MATH5743M: Statistical Learning: Assignment 1

Aishwarya Selvaraj

2024-04-07

Prediction of the Olympic Games:

Introduction:

The Olympics is an event where athletes from all over the world come together to showcase their talent. It's a symbol of athletic achievement and national pride. The competition for dominance in medal tallies was evident between the US and the Soviet Union during the Cold War. The UK and Australia share a rivalry enriched by their sporting traditions. The importance of sportsmanship and fair play is highlighted during the Olympics. The success of a country in the Olympics is usually gauged by the number of medals won, but this measure often doesn't account for the size or economic power of a country.

This study aims to dive deeper into the predictive factors behind Olympic success. By employing statistical models in R, exploring how well a country's population and GDP can forecast the number of Olympic medals won. By critically assess the consistency of these relationships across different Olympics to understand the influence of economic and demographic variables on Olympic outcomes.

PreProcessing:

Preprocessing steps ensures that the dataset is reliable and prepared for further analysis, such as fitting regression models to predict the number of medals a country might win based on its population and GDP.

Firstly loading the dataset and necessary library to my R-Studio:

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

Checking for any missing values in the dataset which could skew analysis and lead to incorrect conclusions:

```
sum(is.na(medal_data))
```

```
## [1] 0
```

Seems like no missing values, which will increase the efficiency of the analysis.

Removing duplicate rows if any:

```
medal_data <- medal_data[!duplicated(medal_data), ]
```

Explorartory Data Analysis:

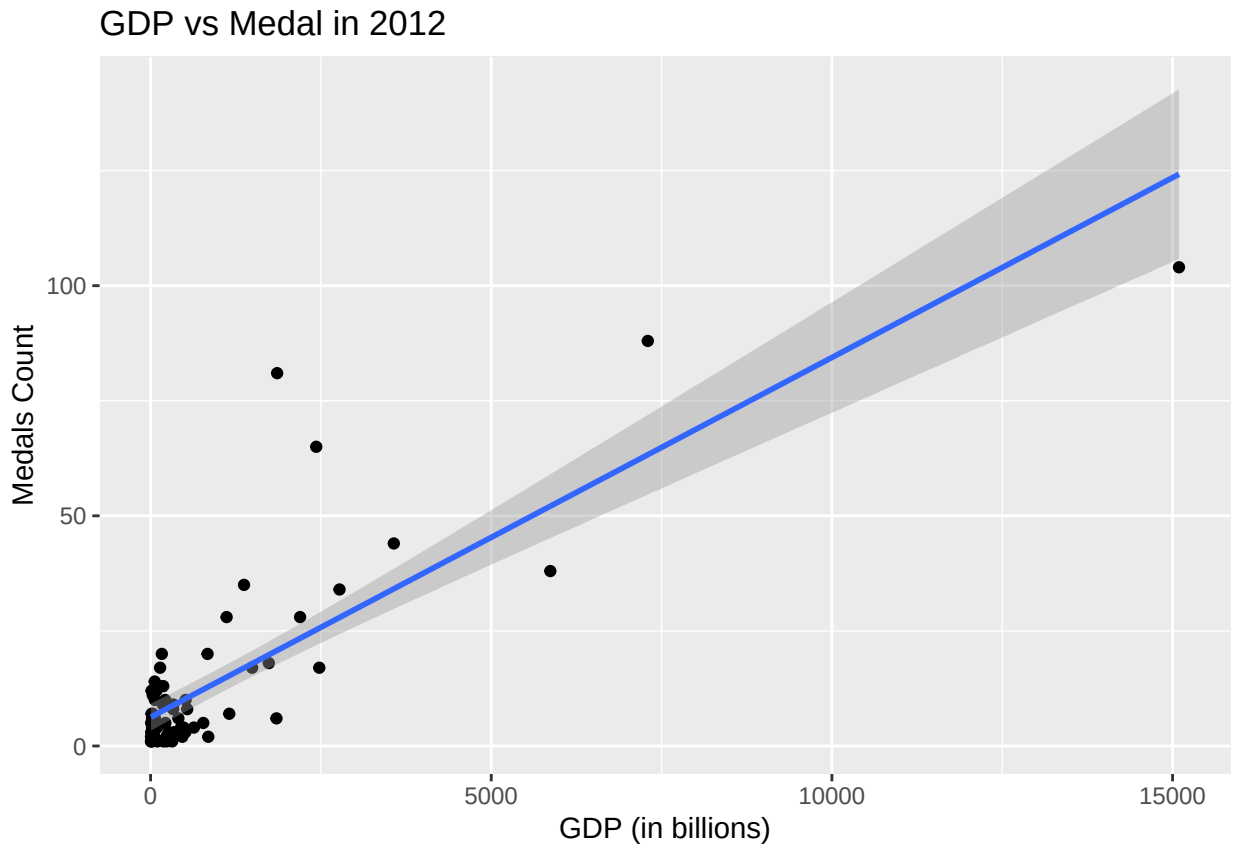
GDP Vs. Medals2012:

Using ggplot2 to create a scatter plot of GDP versus the number of medals won in 2012. geom_point() adds points for each observation, and geom_smooth(method=lm) adds a linear regression line to the plot. The labs() function adds labels to the plot, including a title and axis labels.

```
#importing ggplot library
library(ggplot2)
```

```
#Specifying the x and y axis
#geom_point adds layer representing data points and dots
#geom_smooth to add smooth line representing linear fit
ggplot(medal_data, aes(x=GDP, y=Medal2012)) + geom_point() +
  geom_smooth(method=lm) +
  labs(title="GDP vs Medal in 2012", x="GDP (in billions)", y="Medals Count")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



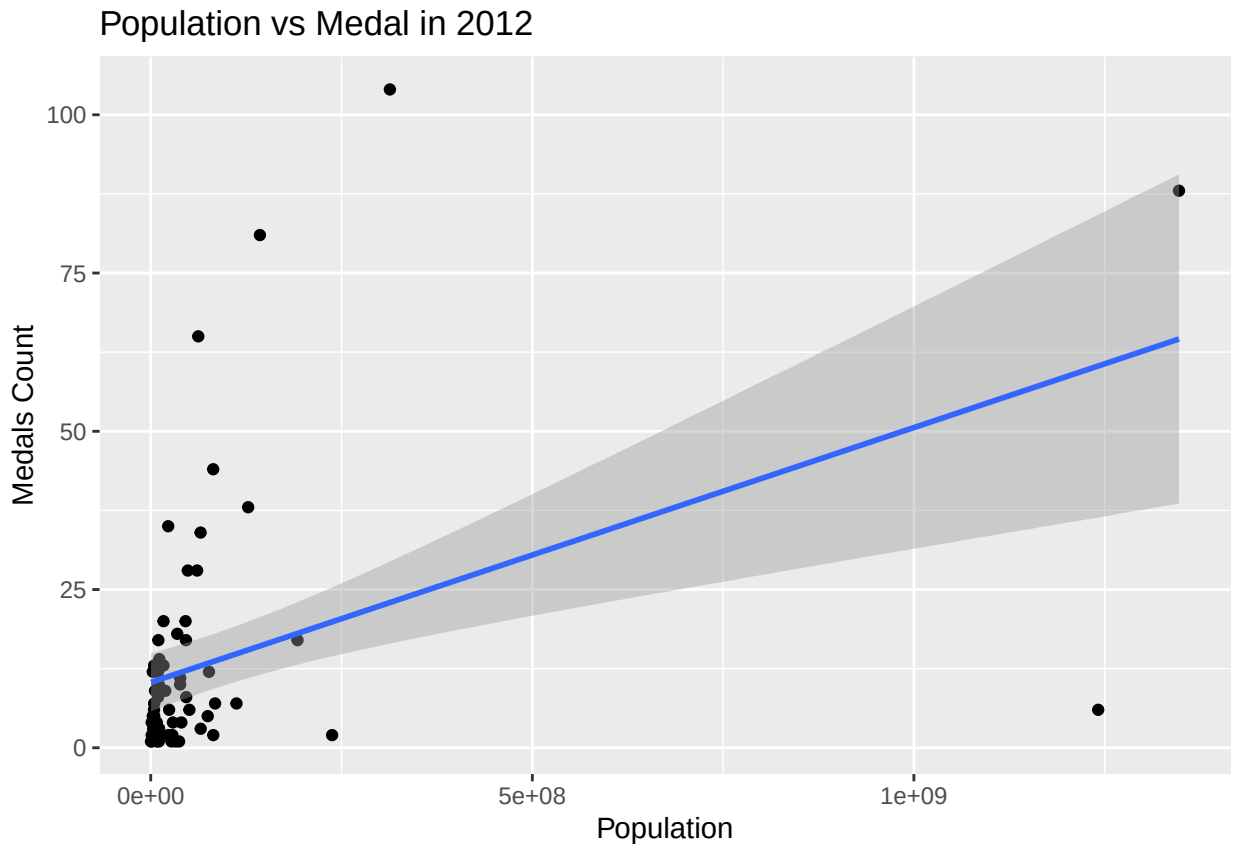
Above displayed image is a scatter plot that visualizes the relationship between GDP (in billions) and the number of medals won in 2012. Each point on the plot represents a different country, with the country's GDP on the x-axis and its medal count on the y-axis. A linear model has been fitted to the data, indicated by the blue line, with the gray area around it representing the confidence interval for the line of best fit. The plot suggests a positive linear relationship, indicating that countries with higher GDPs tend to win more medals.

Population Vs. Medals2012:

Similar to the previous plot, but visualizing the relationship between Population and the number of medals won in 2012 with a linear regression line.

