Group Project Notebook

Davide Danioni, Giorgio Micaletto, Martin Schweitzer

March 21, 2024

1 Group Project

1.1 Loading the libraries

We first start by checking that all the necessary libraries are all downloaded, and we then load them

[1]:
```
listofpackages <- c(
  "MASS",
  "WDI",
  "tidyr",
  "dplyr",
  "VIM",
  "httr",
  "jsonlite",
  "lmtest",
  "forecast",
  "nlme",
  "car",
  "ggplot2",
  "metafor",
  "maps",
  "tseries"
)
if (!require("democracyData")) {
  remotes::install_github("xmarquez/democracyData")
}
newpackages <- listofpackages[!(listofpackages %in% installed.
 ↪packages()[,"Package"])]
if(length(newpackages)) install.packages(newpackages,
  dependencies = TRUE,
  repos = "http://cran.us.r-project.org"
)
```

```
Loading required package: democracyData
```

[2]:
```
library(MASS)
library(WDI)
library(tidyr)
```

```r
library(dplyr)
library(VIM)
library(httr)
library(jsonlite)
library(lmtest)
library(forecast)
library(nlme)
library(car)
library(ggplot2)
library(metafor)
library(democracyData)
library(maps)
library(tseries)
```

1.2 Data Retrieval

We retrieve data from 1995 to 2023 on the following indicators:

- GDP per capita (constant 2010 US$) - NY.GDP.PCAP.CD
- Gross national savings (% of GNI) - NY.GNS.ICTR.ZS
- Population growth (annual %) - SP.POP.GROW
- Fertility rate (total births per woman) - SP.DYN.TFRT.IN
- CO2 emissions (metric tons per capita) - EN.ATM.CO2E.PC
- Political Stability and Lack of Violence - PV.PER.RNK.LOWER, PV.PER.RNK.UPPER
- Research and development expenditure (% of GDP) - GB.XPD.RSDV.GD.ZS
- Freedom status estimation from the Freedom House
- World map data from the maps library

1995 was chosen as the starting point as this is after the fall of the Soviet Union and the end of the Yugoslav wars

[3]:

```r
start_date <- 1995
end_date <- 2023

gdp_per_capita <- WDI(country = "all",
  "NY.GDP.PCAP.CD",
  start = start_date,
  end = end_date
)

saving_rate <- WDI(country = "all",
  "NY.GNS.ICTR.ZS",
  start = start_date,
  end = end_date
)

population_growth <- WDI(country = "all",
  "SP.POP.GROW",
  start = start_date,
  end = end_date
)

fertility <- WDI(country = "all",
  "SP.DYN.TFRT.IN",
  start = start_date,
  end = end_date
)

co2_emission <- WDI(country = "all",
  "EN.ATM.CO2E.PC",
  start = start_date,
  end = end_date
)

pol_stability_lower <- WDI(country = "all",
  "PV.PER.RNK.LOWER",
  start = start_date,
  end = end_date
)

pol_stability_upper <- WDI(country = "all",
  "PV.PER.RNK.UPPER",
  start = start_date,
  end = end_date
)

research <- WDI(country = "all",
  "GB.XPD.RSDV.GD.ZS",
  start = start_date,
  end = end_date
)

dem_data  <- download_fh()

world_map <- map_data("world")

print("Downloaded the dataset")
```

```
Downloading  data...
[1] "Downloaded the dataset"
```

We create dummy variables for `Free` and `Partially Free` countries

[4]:

```
dem_data <- dem_data %>% select(fh_country, year, status)
dem_data$dummy_PF <- ifelse(dem_data$status == "PF", 1, 0)
dem_data$dummy_F <- ifelse(dem_data$status == "F", 1, 0)
colnames(dem_data)[1] <- "country"
dem_data$status <- NULL
```

We create two dummy variables: one for countries whose centroid distance from the equator is ≥ 60 and one for countries whose centroids distance from the equator is ≥ 30 and < 60

[5]:

```
country_centroids <- aggregate(
  cbind(long, lat) ~ region,
  data = world_map,
  FUN = function(x) median(range(x))
)
colnames(country_centroids) <- c("country", "longitude", "latitude")
country_centroids$longitude <- NULL
country_centroids$dummy_30_60 = ifelse(
  abs(country_centroids$latitude) >= 30 & abs(country_centroids$latitude) < 60,
  1,
  0
)
country_centroids$dummy_60_plus = ifelse(
  abs(country_centroids$latitude) >= 60,
  1,
  0
)
country_centroids$latitude <- NULL
```

