Assessed Practical 1

Praveen Gopal Reddy

08/07/2021

Table of Contents

Aim: 1

Regression Tasks: 2

Task 1 2

Task 2 5

Task 3 7

Task 4 8

Task 5 9

Model Selection Tasks 10

Task 1 10

Task 2 12

Task 3 14

Aim:

To investigate how the number of medals a country wins can be predicted from national population and GDP, and how consistent these relationships are

```
# After importing csv file of medal_pop_gdp_data_statlearn.csv and
2021_pop_gdp_staltearn.csv, the dataframes that we use are:
medal_pop_gdp_stlearn and pop_gdp_stlearn_2021
summary(medal_pop_gdp_stlearn)

##    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
```

```
summary(pop_gdp_stlearn_2021)
```

```
##    Country                GDP            Population      
##  Length:71          Min.   :  648.5   Min.   :    396.9  
##  Class :character   1st Qu.: 5184.0   1st Qu.:   5855.0  
##  Mode  :character   Median :12264.5   Median :  18995.0  
##                     Mean   :22968.7   Mean   :  81131.2  
##                     3rd Qu.:34585.2   3rd Qu.:  57513.8  
##                     Max.   :84986.8   Max.   :1444216.1
```

Regression Tasks:

Task 1

Perform a linear regression to predict the medal count in 2008 and 2016 (separately, in two regressions) from Population and GDP. Explain your model and approach to learn the model parameters. Report your results and comment on your findings.

```
#Building linear regression model for 2008 and 2016
model2008 <- glm(Medal2008 ~ GDP+Population, data = medal_pop_gdp_stlearn)
summary(model2008)
```

```
## 
## Call:
## glm(formula = Medal2008 ~ GDP + Population, data = medal_pop_gdp_stlearn)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -27.154   -4.856   -1.702    0.842   51.037  
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.613e+00  1.506e+00   3.728 0.000395 ***
## GDP         7.613e-03  7.353e-04  10.354 1.29e-15 ***
## Population  8.435e-09  7.220e-09   1.168 0.246750    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 133.1455)
## 
##     Null deviance: 29595.1  on 70  degrees of freedom
## Residual deviance:  9053.9  on 68  degrees of freedom
## AIC: 553.72
## 
## Number of Fisher Scoring iterations: 2
```

```
model2016 <- glm(Medal2016 ~ GDP+Population, data = medal_pop_gdp_stlearn)
summary(model2016)
```

