Final Project

David Calin

2023-04-28

Importing Data / Packages

```
library(gapminder)
library(ggplot2)
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(tidyr)
library(moderndive)
library(infer)
library(knitr)
library(epiDisplay)
```

```
## Loading required package: foreign
```

```
## Loading required package: survival
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

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

```
## Loading required package: nnet
```

```
##
## Attaching package: 'epiDisplay'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(rockchalk)
```

```
##
## Attaching package: 'rockchalk'

## The following object is masked from 'package:MASS':
##
##     mvrnorm

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

```
data=read.csv("HW1_Merged.csv")
data=data[,-1]
glimpse(data)
```

```
## Rows: 121
## Columns: 13
## $ country               <chr> "Afghanistan", "Albania", "Algeria", "Angola", "~
## $ continent             <chr> "Asia", "Europe", "Africa", "Africa", "Americas"~
## $ child_mortality       <dbl> 88.00, 13.30, 27.40, 120.00, 14.40, 4.77, 4.33, ~
## $ children_per_women    <dbl> 5.82, 1.65, 2.89, 6.16, 2.37, 1.93, 1.44, 2.14, ~
## $ co2                   <dbl> 0.2900, 1.5600, 3.2800, 1.2400, 4.5700, 18.4000,~
## $ gdp                   <int> 526, 3580, 3920, 3990, 13600, 53500, 43300, 2100~
## $ income_per_person     <int> 1960, 10700, 11000, 7690, 23500, 44900, 51800, 4~
## $ life_expectancy       <dbl> 60.5, 78.1, 74.5, 60.2, 75.9, 82.1, 80.8, 75.0, ~
## $ murder                <dbl> 4130.0, 65.9, 530.0, 824.0, 2450.0, 326.0, 68.5,~
## $ population_density    <dbl> 43.40, 106.00, 15.10, 18.70, 14.70, 2.87, 101.00~
## $ population_total      <dbl> 2.92e+07, 2.95e+06, 3.60e+07, 2.34e+07, 4.09e+07~
## $ total_health_spending <dbl> 37.7, 241.0, 178.0, 123.0, 742.0, 4780.0, 4960.0~
## $ water                 <dbl> 48.3, 91.4, 92.3, 50.4, 98.4, 99.9, 100.0, 100.0~
```

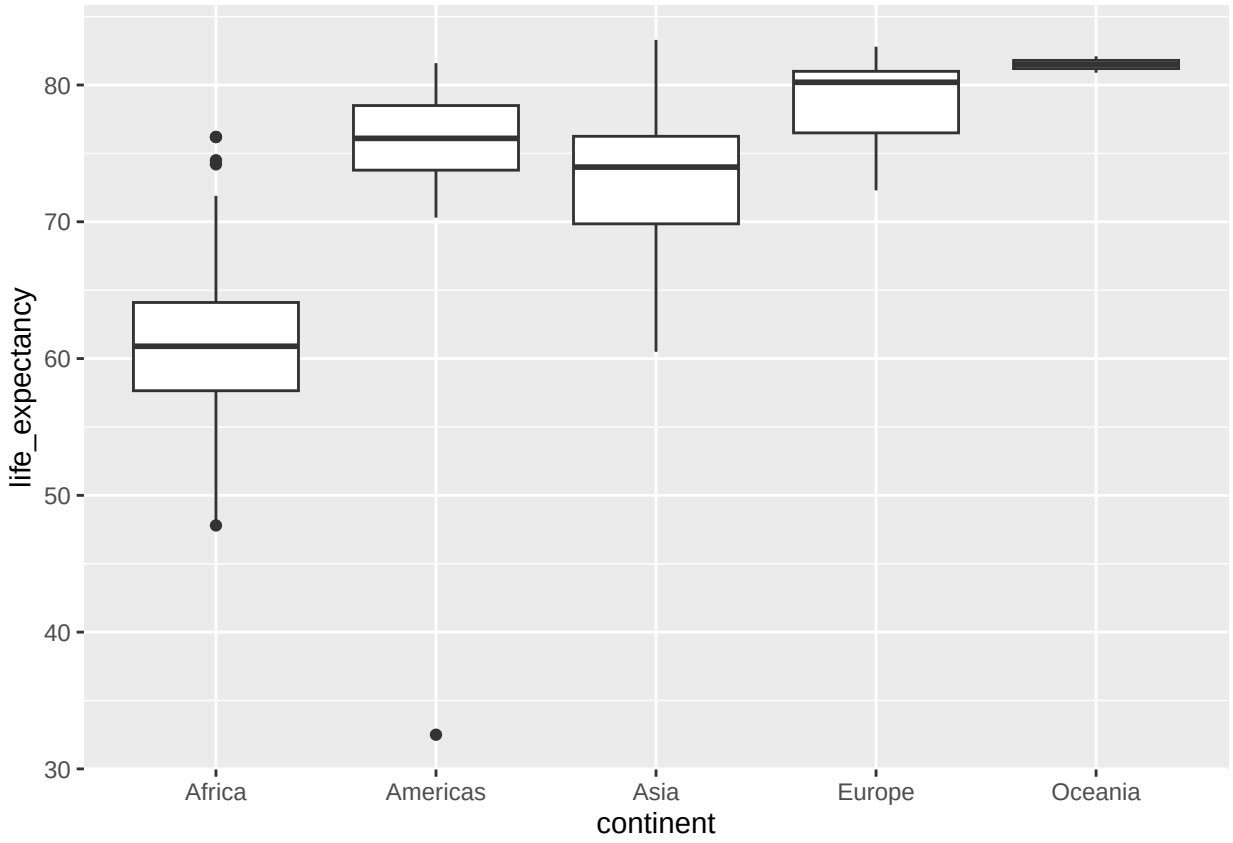
The present dataset (HW1_Merged.csv) is the aggregate of all .csvs used in HW1, for the year 2009, and includes 121 observations with 13 variables. Its categorical variables are country and continent. Its continuous variables are: child mortality rate, number of children per woman, co2 emissions per person, gdp per capita adjsuted for US inflation, income per person, total health spending per person, population density, population total, number of murders, and percentage of people with at least a basic water source.

