Project

Ashish Ashish   Arash Ahmadi

November 7th, 2024

I added two new columns:

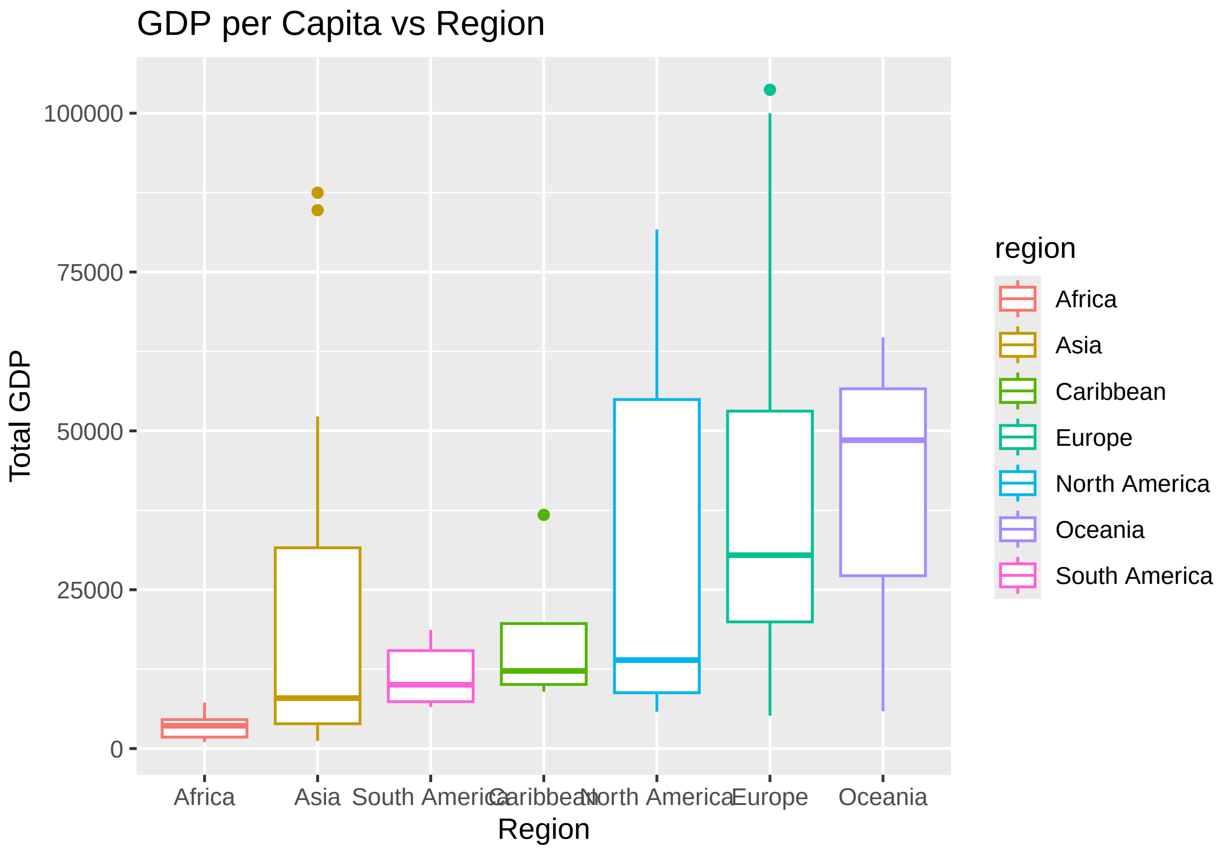
1. toal_gdp, the dataset already has gdp per capita
2. medal_pect, which is total medals divided by total athletes

```
# make the chr to factor
olympics <- olympics %>%
  mutate(country = factor(country),
         country_code = factor(country_code),
         region = factor(region))

olympics <- olympics %>%
  mutate(total_gdp = gdp * population,
         medal_pct = total / athletes)

summary(olympics)
```

```
##       country    country_code           region        gold       
##  Albania   : 1   ALB    : 1    Africa       :12   Min.   : 0.000  
##  Algeria   : 1   ARG    : 1    Asia         :26   1st Qu.: 0.000  
##  Argentina: 1    ARM    : 1    Caribbean    : 4   Median : 1.000  
##  Armenia   : 1   AUS    : 1    Europe       :31   Mean   : 3.644  
##  Australia: 1    AUT    : 1    North America: 7   3rd Qu.: 3.000  
##  Austria   : 1   AZE    : 1    Oceania      : 3   Max.   :40.000  
##  (Other)   :84   (Other):84    South America: 7                   
##      silver          bronze           total             gdp        
##  Min.   : 0.000   Min.   : 0.000   Min.   :  1.00   Min.   :  1014  
##  1st Qu.: 0.000   1st Qu.: 1.000   1st Qu.:  2.00   1st Qu.:  5815  
##  Median : 1.000   Median : 2.000   Median :  5.00   Median : 13061  
##  Mean   : 3.633   Mean   : 4.256   Mean   : 11.53   Mean   : 24478  
##  3rd Qu.: 3.000   3rd Qu.: 5.000   3rd Qu.:  9.00   3rd Qu.: 34485  
##  Max.   :44.000   Max.   :42.000   Max.   :126.00   Max.   :103685  
##                                                                     
##     gdp_year      population         athletes       total_gdp       
##  Min.   :2022   Min.   :   0.100   Min.   :  4.0   Min.   :     895  
##  1st Qu.:2023   1st Qu.:   5.325   1st Qu.: 26.0   1st Qu.:   68724  
##  Median :2023   Median :  12.150   Median : 60.5   Median :  256882  
##  Mean   :2023   Mean   :  69.028   Mean   :111.6   Mean   : 1085174  
##  3rd Qu.:2023   3rd Qu.:  48.550   3rd Qu.:135.5   3rd Qu.:  621058  
##  Max.   :2023   Max.   :1428.600   Max.   :619.0   Max.   :27359719  
```

