Summer Olympics Analysis

2024-08-12

```
options(repos = c(CRAN = "https://cran.rstudio.com/"))
install.packages("dplyr")
```

```
## Installing package into '/Users/swain/Library/R/arm64/4.4/library'
## (as 'lib' is unspecified)
```

```
##
## The downloaded binary packages are in
##  /var/folders/0s/0md080fx6xxg3fvxdh43mm_r0000gn/T//RtmpJ6uSks/downloaded_packages
```

```
install.packages("ggplot2")
```

```
## Installing package into '/Users/swain/Library/R/arm64/4.4/library'
## (as 'lib' is unspecified)
```

```
##
## The downloaded binary packages are in
##  /var/folders/0s/0md080fx6xxg3fvxdh43mm_r0000gn/T//RtmpJ6uSks/downloaded_packages
```

```
library(dplyr, ggplot2)
```

```
##
## 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
```

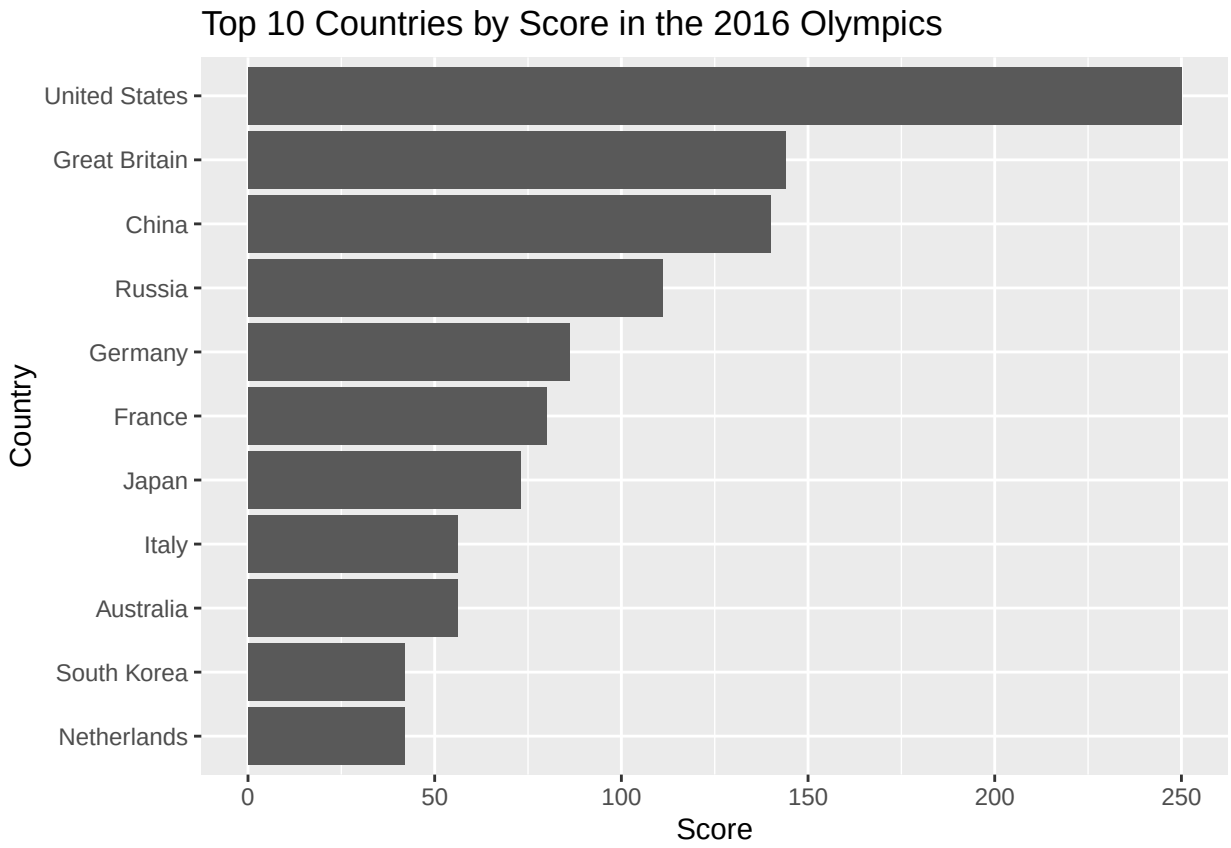
```
olympics <- read.csv("olympics_data.csv")
```

```
olympics2016 <- filter(olympics,year == 2016)
```

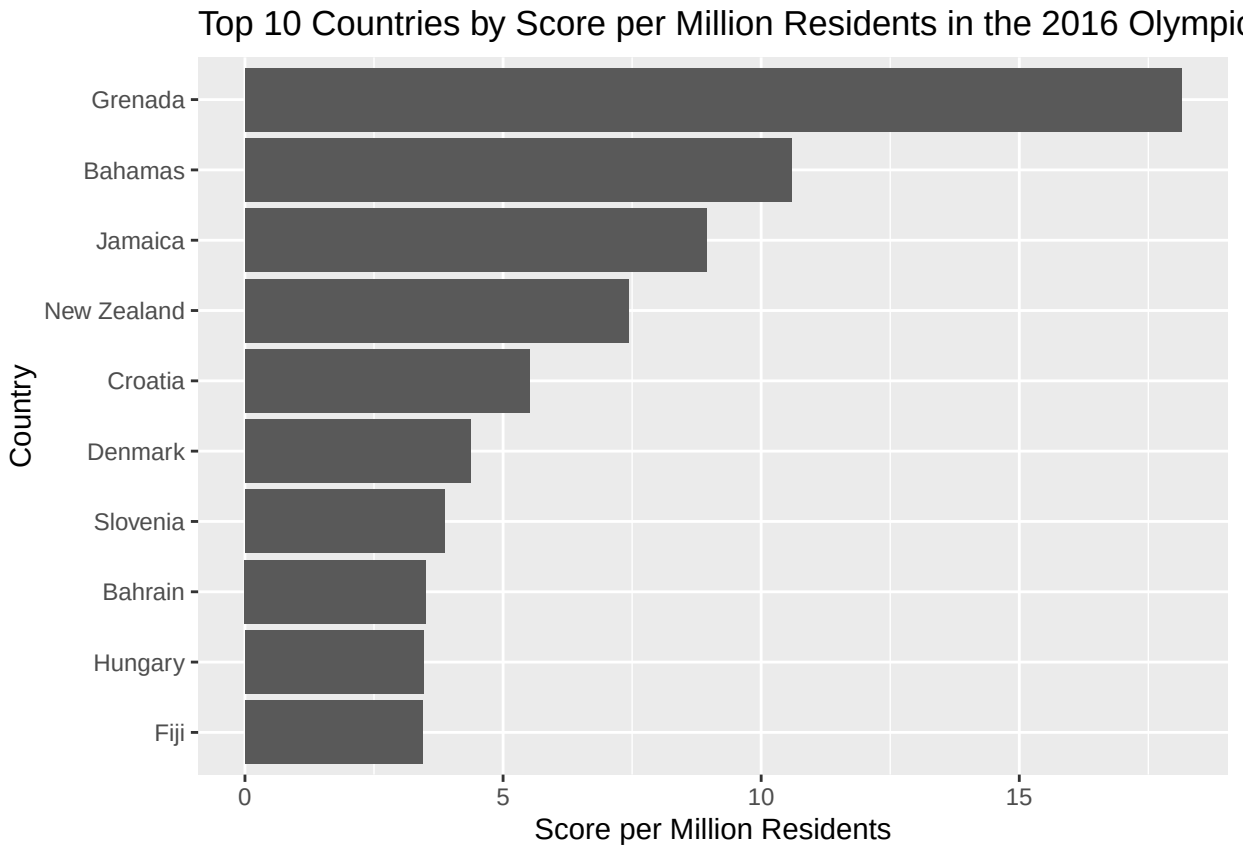
```
