Math 651 Final Project

Mary Peng, Hillary Dunn, and Max Kearns

December 17, 2018

1   Abstract

The Olympics are an important geopolitical event, where countries vie for national pride and acclaim. As such, the models explored in this paper attempt to explain some of the factors that go into a successful Olympic season. Linear and Negative Binomial modelling provide an interesting dissection of how population, GDP, being an Olympic Host, and having a command economy may effect a particular country's chances in the Olympics. We found that each of these factors correlate with a higher medal count, and provide a model to predict the outcomes of future Olympics, given these factors.

2   Introduction

Every four summers, the Olympic Games become the center of the world's attention, as elite athletes seek honor for both themselves and for their countries. Many countries associate tremendous national pride with their medal counts, since a nation's athletic competence also projects its soft power.

Given the fierce competition and high profile nature of medal counts, one may wonder what factors influence the number of medals that a country wins at the summer Olympics. Certainly countries with the largest economies and populations, such as the United States and China, commonly dominate the top of the billboard. However, Azerbaijan, which ranks 91st in population and 72nd in total GDP, also ranked in the top 20 countries by total medal count in the 2016 Summer Olympics.

This paper develops and compares several regression models using predictors like GDP per Capita, Population, whether a country has or will host a summer Olympics, and whether a country was ever a Soviet or Communist state. We pool country-year medal count observations from the 1996, 2004, 2008, and 2012 Olympics, and then project the model onto the 2016 Olympics to understand the model's prediction accuracy. We focus specifically on countries that have won at least one Olympic medal in a given summer Olympics, because of data availability.

3   Methods and Materials

3.1   Data

3.1.1   Dependent Variable

We model on the total medal count, by participating country and year during which a Summer Olympics occurred. The medal count data come from a website called *www.medalspercapita.com*. For any given Summer Olympics, this website only lists the 80 - 90 countries that have won at least one medal, out of the 200 or so nations that participate in each summer Olympics.

Table 1 and Figure 1 both illustrate that the medal counts exhibit right-skewed distribution. Among countries that have won at least one medal, the median medal count is 5, while the mean medal count is 12. The skewed distribution suggests the need for a log transformation (in the case of a linear model), which can also ensure that the predictions are positive numbers. Moreover, given we are modeling on count data, where large counts rarely occur, a Poisson model may be more appropriate than linear regression.

3.1.2   Predictor Variables

Based on existing literature (Bernard and Busse 2006, Goldman Sachs 2016, Bian 2005), potentially significant predictors of medal count include a country's population size, GDP per capita, whether or not a country has hosted or will host a summer Olympics, and whether or not a country has ever been a command economy.

X_1 : *Population*: This variable indicates a country's population size in a given Olympic year. The left histogram in Figure 2 illustrates the right-skewed distribution of population across countries that won at least one medal (with China and India representing the far right). The right chart shows that applying a natural log transformation reduces the skewness and the influence of extreme X variable values on the model.

$X_2 : GDP/Capita$: This variable indicates a country's GDP per capita in a given Olympic year. These data show a similar distribution to the population variable, so we also applied the natural log transformation as well.

$X_3 : Host(1/0)$: This is a binary predictor variable with value equal to 1 if a country has hosted a summer Olympics in the 8 years prior to the Olympics in consideration, or will host in the next 8 years. For example, Greece, which hosted the 2004 Olympics, will have value = 1 for the 1996, 2000, 2004, 2008, and 2012 Olympics. 25 observations are classified as a host by this measure.

$X_4 : CommandEconomy(1/0)$: This binary predictor variable indicates whether the country was once a member of the Soviet Union or Yugoslavia, or ruled by a Communist party. We assembled a list of formerly or currently Soviet, Yugoslavian, or Communist countries using *www.worldatlast.com*, *Wikepedia*'s list of communist parties, and *www.sporcle.com*. 111 data points are classified as former Soviet Union or Communist. We hypothesize that former or current command economy countries will have higher medal count than other similar countries, because these countries can effectively centralize resources to invest into national sports teams.

We generate a correlation matrix (Table 3) and paired scatterplot (Figure 3) to illustrate the relationships between the variables and identify potential dangers of multicollinearity if we use these variables in a linear regression.

The scatter plot matrix in Figure 3 shows positive correlation between count of medals and population size. Overall, we do not observe strong linear relationship (defined here as r > 0.5) between medal count and any of the above predictor variables under consideration. Population size has the strongest linear correlation with medal count (correlation = 0.48), and Command Economy (Y/N) has the weakest linear correlation (correlation = 0.09).

The correlation matrix in Table 3 also suggests that collinearity between variables will not be a significant problem. Log GDP per capita and Command Economy (1/0) has the strongest linear correlation, but their correlation coefficient is still only -0.299. This finding alleviates concern of multicollinearity when we build linear regression using these variables.

3.2 Model Building and Selection

We will consider two types of models: a linear model and a Poisson model, given the count nature of the data.

3.2.1 Linear Model

As previously mentioned, we take the natural log of the medal count variable as the dependent variable, so that it satisfies the normal distribution assumption for linear regression. We then select the best variable subset using C_p statistic, adjusted R^2 value, AIC, and PRESS. Table 4 shows that the model containing all four predictor variables has the smallest C_p statistic, the largest adjusted R^2 value, the smallest AIC, and the smallest PRESS value.

Our final linear regression specification is:

$$\hat{Y} = -9.63 + 0.44X_1 + 0.41X_2 + 0.87X_3 + 1.03X_4 \quad (1)$$

where $Y = \ln(\text{Count})$, $X_1 = \ln(\text{population})$, $X_2 = \ln(\text{gdp/capita})$, $X_3 = \text{host}(1/0)$, and $X_4 = \text{Command Economy } (1/0)$

Table 7 shows the coefficient values and their standard deviations. We conduct inference on each predictor variable, with

$$H_0 : \beta_i \text{ vs. } H_a : \beta_i \neq 0$$

. We reject H_0 at $\alpha = 0.05$ if $t^* = \beta_i/s_i > t(\alpha)$. We find that all 4 predictor variables are statistically significant at $\alpha = 0.05$.

The residual vs. fitted plot shows distinct diagonal bands, particularly one that bounds all other observations from the lower right. This lower bound corresponds to the minimum value of 1 medal count among our observations. The other individual bands reflect the discrete nature of the dependent variable. The residuals vs. fitted plot also shows potential heteroskedasticity, with larger variance among the smaller fitted values and smaller variance among the larger values. This distribution makes sense because we expect the majority of countries to win relatively few medals, and a few countries to win large numbers of medals. We run a Breusch-Pagan test to assess the existence of heteroskedasticity.

Assuming $Var(\epsilon_i) = \sigma_i^2$ such that $\log_e \sigma_i^2 = \gamma_0 + \gamma_1 X_i$:

$$H_0 : \gamma_1, \text{vs. } H_a : \gamma_1 \neq 0$$

At significance level $\alpha = 0.05$, if the p-value of the Breusch-Pagan test is less than α then we reject H_0 and conclude H_a, otherwise we fail to reject H_0. Rejecting H_0 in favor of H_a means that we conclude that the variance is non-constant.

The p-value $= 0.001587 < \alpha$; we thus conclude that our variance is not constant. However, the heteroskedasticity does not cause serious concern; as discussed above, we expect the model to predict high medal count for very few observations (and thus lower variance) and low medal count for the majority of observations (and thus higher variance).