We now merge and clean all the datasets

[6]:

```
data_regression <- merge.data.frame(gdp_per_capita, saving_rate)
data_regression <- merge.data.frame(data_regression, population_growth)
data_regression <- merge.data.frame(data_regression, co2_emission)
data_regression <- merge.data.frame(data_regression, fertility)
data_regression <- merge.data.frame(data_regression, pol_stability_lower)
data_regression <- merge.data.frame(data_regression, research)
data_regression <- merge.data.frame(data_regression, pol_stability_upper)
data_regression <- merge.data.frame(data_regression, country_centroids)
last_year_observed <- max(data_regression$year)
dem_data <- dem_data %>% filter(year >= start_date & year <= last_year_observed)
data_regression <- merge.data.frame(data_regression, dem_data)
subsetted_data_regression = subset(data_regression, year == last_year_observed)
in_subset <- data_regression$country %in% subsetted_data_regression$country
data_regression <- data_regression[in_subset, ]
print("Merged the dataset")
```

```
[1] "Merged the dataset"
```

Sanity check to be sure that only countries are in the dataframe

[7]:

```
print(paste("Number of unique countries:",␣
 ↪length(unique(data_regression$country))))
print(unique(data_regression$country))
```

```
[1] "Number of unique countries: 163"
  [1] "Afghanistan"              "Albania"
  [3] "Algeria"                  "Andorra"
  [5] "Angola"                   "Argentina"
  [7] "Armenia"                  "Australia"
  [9] "Austria"                  "Azerbaijan"
 [11] "Bahrain"                  "Bangladesh"
 [13] "Barbados"                 "Belarus"
 [15] "Belgium"                  "Belize"
 [17] "Benin"                    "Bhutan"
 [19] "Bolivia"                  "Bosnia and Herzegovina"
 [21] "Botswana"                 "Brazil"
 [23] "Bulgaria"                 "Burkina Faso"
 [25] "Burundi"                  "Cambodia"
 [27] "Cameroon"                 "Canada"
 [29] "Central African Republic" "Chad"
 [31] "Chile"                    "China"
 [33] "Colombia"                 "Comoros"
 [35] "Costa Rica"               "Croatia"
 [37] "Cuba"                     "Cyprus"
 [39] "Denmark"                  "Djibouti"
 [41] "Dominica"                 "Dominican Republic"
 [43] "Ecuador"                  "El Salvador"
 [45] "Equatorial Guinea"        "Eritrea"
 [47] "Estonia"                  "Ethiopia"
 [49] "Fiji"                     "Finland"
 [51] "France"                   "Gabon"
 [53] "Georgia"                  "Germany"
 [55] "Ghana"                    "Greece"
 [57] "Grenada"                  "Guatemala"
 [59] "Guinea"                   "Guinea-Bissau"
 [61] "Guyana"                   "Haiti"
 [63] "Honduras"                 "Hungary"
 [65] "Iceland"                  "India"
 [67] "Indonesia"                "Iraq"
 [69] "Ireland"                  "Israel"
 [71] "Italy"                    "Jamaica"
 [73] "Japan"                    "Jordan"
 [75] "Kazakhstan"               "Kenya"
 [77] "Kiribati"                 "Kosovo"
 [79] "Kuwait"                   "Latvia"
 [81] "Lebanon"                  "Lesotho"
 [83] "Liberia"                  "Libya"
 [85] "Liechtenstein"            "Lithuania"
 [87] "Luxembourg"               "Madagascar"
 [89] "Malawi"                   "Malaysia"
 [91] "Maldives"                 "Mali"
 [93] "Malta"                    "Marshall Islands"
 [95] "Mauritania"               "Mauritius"
 [97] "Mexico"                   "Moldova"
 [99] "Monaco"                   "Mongolia"
[101] "Montenegro"               "Morocco"
[103] "Mozambique"               "Myanmar"
[105] "Namibia"                  "Nauru"
[107] "Nepal"                    "Netherlands"
[109] "New Zealand"              "Nicaragua"
[111] "Niger"                    "Nigeria"
[113] "North Macedonia"          "Norway"
[115] "Oman"                     "Pakistan"
[117] "Palau"                    "Panama"
[119] "Papua New Guinea"         "Paraguay"
[121] "Peru"                     "Philippines"
[123] "Poland"                   "Portugal"
[125] "Qatar"                    "Romania"
[127] "Rwanda"                   "Samoa"
[129] "San Marino"               "Sao Tome and Principe"
[131] "Saudi Arabia"             "Senegal"
[133] "Serbia"                   "Seychelles"
[135] "Sierra Leone"             "Singapore"
[137] "Slovenia"                 "Solomon Islands"
[139] "Somalia"                  "South Africa"
[141] "South Sudan"              "Spain"
[143] "Sri Lanka"                "Sudan"
[145] "Suriname"                 "Sweden"
[147] "Switzerland"              "Tajikistan"
[149] "Tanzania"                 "Thailand"
[151] "Timor-Leste"              "Togo"
[153] "Tonga"                    "Tunisia"
[155] "Turkmenistan"             "Uganda"
[157] "Ukraine"                  "United Arab Emirates"
[159] "Uruguay"                  "Uzbekistan"
[161] "Vanuatu"                  "Zambia"
[163] "Zimbabwe"
```