```
ggplot(medal_data, aes(x=Population, y=Medal2012)) + geom_point() +
  geom_smooth(method=lm) +
  labs(title="Population vs Medal in 2012", x="Population", y="Medals Count")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Similar to the previous GDP plot, each point represents a country, with population on the x-axis and medal count on the y-axis. The linear fit line (in blue) suggests there is a positive relationship between population size and the number of medals won, with a wider confidence interval (shown by the grey shaded area) than what we saw with GDP. This could indicate more variability in the number of medals won by countries with similar population sizes compared to those with similar GDP's.

Correlation:

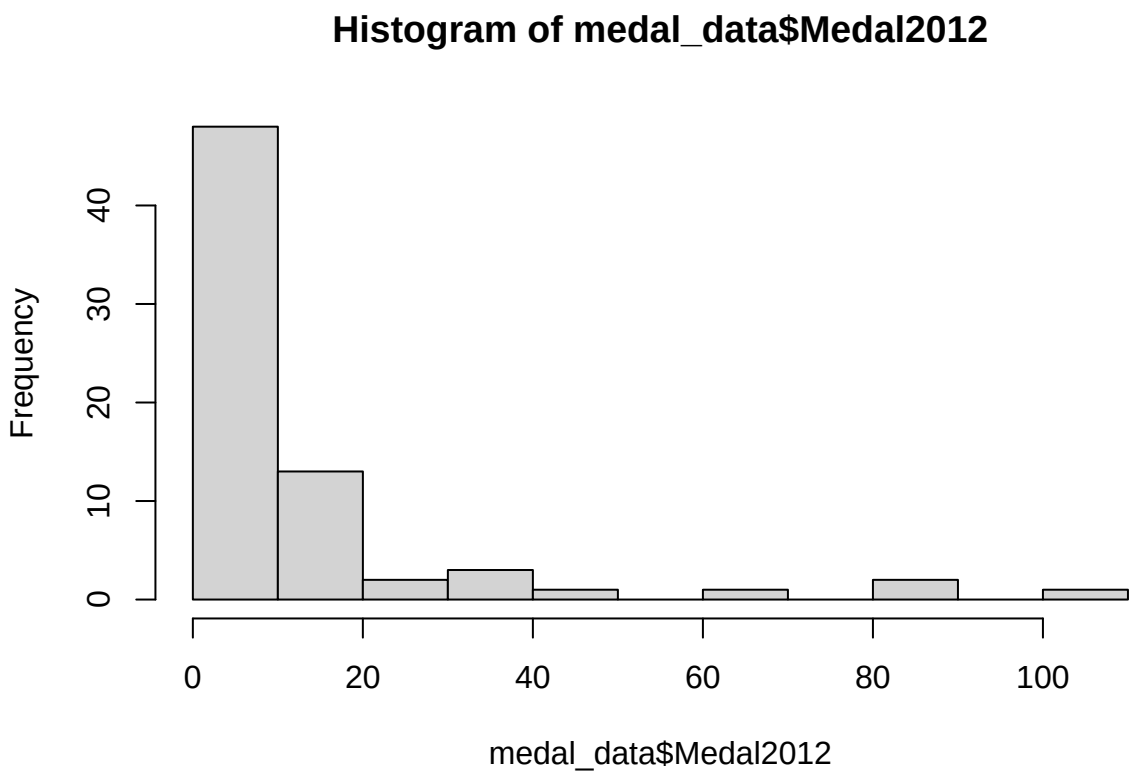
Checking the correlation before proceeding with the analysis:

```
cor(medal_data$GDP, medal_data$Population)
```

```
## [1] 0.4714933
```

The magnitude of 0.471, while not very close to 1, is significant enough to suggest that there is a moderate linear relationship. It's not strong enough to suggest a high predictability based on this relationship alone.

