MAS31004 - individual project

yuanyuan wei

2024-11-13

The analysis employs R packages like dplyr, ggplot2, and tidyr for data manipulation and visualization.

The dataset was cleaned to remove missing values, ensuring a reliable analysis of total medals, weighted scores, and economic indicators.

```
#import the packages
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(ggplot2)

library(tidyr)

library(carData)

library(car)  # For VIF calculation
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
#load the data
setwd("~/Desktop")
data <- read.csv("MedalsStatsData-220251110.csv")

#rename the columns
library(dplyr)

data <- data %>%

  rename(

    total_medals = total_medals,

    weighted_score = weighted_score,

    medals_per_capita = medals_per_capita,

    population = population,

    GDP = GDP..in.millions.,

    GDP_per_capita = GDP_per_capita

  )

print(colnames(data))
```

```
##  [1] "country_code"      "country"           "Gold.Medal"
##  [4] "Silver.Medal"      "Bronze.Medal"      "total_medals"
##  [7] "weighted_score"    "medals_per_capita" "X"
## [10] "female"            "male"              "population"
## [13] "X.1"               "GDP"               "ranking"
## [16] "GDP_per_capita"    "ranking.1"         "region"
```

```
#check the missing values
sapply(data[c("total_medals", "weighted_score", "medals_per_capita", "population", "G
DP", "GDP_per_capita")], function(x) sum(is.na(x)))
```

```
##      total_medals    weighted_score medals_per_capita        population
##                 1                 0                 1                 1
##               GDP    GDP_per_capita
##                 1                 1
```

```
#clean the data
data_clean <- na.omit(data[c("total_medals", "weighted_score", "medals_per_capita",
"population", "GDP", "GDP_per_capita")])
```

```
descriptive_stats <- data_clean %>%

  summarise(

    total_medals_mean = mean(total_medals, na.rm = TRUE),

    total_medals_median = median(total_medals, na.rm = TRUE),

    total_medals_range = paste(min(total_medals, na.rm = TRUE), max(total_medals, na.
rm = TRUE), sep = " to "),

    weighted_score_mean = mean(weighted_score, na.rm = TRUE),

    weighted_score_median = median(weighted_score, na.rm = TRUE),

    weighted_score_range = paste(min(weighted_score, na.rm = TRUE), max(weighted_scor
e, na.rm = TRUE), sep = " to "),

    medals_per_capita_mean = mean(medals_per_capita, na.rm = TRUE),

    medals_per_capita_median = median(medals_per_capita, na.rm = TRUE),

    medals_per_capita_range = paste(min(medals_per_capita, na.rm = TRUE), max(medals_
per_capita, na.rm = TRUE), sep = " to ")

  )

print(descriptive_stats)
```

```
##   total_medals_mean total_medals_median total_medals_range weighted_score_mean
## 1          11.45055                   5           1 to 126            22.30769
##   weighted_score_median weighted_score_range medals_per_capita_mean
## 1                     9             1 to 250               1.097906
##   medals_per_capita_median       medals_per_capita_range
## 1                0.3918811 0.003962537049 to 17.44880955
```

Correlation

The correlation matrix reveals important relationships between our variables:

- Strong Positive Correlations:
  - Total Medals and Weighted Score: r = 0.9965
  - Population and GDP: r = 0.8104
  - Total Medals and GDP: r = 0.7695
  - Weighted Score and GDP: r = 0.7846
- Moderate Positive Correlations:
  - Total Medals and Population: r = 0.3925
  - Total Medals and GDP per Capita: r = 0.2692
- Weak Positive Correlations:
  - Medals per Capita and Population: r = -0.1211
  - Medals per Capita and GDP: r = -0.1149
- Negligible Correlation:
  - Medals per Capita and GDP per Capita: r = -0.0076