Exploratory Data Analysis

```
# We can first explore the mean and median life expectancy at birth by continent.
LE_contient=data %>% group_by(continent) %>% summarise(mean_LE=mean(life_expectancy), median_LE=median(
LE_contient
```

```
## # A tibble: 5 x 3
##   continent mean_LE median_LE
##   <chr>       <dbl>     <dbl>
## 1 Africa       61.6      60.9
## 2 Americas     73.9      76.1
## 3 Asia         72.9      74
## 4 Europe       78.9      80.2
## 5 Oceania      81.5      81.5
```

```r
# Just like HW1, we can use a boxplot to better visualize this and learn more about the data, including
ggplot(data=data, aes(y=life_expectancy, x=continent)) + geom_boxplot()
```



From this graph, we can conclude that the distribution of life expectancy of African countries is approximately normally distributed, but with the most outliers out of any continent. On the other hand, while the distribution life expectancy of European countries appears to be very negatively skewed, there seems to be no significant outliers, if any at all.

```r
# Let's now explore the relationships between the dependent variable (life expectancy in 2009) and all

cor_child_mortality=cor(data$life_expectancy,data$child_mortality)
cor_children_per_women=cor(data$life_expectancy,data$children_per_women)
cor_co2=cor(data$life_expectancy,data$co2)
cor_gdp=cor(data$life_expectancy,data$gdp)
cor_murder=cor(data$life_expectancy,data$murder)
cor_population_density=cor(data$life_expectancy,data$population_density)
cor_population_total=cor(data$life_expectancy,data$population_total)
cor_total_health_spending=cor(data$life_expectancy,data$total_health_spending)
cor_water=cor(data$life_expectancy,data$water)
cor_income=cor(data$life_expectancy,data$income_per_person)

# Ranking Correlations
cor_list=c("cor_child_mortality"=cor_child_mortality,"cor_children_per_women"=cor_children_per_women,"c
ranked_cor_list=order(-cor_list)
cor_table=data.frame(rank=1:length(ranked_cor_list),
                     cor_name=names(cor_list)[ranked_cor_list],
                     cor_value=cor_list[ranked_cor_list])
```

```
cor_table
```

```
##                              rank                  cor_name   cor_value
## cor_water                       1                cor_water  0.82771222
## cor_income                      2               cor_income  0.71441899
## cor_gdp                         3                  cor_gdp  0.64235324
## cor_total_health_spending       4 cor_total_health_spending  0.58489853
## cor_co2                         5                  cor_co2  0.51280110
## cor_population_density          6   cor_population_density  0.14368533
## cor_population_total            7     cor_population_total  0.06078604
## cor_murder                      8              cor_murder -0.03701745
## cor_children_per_women          9   cor_children_per_women -0.78919875
## cor_child_mortality            10      cor_child_mortality -0.91385359
```

```
plot_water=ggplot(data=data, aes(x=water, y=life_expectancy)) + geom_point() +  geom_smooth(method = "lm
plot_income=ggplot(data=data, aes(x=income_per_person, y=life_expectancy)) + geom_point()+ geom_smooth(m
plot_gdp=ggplot(data=data, aes(x=gdp, y=life_expectancy)) + geom_point() + geom_smooth(method = "lm", se
plot_health=ggplot(data=data, aes(x=total_health_spending, y=life_expectancy)) + geom_point() + geom_smo
plot_co2=ggplot(data=data, aes(x= co2, y=life_expectancy)) + geom_point() + geom_smooth(method = "lm", s
plot_pop_density=ggplot(data=data, aes(x= population_density, y=life_expectancy)) + geom_point() + geom_
plot_pop_total=ggplot(data=data, aes(x= population_total, y=life_expectancy)) + geom_point() + geom_smoo
plot_murder=ggplot(data=data, aes(x= murder, y=life_expectancy)) + geom_point() + geom_smooth(method = '
plot_children=ggplot(data=data, aes(x=children_per_women, y=life_expectancy)) + geom_point() + geom_smoo
plot_mortality=ggplot(data=data, aes(x=child_mortality, y=life_expectancy)) + geom_point() + geom_smoothl

library(gridExtra)
```