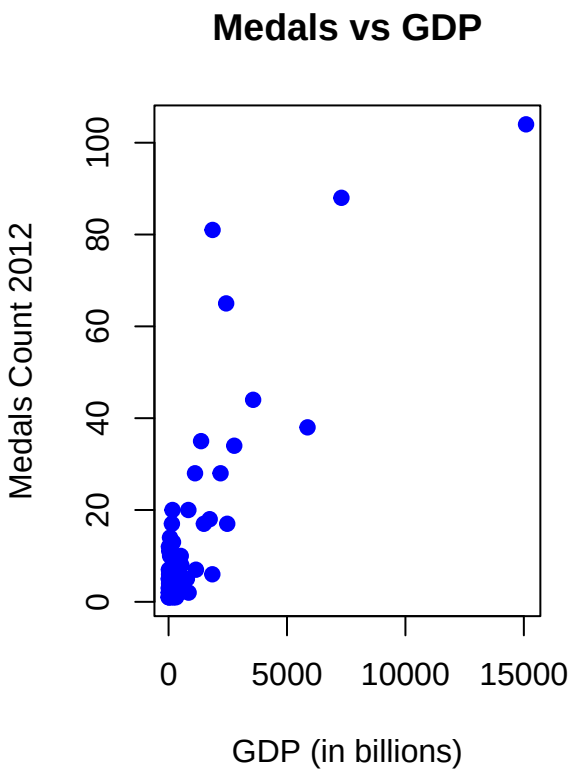
```r
hist(medal_data$Medal2012)
```

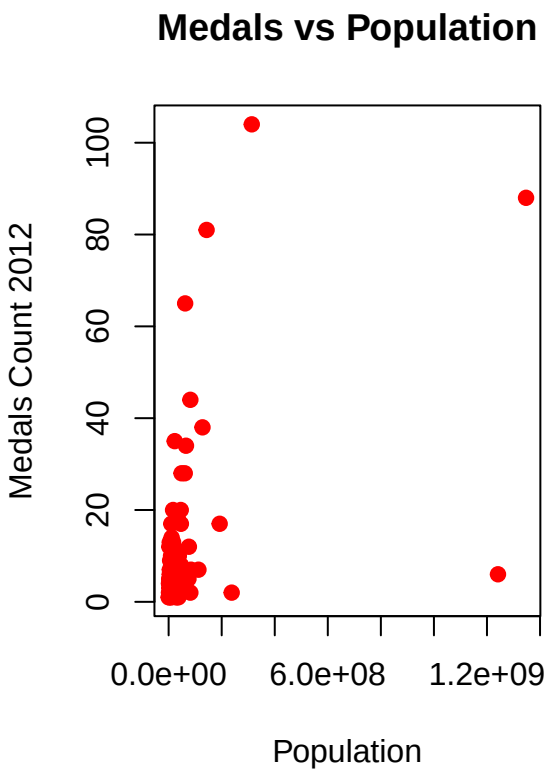


The distribution of medals is right-skewed (positive skew). Given the skewness of the distribution, transforming the dependent variable (e.g., using a logarithmic transformation) might be necessary when using medal counts as a response variable in regression analysis to meet the assumption of normality of residuals(which will be covered in Task2).

Linearity:

```r
par(mfrow = c(1, 2))
plot(medal_data$Medal2012 ~ medal_data$GDP, main = "Medals vs GDP",
xlab = "GDP (in billions)",ylab = "Medals Count 2012", pch = 19, col = 'blue')
plot(medal_data$Medal2012 ~ medal_data$Population, main = "Medals vs Population",
xlab = "Population",ylab = "Medals Count 2012", pch = 19, col = 'red')
```






```
par(mfrow = c(1, 1))
```

Although the relationship between Population and Medal2012 is bit less clear than the relationship between GDP and Medal2012, it quite appears linear to proceed with Linear Regression.

Task 1:

Population and GDP as inputs and medal count in 2012 Olympics as outputs for linear regression model:

To perform validation more robustly proceeding with cross- validation. For K fold cross- validation is divided K equally sized folds. Then for each fold K set the K^{th} fold as the validations set and remaining $K-1$ as training set.

seed() function sets for R's random number generator, ensuring that the results of the random processes like splitting data for cross-validation are reproducible. *trainControl*()sets up a control object for training a model using 10-fold cross-validation, which is a method for evaluating the model's predictive performance. Involved *train*() function from the caret package to fit a linear regression model, trains a linear regression model (lm) on the medal_data dataset, using Medal2012 as the response variable and Population and GDP as predictors. The training process is guided by the previously defined train_control.

```
# Training a linear regression model with cross-validation
library(caret)
library(dplyr)
```

```
set.seed(123)
train_control <-trainControl(method="cv", number=10)
model_cv <- train(Medal2012 ~ Population + GDP, data = medal_data, method = "lm", trControl = train_con
```

Prints the results of the cross-validated linear model:

```
# Printing the cross-validated model results
print(model_cv)
```

```
## Linear Regression
##
## 71 samples
##  2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 63, 64, 63, 65, 62, 65, ...
## Resampling results:
##
##   RMSE     Rsquared   MAE
##   12.7727  0.7309103  8.246891
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

The results show a linear regression model that has been cross-validated with 10 folds. The summary of the cross-validation results provides the root mean squared error (RMSE), R-squared, and mean absolute error (MAE) metrics, which are common measures of model performance.

These metrics collectively describe the model's predictive accuracy and fit quality. The high R-squared value implies a strong linear relationship between the predictors and the response variable as captured by the model. The RMSE and MAE values provide context for the absolute fit of the model to the data, indicating the typical size of the prediction errors.

Fitting a Linear Regression Model:

Now, fitting a linear regression model to the entire dataset without cross-validation. This will be the final model that will interpret.

```
# Training the final linear regression model
final_model <- lm(Medal2012 ~ Population + GDP, data = medal_data)
```

Summary of final linear model, including the coefficients, their significance, R-squared value, and other statistical measures.

```
# Summarizing the final model to get detailed results
summary(final_model)
```

```
##
## Call:
## lm(formula = Medal2012 ~ Population + GDP, data = medal_data)
##
## 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
##
## Residual standard error: 11.5 on 68 degrees of freedom
## Multiple R-squared:  0.6836, Adjusted R-squared:  0.6743
## F-statistic: 73.46 on 2 and 68 DF,  p-value: < 2.2e-16
```

Residuals:

The residuals indicate the differences between the observed values and the model's predicted values. The range of residuals shows some spread, from a minimum of -20.568 to a with a maximum of 60.121, which might suggest the presence of outliers or that the relationship is not perfectly linear.

Coefficients:

1. Intercept: The estimated intercept is around 6.076, which represents the expected number of medals when both Population and GDP are zero.
2. Population: The coefficient for Population is approximately 5.247Ö$10^{-}9$, but it is not statistically significant $p - value = 0.468225$, suggesting that Population alone may not be a good predictor of Medal2012 within this model.
3. GDP:The coefficient for GDP is about 7.564Ö$10^{-}3$ and is statistically significant $p - value = 1.45Ö10^{-15}$), indicating a strong positive relationship between GDP and the number of medals won. Where an increase in GDP by one billion USD is associated with an increase in medals won by approximately 7.564.

Significance Codes:

The asterisks denote the level of significance for each coefficient, with GDP being highly significant (indicated by "***").

Residual Standard Error:

An RSE of 11.5 suggests that on average, the model's predictions deviate from the actual number of medals by about 11.5 medals.

R-squared:

An R-squared value of 0.6836 means that about 68.36% of the variability in medal counts is explained by the model. The Adjusted R-squared of 0.6743 is a modified version of R-squared that takes the number of predictors into account, providing a more accurate measure for models with multiple variables.

F-statistic:

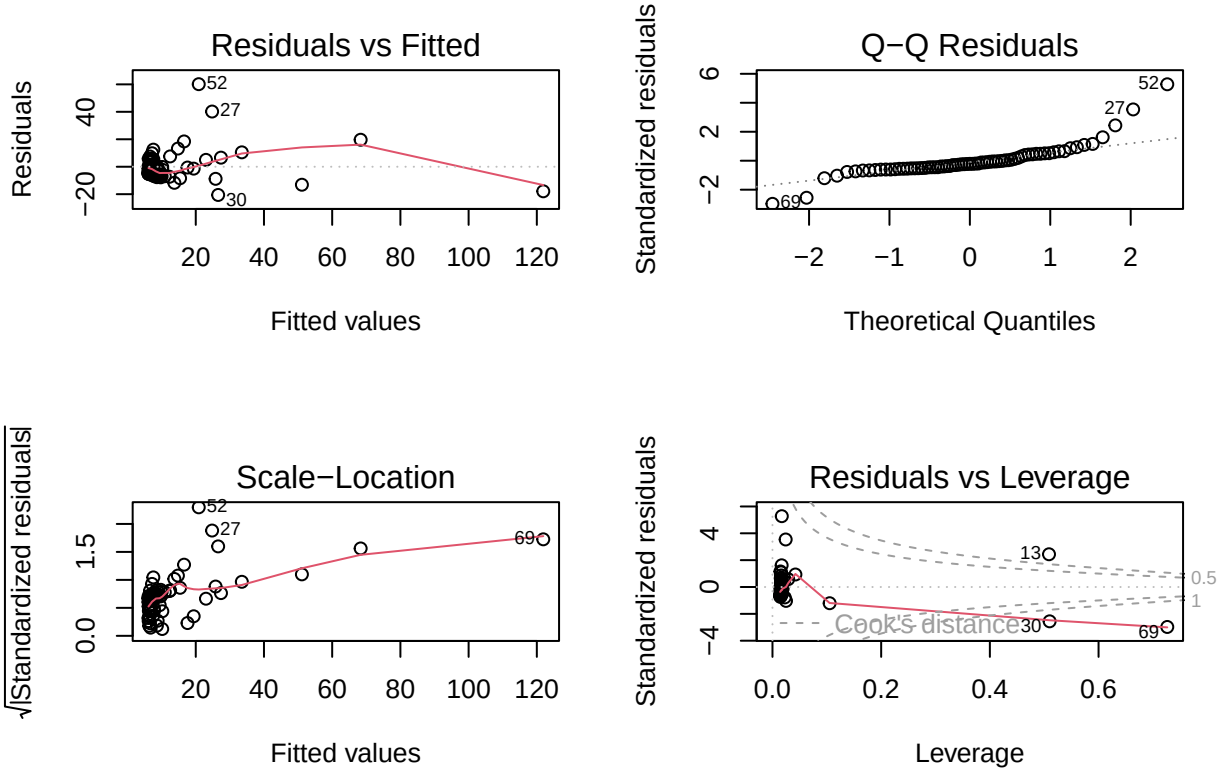
The F-statistic is 73.46, and with a very low p-value (practically zero), can proceed on rejecting the null hypothesis that all regression coefficients are zero, indicating that the overall model is statistically significant. The very low $p - value$ ($< 2.2e^{16}$) here also suggests that the model is statistically significant; that is, the model provides a better fit than a model with no predictors.

In summary, the model suggests that GDP is a significant predictor of Olympic medal counts in 2012, while population size may not be as important in this particular model. The goodness-of-fit measures indicate that the model explains a substantial proportion of the variation in the data, but further diagnostics would be beneficial to ensure that the assumptions of linear regression have been met and to investigate the influence of any outliers or leverage points.

Plotting:

2 by 2 layout plots are used to check the assumptions of the linear regression, such as linearity, normality of residuals, homoscedasticity, and absence of influential outliers.

