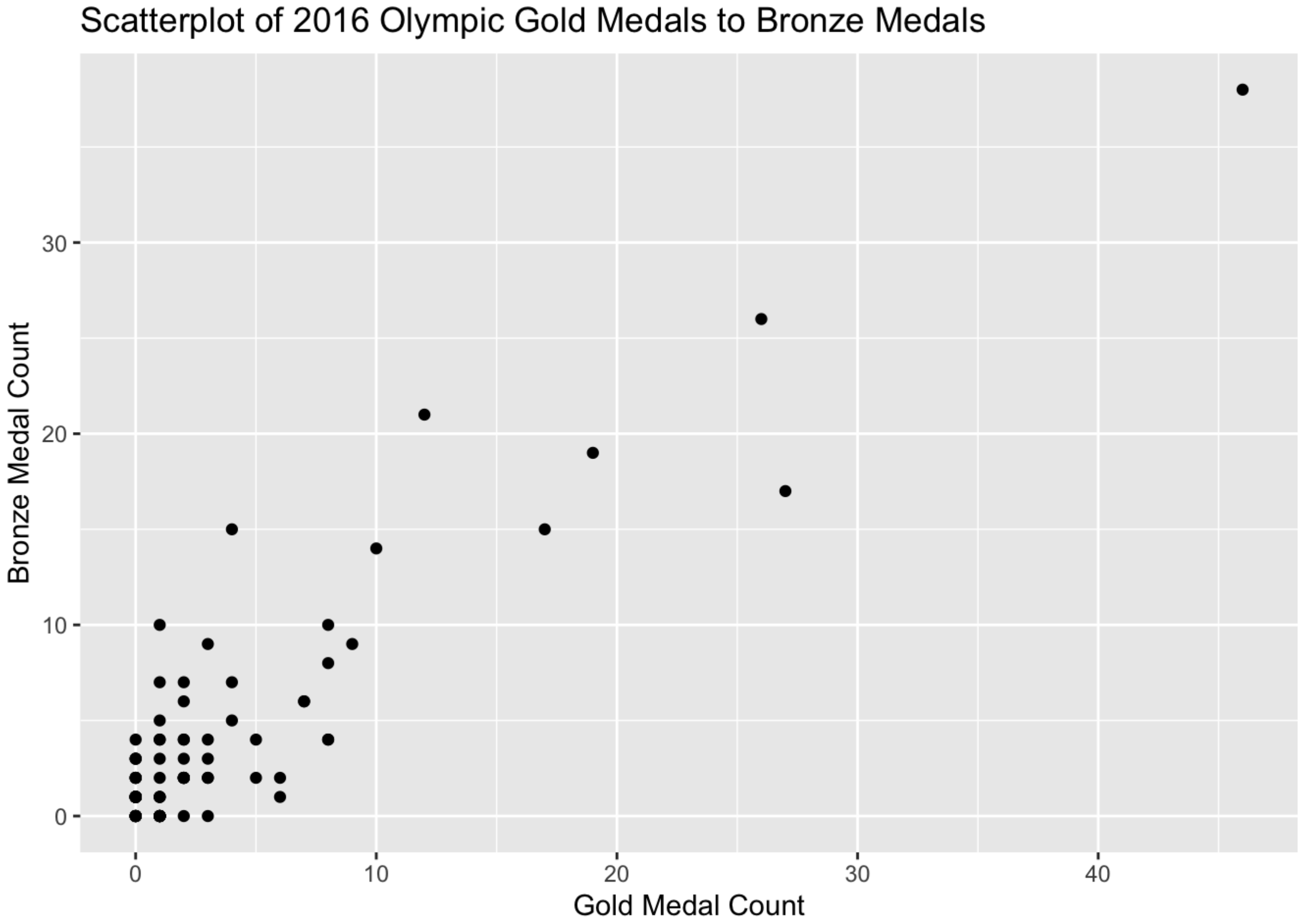
Modeling 2016 Rio Summer Olympic Medal Count (By Country) Data

Introduction

This datset is one of the two that I merged in the last project. It comes from a website called “insidethegames” and includes the gold, silver, and bronze medal counts for each country in the 2016 Rio Summer Olympic games. Additionally, I used a map to record whether the country was located in the northern hemisphere, the southern hemisphere, or along the equator. There are 88 observations for the 88 countries that competed in the summer games in 2016, and the data did not have to be tidied (since I made it). I have always took an interest in the olympics with my grandfather’s history in the olympics. I do not expect to find anything in particular.

EDA

```
# First, I need to import the dataset with 'readxl' and save the dataset
library(readxl)
rio16 <- read_xlsx("2016olympic.xlsx")
# Now I will make a scatterplot with "ggplot2" to see the relationship betweeen the numb
er of gold medals won (if the country won any) and the corresponding number of bronze me
dals.
library(ggplot2)
ggplot(rio16, aes(x = gold.2016, y = bronze.2016)) +
  geom_point() +
  ggtitle("Scatterplot of 2016 Olympic Gold Medals to Bronze Medals") +
  labs(x = "Gold Medal Count", y = "Bronze Medal Count")
```



