stat630_project_group5

Group5

2022-12-05

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
```

```
# import data
olympic <- read_csv("winter_olympic_study.csv")
```

```
## Rows: 175 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (4): host_country, host_city, country_name, country_code
## dbl (8): year, Gold, Silver, Bronze, GDP, gdp_per_capita, Atheletes, Total_P...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(data)
```

```
## NULL
```

```
head(olympic)
```

```
## # A tibble: 6 × 12
##    year host_country host_…¹ count…² count…³  Gold Silver Bronze     GDP gdp_p…⁴
##   <dbl> <chr>        <chr>   <chr>   <chr>   <dbl>  <dbl>  <dbl>   <dbl>   <dbl>
## 1  1984 Yugoslavia   Saraje… Austria AUT         0      0      1 6.80e10   8991.
## 2  1984 Yugoslavia   Saraje… Canada  CAN         2      1      1 3.55e11  13878.
## 3  1984 Yugoslavia   Saraje… Finland FIN         4      3      6 5.29e10  10834.
## 4  1984 Yugoslavia   Saraje… France  FRA         0      1      2 5.31e11   9420.
## 5  1984 Yugoslavia   Saraje… Great … GBR         1      0      0 4.61e11   8179.
## 6  1984 Yugoslavia   Saraje… Italy   ITA         2      0      0 4.38e11   7740.
## # … with 2 more variables: Atheletes <dbl>, Total_Points <dbl>, and abbreviated
## #   variable names ¹host_city, ²country_name, ³country_code, ⁴gdp_per_capita
```

```
sum(is.na(olympic))
```

```
## [1] 0
```

EDA

```
summary(olympic)
```

```
##       year      host_country        host_city         country_name
##  Min.   :1984   Length:175         Length:175         Length:175
##  1st Qu.:1994   Class :character   Class :character   Class :character
##  Median :2002   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2003
##  3rd Qu.:2012
##  Max.   :2018
##  country_code            Gold            Silver           Bronze
##  Length:175         Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  Class :character   1st Qu.: 0.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Mode  :character   Median : 2.000   Median : 2.000   Median : 2.000
##                     Mean   : 2.891   Mean   : 2.857   Mean   : 2.977
##                     3rd Qu.: 4.000   3rd Qu.: 4.500   3rd Qu.: 4.500
##                     Max.   :14.000   Max.   :15.000   Max.   :13.000
##       GDP            gdp_per_capita       Atheletes       Total_Points
##  Min.   :5.026e+08   Min.   :   366.5   Min.   :  1.00   Min.   :  3.00
##  1st Qu.:1.511e+11   1st Qu.: 10906.4   1st Qu.: 45.50   1st Qu.:  9.00
##  Median :4.420e+11   Median : 23087.2   Median : 73.00   Median : 25.00
##  Mean   :1.625e+12   Mean   : 26359.2   Mean   : 81.06   Mean   : 37.71
##  3rd Qu.:1.499e+12   3rd Qu.: 38249.3   3rd Qu.:107.00   3rd Qu.: 55.50
##  Max.   :2.061e+13   Max.   :180366.7   Max.   :241.00   Max.   :173.00
```

```
mean(olympic$Total_Points)
```

```
## [1] 37.70857
```

```
sd(olympic$Total_Points)
```

```
## [1] 36.76143
```

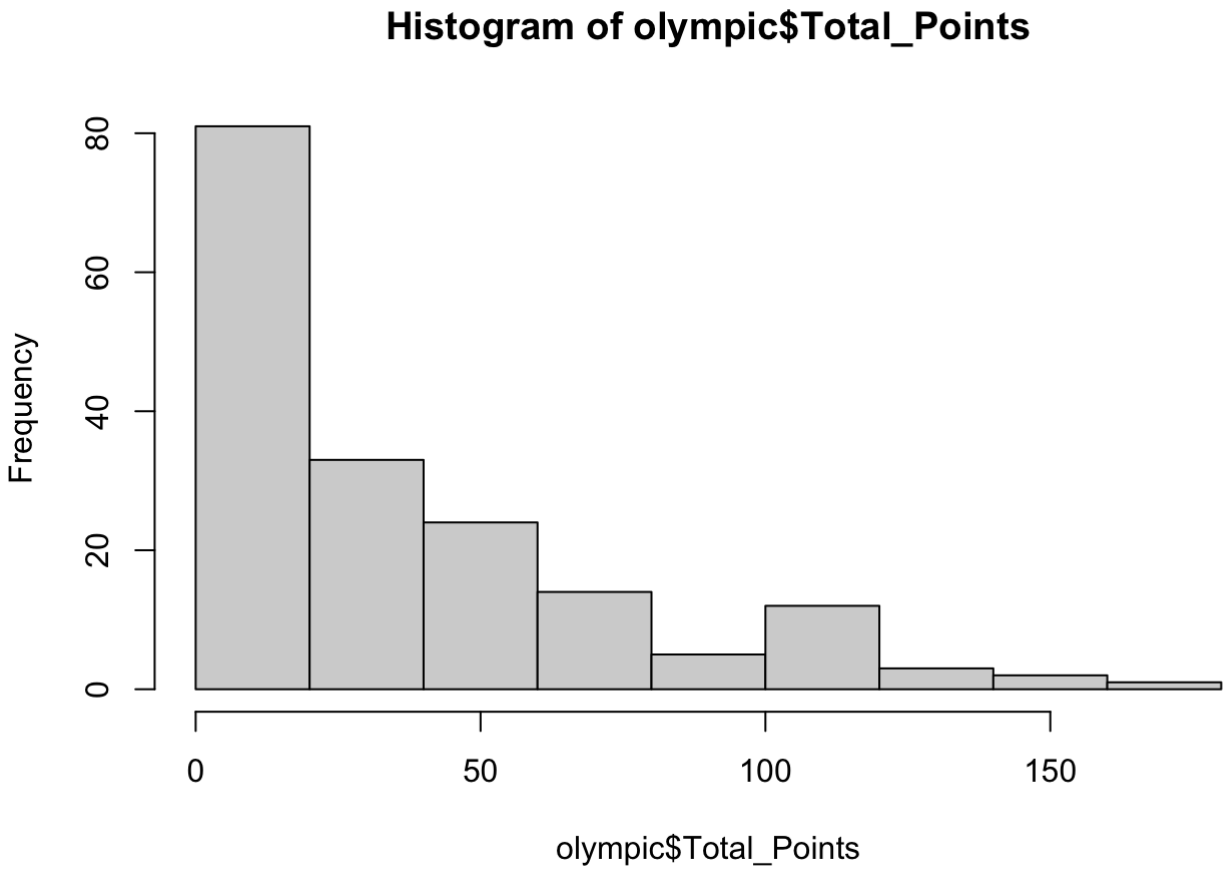
```
mean(olympic$GDP)
```