The Normal Q-Q plot (Figure 4) shows that the residuals generally follow a normal distribution. We validate the normality assumption by running a Lilliefors test.

$$H_0 : \text{Sample comes from a } N(\mu, \sigma^2) \text{ distribution}$$

$$H_a : \text{Sample does not come from a } N(\mu, \sigma^2) \text{ distribution}$$

At significance level $\alpha = 0.05$, we reject H_0 if the p-value is less than α; otherwise we fail to reject H_0.

The p-value= $0.2332 > \alpha$, thus we fail to reject H_0 in favor of H_a and conclude that we can reasonably assume that the error terms follow a normal distribution.

Lastly, we check for influential cases. The Residuals vs. Leverage plot shows a few potential outliers. Appendix C summarizes the influential cases in our model, using Cook's Distance, DFFITS, DFBETAS, and COVRATIO. We consider an observation as an outlier if at least one of the above diagnostics flags it as influential. In total, we flag 37 out of 387 observations as outliers. Many of them are highly populous countries, such as the United States, China, and India. Additionally, many of the influential cases have hosted or will host the summer Olympics and have high medal count. We chose to keep these observations in the model, because they constitute 10% of our sample size and contain valuable information.

Instead, we attempt to correct for non-constant variance and outliers by using robust linear regression, with Huber and Bisquare weights. As shown in table 7, the coefficients generated through robust linear regression, using Bisquare weights, fall within +/- 5% of the corresponding OLS regressions. The robust regression's standard errors are slightly larger than those of the OLS regression. We find similar results when using Huber weights. These results show that the influential cases in fact have limited influence on our model fit.

3.2.2   Generalized Linear Model

Given the dependent variable is medal count, we also considered fitting a Poisson model: We used the same variables as the linear model: $ln(population)$, $ln(GDP/capita)$, $host$, and $command\ economy$, and obtained the following model:

$$\ln(E(Y_i|X_i)) = \ln(\lambda_i) = -11.29 + 0.51X_1 + 0.52X_2 + 0.31X_3 + 1.02X_4$$

All 4 predictors are significant at $\alpha = 0.05$. However, this model has dispersion = 6.621, which far exceeds the assumption of dispersion = 1 for Poisson distributions and would lead to under-estimation of standard error.

To account for the overdisperion, we consider a negative binomial model. Table 5 shows the Mallow's Cp statistic and AIC across different variable subsets using the negative binomial model form. Again, the full

model has the lowest Cp and AIC. It has the following form:

$$\ln(E(Y_i|X_i)) = \ln(\lambda_i) = -10.17 + 0.50X_1 + 0.40X_2 + 0.69X_3 + 1.03X_4$$

The above model has dispersion of 1.13, which gives us confidence in conducting inference on the model coefficients. We use the same hypothesis test as described previously, and find that all 4 predictor variables are statistically significant at $\alpha = 0.05$.

Figure 5 shows the residual plots using the above negative binomial regression. Overall, it appears very similar to the residual vs. fitted plots for the linear regression model, so we will not discuss it in detail.

Independence among observations constitutes a key assumption for negative binomial regressions. Since our model uses panel data, observations over time for the same country will exhibit serial autocorrelation, thereby violating the independence assumption. Moreover, the medal counts among countries within a given year are not completely independent, because of the fixed total number of medals awarded in a single Olympic year.

To address the above non-independence, we tried adding year and country fixed effects to the negative binomial regression. The models take the form:

$$\ln(Y_i t) = \beta_0 + \beta_1 X_1 it + \beta_2 X_2 it + \beta_3 X_3 it + \beta_4 X_4 it + d_t$$

$$\ln(Y_i t) = \beta_0 + \beta_1 X_1 it + \beta_2 X_2 it + \beta_3 X_3 it + \beta_4 X_4 it + v_i$$

where d_i and v_i represent dummies for year t and country i, respectively. The year fixed effect accounts for changing number of sports and number of country participants. The country fixed effect accounts for unobserved factors that vary little over time, such as a country's investment in national sports teams, cultural attitude toward the Olympics, etc.

4 Results

4.1 Comparison of Models

Table 7 compares the coefficients across different model specifications. We observe the following:

- Compared to the OLS linear model, the robust linear model produces slightly larger standard errors, but similar coefficient magnitudes. This finding suggests that outliers do not substantially influence the OLS linear model fit.

- As expected from the dependent variable's over-dispersion, the Poisson regression underestimates the standard error, compared to the other 4 model specifications.

- Negative binomial produces similar coefficient and standard error estimates as OLS and robust linear regressions.

- Adding year / country fixed effects to the negative binomial regression does not substantially change the magnitude of model coefficients.

We focus the remainder of comparisons on the OLS linear regression and negative binomial regression (without fixed effects).

4.2 AIC, MSE, and Actual vs. Predicted

Table 6 compares the AIC and MSE of the two models. Linear regression has both lower AIC and MSE, which would make it the more preferable model. The Actual vs. Predicted plots in Figure 6 corroborate this finding, and show that the higher MSE in the negative binomial model results from a few dramatic over-predictions.

4.3 2016 Out-of-sample Projection:

We supplement the in-sample diagnostics with an out-of-sample projection on the 2016 Rio Summer Olympics. The medal count data for 2016 follows a similar distribution to the overall training data.

To determine the accuracy of each of our models we determined 90% prediction intervals for each of the 81 points in the 2016 medal count data. Then we calculated the ratio of data points that had their actual counts within their respective prediction intervals to total number of data points. 74 out of 81 points had medal counts within their respective 90% prediction intervals.

We performed the same analysis on the negative binomial model, using the *ciTools* package. The package performs bootstrapping to obtain many values of the negative binomial parameter, generates multiple random samples from the negative binomial distributions, and then takes the 90% credible interval as the prediction

interval. As in the linear regression case, the 90% prediction intervals captured the majority of the actual observations.

Table 6 compares the MSE for the 2016 out-of-sample projection. The linear model has substantially lower MSE than the negative binomial model.

5   Discussion and Conclusions

The OLS linear regression with all 4 predictor variables performed the best, out of the various models we have considered. This is surprising, given we had expected the negative binomial model to better capture the dependent variable's distribution, and hence perform as well, if not better, than linear regression.

All of our models indicate that population, GDP per capita, being a past, current, or future Olympics host, and being a command economy all correlate positively with higher medal count. Countries that belong to the command economy club on average earn 2.8 ($\exp(1.03) = 2.801$) times more medals than a similar, democratic country.

Future models can expand upon the methodology in this paper in the following ways:

1) Explore using mixed linear model to address the serial autocorrelation among country observations over time, given we are using panel data.

2) Our models only apply to countries that won at least one medal. Future studies can leverage logistic regression to identify factors correlated with whether or not a country can win at least a single medal, and then combine the forecasted probability of winning a medal with the number of medals we expect a country to win.