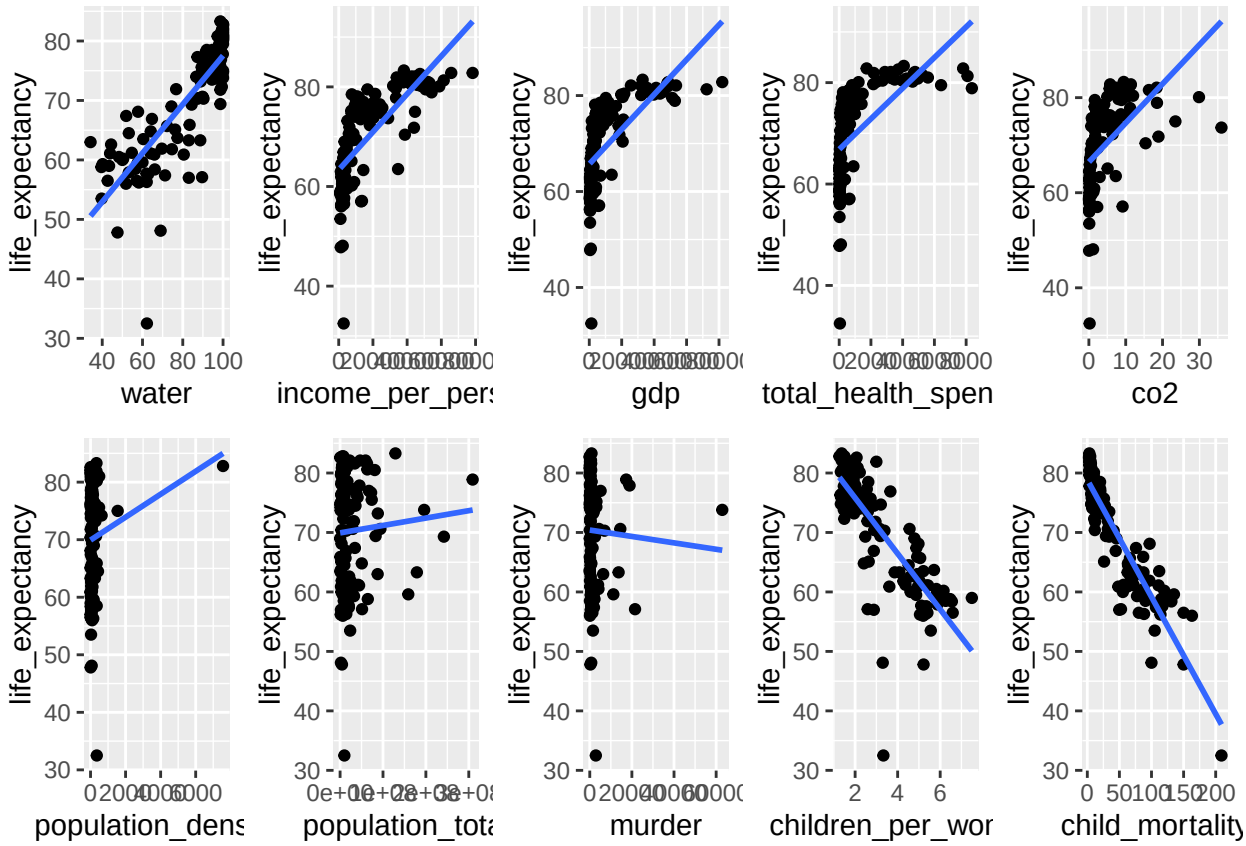
```
##
## Attaching package: 'gridExtra'
```

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

```
grid.arrange(plot_water, plot_income, plot_gdp, plot_health, plot_co2, plot_pop_density, plot_pop_total
```

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

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



Since water and child mortality exhibit and strong, linear correlation (|r|> 0.8) with life expectancy, and children per woman almost reaching that threshold, these three look to be strong contenders for the first-order model. However, all the other scatterplots appear to follow a logarthmic pattern, to varying degrees. So, let's transform them and then re-check their correlations.

```
# Checking Correlations With Log Transformations
cor_log_co2=cor(log(data$life_expectancy),log(data$co2))
cor_log_gdp=cor(log(data$life_expectancy),log(data$gdp))
cor_log_murder=cor(log(data$life_expectancy),log(data$murder))
cor_log_population_density=cor(log(data$life_expectancy),log(data$population_density))
cor_log_population_total=cor(log(data$life_expectancy),log(data$population_total))
cor_log_total_health_spending=cor(log(data$life_expectancy),log(data$total_health_spending))
cor_log_income=cor(log(data$life_expectancy),log(data$income_per_person))

# Ranking Correlations
cor_list2=c("cor_child_mortality"=cor_child_mortality,"cor_children_per_women"=cor_children_per_women,"
ranked_cor_list2=order(-cor_list2)
cor_table2=data.frame(rank=1:length(ranked_cor_list2),
                      cor_name2=names(cor_list2)[ranked_cor_list2],
                      cor_value2=cor_list2[ranked_cor_list2])
cor_table2
```

```
##                                  rank                    cor_name2  cor_value2
## cor_water                           1                    cor_water  0.82771222
## cor_log_income                      2                cor_log_income  0.78791271
## cor_log_gdp                         3                   cor_log_gdp  0.77082161
## cor_log_total_health_spending       4 cor_log_total_health_spending  0.75511823
## cor_log_co2                         5                   cor_log_co2  0.74420258
```