```
par(mfrow=c(2,2))
plot(final_model)
```



Residuals Vs. Fitted:

Residuals Vs. Fitted plot is used to check the assumption of linearity and homoscedasticity. A horizontal line with equally spread residuals above and below it would suggest a good fit. In this plot, the residuals do not appear to be randomly distributed around a horizontal line, indicating potential issues with linearity or homoscedasticity.

Normal Q-Q:

The quantile-quantile plot is used to assess whether the residuals are normally distributed. Points following the straight line indicate normality. The plot shows that while most residuals seem to follow the line, there are some deviations at the ends, suggesting some departure from normality.

Scale-Location:

Also known as a Spread-Location plot, it's used to check homoscedasticity. Ideally, you would want to see a horizontal line with equally spread points. The plot indicates that the variance of residuals might be increasing with the fitted values, suggesting possible heteroscedasticity.

Residuals Vs. Leverage:

Residuals Vs. Leverage helps identify influential observations. Points that stand out far from the rest of the points could be influential. The Cook's distance lines help to visualize how much influence each point has.

A few points stand out, but they seem to be within acceptable Cook's distance, suggesting they might not be overly influential.

The provided diagnostic plots from the linear regression model indicate potential issues with the assumptions required for an ideal model: the residuals suggest possible non-linearity and heteroscedasticity, and the normal Q-Q plot reveals some deviation from normality. While these diagnostics do not render the model unusable, they highlight the need for caution in interpretation and potential model refinements, such as variable transformations or the inclusion of interaction terms. Addressing outliers and influential points may also help improve the model's accuracy and the validity of its assumptions.

Task 2:

Log-transformed outputs to address the skewness:

The log transformation is commonly used in regression models to normalize the distribution of the response variable and stabilize the variance of residuals. The use of cross-validation provides a rigorous assessment of the model's predictive power.

Adding a new column to medal_data where the 2012 medal counts are log-transformed to address the skewness of the distribution. Adding 1 before taking the log ensures there are no issues with taking the logarithm of 0, which is undefined.

```
# Applying log transformation to the medal counts. Adding 1 to avoid log(0)
medal_data$Log_Medal2012 <- log(medal_data$Medal2012 + 1)
```

Setting up a control object for training the model that specifies the use of 10-fold cross-validation. This means the dataset will be divided into 10 parts, the model will be trained on 9 and validated on the 1 remaining part, and this process will be repeated 10 times.

```
# Set up 10-fold cross-validation controls
train_control <- trainControl(method = "cv", number = 10)
```

Training a linear model using the train function from the caret package, which uses the cross-validation parameters defined in train_control. This model uses the log-transformed medal counts as the response variable.

Printing the summary of a log transformed outputs:

```
# Fit a linear regression model using the log-transformed medal counts
model_cv_log <- train(Log_Medal2012 ~ Population + GDP, data = medal_data,
method = "lm", trControl = train_control)
# Print the cross-validated model results
print(model_cv_log)
```

```
## Linear Regression
##
## 71 samples
##  2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 65, 65, 64, 64, 62, 63, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.9153341  0.5102223  0.7297488
```

```
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

The output from the linear regression analysis indicates that after log-transforming the medal counts and using 10-fold cross-validation, the model achieved an RMSE of 0.9153341, an R-squared of 0.5102223, and an MAE of 0.7297488.

Discussing potential benefits and reasons:

The RMSE and MAE are relatively low, indicating small average errors in the model's predictions. The R-squared value suggests that over 50% of the variability in the log-transformed medal count can be explained by the model, which is a reasonable amount considering the complexity of factors influencing Olympic success. These results, particularly the improvement in R-squared from previous untransformed models, justify the use of log transformation and indicate that the model has good predictive accuracy.

Further Diagnostic Check:

Summary() provides detailed summary of the fitted model, including the regression coefficients, their standard errors, t-statistics, p-values, and key model diagnostics like R-squared and F-statistic.

```r
# Fitting the final model on the entire dataset using the log-transformed medal counts
final_model_log <- lm(Log_Medal2012 ~ Population + GDP, data = medal_data)
# Summarize the final model to get detailed results
summary(final_model_log)
```

```
##
## Call:
## lm(formula = Log_Medal2012 ~ Population + GDP, data = medal_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.52067 -0.65159 -0.03983  0.63218  2.04300
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.819e+00  1.041e-01  17.470  < 2e-16 ***
## Population  8.197e-11  4.993e-10   0.164     0.87
## GDP         2.869e-04  5.085e-05   5.641 3.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.798 on 68 degrees of freedom
## Multiple R-squared:  0.3822, Adjusted R-squared:  0.3641
## F-statistic: 21.04 on 2 and 68 DF,  p-value: 7.726e-08
```

The output is a summary of the linear regression model where the response variable, the log-transformed Olympic medal counts Log_Medal2012, is modeled as a function of Population and GDP. The residuals, the differences between observed and predicted log-medal counts, range from -1.52067 to 2.04300, with a median close to zero, suggesting that the model does not have systematic bias. The coefficients for Population and GDP are provided with their standard errors, t-values, and p-values. The intercept and GDP are significant predictors, whereas Population is not. The residual standard error is 0.798, which is the average distance of the data points from the fitted line on the log scale. The model explains about 38.22 of the variance in the log-transformed medal counts, as indicated by the multiple R-squared value.