```
## [1] 1.625446e+12
```

```
sd(olympic$GDP)
```

```
## [1] 3.136443e+12
```

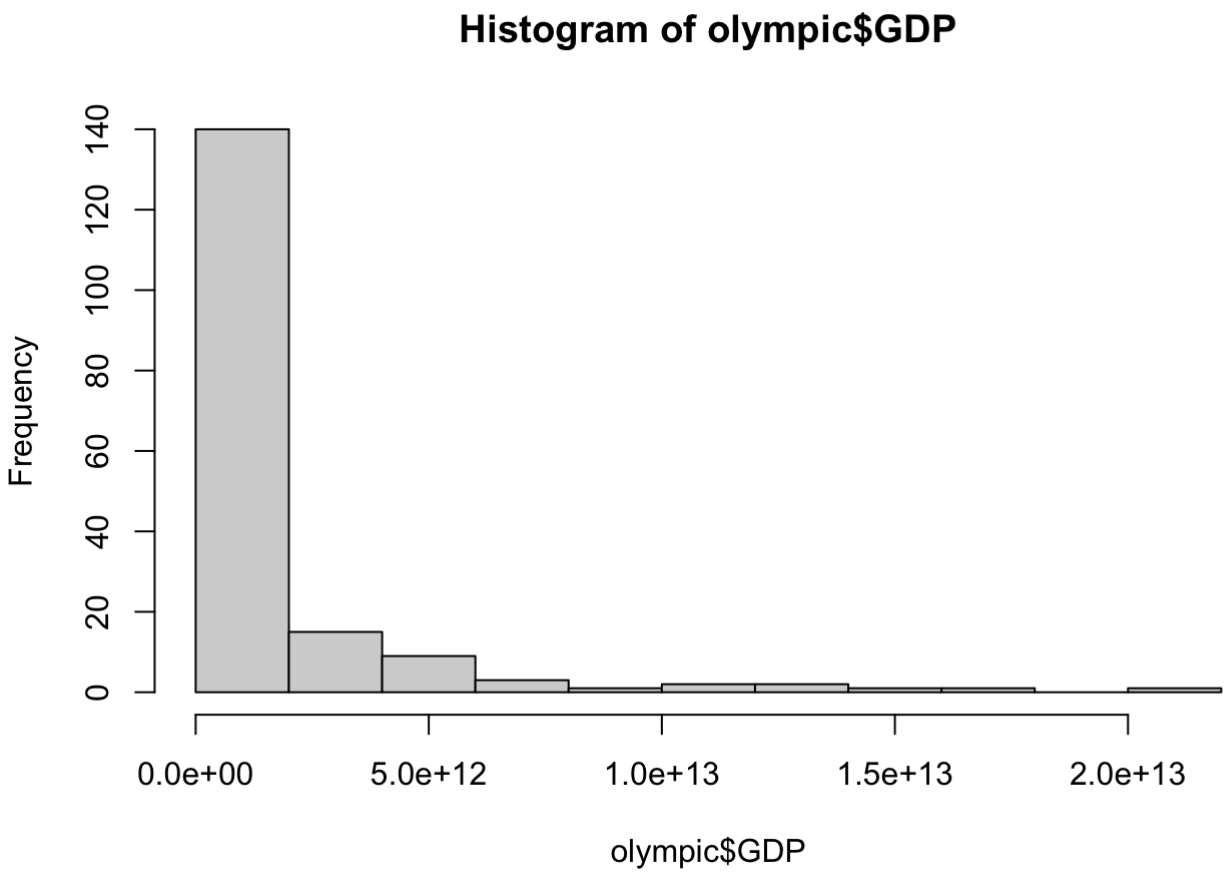
```
hist(olympic$Total_Points)
```




```
olympic %>%
  filter(year == 2018) %>%
  select(country_name, Total_Points) %>%
  arrange(desc(Total_Points)) %>%
  head(10)
```

```
## # A tibble: 10 × 2
##    country_name  Total_Points
##    <chr>                <dbl>
##  1 Norway                 173
##  2 Canada                 128
##  3 United States          104
##  4 South Korea             74
##  5 Sweden                  69
##  6 France                  64
##  7 Austria                 60
##  8 Japan                   56
##  9 Italy                   41
## 10 China                   36
```

```
hist(olympic$GDP)
```



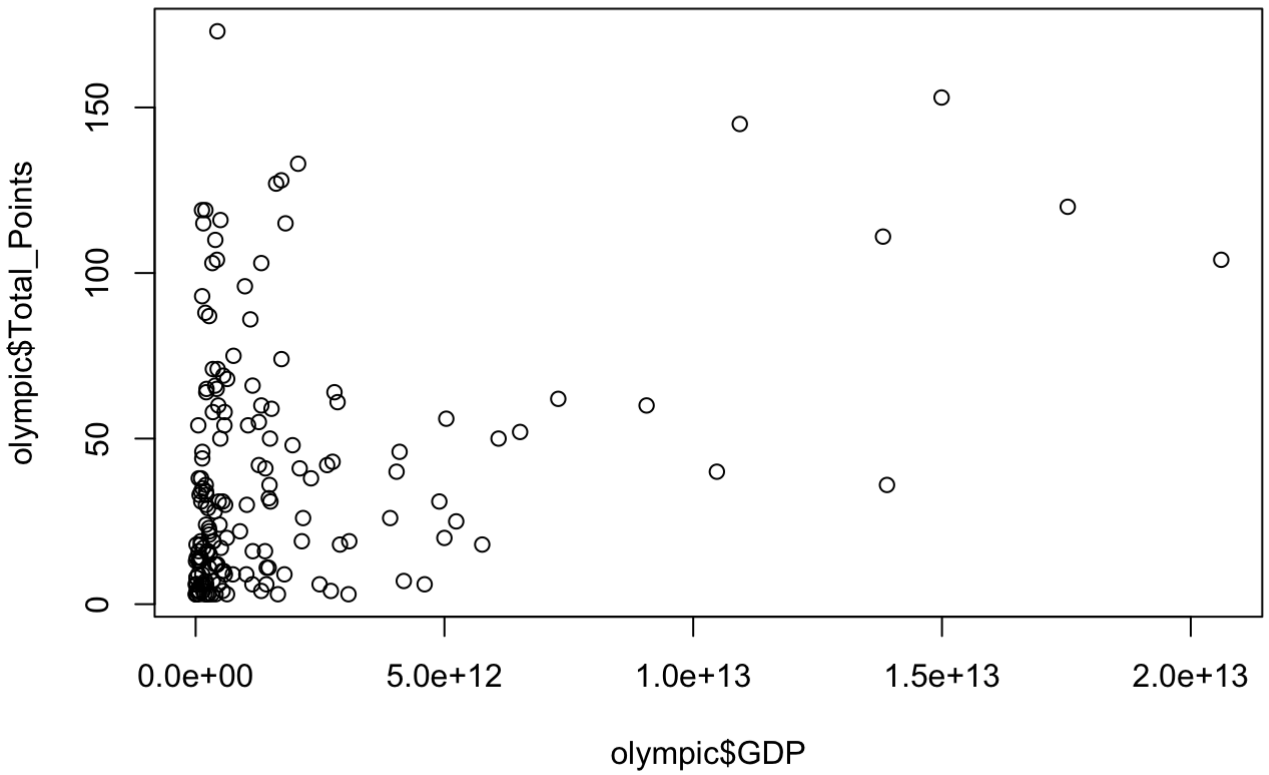

```
olympic %>%
  filter(year == 2018) %>%
  select(country_name, GDP) %>%
  arrange(desc(GDP)) %>%
  head(10)
```