6   Bibliography

7 Appendix A
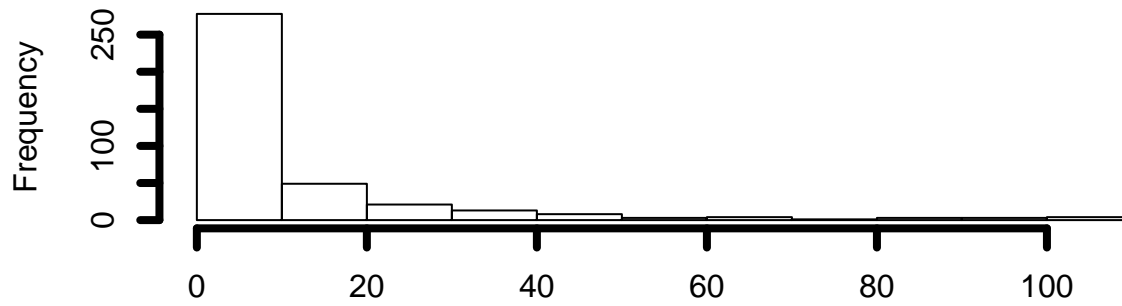
Figure 1: Distribution of Count

Table 1: Summary of Numerical Variables

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Count	1.0000	2.000	5.000	11.78811	12.5	110.0
Population	96369.0000	5150391.000	14447562.000	65765693.91731	48164957.0	1350695000.0
GDP/Cap	123.8762	2159.839	7029.231	15349.91682	24171.9	101668.2

Table 2: Summary of Categorical Variables

	Off	On
Host	362	25
Communist/Soviet	276	111

Table 3: Correlation Matrix

	count	log_pop	log_gdp_per_cap	host	comm_soviet
count	1.0000000	0.4839466	0.2291600	0.4594891	0.0871524

	count	log_pop	log_gdp_per_cap	host	comm_soviet
log_pop	0.4839466	1.0000000	-0.1901839	0.2548345	-0.1480479
log_gdp_per_cap	0.2291600	-0.1901839	1.0000000	0.1636597	-0.2989122
host	0.4594891	0.2548345	0.1636597	1.0000000	-0.0736893
comm_soviet	0.0871524	-0.1480479	-0.2989122	-0.0736893	1.0000000

Untransformed   **Transformed**
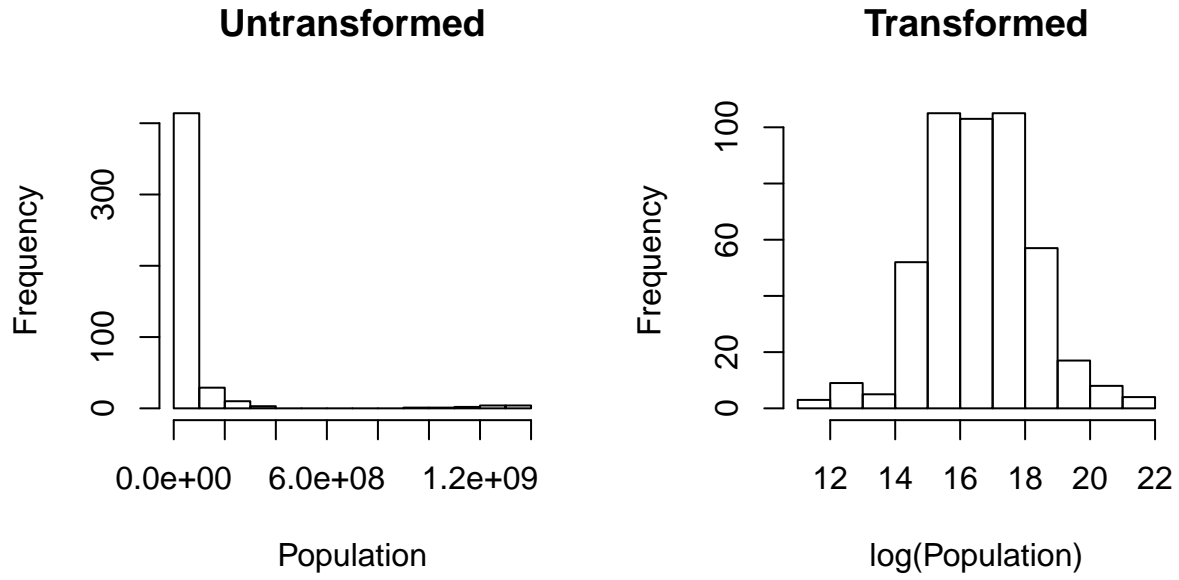
Figure 2: Histogram of the Population Variables

Table 4: Model Selection for Linear Model

	1	2	3	4	Cp	aR2	AIC	PRESS
1	1	0	0	0	250.23	0.22	2165.40	481.0928
1	0	0	1	0	319.74	0.13	2176.89	532.7428
1	0	1	0	0	357.45	0.09	2247.80	561.6892
1	0	0	0	1	418.54	0.01	2265.72	608.1812
2	1	1	0	0	123.00	0.38	2109.14	498.7497
2	1	0	1	0	197.75	0.29	2100.94	384.4047
2	1	0	0	1	220.38	0.26	2154.15	458.7142
2	0	1	1	0	273.23	0.19	2166.76	498.7497
2	0	0	1	1	302.56	0.16	2171.60	520.7314
2	0	1	0	1	317.78	0.14	2238.77	532.9378
3	1	1	0	1	23.77	0.50	2059.22	366.8114
3	1	1	1	0	100.33	0.41	2061.48	309.0094
3	1	0	1	1	164.64	0.33	2084.38	416.4538
3	0	1	1	1	230.32	0.25	2153.04	466.7762

12

	1	2	3	4	Cp	aR2	AIC	PRESS
4	1	1	1	1	5.00	0.53	2008.82	294.4012
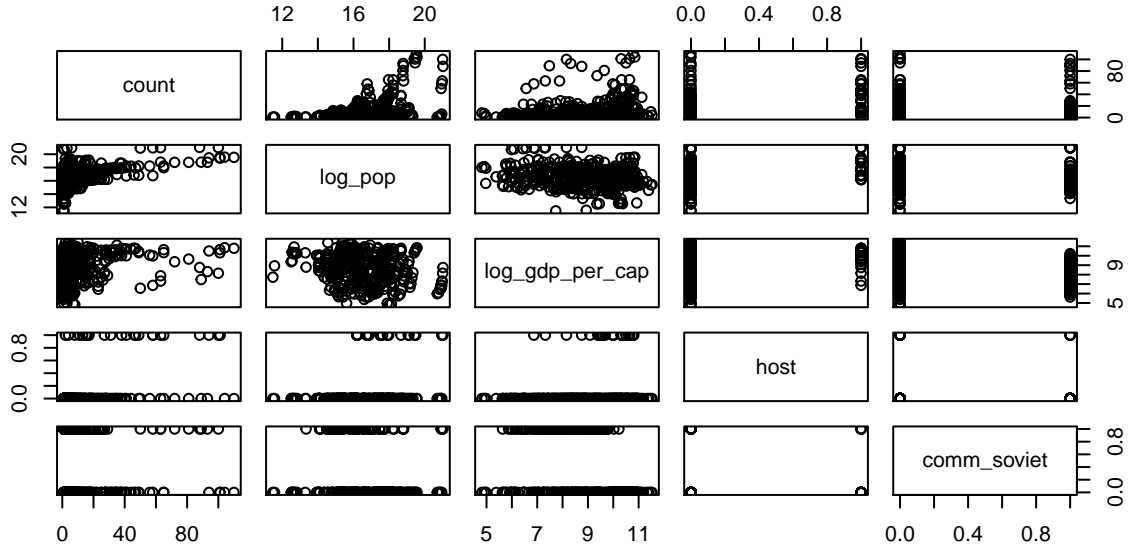
Figure 3: Scatter Matrix