Brief:

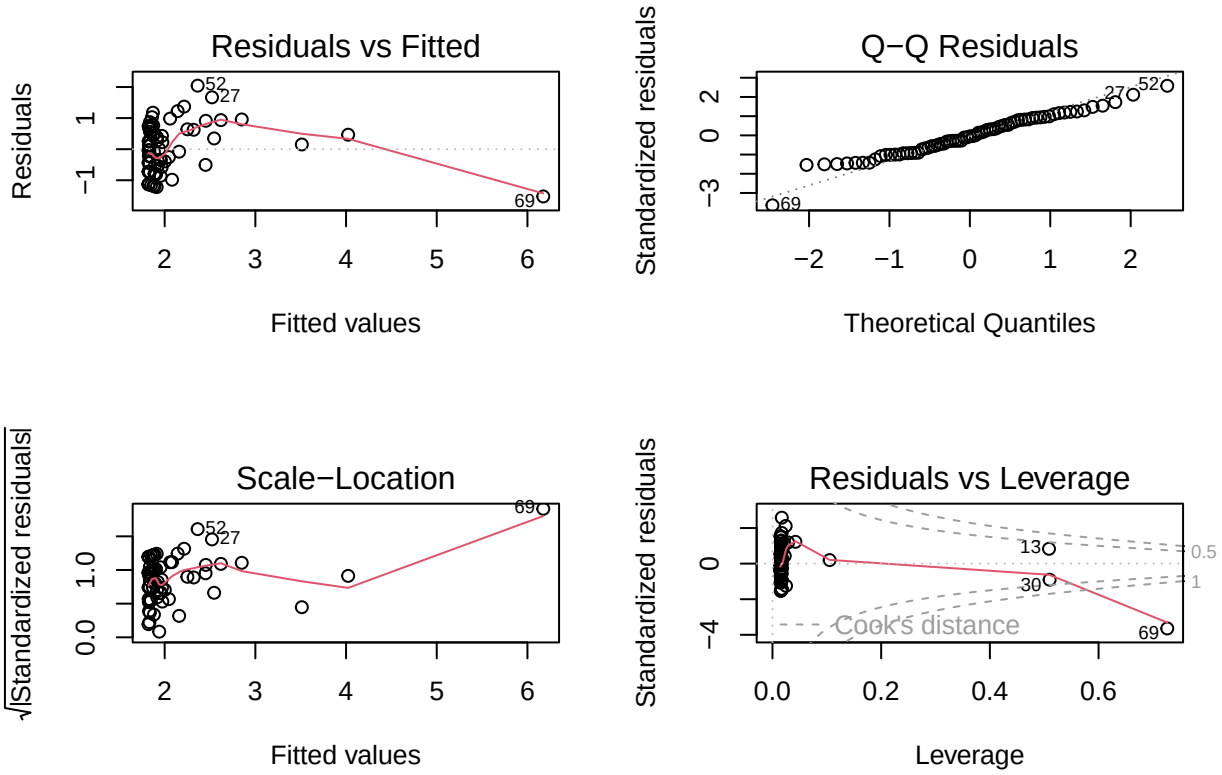
- The statistical significance of the GDP coefficient with a p-value of $3.56e-07$ justifies its inclusion in the model, suggesting a strong association between a country's GDP and its Olympic success on a logarithmic scale.
- The non-significance of the Population coefficient p-value of 0.87 suggests that once GDP is accounted for, additional variation in the medal count explained by Population is not statistically significant in this model.
- The multiple R-squared value of 0.3822, though not exceedingly high, indicates that a substantial portion of the variability in the log-transformed medal counts has been captured by the model, but there is still unexplained variance that might be captured by other factors or by a non-linear model.
- The summary provides justification for the model in terms of its fit to the data, but the relatively low R-squared value and the presence of non-significant predictors indicate that there might be room for model improvement. Additionally, this summary alone does not address whether the model's assumptions have been fully met; this would require further diagnostic checks.

Plotting:

Fitting a linear model to the log-transformed 2012 Olympic medal count data, with population and GDP as predictors. The graphics layout to display four plots on a 2 by 2 grid have been set.

```
# Fit the final model on the entire dataset using the log-transformed medal counts
final_model_log <- lm(Log_Medal2012 ~ Population + GDP, data = medal_data)

# Generate diagnostic plots for the log-transformed model
par(mfrow=c(2,2))  # Set up the plotting area to display a 2x2 grid of plots
plot(final_model_log)  # Produce the diagnostic plots
```



The diagnostic plots provide a visual assessment of the linear regression model's assumptions after log transformation of the response variable.

Residuals Vs. Fitted:

Residuals Vs. Fitted plot still shows some curvature, suggesting that even after the log transformation, the relationship between the predictors and the response variable may not be completely linear. However, the curvature seems less pronounced compared to the untransformed model, indicating an improvement.

Normal Q-Q:

Similar to the untransformed model, the points follow the line in the center section but deviate at the ends. The deviations seem slightly less severe, suggesting that the log transformation may have improved the normality of the residuals, but some deviation still exists.

Scale-Location:

The variance of the residuals still increases with the fitted values, although it might be slightly less apparent than in the untransformed model. This plot suggests that heteroscedasticity is still present, although the effect might be reduced.

Residuals vs Leverage:

The log transformation doesn't seem to have a large effect on the leverage plot. There are still a few points with higher leverage, but their Cook's distance remains within the acceptable range, indicating that they may not be unduly influencing the model.

Discussing potential benefits and reasons for using the transformation:

The log transformation appears to have reduced the pattern in the residuals, indicating an improvement in the linearity of the relationship. There is still evidence of non-normality and heteroscedasticity, but to a lesser extent. Influential points appear similar between the two models. The log transformation has provided some improvements in the model assumptions, but not all issues have been resolved. These diagnostics suggest that further model refinement is necessary, which could involve exploring non-linear relationships or other transformations. The presence of influential points that are not mitigated by the log transformation could warrant further investigation, perhaps requiring a more robust regression technique or the removal of outliers if justified.

Task 3:

To proceed with Task 3 exploring and comparing multiple models including a polynomial linear regression, a Poisson regression, and a negative binomial regression to determine the best fit for predicting Olympic medal counts based on Population and GDP. Setting up each model, comparing them using AIC, and by providing diagnostics to assess their fit.

```
# Load necessary libraries
library(MASS)
library(stats)
```

Polynomial Linear Regression:

Captures non-linear relationships by including squared terms of predictors. Effective for modeling curvature in data relationships that simple linear regression cannot.

```
# Fit a Polynomial Linear Regression Model (including quadratic terms)
poly_model <- lm(Medal2012 ~ Population + I(Population^2) + GDP + I(GDP^2), data = medal_data)
summary(poly_model)
```

Poisson Regression:

Assumes the response variable has a Poisson distribution and models count data based on log-link function. Suitable for count data where mean equals variance (no over-dispersion).

```r
# Fit a Poisson Regression Model
poisson_model <- glm(Medal2012 ~ Population + GDP, family = poisson(), data = medal_data)
summary(poisson_model)
```

Negative Binomial Regression:

Extends Poisson regression by adding a parameter to account for over-dispersion, where the variance exceeds the mean. Best for count data with over-dispersion.

```r
# Fit a Negative Binomial Regression Model
nb_model <- glm.nb(Medal2012 ~ Population + GDP, data = medal_data)
summary(nb_model)
```