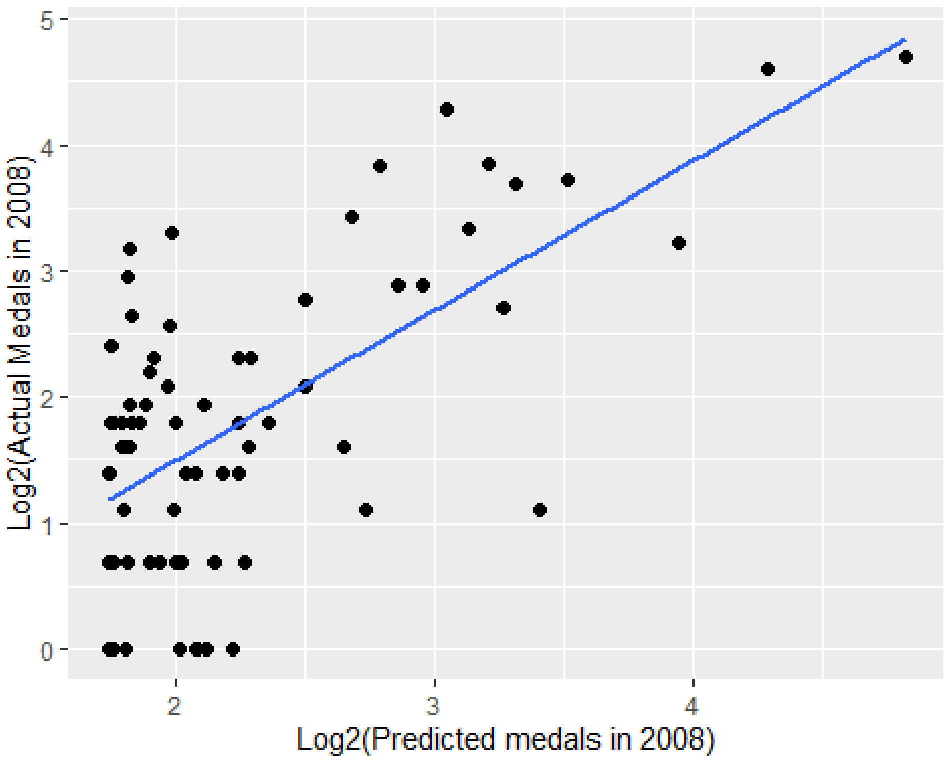
```
##
## Call:
## glm(formula = Medal2016 ~ GDP + Population, data = medal_pop_gdp_stlearn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -14.239   -5.468   -2.679    3.564   40.492
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.287e+00  1.170e+00   5.374 1.01e-06 ***
## GDP          8.499e-03  5.713e-04  14.876  < 2e-16 ***
## Population  -7.135e-09  5.610e-09  -1.272    0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 80.3846)
##
##     Null deviance: 26661.5  on 70  degrees of freedom
## Residual deviance:  5466.2  on 68  degrees of freedom
## AIC: 517.89
##
## Number of Fisher Scoring iterations: 2
```

```
#New dataframe to put predictions of 2008 and 2016:
predict2008_and_2016<- data.frame(Country = medal_pop_gdp_stlearn$Country,
Population = medal_pop_gdp_stlearn$Population, GDP =
medal_pop_gdp_stlearn$GDP)

#Add 2008 actual and predicted new columns to the dataframe
predict2008_and_2016['Actual_2008'] <- medal_pop_gdp_stlearn$Medal2008
predict2008_and_2016['Predicted_2008'] <- predict(model2008)

#Add 2016 actual and predicted new columns to the dataframe
predict2008_and_2016['Actual_2016'] <- medal_pop_gdp_stlearn$Medal2016
predict2008_and_2016['Predicted_2016'] <- predict(model2016)

#display results
summary(predict2008_and_2016)
```

```
##    Country            Population               GDP             Actual_2008
##  Length:71          Min.   :3.537e+05   Min.   :    6.52   Min.   :  1.00
##  Class :character   1st Qu.:5.513e+06   1st Qu.:   51.52   1st Qu.:  2.00
##  Mode  :character   Median :1.673e+07   Median :  229.53   Median :  6.00
##                     Mean   :7.384e+07   Mean   :  903.25   Mean   : 13.11
##                     3rd Qu.:4.958e+07   3rd Qu.:  704.37   3rd Qu.: 13.50
##                     Max.   :1.347e+09   Max.   :15094.00   Max.   :110.00
##  Predicted_2008     Actual_2016     Predicted_2016
##  Min.   :  5.676   Min.   :  1.00   Min.   :  5.955
##  1st Qu.:  6.126   1st Qu.:  3.00   1st Qu.:  6.658
##  Median :  7.482   Median :  7.00   Median :  8.101
##  Mean   : 13.113   Mean   : 13.44   Mean   : 13.437
##  3rd Qu.: 11.320   3rd Qu.: 15.00   3rd Qu.: 11.711
##  Max.   :123.169   Max.   :121.00   Max.   :132.329
```

