_predictions DifferenceInCount
## 69      United States       121             587              -466
## 30              India         2              10                -8
## 19 Dominican Republic         1               5                -4
## 21            Estonia         1               5                -4
## 23            Finland         1               5                -4
## 42            Moldova         1               5                -4
```

```
tail(x = LinearReg_model_Log[, c(1,6,7,8)])
```

```
##                Country Medal2016 log_predictions DifferenceInCount
## 34              Italy        28              10                18
## 4           Australia        29               7                22
## 26            Germany        42              15                27
## 24             France        42              12                30
## 52 Russian Federation        56               9                47
## 27      Great Britain        67              10                57
```

A new column 'DifferenceInCount' is created to store the difference between the predicted medal count and actual medal count for the year 2016 to identify the outliers as performed in Task 1. In this case, we can say from the results below that United States, India, Dominican Republic have performed poorly in the Games. Whereas, countries like Great Britain, Russian Federation, France, Germany and Australia performed well, which are similar to the results observed in Task 1.

```
mae(LinearReg_model_Log$log_predictions,log(LinearReg_model_Log$Medal2016))
```

```
## [1] 13.27068
```

```
rmse(LinearReg_model_Log$log_predictions,log(LinearReg_model_Log$Medal2016))
```

```
## [1] 69.57934
```

```
mae(LinearReg_model_Log$log_predictions,LinearReg_model_Log$Medal2016)
```

```
## [1] 13.71831
```

```
rmse(LinearReg_model_Log$log_predictions,LinearReg_model_Log$Medal2016)
```

```
## [1] 56.60165
```

The mean absolute error (MAE) for the log transformed model is observed to be 13.71.This indicates that the average absolute difference between the observed values and the predicted values is 13.71. The RMSE for the model is 56.60 and indicates that the model is a decent.

The logarithmic transformation is most popular transformation used and some potential benefits are:

- to transform the skewed data to normal, hence makes the model a better fit for the data.
- It improves the linearity between the target and predictor variables.
- Boosts the validity of statistical analysis
- Help to reduce overfitting of the data.

Task 3:

Poisson Regression model are best used for modeling events where the outcomes are counts or more specifically the data is discrete with non-negative integer values.

```
Poisson_model = glm(Medal2012 ~ Population + GDP , data= df,family = poisson(link='log'))
summary(Poisson_model)
```

```
##
## Call:
## glm(formula = Medal2012 ~ Population + GDP, family = poisson(link = "log"),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7459  -2.8253  -1.4710   0.9333  12.4841
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.193e+00  4.034e-02  54.360  < 2e-16 ***
## Population  6.049e-10  9.131e-11   6.625 3.48e-11 ***
## GDP         1.715e-04  6.672e-06  25.708  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1331.81  on 70  degrees of freedom
## Residual deviance:  690.27  on 68  degrees of freedom
## AIC: 962.24
##
## Number of Fisher Scoring iterations: 5
```

The Poisson Regression model was created with Medal count in 2012 as target variable and Population and GDP as input variable. The following observations were made:

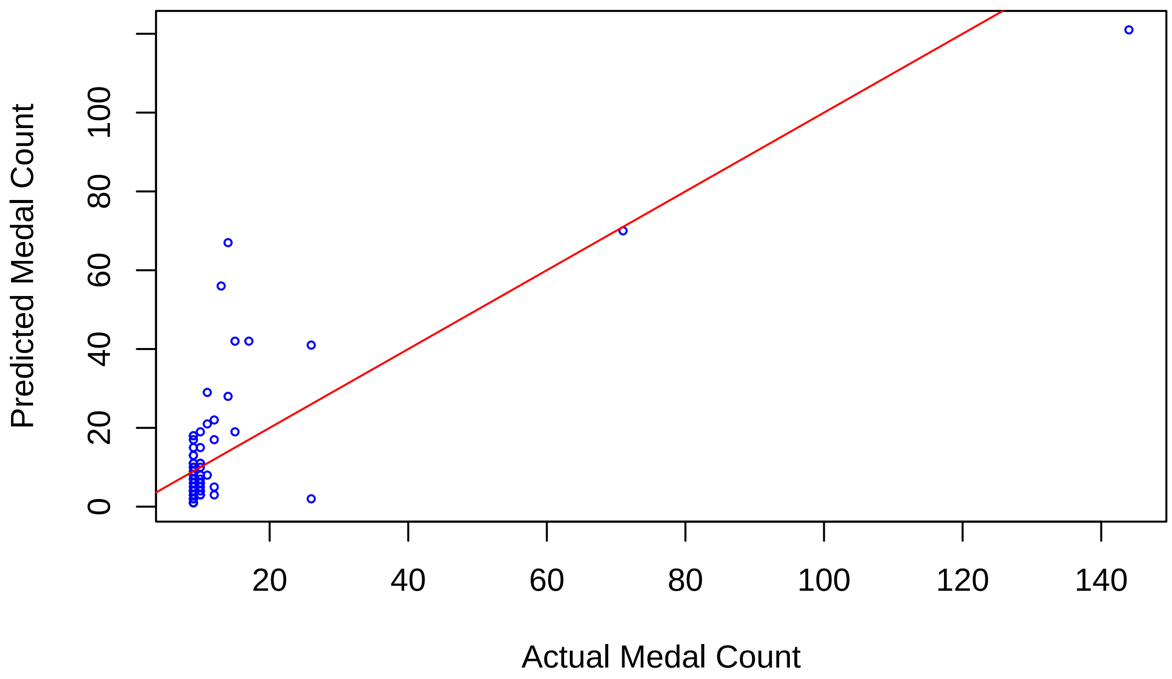
- The AIC returned by the model is 962.24
- The intercept value is 2.193
- The regression coefficient value for Population is very low, which is $6.049e - 10$ and has P-value of $3.48e - 11$ which is also very low.The low value makes the Population variable statistically important.
- In contrast to Population, the GDP has a regression coefficient value of $1.715e - 04$ and has a p-value of $2e - 16$, which is less than 0.5 and hence is statistically significant. This means that one unit change in GDP produces approximately a $2e - 16$ increase in the target variable.