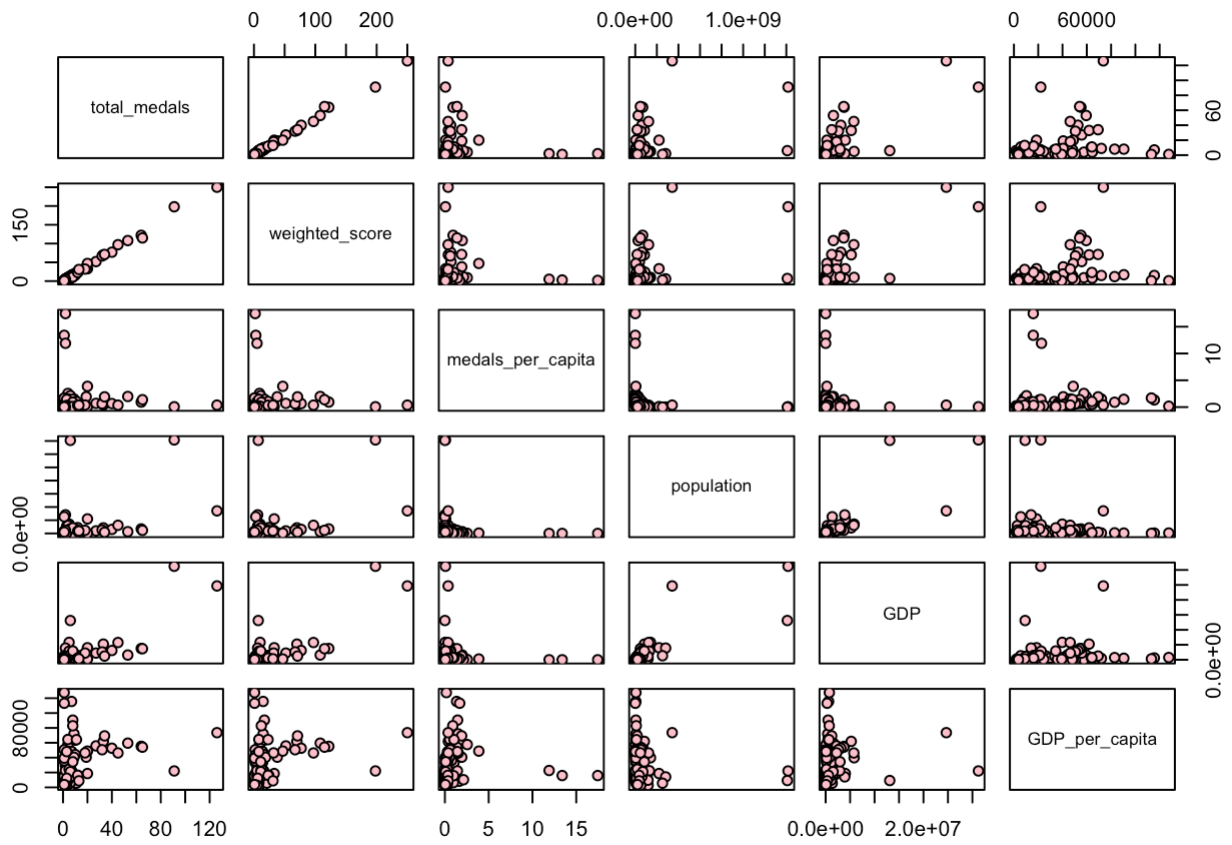
Moderate to weak correlations suggest that larger or wealthier countries do not always perform better on a per capita basis.

```
correlation_matrix <- cor(data_clean)

print(correlation_matrix)
```

```
##                  total_medals weighted_score medals_per_capita population
## total_medals        1.00000000     0.99651841      -0.058101766  0.3925006
## weighted_score      0.99651841     1.00000000      -0.054195713  0.4095094
## medals_per_capita  -0.05810177    -0.05419571       1.000000000 -0.1211274
## population          0.39250057     0.40950938      -0.121127366  1.0000000
## GDP                 0.76946153     0.78456088      -0.114912760  0.8103702
## GDP_per_capita      0.26920610     0.26451505      -0.007640651 -0.1312280
##                           GDP GDP_per_capita
## total_medals       0.76946153    0.269206103
## weighted_score     0.78456088    0.264515048
## medals_per_capita -0.11491276   -0.007640651
## population         0.81037020   -0.131228025
## GDP                1.00000000    0.079742142
## GDP_per_capita     0.07974214    1.000000000
```

Visual Analysis

```
pairs(data_clean, pch = 21, bg = c("pink"))
```



Regression

```
model_total_medals <- lm(total_medals ~ population + GDP + GDP_per_capita, data = dat
a_clean)

model_weighted_score <- lm(weighted_score ~ population + GDP + GDP_per_capita, data =
data)

model_medals_per_capita <- lm(medals_per_capita ~ GDP_per_capita, data = data)

summary(model_total_medals)
```

```
##
## Call:
## lm(formula = total_medals ~ population + GDP + GDP_per_capita,
##     data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.320  -4.308  -2.288   1.351  39.353
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.822e+00  1.811e+00   2.111   0.0377 *
## population     -5.808e-08  8.967e-09  -6.478 5.37e-09 ***
## GDP             5.692e-06  4.299e-07  13.240  < 2e-16 ***
## GDP_per_capita  6.461e-05  4.242e-05   1.523   0.1314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.93 on 87 degrees of freedom
## Multiple R-squared:  0.7541, Adjusted R-squared:  0.7456
## F-statistic: 88.95 on 3 and 87 DF,  p-value: < 2.2e-16
```

```
summary(model_weighted_score)
```

```
##
## Call:
## lm(formula = weighted_score ~ population + GDP + GDP_per_capita,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.793  -9.797  -4.633   2.654  78.589
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.608e+00  3.515e+00   1.880   0.0634 .
## population     -1.153e-07  1.747e-08  -6.596 3.04e-09 ***
## GDP             1.158e-05  8.378e-07  13.826  < 2e-16 ***
## GDP_per_capita  1.267e-04  8.264e-05   1.533   0.1288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.35 on 88 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7705, Adjusted R-squared:  0.7626
## F-statistic: 98.47 on 3 and 88 DF,  p-value: < 2.2e-16
```

```
summary(model_medals_per_capita)
```

```
##
## Call:
## lm(formula = medals_per_capita ~ GDP_per_capita, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1001 -0.9928 -0.6908 -0.0792 16.3513
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.108e+00  4.421e-01   2.505    0.014 *
## GDP_per_capita -6.356e-07  1.032e-05  -0.062    0.951
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.574 on 90 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  4.212e-05,  Adjusted R-squared:  -0.01107
## F-statistic: 0.003791 on 1 and 90 DF,  p-value: 0.951
```

```
#Check for multicollinearity

vif(model_total_medals)
```

```
##     population            GDP GDP_per_capita
##       3.281939       3.246062       1.133900
```

```
vif(model_weighted_score)
```

```
##     population            GDP GDP_per_capita
##       3.284332       3.249589       1.133881
```

