SS 3860B - Winter 2022 Final Project

Submitted by:
Shivanika Shah (251139213)
Unnikrishnan Sivakumaran Nair (251238730)

2016 Olympics in Rio de Janeiro

Predicting medals won by each country in 2016 Olympics

TABLE OF CONTENTS

1. Introduction

The purpose of this project is to explore the use of the count regression techniques such as Poisson Regression and Negative Binomial Regression for the prediction of the total medals won by each country participating in the 2016 Olympics held in Rio de Janeiro.

Since the response is unbounded count (0,1,2,3…), we assume that a count regression model would be well suited for prediction of the total medals won by each country and to explain this in terms of the covariates used. Since the counts are not expected to be sufficiently large that a normal approximation is justified and hence using a normal linear model would not be appropriate.

2. Dataset Source and Detailed Description

The data set used for the analysis was obtained from http://www.stats.gla.ac.uk/~tereza/rp/rioolympics.csv. It contains the medals won by each country in the Olympics held in the years 2004, 2008, 2012 and 2016 and some of the country related information such as GDP, Population, number of athletes, if it is a Soviet Union nation or if it is a Communist nation. An attempt has been made to predict the total number of medals won by each country based on these country related parameters.

Response variable:	*tot*	: Total medals won by each country
Covariates:*		
	gdp	: GDP
	pop	: Population
	num_athletes	: Number of Athletes
	comm	: Communist Nation or not
	muslim	: Muslim Nation or not
	soviet	: Was part of the Soviet Union or not
	oneparty	: Is lead by a single party or not
	host	: Previously an Olympic host or not
	totgold	: Total number of Gold medals

*Covariate information collected for each year when Olympics took place.

All the information related to the covariate fields are available for the Olympics years 2004, 2008, 2012 and 2016. The features GDP, Population, Number of athletes and the number of gold medals won by each country are numeric values and the features such as Comm, Muslim, OneParty and host are factor variables.

3. Methods

Since the response is an unbounded count variable, we use the count regression techniques to model for the response variable; the total number of the medals won by each of the country.

3.1. Data Pre-processing and Exploratory Data Analysis

The data used for the analysis had country related information for each of the years when Olympics was conducted. Data was first transformed in such a way that there is one column for each of the features for analysis. The data corresponding to each year was considered as multiple levels within the data, for modelling purposes.

For the field GDP, we found that there was missing data for the countries Afghanistan, Cuba and Syria. These countries appeared in the sanctions list and that could be one of the reasons for no mention of GDP for these countries in the data.

3.1.1. Removing Unwanted Records

Since we don’t have any information for the country Syria for the year 2016 for which we are trying to predict, it doesn’t make sense to impute the values, and hence we have removed the entry for Syria.

3.1.2. Missing Value Treatment

For the countries Afghanistan and Cuba, we have adopted the linear interpolation technique to impute the missing information for the GDP for the years when the value is missing.

3.1.3. Outlier Analysis

Outliers in the data may be due to some variability in the measurement or some experimental error and in such cases these data are excluded from the data set as outliers can cause problems in the statistical analyses.

It can be seen from the plotted histograms, that there are a lot of countries that have only a small number of medals in the Olympics and most of the countries have only a limited number of athletes participating in the Olympics.

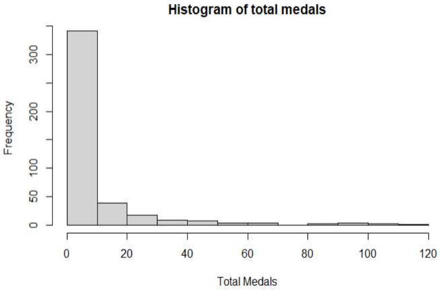


Figure 1: Histogram of Total Medals

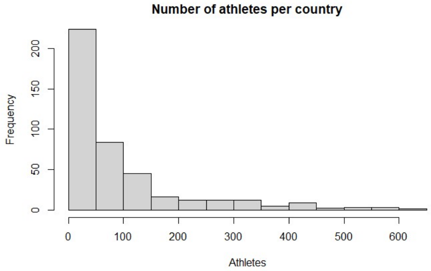


Figure 2: Number of Athletes per Country

Hence, we try to find out the outlier points in the data set that can influence the prediction value and hence cause variations in the results. For this, a half-normal plot is created. From the plot, it can be observed that the data point with the index of 424 is a potential outlier and could be removed. This is confirmed by the use of the plot with the Cooks-distance. Hence, we remove this data point from the data set.

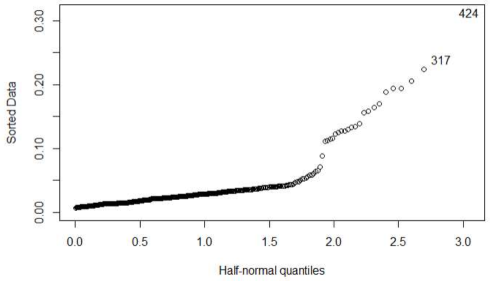


Figure 3: Half-norm Plot

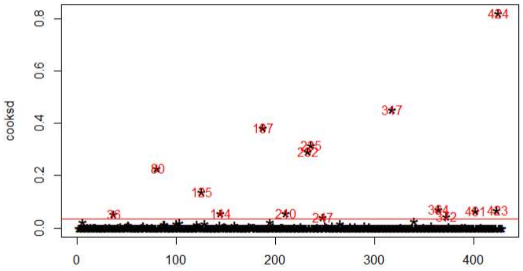


Figure 4: Cooks Distance Plot

3.1.4. Identifying Correlations

To identify relationships between the various predictors and see if they are correlated, we create a correlation plot and check VIF values. A high VIF value can cause us to draw poor conclusions about our beta coefficients, thus, it is important to understand which predictor variables are highly correlated and remove them from our analysis.