It looks as though the more gold medals a country won in the olympics, the more bronze medals they won as well. To further explore this, I will make a correlation matrix of the 2016 medal count data.

```
# First, I will use dplyr functions to remove my categorical variables for the correlati
on matrix with "select".
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

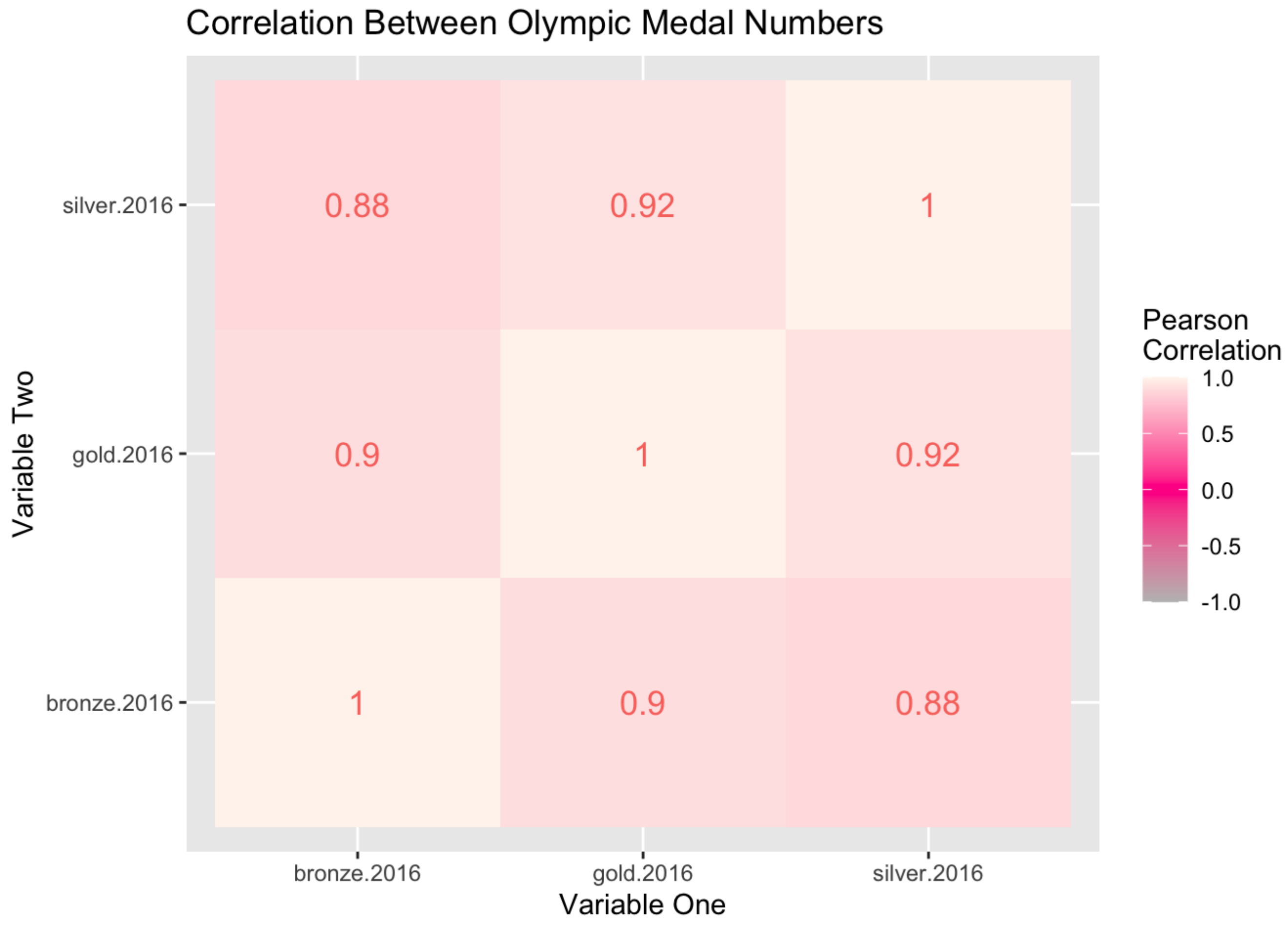
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
## ✓ tibble  3.0.5     ✓ purrr   0.3.3
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
newrio16 <- rio16 %>% select(-country, -ns.hemisphere)
# Then, I will use 'cor' to make a correlation matrix with pairwise observations in orde
r to find the correlation between any two of my numeric variables.
cor(newrio16, use = "pairwise.complete.obs") %>%
# Now, I will save the matrix as a datarame with 'as.data.frame'.
as.data.frame() %>%
# Now, I will convert my row names to explicit variables.
rownames_to_column %>%
# Using 'pivot_longer', I will make every correlation appear in the same column.
pivot_longer(-1, names_to = "other_var", values_to = "correlation") %>%
# Now, I will make the correlation heatmap using 'ggplot' and 'geom_tile'. My x and y ax
es are going to be the two variables, and the fill will be their correlation values. Thi
s will be filled within 'aes'
ggplot(aes(rowname, other_var, fill = correlation)) +
geom_tile() +
# To make the graph pretty, I will use 'scale_fill_gradient2', set the midpoint to zero,
the limits to one and negative one, and rename the color key. Additionally, I used 'ggti
tle' and 'labs' to add a title to the graph and rename the axes, respectively.
scale_fill_gradient2(low = "gray", high = "seashell1", mid = "deeppink", midpoint = 0, l
imit = c(-1,1), space = "Lab", name = "Pearson\nCorrelation") +
ggtitle("Correlation Between Olympic Medal Numbers") +
labs(x = "Variable One", y = "Variable Two") +
# Additionally, I added the correlation values to the heatmap with 'geom_text', rounded
 the values to two decimal places with 'round', colored the text, set the font size, and
removed the legend for the text.
geom_text(aes(label = round(correlation, 2), color = "roseybrown4", size = 4), show.lege
nd = FALSE)
```



One finding that was similar to the previous project is the particularly strong, positive correlations between the number of different medals won in 2016.

```
# Let's look at the average number of gold medals won in the 2016 olympics total, and th
en look at the average gold medals by hemisphere. First, I will use dplyr functions "sel
ect" and "summarize" to select gold medals and get the mean of them.
rio16 %>% select(gold.2016) %>% summarize(mean.gold = mean(gold.2016))
```

```
## # A tibble: 1 x 1
##   mean.gold
##       <dbl>
## 1      3.52
```

The average gold medals won in the 2016 olympics per country is about 3-4 medals. Now, let’s group by hemisphere.

```
# I will use the same code from the total mean except add 'group_by" to group by hemisph
ere.
rio16 %>% select(gold.2016, ns.hemisphere) %>% group_by(ns.hemisphere) %>% summarize(mea
n.gold = mean(gold.2016))
```

```
## # A tibble: 3 x 2
##   ns.hemisphere mean.gold
## * <chr>             <dbl>
## 1 both               3.67
## 2 northern           3.55
## 3 southern           3
```

It looks like the average number of gold medals for each hemisphere was between 3-4 gold medals.

MANOVA

```
# Now, I will perform a MANOVA to see if any of the different medal counts show a mean difference across the different hemispheres. I will use 'manova' and 'cbind' to run the MANOVA.
manova_rio <- manova(cbind(gold.2016, silver.2016, bronze.2016) ~ ns.hemisphere, data = rio16)

# Now, we can look at the output of the manova.
summary(manova_rio)
```

```
##               Df   Pillai approx F num Df den Df Pr(>F)
## ns.hemisphere  2 0.065901   0.9427      6    166 0.4661
## Residuals     84
```

