MATH5743M: Statistical Learning:Assignment1

Predicting the Olympic Games

Surapot Nonpassopon

2024-04-12

Using Regression Models to Predicting the Olympic Games.

Data understanding and preprocessing.

1. First, the data set file "medal_pop_gdp_data_statlearn.csv" must be inputted into the R studio. Second, the head function can be used to observe the example's first five data lines.

```
# Import the dataset first
setwd("~/Desktop/Semaster2/Stat Learning/Assignment/A1")
medal <- read.csv("medal_pop_gdp_data_statlearn.csv", stringsAsFactors = FALSE, encoding = 'UTF-8')

# Data observation
head(medal)
```

```
##      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 explanation of each variable is provided in Table 1 below.

Table 1: The explanation of dataset variables.

Variable	Explanation
Country	The country name (recognized by the IOC)
GDP	The GDP rate (in billions of US dollars)
Population	The number of populations in each country
Medal2008	The number of medal wins in 2008 Olympic games in Beijing
Medal2012	The number of medal wins in 2012 Olympic games in London
Medal2016	The number of medal wins in 2016 Olympic games in Rio

2. The missing value must then be removed before processing.

```
# Removing the missing value
medal <- na.omit(medal)
```

3. After that, using the R function of str () to understand the function of the variable, for example, Country is "chr", which means the type of Character. Moreover, the R function of the summary is used to inform the mean, median, minimum value, maximum value and quantile range of numeric columns of the dataset.

```r
# Data understanding
str(medal)
```

```
## 'data.frame':    71 obs. of  6 variables:
##  $ Country   : chr  "Algeria" "Argentina" "Armenia" "Australia" ...
##  $ GDP       : num  188.7 446 10.2 1371.8 63.4 ...
##  $ Population: int  37100000 40117096 3268500 22880619 9111100 353658 1234571 9461400 10951266 19237
##  $ Medal2008 : int  2 6 6 46 7 2 1 19 2 15 ...
##  $ Medal2012 : int  1 4 3 35 10 1 1 12 3 17 ...
##  $ Medal2016 : int  2 4 4 29 18 2 2 9 6 19 ...
```

```r
summary(medal)
```

```
##    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
```

4. Following that, it could use the visualization as a histogram in Figure 1 to represent the shape of our data, which seems to be the right skew. This means there are a few very high values that pull the tail of this distribution to the right.

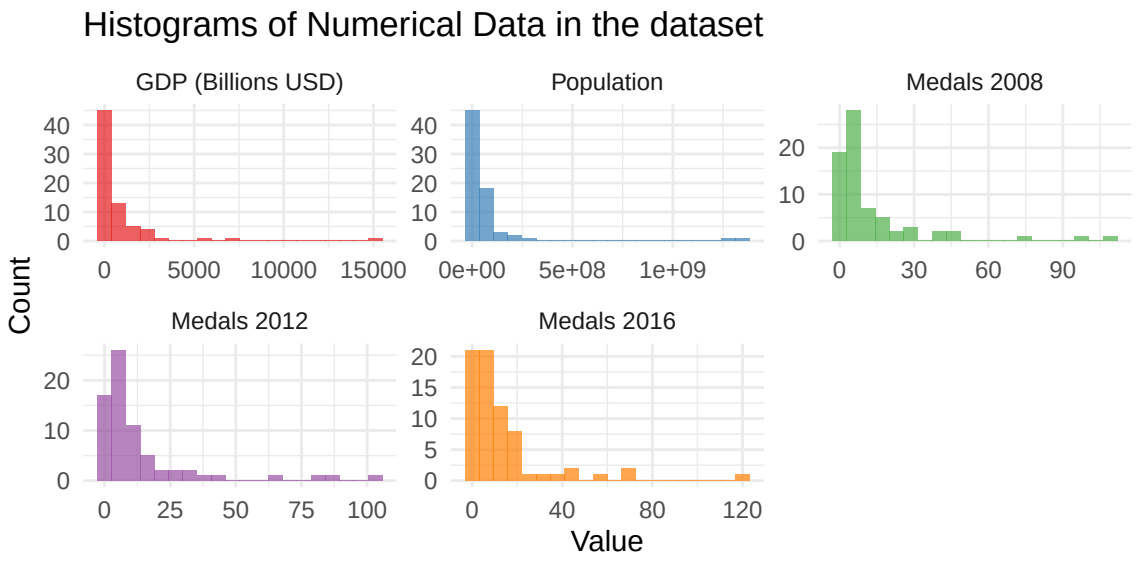

Figure 1: The Histogram of Numerical Data

Assigment Task

Task1: Use Population and GDP as inputs and medal count in the 2012 Olympics as outputs for the linear regression model.

To progress linear regression in R, the function of glm() needs to be used while setting the input variables Population and GDP, which are independent variables used to make the predictions. Additionally, setting the medal count in 2012 as the output is a dependent variable that needs to be predicted.