AIC measures the quality of each model, relative to each of the others. A lower AIC value indicates a better model. AIC rewards goodness of fit but also includes a penalty that increases with the number of estimated parameters, which helps to prevent overfitting.AIC not only compares the goodness of fit but also adjusts for the model's complexity.

```r
# Calculate AIC values
aic_poly <- AIC(poly_model)
aic_poisson <- AIC(poisson_model)
aic_nb <- AIC(nb_model)
# Create a data frame to display AIC comparisons
aic_comparison <- data.frame(
  Model = c("Polynomial Linear", "Poisson", "Negative Binomial"),
  AIC = c(aic_poly, aic_poisson, aic_nb)
)
# Display the AIC values
print(aic_comparison)
```

```
##               Model      AIC
## 1 Polynomial Linear 545.6235
## 2           Poisson 962.2420
## 3 Negative Binomial 475.9745
```

Properties Addressed:

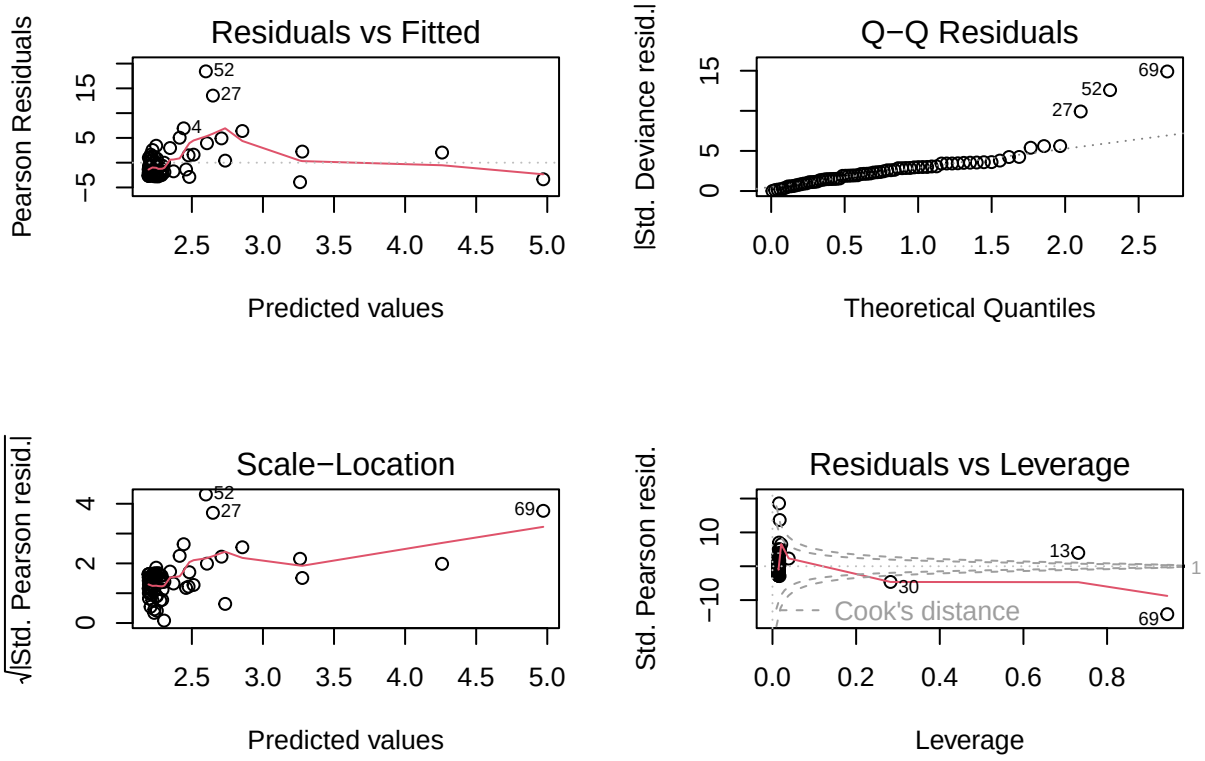
1. Over-Dispersion: Both Poisson and Polynomial models may not handle over-dispersion inherently present in medal counts effectively, potentially leading to inefficient, biased estimates that could misinform the predictions. Negative Binomial regression specifically accounts for over-dispersion.
2. Non-linearity: Polynomial regression addresses non-linearity but does not handle count data properties like over-dispersion, which are crucial for data involving counts like medal tally.
3. Zero-inflation: Not specifically modeled by any of these three, but Negative Binomial can better accommodate excess zeros than Poisson through its dispersion parameter.
4. Right-Skewed Distribution: The distribution of medal counts is highly skewed to the right, meaning most of the data points (countries) have low medal counts, with a few outliers having very high counts.
5. Non-linear Relationships: The relationships between predictors (Population and GDP) and the response variable (medal counts) are likely non-linear, with increases in GDP and Population potentially having diminishing returns in terms of medal count increments.

Model Development and Explanation:

Negative Binomial Regression is chosen due to its ability to handle over-dispersion effectively. Unlike the Poisson regression, which assumes the mean equals the variance, the negative binomial model has an extra parameter to account for the variance, making it more flexible for data like Olympic medal counts. The logarithm link function used in this model is standard for count data and helps in modeling the multiplicative effects of the predictors. *Over-dispersion Parameter (Theta)* model's flexibility in adjusting the dispersion parameter helps it to fit the data better than a Poisson model, particularly when dealing with high variance typical to count data distributions like that of Olympic medals.

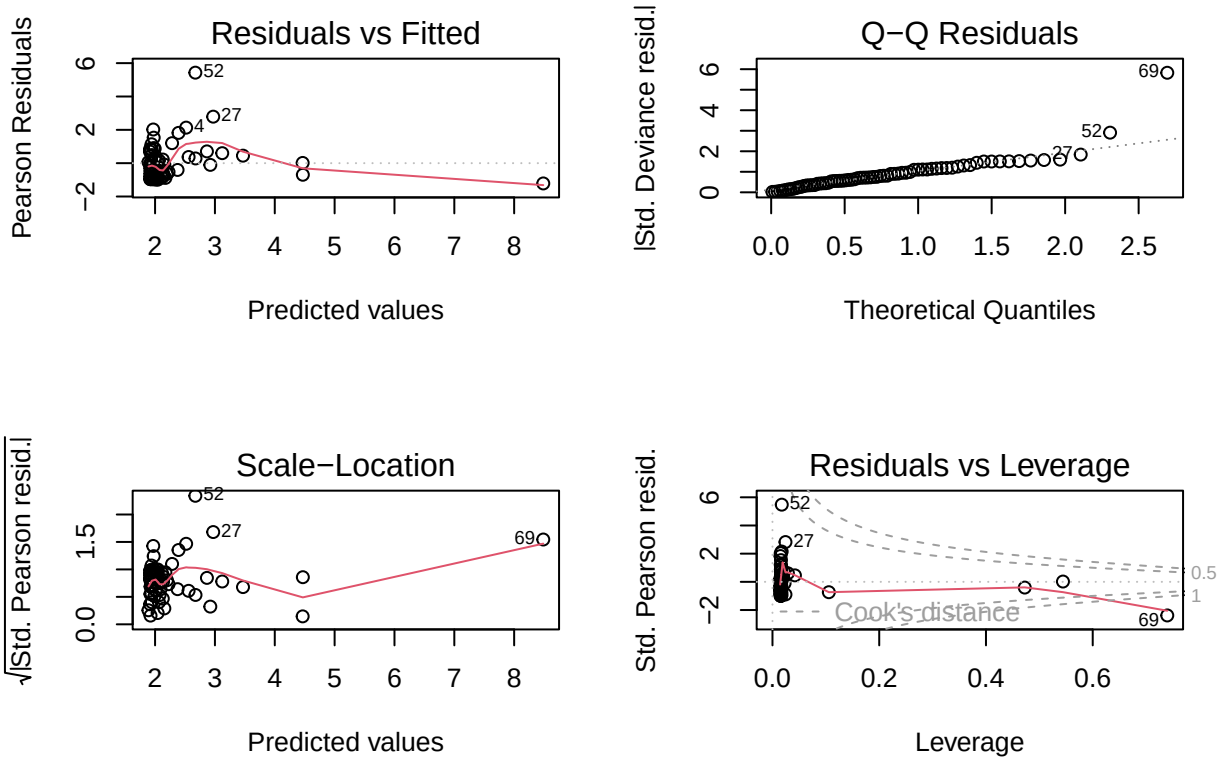
Plotting:

```
par(mfrow=c(2,2))
plot(poisson_model)
```



