2016 Summer Olympic and Total Olympic Medals by Country Analysis

```
# Yasamin Yazdani, yy8429
```

Introduction

The title of this report is called “2016 Summer Olympic and Total Olympic Medals by Country Analysis”. This report includes two datasets. The first one, “data16”, is the number of gold, silver, and bronze medals won by country in the 2016 summer olympics as well as whether the countries are located in the Northern Hemisphere, the Southern Hemisphere, or both. The seconds one, “totaldata”, is the grand total of gold, silver, bronze, and toal olympic medals won by country from 1896 to the present. These datasets were acquired from online websites (total data- https://worldpopulationreview.com/country-rankings/olympic-medals-by-country , 2016 data- https://www.insidethegames.biz/games/7/medals) of world data reports on topics like sports. I also used maps to locate countries in the Northern and Southern hemispheres. With these resources, I made my own dataset. I found this data interesting because my grandfather was an olympic gold medalist from Iran. I do not anticipate any specific associations, but I do expect to see that the United States is the top producer of all the different medal cetegories since the start of the olympics and in 2016. I also expect the Northern Hemisphere to be the producer of most olympic medals.

Importing the Datasets

```
# I will use the package "readxl" to import the two datasets.
library(readxl)
# Now I will import the datasets. The first one is the 2016 medals by country
data16 <- read_xlsx("2016olympic.xlsx")
totaldata <- read_excel("totalolympic.xlsx")
```

Tidying the Dataset

The datasets that I have used for this project are already tidy because I have created the datasets, therefore I do not need to reshape them in any way. This means that I can move onto joining the two datasets.

Joining The Datasets

```
# Now, I will merge the two datasets together into one dataset. First, I need
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
```

```
# Now, I will join my data using the "left_join" function. I used this type of
mydata1 <- left_join(data16, totaldata, by = c("country"))
# Now, I will take a look at the new dataset and compare it to the other two t
glimpse(mydata1)
```

```
## Rows: 87
## Columns: 9
## $ country      <chr> "United States", "United Kingdom", "China", "Russia",
## $ gold.2016    <dbl> 46, 27, 26, 19, 17, 12, 10, 9, 8, 8, 8, 8, 7, 7, 6, 6
## $ silver.2016  <dbl> 37, 23, 18, 18, 10, 8, 18, 3, 12, 11, 7, 3, 6, 4, 6,
## $ bronze.2016  <dbl> 38, 17, 26, 19, 15, 21, 14, 9, 8, 10, 4, 4, 6, 6, 1,
## $ ns.hemisphere <chr> "northern", "northern", "northern", "northern", "nort
## $ gold.total   <dbl> 1127, 274, 237, 195, 283, 156, 248, 121, 246, 152, 13
## $ silver.total <dbl> 907, 299, 195, 163, 282, 158, 276, 112, 214, 168, 136
## $ bronze.total <dbl> 793, 310, 176, 188, 290, 183, 316, 104, 241, 192, 149
## $ grand.total  <dbl> 2827, 883, 608, 546, 855, 497, 840, 337, 701, 512, 41
```

```
glimpse(data16)
```

```
## Rows: 87
## Columns: 5
## $ country       <chr> "United States", "United Kingdom", "China", "Russia",
## $ gold.2016     <dbl> 46, 27, 26, 19, 17, 12, 10, 9, 8, 8, 8, 8, 7, 7, 6, 6
## $ silver.2016   <dbl> 37, 23, 18, 18, 10, 8, 18, 3, 12, 11, 7, 3, 6, 4, 6,
## $ bronze.2016   <dbl> 38, 17, 26, 19, 15, 21, 14, 9, 8, 10, 4, 4, 6, 6, 1,
## $ ns.hemisphere <chr> "northern", "northern", "northern", "northern", "nort
```

```
glimpse(totaldata)
```

```
## Rows: 135
## Columns: 5
## $ country      <chr> "United States", "United Kingdom", "Germany", "France"
## $ gold.total   <dbl> 1127, 274, 283, 248, 246, 202, 237, 195, 188, 137, 176
## $ silver.total <dbl> 907, 299, 282, 276, 214, 216, 195, 163, 174, 166, 149,
## $ bronze.total <dbl> 793, 310, 290, 316, 241, 234, 176, 188, 158, 198, 173,
## $ grand.total  <dbl> 2827, 883, 855, 840, 701, 652, 608, 546, 520, 501, 498
```

It appears that the total olympic dataset had more cases than the 2016 one. As a result, all the countries from the 2016 data (87) were retained and all of the countries that were from the total data who did not compete in the olympics were dropped off on the merged dataset. I will do a simple calculation below to determine what that number is.

```
135-87
```

```
## [1] 48
```

A total of 48 cases/countries were dropped in the new, joined dataset. This means that out of the 135 total countries to have ever competed in the olympics, 87 of them were at the 2016 olympics.