```
linear_medal <- glm(Medal2012 ~ Population + GDP, data = medal)
summary(linear_medal)
```

```
##
## Call:
## glm(formula = Medal2012 ~ Population + GDP, data = medal)
##
## 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
```

From the summary of the linear regression model, in terms of Population, the linear regression model represents that the model estimate is very small, with the value 5.247×10^{-9}, which could have a non-significant effect on the number of medals won in the 2012 Olympics. In contrast, GDP might have a significant effect on the number of medals won in the London Olympics due to the estimated rate of around 0.007564 with a small number of standard deviation errors. It could be predicted that the high GDP is associated with the greater number of medals won in the Olympics 2012.

Focusing on the AIC values reveals this model's performance. This model has an AIC of 553.19, which will be used to compare to other models for the other tasks.

After that, the confidence interval 95% of each variable has been computed through these steps.

```
# Finding the confident interval of population
# Using 71 - 3 = 68 as degree of freedoms.
# The quantile that need is 0.975
linear_medal_table <- summary(linear_medal)$coefficients
t_critical = qt(0.975, 68)
estimate_pop_1 = linear_medal_table[2,1]
sterr_pop_1 = linear_medal_table[2,2]

interval_pop_min_1 = estimate_pop_1 - t_critical * sterr_pop_1
interval_pop_max_1 = estimate_pop_1 + t_critical * sterr_pop_1

print(paste(c('Population Min: ', interval_pop_min_1), collapse=""))
```

```
## [1] "Population Min: -9.10593444565585e-09"
```

```r
print(paste(c('Population Max: ', interval_pop_max_1), collapse=""))
```

```
## [1] "Population Max: 1.95994344112297e-08"
```

```r
# Finding the confident interval of GDP
# Using 71 - 3 = 68 as degree of freedoms.
# The quantile that need is 0.975
t_critical = qt(0.975, 68)
estimate_GDP_1 = linear_medal_table[3,1]
sterr_GDP_1 = linear_medal_table[3,2]

interval_GDP_min_1 = estimate_GDP_1 - t_critical * sterr_GDP_1
interval_GDP_max_1 = estimate_GDP_1 + t_critical * sterr_GDP_1

print(paste(c('GDP Min: ', interval_GDP_min_1), collapse=""))
```

```
## [1] "GDP Min: 0.00610231906043589"
```

```r
print(paste(c('GDP Max: ', interval_GDP_max_1), collapse=""))
```

```
## [1] "GDP Max: 0.00902584306021206"
```

To make it better to see the result, the confidence interval of population and GDP in this linear regression model are represented in Table 2.

Table 2: The 95% confidence interval of Population and GDP for the linear regression model.

Confident.interval	Min	Max
Population	-9.11×10^{-9}	1.96×10^{-8}
GDP	0.006	0.009

From Table 2, the confidence interval of the population that includes 0 indicates that the size of the population may not have a statistically significant impact on the number of medals won in the 2012 Olympics. By contrast, the GDP show a positive value of 95% confidence interval, which could be represented as if the GDP increased, the number of medals won in the Olympics 2012 would increase as well.

Task 2: Repeat task 1 for log-transformed outputs. Discuss potential benefits and reasons for using the transformation.

Potential benefits and reasons for using log-transformation.

- **Normalisation of skewed data:** It can be seen from Figure 1 that the data's numerical value distribution is right-skewed, which could affect the performance of linear regression models. The taking-logarithms transformation could normalize the distribution and help improve model performance.
- **Improving the model fit:** After transforming the right-skewed data by taking a log, the shape of the data might be approximate like the normal distribution, which could lead to a better fit to the regression model that could be performing better.
- **Reducing the impact of outliners of data:** if the data contains the outliners, it might have an effect when the data is used to fit the model, which might lead to wrong results. Thus, when using log transformation, it will compress the scale of the data, which can reduce the impact of the outliner that could affect the regression result.

Performing the linear regression with a log-transformed output model. To perform the log-transformed outputs, it will use the lm () function, the same as task1, setting the input as population and GDP and setting the output by taking the log to the value of the number of medals in 2012.