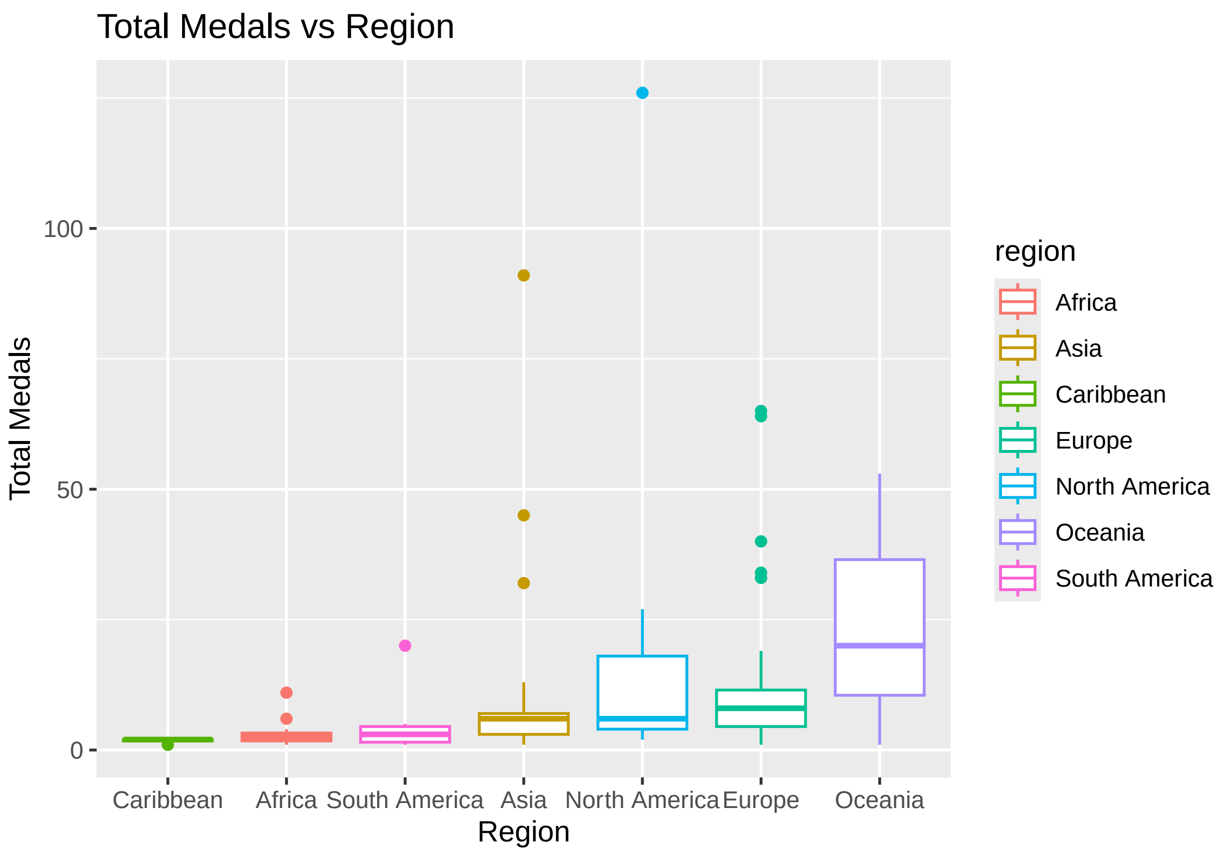
```
##
##    medal_pct
##  Min.    :0.01911
##  1st Qu.:0.05430
##  Median :0.09046
##  Mean   :0.11846
##  3rd Qu.:0.14509
##  Max.   :0.50000
##
```

```
olympics %>%
  ggplot(aes(x=reorder(region, gdp, FUN = median), y=gdp, col=region)) +
  geom_boxplot() +
  labs(title = "GDP per Capita vs Region",
       x = "Region",
       y = "Total GDP")
```



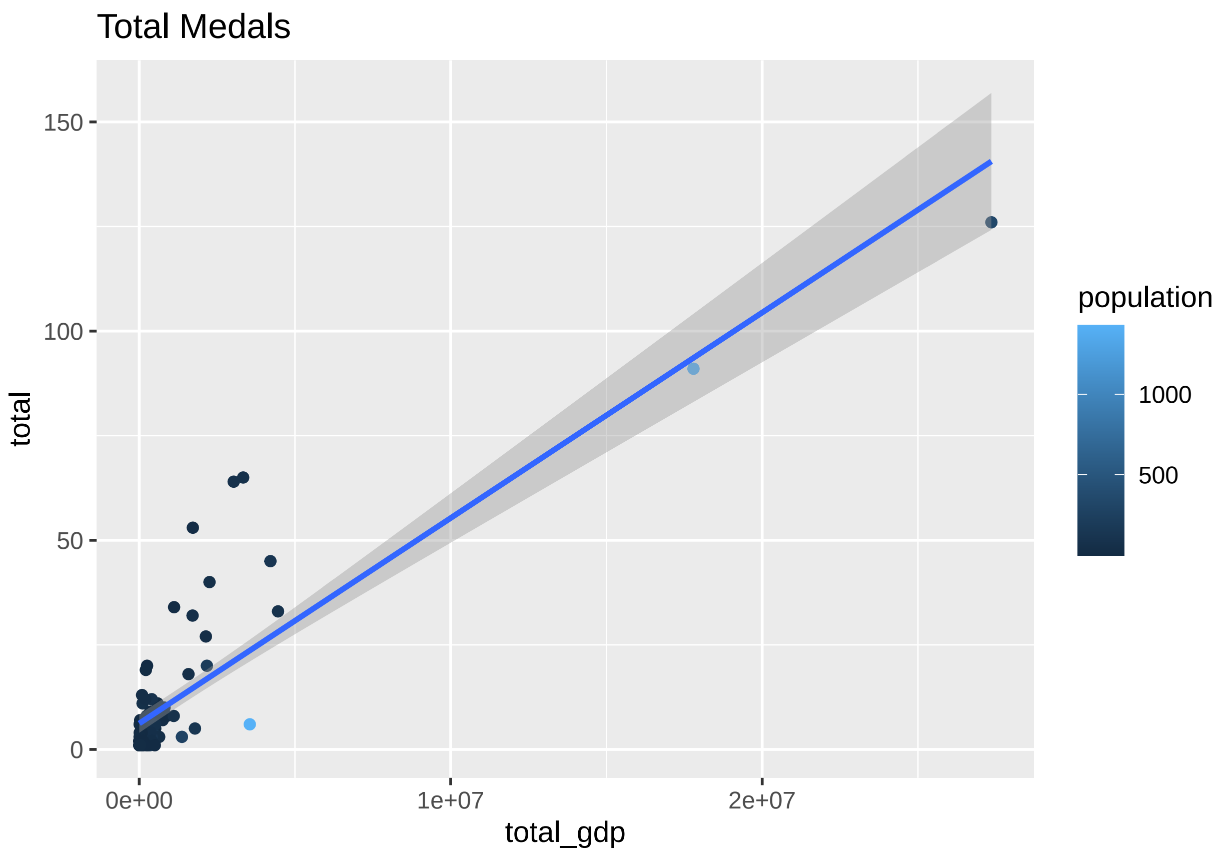
```
ggsave("chart1.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
```

```
olympics %>%
  ggplot(aes(x=reorder(region, total, FUN = median), y=total, col=region)) +
  geom_boxplot() +
  labs(title = "Total Medals vs Region",
       x = "Region",
       y = "Total Medals")
```