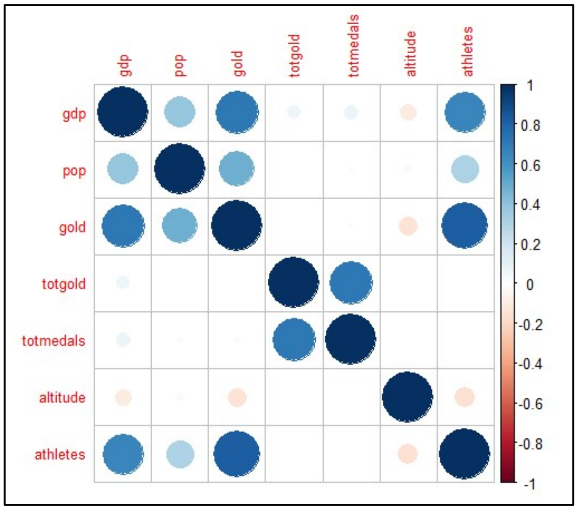


Figure 5: Correlations Plot

From the above plot, we observed that there is slight correlation between gdp, gold and athletes, totgold and totmedals. We don’t remove these predictors as the correlations and VIF values aren’t too high.

3.2. Main Analysis

3.2.1. Poisson Regression

After the data processing stage, we try to model for the response variable using the Poisson Regression technique. For testing if Poisson is an ideal model, we check the

Poissonness Plot. It can be observed that the points lie in a straight line and hence Poissonness of the response variable can be assumed.

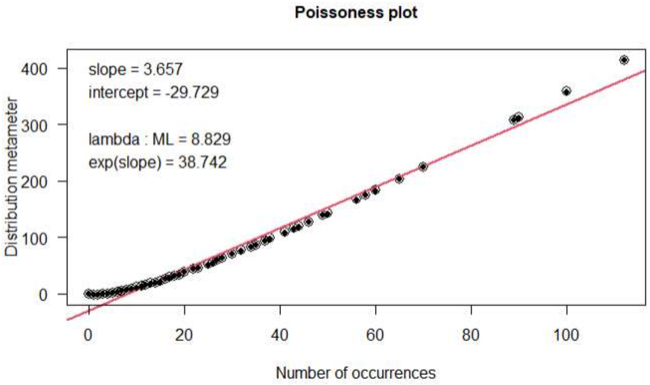
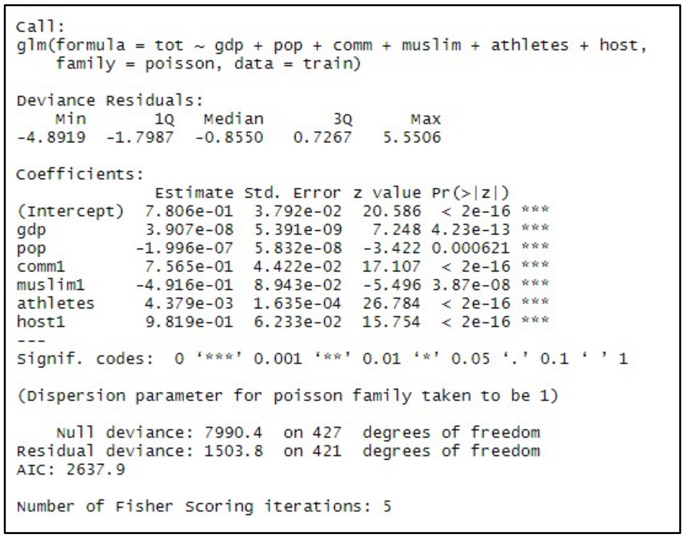


Figure 6: Poissoness Plot

In the first attempt to create a predictive model for the total number of medals won by each country, we try to use all the country related parameters and tried to fit a model. The first model has a lot of non-significant variables and after removing the non-significant variables from the model, we obtain the following result. The goodness of fit tests such as deviance test gives a p-value of 4.5e-93 which indicates a lack of fit.