Exploring Summary Statistics

```
# Now I am going to do some exploring with the 6 core dplyr functions. First,
project1 <- mydata1 %>% mutate(rioproportion = (gold.2016 + silver.2016 +bronz
# Now, I used the "summarize" function on my new dataset to pick the variables
project1  %>% summarize(rioproportion, country) %>% arrange(desc(rioproportior
```

```
## # A tibble: 87 x 2
##    rioproportion country
##            <dbl> <chr>
##  1         100   Fiji
##  2         100   Kosovo
##  3         100   Jordan
##  4          66.7 Bahrain
##  5          66.7 Ivory Coast
##  6          53.3 Serbia
##  7          50   Vietnam
##  8          50   Grenada
##  9          50   Burundi
## 10          50   Niger
## # … with 77 more rows
```

Based on this analysis, it looks like three countries, Fiji, Kosovo, and Jordan, won all of their olympic medals at the 2016 summer olympics in Rio.

```
# Now, lets look at the average of the total medals won when grouped by the he
Project2 <- mydata1 %>% group_by(ns.hemisphere) %>% summarise(hemisphere.avg =
# Now I will take a look at the table by simply typing the name of the dataset
Project2
```

```
## # A tibble: 3 x 2
##   ns.hemisphere hemisphere.avg
##   <chr>                  <dbl>
## 1 northern                194.
## 2 southern                132.
## 3 both                     63
```

As you can see, the northern hemisphere has more olympic medals than the southern hemisphere and countries in both hemispheres. This may be because most of the world’s countries are located in the northern hemisphere, thus leading to more opportunities and people to win olympic medals.

```
# Now, I will explore the standard deviation of silver medals won in 2016 for
Project3 <- mydata1 %>% filter(ns.hemisphere == "southern") %>% mutate(sd.silv
Project3
```

```
## # A tibble: 6 x 2
##   sd.silver.16 country
##          <dbl> <chr>
## 1         4.68 Australia
## 2         4.68 New Zealand
## 3         4.68 Argentina
## 4         4.68 South Africa
## 5         4.68 Fiji
## 6         4.68 Burundi
```

There are six sounthern hemisphere countries in the 2016 olympics, and the standard deviation of silver medals for the 2016 olympics is about 4.7 medals.

```
# Lets look at the variance in the bronze medals of 2016. This is done by crea
project4 <- mydata1 %>% summarize(var.bronze.16 = var(bronze.2016))
project4
```

```
## # A tibble: 1 x 1
##   var.bronze.16
##           <dbl>
## 1          39.3
```

The variance in bronze medals for the 2016 olympics is 39.27453 square medals.

```
# Let's now see how many distinct values there are for the total gold medals i
project5 <- mydata1 %>% summarize(distinct.gold = n_distinct(gold.total))
project5
```

```
## # A tibble: 1 x 1
##   distinct.gold
##           <int>
## 1            50
```

There are 50 unique values for total gold medals. This means that for 37 of the countries (87-50), their number of gold medals overlaps with at least one other country.

```
# Now we will look at the 95th percentile of silver medals in all of the olymp
project6 <- mydata1 %>% summarize(silver.95th = quantile(silver.total, probs =
project6
```

```
## # A tibble: 1 x 1
##   silver.95th
##         <dbl>
## 1        215.
```

It looks like the 95th percentile of total silver medals is 215.4 medals.

```
# Now, we will look at the minimum number of total bronze medals. To do this,
project7 <- mydata1 %>% select(country, bronze.total) %>% summarize(bronze.mir
project7
```

```
## # A tibble: 1 x 1
##   bronze.min
##        <dbl>
## 1          0
```

```
# It looks like zero medals was the minimum for bronze medals. Let's see what
mydata1 %>% filter(bronze.total == "0") %>% select(country, bronze.total)
```

```
## # A tibble: 7 x 2
##   country bronze.total
##   <chr>          <dbl>
## 1 Bahrain            0
## 2 Vietnam            0
## 3 Fiji               0
## 4 Kosovo             0
## 5 Jordan             0
## 6 Grenada            0
## 7 Burundi            0
```

The seven countries with zero bronze medals are Bahrain, Vietnam, Fiji, Kosovo, Jordan, Grenada, and Burundi.

```
# Now, we will look at the maximum total number of medals. To do this, I will
project8 <- mydata1 %>% select(country, grand.total) %>% summarize(total.max =
project8
```

```
## # A tibble: 1 x 1
##   total.max
##       <dbl>
## 1      2827
```

```
# It looks like 2827 medals was the maximum for total medals won in all the ol
mydata1 %>% filter(grand.total == "2827") %>% select(country, grand.total)
```

```
## # A tibble: 1 x 2
##   country       grand.total
##   <chr>               <dbl>
## 1 United States        2827
```

The United States had the highest number of total olympic medals. Go USA!

```
# Now I want to see the number of observations for 'ns.hemisphere' after group
project9 <- mydata1 %>% select(ns.hemisphere, gold.2016, gold.total) %>% group
project9
```

```
## # A tibble: 3 x 3
##   ns.hemisphere `n()` gold.correlation
## * <chr>         <int>            <dbl>
## 1 both              3            0.919
## 2 northern         78            0.873
## 3 southern          6            0.964
```

There appears to be a strong, positive correlation (0.96) between the number of gold medals in 2016 and the number of gold medals total for the southern hemisphere countries. This correlation stays strong, but decreases going to the countries in both hemispheres and finally to northern hemisphere countries. There are 6 countries in the southern hemisphere, 78 in the northern hemisphere, and 3 countries in both.

Making a Table of Summary Statistics Using Dashes and Vertical Bars

Summary	Statistic	Value	Unit	Description
1	Proportion of 2016 medals to total medals	0-100	%	Findings- Fiji, Kosovo, and Jordan won their first olympic medal(s) in 2016.