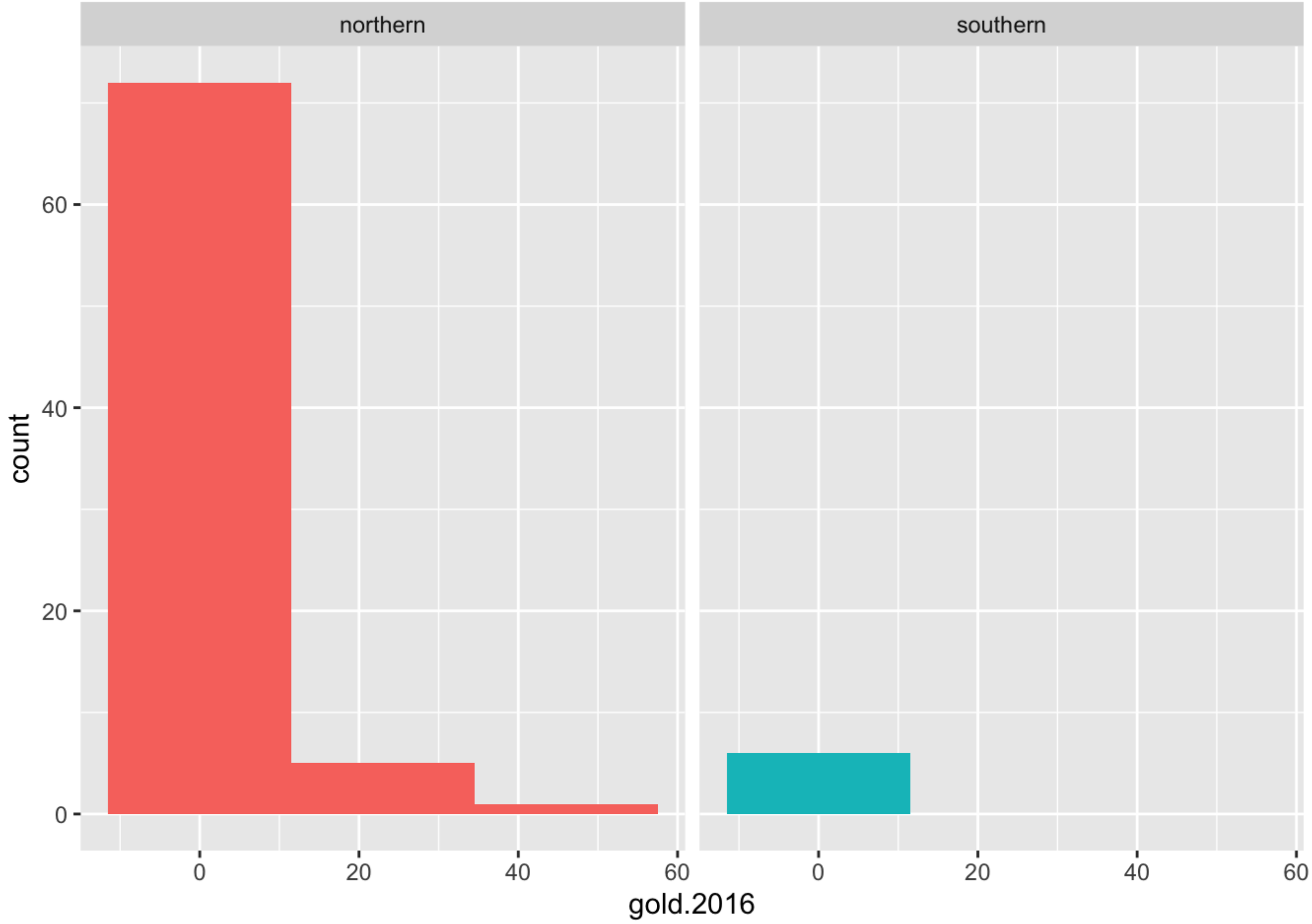
```
summary.aov(manova_rio)
```

```
##  Response gold.2016 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## ns.hemisphere  2    1.8   0.881  0.0182 0.9819
## Residuals     84 4060.0  48.333
##
##  Response silver.2016 :
##               Df  Sum Sq Mean Sq F value Pr(>F)
## ns.hemisphere  2    8.38   4.192   0.124 0.8835
## Residuals     84 2839.29  33.801
##
##  Response bronze.2016 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## ns.hemisphere  2   15.4   7.715  0.1927 0.8251
## Residuals     84 3362.2  40.026
```

It does not look like any of these medal counts show a significant mean difference across the different hemispheres given the p-value is well over 0.05.In other words, it appears that the location of the country in terms of hemisphere does not significantly affect the average number of gold, silver, and bronze medals won in the 2016 olympics. Manova has several assumptions, including one for a random sample and independent observations, multivariate normality of the response variables (which is likely not met), having no extreme uni/multivariate outliers (which is likely not met), linear relationships with the response variables without multicolinearity, and homogenieity of within-groups covariant matricies. Overall, MANOVA was the only test that needed to be completed for this step of the project since there were no significant findings.

Randomization Testing

```
# We will now see whether the average number of gold medals won in 2016 varies based on
 which hemisphere it came from (but only northern and southern). The null hypothesis is
 that the average number of gold medals in 2016 does not vary based on which hemisphere
 the country came from, and the alternative hypothesis is that the average number of gol
d medals does vary based on which hemisphere it came from.
# I will remove the countries from both hemispheres using 'select'
rio2 <- rio16 %>% filter(!ns.hemisphere == "both")
# First, we will look at the distribution of the number of gold medals for each hemisphe
re using 'ggplot'
ggplot(rio2, aes(gold.2016,fill=ns.hemisphere)) +
  geom_histogram(bins=3) +
  facet_wrap(~ns.hemisphere,ncol=3) +
  theme(legend.position="none")
```



It looks like there are several more gold medals won by northern hemisphere countries than anywhere else. The distribution also does not look normal.

```
# Now, I will calculate the mean difference between the three hemisphere categories with
dplyr functions 'group_by' and 'summarize'.
true_diff <- rio2 %>%
  group_by(ns.hemisphere) %>%
  summarize(means = mean(gold.2016)) %>%
  summarize(mean_diff = diff(means))

true_diff
```

```
## # A tibble: 1 x 1
##   mean_diff
##       <dbl>
## 1    -0.551
```

The mean difference in gold medals between the southern and northern hemispheres is 0.5512821.

```
# To get a small gist of a randomization sample, I will resample 2016 gold medal counts
 across the different hemispheres by creating a new dataframe and sampling the gold meda
ls using 'sample'.
perm1 <- data.frame(hemisphere = rio2$ns.hemisphere, gold.medals = sample(rio2$gold.201
6))
head(perm1)
```

```
##   hemisphere gold.medals
## 1   northern           0
## 2   northern           0
## 3   northern           1
## 4   northern           0
## 5   northern           2
## 6   northern           3
```

```
# Similarly to the true mean difference calculation from above, I will calculate the mea
n difference from this sample that I just created.
perm1 %>%
  group_by(hemisphere) %>%
  summarize(means = mean(gold.medals)) %>%
  summarize(mean_diff = diff(means))
```

```
## # A tibble: 1 x 1
##   mean_diff
##       <dbl>
## 1      2.86
```

It looks like the mean difference of medals between the northern and southern hemispheres in this sample was 1.807692.

```
# I will repeat the same procedure again, creating a new sample. This just shows the ove
rall gist of a randomization sample, where this process can be repeated several times.
perm2 <- data.frame(hemisphere = rio2$ns.hemisphere, gold.medals = sample(rio2$gold.201
6))
head(perm2)
```

```
##   hemisphere gold.medals
## 1   northern           1
## 2   northern           6
## 3   northern          27
## 4   northern           0
## 5   northern           0
## 6   northern           0
```

```
# Again, I will find the mean difference of this new sample.
perm2 %>%
  group_by(hemisphere) %>%
  summarize(means = mean(gold.medals)) %>%
  summarize(mean_diff = diff(means))
```