Table 5: Model Selection Diagnostics for a Negative Binomial Model

Pop	GDP/C	Host	Soviet	Parameters	Cp.nb	AIC.nb
1	0	0	0	2	198.46	2497.82
0	0	1	0	2	215.98	2633.68
0	1	0	0	2	336.41	2658.69
0	0	0	1	2	370.52	2695.53
1	0	1	0	3	108.70	2474.29
1	1	0	0	3	119.18	2428.33
1	0	0	1	3	180.88	2489.15
0	1	1	0	3	199.48	2601.80
0	0	1	1	3	206.80	2626.84
0	1	0	1	3	318.21	2614.08
1	1	0	1	4	58.39	2338.79
1	1	1	0	4	60.95	2417.52
1	0	1	1	4	87.77	2457.19
0	1	1	1	4	178.38	2556.83

14

Pop	GDP/C	Host	Soviet	Parameters	Cp.nb	AIC.nb
1	1	1	1	5	5.00	2319.23
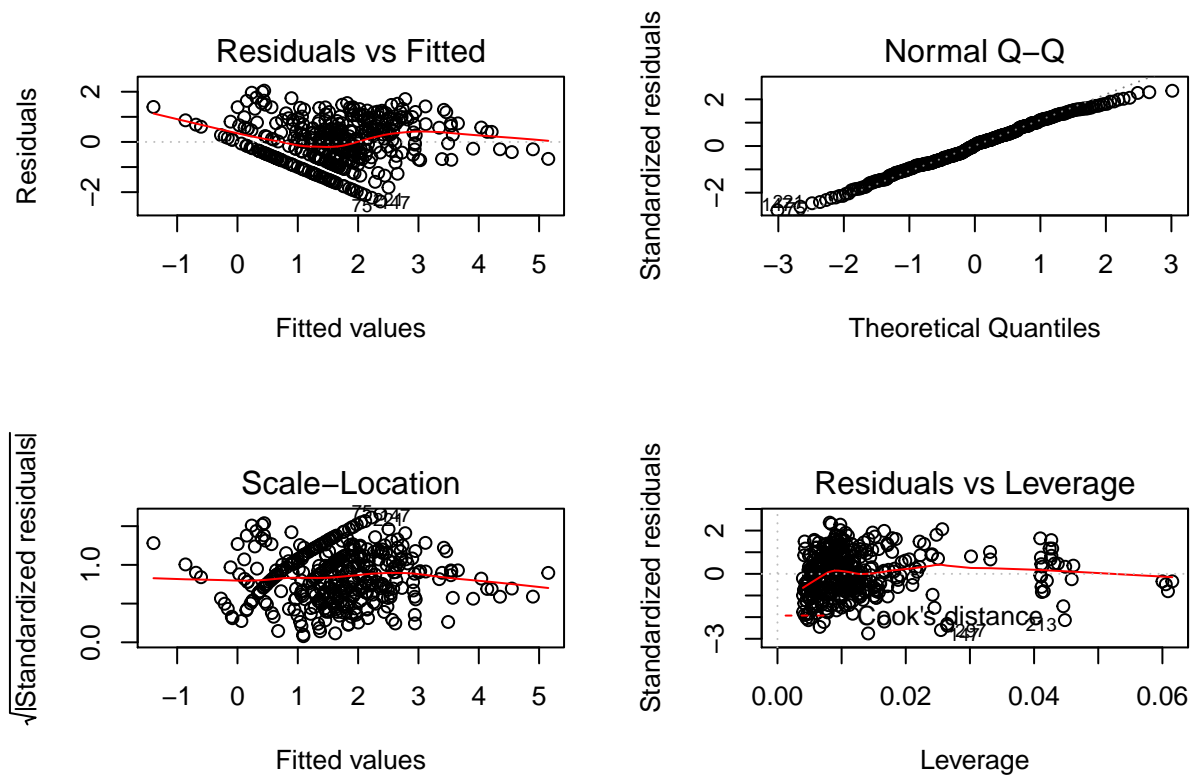
Figure 4: Exploratory Linear Model Plots

Table 6: Linear vs. GLM Model Comparison (Absolute Scale)

	AIC	MSE	MSE (out of sample)
Linear	2008.82	442.44	279.47
Negative Binomial	2319.23	480.74	675.54
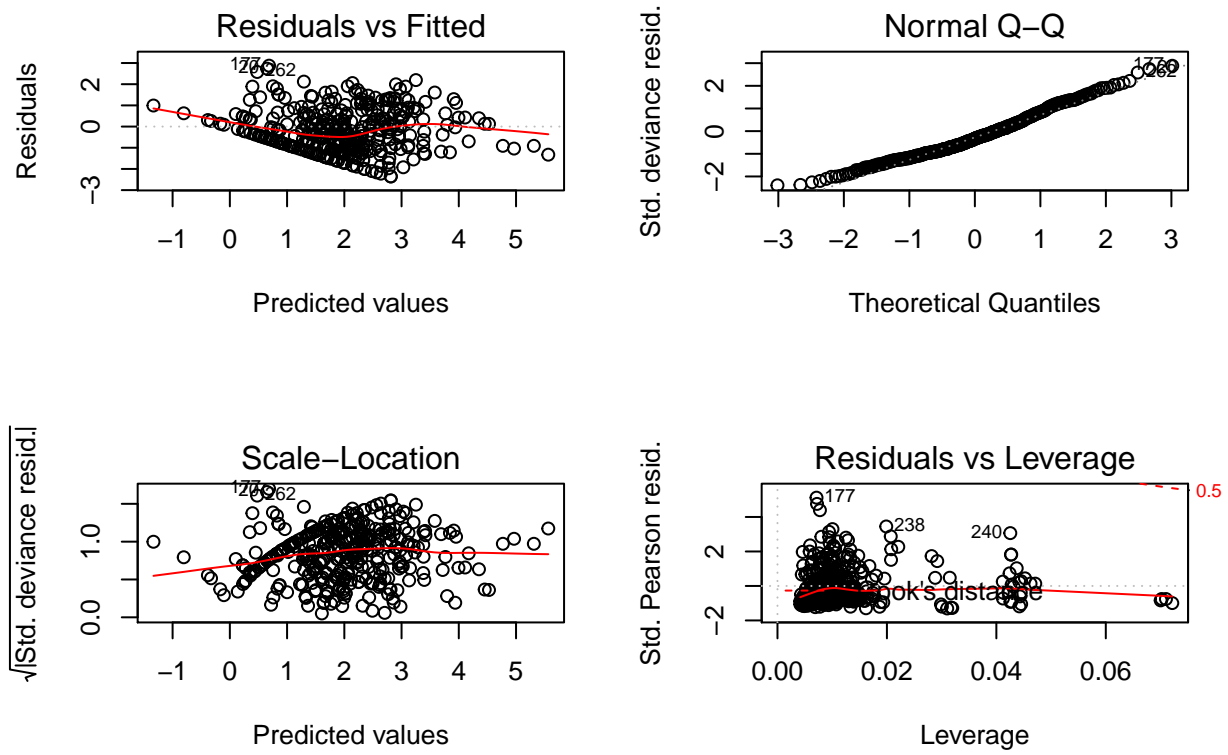
Figure 5: Exploratory Negative Binomial Model Plots