Summary	Statistic	Value	Unit	Description
2	Average of total medals won by hemisphere	63-194	medals	Findings- The northern hemisphere countries have the highest average (194 medals).
3	SD of 2016 silver medals (southern hemisphere)	4.68	silver medals	Findings- SD is 4.68 medals and there are 6 countries in the southern hemisphere.
4	Variance in bronze medals in 2016	39.27	medals-squared	Findings- The variaence is 39.27 bronze medals-squared for the 2016 olympics.
5	Distinct gold medal values for total medals	50	distinct values	Findings- 50 unique gold medal values means there're overlaps for 37 countries.
6	95th percentile of total silver medals	215.4	silver medals	Findings- The 95th percentile for total silver medals is 215.4 medals.
7	Minimum of total bronze medals	0	bronze medals	Findings- Bahrain, Vietnam, Fiji, Kosovo, Jordan, Greneda, & Burundi have 0 bronze medals
8	Maximum total number of medals	2827	total medals	Findings- The USA has the most medals (2827) out of all the olympic countries!

Summary	Statistic	Value	Unit	Description
9a	Number of observations/countries by hemisphere	3-78	countries	Findings- Only six countries are from the southern hemisphere in the dataset.
9b/10	Correlation-2016 gold/total gold by hemisphere	.87-.96	n/a	Findings- There is a strong positive correlation in all three hemispheres.

Making Visualizations

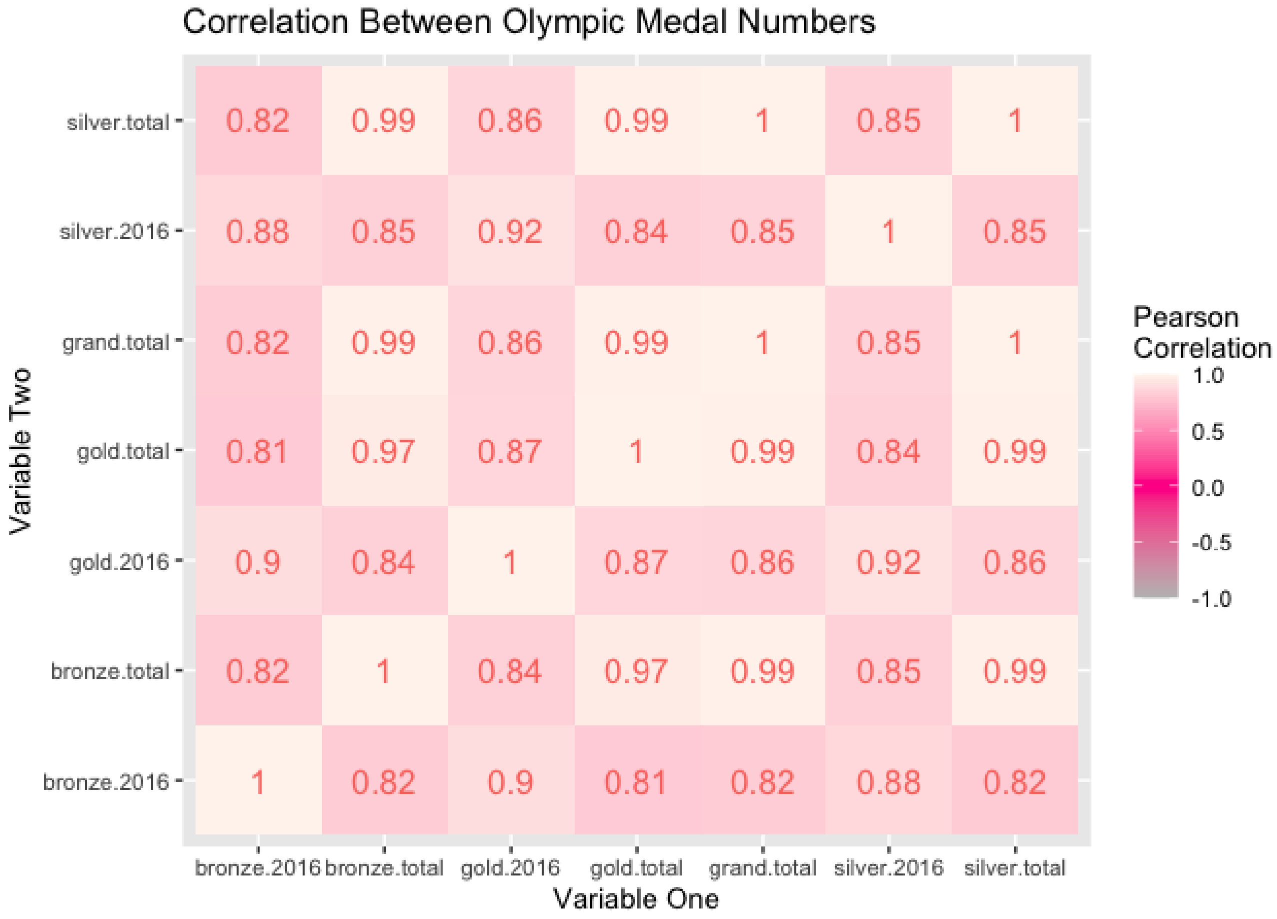
```
# To start making visualizations, I will need the "ggplot2" package. Additioar
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3
```

```
## ✓ 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()
```