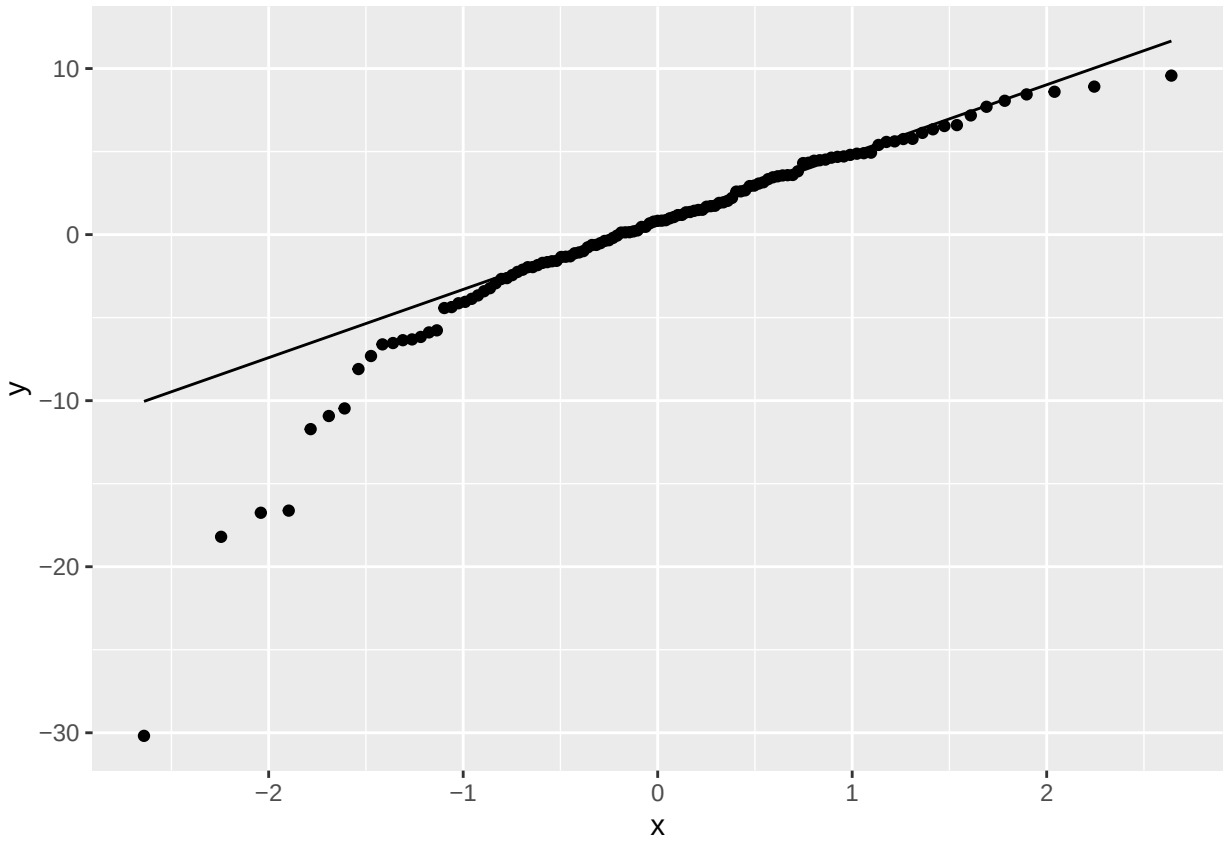
```
## cor_log_population_density       6    cor_log_population_density  0.14264523
## cor_log_population_total         7      cor_log_population_total  0.03515532
## cor_log_murder                   8                cor_log_murder -0.26757941
## cor_children_per_women           9        cor_children_per_women -0.78919875
## cor_child_mortality             10           cor_child_mortality -0.91385359
```

With the log transformation, there are still two predictors with a correlation of $|r| > 0.8$, but there are many more that are close ($|r| > 0.74$). We will consider them, and discard the rest. Thus, log(population density), log(population total), and log(murder) are thrown out as contenders for the single regression model.

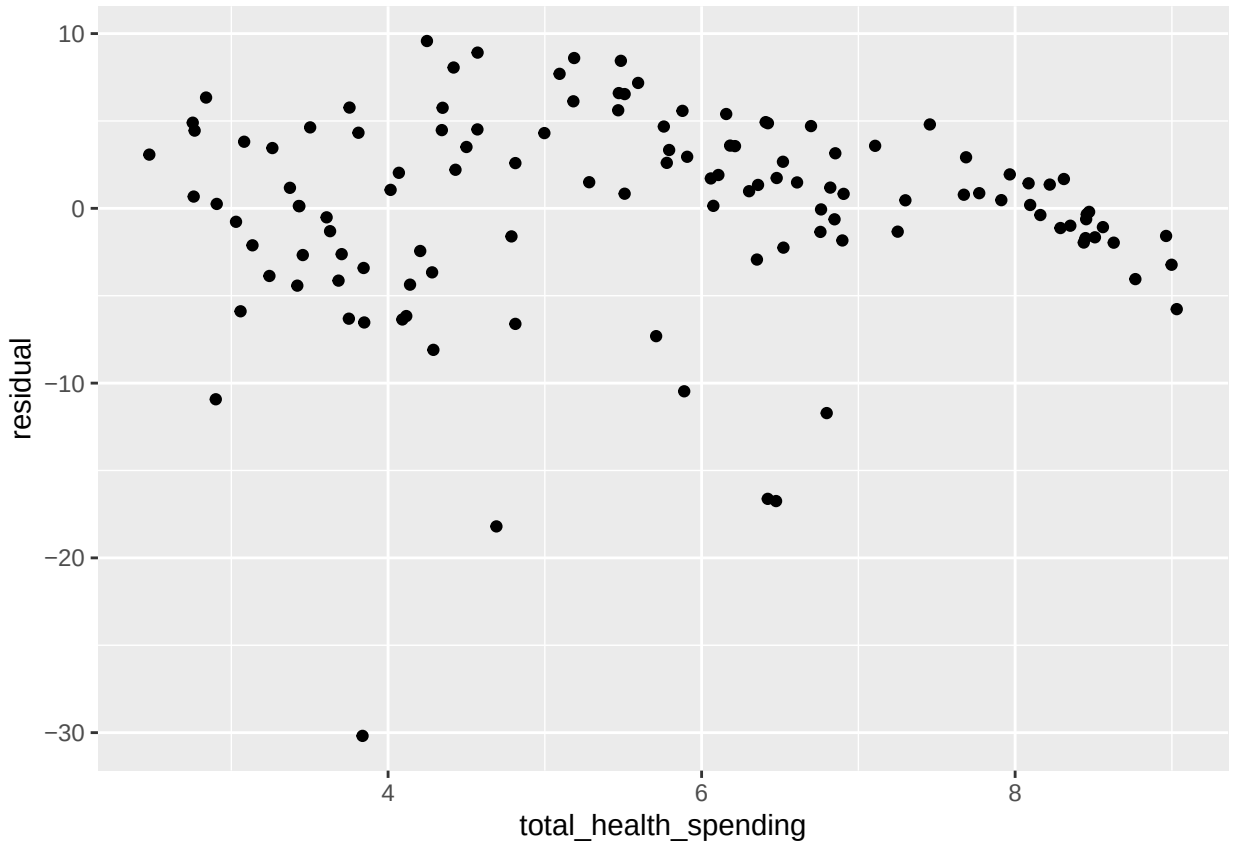
```
data_log=data
data_log$child_mortality=log(data_log$child_mortality)
data_log$gdp=log(data_log$gdp)
data_log$co2=log(data_log$co2)
data_log$murder=log(data_log$murder)
data_log$population_total=log(data_log$population_total)
data_log$population_density=log(data_log$population_density)
data_log$total_health_spending=log(data_log$total_health_spending)
data_log$income_per_person=log(data_log$income_per_person)
#Let's make some discrete variables
data_log$no_outlier_continent=ifelse((data_log$continent=="Europe" |data_log$continent== "Asia" | data_l

model1.7=lm(data=data_log,life_expectancy~total_health_spending)
get_regression_table(model1.7)
```

```
## # A tibble: 2 x 7
##   term                 estimate std_error statistic p_value lower_ci upper_ci
##   <chr>                   <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept               46.4      1.71       27.2       0    43.1     49.8
## 2 total_health_spending    4.23     0.288      14.7       0     3.66     4.80
```

```
points1.7=get_regression_points(model1.7)
ggplot(points1.7, aes(sample=residual)) + stat_qq() + stat_qq_line()
```