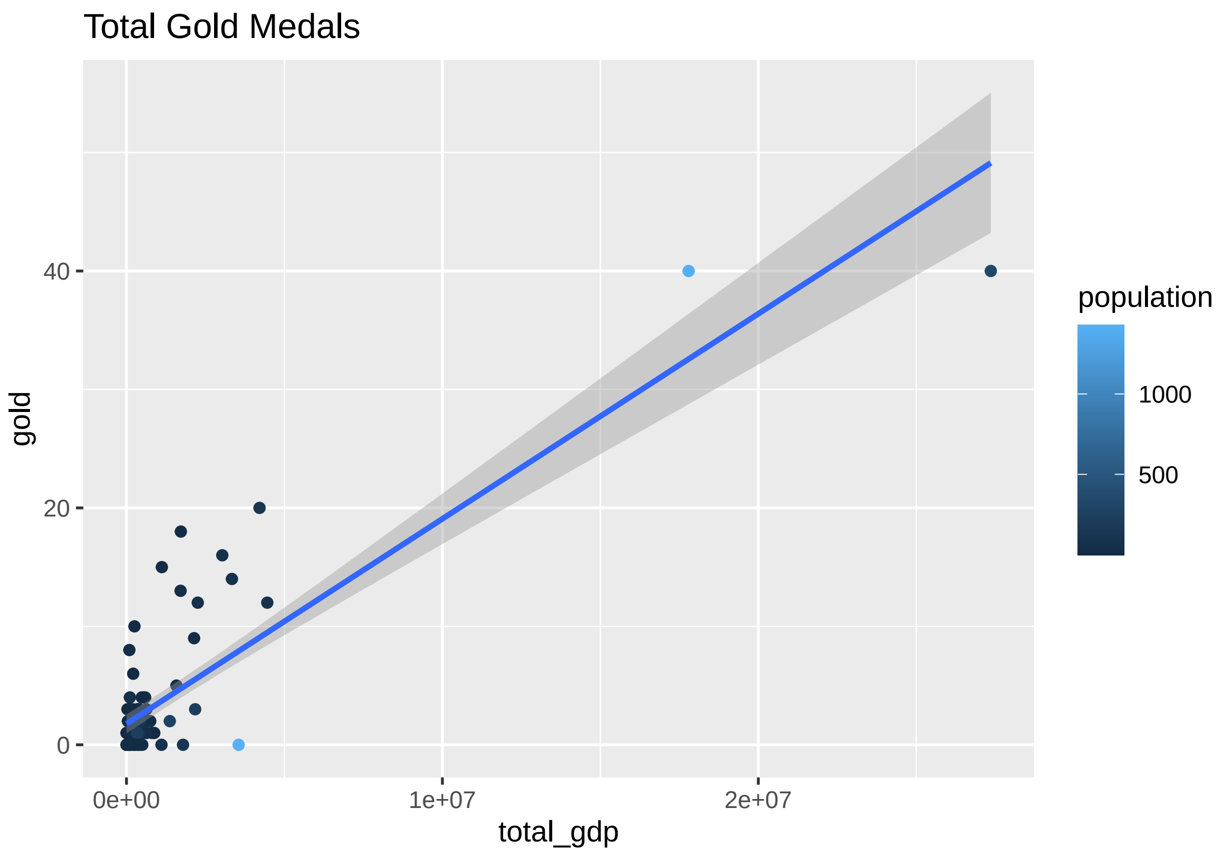


```r
ggsave("chart1.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
```

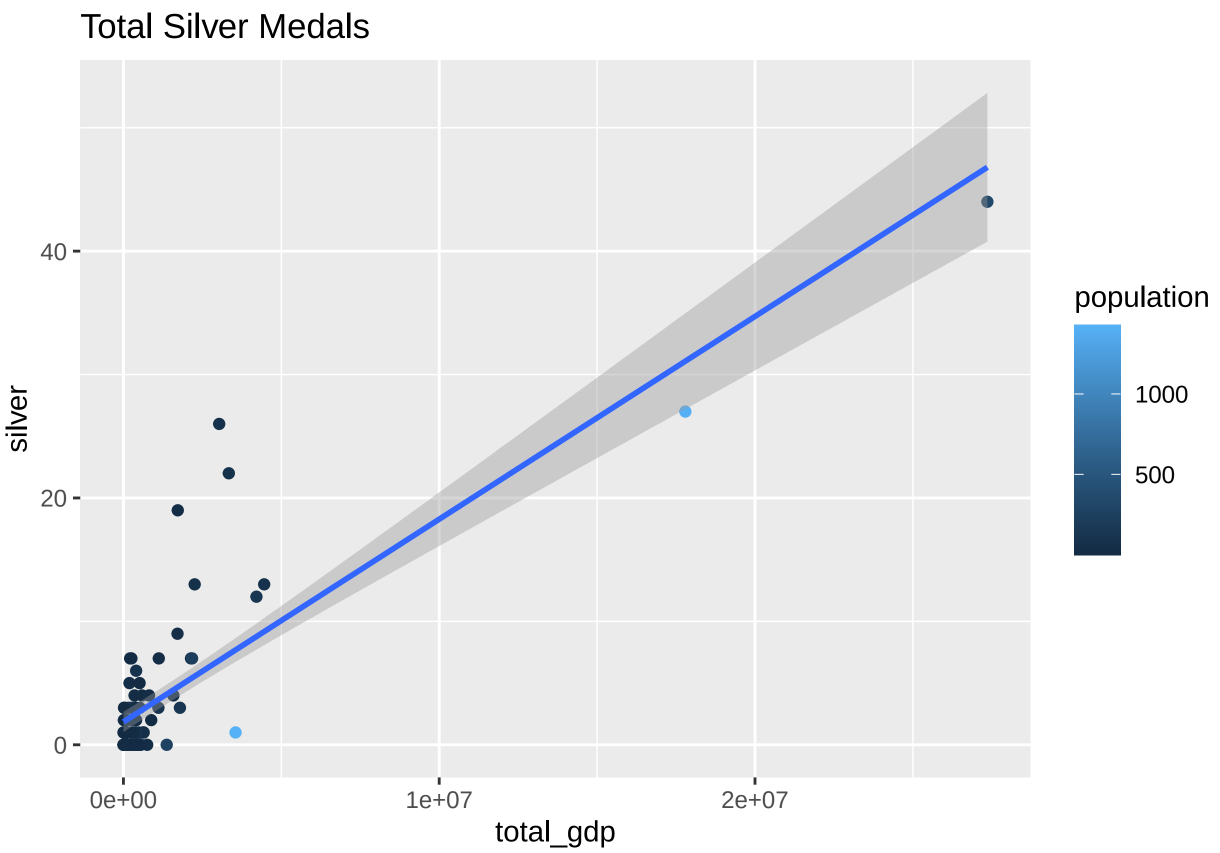
```r
olympics %>%
  ggplot(aes(x=total_gdp, y=total, col=population)) +
  geom_point() +
  geom_smooth(method='lm') +
  labs(
    title = "Total Medals"
  )
```