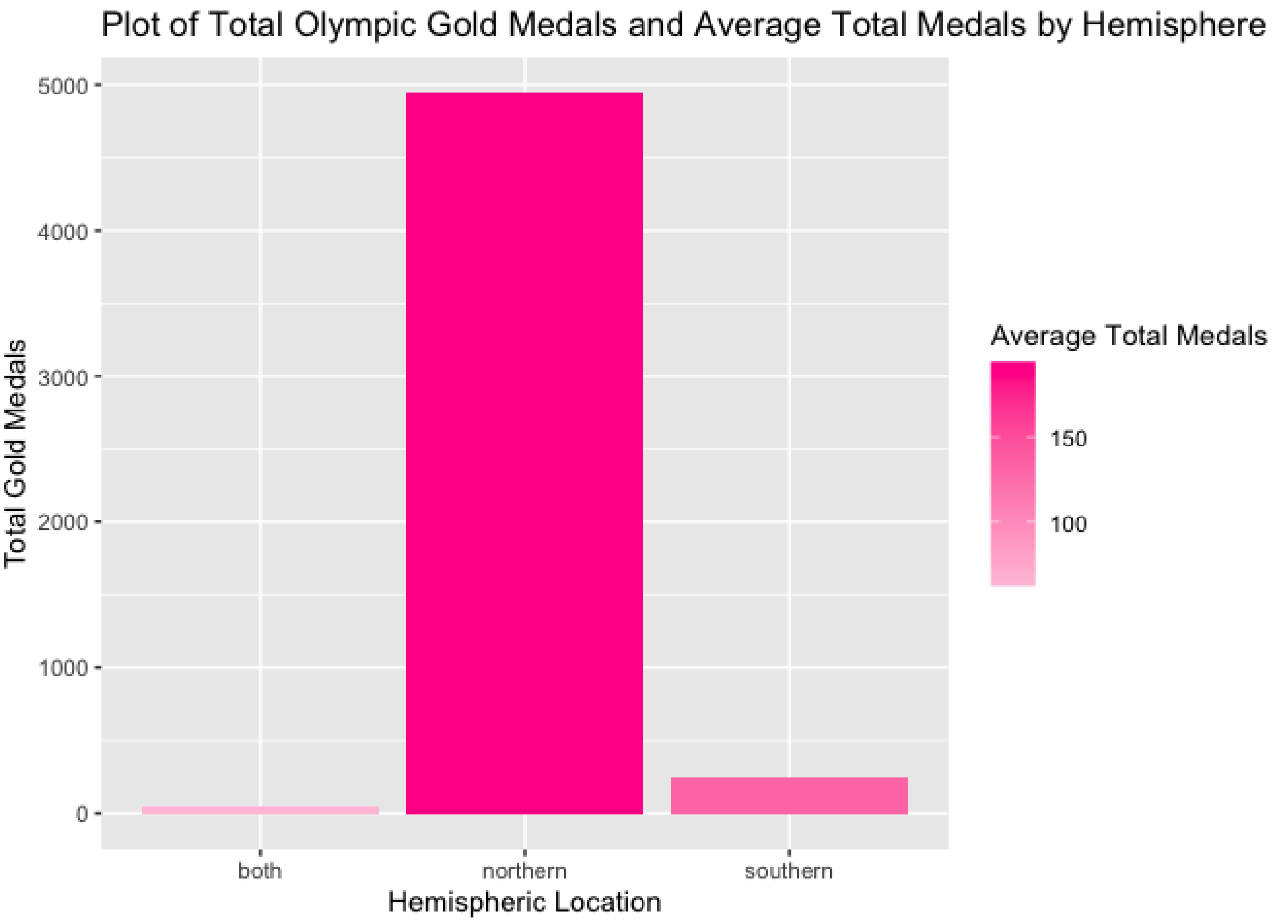
```
# Now, I will make a correlation heatmap. First, I need to remove my categoric
projectcor1 <- mydata1 %>% select(-country, -ns.hemisphere)
# Then, I will use 'cor' to make a correlation matrix with pairwise observatic
cor(projectcor1, use = "pairwise.complete.obs") %>%
# Now, I will save the matrix 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 colum
  pivot_longer(-1, names_to = "other_var", values_to = "correlation") %>%
# Now, I will make the correlation heatmap using 'ggplot' and 'geom_tile'. My
  ggplot(aes(rowname, other_var, fill = correlation)) +
  geom_tile() +
# To make the graph pretty, I will use 'scale_fill_gradient2', set the midpoir
  scale_fill_gradient2(low = "gray", high = "seashell1", mid = "deeppink", mic
  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'
  geom_text(aes(label = round(correlation, 2), color = "roseybrown4", size = 4
```



One thing I found particularly unique is that there is a very strong correlation between any two variables. The lowest correlation value between two distinct variables is 0.81 and is between ‘gold.total’ and ‘’bronze.2016’, and the highest correlation between two distinct variables is 0.99. This value is for variables ‘grand.total’ and ‘bronze.total’, ‘grand.total’ and ‘gold.total’, ‘silver.total’ and ‘bronze.total’, and ‘silver.total’ and ‘gold.total’.