```
# Task 2: Processing linear regression with log-transformed output
log_linear_medal <- glm(log(Medal2012) ~ Population + GDP, data = medal)
summary(log_linear_medal)
```

```
##
## Call:
## glm(formula = log(Medal2012) ~ Population + GDP, data = medal)
##
## 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
```

From the summary of the log-transformed linear regression model, it could be seen that population estimates contain a small value, which means the number of populations is not statistically significant in predicting the medal won in 2012. However, The GDP might have a significant effect on the medal won based on the estimated value around 3.16×10^{-4} with a small standard deviation error.

Focusing on the AIC values reveals this model's performance. The AIC value is better than task 1. It might be recommended that this model has a better balance between model fit and complexity than a normal linear regression model.

After that, the confidence interval of each variable has been computed through these steps.

```
# Finding the confident interval of population
# Using 71 - 3 = 68 as degree of freedoms.
# The quantile that need is 0.975
log_linear_medal_table <- summary(log_linear_medal)$coefficients
t_critical = qt(0.975, 68)
estimate_pop_2 = log_linear_medal_table[2,1]
sterr_pop_2 = log_linear_medal_table[2,2]

interval_pop_min_2 = estimate_pop_2 - t_critical * sterr_pop_2
interval_pop_max_2 = estimate_pop_2 + t_critical * sterr_pop_2

print(paste(c('Population Min: ', interval_pop_min_2), collapse=""))
```

```
## [1] "Population Min: -1.0984460237347e-09"
```

```
print(paste(c('Population Max: ', interval_pop_max_2), collapse=""))
```

```
## [1] "Population Max: 1.31945505913133e-09"
```

```
# Finding the confident interval of GDP
# Using 71 - 3 = 68 as degree of freedoms.
# The quantile that need is 0.975
t_critical = qt(0.975, 68)
estimate_GDP_2 = log_linear_medal_table[3,1]
sterr_GDP_2 = log_linear_medal_table[3,2]

interval_GDP_min_2 = estimate_GDP_2 - t_critical * sterr_GDP_2
interval_GDP_max_2 = estimate_GDP_2 + t_critical * sterr_GDP_2