Table 7: Coefficient Comparison

	Linear	Robust (Bisquare)	Poisson	Neg Binom	Neg Binom (fixed effects
Intercept	-9.63 (0.638)	-10.04 (0.67)	-11.23 (0.254)	-10.17 (0.647)	-10.57 (0.641)
log(GDP/Cap)	0.408 (0.032)	0.412 (0.034)	0.521 (0.014)	0.402 (0.029)	0.452 (0.033)
log(Pop)	0.443 (0.029)	0.464 (0.031)	0.505 (0.010)	0.498 (0.032)	0.506 (0.028)
Host	0.867 (0.190)	0.829 (0.199)	0.311 (0.042)	0.693 (0.159)	0.608 (0.153)
Soviet/Comm	1.033 (0.105)	1.04 (0.110)	1.02 (0.037)	1.03 (0.098)	1.08 (0.097)
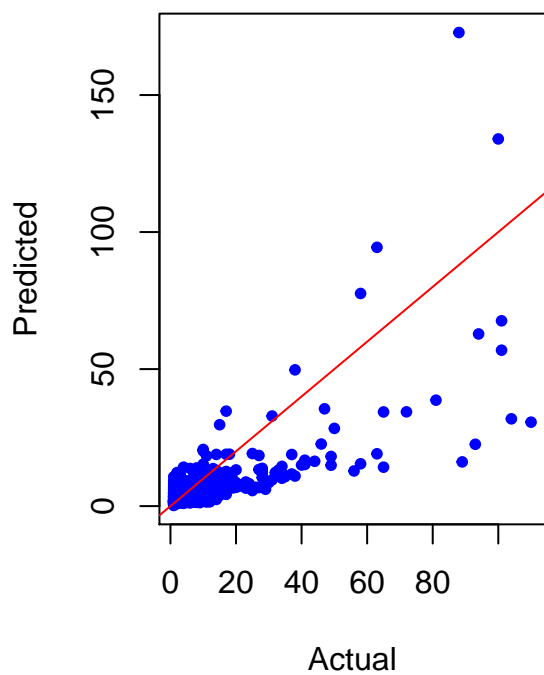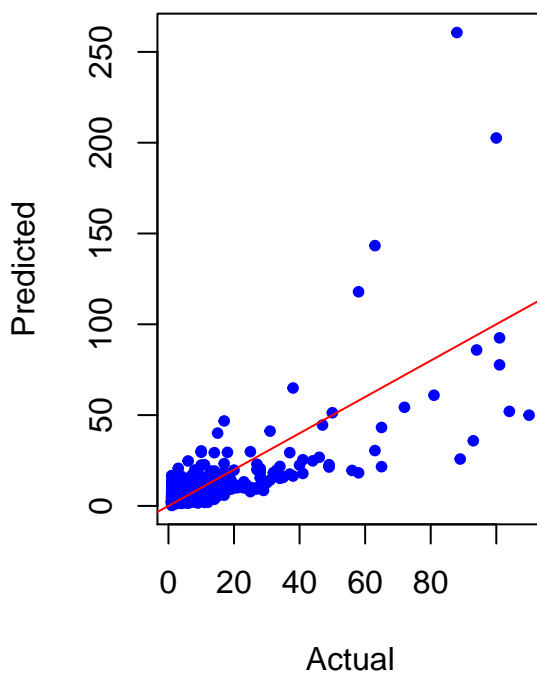
Figure 6: Comparison of Predicted vs. Actual
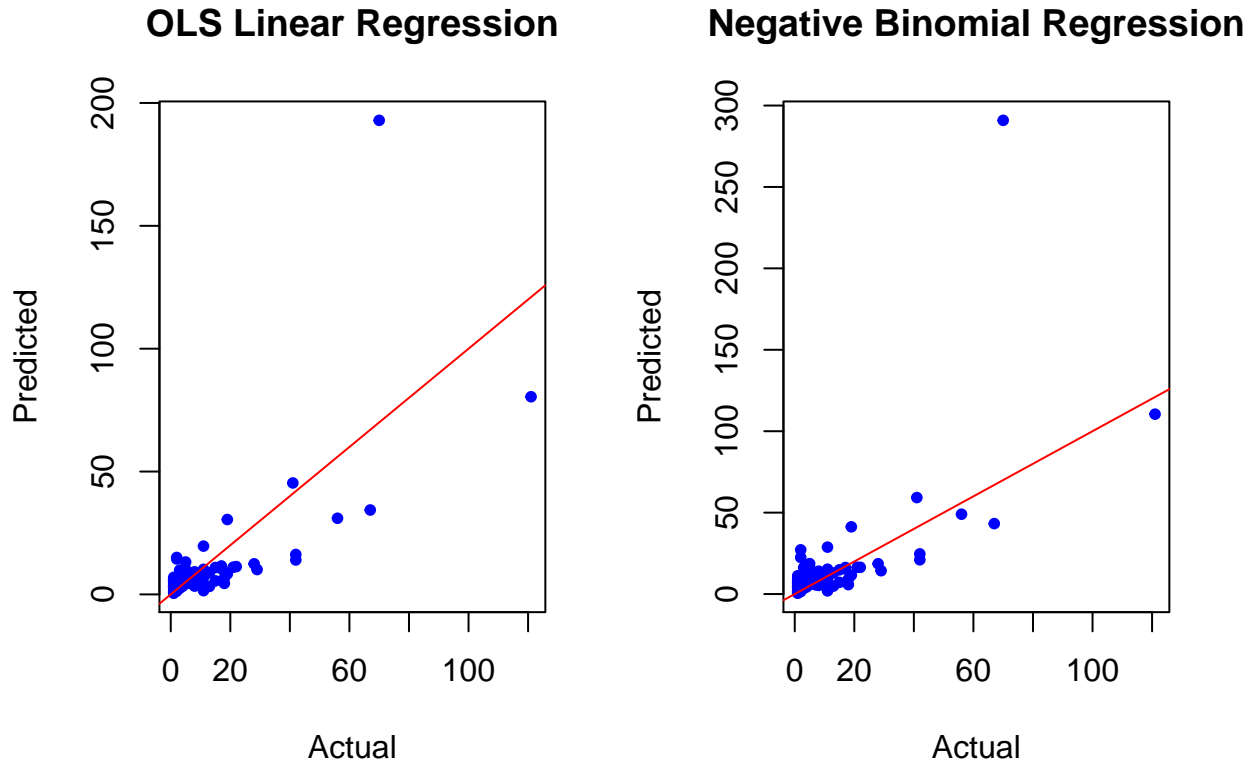
Figure 7: Out of Sample Comparison of Predicted vs. Actual