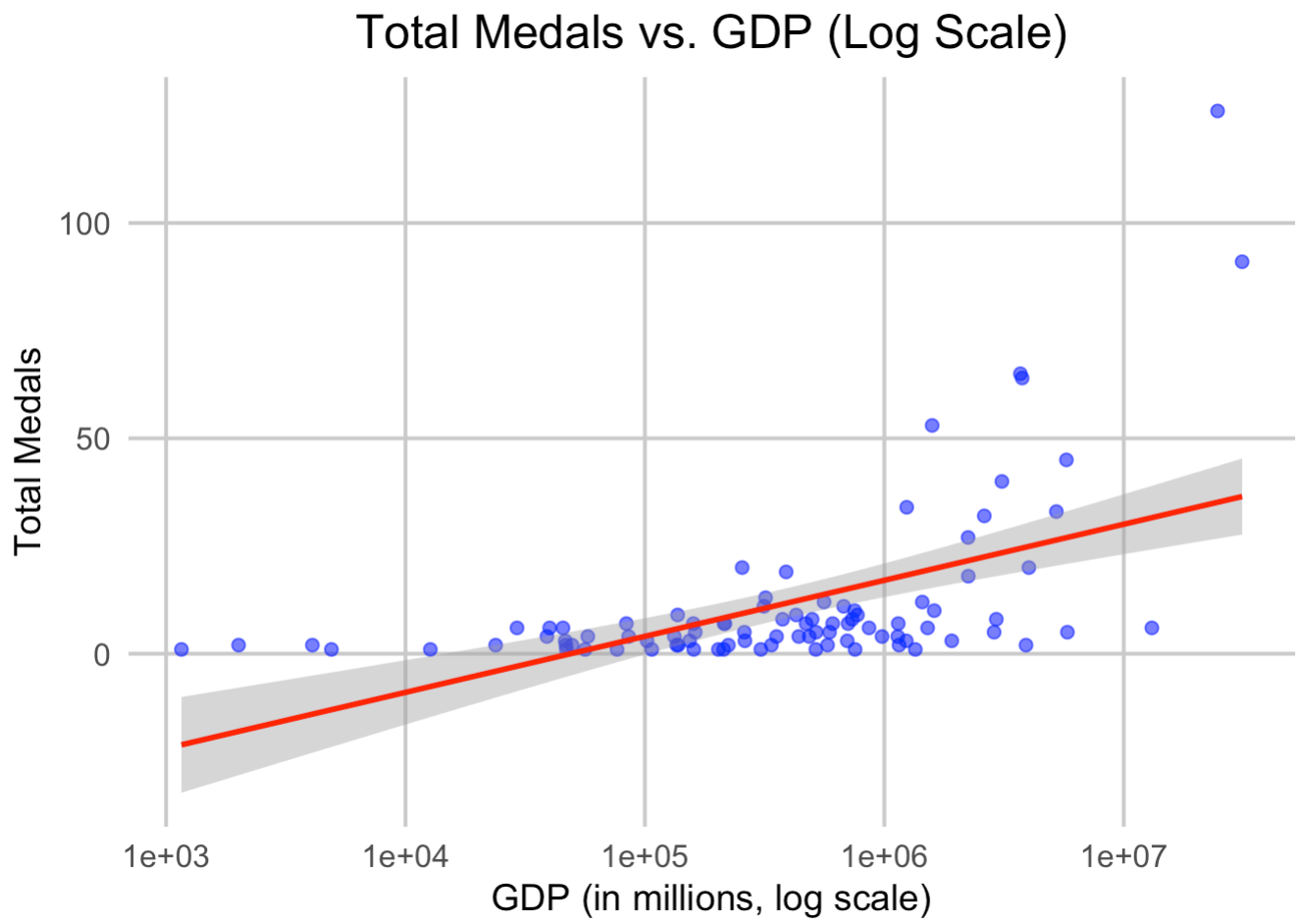
graphs

```
#total medals log scale graph
ggplot(data, aes(x = GDP, y = total_medals)) +

  geom_point(color = "blue", alpha = 0.6, size = 2) +  # Increase point size

  geom_smooth(method = "lm", color = "red", se = TRUE) +  # Add confidence interval

  scale_x_log10() +  # Log scale for GDP

  labs(title = "Total Medals vs. GDP (Log Scale)",

       x = "GDP (in millions, log scale)",

       y = "Total Medals") +

  theme_minimal(base_size = 15) +  # Increase base size for readability

  theme(plot.title = element_text(hjust = 0.5),  # Center title

        panel.grid.major = element_line(color = "grey80"),  # Light grid lines

        panel.grid.minor = element_blank(),  # Remove minor grid lines

        axis.text = element_text(size = 12),  # Axis text size

        axis.title = element_text(size = 14))  # Axis title size
```