```
## # A tibble: 10 × 2
##    country_name      GDP
##    <chr>           <dbl>
##  1 United States 2.06e13
##  2 China         1.39e13
##  3 Japan         5.04e12
##  4 Great Britain 2.90e12
##  5 France        2.79e12
##  6 Italy         2.09e12
##  7 South Korea   1.72e12
##  8 Canada        1.72e12
##  9 Australia     1.43e12
## 10 Spain         1.42e12
```

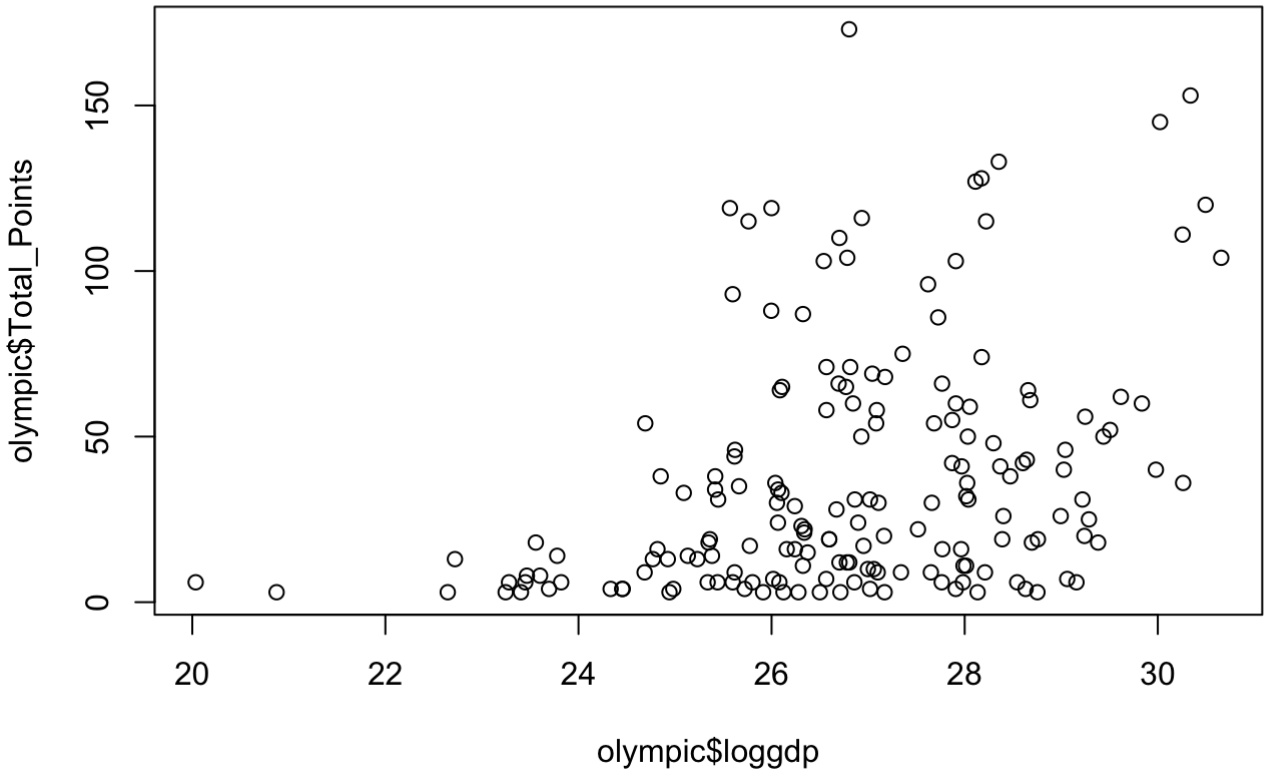
Transformation

```
olympic$loggdp <- log(olympic$GDP)
```

```
plot(olympic$Total_Points ~ olympic$GDP)
```




```
plot(olympic$Total_Points ~ olympic$loggdp)
```




```
mean(olympic$loggdp)
```

```
## [1] 26.81276
```

```
sd(olympic$loggdp)
```

```
## [1] 1.823715
```

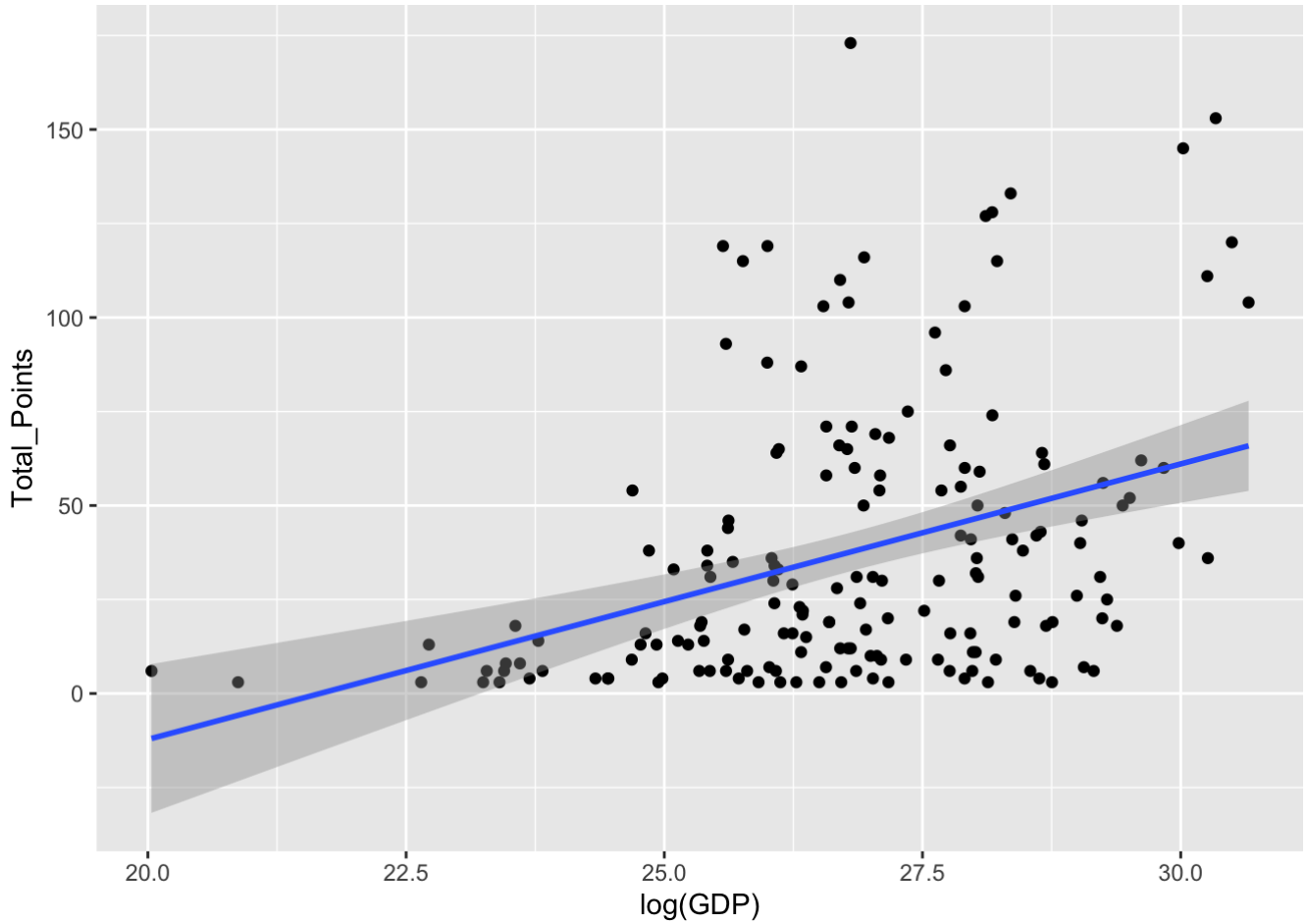
Analysis 1

Check assumptions

1. Linearity