8 Appendix B: Influential Cases

	country	count	year	gdp	pop	host	comm_soviet
2	China	100	2008	4598206091384	1324655000	1	1
4	United Kingdom	47	2008	2890564338235	61806995	1	0
5	Australia	46	2008	1052584601611	21249200	1	0
17	Brazil	15	2008	1695824571927	192979029	1	0
20	Jamaica	11	2008	13678606692	2790122	0	0
48	Greece	4	2008	354460802549	11077841	1	0
75	Vietnam	1	2008	99130304099	86707801	0	1
86	United States	101	2004	12274928000000	292805298	1	0
88	China	63	2004	1955347004963	1296075000	1	1
90	Australia	49	2004	611904253806	20127400	1	0
94	United Kingdom	31	2004	2398555474185	59987905	1	0

	country	count	year	gdp	pop	host	comm_soviet
102	Greece	16	2004	240521260988	10955141	1	0
147	India	1	2004	699688852930	1126135777	0	0
238	China	88	2012	8560547314679	1350695000	1	1
240	United Kingdom	65	2012	2662085168499	63700300	1	0
242	Japan	38	2012	6203213121334	127629000	1	0
252	Brazil	17	2012	2465188674415	200560983	1	0
258	Jamaica	12	2012	14800165407	2840992	0	0
294	Greece	2	2012	245670666639	11045011	1	0
302	Saudi Arabia	1	2012	735974843360	29086357	0	0
303	Venezuela	1	2012	381286237848	29893080	0	0
318	United States	94	2000	10284779000000	282162411	1	0
320	China	58	2000	1211346869605	1262645000	1	1
321	Australia	58	2000	415034227218	19153000	1	0
336	Greece	13	2000	130133845771	10805808	1	0
340	Spain	11	2000	595402616547	40567864	1	0
343	Jamaica	9	2000	8985352832	2656864	0	0
378	India	1	2000	462146799338	1053050912	0	0
395	United States	101	1996	8100201000000	269394000	1	0
398	China	50	1996	863746717504	1217550000	0	1
399	Australia	41	1996	400302731411	18311000	1	0
402	South Korea	27	1996	598099073901	45524681	1	0
409	Spain	17	1996	640998292395	39889852	1	0
420	Greece	8	1996	145861612826	10608800	1	0
452	India	1	1996	387656017799	978893217	0	0
453	Mexico	1	1996	410975595310	95687452	0	0
468	Tonga	1	1996	219583570	96369	0	0

	dfb.1_	dfb.lg_p	dfb.l____	dfb.host	dfb.cmm_	dffit	cov.r	cook.d	hat
2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
17	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
20	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
48	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
75	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
86	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
88	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
90	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
94	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
102	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
147	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
157	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
159	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
161	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
171	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
177	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
213	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
221	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
222	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
237	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
239	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
240	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
255	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
259	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
262	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
297	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
314	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
317	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
318	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE

	dfb.1_	dfb.lg_p	dfb.l____	dfb.host	dfb.cmm_	dffit	cov.r	cook.d	hat
321	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
328	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
339	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
371	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
372	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
387	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