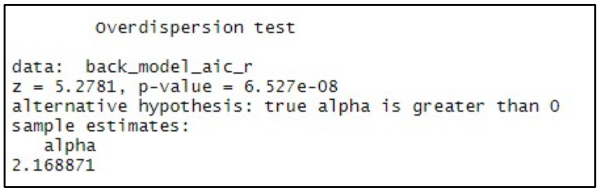
```
Call:
glm(formula = tot ~ gdp + pop + comm + muslim + athletes + host,
    family = poisson, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.8919  -1.7987  -0.8550   0.7267   5.5506

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  7.806e-01  3.792e-02  20.586  < 2e-16 ***
gdp          3.907e-08  5.391e-09   7.248 4.23e-13 ***
pop         -1.996e-07  5.832e-08  -3.422 0.000621 ***
comm1        7.565e-01  4.422e-02  17.107  < 2e-16 ***
muslim1     -4.916e-01  8.943e-02  -5.496 3.87e-08 ***
athletes     4.379e-03  1.635e-04  26.784  < 2e-16 ***
host1        9.819e-01  6.233e-02  15.754  < 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: 7990.4  on 427  degrees of freedom
Residual deviance: 1503.8  on 421  degrees of freedom
AIC: 2637.9

Number of Fisher Scoring iterations: 5
```

Figure 7: First Poisson Model Summary

We also tested for interactions and did not find any significant difference with the inclusion of the interactions and test for dispersion was performed

3.2.1.1. Dispersion Test

For the model, the dispersion factor was 3.82 and we performed a dispersion test to check the presence of overdispersion and we get a p-value of 2.1e-11 which indicated that the overdispersion is quite significant. In order to account for the overdispersion, we can try the Quasi-Poisson model or the negative binomial model. Here, since the variance is higher than the mean, a negative binomial model is preferred.



```
        Overdispersion test

data:  back_model_aic_r
z = 5.2781, p-value = 6.527e-08
alternative hypothesis: true alpha is greater than 0
sample estimates:
   alpha
2.168871
```

Figure 8: Dispersion Test for back_model_aic_r Model

3.2.1.2. Model Adequacy

- From the QQ plot of the deviance residuals, it can be observed that there are large residuals. Hence there is a deviation of the residuals from the normal. But this is something that is expected for Poisson.

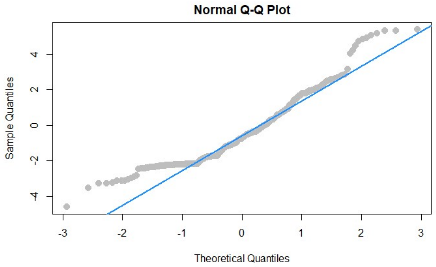


Figure 9: Normal QQ Plot

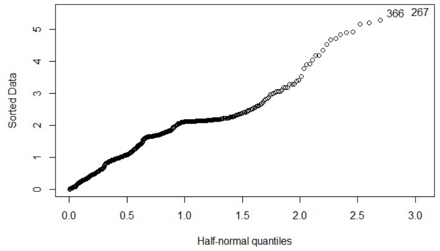


Figure 10: Half-norm Plot

- From the half normal probability plot of residuals by plotting ordered absolute residuals vs expected normal values. It can be seen that there are not many influential observations.

- A plot of the fitted values vs the residuals is also created and we can see that the points are nearly evenly distributed.

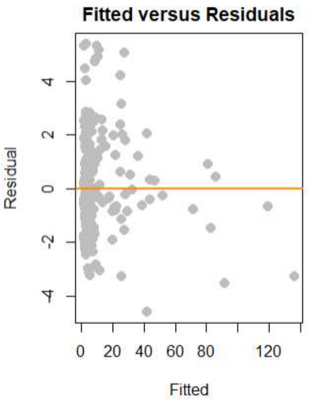
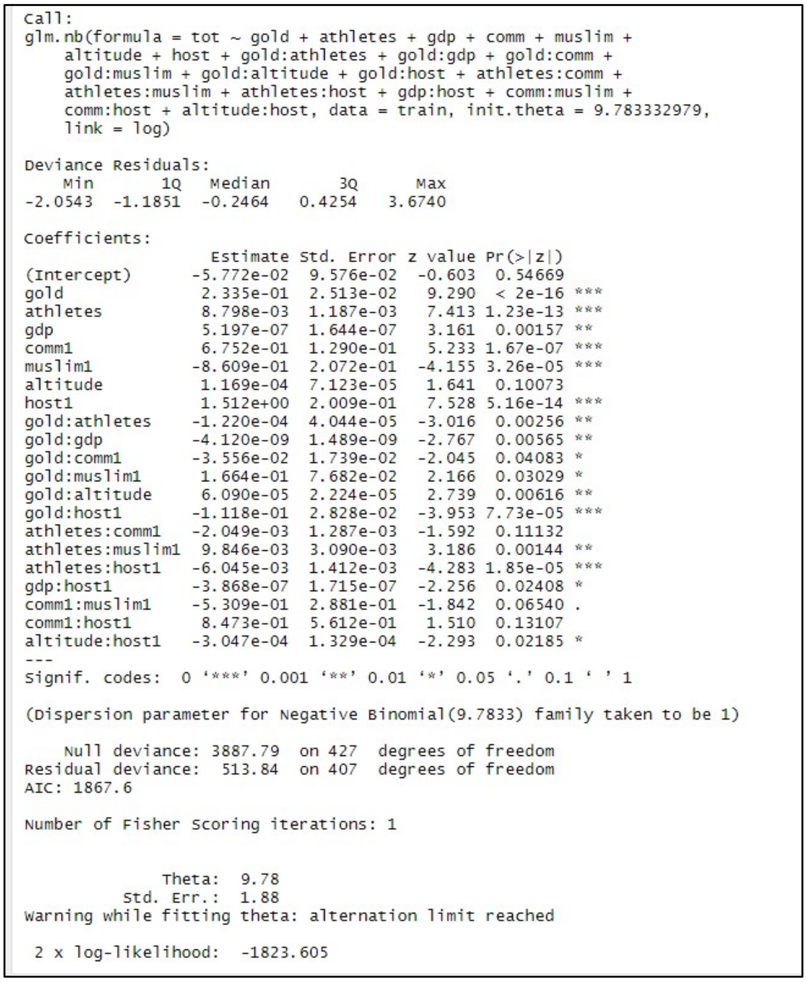