```
# 1. Linearity
ggplot(olympic, aes(x = log(GDP), y = Total_Points)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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



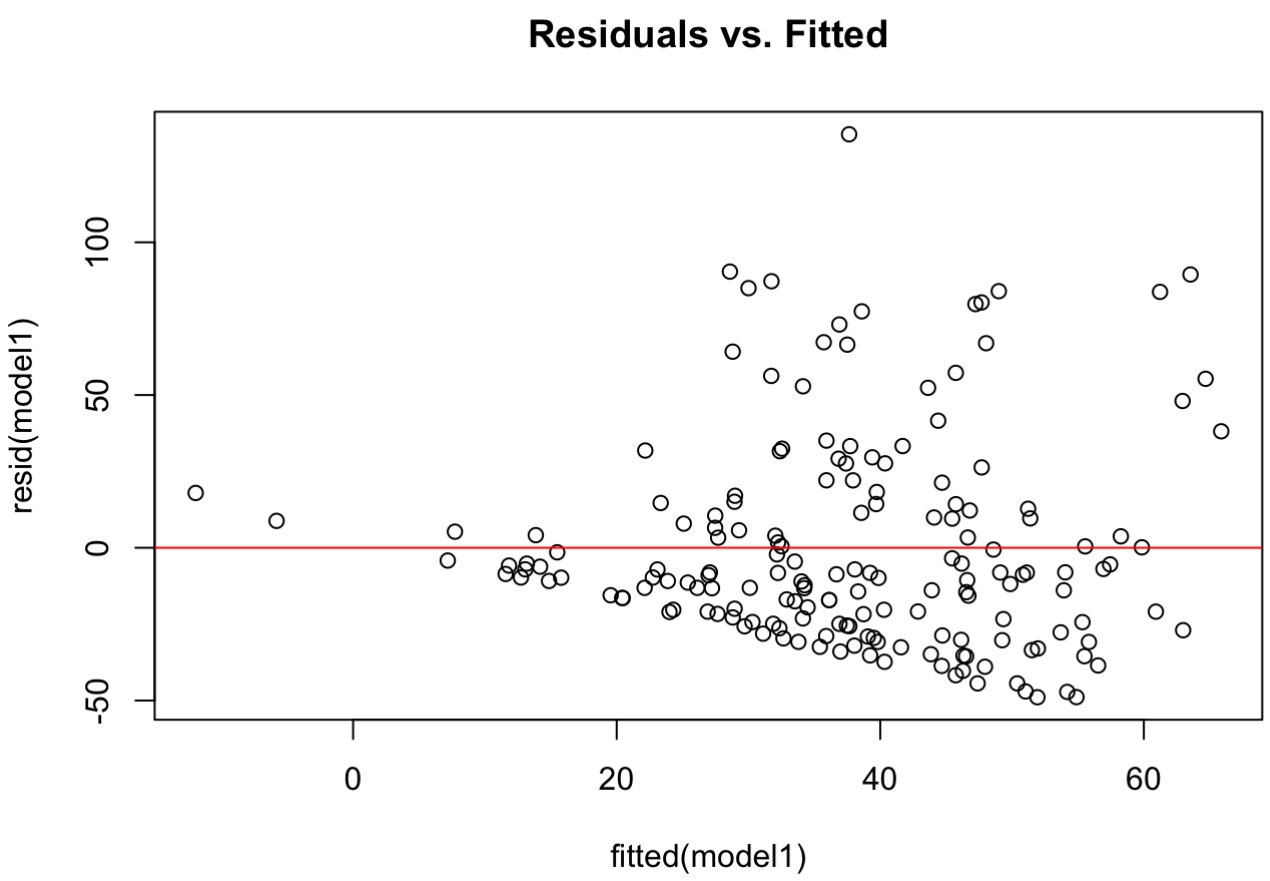
2. Indepence

Each country is independent of one another.We assume that all the country's GDP and number of medals are independent.