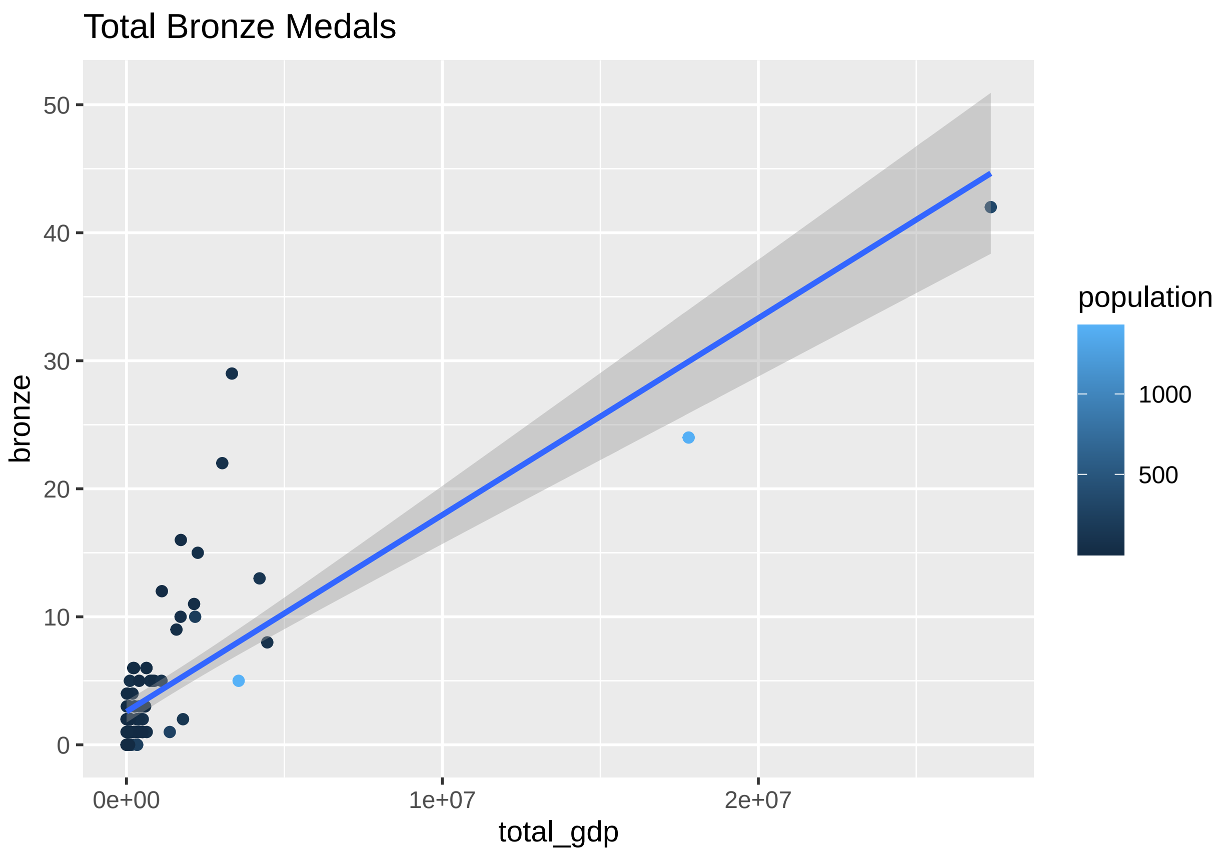
```r
ggsave("chart1.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
olympics %>%
  ggplot(aes(x=total_gdp, y=gold, col=population)) +
  geom_point() +
  geom_smooth(method='lm') +
  labs(
    title = "Total Gold Medals"
  )
```



```
ggsave("chart2.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
olympics %>%
  ggplot(aes(x=total_gdp, y=silver, col=population)) +
  geom_point() +
  geom_smooth(method='lm') +
  labs(
    title = "Total Silver Medals"
  )
```



```
ggsave("chart3.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
olympics %>%
  ggplot(aes(x=total_gdp, y=bronze, col=population)) +
  geom_point() +
  geom_smooth(method='lm') +
  labs(
    title = "Total Bronze Medals"
  )
```

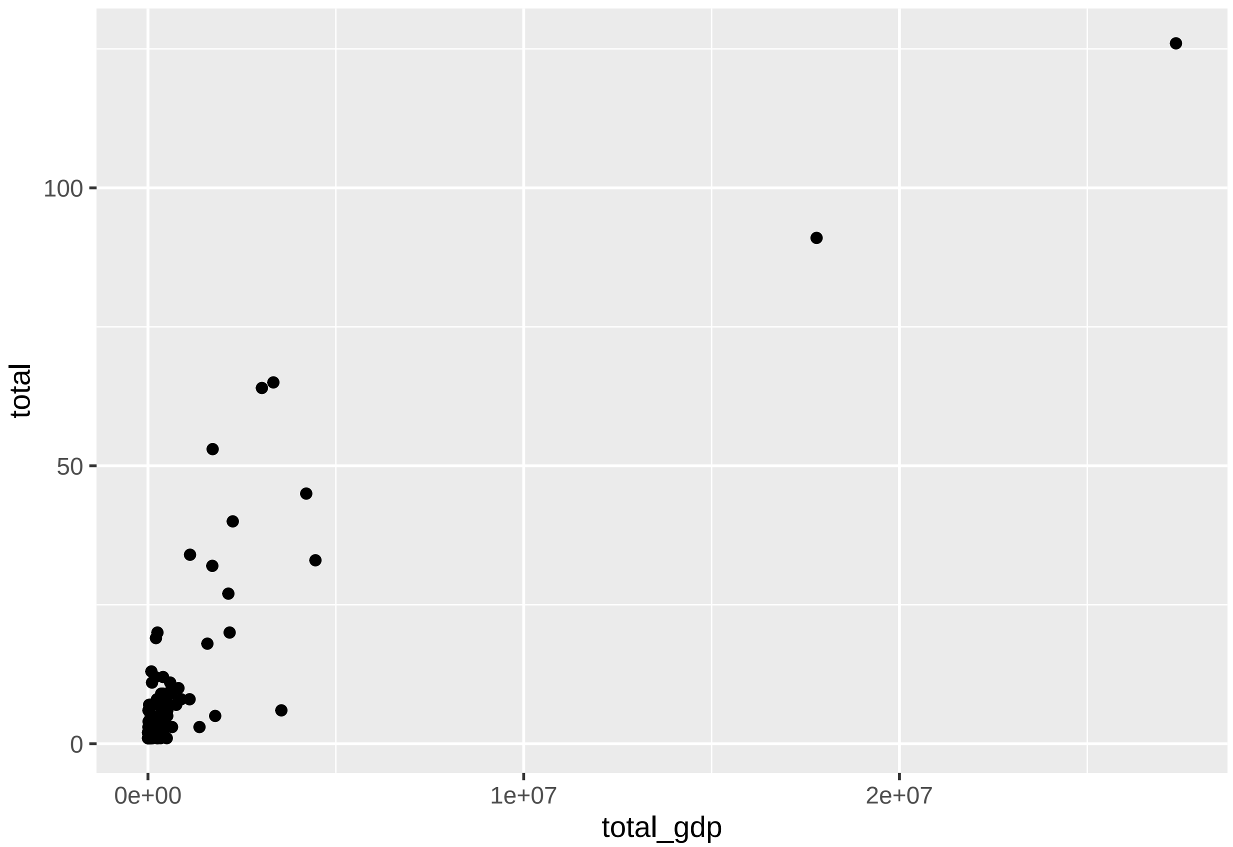


```r
ggsave("chart4.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
```