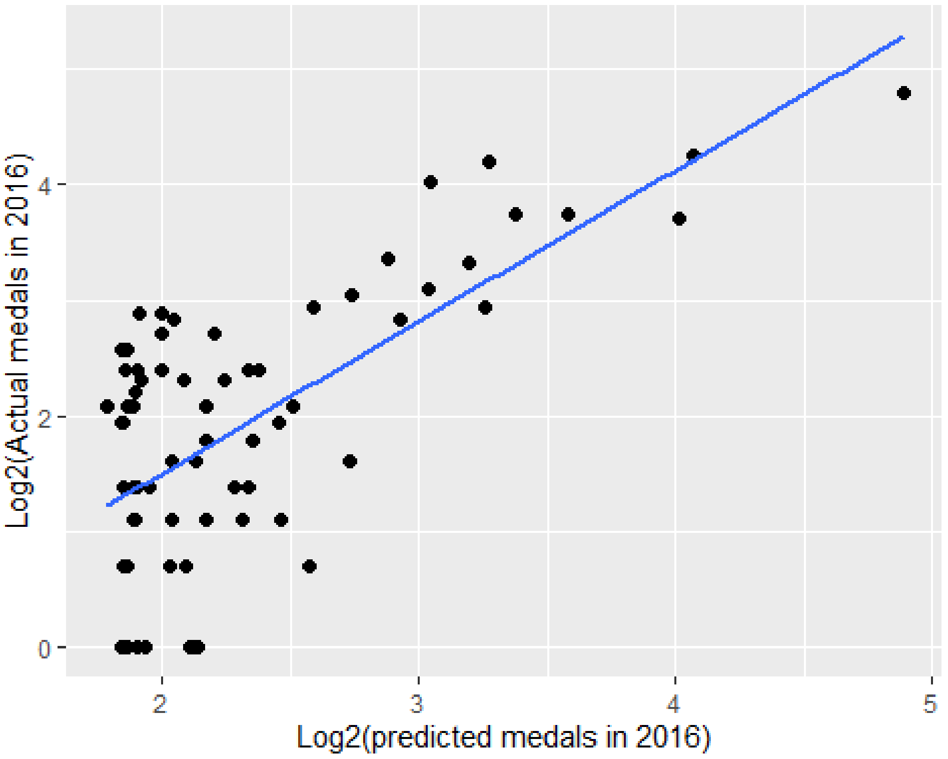
```
#plot the relation between actual and predicted values.
library(ggplot2)
plot_2008 <- ggplot(predict2008_and_2016, aes(x=log(Predicted_2008),
y=log(Actual_2008))) +
  geom_point(size=2, shape=16) + xlab("Log2(Predicted medals in 2008)") +
  ylab("Log2(Actual Medals in 2008)") + geom_smooth(method = lm,se=FALSE)

print(plot_2008)
```

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



```
plot_2016<- ggplot(predict2008_and_2016, aes(x=log(Predicted_2016),
y=log(Actual_2016))) +
  geom_point(size=2, shape=16) + xlab("Log2(predicted medals in 2016)") +
  ylab("Log2(Actual medals in 2016)") + geom_smooth(method = lm,se=FALSE)

print(plot_2016)
```

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



By looking at summary of model2008 and model2016, the estimates of Coefficients of GDP in both models is very significant. Because the P-value is closer to zero at 95% confidence interval. So GDP variable has strong relation on output variable.

On other hand, the coefficient of population variable in both models is not significant since P-value(0.2) is greater than 0.05 at 95% confidence interval.

From plot diagrams we can see that most of the data points are above and below the regression line. which means residual are positive and negative. when residual is positive, the line underestimates the actual data value of y. when residual is negative, the line overestimates the actual data value of y.