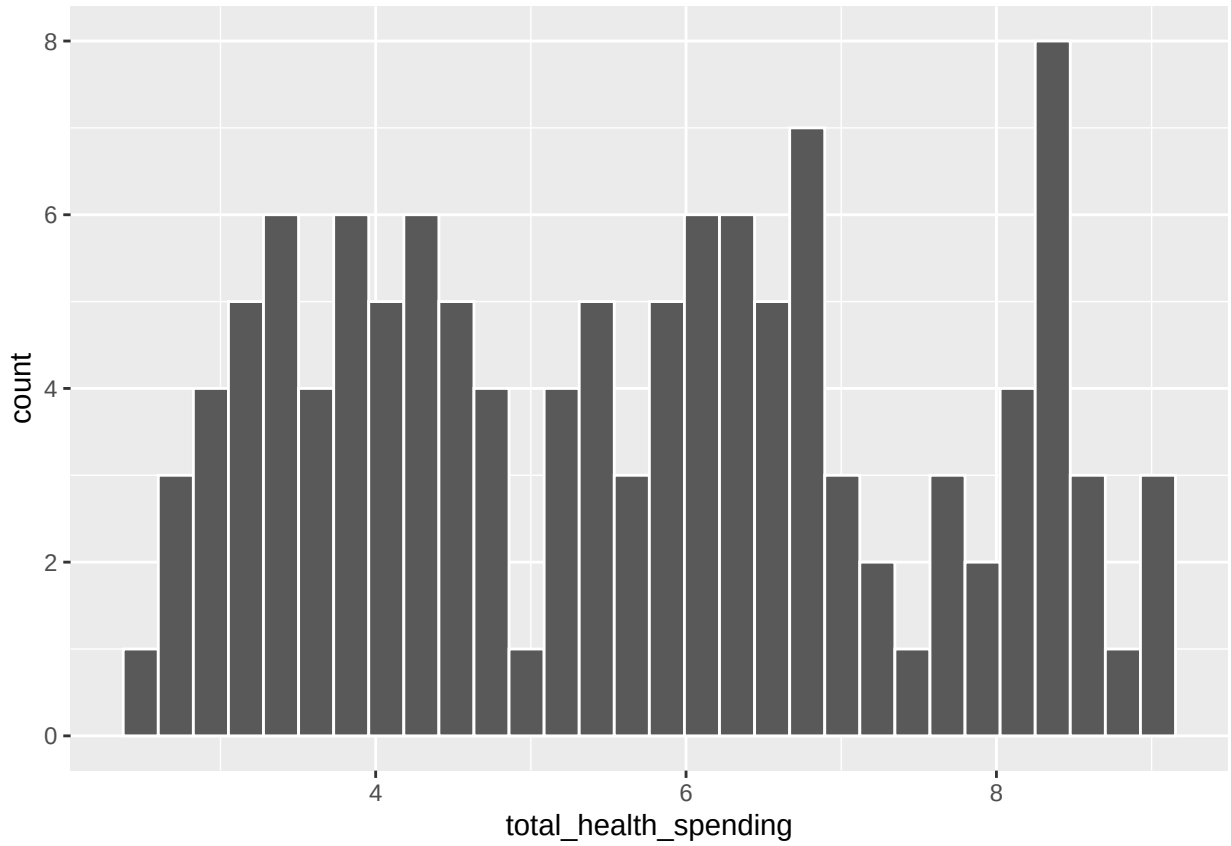



```
ggplot(points1.7, aes(x=total_health_spending, y=residual)) + geom_point()
```




```
ggplot(data=data_log, aes(x=total_health_spending)) + geom_histogram(color="white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
get_regression_summaries(model1.7)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.644         0.641  33.2  5.76  5.81      215.       0     1   121
```

Linearity / Mean 0 => Pos/Neg patterns in residual generally speaking, NOT VIOLATED

Independency: Highly correlated, VIOLATED

Normality: Histogram of residuals multimodal, VIOLATED ... Values under -0.5 do not fit QQPlot, VIOLATED

Equal Variaance: Resiual is suggested to be heteroskadciscuous (not cone-shaped), NOT VIOLATED

Overall, after comparing QQ plots, histograms of residuals, and residuals scatterplots, I have determined model1.7 (total health spending) was the best single regressor in predicting life expectancy in 2009.