3. Equal Variance

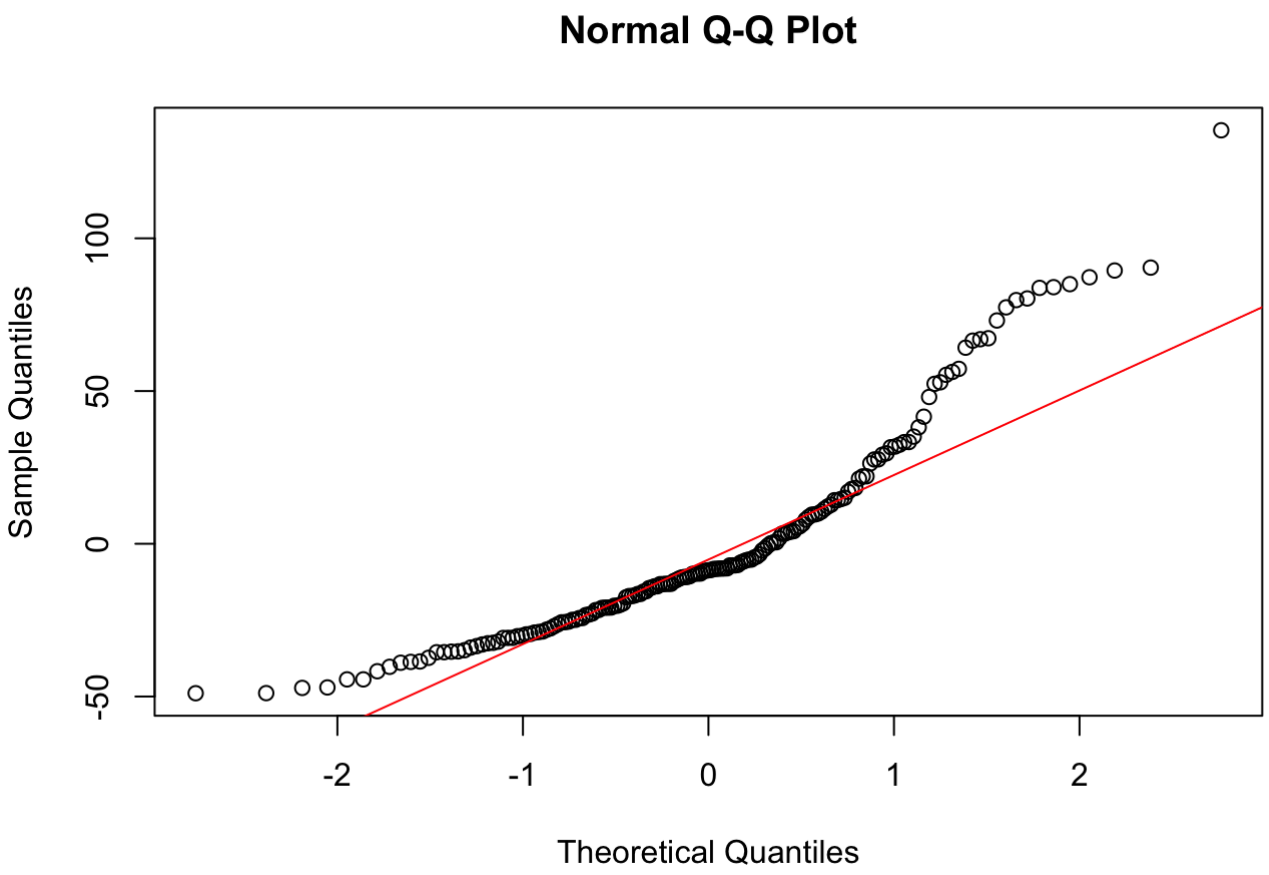
```
model1 <- lm(Total_Points ~ log(GDP), data = olympic)

plot(resid(model1) ~ fitted(model1), main = "Residuals vs. Fitted")
abline(h = 0, col = "red")
```



4. Normality

```
qqnorm(resid(model1))
qqline(resid(model1), col = "red")
```



Linear Regression

```
model1
```

```
##
## Call:
## lm(formula = Total_Points ~ log(GDP), data = olympic)
##
## Coefficients:
## (Intercept)     log(GDP)
##    -158.733        7.326
```

```
summary(model1)
```

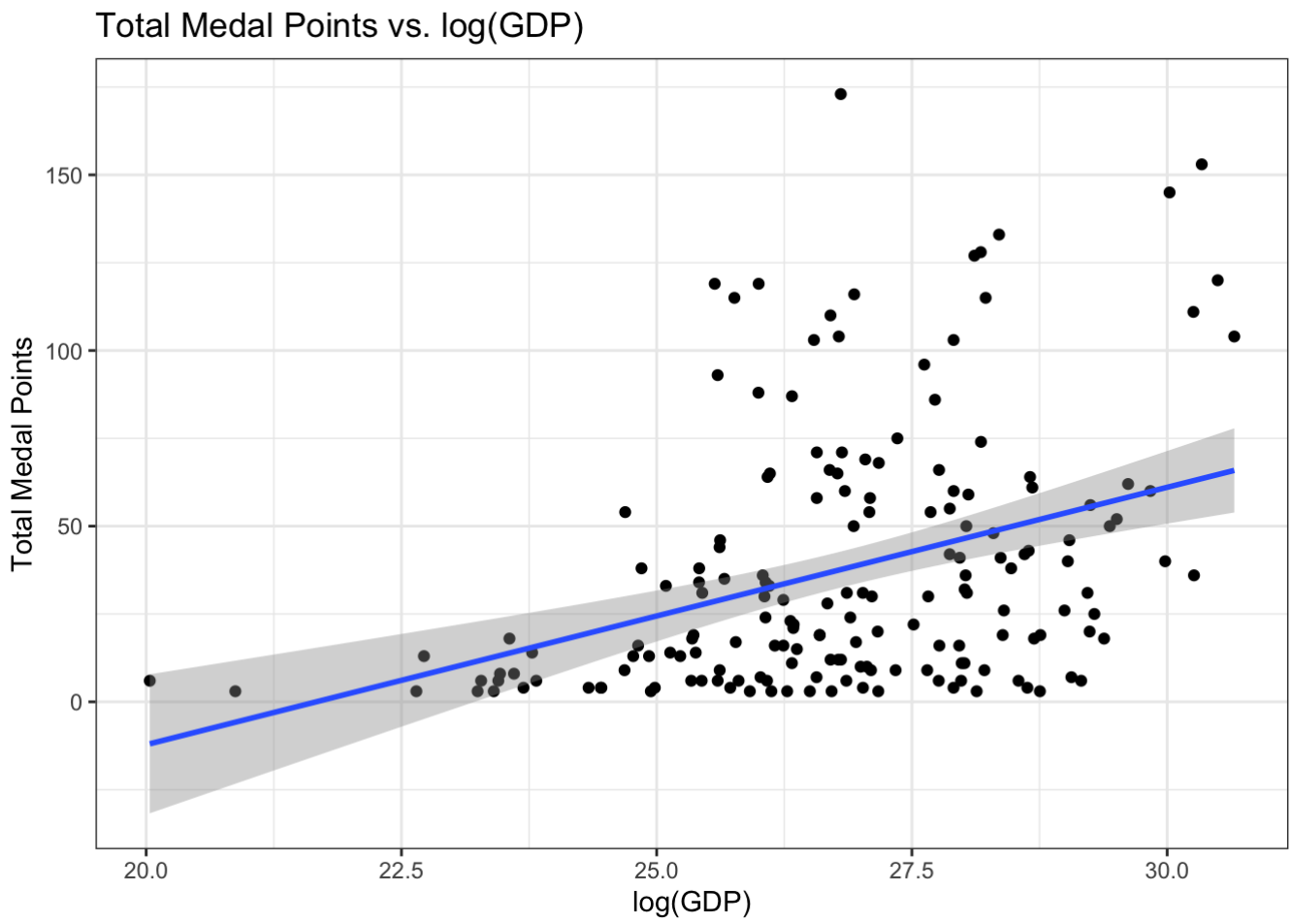
```
##
## Call:
## lm(formula = Total_Points ~ log(GDP), data = olympic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.925 -23.823  -8.679  13.521 135.361
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -158.733     38.369  -4.137 5.48e-05 ***
## log(GDP)       7.326      1.428   5.132 7.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.35 on 173 degrees of freedom
## Multiple R-squared:  0.1321, Adjusted R-squared:  0.1271
## F-statistic: 26.33 on 1 and 173 DF,  p-value: 7.662e-07
```

```
confint(model1)
```

```
##                   2.5 %    97.5 %
## (Intercept) -234.465299 -83.00056
## log(GDP)       4.508402  10.14443
```

```
ggplot(olympic, aes(x = log(GDP), y = Total_Points)) +
  geom_point() +
    geom_smooth(method = "lm")+
  labs(x = "log(GDP)",
       y = "Total Medal Points",
       title = "Total Medal Points vs. log(GDP)")+
  theme_bw()
```