Figure 11: Fitted vs Residuals Plot

3.2.2. Negative Binomial

When overdispersion is found, the negative binomial model is a good alternative to Poisson. We fit a generalized linear model with the response following a negative binomial distribution and with a logarithm link function. We start fitting a model whose systematic part includes all linear terms and then we fit a model with only significant terms. It's observed that the model a better fir with a Chi square test statistic of 463.1855 and dispersion factor of 1.9779.

Using forward selection, backward selection and stepwise selection – AIC, BIC model selection methods, backward AIC selection model gives better results with an AIC of 2114.2. We further incorporate interaction terms in our model which improves our AIC score to 1867.6 but increases the dispersion factor to 9.7833.



```
Call:
glm.nb(formula = tot ~ gold + athletes + gdp + comm + muslim +
    altitude + host + gold:athletes + gold:gdp + gold:comm +
    gold:muslim + gold:altitude + gold:host + athletes:comm +
    athletes:muslim + athletes:host + gdp:host + comm:muslim +
    comm:host + altitude:host, data = train, init.theta = 9.783332979,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0543  -1.1851  -0.2464   0.4254   3.6740

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -5.772e-02  9.576e-02  -0.603  0.54669
gold             2.335e-01  2.513e-02   9.290  < 2e-16 ***
athletes         8.798e-03  1.187e-03   7.413 1.23e-13 ***
gdp              5.197e-07  1.644e-07   3.161  0.00157 **
comm1            6.752e-01  1.290e-01   5.233 1.67e-07 ***
muslim1         -8.609e-01  2.072e-01  -4.155 3.26e-05 ***
altitude         1.169e-04  7.123e-05   1.641  0.10073
host1            1.512e+00  2.009e-01   7.528 5.16e-14 ***
gold:athletes   -1.220e-04  4.044e-05  -3.016  0.00256 **
gold:gdp        -4.120e-09  1.489e-09  -2.767  0.00565 **
gold:comm1      -3.556e-02  1.739e-02  -2.045  0.04083 *
gold:muslim1     1.664e-01  7.682e-02   2.166  0.03029 *
gold:altitude    6.090e-05  2.224e-05   2.739  0.00616 **
gold:host1      -1.118e-01  2.828e-02  -3.953 7.73e-05 ***
athletes:comm1  -2.049e-03  1.287e-03  -1.592  0.11132
athletes:muslim1  9.846e-03  3.090e-03   3.186  0.00144 **
athletes:host1  -6.045e-03  1.412e-03  -4.283 1.85e-05 ***
gdp:host1       -3.868e-07  1.715e-07  -2.256  0.02408 *
comm1:muslim1   -5.309e-01  2.881e-01  -1.842  0.06540 .
comm1:host1      8.473e-01  5.612e-01   1.510  0.13107
altitude:host1  -3.047e-04  1.329e-04  -2.293  0.02185 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(9.7833) family taken to be 1)

    Null deviance: 3887.79  on 427  degrees of freedom
Residual deviance:  513.84  on 407  degrees of freedom
AIC: 1867.6

Number of Fisher Scoring iterations: 1


              Theta:  9.78
          Std. Err.:  1.88
Warning while fitting theta: alternation limit reached

 2 x log-likelihood:  -1823.605
```

Figure 12: Model Summary of Selected Negative Binomial Model

3.2.2.1. Handling Overdispersion

The deviance goodness of fit test is used to compare the observed with the estimated frequencies. The deviance of the final model is 513.8363 and the residual degrees of freedom is 407 so D/dfres = 1.262497 with deviance χ 2 -test p-value ≈ 0.000248.

Hence, we do not reject the null hypothesis and conclude that the model fits the data well. In addition, the ratio D/dfres is very close to the ideal value 1, so this model does not suffer from overdispersion.

3.2.2.2. Model Adequacy

- We check for linearity assumption for our finally selected model from the graph between model residuals and the fitted values. We find that the linearity assumption is violated as the residuals are not equally distributed. It can also see that the equal variance is not completely followed.

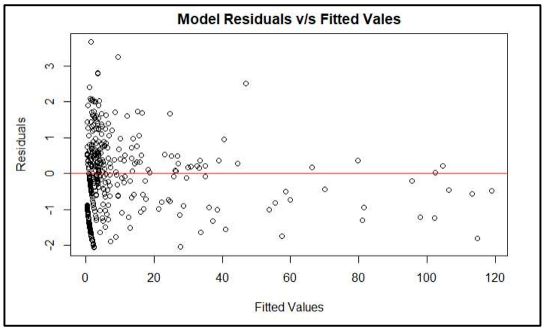


Figure 13: Model Residuals vs Fitted Values

- Further plotting the QQ plot and performing Shapiro test, we conclude that even Normality assumption holds violated as per below.