print(paste(c('GDP Min: ', interval_GDP_min_2), collapse=""))
```

```
## [1] "GDP Min: 0.000192975139143463"
```

```
print(paste(c('GDP Max: ', interval_GDP_max_2), collapse=""))
```

```
## [1] "GDP Max: 0.000439228440401706"
```

To make it more clear, the confidence interval of population and GDP in this log-linear regression model are represented in Table 3.

Table 3: The 95% confidence interval of Population and GDP for the log-transformed linear regression model.

Confident.interval	Min	Max
Population	-1.098×10^{-9}	1.319×10^{-9}
GDP	0.0002	0.0004

Table 3 illustrated that the 95% confidence interval of Population and GDP, including the 0, could not be confident that the increase or decrease of the population has an effect on the number of Olympic 2012 medals. In terms of GDP, it could be seen that the positive value of the confidence interval suggests that when the GDP increases, the number of medals won in 2012 could increase as well.

Task 3: List relevant properties of the outputs. Develop and explain your own regression model that uses the same inputs and outputs from task 1 but is different from the models in tasks 1 and 2. Justify your choice and discuss how your model takes into account the listed properties. Discuss potential benefits of your approach compared to models 1 and 2.

First, draw the box plot to find the list of relevant properties of the outputs.

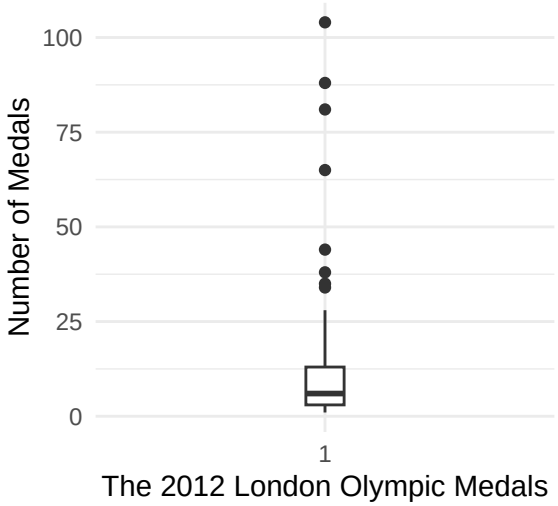

Figure 2: Box Plot of 2012 Olympic Medals

List of relevant properties of the outputs that can be observed from the box plot.

- **Median:** The box plot's median is above 0, which means that at least half of the countries in this dataset won the medal in 2012.
- **Spread:** The interquartile range (IQR) seems very small compared to the entire range of the dataset.
- **Range:** The overall range from minimum to maximum, excluding the outliers, seems wide, showing the variability in the number of medals won.
- **Skewness:** The box plot clearly represents the right skew, as evidenced by the long upper whisker, and the median is close to the bottom of the box.
- **Outliers:** There are several outliers indicated by the points above the upper whisker.

Second, draw the scatter plots to find the list of relevant properties of the outputs.

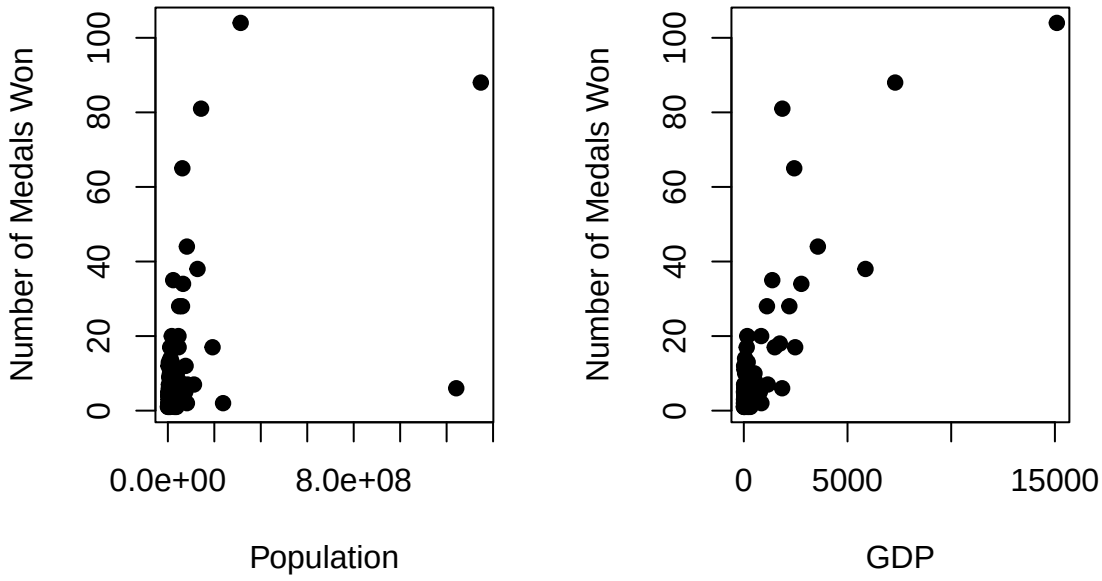

Figure 3: Scatter plots showing the relationship between 2012 Olympic Medals and Population and GDP

List of relevant properties of the outputs that can be observed from the scatter plot.

- **Relationships:** Both scatter plots suggest that the relation between output and input variables is not clearly shown as linear. Thus, the non-linear model might be used to fit this data.

In conclusion, given the right skew of the data and the outliers, it might be shown that our data might not be suitable for the normal distribution, which has already been solved by task 2. Therefore, we need to find another model to tackle the problem of the non-linear relationship of the data that might be used in the polynomial regression model.

Develop and explain the regression model. The model will choose to handle the non-linear relationship between the dependent variable and the independent variable in a quadratic model. The quadratic model was chosen instead of the cubic model because if the model is cubic model, it will produce a close fit to the data that might lead to the problem of overfitting.

The equation of this model could be written as equation 1.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \varepsilon \tag{1}$$

- y is the dependent variable.
- x_1 and x_2 are the independent variables.
- x_1^2 and x_2^2 are the squared terms of x_1 and x_2, providing the quadratic effects of each variable.
- β_0 is the intercept.
- β_1, β_2, β_3, and β_4 are the coefficients for linear and quadratic terms.

- ε is the error term.

By inputting the variable of our data in equation 1 to receive equation 2 that could be used in R.

- The dependent variable is the number of medals won in the 2012 Olympics.
- The independent variables are the number of population and GDP.

$$\text{Medal2012} = \beta_0 + \beta_1(\text{Population}) + \beta_2(\text{Population}^2) + \beta_3(\text{GDP}) + \beta_4(\text{GDP}^2) + \varepsilon \tag{2}$$

Equation 2 would be used in R to fit the data to the quadratic model. First, a data frame of each variable would need to be created. After that, each variable from the data frame is inputted into the glm() function.