olympics2016 <- olympics2016 %>% mutate(score=gold*3+silver*2+bronze*1) #scoring system
olympics2016 <- olympics2016 %>% mutate(score_per_million=(score/population)) #score per million popula
olympics2016 <- olympics2016 %>% mutate(score_per_gdp=(score/gdp)*100) #score per $1 billion GDP
```

```
olympics2016score <- slice_max(filter(olympics2016, year == 2016),score, n=10) #Top 10 scorers in 2016
olympics2016scorepc <- slice_max(filter(olympics2016, year == 2016),score_per_million, n=10) #Top 10 sc
olympics2016scoregdp <- slice_max(filter(olympics2016, year == 2016),score_per_gdp, n=10) #Top 10 score
```

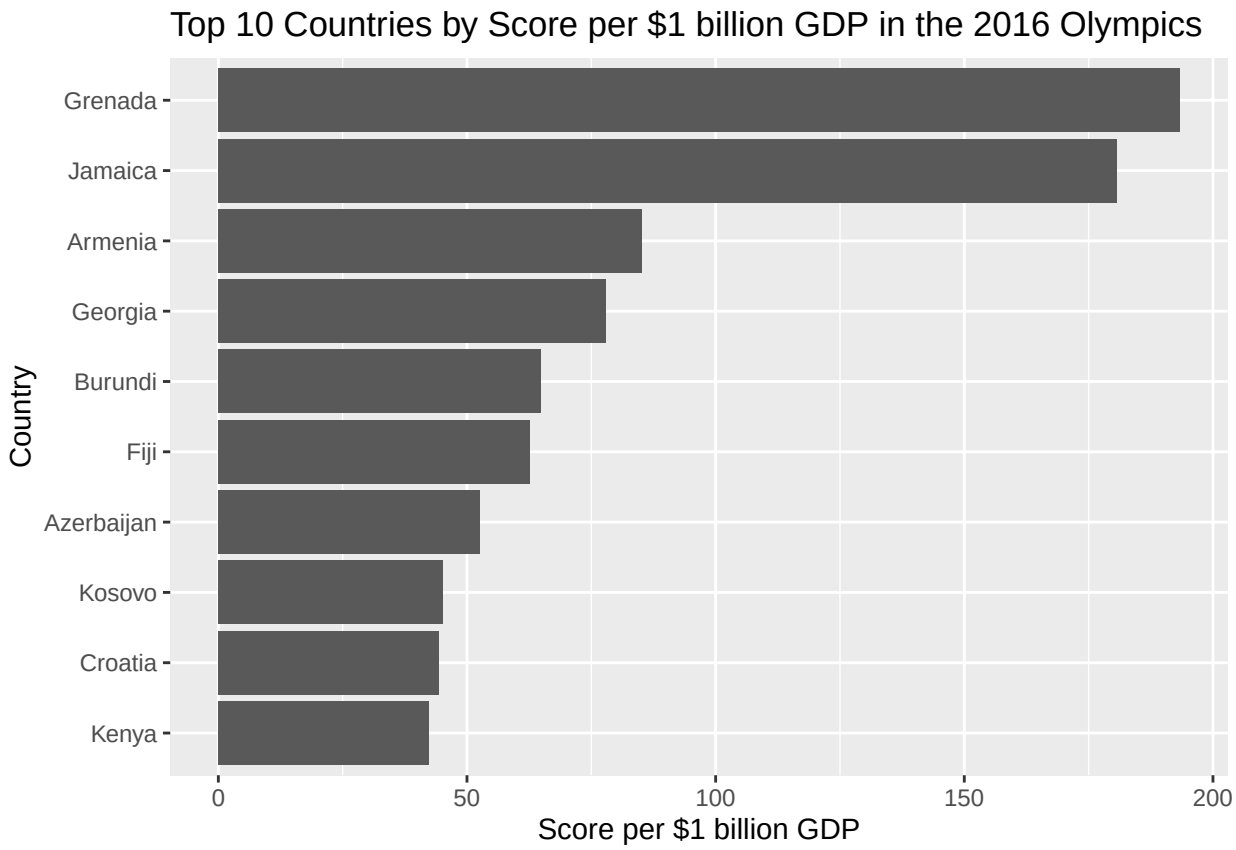
```
#Bar graphs

library(ggplot2)

ggplot(olympics2016score, aes(x = reorder(country, score), y = score)) + geom_bar(stat = "identity") + c
```



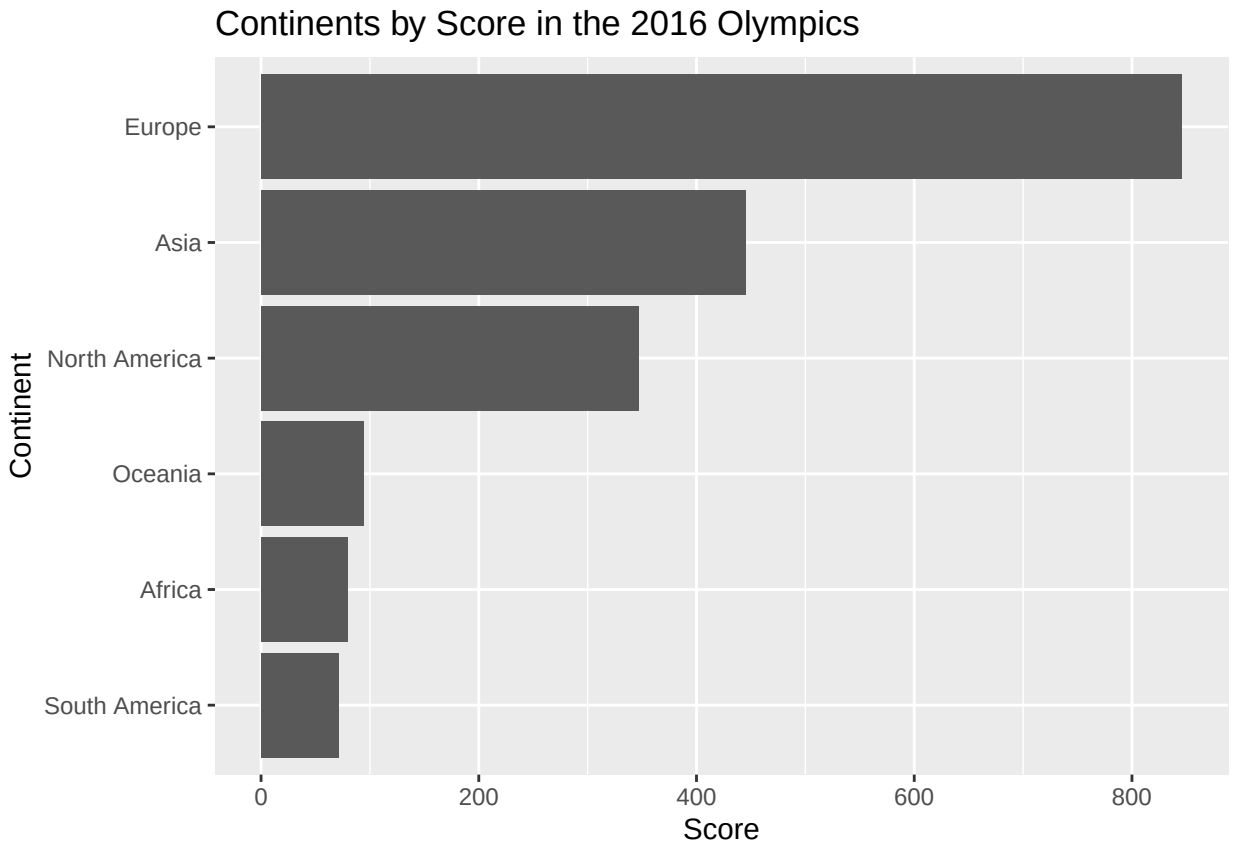

```
ggplot(olympics2016scorepc, aes(x = reorder(country, score_per_million), y = score_per_million)) + geom_
```