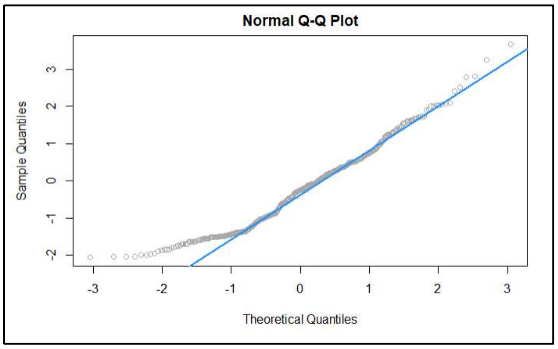


Figure 14: Normal QQ Plot

3.2.3. Zero Inflated Poisson

From data analysis we noticed that our data probably has excess zeros. So, it will be appropriate to fit zero-inflated Poisson and zero-inflated negative binomial model. Using all the predictors, we fit a zero-inflated model and observe that the model with negative binomial distribution gives best RMSE results as it performs better with over dispersed data, i.e. variance much larger than the mean.

4. Results

For the given data, data is transformed for country-wise analysis. The unwanted records for Syria are removed, missing values for Afghanistan and Cuba interpolated and outliers are treated. It's observed that there is no significant high correlation between the predictors and thus, we don't exclude them from our analysis.

For further analysing the model, the model is trained using the data for the years - 2000, 2004, 2008 and 2012. The trained model is tested model by making predictions on the data for the year 2016.

Model	Train RMSE	Test RMSE	AIC
Poisson	17.65	17.84	1890.2
Negative Binomial	17.57	17.73	1867.6
Zero-inflated Poisson	11.02	7.21	1993.03

Table 1: Model Performance

5. Conclusion and Discussion

It can be observed from the results that the zero-inflated Poisson model performs best out of the three model. However, the AIC score is better for the negative binomial model and it conforms to the checks performed for the model adequacy. It was observed that there was overdispersion in the Poisson model which we tried to overcome by using the negative binomial model.

The final model of the negative binomial regression suggests that the most important covariates are number of participating athletes, number of gold medals and whether the country has hosted Olympics previously or not.

6. References

[1] (2014, April 16). *Checking (G)LM model assumptions in R*. biologyforfun. Retrieved April 21, 2022, from https://biologyforfun.wordpress.com/2014/04/16/checking-glm-model-assumptions-in-r/

[2] (1960, April 1). *Assumptions of generalised linear model*. Cross Validated. Retrieved April 21, 2022, from https://stats.stackexchange.com/questions/32285/assumptions-of-generalised-linear-model

[3] Introduction to SAS. UCLA: Statistical Consulting Group. from https://stats.idre.ucla.edu/sas/modules/sas-learning-moduleintroduction-to-the-features-of-sas/ (accessed August 22, 2016)

[4] https://www.kaggle.com/datasets/rio2016/olympic-games?resource=download&select=countries.csv

7. Appendix

Summary Table for Code

Section Number	Section Name
1	Data Handling
2	Exploratory Data Analysis
2.1	Removing Outliers
2.2	Checking Correlations
2.3	Variable Analysis
3	Modelling
3.1	Poisson Regression Model
3.1.1	Model Adequacy
3.1.2	Model Evaluation
3.2	Negative Binomial
3.2.1	Model Adequacy
3.2.2	Model Evaluation
3.3	Zero-Inflated Model
3.3.1	Model Evaluation

Dataset:

rioolympics.csv

Code:

FinalProject_GLM.Rmd

FinalProject_GLM.html

GLM Final Project - 2016 Olympics in Rio de Janeiro

Unnikrishnan Sivakumaran Nair and Shivanika Shah

14/04/2022

```
#Importing libraries
library(corrplot)

library(AER)

library(ggplot2)
library(dplyr)

library(caret)

library(Metrics)
```

1. Data Handling

```
#Importing Data
data_olympics<-read.csv("rioolympics.csv",na.strings="#N/A")
str(data_olympics)

# now we check to see which countries have missing data.
data_olympics[!complete.cases(data_olympics),]$country

data_olympics$gdp00[data_olympics$country=="Afghanistan"]<-data_olympics$gdp0
4[data_olympics$country=="Afghanistan"]

# For cuba we chose to linearly extrapolate the data.
cuba_gdp=t(as.matrix(data_olympics[21,3:6]))
cuba_years=as.matrix(seq(1:4))
data<-cbind(cuba_years,cuba_gdp)
data<-as.data.frame(data)

colnames(data)<-c("v1","v2")
mod_cuba<-lm(v2~v1,data)


# We predict and add the value to the gdp16
data_olympics$gdp16[data_olympics$country=="Cuba"]<-predict(mod_cuba,newdata=
data.frame(v1=5))[[1]]

# For Syria it doesnt make sense to predict the value so we will ignore that
country
data_olympics<-data_olympics[-92,]
```