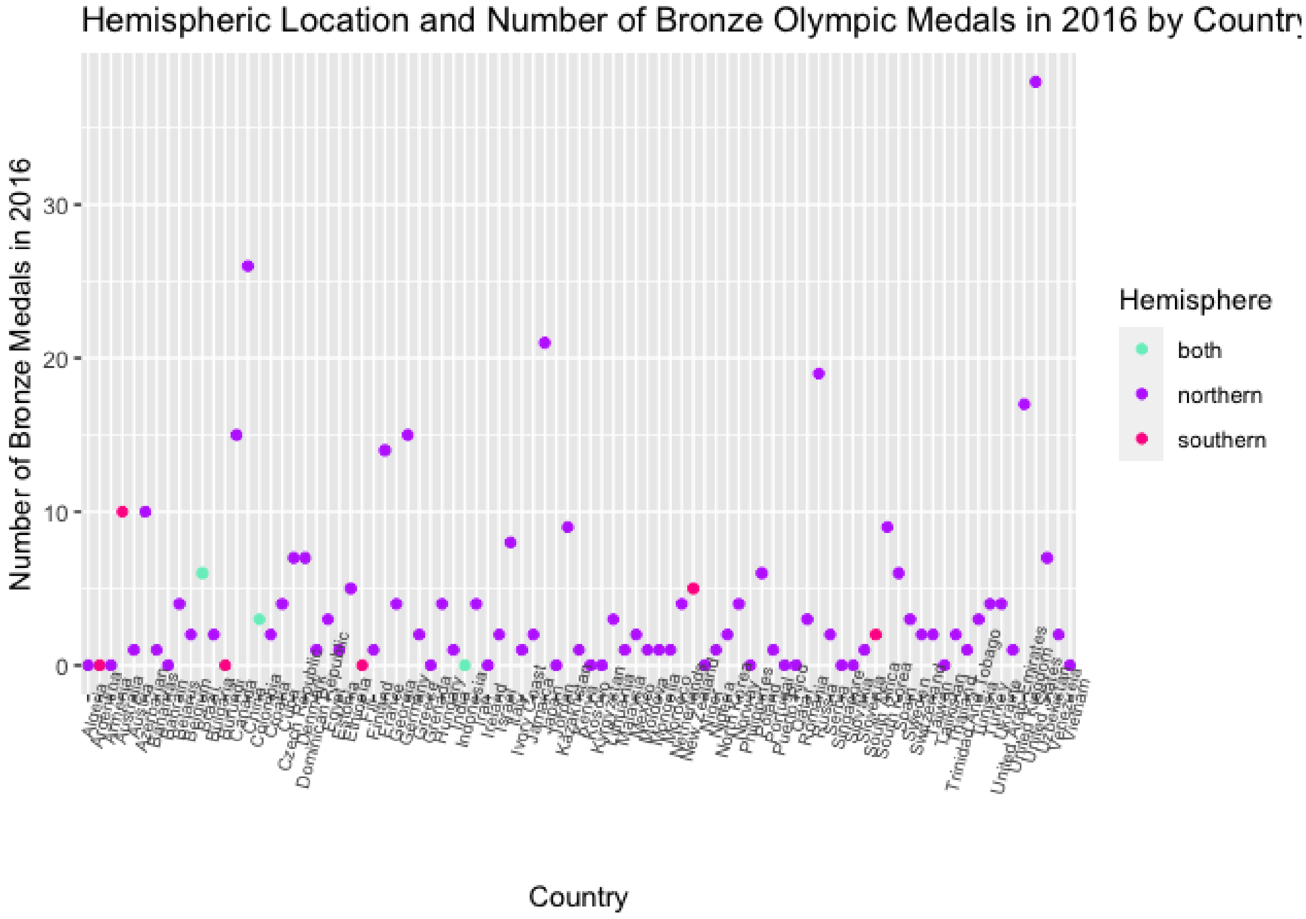
```
# Now I want to make a barchart. First, I will create a new dataset with three
projectplot1 <- mydata1 %>% select(-country, -silver.2016, -silver.total, -brc

# Now, I will use 'ggplot' to make a bar chart. I will use the dataset I just
ggplot(projectplot1, aes(x = ns.hemisphere, y = totalgold, fill = averagetotal
  # Now I will use 'geom_col' to make a bar chart.
  geom_col() +
  # To make the graph pretty, I used 'scale_fill_gradient2' to have the lower
  scale_fill_gradient2(low = "white", high = "deeppink", name = "Average Total
  ggtitle("Plot of Total Olympic Gold Medals and Average Total Medals by Hemis
  labs(x = "Hemispheric Location", y = "Total Gold Medals")
```



If we look at the chart, we can see that countries located in both hemispheres have the lowest gold medal count and the lowest average total medals. The southern hemispheric countries are not too far behind the countries in both hemispheres, and the northern hemisphere has an overwhelming advantage over all the other countries in terms of total gold medals and average total medals. Go team USA!

```
# Now I want to use 'ggplot' on my original merged 'mydata1' dataset and use '
ggplot(mydata1, aes(x = country, y = bronze.2016, color = ns.hemisphere)) +
  geom_point() +
# Now, I need to change the alignment and size of the x-axis text using 'theme
  theme(axis.text.x = element_text(angle = 75, size = 7)) +
# To make the graph pretty, I used 'scale_color_manual' to manually fill the h
  scale_color_manual(values = c("aquamarine2", "darkorchid1", "deeppink"), nam
  ggtitle("Hemispheric Location and Number of Bronze Olympic Medals in 2016 by
  labs(x = "Country", y = "Number of Bronze Medals in 2016")
```



One thing that surprised me in this graph is that most of the bronze medals are won by northern hemisphere countries. This may be because there is a disproportional amount of them relative to the other two categories, but I still expected to maybe see a southern hemisphere country mixed in at the top of the scatterplot with the northern hemisphere countries.

Performing PCA

```
# Last but not least, I will perform a PCA of the dataset. I will use the data
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://gc
```

```
# Now, I will start the PCA. I will also use 'scale' to scale the data to zerc
pca1 <- projectcor1 %>%
  scale() %>%
  prcomp()
```

```
# Now, let's take a look at the results of the PCA using 'names'.
names(pca1)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

It looks like we have a list of standard deviations on each principal component (sdev) as well as the rotated data (x) amongst other things.