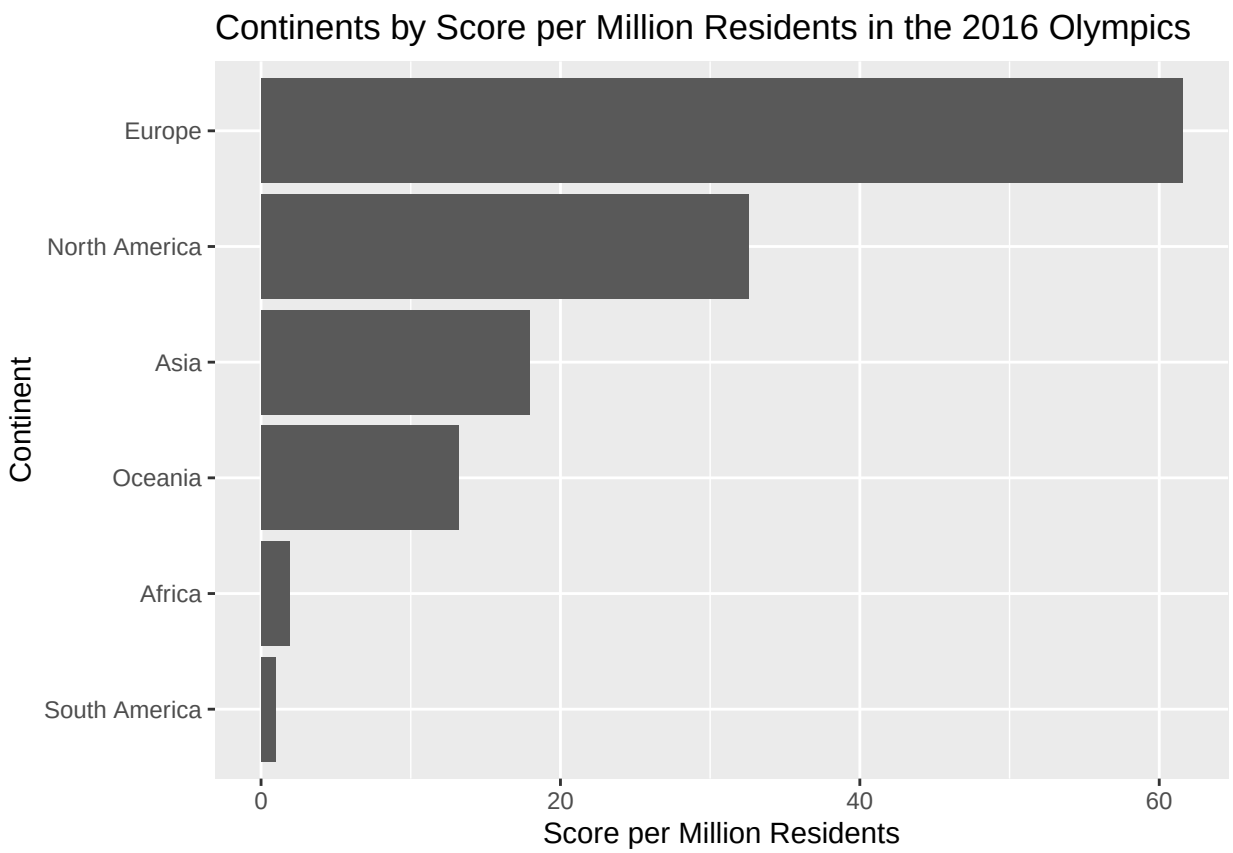

```
ggplot(olympics2016scoregdp, aes(x = reorder(country, score_per_gdp), y = score_per_gdp)) + coord_flip()
```



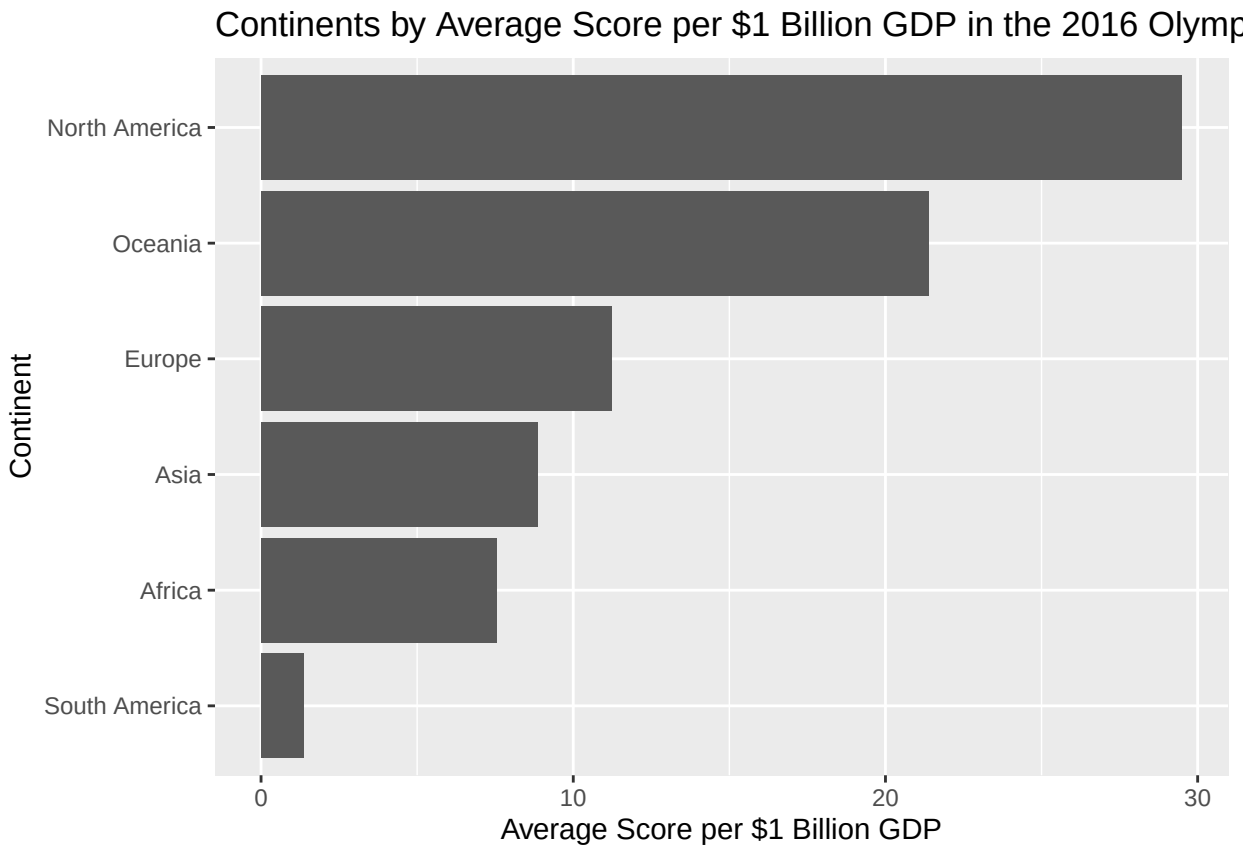

```r
#Defining continent metrics. I felt I should average instead of add GDP metric
continentscores <- olympics2016 %>% group_by(continent) %>% summarize(
  continentscore = sum(score, na.rm = TRUE),
  continentscorepc = sum(score_per_million, na.rm = TRUE),