2. Exploratory Data Analysis

```
# Merging data year wise
year00<-data_olympics[,c("gdp00","pop00","soviet","comm","muslim","oneparty",
"gold00","tot00","totgold00","totmedals00","altitude","athletes00","host")]
year04<-data_olympics[,c("gdp04","pop04","soviet","comm","muslim","oneparty",
"gold04","tot04","totgold04","totmedals04","altitude","athletes04","host")]
year08<-data_olympics[,c("gdp08","pop08","soviet","comm","muslim","oneparty",
"gold08","tot08","totgold08","totmedals08","altitude","athletes08","host")]
year12<-data_olympics[,c("gdp12","pop12","soviet","comm","muslim","oneparty",
"gold12","tot12","totgold12","totmedals12","altitude","athletes12","host")]
colnames(year00)<-colnames(year04)<-colnames(year08)<-colnames(year12)<-c("gd
p","pop","soviet","comm","muslim","oneparty","gold","tot","totgold","totmedal
s","altitude","athletes","host")

data_olympics_c<-rbind(year00,year04,year08,year12)
rownames(data_olympics_c) <- 1:nrow(data_olympics_c)

year16<-data_olympics[,c("gdp12","pop16","soviet","comm","muslim","oneparty",
"gold16","tot16","totgold16","totmedals16","altitude","athletes16","host")]
colnames(year16)<-c("gdp","pop","soviet","comm","muslim","oneparty","gold","t
ot","totgold","totmedals","altitude","athletes","host")
data_olympics_test<-year16
rownames(data_olympics_c) <- 1:nrow(data_olympics_c)
```

2.1. Removing Outliers

```
#outlier analysis

summary(data_olympics_c[,c("gdp","pop","gold","tot","altitude","athletes")])

hist(x=data_olympics_c$tot,xlab="Total Medals",main = "Histogram of total med
als")

hist(x=data_olympics_c$athletes,xlab="Athletes",main="Number of athletes per
country")

# Cooks distance
cooksd<-cooks.distance(lm(tot~.,data_olympics_c))
plot(cooksd,pch="*",cex=2)
abline(h=10*mean(cooksd,na.rm=T),col="red")
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd, na.r
m=T),names(cooksd),""), col="red")
```

```
library(faraway)

lmodi<-lm(tot~.,data_olympics_c)
halfnorm(hatvalues(lmodi))

data_olympics_c[hatvalues(lmodi)>0.3,]
```

```
data_olympics_1<-data_olympics_c[-424,]

data_olympics_1
```

2.2 Checking Correlations

```
df <- subset (data_olympics_1, select = - c(host, oneparty, muslim, comm, sov
iet, tot))
head(df)

# Correlations plot
correlations <- cor(df)
corrplot(correlations, method="circle",tl.cex=0.8)

# Correlation Matrix
cor_matrix = cor(df)

cor_matrix

drop = findCorrelation(cor_matrix, cutoff = .85) #function that returns a vec
tor of integers corresponding to columns to remove to reduce pair-wise correl
ations.
drop = names(df)[drop]
medals_data_corr_rem = df[ , !(names(df) %in% drop)]

correlations <- cor(medals_data_corr_rem)
corrplot(correlations, method="circle",tl.cex=0.8)
```

2.3 Variables Analysis

```
ggplot(data_olympics_1,aes(x=tot,y=gold))+
  geom_point(size=1)+
  xlab("Total number of medals")+
  ylab("proportion of golds to total medals")+geom_smooth(method="lm",col="re
d")



# Compare proportion of medals with number of athletes.

ggplot(data_olympics_1,aes(x=athletes,y=tot/totmedals))+
  geom_point()+geom_smooth()

summary(lm((tot/totmedals)~athletes,data_olympics_1))

# Compare proportion of medals with gdp
ggplot(data_olympics_1,aes(x=gdp,y=tot/totmedals))+
  geom_point()

summary(lm((tot/totmedals)~gdp,data_olympics_1))

# Check muslim countries
```

```
ggplot(data_olympics_1,aes(x=muslim,y=tot))+geom_boxplot()+xlab("Muslim or no
t?")+
  theme(panel.background = element_rect(fill="transparent",colour = NA),
        plot.background = element_rect(fill="transparent",colour=NA),
        panel.border = element_rect(fill=NA,colour="black",size=1))

# Check ex soviet countries
ggplot(data_olympics_1,aes(x=soviet,y=tot))+geom_boxplot()+xlab("Soviet or no
t?")+
  theme(panel.background = element_rect(fill="transparent",colour = NA),
        plot.background = element_rect(fill="transparent",colour=NA),
        panel.border = element_rect(fill=NA,colour="black",size=1))

# Check communist
ggplot(data_olympics_1,aes(x=comm,y=tot))+geom_boxplot()+xlab("Socialist or n
ot?")+
  theme(panel.background = element_rect(fill="transparent",colour = NA),
        plot.background = element_rect(fill="transparent",colour=NA),
        panel.border = element_rect(fill=NA,colour="black",size=1))

# Check host
ggplot(data_olympics_1,aes(x=host,y=tot))+geom_boxplot()+xlab("Host or not?")
+
  theme(panel.background = element_rect(fill="transparent",colour = NA),
        plot.background = element_rect(fill="transparent",colour=NA),
        panel.border = element_rect(fill=NA,colour="black",size=1))

ggplot(data_olympics_1,aes(x=gdp,y=(tot)**2))+geom_point()+geom_smooth()
```