Task 2

How consistent are the effects of Population and GDP over time?

```
#coefficients of 2008 model
summary_2008_coeff=summary(model2008)$coefficients
#coefficients of 2016 model
summary_2016_coeff=summary(model2016)$coefficients

#lets construct confidence interval for coefficients of estimates of GDP and
Population
t_critical=qt(0.975,68)
#2008 GDP
estimate_gdp_2008=summary_2008_coeff[2,1]
sterr_gdp_2008=summary_2008_coeff[2,2]
```

```
interval_min_gdp_2008=estimate_gdp_2008-t_critical*sterr_gdp_2008
interval_max_gdp_2008=estimate_gdp_2008+t_critical*sterr_gdp_2008
print(paste(c("estimate_gdp_2008:",estimate_gdp_2008),collapse=""))

## [1] "estimate_gdp_2008:0.00761313496705"

print(paste(c("min_gdp_2008:",interval_min_gdp_2008),collapse=""))

## [1] "min_gdp_2008:0.00614591191989105"

print(paste(c("max_gdp_2008:",interval_max_gdp_2008),collapse=""))

## [1] "max_gdp_2008:0.00908035801420896"

#2016 GDP
estimate_gdp_2016=summary_2016_coeff[2,1]
sterr_gdp_2016=summary_2016_coeff[2,2]
interval_min_gdp_2016=estimate_gdp_2016-t_critical*sterr_gdp_2016
interval_max_gdp_2016=estimate_gdp_2016+t_critical*sterr_gdp_2016
print(paste(c("estimate_gdp_2016:",estimate_gdp_2016),collapse=""))

## [1] "estimate_gdp_2016:0.00849859830471469"

print(paste(c("min_gdp_2016:",interval_min_gdp_2016),collapse=""))

## [1] "min_gdp_2016:0.00735856045033984"

print(paste(c("max_gdp_2016:",interval_max_gdp_2016),collapse=""))

## [1] "max_gdp_2016:0.00963863615908954"

#2008 population
estimate_pop_2008=summary_2008_coeff[3,1]
sterr_pop_2008=summary_2008_coeff[3,2]
interval_min_pop_2008=estimate_pop_2008-t_critical*sterr_pop_2008
interval_max_pop_2008=estimate_pop_2008+t_critical*sterr_pop_2008
print(paste(c("estimate_pop_2008:",estimate_pop_2008),collapse=""))

## [1] "estimate_pop_2008:8.43482436477738e-09"

print(paste(c("min_pop_2008:",interval_min_pop_2008),collapse=""))

## [1] "min_pop_2008:-5.97148075440265e-09"

print(paste(c("max_pop_2008:",interval_max_pop_2008),collapse=""))

## [1] "max_pop_2008:2.28411294839574e-08"

#2016 population
estimate_pop_2016=summary_2016_coeff[3,1]
sterr_pop_2016=summary_2016_coeff[3,2]
interval_min_pop_2016=estimate_pop_2016-t_critical*sterr_pop_2016
interval_max_pop_2016=estimate_pop_2016+t_critical*sterr_pop_2016
print(paste(c("estimate_pop_2016:",estimate_pop_2016),collapse=""))
```

```
## [1] "estimate_pop_2016:-7.13513543637737e-09"

print(paste(c("min_pop_2016:",interval_min_pop_2016),collapse=""))

## [1] "min_pop_2016:-1.83288889759918e-08"

print(paste(c("max_pop_2016:",interval_max_pop_2016),collapse=""))

## [1] "max_pop_2016:4.05861810323709e-09"
```

From above results,it is obvious that estimate of GDP is consistent in 2008 and 2016 because the estimate value is within the confidence interval. It is also closer to zero indicating significance level in both years.For the estimate of population in 2008 and 2016 it is different but insignificant in terms of p-value in both years.

Task 3

Using the regression for the 2008 medal count make a prediction for the results of 2012.