```
# Now, we will start to visualize the results by simply showing the PCA that w
pca1
```

```
## Standard deviations (1, .., p=7):
## [1] 2.521520e+00 6.368481e-01 3.572325e-01 2.837612e-01 1.597018e-01
## [6] 5.215940e-02 3.345029e-16
## 
## Rotation (n x k) = (7 x 7):
##                     PC1        PC2         PC3         PC4         PC5
## gold.2016    -0.3717343 -0.4065963 -0.27105032 -0.72747863  0.30417422
## silver.2016  -0.3673712 -0.4218391 -0.57793068  0.57375791 -0.15364697
## bronze.2016  -0.3590301 -0.5076085  0.76230398  0.13219134 -0.11854171
## gold.total   -0.3858101  0.3039427 -0.03100003 -0.22396378 -0.67366155
## silver.total -0.3876367  0.3246470  0.01163857 -0.01060899  0.04102990
## bronze.total -0.3846741  0.3177671  0.09910151  0.27167688  0.64216742
## grand.total  -0.3884116  0.3167208  0.02242596 -0.00463742 -0.04400202
##                      PC6           PC7
## gold.2016     0.03574510  2.642315e-16
## silver.2016  -0.01675529  3.925464e-18
## bronze.2016  -0.02823007 -1.762236e-16
## gold.total    0.38444931 -3.256186e-01
## silver.total -0.81524385 -2.789052e-01
## bronze.total  0.43016261 -2.645782e-01
## grand.total   0.01345157  8.638187e-01
```

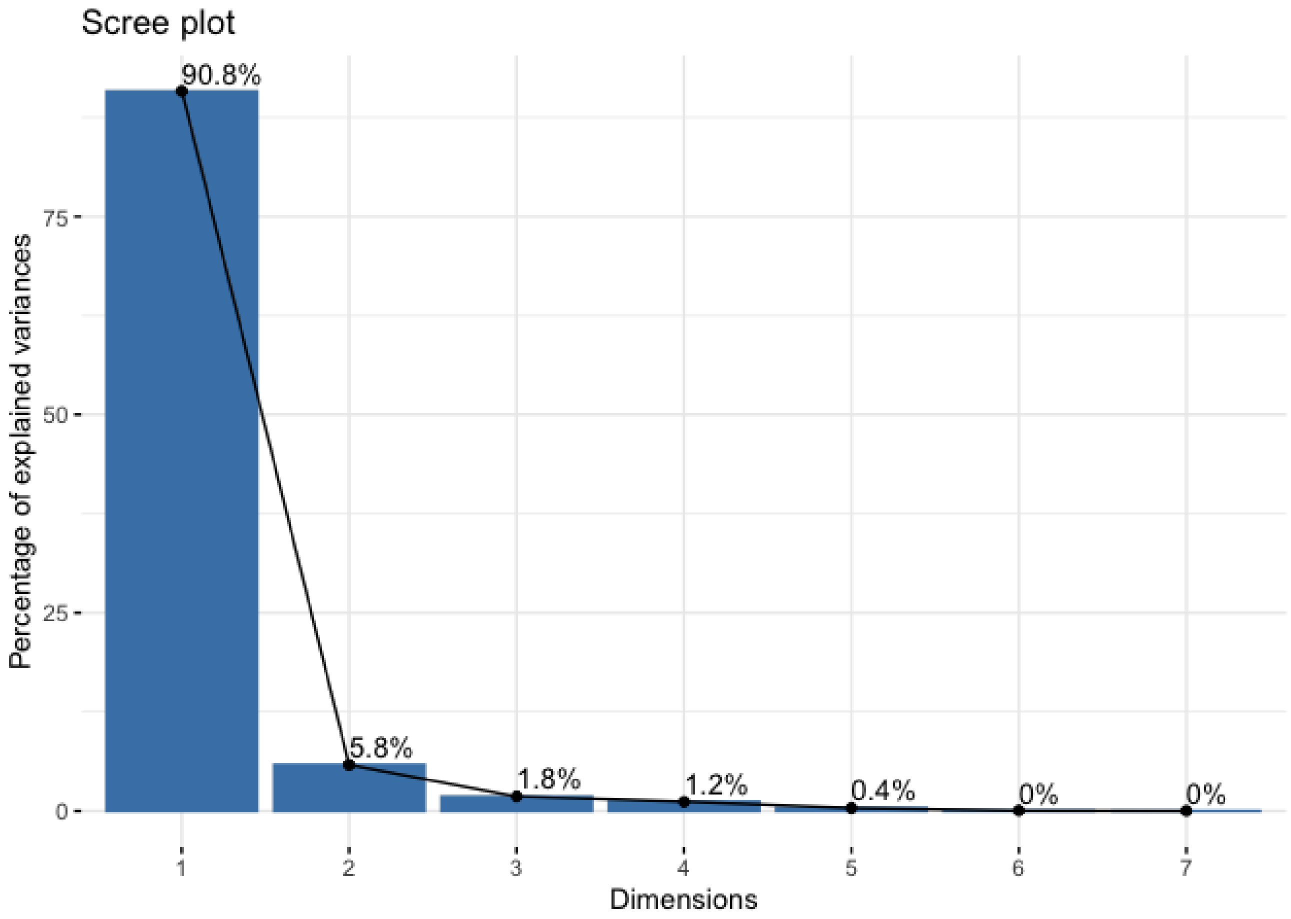
This gives us a better look at the standard deviations on each principal component.