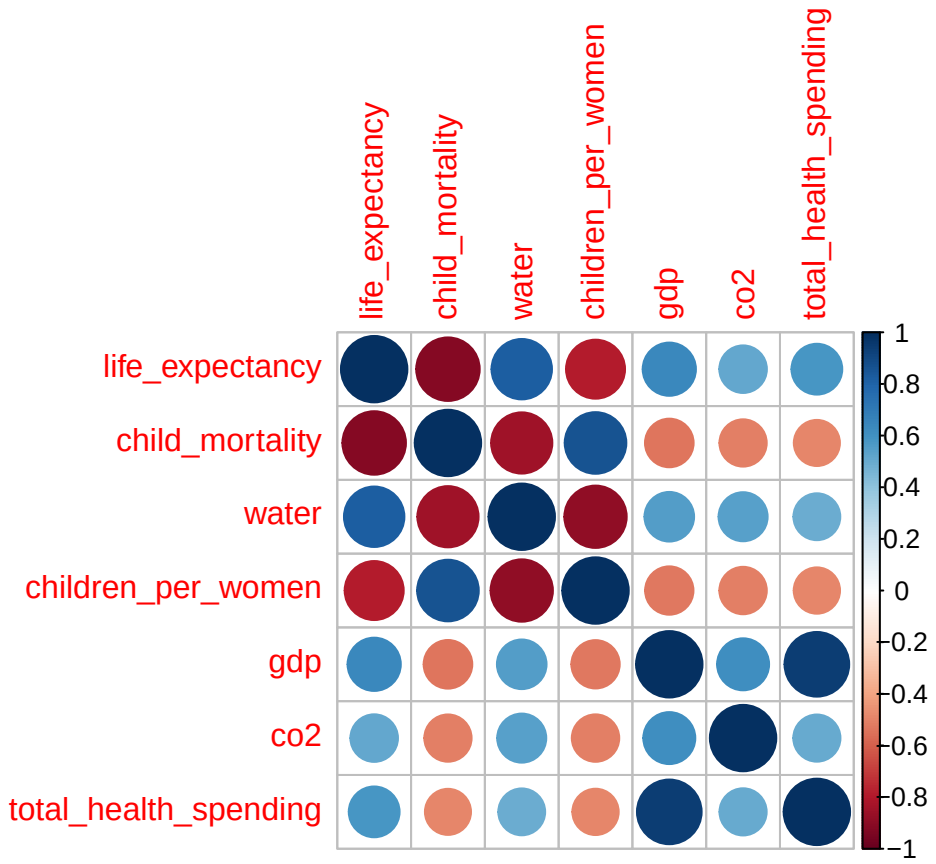
Multicolinearity Test

Before we run multiple regression models, let's test for multicollinearity to increase efficiency. Instead of checking each correlation individually like I did before, we can use a correlation matrix.

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
cor_matrix = cor(data[, c("life_expectancy", "child_mortality", "water","children_per_women", "gdp", "c
corrplot(cor_matrix)
```

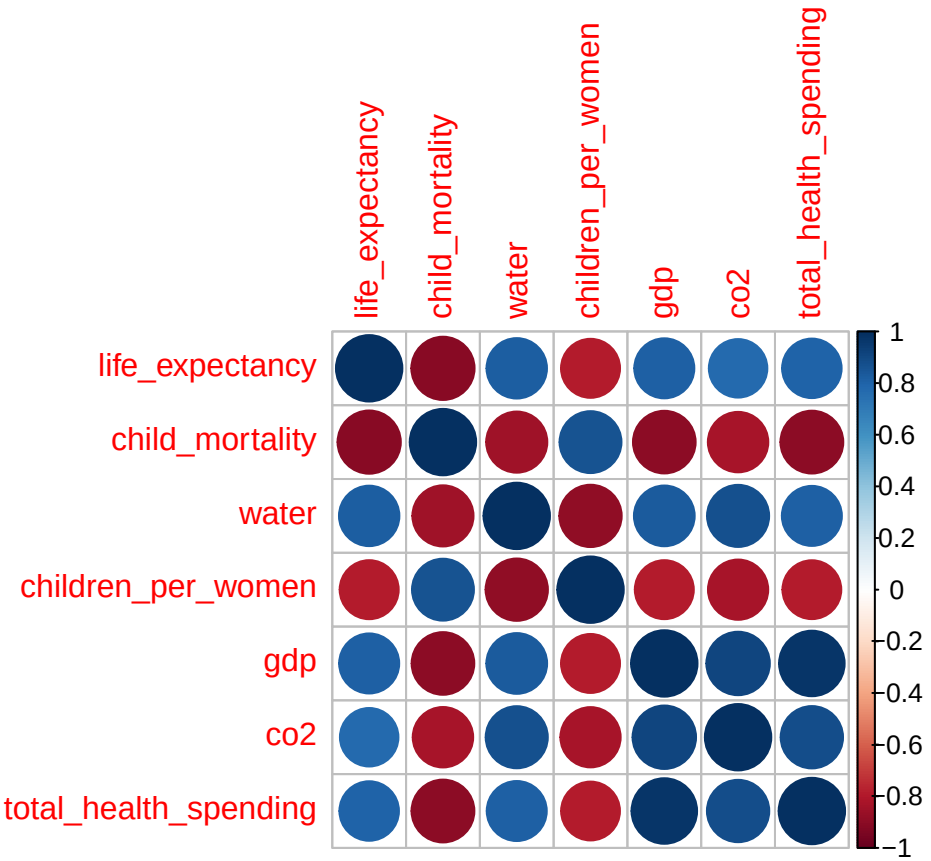


The matrix shows very

dark circles at the intersection of Total Health Spending and GDP & Children Per Woman and Water indicating that r $>$ 0.8, suggestive of colinearity.

Now, let's test for multicolinearity when the variables are logarithmically transformed.