```
#Creating a duplicate dataframe
Poisson_df = df

##Predicting the results and storing them in exponential form in the dataframe
Poisson_df$poi_predictions = round(predict(Poisson_model, newdata = new_df, type = "response"))
print(Poisson_df$poi_predictions)
```

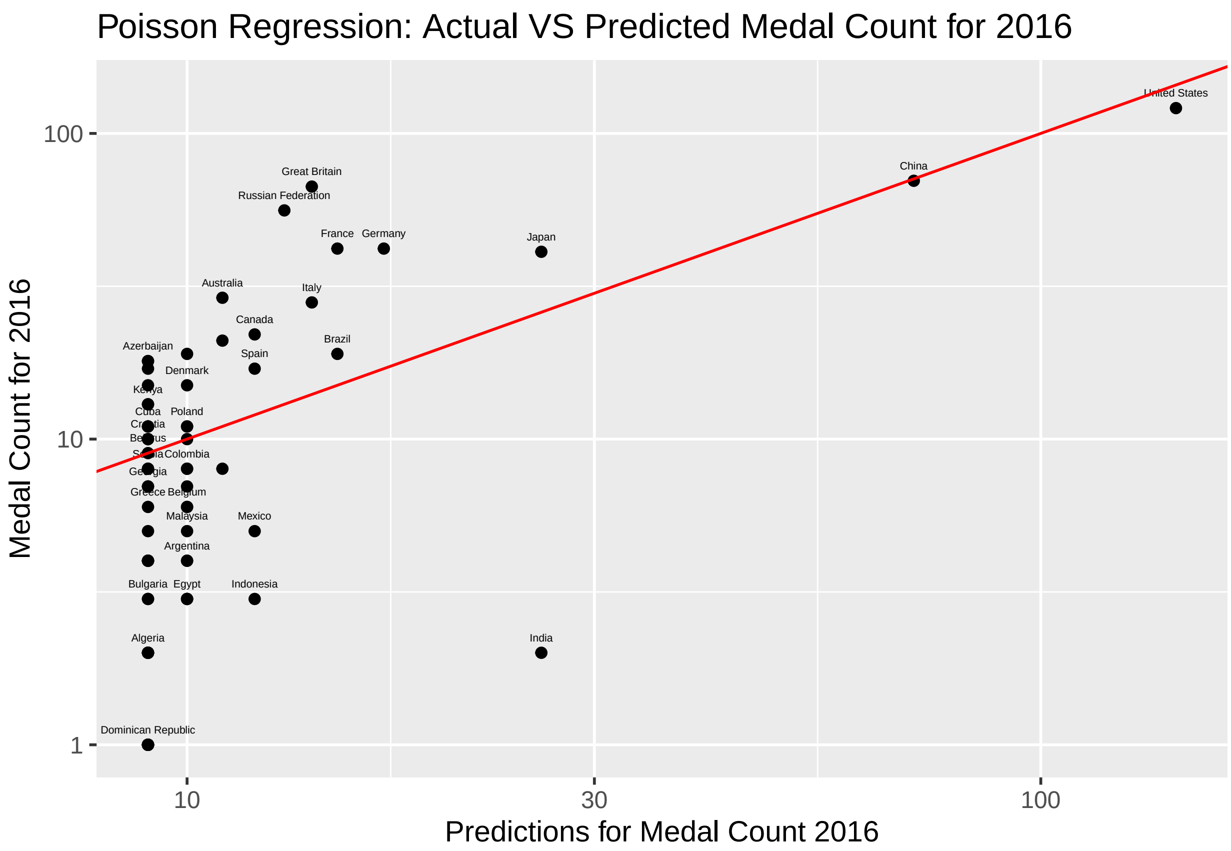
```
##  [1]   9  10   9  11   9   9   9   9  10  15   9  12  71  10   9   9   9  10   9
## [20]  10   9   9   9  15   9  17  14   9   9  26  12  10   9  14   9  26   9   9
## [39]   9  10  12   9   9   9  10   9   9  10  10   9   9  13   9   9   9   9  10
## [58]  11  12  10  10  10   9  10   9   9  11   9 144   9  10
```