```
# Now we can start to add the 'ns.hemisphere' variable abck to the PCA data 'p
pca2 <- data.frame(pca1$x, country = mydata1$country, hemisphere = mydata1$ns.
head(pca2)
```

```
##           PC1        PC2        PC3         PC4         PC5          PC6
## 1 -17.522979  1.3847117 -0.2628060 -0.52300229 -0.72521187 -0.008559626
## 2  -6.236359 -1.4151737 -1.0792590 -0.05500434  0.63657781 -0.080174121
## 3  -5.173262 -2.7110057  0.4188144 -0.51163422 -0.03890958 -0.020599523
## 4  -4.144673 -1.9314758 -0.1433790  0.18339216  0.05590058  0.123142063
## 5  -4.620993  0.1828435  0.3542387 -0.39651128  0.41663737 -0.020247663
## 6  -3.046085 -1.1036676  1.3811752  0.02214475  0.14246674  0.010945528
##              PC7        country hemisphere
## 1 -1.154632e-14  United States   northern
## 2 -1.332268e-15 United Kingdom   northern
## 3 -2.220446e-15          China   northern
## 4 -8.881784e-16         Russia   northern
## 5 -1.998401e-15        Germany   northern
## 6 -6.661338e-16          Japan   northern
```

We will use this later when making a scatter plot of the principal components.

```
# Then, we will determine how many principal components to consider using a sc
fviz_screeplot(pca1, addlabels = TRUE)
```



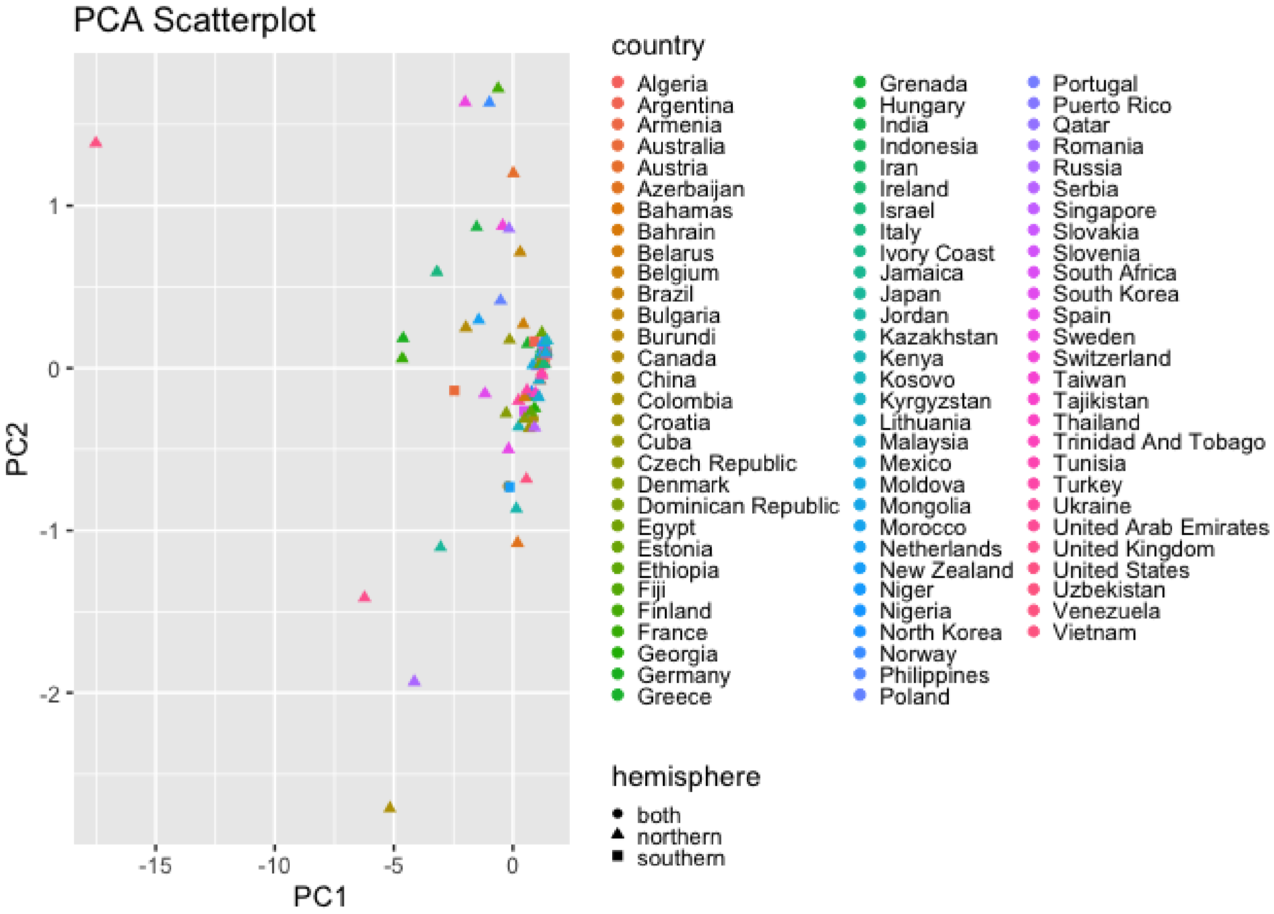
```
get_eigenvalue(pca1)
```

```
##         eigenvalue variance.percent cumulative.variance.percent
## Dim.1 6.358064e+00     9.082948e+01                    90.82948
## Dim.2 4.055755e-01     5.793935e+00                    96.62342
## Dim.3 1.276151e-01     1.823072e+00                    98.44649
## Dim.4 8.052043e-02     1.150292e+00                    99.59678
## Dim.5 2.550468e-02     3.643525e-01                    99.96113
## Dim.6 2.720603e-03     3.886575e-02                   100.00000
## Dim.7 1.118922e-31     1.598459e-30                   100.00000
```

Based on this analysis, we should consider one principal componenet. This is because the second through seventh principal components do not greatly contribute to the variance of the data, while the first principal component contributes to about 91% of the variance (eigenvalue

6.36).