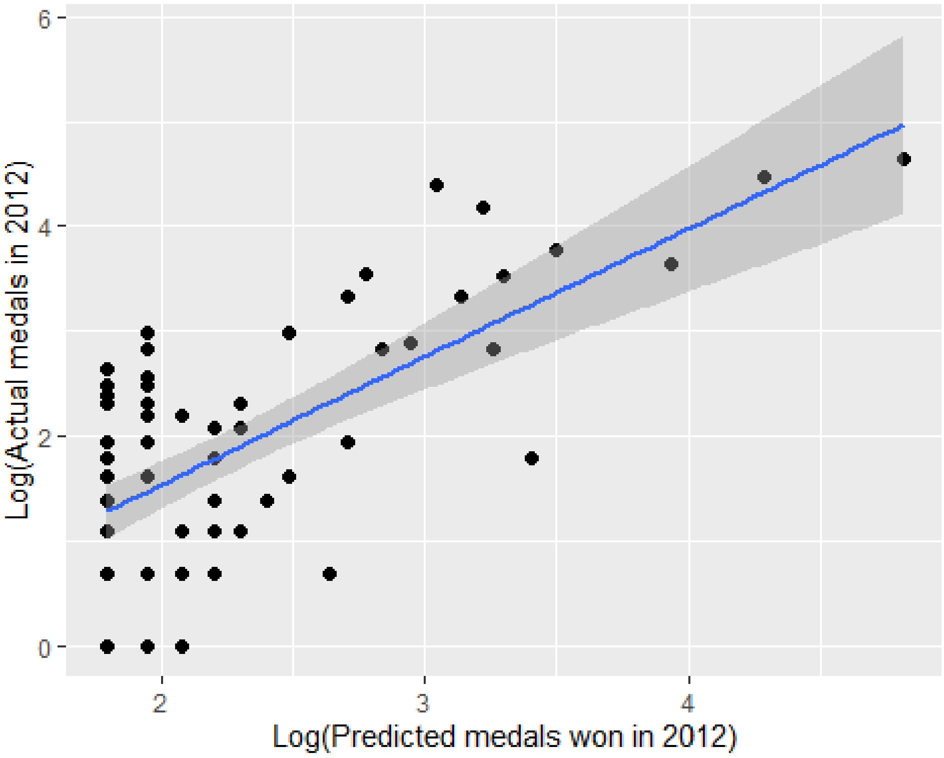
```
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

newdata = data.frame(Country = medal_pop_gdp_stlearn$Country, Population =
medal_pop_gdp_stlearn$Population, GDP = medal_pop_gdp_stlearn$GDP)

predictions_2012 = predict(model2008, newdata)

#Added predictions and actual to dataframe
newdata['Predicted_2012'] <- predictions_2012
newdata ['Actual_2012' ]<- medal_pop_gdp_stlearn$Medal2012
#Round to whole numbers
newdata <- newdata %>% mutate(across(starts_with("Predicted_2012"), round,
0))
#print predicted and actual summary
summary(newdata[4:5])

##  Predicted_2012    Actual_2012
##  Min.   :  6.00   Min.   :  1.0
##  1st Qu.:  6.00   1st Qu.:  3.0
##  Median :  7.00   Median :  6.0
##  Mean   : 13.11   Mean   : 13.3
##  3rd Qu.: 11.50   3rd Qu.: 13.0
##  Max.   :123.00   Max.   :104.0
```

From summary data mean and median for predicted and actual is almost same.

Task 4

Plot your predictions against the actual results of 2012. If the results are hard to see, use a transformation of the axes to improve clarity. Comment on your findings. How good are the predictions? Which countries are outliers from the trend?

```
library(ggplot2)
plot_task4 <- ggplot(newdata, aes(x=log(Predicted_2012), y=log(Actual_2012)))
+
  geom_point(size=2, shape=16) + xlab("Log(Predicted medals won in 2012)") +
  ylab("Log(Actual medals in 2012)") + geom_smooth(method = lm)

plot_task4

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



```
# Get fit, and make a variable with labels for points outside CIs
fit <- lm(newdata$Actual_2012~newdata$Predicted_2012)

#creates a dataframe with the fit, lower and upper CI's
dat_upr_lwr <- predict.lm(fit, interval="confidence")
```

```
#The below condition creates new column "outside" with a country name if it
is outside the CI's
newdata$outside <- ifelse(newdata$Actual_2012 < dat_upr_lwr[,"upr"] &
newdata$Actual_2012 > dat_upr_lwr[,"lwr"], "", as.character(newdata$Country))
#display countries which are outliers
newdata$outside
```