The diagnostics suggest that the Poisson model may not be the best fit for this data due to signs of over-dispersion. The normality assumption doesn't hold, which isn't surprising for Poisson-distributed data, but the heteroscedasticity and potential over-dispersion could mean that the Poisson model underestimates the true variability in the data. The lack of strong influential points is a positive, but the model's fit could likely be improved by using a model that accounts for over-dispersion, such as the Negative Binomial model, which already identified as having a better AIC value.

```
par(mfrow=c(2,2))
plot(nb_model)
```



The pattern in the Residuals Vs. Fitted plot and the Scale-Location plot could be a sign of over-dispersion, which is well-handled by the Negative Binomial model through its dispersion parameter. The point labeled 69 in all plots appears to be an outlier. It's worth investigating to determine if it represents a data entry error, a rare event, or a country with a very high medal count. While the Negative Binomial model helps to address over-dispersion and is generally a good fit for count data, the presence of patterns in the diagnostic plots suggests that the model assumptions may not perfectly hold. These patterns do not necessarily invalidate the model but suggest areas where model fit could potentially be improved. Despite these indications, the Negative Binomial model's lower AIC value compared to the Poisson model suggests it is more appropriate for this data, likely because it better accommodates the distribution and variance of the medal counts.

Parameter learning and Interpretation:

Coefficients	Estimate	Std. Error	z value	Pr>
Intercept	1.920e+00	1.145e-01	16.762	<2e-16 ***
Population	-5.258e-10	5.280e-10	-0.996	0.319
GDP	4.460e-04	5.207e-05	8.565	<2e-16 ***

Intercept 1.920 represents the log count of medals when GDP and Population are zero. Population Coefficient $-5.258e-10$ Suggests a very small decrease in medal counts with increasing population, although not statistically significant $p = 0.319$. GDP Coefficient $4.460e-04$ Indicates that as GDP increases by one unit, the log count of medals is expected to increase by approximately 0.000446, which is statistically significant $p < 2e-16$. This reflects the positive impact of economic capability on winning medals. Dispersion Parameter $Theta = 1.547$ helps in adjusting the variance separately from the mean, providing a better fit for the over-dispersed count data.

The Negative Binomial regression model, with its ability to handle over-dispersion and accommodate the count nature of the data with zero-inflation and right-skewness, provides a robust, statistically sound framework for modeling Olympic medal counts. It surpasses the Poisson and Polynomial Linear regression models in terms of fit and appropriateness for the data's distribution, justified by the lowest AIC score among the models tested. This makes it an ideal choice for predictive modeling and inferential analysis in sports analytics and similar fields where count data with over-dispersion is common.

Comparing My model to Task 1 and 2:

The Negative Binomial regression stands out as a superior model for count data like Olympic medal counts compared to the Linear and Log-Transformed Linear models, mainly because it effectively handles over-dispersion and provides predictions that are inherently aligned with the nature of the data (non-negative counts). While the Linear model assumes constant variance and the Log-Transformed Linear model improves this by stabilizing variance, neither directly accounts for the variability often found in count data. The Negative Binomial model's extra dispersion parameter offers more flexibility, allowing it to accommodate the greater variance seen as the count values rise, leading to more accurate standard errors and confidence intervals.

Task 4:

lm_model for a Linear Regression model. lm_log_model for a Log-Transformed Linear Regression model, applying a log transformation to the response variable to potentially normalize the residuals and stabilize variance. nb_model for a Negative Binomial model, which is suitable for count data and addresses over-dispersion.

```
# Fit a Linear Regression Model
lm_model <- lm(Medal2012 ~ Population + GDP, data = medal_data)
# Fit a Log-Transformed Linear Regression Model
lm_log_model <- lm(log(Medal2012 + 1) ~ Population + GDP, data = medal_data)
# Fit a Negative Binomial Regression Model
nb_model <- glm.nb(Medal2012 ~ Population + GDP, data = medal_data)
# Calculate AIC values
aic_lm <- AIC(lm_model)
aic_lm_log <- AIC(lm_log_model)
aic_nb <- AIC(nb_model)
```

The AIC for each model to compare their relative quality. Lower AIC values indicate a better model fit when considering both the likelihood of the model and the number of parameters used. Creating a data frame that consolidates all AIC values for a side-by-side comparison.

```
# Create a data frame to display AIC comparisons
aic_comparison <- data.frame(
  Model = c("Linear", "Log-Linear", "Negative Binomial"),
  AIC = c(aic_lm, aic_lm_log, aic_nb)
)
# Print the AIC values to compare models
print(aic_comparison)
```

```
##               Model      AIC
## 1            Linear 553.1870
## 2        Log-Linear 174.3901
## 3 Negative Binomial 475.9745
```

The linear model has a relatively higher AIC compared to the log-linear and negative binomial models. This suggests that, while it may provide a baseline model, it doesn't handle the intricacies of the count data as

effectively as the other models. The log-linear model dramatically improves on the linear model, showing a much lower AIC. This indicates that the transformation applied to the count of medals (logarithm) has likely helped stabilize the variance and linearise the relationship between the predictors and the outcome, resulting in a better fit. Although the negative binomial model has a higher AIC than the log-linear model, it is still preferable over the basic linear model. The negative binomial model accounts for over-dispersion inherent in count data, which the basic linear regression does not.

Comparing the plots from Task 1,2 and 3:

The **log-linear model stands out as the most fitting according to AIC values**, but AIC should not be the only factor in model selection. The practical applicability of the models and their adherence to the data distribution assumptions must also be taken into account.Specifically, the log-linear model presupposes that the logarithmic transformation of the count data leads to normally distributed errors with constant variance. If this model satisfies these conditions, it may be considered the most suitable. Conversely, should over-dispersion persist, the negative binomial model, despite a higher AIC, could be deemed more appropriate due to its capacity to manage this issue, something that AIC does not directly evaluate.

The Negative Binomial model shows an overall improvement in the diagnostic plots compared to the Linear and Log-Transformed Linear models, with a better handle on the variance and a more random pattern in the residuals. However, there are still some signs of potential heteroscedasticity and a few outliers, which are common in count data and do not necessarily invalidate the model. The residuals' behavior in the Negative Binomial model diagnostics suggests it is better equipped to handle the properties of the count data, particularly the over-dispersion.While the Log-Transformed Linear model had the lowest AIC, indicating a good fit to the data, it's crucial to note that the Negative Binomial model may provide more reliable inferences due to its nature of handling count data. The final model selection should balance the AIC information with the diagnostic plots' insights, taking into consideration the theoretical understanding of the data generation process. The decision would ideally favor the model that has a **low AIC**.