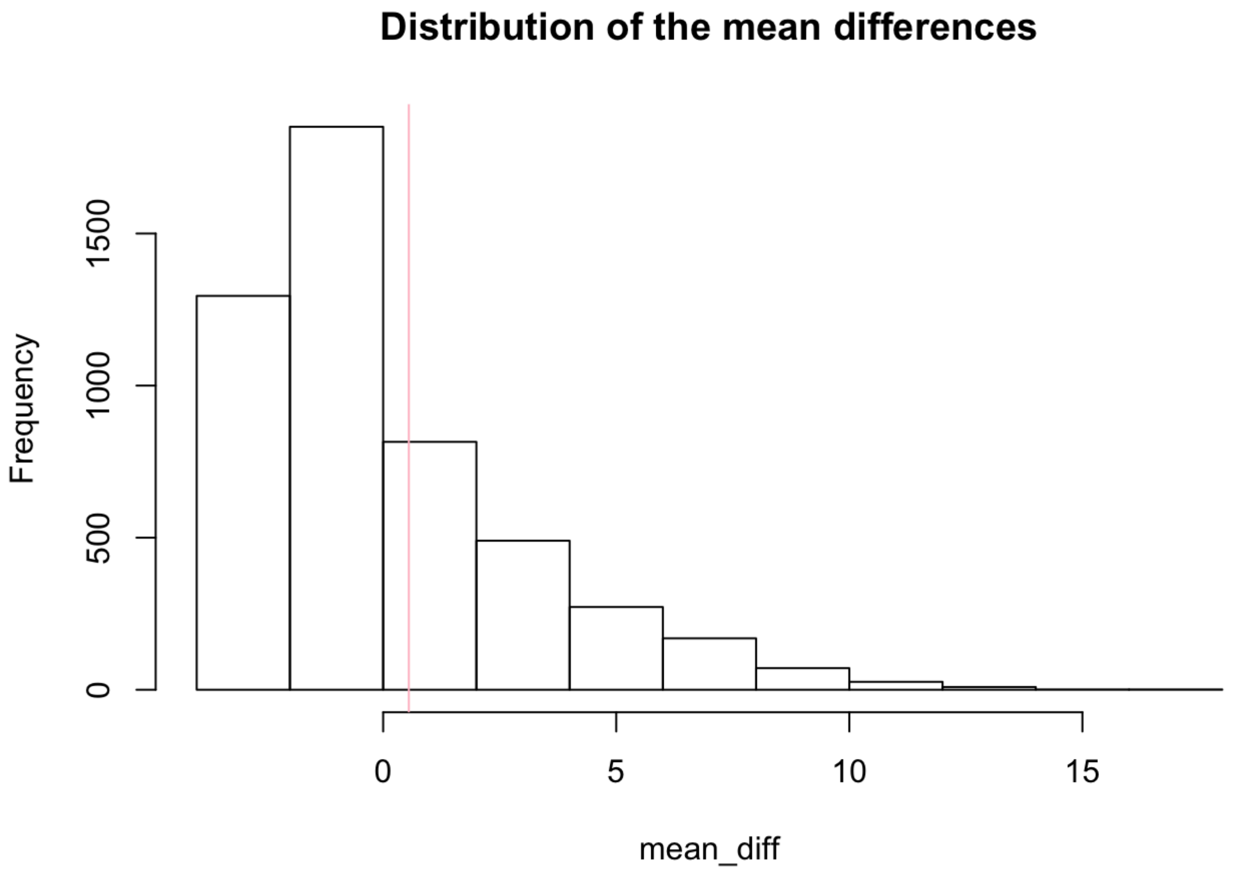
```
## # A tibble: 1 x 1
##   mean_diff
##       <dbl>
## 1     -2.53
```

It looks like the mean difference of medals between the norther and southern hemispheres in this sample is now 2.705128.

```
# In reality, I would want to recreate this randomization of samples many times over. In
order to do this, I need to create an empty vector to store the mean differences. I will
do this with 'vector'.
mean_diff <- vector()
```

```
# Now, I will do the same exact procedure using 'sample' to create a new sample, except
 this time, I will use a 'for' loop to repeat it 5000 times. I will also store the mean
 differences of these runs in the empty vector that I previously created.
for(i in 1:5000){
  temp <- data.frame(hemisphere = rio2$ns.hemisphere, gold.medals = sample(rio2$gold.201
6))

  mean_diff[i] <- temp %>%
    group_by(hemisphere) %>%
    summarize(means = mean(gold.medals)) %>%
    summarize(mean_diff = diff(means)) %>%
    pull
}
```

```
# Previously, I calculated the mean difference between the three hemisphere categories t
o be 0.5512821. I can show the distribution of the mean differences from the randomizati
on test along with  a vertical line showing the value of the true difference using 'his
t' and 'abline'.
{hist(mean_diff, main="Distribution of the mean differences"); abline(v = 0.5512821, col
="pink")}
```



```
# Now, I can calculate the two-sided p-value using 'mean'.
mean(mean_diff > 0.5512821| mean_diff < -0.5512821)
```

```
## [1] 0.8588
```

Based on the results (p = 0.8554), it looks like I failed to reject the null hypothesis, and there is evidence to suggest that the average number of gold medals in 2016 does not vary based on which hemisphere the country came from.

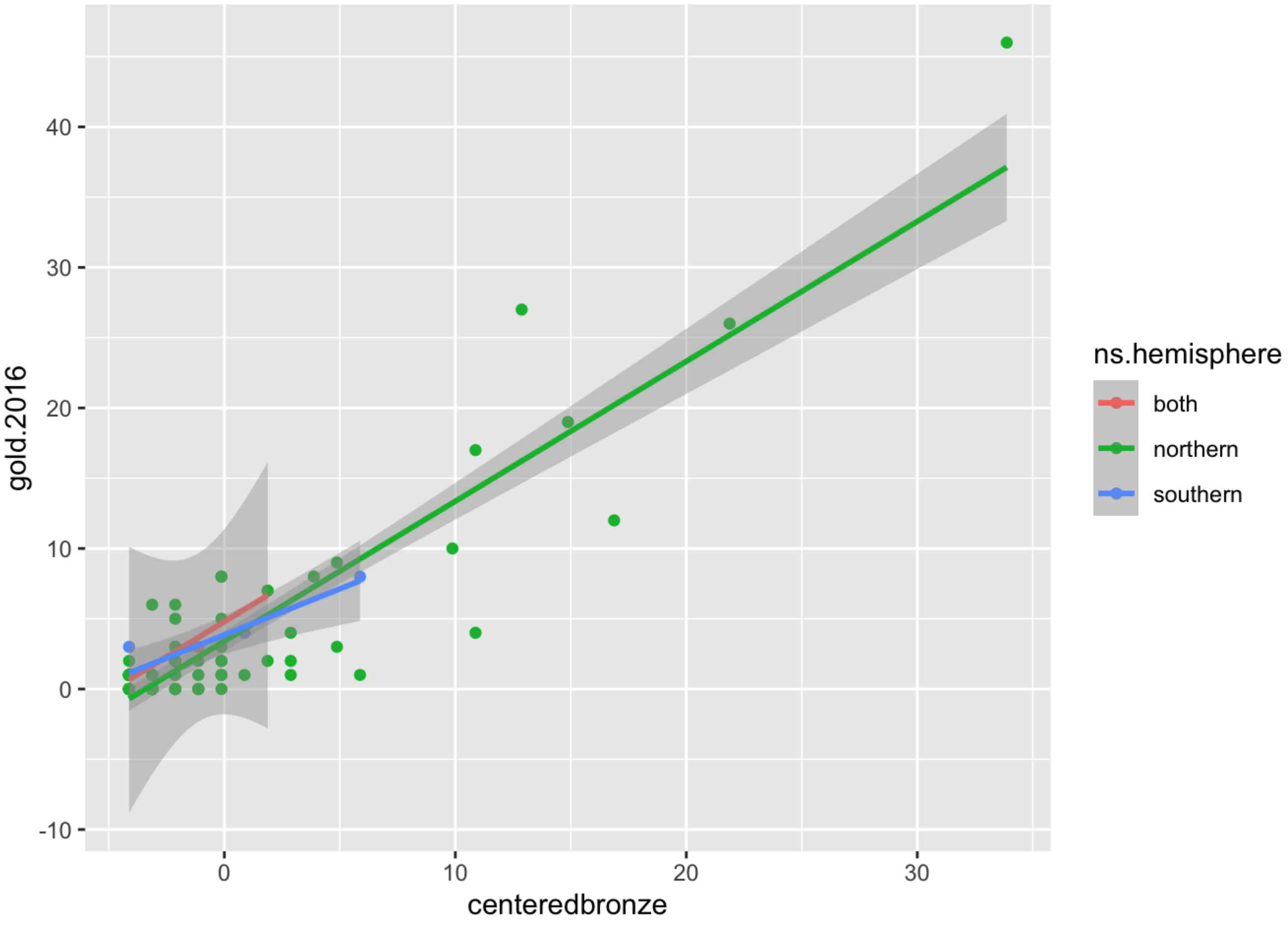
Linear Regression Model

```
# First, I will center the 2016 bronze medals around its mean
rio16$centeredbronze <- rio16$bronze.2016 - mean(rio16$bronze.2016)
# Then, I will fit a multiple regression model with 2016 gold medals and the hemisphere
 and centered bronze medal count as the predictors (and I will get their interactions us
ing '*' too).
fit <- lm(gold.2016 ~ ns.hemisphere * centeredbronze, data = rio16)
summary(fit)
```

```
##
## Call:
## lm(formula = gold.2016 ~ ns.hemisphere * centeredbronze, data = rio16)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2330  -1.2833  -0.1423   1.2579  10.7762
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            4.793103   1.963924   2.441   0.0168 *
## ns.hemispherenorthern                 -1.383962   1.994870  -0.694   0.4898
## ns.hemispheresouthern                 -0.945254   2.376222  -0.398   0.6918
## centeredbronze                         1.000000   0.728436   1.373   0.1736
## ns.hemispherenorthern:centeredbronze -0.004573   0.730442  -0.006   0.9950
## ns.hemispheresouthern:centeredbronze -0.344330   0.805467  -0.427   0.6702
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.09 on 81 degrees of freedom
## Multiple R-squared:  0.8095, Adjusted R-squared:  0.7978
## F-statistic: 68.85 on 5 and 81 DF,  p-value: < 2.2e-16
```