9 Appendix C

```r
knitr::opts_chunk$set(echo = FALSE, comment = NA)
base <-read.csv('data/base_data.csv', stringsAsFactors = F)
P__disp <- function(x) {
  pr <- sum(residuals(x, type="pearson")^2)
  dispersion <- pr/x$df.residual
  c(pr, dispersion)
}
library(dplyr)
library(qpcR)
library(MuMIn)
base.total = base[which(base$year!=2016),]
Olympic = base.total[,c(2,3,4,5,6,7,8)]
#Clean the data (Mary to add)
#Make new dataframe with GDP / capita
Olympic_v2 <- data.frame(year = Olympic$year, country = Olympic$country, count = Olympic$count, log_pop
library(knitr)
summary(base$count)
hist(base$count)
hist(base[which(base$year == c("2008")),c("count")])
summary(base$gdp)
summary(base$pop)
```

```r
sum(base$comm_soviet)

nrow(base)-sum(base$comm_soviet)

sum(base$host)

par(mfrow=c(1,2))

#Histogram Population

hist(base$pop,main = "Untransformed",xlab = "Population")

#Histogram log(Population)

hist(log(base$pop),main = "Transformed",xlab = "log(Population)")

#Log(GDP)

boxplot(log(base$gdp),ylab = "log(GDP)")

#log(Population)

boxplot(log(base$pop),ylab = "log(Population)")

pairs(Olympic[,c(2,4,5,6,7)])

kable(cor(Olympic[,c(2,4,5,6,7)]))

library(leaps)

#CP

olympic.leapCP <- leaps(y=log(Olympic_v2$count), x=Olympic_v2[,4:7])

#R2a

olympic.leapR2a <- leaps(y=log(Olympic_v2$count), x=Olympic_v2[,4:7], method = 'adjr2')

  ### Code for AIC

xList <- names(Olympic_v2)[4:7]

  #### Remove the last row that has all False's

vec <- olympic.leapCP$which

  ### Name the columns in the grid

names(vec) <- paste("X", 1:4, sep="")

  #### Build matrix of formula for every row

allModelsList <- apply(vec, 1, function(x) as.formula(

  paste(c("count ~ 1", xList[x]), collapse = "+")))

  ### Calculate the coefficients for all 16 models

allModelsResults.lm <- lapply(allModelsList,

                        function(x) lm(x, data=Olympic_v2))

olympic.lmfinal <- lm(log_count ~ log_pop + log_gdp_per_cap + host + comm_soviet, data = Olympic_v2)
```

```r
par(mfrow=c(2,2))

plot(olympic.lmfinal)

library(lmtest)

bptest(log_count ~ log_pop + log_gdp_per_cap + host + comm_soviet,data = Olympic_v2,studentize = FALSE)

library(nortest)

lillie.test(olympic.lmfinal$residuals)

Olympic.pois<-glm(count~log_pop + log_gdp_per_cap + host + comm_soviet, data = Olympic_v2, family = pois

P__disp(Olympic.pois)

library(MASS)

olympic.nb <- glm.nb(count~log_pop + log_gdp_per_cap + host + comm_soviet, data = Olympic_v2)

library(leaps)

olympic.nb_leap <- leaps(y=Olympic_v2$count, x=Olympic_v2[,4:7])

Cp.nb<-round(olympic.nb_leap$Cp, 2)

which<-olympic.nb_leap$which

rownames(which) <-NULL

colnames(which)<-c('Pop', 'GDP/C', 'Host', 'Soviet')

xList <- names(Olympic_v2)[4:7]

vec <- olympic.nb_leap$which

#Name the columns in the grid

names(vec) <- paste("X", 1:4, sep="")

#Build matrix of formula for every row

allModelsList <- apply(vec, 1, function(x) as.formula(
  paste(c("count ~ 1", xList[x]), collapse = "+")))

#Calculate the coefficients for all 16 models

allModelsResults <- lapply(allModelsList,
                           function(x) glm.nb(x, data=Olympic_v2))

AIC.nb<-matrix(unlist(lapply(allModelsResults, function(x) round(extractAIC(x),2))), ncol = 2, byrow =

library(knitr)

#2016 Data

base.2016 = base[which(base$year==2016),]

attach(base.2016)

base.2016 = data.frame(country,count,year,log_pop=log(pop),log_gdp_per_cap = log(gdp/pop),host,comm_sov
```

```r
#Linear Fitted
fitted.lm = -9.63058+0.40830*base.2016$log_gdp_per_cap+0.44275*base.2016$log_pop+0.86726*base.2016$host-
#olympic.lmfinal




#Prediction intervals: Linear Model
nrow(base.2016)-5
t=qt(1-0.1/2,76)
mse.lm = 0.751
X = as.matrix(cbind(rep(1,nrow(Olympic_v2)),Olympic_v2$log_gdp_per_cap,Olympic_v2$log_pop,Olympic_v2$hos

spred = c()
for (i in 1:nrow(base.2016)){
  xh = as.matrix(cbind(1,base.2016$log_gdp_per_cap[i],base.2016$log_pop[i],base.2016$host[i],base.2016$
  s2 = mse.lm*(1+xh%*%solve(t(X)%*%X)%*%t(xh))
  spred = c(spred,sqrt(s2))
}


lower.pred = fitted.lm - qt(1-0.10/2,76)*spred
upper.pred = fitted.lm + qt(1-0.10/2,76)*spred


forecast.lm = cbind(log(base.2016$count),lower.pred,upper.pred)
ininterval.lm = cbind(forecast.lm[,2]<=forecast.lm[,1] & forecast.lm[,3]>=forecast.lm[,1])
length(which(ininterval.lm))


#NB Method 2 (works!)
library(tidyverse)
library(ciTools)
library(MASS)
```

```r
set.seed(20181215)


base.2016 = base[which(base$year==2016),]

attach(base.2016)

base.2016 = data.frame(country,count,year,log_pop=log(pop),log_gdp_per_cap = log(gdp/pop),host,comm_sov


newData <- data.frame(base.2016[,c(2,4,5,6,7)])

train_data <- Olympic_v2[,3:7]


#Generate prediction intervals

olympic.nb = glm.nb(count ~ log_pop + log_gdp_per_cap + host + comm_soviet,data=train_data)


#add_pi comes from the library ciTools

olympic.nb_pint <- add_pi(tb=newData,fit=olympic.nb, names=c("lpb", "upb"), alpha=0.1, nSims=20000)



olympic.nb_pint %>%

  mutate(inside=(count >lpb & count<upb))->preds.nb


summary(preds.nb)



####finding MSEs

preds<-data.frame(matrix(cbind(base.2016$count, preds.nb$pred, exp(fitted.lm)), ncol = 3))


colnames(preds)<-c('Actual', 'Prediction.nb', 'Prediction.lm')


mse.nb<-mean((preds$Actual-preds$Prediction.nb)^2)

mse.lm<-mean((preds$Actual-preds$Prediction.lm)^2)

hist(Olympic_v2$count, lwd = 4, xlab = '', main = '')

par(mfrow=c(1,2))

#Histogram Population
```

```r
hist(base$pop,main = "Untransformed",xlab = "Population")
#Histogram log(Population)
hist(log(base$pop),main = "Transformed",xlab = "log(Population)")
options(scipen = 999)


kable(rbind('Count'=summary(Olympic_v2$count), 'Population'=summary(exp(Olympic_v2$log_pop)), 'GDP/Cap'=
this<-rbind('Host' = unname(table(Olympic_v2$host)),
'Communist/Soviet' = unname(table(Olympic_v2$comm_soviet)))


colnames(this)<-c('Off', 'On')


kable(this, caption = 'Summary of Categorical Variables')
plot(Olympic_v2[,c(3,4,5,6,7)])
kable(cor(Olympic_v2[,c(3,4,5,6,7)]), caption = 'Correlation Matrix')
#PRESS (Non-Mac)
library(qpcR)
olympic.lm = PRESS(lm(log_count~log_pop+log_gdp_per_cap+host+comm_soviet, data = Olympic_v2))
olympic.lmX1 = PRESS(lm(log_count~log_pop, data = Olympic_v2))
olympic.lmX2 = PRESS(lm(log_count~log_gdp_per_cap, data = Olympic_v2))
olympic.lmX3 = PRESS(lm(log_count~host, data = Olympic_v2))
olympic.lmX4 = PRESS(lm(log_count~comm_soviet, data = Olympic_v2))
olympic.lmX1X2 = PRESS(lm(log_count~log_pop+log_gdp_per_cap, data = Olympic_v2))
olympic.lmX1X3 = PRESS(lm(log_count~log_gdp_per_cap+host, data = Olympic_v2))
olympic.lmX1X4 = PRESS(lm(log_count~log_pop+comm_soviet, data = Olympic_v2))
olympic.lmX2X3 = PRESS(lm(log_count~log_gdp_per_cap+host, data = Olympic_v2))
olympic.lmX2X4 = PRESS(lm(log_count~log_gdp_per_cap+comm_soviet, data = Olympic_v2))
olympic.lmX3X4 = PRESS(lm(log_count~host+comm_soviet, data = Olympic_v2))
olympic.lmX1X2X3 = PRESS(lm(log_count~log_pop+log_gdp_per_cap+host, data = Olympic_v2))
olympic.lmX1X2X4 = PRESS(lm(log_count~log_pop+log_gdp_per_cap+comm_soviet, data = Olympic_v2))
olympic.lmX2X3X4 = PRESS(lm(log_count~log_gdp_per_cap+host+comm_soviet, data = Olympic_v2))
olympic.lmX1X3X4 = PRESS(lm(log_count~log_pop+host+comm_soviet, data = Olympic_v2))
olympic.lm_press <- rbind(olympic.lmX1$stat,
```

```
                             olympic.lmX3$stat,

                             olympic.lmX2$stat,

                             olympic.lmX4$stat,

                             olympic.lmX1X3$stat,

                             olympic.lmX1X2$stat,

                             olympic.lmX1X4$stat,

                             olympic.lmX2X3$stat,

                             olympic.lmX3X4$stat,

                             olympic.lmX2X4$stat,

                             olympic.lmX1X2X3$stat,

                             olympic.lmX1X2X4$stat,

                             olympic.lmX1X3X4$stat,

                             olympic.lmX2X3X4$stat,

                             olympic.lm$stat)
#Summary
Diagnostics = cbind(olympic.leapCP$which, Cp=round(olympic.leapCP$Cp,2), aR2=round(olympic.leapR2a$adjr2
      AIC=matrix(unlist(lapply(allModelsResults.lm, function(x) round(extractAIC(x),2))), ncol=2, byrow=
#PRESS wasn't showing as column name
colnames(Diagnostics) = c("1","2","3","4","Cp","aR2","AIC","PRESS")
par(mfrow=c(2,2))
plot(olympic.lmfinal)
par(mfrow=c(2,2))
plot(olympic.nb)
kable(Diagnostics, caption = 'Model Selection for Linear Model')
kable(cbind(which, Parameters = olympic.nb_leap$size, Cp.nb, AIC.nb), caption = 'Model Selection Diagnos
knitr::opts_chunk$set(fig.pos = 'H')
par(mfrow=c(1,2))
plot(x=Olympic_v2$count, y = exp(olympic.lmfinal$fitted.values), col='blue', pch=20, xlab='Actual',
     ylab='Predicted', main='OLS Linear Regression')
abline(a=0,b=1, col='red')
plot(x=Olympic_v2$count, y = olympic.nb$fitted.values, col='blue', pch=20, xlab='Actual',
     ylab='Predicted', main='Negative Binomial Regression')
```

```r
abline(a=0,b=1, col='red')
par(mfrow=c(1,2))
plot(x=base.2016$count, y = exp(fitted.lm), col='blue', pch=20, xlab='Actual',
     ylab='Predicted', main='OLS Linear Regression')
abline(a=0,b=1, col='red')
plot(x=base.2016$count, y = preds.nb$pred, col='blue', pch=20, xlab='Actual',
     ylab='Predicted', main='Negative Binomial Regression')
abline(a=0,b=1, col='red')
df_20<-data.frame(matrix(c(2008.82, 442.44, 279.47, 2319.23,  480.74, 675.54), nrow = 2, byrow = T))
colnames(df_20)<-c('AIC', 'MSE', 'MSE (out of sample)')
rownames(df_20)<-c('Linear', 'Negative Binomial')
kable(df_20, caption = 'Linear vs. GLM Model Comparison (Absolute Scale)')
df_133<-data.frame(matrix(c(
'-9.63 (0.638)', '-10.04 (0.67)','-11.23 (0.254)','-10.17 (0.647)','-10.57 (0.641)',
'0.408 (0.032)','0.412 (0.034)','0.521 (0.014)','0.402 (0.029)','0.452 (0.033)',
'0.443 (0.029)','0.464 (0.031)','0.505 (0.010)','0.498 (0.032)','0.506 (0.028)',
'0.867 (0.190)','0.829 (0.199)','0.311 (0.042)','0.693 (0.159)','0.608 (0.153)',
'1.033 (0.105)','1.04 (0.110)','1.02 (0.037)','1.03 (0.098)','1.08 (0.097)'
), ncol = 5, byrow = T))
colnames(df_133)<-c(
'Linear',
'Robust (Bisquare)',
'Poisson',
'Neg Binom',
'Neg Binom (fixed effects')
rownames(df_133)<-c('Intercept','log(GDP/Cap)','log(Pop)','Host','Soviet/Comm')
kable(df_133, caption = 'Coefficient Comparison')
#Influential cases
olympic.lm_inf=influence.measures(olympic.lmfinal)$is.inf
idx=which(apply(olympic.lm_inf,1,any))
#Influential Cases
kable(Olympic[idx,])
```

```r
#Inluential Cases by Test
kable(olympic.lm_inf[idx,])


kable(cbind('Country'=as.character(base.2016$country), 'Count' = preds.nb$count, 'NB Predictions' = rou
```