```
plot(Poisson_df$poi_predictions, Poisson_df$Medal2016, col="blue", xlab = "Actual Medal Count", ylab =
abline(a=0,b=1, col = "red")
```



From the graph, it is evident that majority of the countries have won medals fewer than 20. To see the results more clearly, we have applied logarithmic transformation.

```
ggplot(data=Poisson_df, aes(x=poi_predictions, y=Medal2016,label = Country)) +
  geom_point() +
  scale_y_continuous(trans='log10')+
  scale_x_continuous(trans='log10')+
  xlab("Predictions for Medal Count 2016")+
  ylab("Medal Count for 2016")+
  ggtitle("Poisson Regression: Actual VS Predicted Medal Count for 2016")+
  geom_text(size=1.5,nudge_y = 0.05,  check_overlap = TRUE)+
  geom_abline(slope = 1, intercept = 0, col = 'red')
```



Here different set of countries are observed to be close to the imaginary line from those seen with Linear regression or Log transformed Linear Regression. With Poisson distribution countries like Colombia, China and United States have medal count close to the predicted medal count.

```
Poisson_df$DifferenceInCount <- (Poisson_df$Medal2016 - Poisson_df$poi_predictions)
#sorting the dataframe to obtain outliers
Poisson_df = Poisson_df[with(Poisson_df, order(DifferenceInCount)),]
head(x = Poisson_df[, c(1,6,7,8)])
```

```
##                 Country Medal2016 poi_predictions DifferenceInCount
## 30                India         2              26               -24
## 69        United States       121             144               -23
## 31            Indonesia         3              12                -9
## 19   Dominican Republic         1               9                -8
## 21              Estonia         1               9                -8
## 23              Finland         1               9                -8
```

```
tail(x = Poisson_df[, c(1,6,7,8)])
```

```
##                 Country Medal2016 poi_predictions DifferenceInCount
## 36                Japan        41              26                15
## 4             Australia        29              11                18
## 26              Germany        42              17                25
## 24               France        42              15                27
## 52   Russian Federation        56              13                43
## 27        Great Britain        67              14                53
```

A new column 'DifferenceInCount' is created to store the difference between predicted and actual medal count for 2016 to find outliers and India, United States and Indonesia looks like potential outliers that performed poorly and Great Britain, Russian Federation, France performed well at the games.

```
mae(Poisson_df$poi_predictions,Poisson_df$Medal2016)
```

```
## [1] 7.887324
```

```
rmse(Poisson_df$poi_predictions,Poisson_df$Medal2016)
```

```
## [1] 11.85594
```

The mean absolute error (MAE) for the log transformed model is observed to be 7.88.This indicates that the average absolute difference between the observed values and the predicted values is 7.88. The RMSE for the model is 11.85 and indicates that the model is a decent.