```
##  [1] "Algeria"            "Argentina"          "Armenia"
##  [4] "Australia"          "Azerbaijan"         "Bahamas"
##  [7] "Bahrain"            "Belarus"            "Belgium"
## [10] "Brazil"             "Bulgaria"           ""
## [13] "China"              ""                   ""
## [16] "Cuba"               ""                   ""
## [19] "Dominican Republic" "Egypt"              "Estonia"
## [22] ""                   "Finland"            "France"
## [25] ""                   "Germany"            "Great Britain"
## [28] "Greece"             "Hungary"            "India"
## [31] "Indonesia"          "Iran"               ""
## [34] "Italy"              "Jamaica"            "Japan"
## [37] "Kazakhstan"         "Kenya"              ""
## [40] "Malaysia"           "Mexico"             "Moldova"
## [43] ""                   "Morocco"            "Netherlands"
## [46] "New Zealand"        ""                   "Norway"
## [49] ""                   "Portugal"           ""
## [52] "Russian Federation" ""                   "Singapore"
## [55] ""                   ""                   "South Africa"
## [58] "South Korea"        ""                   ""
## [61] "Switzerland"        "Taiwan"             "Tajikistan"
## [64] "Thailand"           ""                   "Tunisia"
## [67] "Turkey"             "Ukraine"            ""
## [70] ""                   "Venezuela"
```

The predicted and actual medal counts were transformed to log2(medal count) to improve the clarity. In the graph above, we can see that there are a significant number of outliers. we have also used the geom_smooth() method which gives a regression line and 95% confidence interval.Clearly we can see there are number of data points outside, providing poor prediction results.

The dataframe “newdata” with column name outside gives us countries which are potentially outliers from the above graph. there are total 37 countries which are above or below the Confidence interval and 34 countries within the band.

Task 5

Using the regression for the 2016 medal count, make prediction for the unknown results of the upcoming 2021 Olympic games.

```
library(dplyr)
predictions_2021 = predict(model2016, pop_gdp_stlearn_2021)
```

```
#added predictions to existing dataframe
pop_gdp_stlearn_2021['Predicted_2021'] <- predictions_2021
#Round to whole numbers
pop_gdp_stlearn_2021 <- pop_gdp_stlearn_2021 %>%
mutate(across(starts_with("Predicted_2021"), round, 0))
#display predicted summary
summary(pop_gdp_stlearn_2021[4])
```

```
##  Predicted_2021
##  Min.   : 12.0
##  1st Qu.: 50.5
##  Median :111.0
##  Mean   :201.5
##  3rd Qu.:300.5
##  Max.   :729.0
```

Model Selection Tasks

Task 1

Fit linear regressions models for the total medal count in 2008 using: (i) Population alone; (ii) GDP alone; (iii) Population and GDP. Perform model selection using the Akaike Information Criterion and report your results.

```
#Build all three models
model1 <- glm(Medal2008~GDP, data = medal_pop_gdp_stlearn)
model2 <- glm(Medal2008~Population, data = medal_pop_gdp_stlearn)
model3 <- glm(Medal2008~GDP+Population, data = medal_pop_gdp_stlearn)

#summary of glm includes AIC
summary(model1)
```

```
##
## Call:
## glm(formula = Medal2008 ~ GDP, data = medal_pop_gdp_stlearn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -27.914   -5.115   -1.939    0.782   51.234
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.8702743  1.4933348   3.931 0.000199 ***
## GDP         0.0080182  0.0006501  12.333  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 133.8499)
##
```

```
##     Null deviance: 29595.1  on 70  degrees of freedom
## Residual deviance:  9235.6  on 69  degrees of freedom
## AIC: 553.13
##
## Number of Fisher Scoring iterations: 2
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = Medal2008 ~ Population, data = medal_pop_gdp_stlearn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -61.115   -8.323   -5.075    1.689   86.424
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.887e+00  2.307e+00   4.285  5.8e-05 ***
## Population  4.368e-08  1.015e-08   4.305  5.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 338.0881)
##
##     Null deviance: 29595  on 70  degrees of freedom
## Residual deviance: 23328  on 69  degrees of freedom
## AIC: 618.92
##
## Number of Fisher Scoring iterations: 2
```