```
cor_matrix_log = cor(data_log[, c("life_expectancy", "child_mortality", "water","children_per_women", "
corrplot(cor_matrix_log)
```



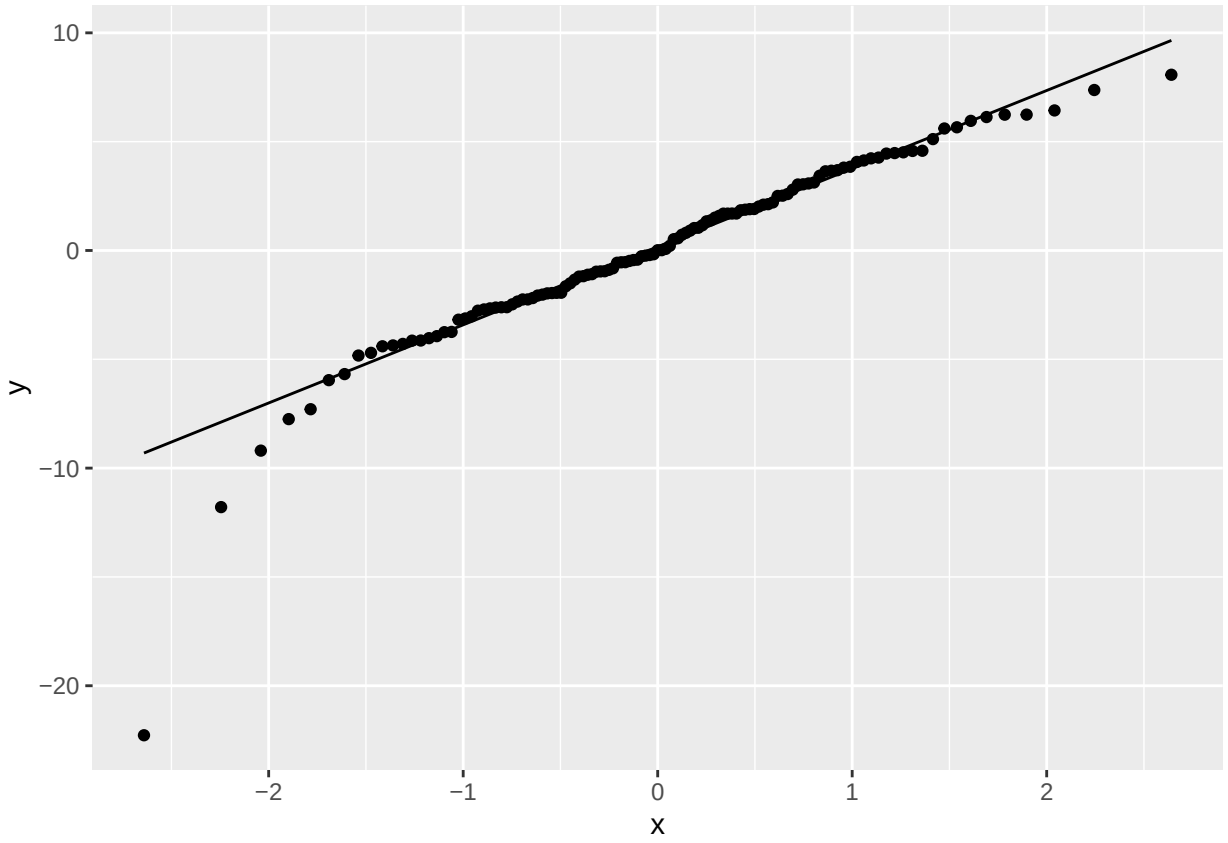
The shows very dark circles at the intersection of GDP and Child Mortality, Health Spending and Child Mortality, GDP & Total Health Spending, and Children Per Woman & Water. This will inform us in deciding potential predictor combinations.

Multiple Regression Models & Their Effectiveness

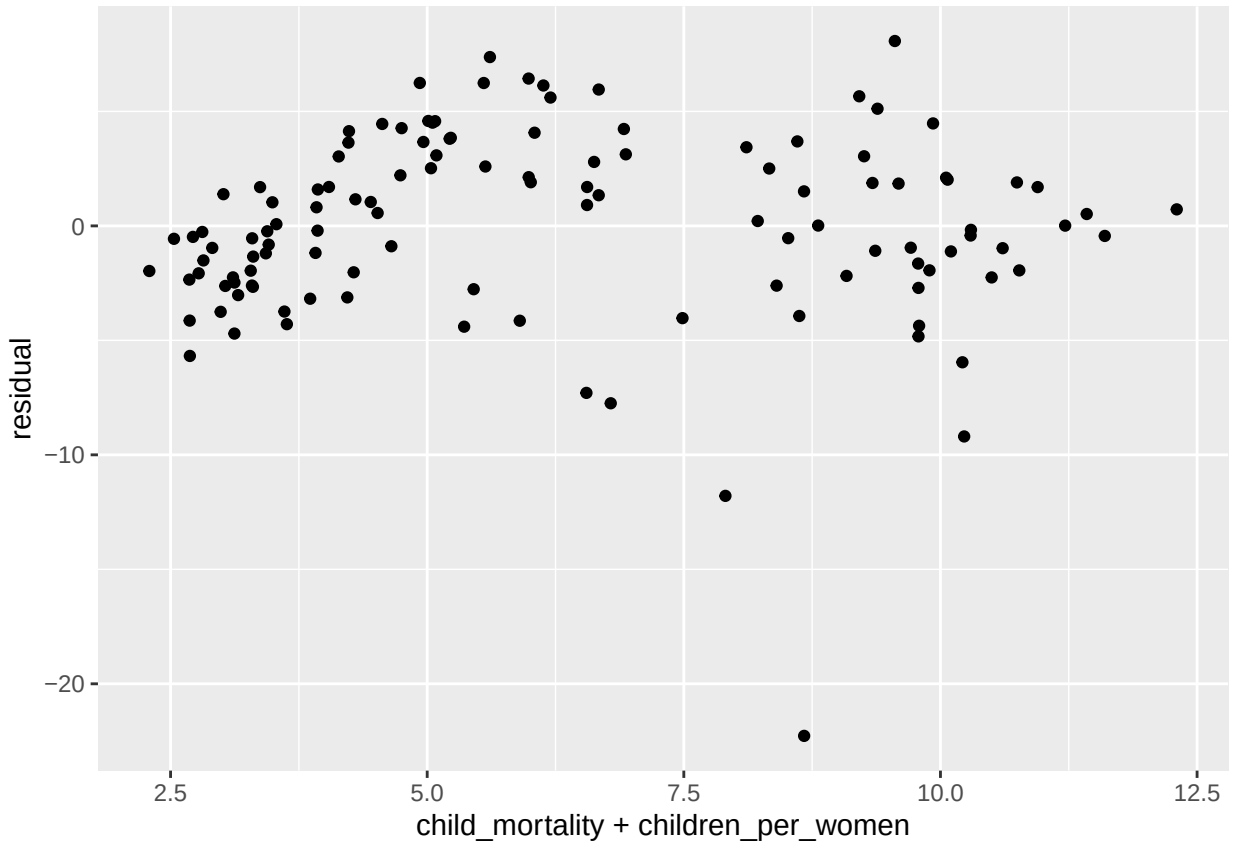
```
model2.11=lm(data=data_log,life_expectancy~child_mortality+children_per_women)
get_regression_table(model2.11)
```