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

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale rang
e
## (`geom_point()`).
```

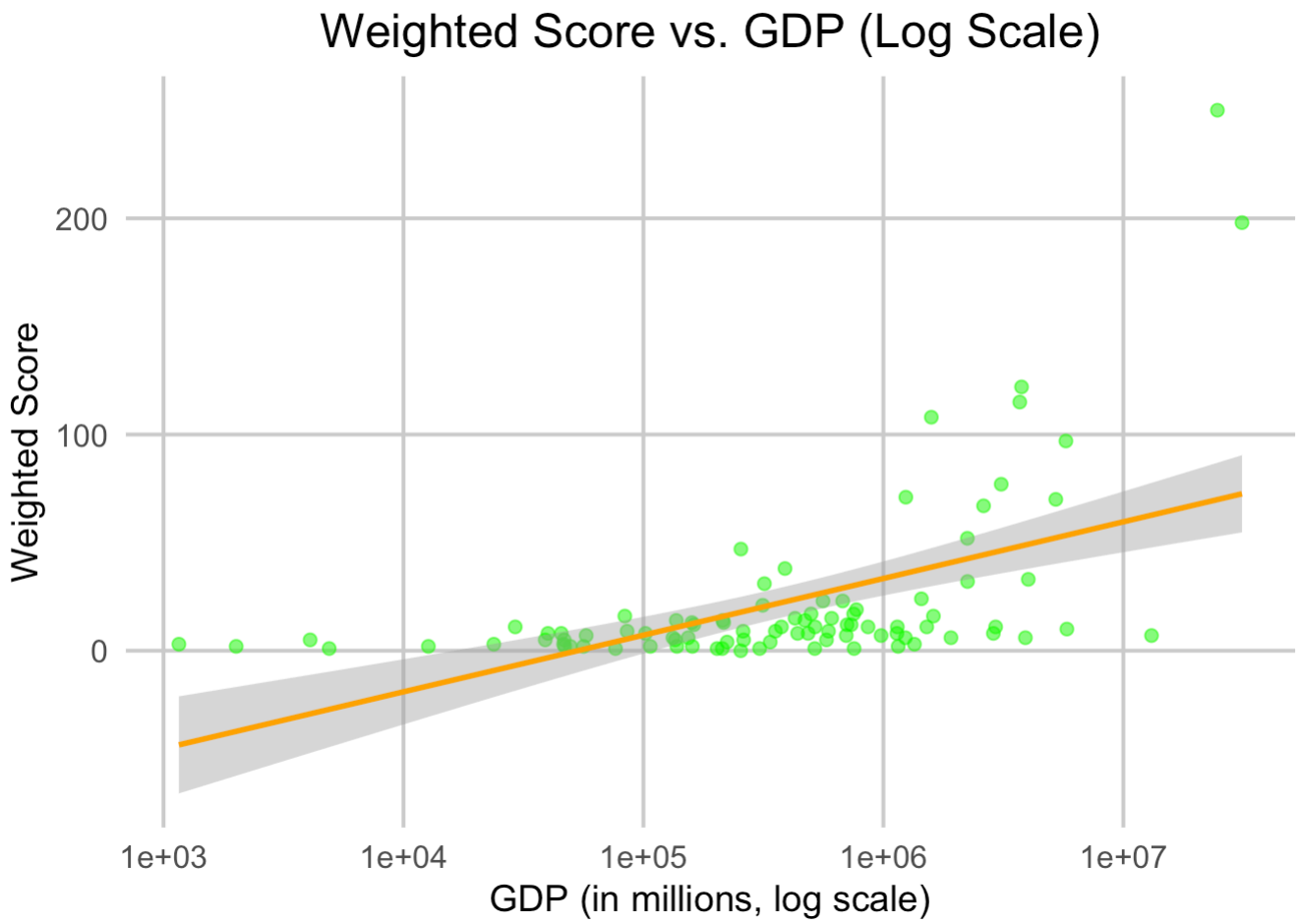



```
# Create the Weighted Score vs. GDP graph with enhancements

ggplot(data, aes(x = GDP, y = weighted_score)) +

  geom_point(color = "green", alpha = 0.6, size = 2) +  # Increase point size

  geom_smooth(method = "lm", color = "orange", se = TRUE) +  # Add confidence interva
l

  scale_x_log10() +  # Log scale for GDP

  labs(title = "Weighted Score vs. GDP (Log Scale)",

       x = "GDP (in millions, log scale)",

       y = "Weighted Score") +

  theme_minimal(base_size = 15) +  # Increase base size for readability

  theme(plot.title = element_text(hjust = 0.5),  # Center title

        panel.grid.major = element_line(color = "grey80"),  # Light grid lines

        panel.grid.minor = element_blank(),  # Remove minor grid lines

        axis.text = element_text(size = 12),  # Axis text size

        axis.title = element_text(size = 14))  # Axis title size
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