To remove any NaNs present in the data, we use k-Nearest Neighbours(kNN), a non-parametric model that imputes the value of a point based on the average values of the k nearest points in the dataset. Mathematically what it does is the following:

For an observation X_i with missing data, calculate the distance between X_i and all other observations in the dataset that have a value for the missing feature. The distance R uses is euclidian and in an n-dimensional space is given by:

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (X_{ik} - X_{jk})^2} \tag{1}$$

Then the k observations closest to X_i based on the calculated distances are selected and the missing value in X_i are substituted with the mean of the observed values from the k nearest neighbors. By using the most similar observations for imputation, kNN ensures that the imputed values are more contextually appropriate, preserving the data's underlying structure and relationships.

We chose kNN because, unlike model-based approaches that require assumptions about the distribution of data or the relationship between variables, kNN's non-parametric nature makes it robust to deviations from such assumptions.

[8]:

```
data_regression <- kNN(data_regression, k = 10)
```

As there are a lot of extreme values in the political stability estimate, we decided it was best to aggregate them using a weighted average, where the weights are calculated as followed

- For the lower bound it is: $\frac{1}{2} + \frac{political\ stability_*}{200}$, which means that it starts at $\frac{1}{2}$ and linearly increases up to 1, when $political\ stability_* = 100$ where $political\ stability_*$ is the lower bound of the political stability estimation
- For the upper bound it is: $\frac{1}{2} + \frac{100 - political\ stability^*}{200}$, which means that it starts at 1 and linearly decreases up to $\frac{1}{2}$, when $political\ stability^* = 100$ where $political\ stability^*$ is the upper bound of the political stability estimation

This is done because linear models can struggle with extreme values, and these extremes can skew the estimation, making it difficult to accurately assess the impact of political stability on GDP.

[10]:

```
weighted_average <- function(lower, upper) {
  weight_l <- 0.5 + (lower / 200)
  weight_u <- 0.5 + ((100 - upper) / 200)
  rtv <- (lower * weight_l + upper * weight_u) / (weight_l + weight_u)
  return(rtv)
}
data_regression$pol_stability <- mapply(
  weighted_average,
  data_regression$PV.PER.RNK.LOWER,
  data_regression$PV.PER.RNK.UPPER
)
```

We now ensure that all the necessary columns are in numerical form and remove any that are unnecessary for the assignment

[11]:

```
data_regression$gdp <- as.numeric(data_regression$NY.GDP.PCAP.CD)
data_regression$saving <- as.numeric(data_regression$NY.GNS.ICTR.ZS)
data_regression$pop_growth <- as.numeric(data_regression$SP.POP.GROW)
data_regression$fertility <- as.numeric(data_regression$SP.DYN.TFRT.IN)
data_regression$co2_emission <- as.numeric(data_regression$EN.ATM.CO2E.PC)
data_regression$pol_stability <- as.numeric(data_regression$pol_stability)
```

```
data_regression$research <- as.numeric(data_regression$GB.XPD.RSDV.GD.ZS)
```

[12]:
```
data_regression$iso2c <- NULL
data_regression$iso3c <- NULL
data_regression$NY.GDP.PCAP.CD <- NULL
data_regression$NY.GNS.ICTR.ZS <- NULL
data_regression$SP.POP.GROW <- NULL
data_regression$SP.DYN.TFRT.IN <- NULL
data_regression$EN.ATM.CO2E.PC <- NULL
data_regression$PV.PER.RNK.LOWER <- NULL
data_regression$PV.PER.RNK.UPPER <- NULL
data_regression$GB.XPD.RSDV.GD.ZS <- NULL
data_regression$country_imp <- NULL
data_regression$NY.GDP.PCAP.CD_imp <- NULL
data_regression$NY.GNS.ICTR.ZS_imp <- NULL
data_regression$SP.POP.GROW_imp <- NULL
data_regression$SP.DYN.TFRT.IN_imp <- NULL
data_regression$EN.ATM.CO2E.PC_imp <- NULL
data_regression$iso2c_imp <- NULL
data_regression$iso3c_imp <- NULL
data_regression$year_imp <- NULL
data_regression$PV.PER.RNK.LOWER_imp <- NULL
data_regression$PV.PER.RNK.UPPER_imp <- NULL
data_regression$GB.XPD.RSDV.GD.ZS_imp <- NULL
data_regression$dummy_F_imp <- NULL
data_regression$dummy_PF_imp <- NULL
data_regression$dummy_30_60_imp <- NULL
data_regression$dummy_60_plus_imp <- NULL
```