```
summary(model3)
```

```
##
## Call:
## glm(formula = Medal2008 ~ GDP + Population, data = medal_pop_gdp_stlearn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -27.154   -4.856   -1.702    0.842   51.037
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.613e+00  1.506e+00   3.728 0.000395 ***
## GDP         7.613e-03  7.353e-04  10.354 1.29e-15 ***
## Population  8.435e-09  7.220e-09   1.168 0.246750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 133.1455)
##
```

```
##     Null deviance: 29595.1  on 70  degrees of freedom
## Residual deviance:  9053.9  on 68  degrees of freedom
## AIC: 553.72
##
## Number of Fisher Scoring iterations: 2
```

From summary of all three models. we have following AIC's values. model1 : 553.13 model2 : 618.92 model3 : 553.72

Model 2 is larger than other two models.The AIC's of model1 and model3 are almost similar. Model1 has minimum AIC. lets compare model1 and model3. The relative likelihood of model 'i' is given by exp((AICmin – AICi)/2) so model3 exp(553.13-553.72/2)=0.744 times as probable as the first model to minimize the information loss.

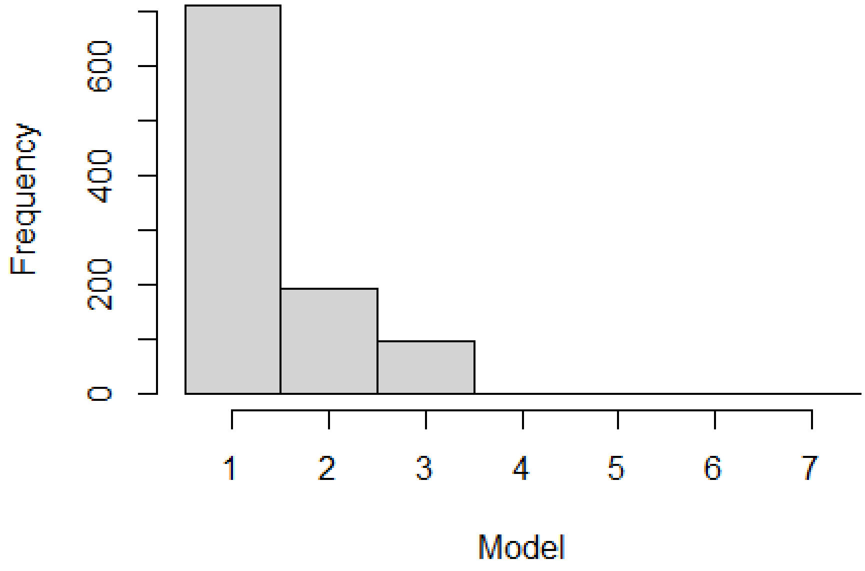
we have another valid point as per rule of thumb is that models within 1-2 of the minimum AIC have substantial support, hence model3 is acceptable.

Task 2

Use cross-validation to perform a model selection between (i) Population alone; (ii) GDP alone; (iii) Population and GDP. Comment on and report your results. Do your results agree with the model selected by the AIC?

```
z1 = medal_pop_gdp_stlearn$Medal2012
z2 = medal_pop_gdp_stlearn$GDP
z3 = medal_pop_gdp_stlearn$Population

new_data_frame = data.frame(z3,z2,z1)
#create for loop for cross validation of 1000 times

medal_winner = rep(NA, 1000)
for (iter in 1:1000){

#create random 60% of the data
random_pick<-runif(nrow(new_data_frame))>0.60

#assign 60% of the data to train set [ which is 42 rows]
training<-new_data_frame[random_pick,]

#assign remaining 40% of the data to test set [ which is 29 rows]
testing <-new_data_frame[!random_pick,]# about 40% testing (29 rows testing)

#list all model selection --gdp,population, gdp+population
model_selection = c("z1~z2", "z1~z3","z1~z2+z3")

#Replicate elements of vectors and lists
predictive_log_likelihood = rep(NA, length(model_selection))

for (i in 1:length(model_selection)){
```

```
#build model for training set
current_model = glm(formula = model_selection[i], data = training)
sigma = sqrt(summary(current_model)$dispersion)
#predict the model from testing data
predict_test = predict(current_model, testing)