  continentscoregdp = mean(score_per_gdp, na.rm = TRUE) )

#Continent bar graphs
ggplot(continentscores, aes(x = reorder(continent, continentscore), y = continentscore)) + coord_flip()
```




```
ggplot(continentscores, aes(x = reorder(continent, continentscorepc), y = continentscorepc)) + coord_fl
```

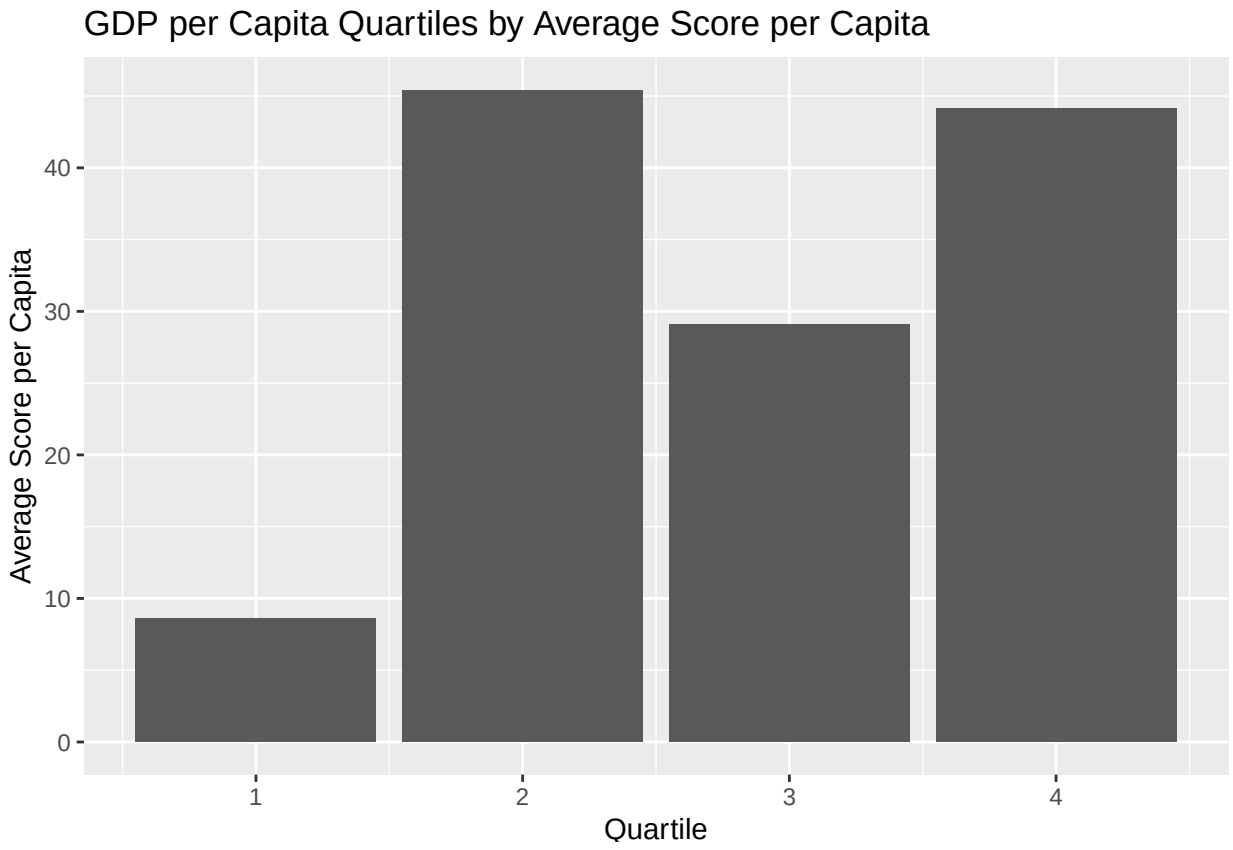



```
ggplot(continentscores, aes(x = reorder(continent, continentscoregdp), y = continentscoregdp)) + coord_
```




```
#Creating variables for GDP per capita and GDP per capita quartile
olympics2016 <- olympics2016 %>% mutate(gdp_percapita = gdp/population)
olympics2016 <- olympics2016 %>% mutate(gdp_percapita_quartile = ntile(gdp_percapita,4))

#Bar graph
ggplot(olympics2016, aes(x = gdp_percapita_quartile, y = score/population)) + geom_bar(stat = "identity
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```


GDP per Capita Quartiles by Average Score per Capita
Average Score per Capita
0
10
20
30
40
1
2
3
4
Quartile