3. Modelling

3.1 Poisson Regression Model

```
train <- data_olympics_c

test <- data_olympics_test[,colnames(train)]

# Full initial model
mod_poison1<-glm(tot~gdp+pop+soviet+comm+muslim+oneparty+altitude+athletes+ho
st,family=poisson,data=train)
summary(mod_poison1)

# Removing non-significant variables

mod_poison2<-glm(tot~gdp+pop+comm+muslim+athletes+host,family=poisson,data=tr
ain)
summary(mod_poison2)

pchisq(deviance(mod_poison2),df.residual(mod_poison2),lower=FALSE)

## [1] 1.906498e-121
```

```
#We observe a lack of fit. So checking for overdispersion.

#Checking dispersion factor
dp <- sum(residuals(mod_poison2,type="pearson")^2)/mod_poison2$df.res
dp

## [1] 3.677141

#Testing if the dispersion is significant


# Here the dispersion is positive and its high. Hence we try Quasi-Poisson mo
del may not be appropriate.

## Interactions test

mod_poison_p2<-glm(tot~(gdp+pop+comm+muslim+athletes+host)^2,family=poisson,d
ata=train)
summary(mod_poison_p2)

back_model_aic = step(mod_poison_p2, direction = "backward", trace = 0)
summary(back_model_aic)

back_model_aic_r<-glm(formula = tot ~ gdp + comm + muslim + athletes + host +
    gdp:pop + gdp:comm + pop:muslim + pop:athletes +
    pop:host + comm:athletes + comm:host + muslim:athletes +
    athletes:host, family = poisson, data = train)

summary(back_model_aic_r)

#Testing if the dispersion is significant for best model till now - back_mode
l_aic_r
dispersiontest(back_model_aic_r,trafo=1)

pchisq(deviance(back_model_aic_r),df.residual(back_model_aic_r),lower=FALSE)

## [1] 4.92007e-74

# p-value of 2.1e-11 which indicated that the overdispersion is quite signifi
cant
```

3.1.1 Model Adequacy

```
# QQ Plot for Normality
qqnorm(resid(mod_poison2), col = "grey",pch=20,cex=2)
qqline(resid(mod_poison2), col = "dodgerblue", lwd = 2)
```

```
# Checking Linearity and Equal variance
par(mfrow=c(1,2))
plot(fitted(mod_poison2), resid(mod_poison2), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```

```
library(faraway)
halfnorm(residuals(mod_poison2))
```

```
library(vcd)

distplot(train$tot, type="poisson")
```

3.1.2. Model Evaluation

```
test_data<- test[,c("gdp","pop","comm","muslim","athletes","host")]

#Test RMSE
pred_values<-predict.glm(mod_poison2,test_data)

actual<-test$tot
rmse(actual,pred_values)

## [1] 17.84109

#Train RMSE
pred_values<-predict.glm(mod_poison2,train)

actual<-train$tot
rmse(actual,pred_values)

## [1] 17.64796

# Model 3 : Backward Selection - AIC
full_model = mod_poison2
summary(full_model)
```

3.2 Negative Binomial

```
library(MASS)

m1 <- glm.nb(tot ~ ., data = train)

summary(m1)

vif(m1)

#install.packages('pscl')
library(pscl)

odTest(m1)

#As test statistic 314.0968 exceeds 2.7055 with a p-value of  = < 2.2e-16  th
us, null of the Poisson restriction is rejected in favour of the negative bin
omial regression.

#install.packages('msme')
library(msme)

P__disp(m1)

## pearson.chi2   dispersion
##   444.877366     1.071994

# nb model using significant predictors only
drop1(m1, test = "Chisq")

m2 <- glm.nb(tot ~ gdp + comm + muslim + gold + altitude + athletes + host, d
ata = train)
summary(m2)

vif(m2)

##       gdp     comm1   muslim1      gold  altitude  athletes     host1
## 1.3462066 0.9902429 1.1598321 2.4678158 0.9122724 3.0066358 1.7594003

#install.packages('pscl')
library(pscl)
odTest(m1)

#As test statistic 463.1855 exceeds 2.7055 with a p-value of  = < 2.2e-16  th
us, null of the Poisson restriction is rejected in favour of the negative bin
omial regression.

#install.packages('msme')
library(msme)
P__disp(m1)
```

```
## pearson.chi2   dispersion
##   444.877366     1.071994
```

```
drop1(m2, test = "Chisq")
```

```
library(faraway)
AIC(m1,m2)
```

```
##    df      AIC
## m1 14 2120.325
## m2  9 2114.205
```

```
# m2 is better

m2 <- glm.nb(tot ~ gdp + comm + muslim + gold + altitude + athletes + host, d
ata = train)
full_model = m2
summary(full_model)

null_model = glm.nb(tot ~ 1, data = train)
summary(null_model)

# AIC and BIC

# Model 1 : Forward selection - AIC
null_model = lm(tot ~ 1, data = train)
for_model_aic = step(null_model, scope = tot ~ gdp + comm + muslim + gold + a
ltitude + athletes +
    host, direction = "forward", trace = 0)
summary(for_model_aic)