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



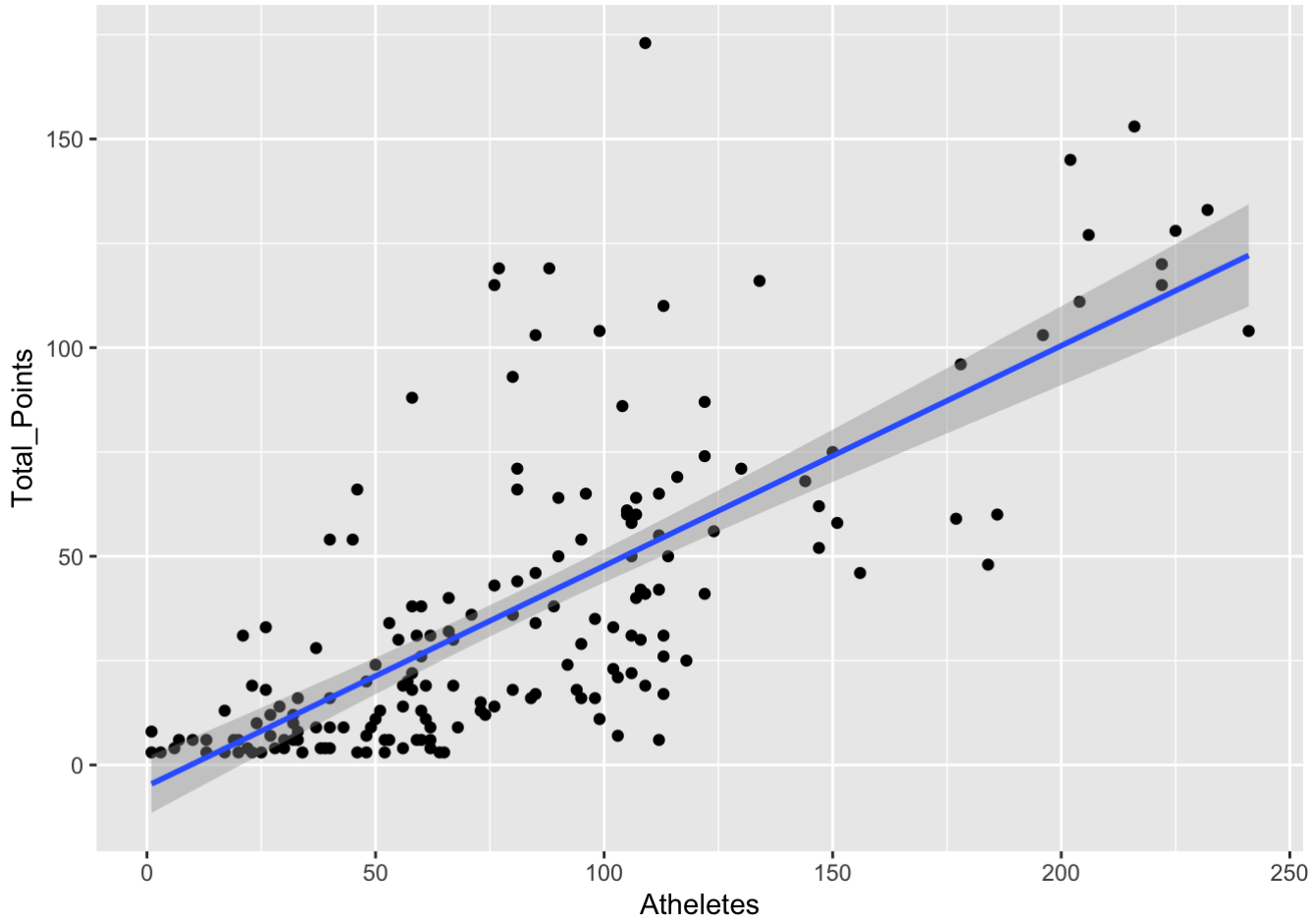
Analysis 2

Check assumptions

1. Linearity

```
# 1. Linearity
ggplot(olympic, aes(x = Atheletes, y = Total_Points)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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



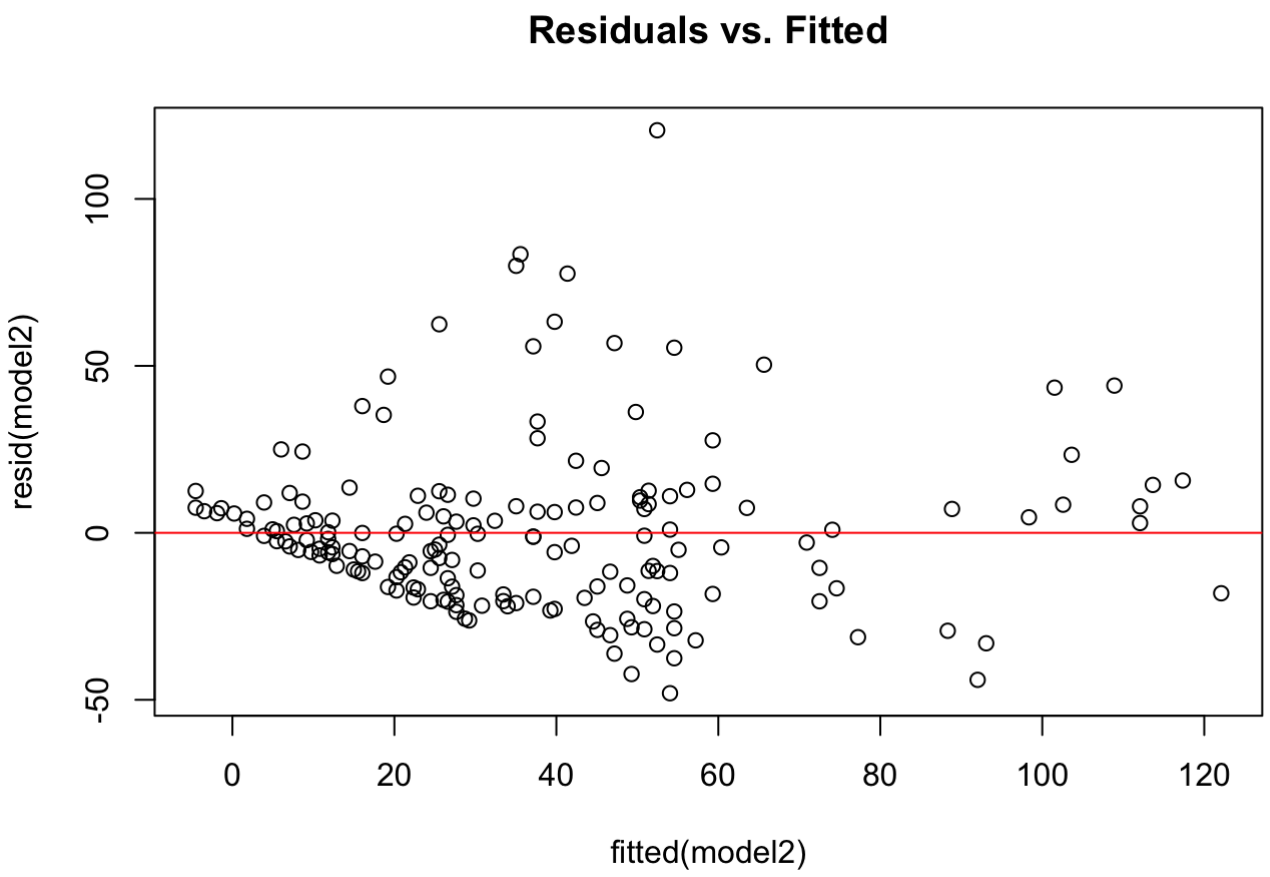
2. Indepence

Each country is independent of one another.We assume that all the number of Athletes and number of medals are independent.