```
## # A tibble: 3 x 7
##   term               estimate std_error statistic p_value lower_ci upper_ci
##   <chr>                 <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept              92.0      1.02      90.5    0        90.0     94.0
## 2 child_mortality        -6.94     0.606    -11.4    0        -8.14    -5.74
## 3 children_per_women     -0.042    0.467     -0.09   0.929    -0.966    0.882
```

```
points2.11=get_regression_points(model2.11)
ggplot(points2.11, aes(sample=residual)) + stat_qq() + stat_qq_line()
```

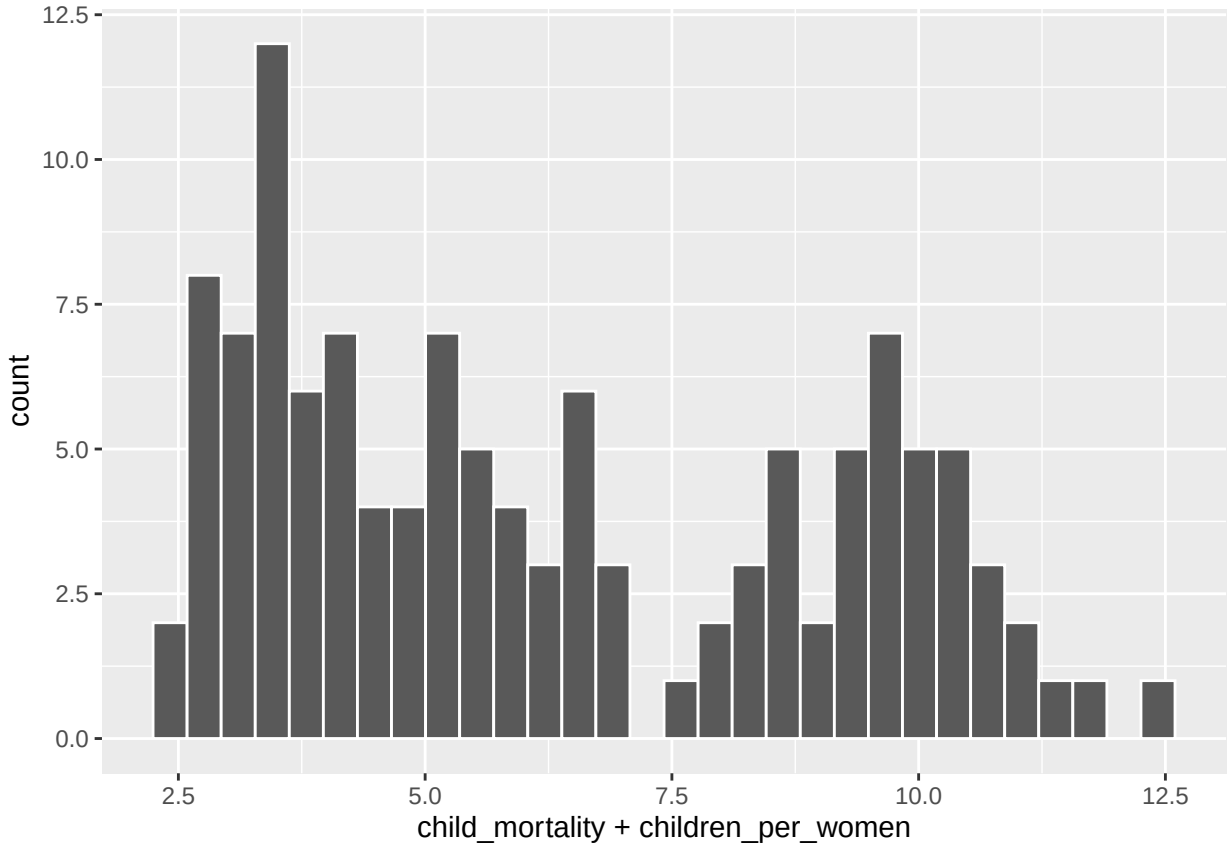



```
ggplot(points2.11, aes(x=child_mortality+children_per_women, y=residual)) + geom_point()
```




```
ggplot(data=data_log, aes(x=child_mortality+children_per_women)) + geom_histogram(color="white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
get_regression_summaries(model2.11)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.821         0.818  16.7  4.08  4.13      271.       0     2   121
```

3 assumptions are not violated so this model is the best one tested.