```
# create the data frame to progess the quadratic  model.
mydata = data.frame(Medal2012 = medal$Medal2012,
                    Population = medal$Population, Population2 = medal$Population^2, GDP = medal$GDP, GI

# Fit the quadratic model (including squared terms)
quadratic_model = glm(Medal2012 ~ Population + Population2 + GDP + GDP2, data = mydata)
summary(quadratic_model)
```

```
##
## Call:
## glm(formula = Medal2012 ~ Population + Population2 + GDP + GDP2,
##     data = mydata)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.612e+00  1.721e+00   2.679  0.00930 **
## Population  -3.603e-08  3.682e-08  -0.979  0.33131
## Population2  2.355e-17  2.652e-17   0.888  0.37779
## GDP          1.379e-02  1.957e-03   7.045 1.35e-09 ***
## GDP2        -4.408e-07  1.306e-07  -3.374  0.00124 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 115.6967)
##
##     Null deviance: 28403  on 70  degrees of freedom
## Residual deviance:  7636  on 66  degrees of freedom
## AIC: 545.62
##
## Number of Fisher Scoring iterations: 2
```

Overall, as can be seen from the model summary, the coefficients show that GDP terms, with both linear and quadratic terms, could have statistical significance in predicting the Olympic medals won in 2012. The linear GDP term shows a positive relationship. The quadratic term is negative, implying the point of diminishing returns when increasing the GDP doesn't help a country win any more medals. In terms of population, it does not appear to have a significant effect on the medal won with both linear and quadratic terms.

The AIC of this model is 545.62, which could be compared to the other models in Task 1 and 2 to see which models are better for fitting the data.

Summarise potential benefits and reasons for using the quadratic equation to handle the list of properties.

- **Non-linear Relationship:** It could be used to capture the non-linear pattern of data, which a simple linear regression model might miss. As can be seen from the scatter plot, the relationship between input(population and GDP) and the number of medals won does not seem to be a straight line.
- **Better fit of the model:** Including the quadratic terms might provide a better fit of the dataset that could capture more variance, which leads to the improvement of the accuracy of the model.
- **Diminishing or Increasing Returns:** The quadratic model is useful to show the patterns where the data starts to slow down or speed up. For example, from our model in GDP, the quadratic term is negative, implying the point of diminishing returns when increasing the GDP doesn't help a country win any more medals.
- **Avoiding Overfitting:** The quadratic model is good for capturing the complexity of the data and maintaining its simplicity. Compared to the cubic model, the cubic model might be better for capturing the data, but it could cause overfitting.

Task 4: Carry out model selection using AIC to determine which model from tasks 1 to 3 performs best. In addition to using AIC, analyse and inspect the models to determine which model would you choose to accurately predict the medal count. Report your results. Justify your reasoning and choice.

The Akaike Information Criterion (AIC) was used to measure the model selection in statistical models in this task. It measures the quality of statistical models by balancing model fit and model complexity, penalising the models with more parameters.

Table 4: Comparison of AIC Values Across Different Regression Models

Model	AIC
Task1: Linear regression model	553.19
Task2: Linear regression model with log-transformation output.	201.85
Task3: Quadratic model	545.62

Choosing the model based on AIC. When choosing a model based on the AIC, the model with the lowest AIC is preferred. The lower AIC score suggests a better balance between model fit and complexity. Based on the AIC values presented in the table, the linear regression model with log-transformation output (Task2) has the lowest AIC, around 200, while the two other models have approximately an AIC of 550, suggesting the most appropriate model and might be the most accurate model among the three predicting models. Thus, the final choice of a model based on AIC is the linear regression model with log-transformation output (Task2).

Justify reason and choice. (linear regression model with log-transformation output)

- **Lowest AIC:** The linear regression model with log-transformation output has the lowest AIC (201.85), indicating a better balance between model fit and complexity. Moreover, lower AIC suggests this model might perform better in the new unseen data.
- **Right-Skewed Data:** using the log-transformation could address the right skew of medals won in 2012, which might improve the model fit.
- **Outliner Impact:** As mentioned in task 2, the log transformation could handle the effect of outliers. Taking the log might help compress the scale of data and reduce the impact of outliers.

Task 5: Using the model from task 4, derive and compute the probability that the UK wins at least one medal given the estimated model parameters.

The model selected in task 4 is the Linear regression model with log-transformation output, which will be used to derive and compute the probability that Great Britain wins at least one medal through these 5 steps.

1. Obtain the model coefficients.

The 'coef' function was used to receive the coefficient from the linear regression model with log transformation. These coefficients represent the relationship between the input variable (population and GDP) and the logarithm of the output variable (number of medals won in 2012).