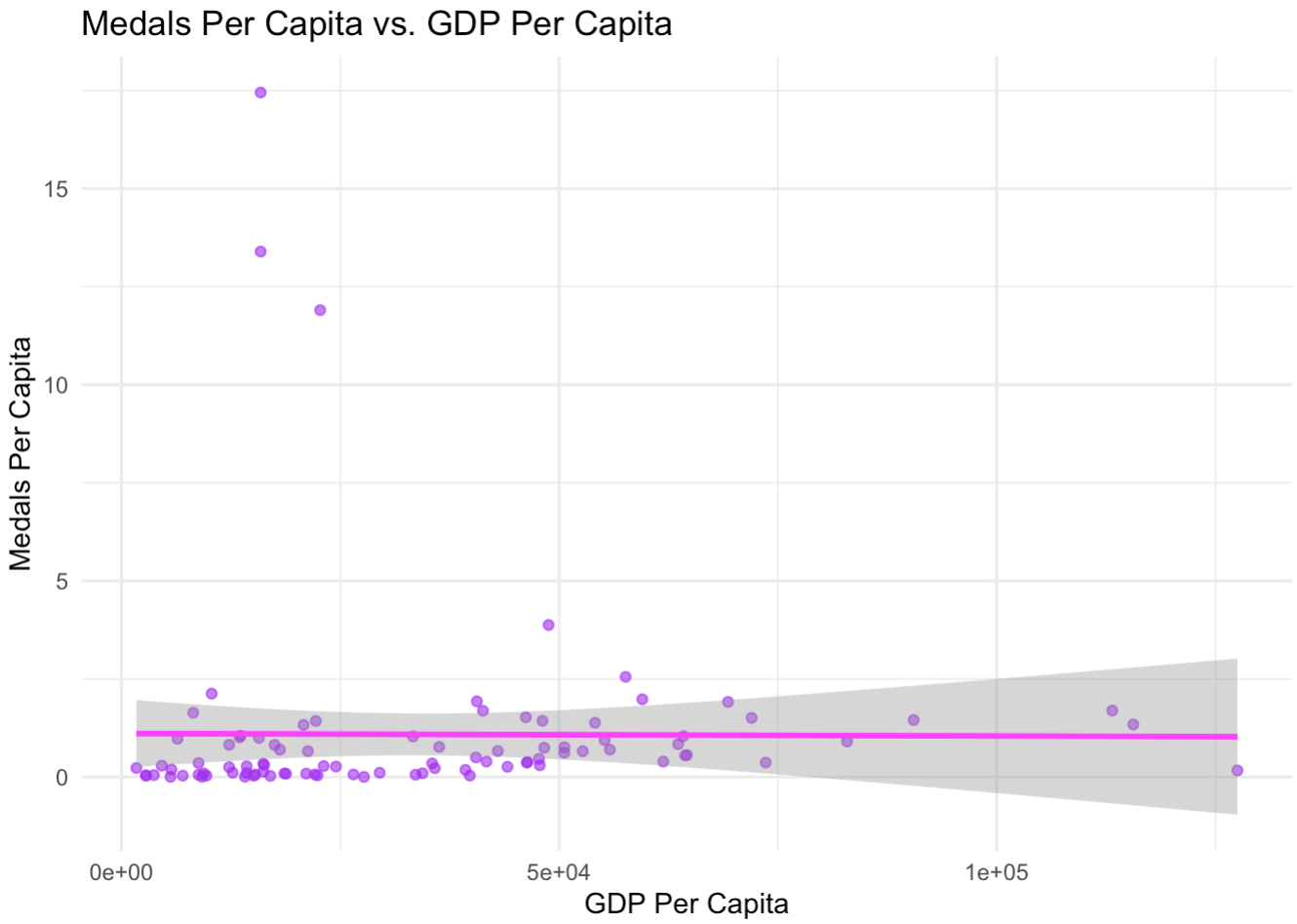



```
#medals per capita vs GDP per capita
ggplot(data, aes(x = GDP_per_capita, y = medals_per_capita)) +

  geom_point(color = "purple", alpha = 0.6) +

  geom_smooth(method = "lm", color = "magenta") +

  labs(title = "Medals Per Capita vs. GDP Per Capita",

       x = "GDP Per Capita",

       y = "Medals Per Capita") +

  theme_minimal()
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Regional Analysis

```
library(dplyr)
regional_stats <- data %>%

  group_by(region) %>%

  summarise(

    avg_total_medals = mean(total_medals, na.rm = TRUE),

    avg_weighted_score = mean(weighted_score, na.rm = TRUE),

    avg_medals_per_capita = mean(medals_per_capita, na.rm = TRUE)