Check Conditions:

Linearity

```r
ggplot(olympics, aes(x=total_gdp, y=total)) +
  geom_point()
```



```r
ggsave("chart1.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
```

Independence:

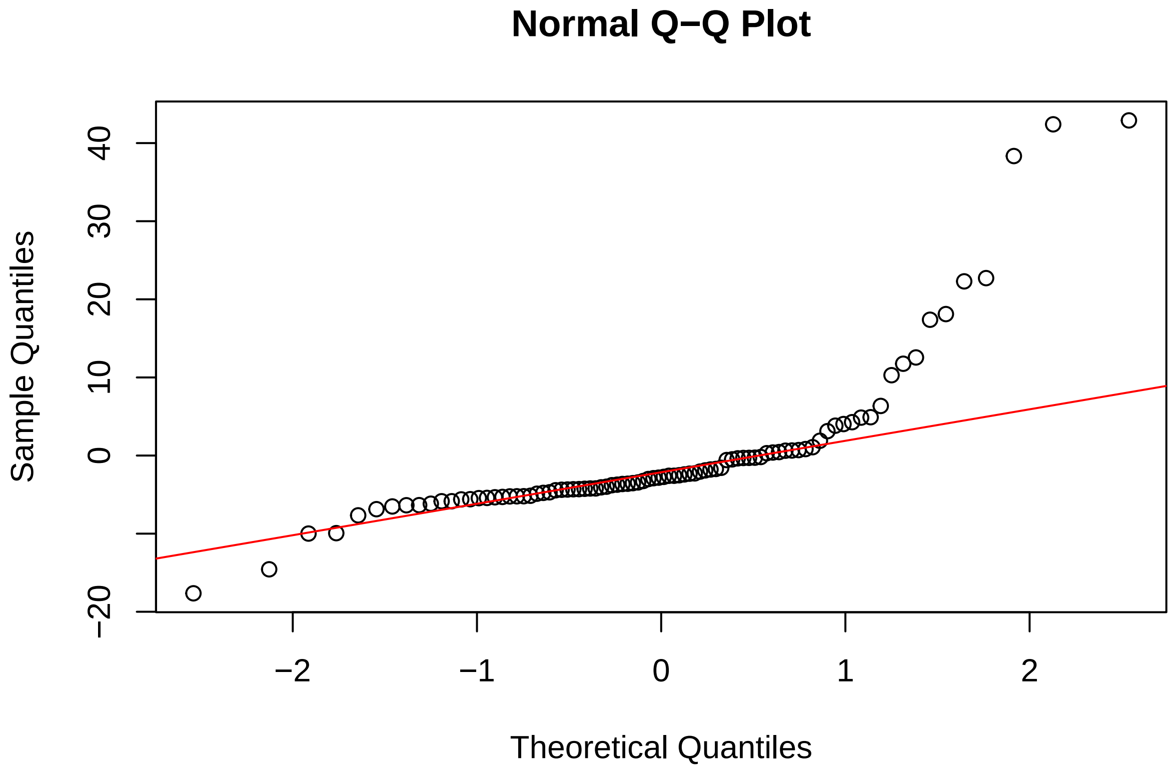
Normality of the residuals

```r
# Run SLR model
mod <- lm(total ~ total_gdp, data = olympics)
summary(mod)
```

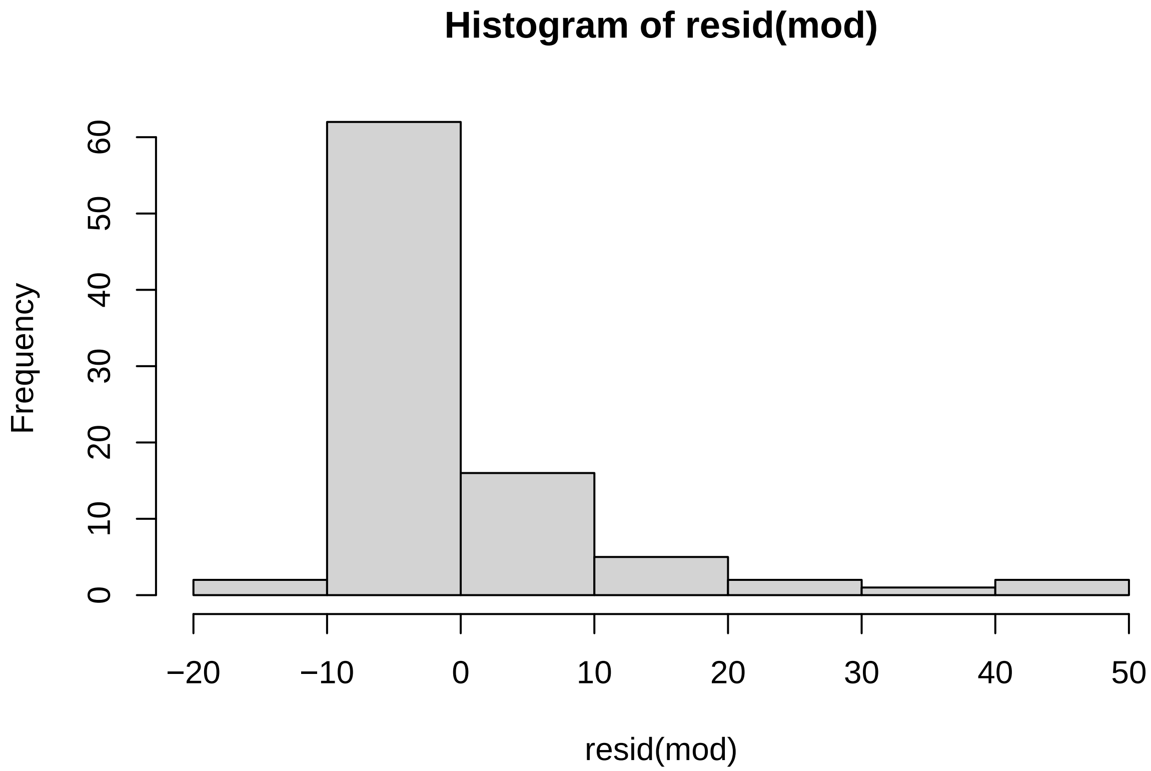
```
##
## Call:
## lm(formula = total ~ total_gdp, data = olympics)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.637  -4.860  -2.783   0.576  42.905
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.204e+00  1.123e+00   5.527 3.28e-07 ***
## total_gdp   4.911e-06  3.108e-07  15.798  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 10.16 on 88 degrees of freedom
## Multiple R-squared:  0.7393, Adjusted R-squared:  0.7364
## F-statistic: 249.6 on 1 and 88 DF,  p-value: < 2.2e-16
```

```
# Check Normality Assumption
qqnorm(resid(mod))
qqline(resid(mod), col = "red")
```