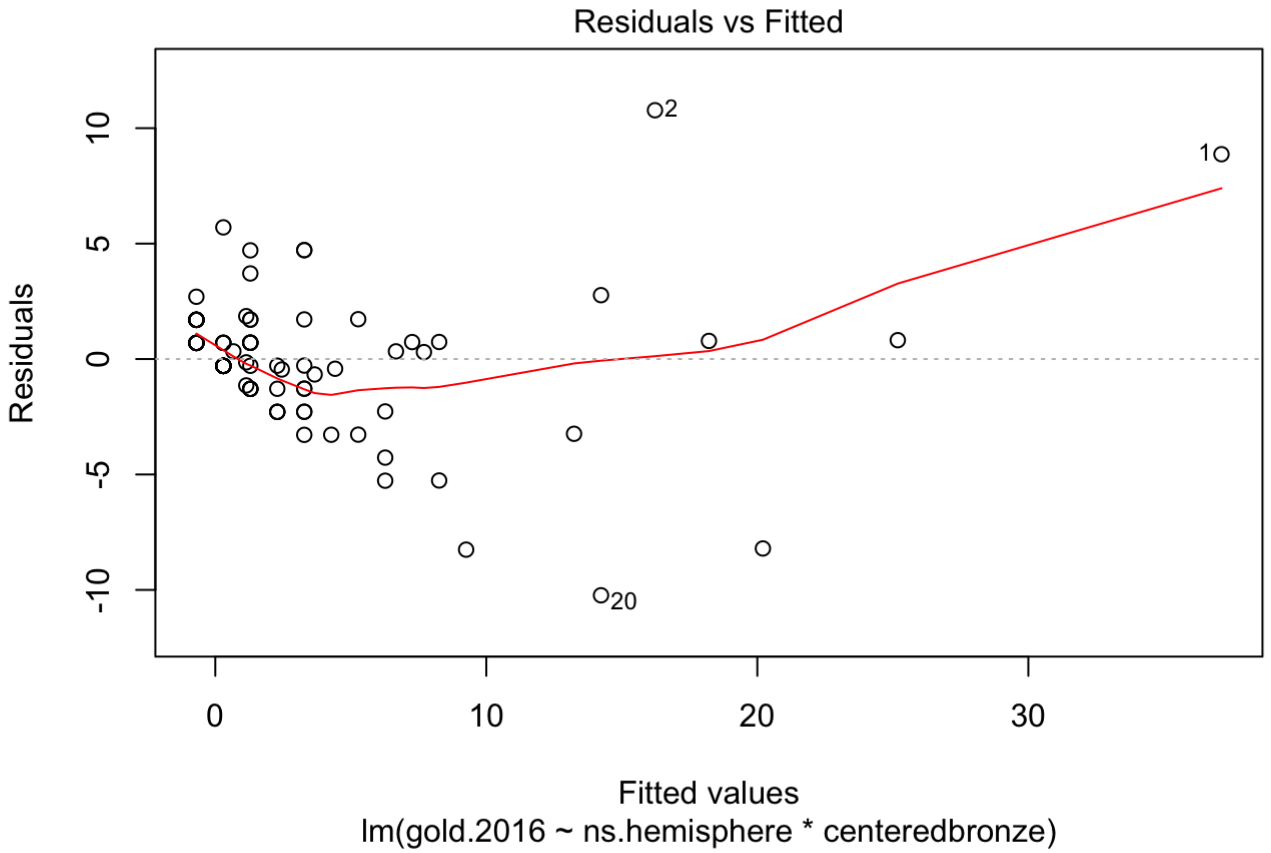
```
# Using 'ggplot', 'geom_point', and 'geom_smooth', I can visualize the relationship betw
een the centerd bronze medals, hemisphere, and number of gold medals.
ggplot(rio16, aes(x = centeredbronze, y = gold.2016, color = ns.hemisphere)) +
  geom_point() +
  geom_smooth(method=lm)
```