  )

print(regional_stats)
```

```
## # A tibble: 12 × 4
##    region              avg_total_medals avg_weighted_score avg_medals_per_capita
##    <chr>                          <dbl>              <dbl>                 <dbl>
##  1 ""                              1                  1                NaN
##  2 "Africa"                        3.25               6.42               0.274
##  3 "Australia and Oce…            24.7               52.3                2.30
##  4 "Central America a…             3.11               5.33               5.22
##  5 "Central Asia"                  6.8               12.8                0.400
##  6 "East and Southeas…            16.8               34.9                0.231
##  7 "Europe"                       13.8               26.2                0.917
##  8 "Europe "                     NaN                  0                  0
##  9 "Middle East"                   5.67              10.8                0.969
## 10 "North America"                52.7              103.                 0.368
## 11 "South America"                 5.83              10.2                0.108
## 12 "South Asia"                    3.5                5                  0.00411
```

```
#ANOVA for regional differences
anova_total_medals <- aov(total_medals ~ region, data = data)
summary(anova_total_medals)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## region      10   8394   839.4   2.556 0.00979 **
## Residuals   81  26603   328.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

```
avg_weighted_score <- aov(weighted_score ~ region, data = data)
summary(avg_weighted_score)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## region      11  34457    3132   2.316 0.0158 *
## Residuals   81 109556    1352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

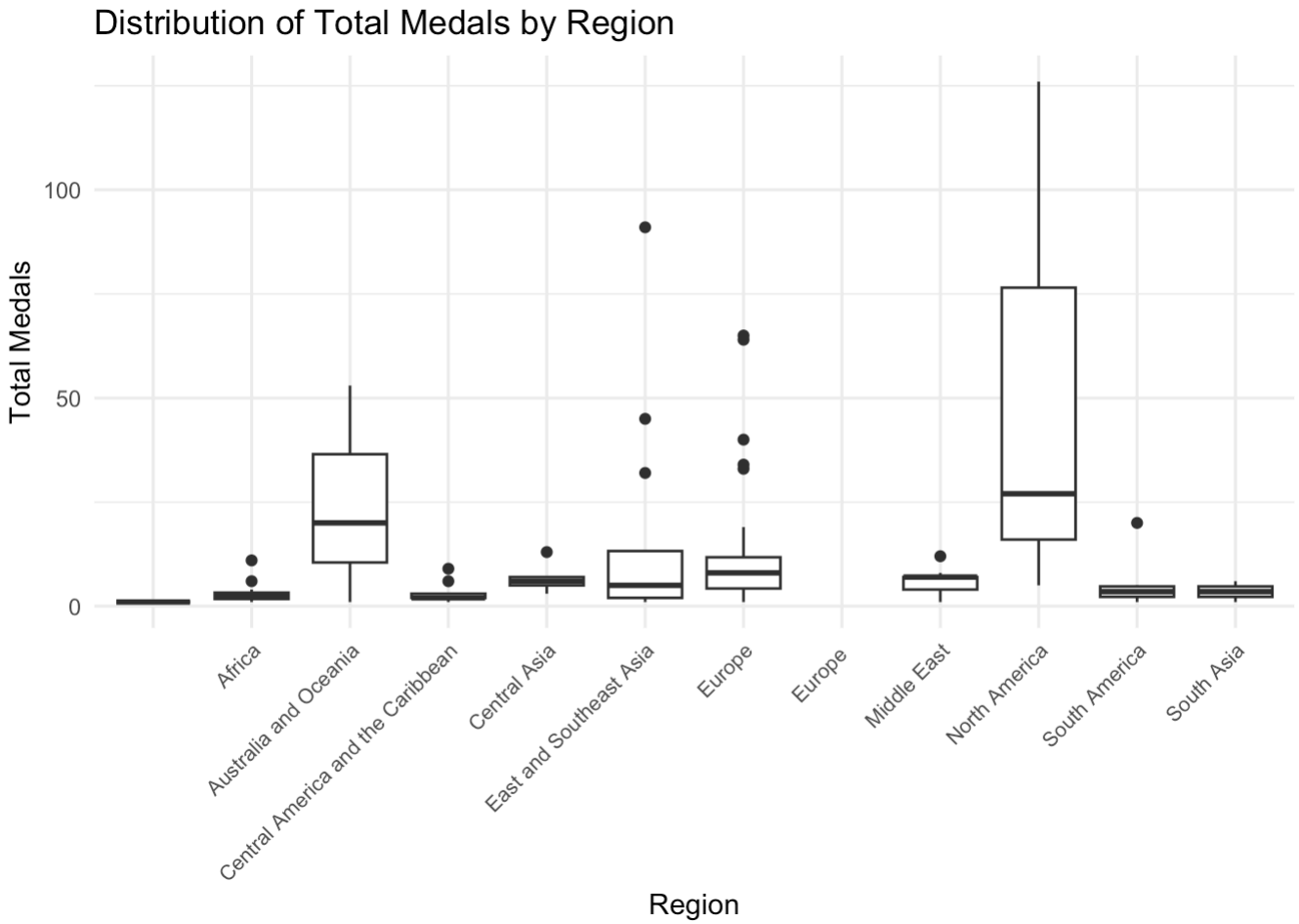
```
avg_medals_per_capita <- aov(medals_per_capita ~ region, data = data)
summary(avg_medals_per_capita)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## region      10  188.9   18.89   3.756 0.000354 ***
## Residuals   81  407.4    5.03
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