Taking the logarithms, removing any $\pm\infty$ generated by taking the logarithm and imputing the NaNs created

[13]:
```
data_regression$gdp <- log(data_regression$gdp)
data_regression$saving <- log(data_regression$saving)
data_regression$pop_growth <- log(data_regression$pop_growth)
data_regression$co2_emission <- log(data_regression$co2_emission)
data_regression$fertility <- log(data_regression$fertility)
data_regression$research <- log(data_regression$research)
```

```
Warning message in log(data_regression$saving):
‘‘NaNs produced’’
Warning message in log(data_regression$pop_growth):
‘‘NaNs produced’’
```

[14]:
```
data_regression$co2_emission  <- ifelse(
  is.infinite(
    data_regression$co2_emission
  ),
  NA,
  data_regression$co2_emission
)
data_regression$fertility  <- ifelse(
  is.infinite(
    data_regression$fertility
  ),
  NA,
  data_regression$fertility
)
data_regression$research  <- ifelse(
  is.infinite(
    data_regression$research
  ),
  NA,
  data_regression$research
)
```

[15]:

```
data_regression <- kNN(data_regression, k = 10)
data_regression$country_imp <- NULL
data_regression$saving_imp <- NULL
data_regression$pop_growth_imp <- NULL
data_regression$gdp_imp <- NULL
data_regression$year_imp <- NULL
data_regression$co2_emission_imp <- NULL
data_regression$fertility_imp <- NULL
data_regression$pol_stability_imp <- NULL
data_regression$research_imp <- NULL
data_regression$dummy_PF_imp <- NULL
data_regression$dummy_F_imp <- NULL
data_regression$dummy_30_60_imp <- NULL
data_regression$dummy_60_plus_imp <- NULL
```

Sanity check to be sure that our dataset follows our expectations

[16]:

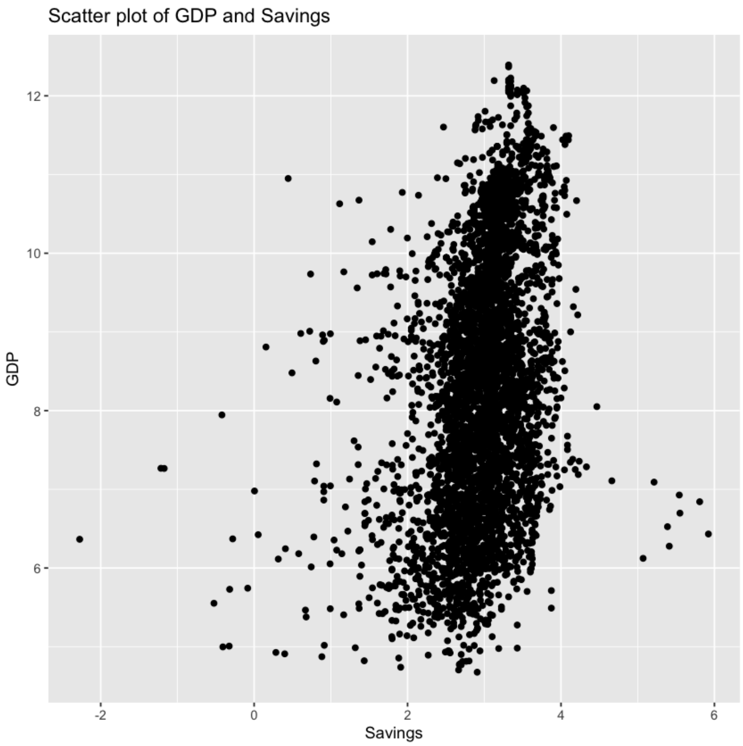
```
print(colnames(data_regression))
print(summary(data_regression))
```