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

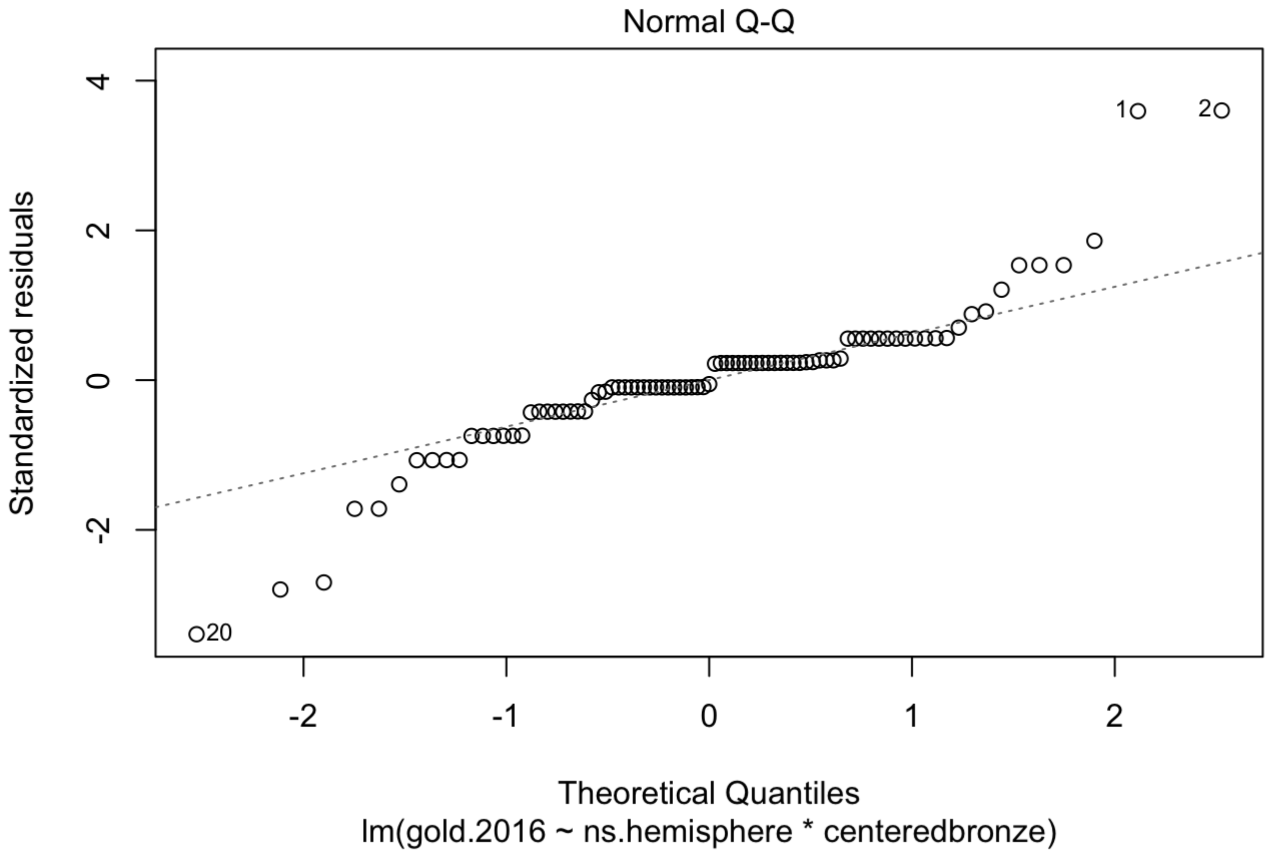


The coeficients represent how the different variables have an impact on the average number of gold medals. Based on these results, there was no significant impact or interaction, but I will interpret the results. Controlling for other variables, a country located in the northern hemisphere will have about 1.4 less gold medals than one in both hemispheres and a southern hemisphere country will have about 0.9 less. For every additional bronze medal, the average gold medals increases by about one medal. A country located in the northern hemisphere with a bronze medal has 0.005 gold medals less than a country in both hemispheres with a bronze medals, and a country in the southern hemisphere with a bronze medal has about 0.3 gold medals less. This model explains roughly 80% of the variation in the 2016 gold medal count.

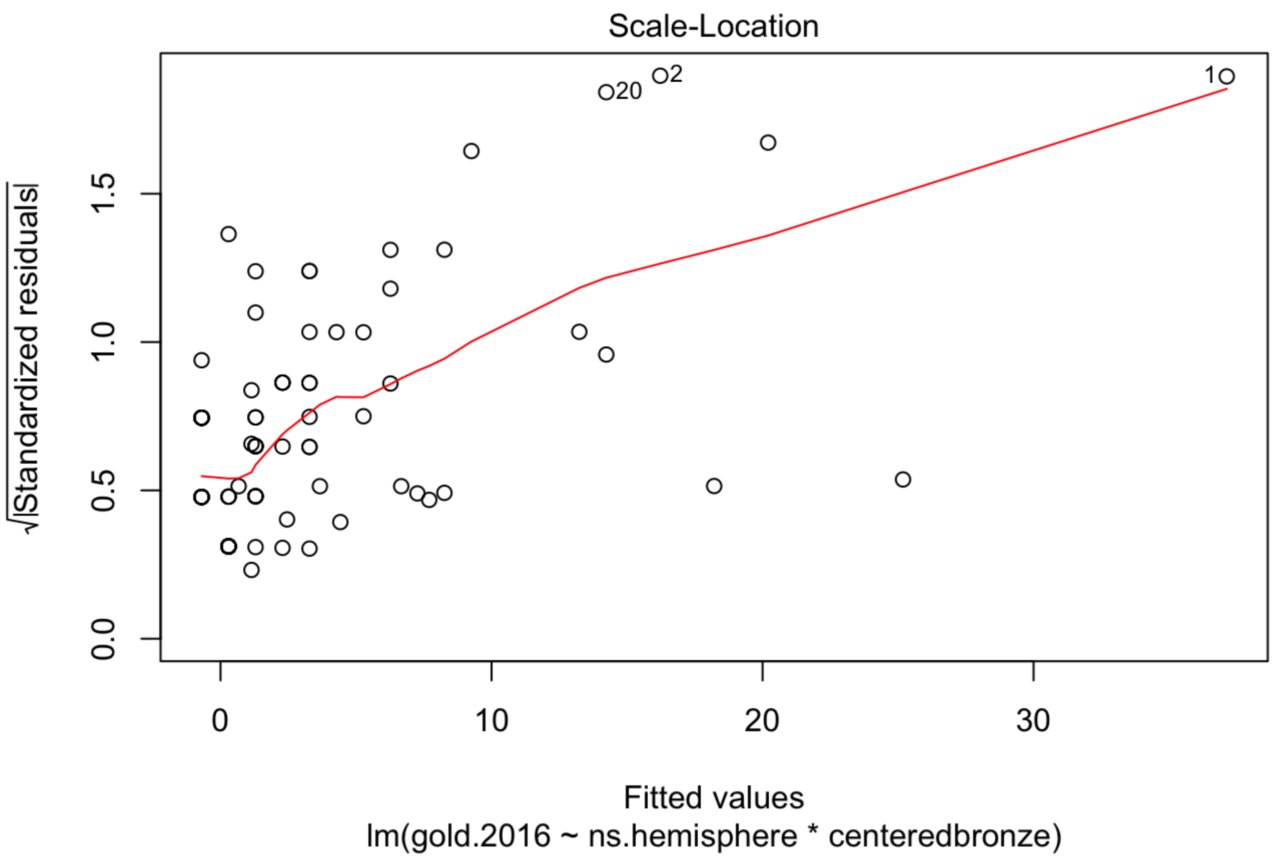
```
# Now, I need to check assumptions for linearity, normality, and homoscedasticity.
plot(fit, 1)
```


Residuals vs Fitted
Residuals
10
5
0
-5
-10
2
1
20
0
10
20
30
Fitted values
lm(gold.2016 ~ ns.hemisphere * centeredbronze)

```
plot(fit, 2)
```


Normal Q-Q
Standardized residuals
4
2
0
-2
1
2
20
-2
-1
0
1
2
Theoretical Quantiles
lm(gold.2016 ~ ns.hemisphere * centeredbronze)

```
plot(fit, 3)
```



It looks like the linearlity assumption is violated, normality looks like its violated, and the homoscedasticity assumption looks like it is not met as well.

```
# Now, I will compute the robust standard error using the 'sandwich' and 'lmtest' packag
es and the 'coeftest' function.
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
coeftest(fit, vcov = vcovHC(fit))
```

```
## 
## t test of coefficients:
## 
##                                       Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                          4.7931034  1.1322406  4.2333 6.035e-05
## ns.hemispherenorthern               -1.3839624  1.1933984 -1.1597   0.24959
## ns.hemispheresouthern               -0.9452542  1.2706584 -0.7439   0.45908
## centeredbronze                       1.0000000  0.4714045  2.1213   0.03695
## ns.hemispherenorthern:centeredbronze -0.0045729  0.4989844 -0.0092   0.99271
## ns.hemispheresouthern:centeredbronze -0.3443299  0.5015357 -0.6866   0.49433
##                                         
## (Intercept)                          ***
## ns.hemispherenorthern                   
## ns.hemispheresouthern                   
## centeredbronze                       *  
## ns.hemispherenorthern:centeredbronze    
## ns.hemispheresouthern:centeredbronze    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The coefficients represent how the different variables have an impact on the average number of gold medals. Based on these results, there a significant result from the centered bronze variable on the averagae gold medal count. Controlling for other variables, for every additional bronze medal, the average gold medals increases by about one medal (p = 0.03695). Other than this significant finding, the coefficients for the interaction terms remained the same, and the coefficients for the hemispheric terms decreased.