#calculate the predictive log probability
predictive_log_likelihood[i] = sum(dnorm(testing$z1, predict_test, sigma,
log=TRUE))

}
medal_winner[iter] = which.max(predictive_log_likelihood)
}

#plot the histogram which shows how many times model won out of 1000 cross
validation
hist(medal_winner, breaks = seq(0.5, 7.5, 1), xlab='Model', ylab='Frequency',
main='')
```



As you see we have performed 1000 times cross validation with 60% training set and 40% test set. From histogram plot,the log probability of model 1 (model2008~GDP) was the highest. Model 1 has 700 out of 1000,which is 70% winning of the time.This results doesn’t agree with AIC in task 1 where model 1 achieved lowest AIC 553.13

similarly model3(model~gpd+population) has highest AIC in task 1 but after running 1000 cross validations , model 2(model2008~population) has 20%(200/1000) compared to 10% (100/1000) for model 3.

Task 3

Using the three fitted models from Model Selection Task 1, predict the results of Rio 2012. Which model predicts best? Justify your reasoning. Compare this result with the earlier results on model performance

```
library(dplyr)
#install.packages("Metrics")
library(Metrics)

## Warning: package 'Metrics' was built under R version 4.0.5

#create new dataframe from medal_pop_gdp_stlearn
predict_data = data.frame(Country = medal_pop_gdp_stlearn$Country, Population
= medal_pop_gdp_stlearn$Population, GDP =
medal_pop_gdp_stlearn$GDP,Actual=medal_pop_gdp_stlearn$Medal2012)

#prediction based on GDP
model1_prediction = predict.glm(model1, predict_data)
#prediction based on population
model2_prediction = predict.glm(model2, predict_data)
#prediction based on GDP and Population
model3_prediction = predict.glm(model3, predict_data)

#Add 2012 medal predictions to the predict_data dataframe

predict_data['Model 1 Predictions GDP'] <- model1_prediction
predict_data['Model 2 Predictions Population'] <- model2_prediction
predict_data['Model 3 Predictions GDP and Population'] <- model3_prediction

predict_data <- predict_data %>% mutate(across(starts_with("Model"), round,
0))

#display summary of predict_data dataframe
summary(predict_data)

##    Country             Population             GDP               Actual
##  Length:71          Min.   :3.537e+05   Min.   :    6.52   Min.   :  1.0
##  Class  :character  1st Qu.:5.513e+06   1st Qu.:   51.52   1st Qu.:  3.0
##  Mode   :character  Median :1.673e+07   Median :  229.53   Median :  6.0
##                     Mean   :7.384e+07   Mean   :  903.25   Mean   : 13.3
##                     3rd Qu.:4.958e+07   3rd Qu.:  704.37   3rd Qu.: 13.0
##                     Max.   :1.347e+09   Max.   :15094.00   Max.   :104.0
##  Model 1 Predictions GDP Model 2 Predictions Population
##  Min.   :  6.0           Min.   :10.00
##  1st Qu.:  6.0           1st Qu.:10.00
##  Median :  8.0           Median :11.00
##  Mean   : 13.1           Mean   :13.07
##  3rd Qu.: 11.5           3rd Qu.:12.00
##  Max.   :127.0           Max.   :69.00
##  Model 3 Predictions GDP and Population
##  Min.   :  6.00
##  1st Qu.:  6.00
##  Median :  7.00
##  Mean   : 13.11
##  3rd Qu.: 11.50
##  Max.   :123.00
```

```
#To compare models which one predicts best, I have used the Root mean squared error

rmse(predict_data$Actual,predict_data$`Model 1 Predictions GDP`)

## [1] 11.34355

rmse(predict_data$Actual,predict_data$`Model 2 Predictions Population`)

## [1] 17.98982

rmse(predict_data$Actual,predict_data$`Model 3 Predictions GDP and Population`)

## [1] 11.24315
```

As we know a smaller RMSE indicates a better fit of the data.The model 3 has smaller RMSE hence model 3 is best model to predict the medals based on GDP and Population.