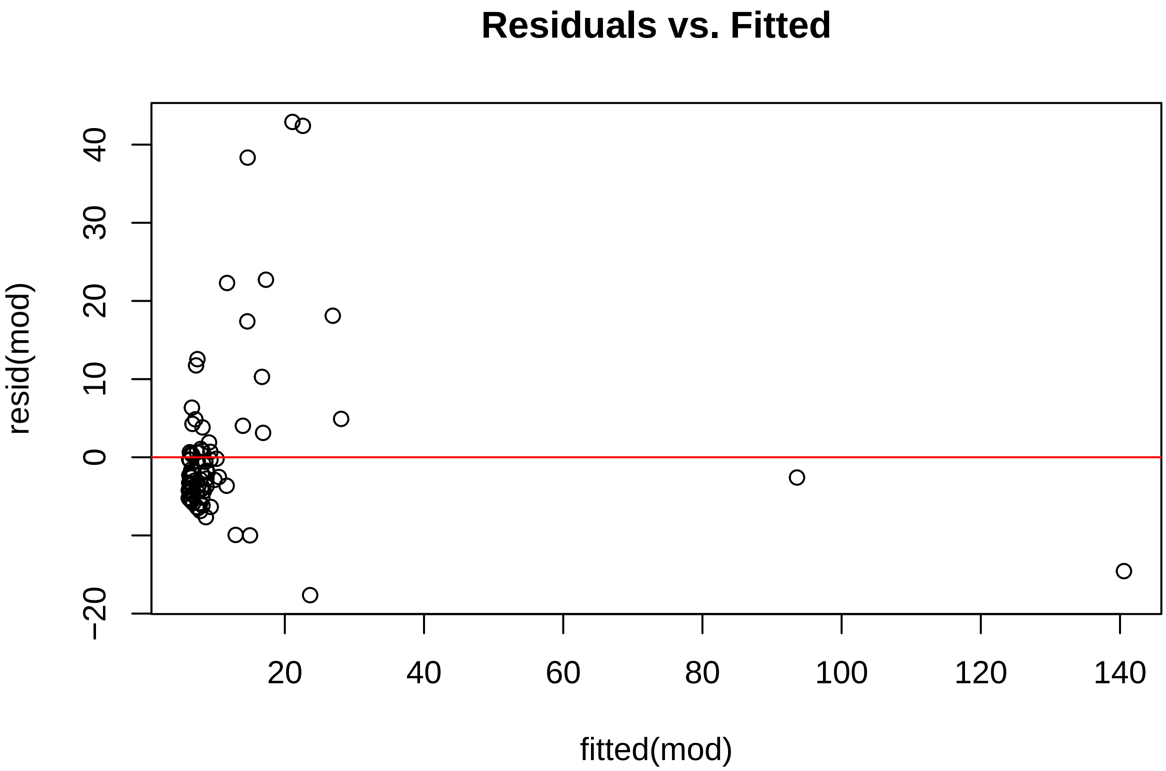
```
ggsave("chart1.png", plot = last_plot(), width = 8, height = 6, dpi = 300)

hist(resid(mod))
```



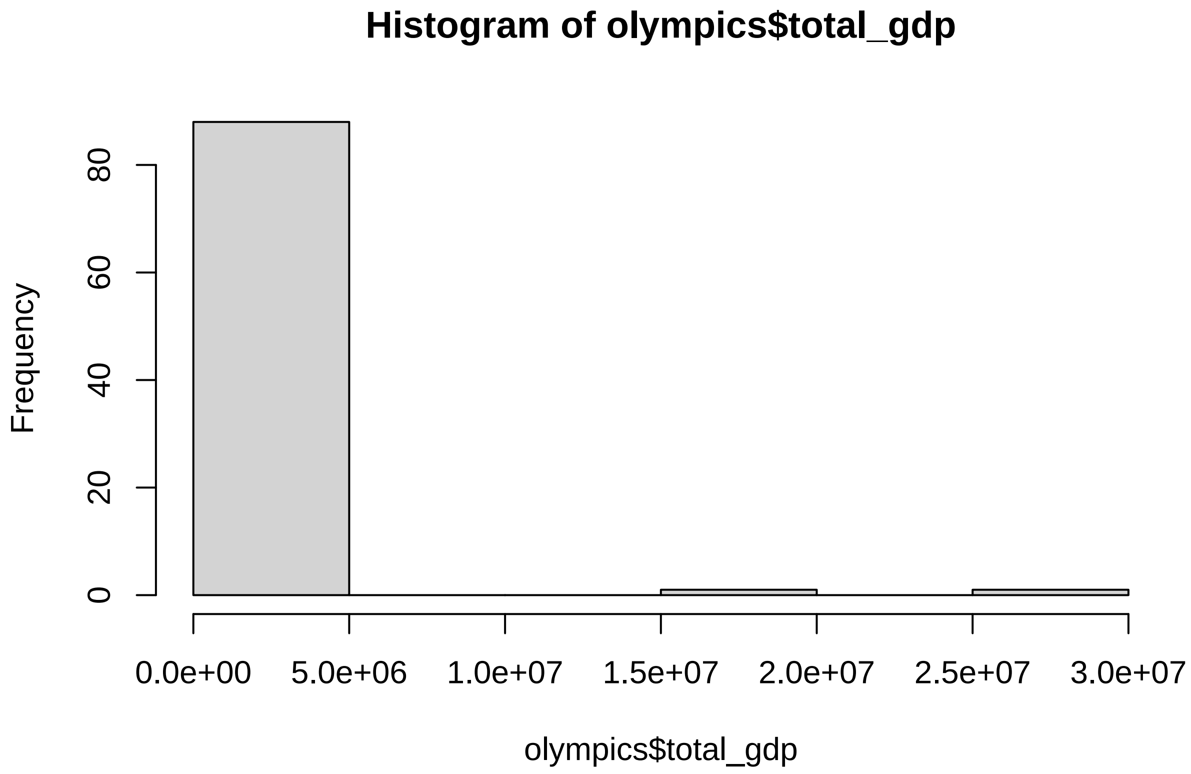
```r
ggsave("chart2.png", plot = last_plot(), width = 8, height = 6, dpi = 300)
```