```
# Now, I will compute the bootsrtapped standard errors. I will use 'replicate' to repeat
the process.
samp_SEs <- replicate(5000, {
  # Now I will bootsrtap the data with 'sample_frac'...
  boot_data <- sample_frac(rio16, replace = TRUE)
  # and fit the same regression model...
  fitboot <- lm(gold.2016 ~ ns.hemisphere * centeredbronze, data = rio16)
  # and save the coefficients.
  coef(fitboot)
})

# Now I can get the estimated standard errors from the sample by first transposing the m
atricies from 'samp_SEs', changing it into a dataframe, and using 'summarize_all' to get
the standard error.
samp_SEs %>%
  t %>%
  as.data.frame %>%
  summarize_all(sd)
```

```
##   (Intercept) ns.hemispherenorthern ns.hemispheresouthern centeredbronze
## 1           0                     0                     0              0
##   ns.hemispherenorthern:centeredbronze ns.hemispheresouthern:centeredbronze
## 1                                    0                                    0
```

After bootsrtapping, all of the coefficients were zero. this indicates that controlling for all other variables, none of the predictors and their interactions resulted in an increase or decrease in the average gold medals won in 2016. This is very different from the original and robust coefficients, where the predictors and their interactions were shown to inflence the average number of gold medals to some extent.

Logistic Regression

```
# Now I will do a logistic regression. First, I have to create a binomial variable. I wi
ll use the dataset where I have removed the countries located in "both" hemispheres.
rio2 <- rio2 %>%
  mutate(y = ifelse(ns.hemisphere == "northern", 1, 0))
# Now I can run a logistic regression
fit2 <- glm(y ~ gold.2016 + bronze.2016,  data = rio2, family = "binomial")
summary(fit2)
```

```
##
## Call:
## glm(formula = y ~ gold.2016 + bronze.2016, family = "binomial",
##     data = rio2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5080   0.3245   0.3912   0.4240   0.5468
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3638     0.4958   4.767 1.87e-06 ***
## gold.2016    -0.1178     0.1637  -0.720    0.472
## bronze.2016   0.1680     0.1986   0.846    0.398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 83  degrees of freedom
## Residual deviance: 42.347  on 81  degrees of freedom
## AIC: 48.347
##
## Number of Fisher Scoring iterations: 6
```

```
# We can also calculate the odds ratio of the model with 'exp' and 'coef'.
exp(coef(fit2))
```

```
## (Intercept)   gold.2016 bronze.2016
##  10.6315848   0.8888436   1.1829158
```

There are no significant findings, and the coefficients show the expected change in the odds of being from the northern hemisphere. Based on these results and the odds ratio calculations, for a one unit increase in the number of gold medals in 2016, a country is 0.89 times more likely to be in from the northern hemisphere, and for a one unit increase in bronze medals from 2016, a country is 1.18 times more likely to be from the northern hemisphere.

```
# Now, it is time to make a confusion matrix. TI will first need to make column of predi
cted probabilities using 'predict' and based on these probabilities, I can classify the
 country as being either in the northern or southern hemisphere by applying a cutoff aro
und 0.89 with 'ifelse'
rio2$prob1 <- predict(fit2, type = "response")
rio2$predicted <- ifelse(rio2$prob1 > .89, "northern", "southern")

# Now, I will set up a confusion matrix to compare true to predicted condition
table(true_condition = rio2$ns.hemisphere, predicted_condition = rio2$predicted) %>%
  addmargins
```

```
##               predicted_condition
## true_condition northern southern Sum
##       northern       75        3  78
##       southern        5        1   6
##       Sum            80        4  84
```

```
# Accuracy defines correctly defined cases within the confusion matrix.
(75 + 1)/84
```

```
## [1] 0.9047619
```

```
# This matrix was accurate about 90.5% of the time.

# Sensitivity (True Positive Rate, TPR)
4/84
```

```
## [1] 0.04761905
```

```
# It detected a true positive case about 4.8% of the time.