10   Appendix D

Table 10: 2016 Out-of-sample Predicted Values

Country	Count	NB Predictions	NB Interval	Linear Predictions	Linear Intervals
United States	121	110.48	(19, 269)	80.44	(18, 352)
China	70	290.9	(53, 711)	192.93	(44, 853)
United Kingdom	67	43.25	(7, 107)	34.35	(8, 150)
Russia	56	49.03	(8, 117)	30.99	(7, 133)
Germany	42	24.67	(4, 60)	16.25	(4, 69)
France	42	21.05	(3, 51)	14.02	(3, 60)
Japan	41	59.25	(10, 143)	45.34	(10, 198)
Australia	29	14.33	(2, 35)	10.12	(2, 43)
Italy	28	18.62	(3, 45)	12.45	(3, 53)
Canada	22	16.41	(2, 40)	11.31	(3, 48)
South Korea	21	16.41	(2, 40)	11.07	(3, 47)
Netherlands	19	11.6	(1, 29)	8.35	(2, 36)
Brazil	19	41.29	(7, 101)	30.47	(7, 133)
New Zealand	18	5.81	(0, 15)	4.49	(1, 19)
Azerbaijan	18	9.17	(1, 23)	6.7	(2, 29)
Spain	17	15.4	(2, 37)	10.45	(2, 44)
Kazakhstan	17	16.31	(2, 40)	11.57	(3, 49)
Hungary	15	14.87	(2, 36)	10.94	(3, 47)
Denmark	15	7.19	(1, 18)	5.5	(1, 23)
Kenya	13	4.89	(0, 13)	3.25	(1, 14)
Uzbekistan	13	12.93	(2, 32)	8.81	(2, 38)
Poland	11	28.81	(5, 70)	19.65	(5, 84)
Sweden	11	9.33	(1, 23)	6.92	(2, 29)
Ukraine	11	15.26	(2, 38)	10.2	(2, 44)
Jamaica	11	1.95	(0, 6)	1.53	(0, 6)
Czech Republic	10	6.36	(0, 16)	4.67	(1, 20)
South Africa	10	8.82	(1, 22)	5.86	(1, 25)
Croatia	10	9.55	(1, 24)	7.37	(2, 31)

Country	Count	NB Predictions	NB Interval	Linear Predictions	Linear Intervals
Belarus	9	10.03	(1, 25)	7.35	(2, 31)
Iran	8	10.5	(1, 26)	6.84	(2, 29)
Colombia	8	8.51	(1, 21)	5.7	(1, 24)
Ethiopia	8	5.32	(0, 14)	3.38	(1, 14)
Turkey	8	14.04	(2, 34)	9.19	(2, 39)
Serbia	8	8.93	(1, 22)	6.65	(2, 28)
Switzerland	7	10.2	(1, 26)	7.66	(2, 33)
Georgia	7	5.66	(0, 15)	4.36	(1, 19)
Thailand	6	10.28	(1, 25)	6.76	(2, 29)
Greece	6	6.33	(0, 16)	4.65	(1, 20)
Belgium	6	9.09	(1, 23)	6.69	(2, 28)
Mexico	5	16.05	(2, 39)	10.22	(2, 44)
Malaysia	5	8.34	(1, 21)	5.75	(1, 24)
Romania	5	18.68	(3, 46)	13.19	(3, 56)
Norway	4	7.69	(1, 20)	5.93	(1, 25)
Slovenia	4	8.44	(1, 21)	6.79	(2, 29)
Slovakia	4	4.36	(0, 12)	3.32	(1, 14)
Argentina	4	11.09	(1, 27)	7.51	(2, 32)
Lithuania	4	8.56	(1, 22)	6.75	(2, 29)
Armenia	4	4.88	(0, 13)	3.81	(1, 16)
Indonesia	3	16.23	(2, 40)	9.88	(2, 42)
Egypt	3	9.74	(1, 24)	6.27	(1, 27)
Tunisia	3	3.45	(0, 9)	2.5	(1, 11)
Bulgaria	3	10.2	(1, 26)	7.61	(2, 32)
India	2	27.15	(4, 65)	15.03	(3, 65)
Algeria	2	6.66	(1, 17)	4.5	(1, 19)
Mongolia	2	5.01	(0, 13)	3.91	(1, 17)
Vietnam	2	22.51	(3, 54)	14.44	(3, 62)
Ireland	2	7.04	(1, 18)	5.45	(1, 23)
Israel	2	7.58	(1, 19)	5.66	(1, 24)
Kuwait	2	4.62	(0, 12)	3.59	(1, 15)

Country	Count	NB Predictions	NB Interval	Linear Predictions	Linear Intervals
Bahamas	2	1.5	(0, 5)	1.33	(0, 6)
Bahrain	2	2.54	(0, 7)	2.09	(0, 9)
Austria	1	8.25	(1, 21)	6.16	(1, 26)
Finland	1	6.47	(0, 17)	4.96	(1, 21)
Niger	1	1.84	(0, 5)	1.27	(0, 5)
Nigeria	1	11.23	(1, 28)	6.94	(2, 30)
Morocco	1	5.53	(0, 14)	3.76	(1, 16)
Philippines	1	9.47	(1, 24)	6.06	(1, 26)
Tajikistan	1	4.58	(0, 12)	3.34	(1, 14)
Jordan	1	3.28	(0, 9)	2.4	(1, 10)
Portugal	1	6.47	(0, 17)	4.76	(1, 20)
United Arab Emirates	1	8	(1, 20)	5.95	(1, 25)
Dominican Republic	1	4.27	(0, 11)	3.12	(1, 13)
Estonia	1	6.22	(0, 16)	5.13	(1, 22)
Moldova	1	4.17	(0, 11)	3.21	(1, 14)
Burundi	1	1.18	(0, 4)	0.85	(0, 4)
Fiji	1	1.12	(0, 4)	0.93	(0, 4)
Qatar	1	5.02	(0, 13)	4.02	(1, 17)
Kosovo	1	3.89	(0, 10)	3.12	(1, 13)
Trinidad and Tobago	1	2.18	(0, 6)	1.8	(0, 8)
Singapore	1	7.2	(1, 18)	5.52	(1, 24)
Grenada	1	0.5	(0, 2)	0.47	(0, 2)