3. Equal Variance

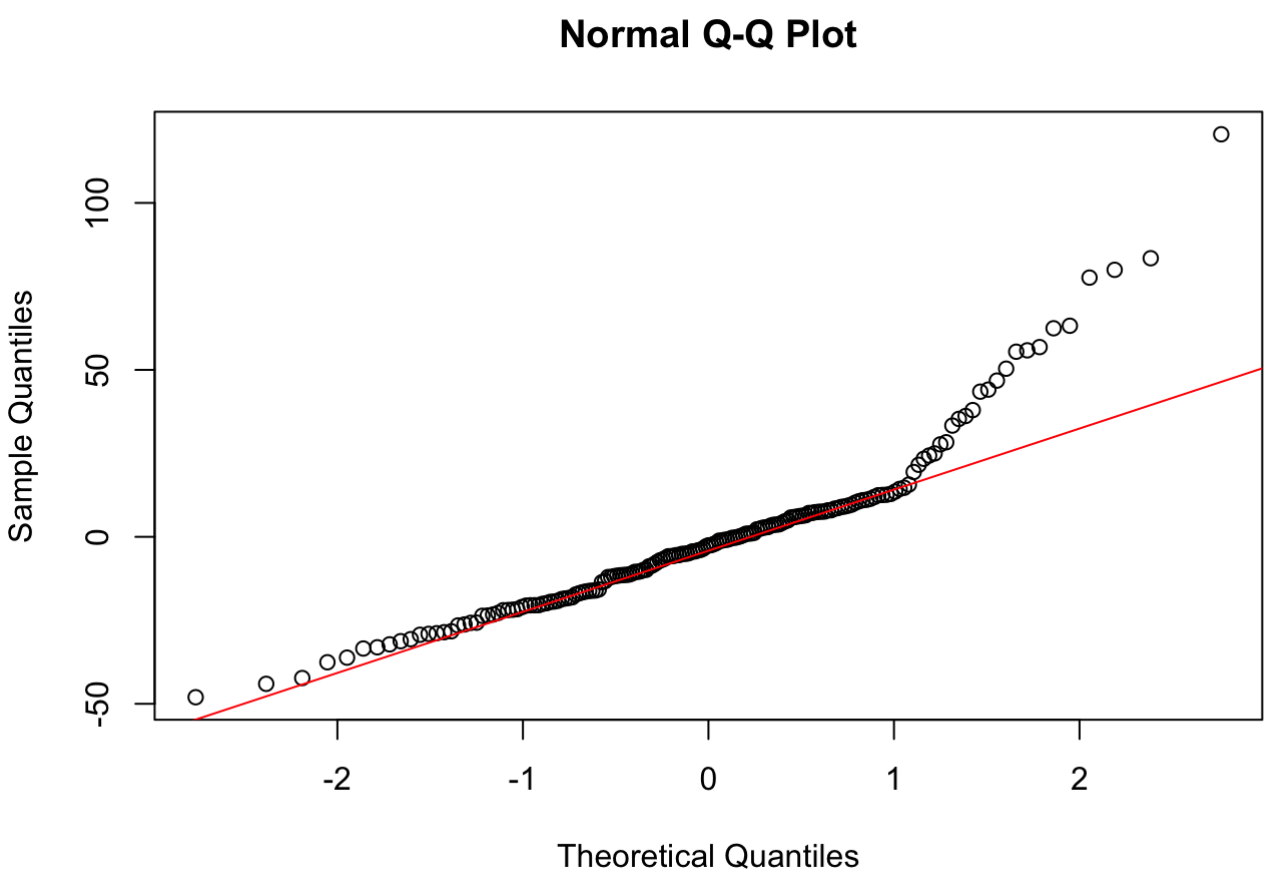
```
model2 <- lm(Total_Points ~ Atheletes, data = olympic)

plot(resid(model2) ~ fitted(model2), main = "Residuals vs. Fitted")
abline(h = 0, col = "red")
```



4. Normality

```
qqnorm(resid(model2))
qqline(resid(model2), col = "red")
```



Linear Regression

```
model2
```

```
##
## Call:
## lm(formula = Total_Points ~ Atheletes, data = olympic)
##
## Coefficients:
## (Intercept)    Atheletes
##     -5.0620       0.5276
```