# Specificity (True Negative Rate, TNR)
75/78
```

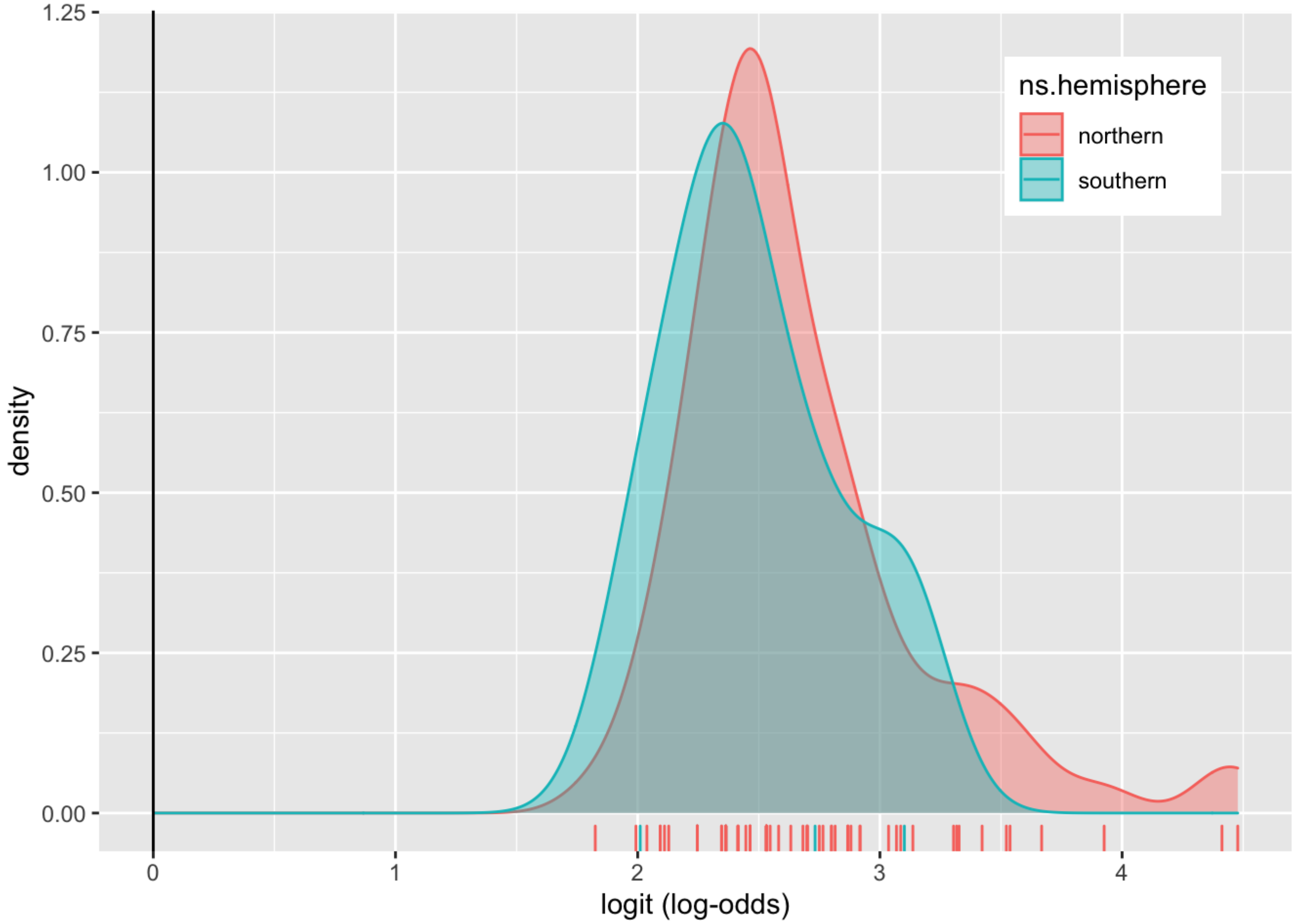
```
## [1] 0.9615385
```

```
# It detected a true negative about 96.2% of the time.

# Precision (Positive Predictive Value, PPV)
1/4
```

```
## [1] 0.25
```

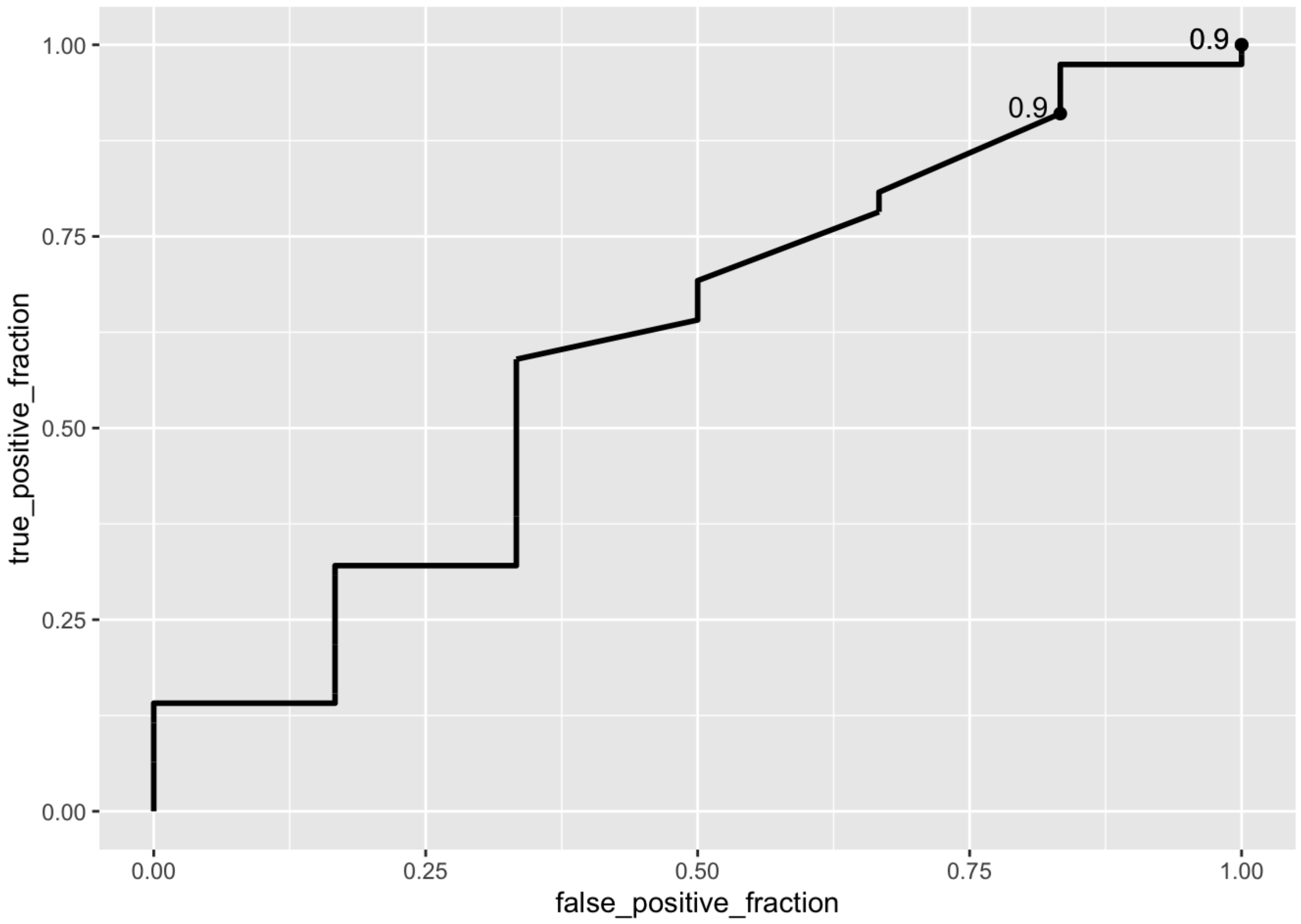
```
# When the model predicted 'southern' and it was actually 'southern' this event occured
 roughly 25% of the time.
```

```
# Now, I can find the predicted log odds with 'predict' to plot them.
rio2$logit <- predict(fit2, type = "link")

# Using 'geom_density', I can make a density plot of log-odds for each hemisphere to see
how difficult it is to predict what hemisphere the country comes from based on the predi
ctors (gold and bronze medals from 2016).
rio2 %>%
  ggplot() +
  geom_density(aes(logit, color = ns.hemisphere, fill = ns.hemisphere), alpha = .4) +
    geom_rug(aes(logit, color = ns.hemisphere)) +
  theme(legend.position = c(.85,.85)) +
  geom_vline(xintercept = 0) +
  xlab("logit (log-odds)")
```



Based on this plot, it looks like these may not be great predictors for assessing which hemisphere the country that competed in the 2016 summer olympics came from.

```
# Now, to make a ROC curve, I need to use the 'plotROC' package that I will call using
 'library'.
library(plotROC)

# Then, I can plot a ROC curve with 'geom_roc' depending on values of y and its probabil
ities in 'aes'. I will also set cutoffs on the graph using 'cutoffs.at'.
ROCplot1 <- ggplot(rio2) +
  geom_roc(aes(d = y, m = prob1), cutoffs.at = list(0.1, 0.5, 0.9))
ROCplot1
```



```
# Now, I can calculate the AUC using 'calc_auc' on the ROC plot I previously made to ass
ess
AUCs <- calc_auc(ROCplot1)$AUC
AUCs
```

```
## [1] 0.607906
```

The ROC plot indicates that the model is not very accurately predictive. The AUC of the ROC plot it 0.607906, and the higher the AUC the better. This indicates that the model predicts the hemisphere of the country correctly about 60.8% of the time.