Equal (constant) Variance of the residuals

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

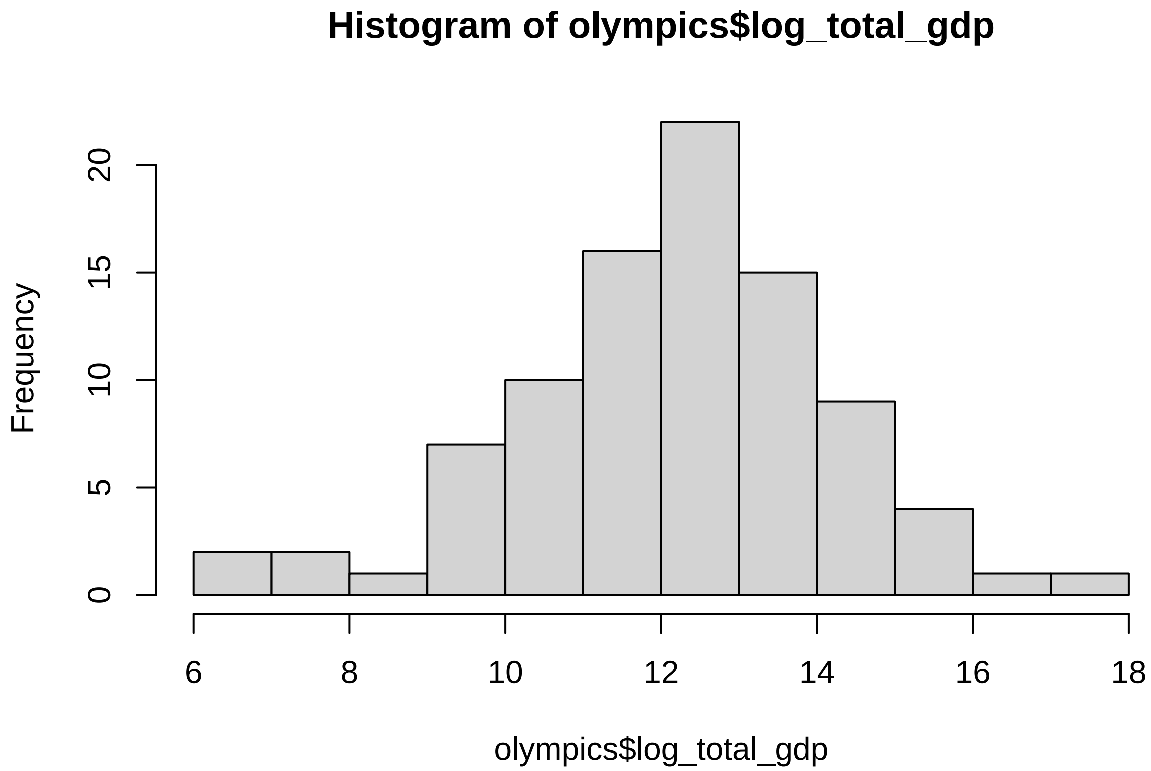


```r
hist(olympics$total_gdp)
```



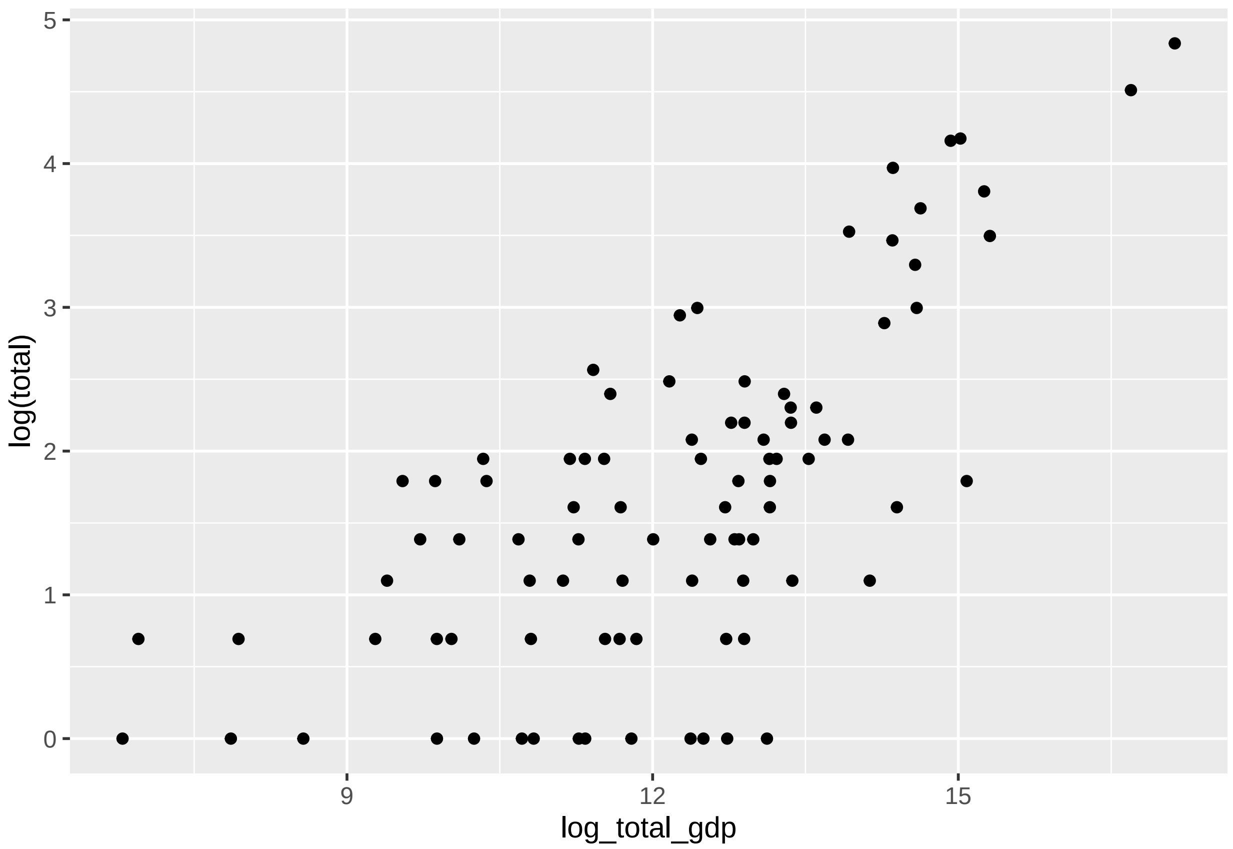
I checked the assumptions for total gdp, and most of them were not meet, therefore, we transformed the gdp by "log function" and retest all the assumptions.

```
olympics <- olympics %>%
  mutate(log_total_gdp = log(total_gdp))

hist(olympics$log_total_gdp)
```