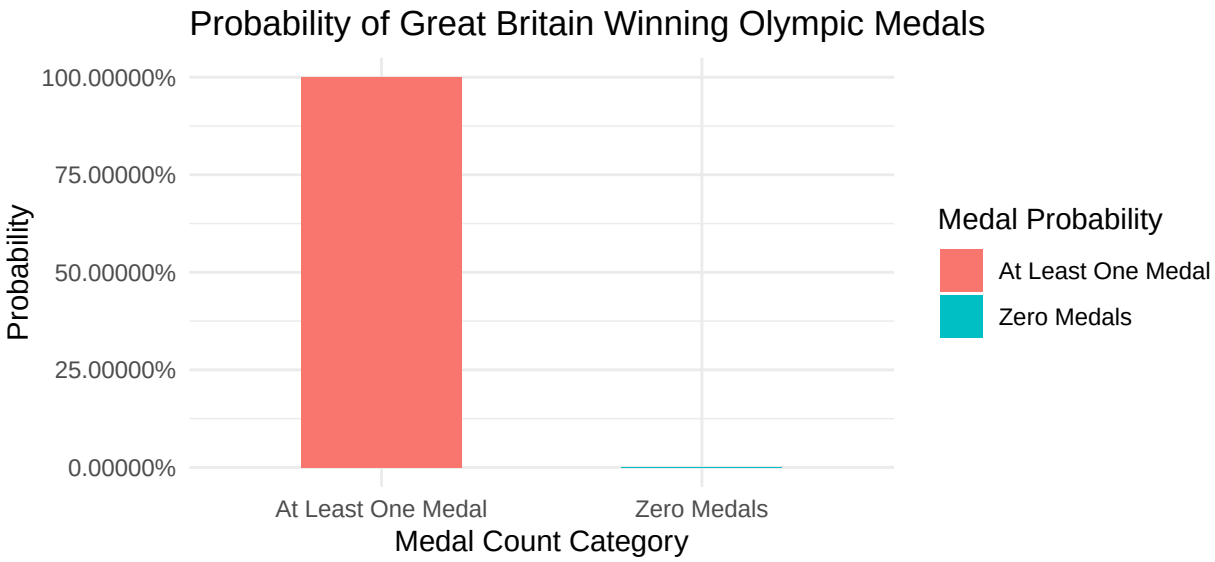
Task 5:

Probability that UK wins at least one medal using log-transformed linear regression:

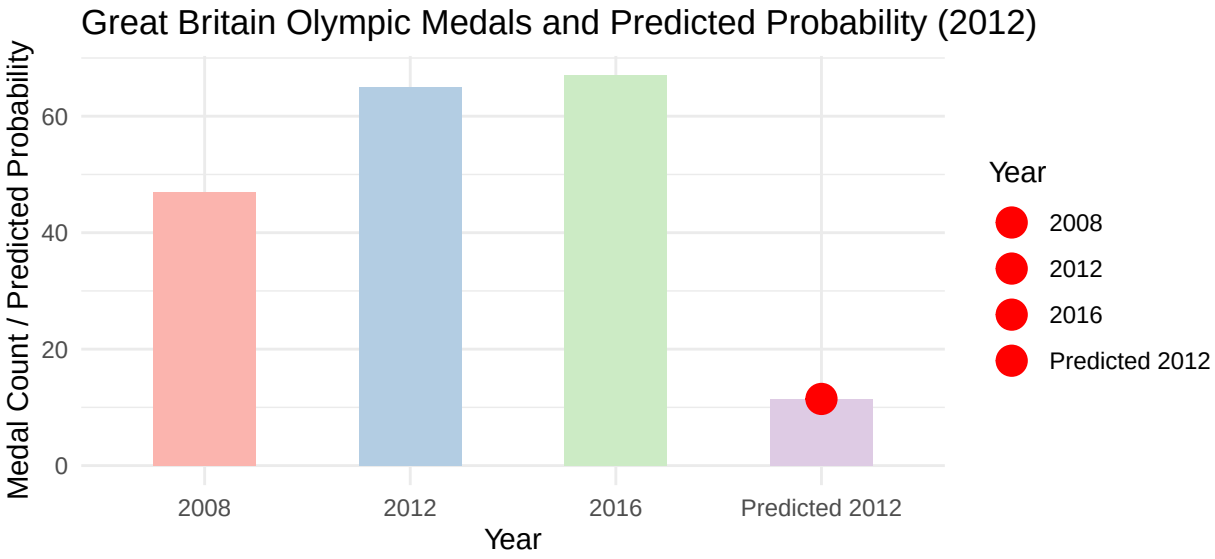
```
uk_data <- medal_data[medal_data$Country == "Great Britain", ]
log_predicted_count_uk <- predict(lm_log_model, newdata = uk_data, type = "response")
predicted_count_uk <- exp(log_predicted_count_uk) - 1
prob_zero_medals <- dpois(0, lambda = predicted_count_uk)
prob_at_least_one_medal <- 1 - prob_zero_medals
prob_at_least_one_medal
```

```
## [1] 0.9999893
```

Based on the output, the calculated probability of Great Britain winning at least one medal is approximately 99.99893%. This very high probability suggests that, according to the log-linear model's predictions and the given Population and GDP data, it is almost certain that Great Britain would win at least one medal.



The box plot below dynamically retrieves historical Olympic medal data for Great Britain and visualizes this data alongside a predicted value for the 2012 Olympic Games. This effectively presents a historical overview of Great Britain's Olympic performance in terms of medal counts and shows how the model's prediction for 2012 compares to the actual counts. The visualization can be useful for communicating both the model's capabilities and historical trends to a broad audience.



The actual medal counts for the years 2008 and 2016 show a clear upward trend, with the number of medals increasing significantly from 2008 to 2012 and maintaining a high level in 2016. The predicted count for 2012 (shown as a red dot) is visibly lower than the actual medal count for that year. This suggests that while the model predicts Great Britain to win medals, it underestimates the number compared to the actual performance. Despite the underestimation, the prediction is not on the zero mark, indicating that the model captures the likelihood of winning medals well. However, the exact count is not accurate, which could be due to factors not accounted for in the model, such as home advantage, as 2012 was the year London hosted the Olympics. It is important to note that the red dot represents a probability-based predicted count. The very high probability suggests almost certainty in winning medals, but it does not provide a precise number. The actual counts of medals are represented by bars for the respective years. The graph serves to compare the model's prediction to actual historical outcomes, highlighting areas where the model fits well and where it might need refinement. It also visually communicates the effectiveness of the model in a way that is easy

to understand for those who may not be familiar with statistical models and their outputs.

Conclusion:

In conclusion, this analytical exercise demonstrates the power and nuances of statistical modeling in sports analytics. The assignment showed that while GDP and population size are significant predictors of Olympic success, the reality of such competitive events is influenced by a myriad of factors, some quantifiable and others less so. The models developed here provide a statistical foundation that, with further refinement and incorporation of additional variables, could yield even more accurate predictions of Olympic outcomes.

References:

1. Lecture notes
2. https://www.datacamp.com/tutorial/linear-regression-R
3. https://www.scribbr.com/statistics/simple-linear-regression/
4. https://www.tutorialspoint.com/r/r_linear_regression.htm
5. https://www.atlassian.com/data/charts/what-is-a-scatter-plot
6. https://rpruim.github.io/s341/S19/from-class/MathinRmd.html#:~:text=Math%20inside%20RMarkdown,10n%3D1n2
7. https://www.scribbr.com/statistics/linear-regression-in-r/
8. https://stats.stackexchange.com/questions/193959/does-cross-validation-on-simple-or-multiple-linear-regression-make-sense
9. https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf
10. https://corporatefinanceinstitute.com/resources/data-science/heteroskedasticity/#:~:text=Heteroskedasticity%20vs.&
11. https://pro.arcgis.com/en/pro-app/3.1/tool-reference/spatial-statistics/generalized-linear-regression.htm#:~:text=Illustration-,Summary,and%20count%20(Poisson)%20models.