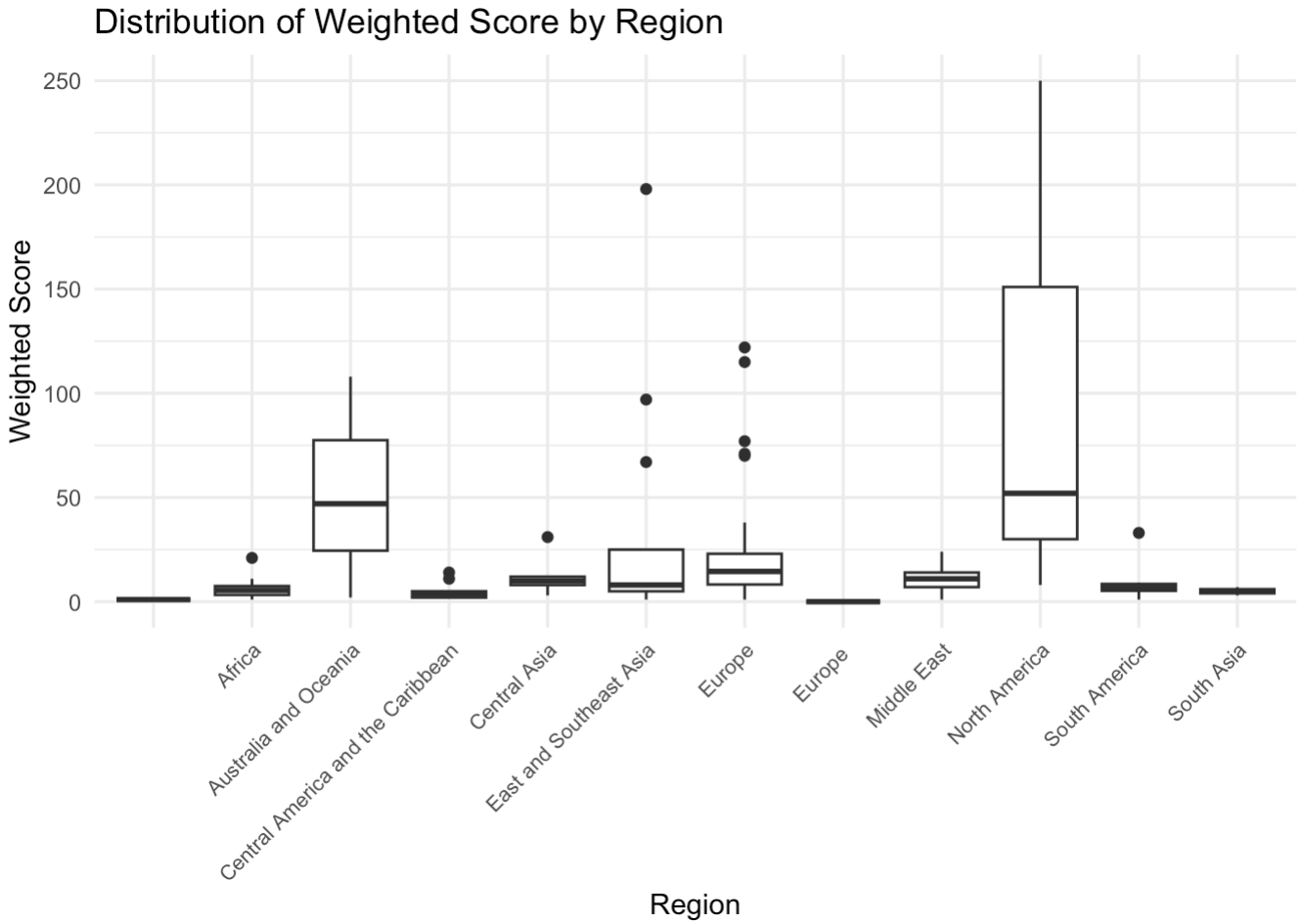
Visualisation

```
# Visualization: Box plots for regional comparison

library(ggplot2)

ggplot(data, aes(x = region, y = total_medals)) +

  geom_boxplot() +

  theme_minimal() +

  labs(title = "Distribution of Total Medals by Region", x = "Region", y = "Total Med
als") +

  theme(

    axis.text.x = element_text(angle = 45, hjust = 1, size = 8)  # Adjust size and an
gle of x-axis text

  )
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

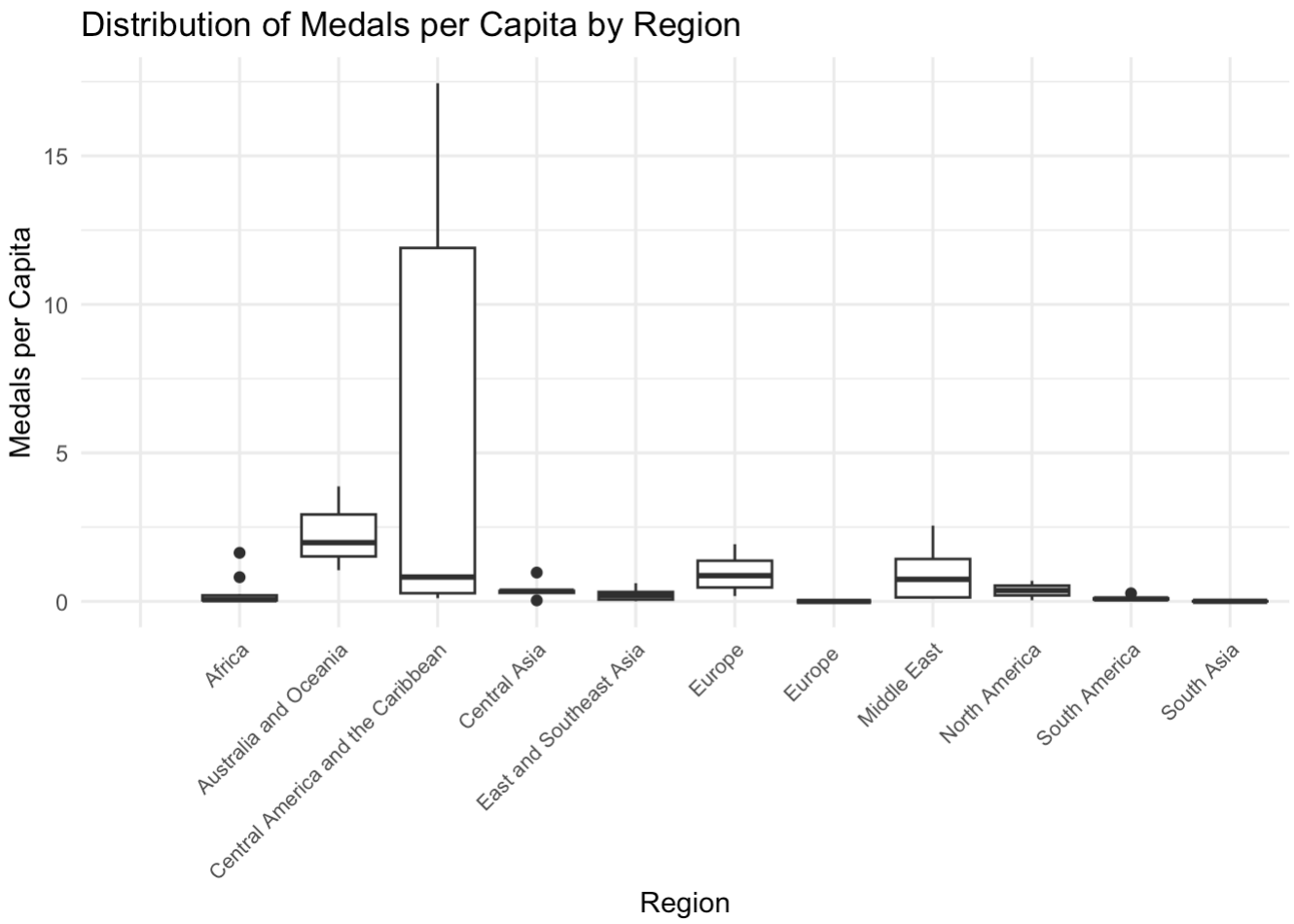