# Model 2 : Forward selection - BIC
n = nrow(train)
for_model_bic = step(null_model, scope = tot ~ gdp + comm + muslim + gold + a
ltitude + athletes +
    host, direction = "forward", trace = 0, k = log(n))
summary(for_model_bic)

# Model 3 : Backward Selection - AIC
back_model_aic = step(full_model, direction = "backward", trace = 0)
summary(back_model_aic)

# Model 4 : Backward Selection - BIC
n = nrow(train)
back_model_bic = step(full_model, direction = "backward", trace = 0, k = log(
n))
summary(back_model_bic)

# Model 5 : Stepwise Selection - AIC
step_aic = step(null_model, scope = tot ~ gdp + comm + muslim + gold + athlet
es +
    host, direction = "both", trace = 0)
summary(step_aic)

plot(step_aic)

# Model 6 : Stepwise Selection - BIC
n = nrow(train)
step_bic = step(full_model, direction = "both", trace = 0, k = log(n))
summary(step_bic)

# Model 3 is better.


summary(back_model_aic)

vif (back_model_aic)

##       gdp     comm1   muslim1      gold  altitude  athletes     host1
## 1.3462066 0.9902429 1.1598321 2.4678158 0.9122724 3.0066358 1.7594003

## Interaction Terms - taking most significant terms

interaction <- glm.nb(tot ~ (gold + athletes + gdp + comm + muslim + altitude
+ host)^2, data = train)

## Warning in glm.nb(tot ~ (gold + athletes + gdp + comm + muslim + altitude
+ :
## alternation limit reached

summary(interaction)

drop1(interaction, test = 'Chisq')

back_model_aic_int = step(interaction, direction = "backward", trace = 0)

summary(back_model_aic_int)

#Deviance
dev.negbin<-summary(back_model_aic_int)$deviance
df.negbin<-summary(back_model_aic_int)$df.residual
dev.negbin/df.negbin

## [1] 1.262497

1-pchisq(dev.negbin,df.negbin)

## [1] 0.0002484877

#no overdispersion
```

3.2.1. Model Adequacy

```
library(lmtest)

# 1. Linearity - Violated
plot(fitted.values(back_model_aic_int), residuals(back_model_aic_int), xlab =
"Fitted Values", ylab = "Residuals",
     main = "Model Residuals v/s Fitted Vales")
abline(h = 0, col = 'red')
```

```
# 2. Equal Variance  - Violated
bptest(back_model_aic_int)

# p-value is 2.2e-16

# 3. Normality - Violated
qqnorm(resid(back_model_aic_int), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(back_model_aic_int), col = "dodgerblue", lwd = 2)

residuals <-resid(back_model_aic_int)
shapiro.test(residuals)

# p-value = 2.011e-08

library(faraway)
halfnorm(residuals(back_model_aic_int))
```

3.2.2. Model Evaluation

```
test_data<- test[,c("gdp","gold","comm","muslim","altitude","athletes","host"
)]

train$predy <- predict(back_model_aic_int,type="response")
par(mfrow=c(1,1))
plot(tot ~ predy, data = train,pch = 16,xlab="Predicted response",ylab="Actua
l response")

predict.glm(back_model_aic_int,type="response",se.fit=T)


#Test RMSE
pred_values<-predict.glm(back_model_aic_int,test_data)

actual<-test$tot
rmse(actual,pred_values)

## [1] 17.73392

#Train RMSE
pred_values<-predict.glm(back_model_aic_int,train)
```

```
actual<-train$tot
rmse(actual,pred_values)

## [1] 17.57536

# Model 3 : Backward Selection - AIC
full_model = back_model_aic_int


back_model_aic = step(full_model, direction = "backward", trace = 0)
summary(back_model_aic)
```

3.3 Zero-Inflated Model

```
library(pscl)
mod_poison3 <- zeroinfl(tot~gdp+pop+comm+muslim+athletes+host,dist="poisson",
data=train)

AIC(mod_poison2, mod_poison3)

##             df      AIC
## mod_poison2  7 2637.852
## mod_poison3 14 2281.225

mod_poison4 <- zeroinfl(tot~gdp+pop+comm+muslim+athletes+host,dist="negbin",d
ata=train)

AIC(mod_poison4, mod_poison3)

##             df      AIC
## mod_poison4 15 1998.424
## mod_poison3 14 2281.225

summary(mod_poison4)

#mod_poison4 is better
```

3.3.1 Model Evaluation

```
test_data<- test[,c("gdp","pop","comm","muslim","athletes","host")]

#Test RMSE
pred_values<-predict(mod_poison4,test_data)

actual<-test$tot
rmse(actual,pred_values)

## [1] 7.210511

#Train RMSE
pred_values<-predict(mod_poison4,train)
```

```
actual<-train$tot
rmse(actual,pred_values)

## [1] 11.01851

# Model 3 : Backward Selection - AIC
full_model = mod_poison4
summary(full_model)

back_model_aic = step(full_model, direction = "backward", trace = 0)

summary(back_model_aic)

AIC(back_model_aic, mod_poison4)

##                df      AIC
## back_model_aic  9 1993.030
## mod_poison4    15 1998.424
```