```
# Now, I will plot the data according to principle component considered from a
ggplot(pca2, aes(x = PC1, y = PC2, color = country)) +
  geom_point(aes(shape = hemisphere)) +
# This original plot can barely be seen because the legend is so large and ext
  theme(legend.key = element_blank(), legend.key.size = unit(1, "point")) +
# To continue making the legend smaller, we will assign 30 countries (which th
  guides(color = guide_legend(nrow = 30)) +
  ggtitle("PCA Scatterplot")
```



As we can see, the principal components puts the dataframe into clusters of observations within the dataset according to based on their similarity. Although the second PC was not nearly as contribtive to the variance in data, I have included it in the visualization as well.

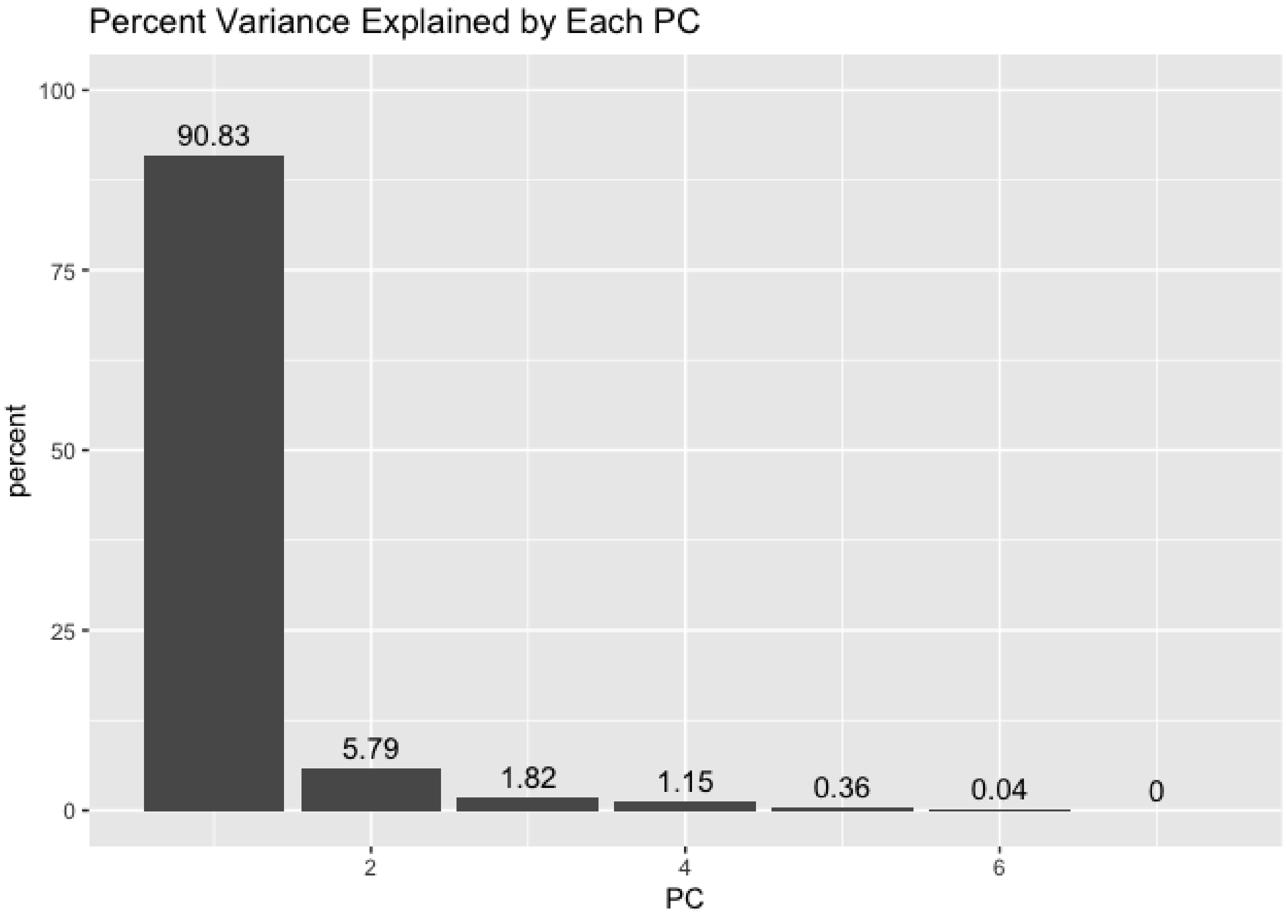
```
# We cen then do some calculations to see how much variance is caused by each
pca3 <- 100* (pca1$sdev^2 / sum(pca1$sdev^2))
pca3
```

```
## [1] 9.082948e+01 5.793935e+00 1.823072e+00 1.150292e+00 3.643525e-01
## [6] 3.886575e-02 1.598459e-30
```

Do these values look familiar? That is because it is shown as ‘variance.percent’ when we used ‘get_eigenvalue’ earlier.

```
# Now, we will save these percentages as a dataframe with 'data.frame'.
pca4 <- data.frame(percent = pca3, PC = 1:length(pca3))
```

```
# Now, we will use these percentages to see how they compare to one-another ac
ggplot(pca4, aes(x = PC, y = percent)) +
  geom_col() +
# Now I will label the graphs with their percentages with 'geom.text', roudinc
  geom_text(aes(label = round(percent, 2)), size = 4, vjust = -0.5) +
# Finally, I will set a limit to the y-axis of 100 to ensure all of the variar
  ylim(0, 100) +
  ggtitle("Percent Variance Explained by Each PC")
```



As we can see, the first PC contributes far greater to the variance of the data than all the other PCs. It contributes to 90.83% of the variance.