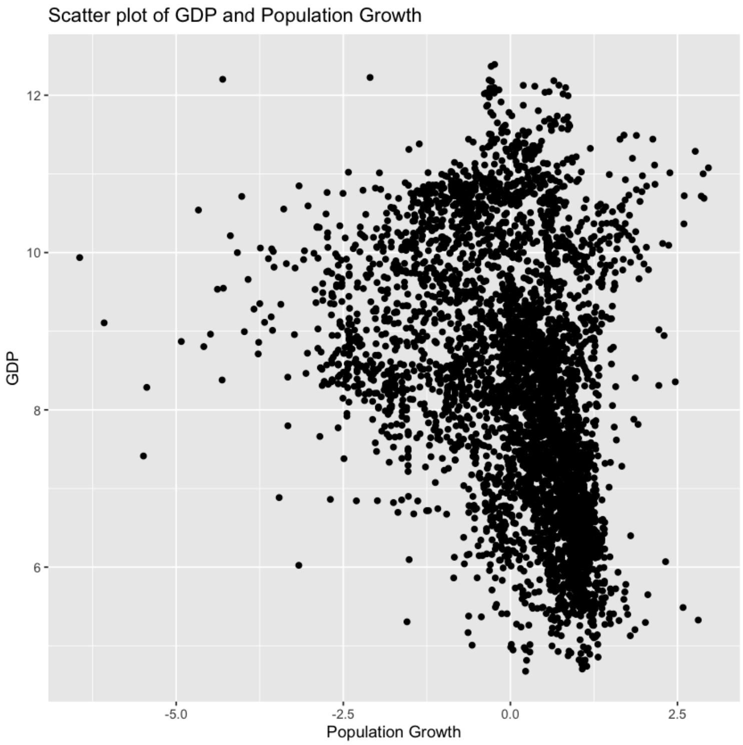
```
 [1] "country"       "year"          "dummy_30_60"   "dummy_60_plus"
 [5] "dummy_PF"      "dummy_F"       "pol_stability" "gdp"
 [9] "saving"        "pop_growth"    "fertility"     "co2_emission"
[13] "research"
   country               year       dummy_30_60      dummy_60_plus
 Length:4508        Min.   :1995   Min.   :0.0000   Min.   :0.00000
 Class :character   1st Qu.:2002   1st Qu.:0.0000   1st Qu.:0.00000
 Mode  :character   Median :2009   Median :0.0000   Median :0.00000
                    Mean   :2009   Mean   :0.3647   Mean   :0.03106
                    3rd Qu.:2016   3rd Qu.:1.0000   3rd Qu.:0.00000
                    Max.   :2022   Max.   :1.0000   Max.   :1.00000
   dummy_PF          dummy_F       pol_stability       gdp
 Min.   :0.0000   Min.   :0.0000   Min.   : 0.00   Min.   : 4.676
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:28.82   1st Qu.: 6.969
 Median :0.0000   Median :0.0000   Median :47.78   Median : 8.268
 Mean   :0.3321   Mean   :0.4439   Mean   :48.03   Mean   : 8.271
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:70.74   3rd Qu.: 9.470
 Max.   :1.0000   Max.   :1.0000   Max.   :96.35   Max.   :12.392
     saving          pop_growth          fertility        co2_emission
 Min.   :-2.274   Min.   :-6.44493   Min.   :-0.1625   Min.   :-3.8263
 1st Qu.: 2.724   1st Qu.:-0.42255   1st Qu.: 0.5365   1st Qu.:-0.5622
 Median : 3.040   Median : 0.34207   Median : 0.9365   Median : 0.8563
 Mean   : 2.971   Mean   : 0.08756   Mean   : 0.9857   Mean   : 0.5316
 3rd Qu.: 3.292   3rd Qu.: 0.86320   3rd Qu.: 1.4237   3rd Qu.: 1.8252
 Max.   : 5.922   Max.   : 2.96323   Max.   : 2.0510   Max.   : 3.8640
    research
 Min.   :-5.2140
 1st Qu.:-1.6810
 Median :-1.1316
 Mean   :-1.0095
 3rd Qu.:-0.3806
 Max.   : 1.7414
```

1.3 Data Exploration

We will now explore the distributions of the various variables to better understand the data we have at hand

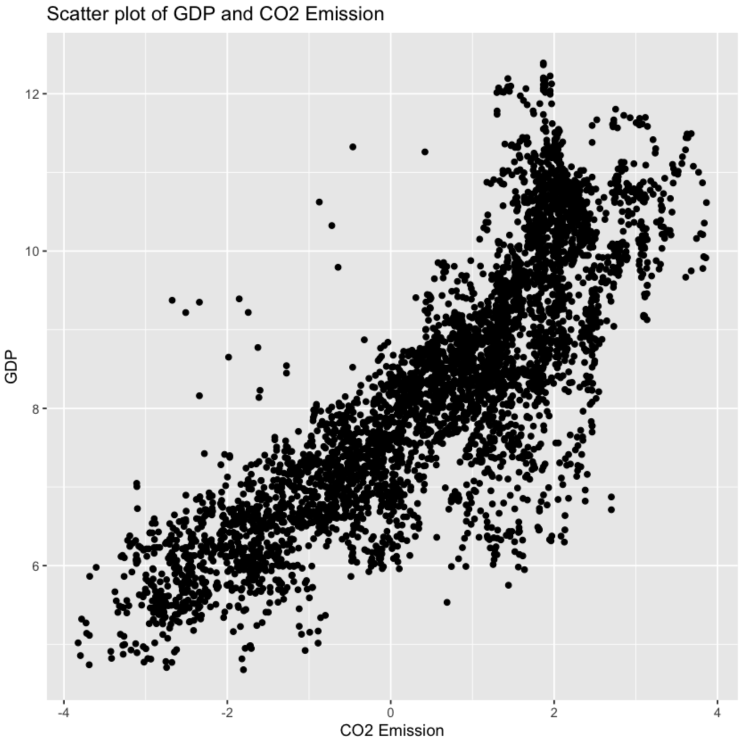
[17]:
```
ggplot(data_regression, aes(x = saving, y = gdp)) +
  geom_point() +
  labs(title = "Scatter plot of GDP and Savings",
       x = "Savings",
       y = "GDP")
ggplot(data_regression, aes(x = pop_growth, y = gdp)) +
  geom_point() +
  labs(title = "Scatter plot of GDP and Population Growth",
       x = "Population Growth",
       y = "GDP")
```