```
# Visualization: Box plots for regional comparison of Weighted Scores
library(ggplot2)

ggplot(data, aes(x = region, y = weighted_score)) +

  geom_boxplot() +

  theme_minimal() +

  labs(title = "Distribution of Weighted Score by Region", x = "Region", y = "Weighte
d Score") +

  theme(

    axis.text.x = element_text(angle = 45, hjust = 1, size = 8)  # Adjust size and an
gle of x-axis text

  )
```



```
# Visualization: Box plots for regional comparison of Medals per Capita
library(ggplot2)

ggplot(data, aes(x = region, y = medals_per_capita)) +

  geom_boxplot() +

  theme_minimal() +

  labs(title = "Distribution of Medals per Capita by Region", x = "Region", y = "Meda
ls per Capita") +

  theme(

    axis.text.x = element_text(angle = 45, hjust = 1, size = 8)  # Adjust size and an
gle of x-axis text

  )
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



efficiency analysis

We calculated an efficiency index to identify countries that perform exceptionally well relative to their economic resources:

```
#Efficiency Analysis

data <- data %>%

  mutate(efficiency_index = weighted_score / (GDP / 1e9))

top_efficient_countries <- data %>%

  arrange(desc(efficiency_index)) %>%

  select(country, efficiency_index) %>%

  head(10)

print(top_efficient_countries)
```

```
##        country efficiency_index
## 1     Dominica        2588438.3
## 2  Saint Lucia        1224589.8
## 3      Grenada         996015.9
## 4      Jamaica         376390.1
## 5   Cabo Verde         203956.8
## 6    DPR Korea         200000.0
## 7      Georgia         191259.4
## 8  New Zealand         184480.1
## 9   Kyrgyzstan         175975.0
## 10        Fiji         157492.7
```