```
summary(model2)
```

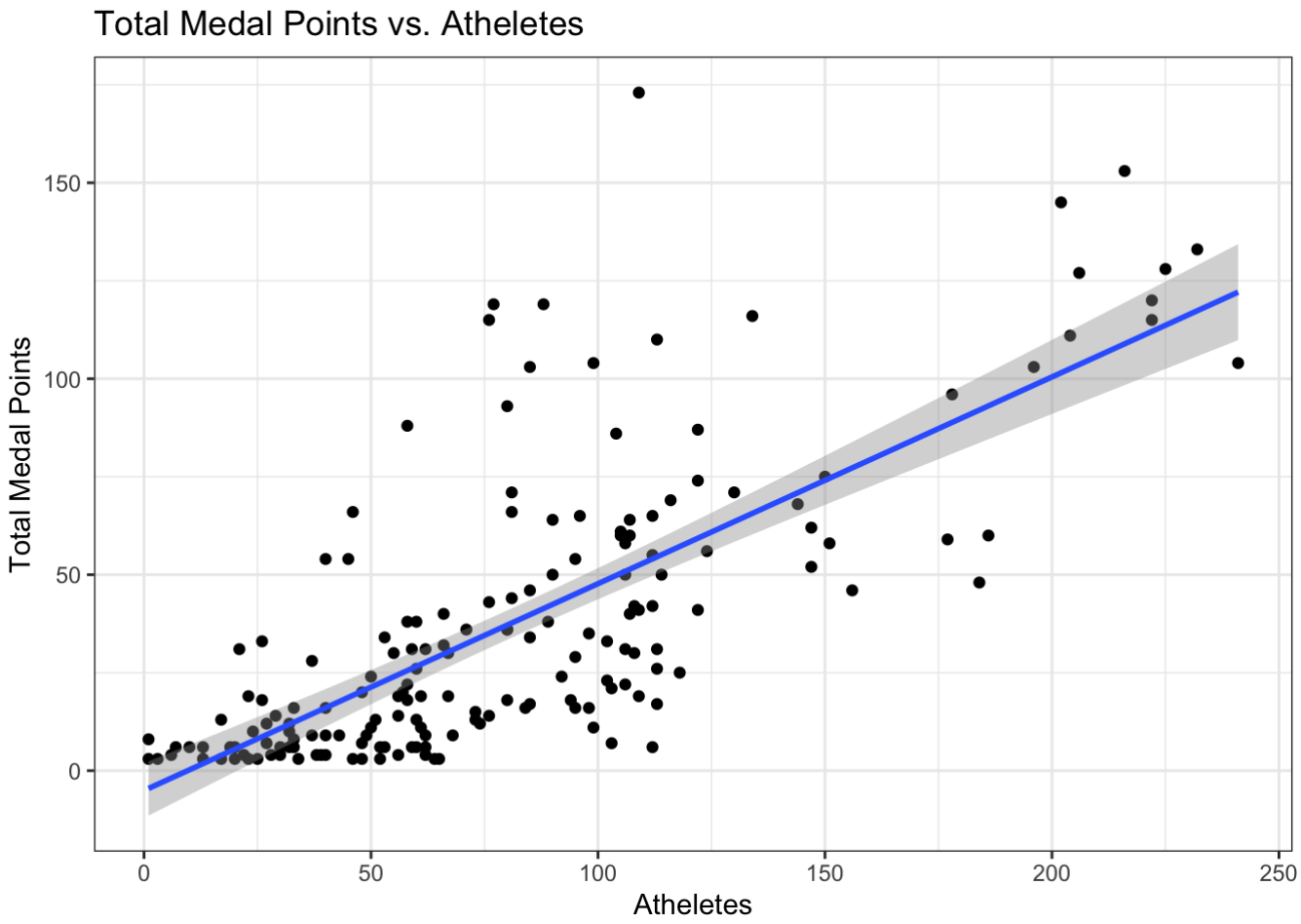
```
##
## Call:
## lm(formula = Total_Points ~ Atheletes, data = olympic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.032 -16.492  -2.546   8.195 120.551
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.06204    3.54032   -1.43    0.155
## Atheletes    0.52762    0.03695   14.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.98 on 173 degrees of freedom
## Multiple R-squared:  0.5411, Adjusted R-squared:  0.5384
## F-statistic:   204 on 1 and 173 DF,  p-value: < 2.2e-16
```

```
confint(model2)
```

```
##                   2.5 %    97.5 %
## (Intercept) -12.0498112 1.9257374
## Atheletes     0.4547016 0.6005439
```

```
ggplot(olympic, aes(x = Atheletes, y = Total_Points)) +
  geom_point() +
    geom_smooth(method = "lm")+
  labs(x = "Atheletes",
       y = "Total Medal Points",
       title = "Total Medal Points vs. Atheletes")+
  theme_bw()
```

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

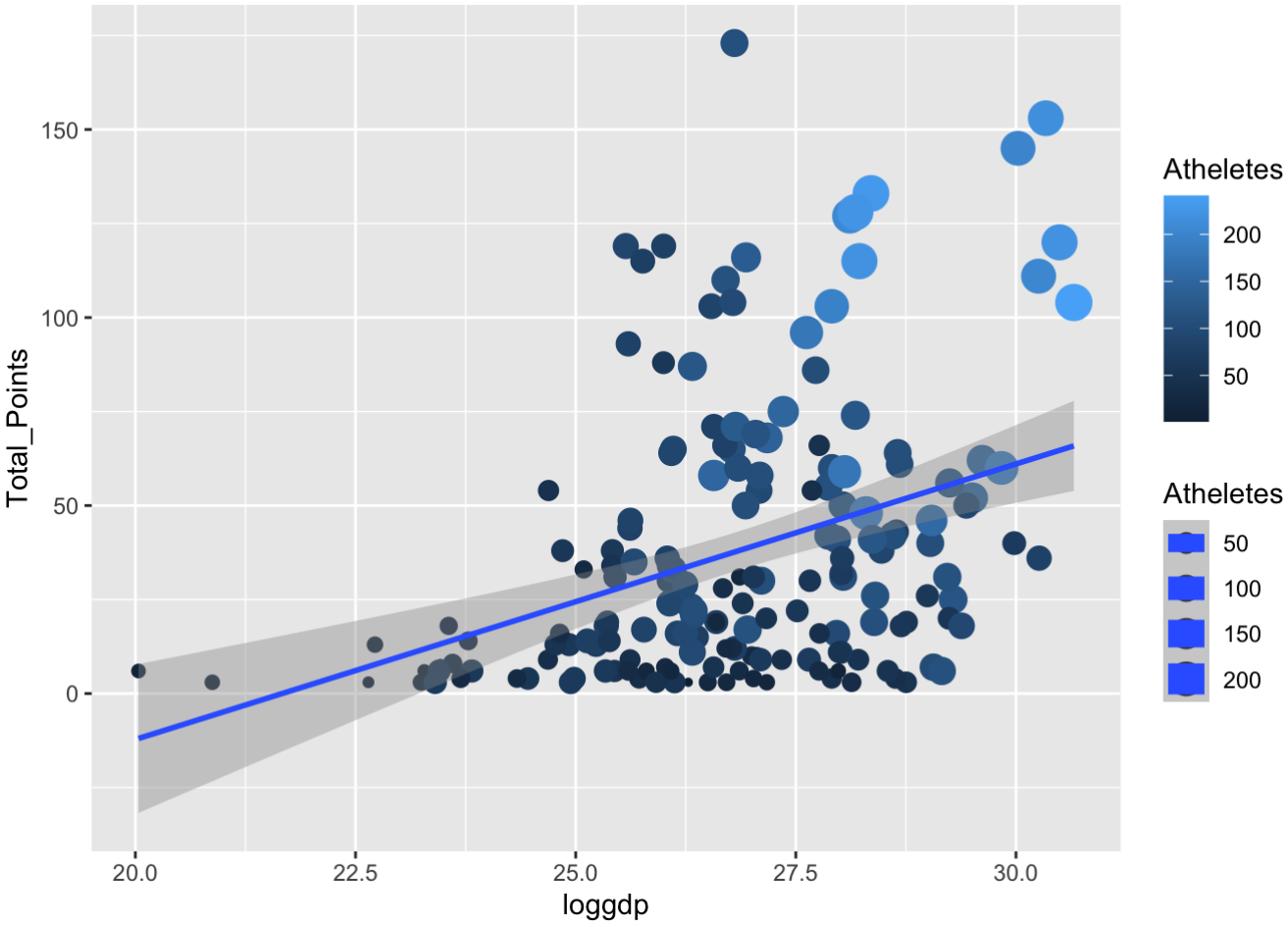



```
olympic %>%
  ggplot(aes(x = loggdp, y = Total_Points, col=Atheletes, size=Atheletes)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
```

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

```
## Warning: The following aesthetics were dropped during statistical transformation:
## colour, size
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```


Total_Points
150
100
50
0
20.0
22.5
25.0
27.5
30.0
loggdp
Atheletes
200
150
100
50
Atheletes
50
100
150
200