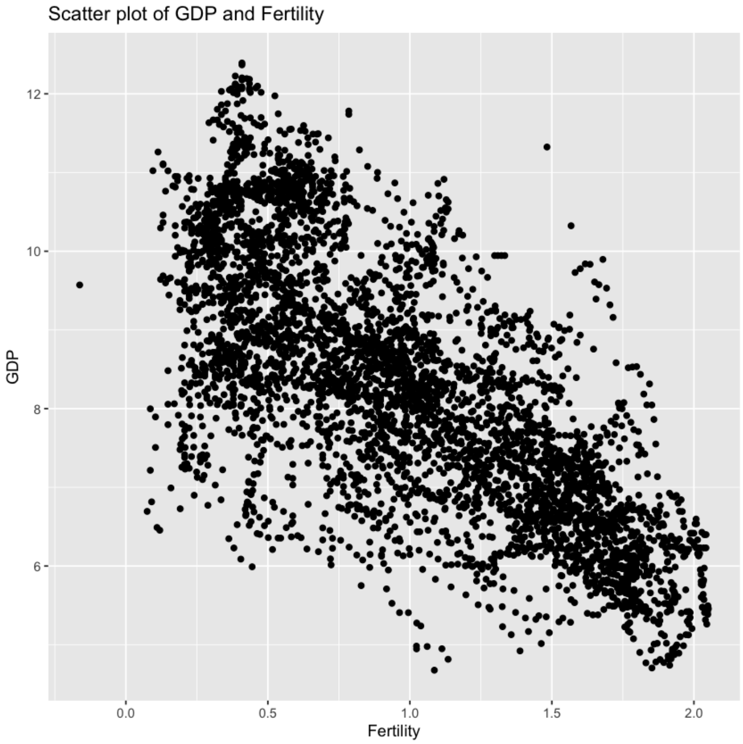




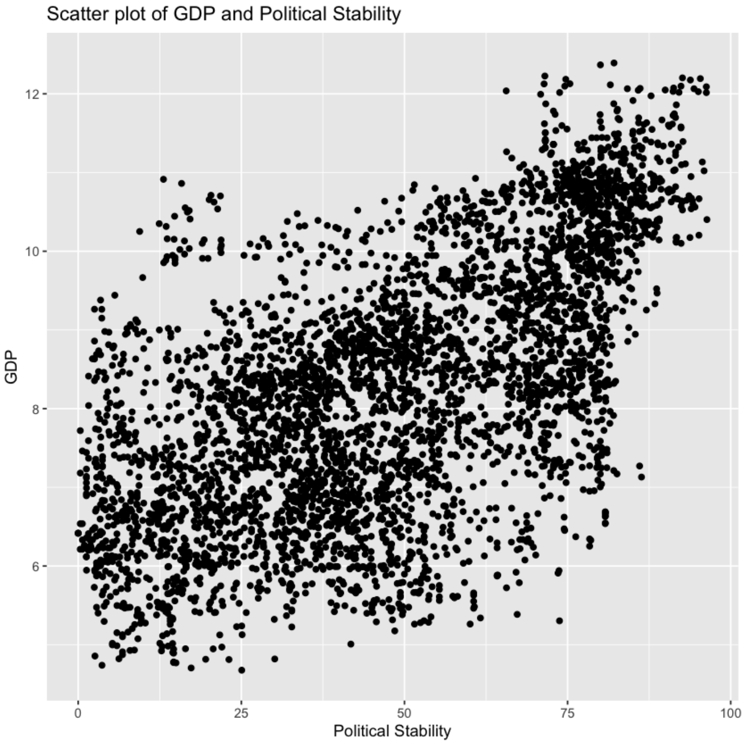
[18]:

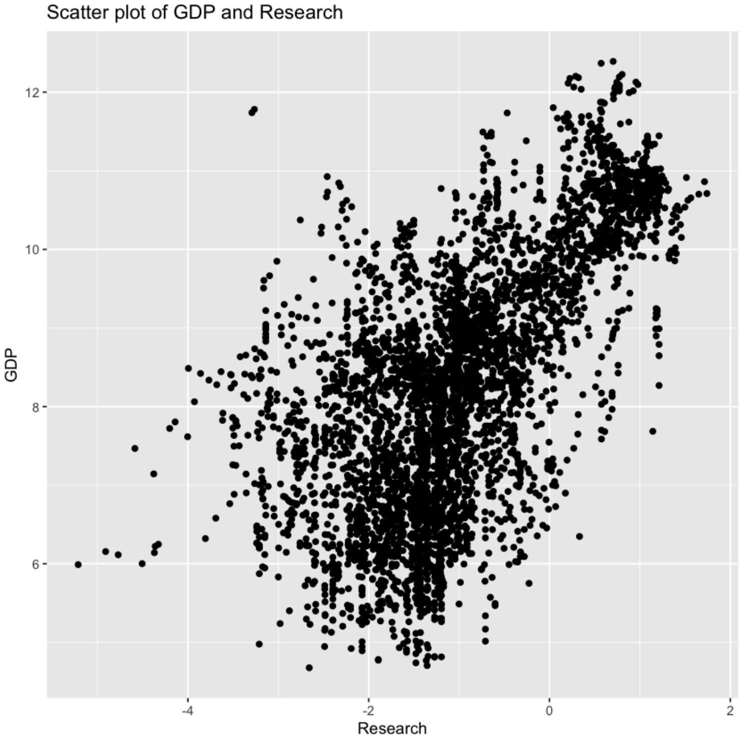
```r
ggplot(data_regression, aes(x = co2_emission, y = gdp)) +
  geom_point() +
  labs(title = "Scatter plot of GDP and CO2 Emission",
       x = "CO2 Emission",
       y = "GDP")
ggplot(data_regression, aes(x = fertility, y = gdp)) +
  geom_point() +
  labs(title = "Scatter plot of GDP and Fertility",
       x = "Fertility",
       y = "GDP")
```





```
[19]: ggplot(data_regression, aes(x = pol_stability, y = gdp)) +
        geom_point() +
        labs(title = "Scatter plot of GDP and Political Stability",
             x = "Political Stability",
             y = "GDP")
      ggplot(data_regression, aes(x = research, y = gdp)) +
        geom_point() +
        labs(title = "Scatter plot of GDP and Research",
             x = "Research",
             y = "GDP")
```





From the plots we can infer that: - GDP and Savings are mostly uncorrelated, with most observations of Saving being around 3 - There seems to be a correlation between GDP and the other predictors, with varying degree of variance

From here we can infer that all predictors display significant skewness

1.4 Linear Regression

We now do a linear regression. We start from the model given by the assignment, i.e.

$$\log(gdp) = \log(savings) + \log(population\ growth)$$

[30]:

```
model <- lm(gdp ~ saving + pop_growth, data = data_regression)
```

We now check the summary of the model

[31]:

```
summary(model)
```

```
Call:
lm(formula = gdp ~ saving + pop_growth, data = data_regression)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4033 -1.0207 -0.1898  0.9567  5.4765 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.38067    0.11555   46.56   <2e-16 ***
saving       0.98910    0.03823   25.87   <2e-16 ***
pop_growth  -0.55795    0.01972  -28.30   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.396 on 4505 degrees of freedom
Multiple R-squared:  0.2517,	Adjusted R-squared:  0.2514 
F-statistic: 757.7 on 2 and 4505 DF,  p-value: < 2.2e-16
```

As the residuals are very spread out we check that there's no outliers by running the Breusch-Pagan test. The Breusch-Pagan test for homoscedasticity is designed to assess the presence of heteroscedasticity in a regression model. The null hypothesis (H_0) of the test is that the variance of the errors (σ_i^2) is constant across observations, implying homoscedasticity ($\sigma_i^2 = \sigma^2$). The alternative hypothesis (H_1) suggests the existence of a relationship between the variance of the errors and one or more explanatory variables

$$\sigma_i^2 = f(\gamma + \delta Z) \tag{2}$$

where Z could be any subset of the explanatory variables in the model, their transformations, or even different variables not included in the regression model.