```
# 1st: obtain the estimated coefficients from model
coefficients <- coef(log_linear_medal)
```

2. Find the population and GDP of Great Britain.

```
# 2nd: Get the values of 'Population' and 'GDP' from Great Britain
GB_population <- medal$Population[medal$Country == 'Great Britain']
GB_gdp <- medal$GDP[medal$Country == 'Great Britain']
```

3. Predict the log of the number of medals won in 2012.

This could be computed by each coefficient being multiplied by its corresponding variable and summed up together, including the intercept.

```
# 3nd: Calculate the predict value
log_predicted_medals2012 <- coefficients['(Intercept)'] +
  coefficients['Population'] * GB_population +
  coefficients['GDP'] * GB_gdp

print(log_predicted_medals2012)
```

```
## (Intercept)
##     2.34493
```

4. Converting the result from the log scale to the original scale.

The exponential function('exp') was used to append the predicted result to convert from the log scale to the original scale, which is the number of medals.

```
# 4th: Convert log back to the original scale by taking exponential
predicted_medals2012 <- exp(log_predicted_medals2012)
print(predicted_medals2012)
```

```
## (Intercept)
##    10.43255
```

5. Calculate the probability of winning at least one medal.

The final step is using the properties of the Poisson distribution to calculate the probability of winning at least one medal. The probability of winning zero medals in a Poisson distribution is $P(X = 0) = e^{-\lambda}$, in which the λ is the expected won count. Subtract $P(X = 0)$ from 1 to get the probability of winning one or more medals as the equation 3.

$$P(X \geq 1) = 1 - P(X = 0) \tag{3}$$

```
# 5th: The probability of winning at least one medal is 1 minus the probability of winning zero medals.
# For the possion distribution: P(X = 0) = exp(-lambda)
# where lambda is the expected won (predicted_medals2012)
```

```r
prob_at_least_one_medal <- 1- exp(-predicted_medals2012)
print(paste(c('The probability of winning at least one medal of UK ', prob_at_least_one_medal ), collaps
```

```
## [1] "The probability of winning at least one medal of UK 0.999970542015371"
```

The result is represented at approximately 99.997% based on the linear regression model with log-transformation output, which is a high probability that suggests that Great Britain has a very high probability of winning one or more medals in the Olympic Games 2012.