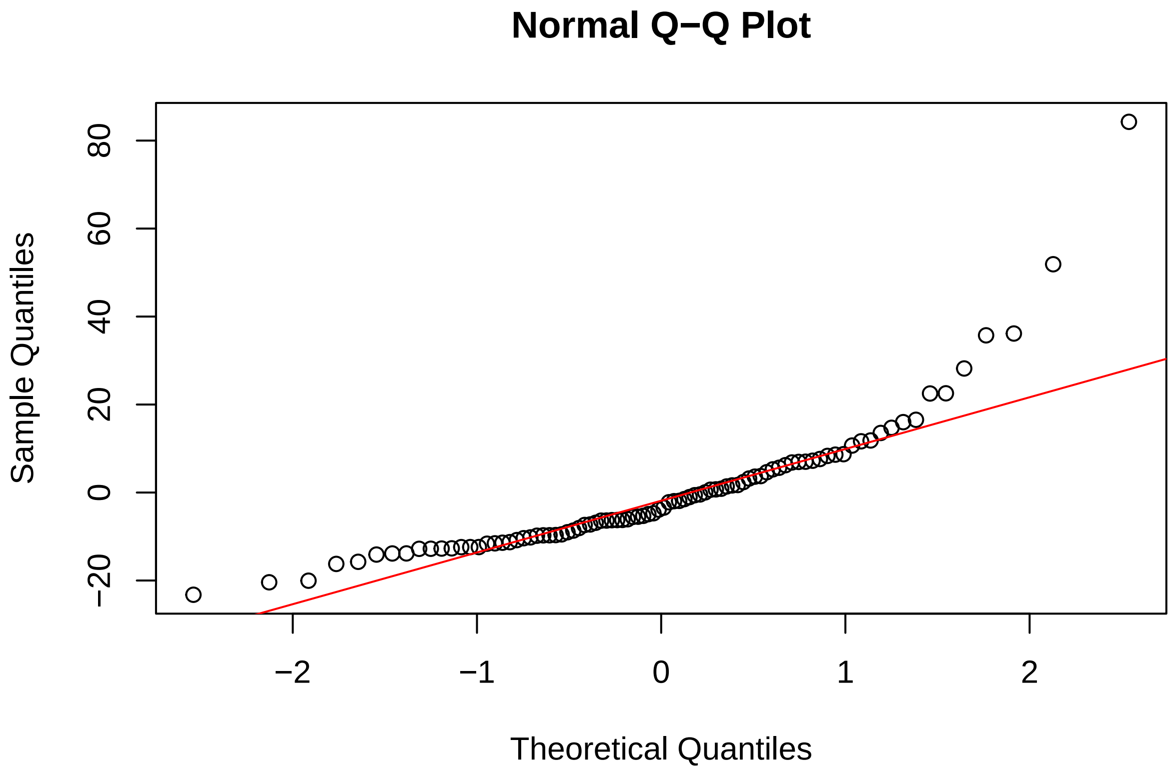


Check Assumptions

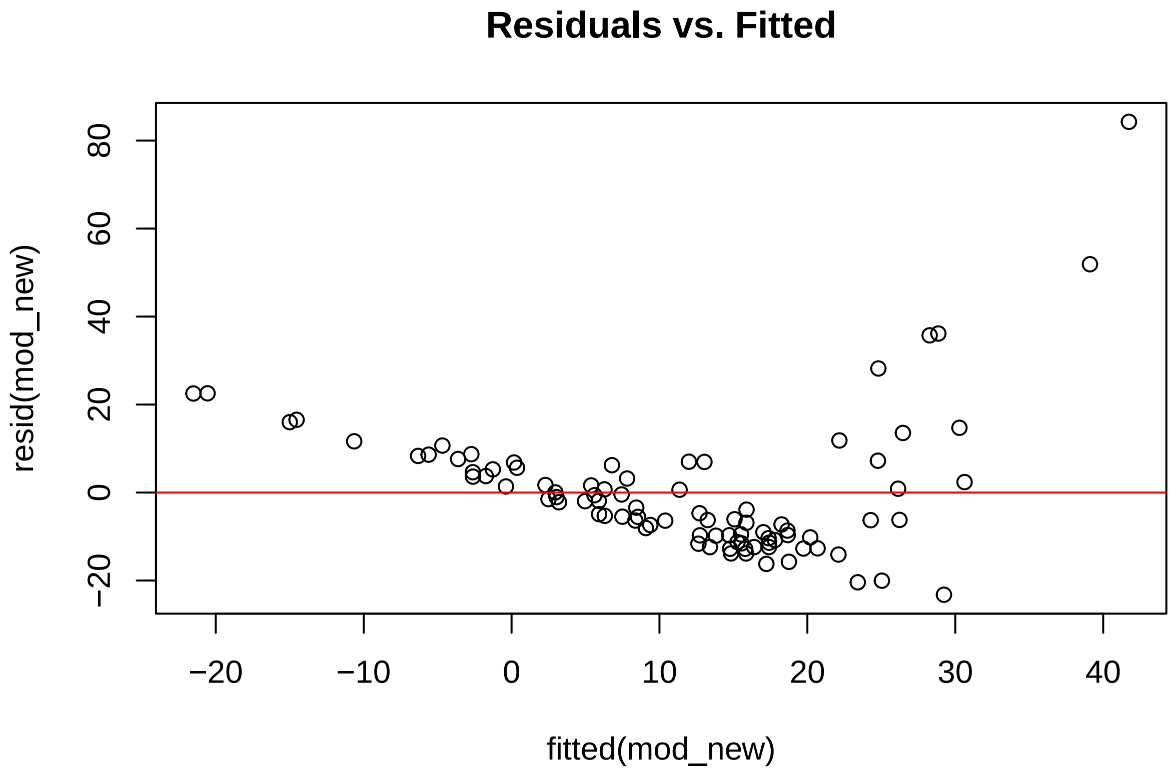
```
ggplot(olympics, aes(x = log_total_gdp, y = log(total))) +
  geom_point()
```



```
# Run model with transformed x
mod_new <- lm(total ~ log_total_gdp, data = olympics)

# Check Normality Assumption
qqnorm(resid(mod_new))
qqline(resid(mod_new), col = "red")
```



```r
# Check Equal (Constant) Variance Assumption
plot(resid(mod_new) ~ fitted(mod_new), main = "Residuals vs. Fitted")
abline(h = 0, col = "red")
```



```r
summary(mod_new)
```

```
##
## Call:
## lm(formula = total ~ log_total_gdp, data = olympics)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.233  -9.794  -3.659   6.069  84.259
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -63.142     10.448  -6.044 3.55e-08 ***
## log_total_gdp    6.125      0.846   7.239 1.61e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.75 on 88 degrees of freedom
## Multiple R-squared:  0.3733, Adjusted R-squared:  0.3661
## F-statistic: 52.41 on 1 and 88 DF,  p-value: 1.608e-10
```

Refrences:

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://repository.gatech.edu/server/api/core/bitstreams/1aa2b537-c3de-4177-8295-3fcd3a03a965/content

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://faculty.tuck.dartmouth.edu/images/uploads/faculty/andrew-bernard/olymp60restat_finaljournalversion.pdf