[32]:

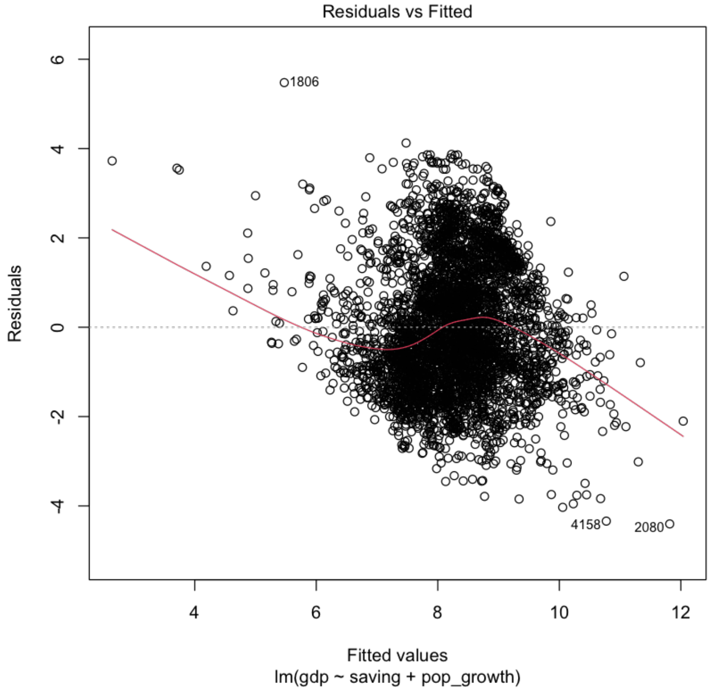
```
bptest(model)
```

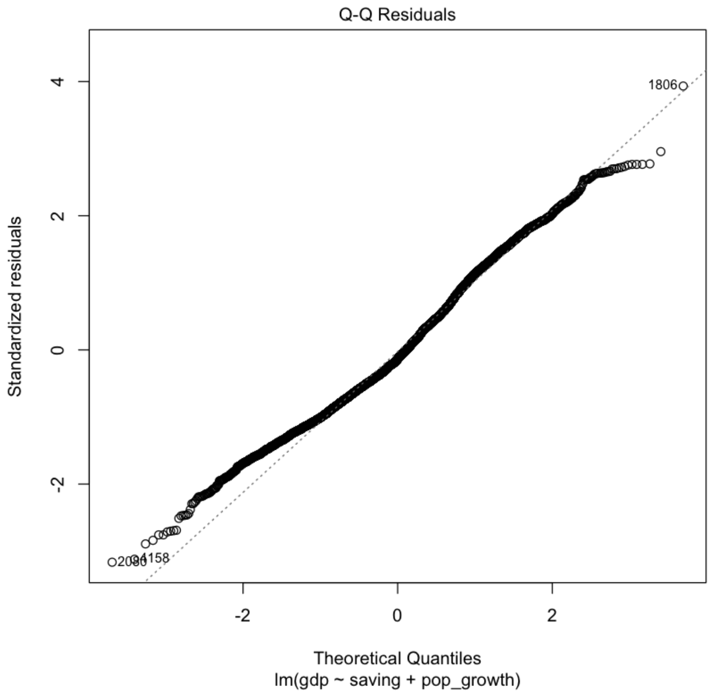
```
	studentized Breusch-Pagan test

data:  model
```

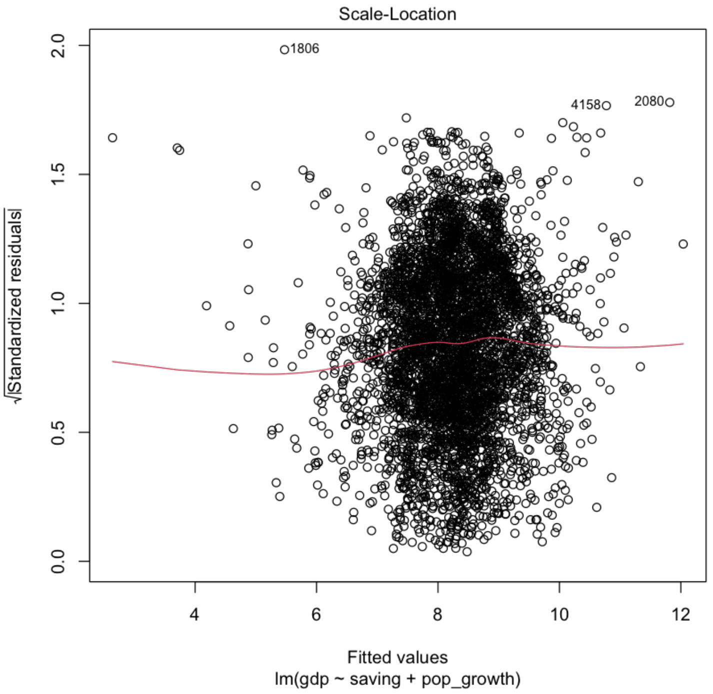
```
BP = 125.53, df = 2, p-value < 2.2e-16
```

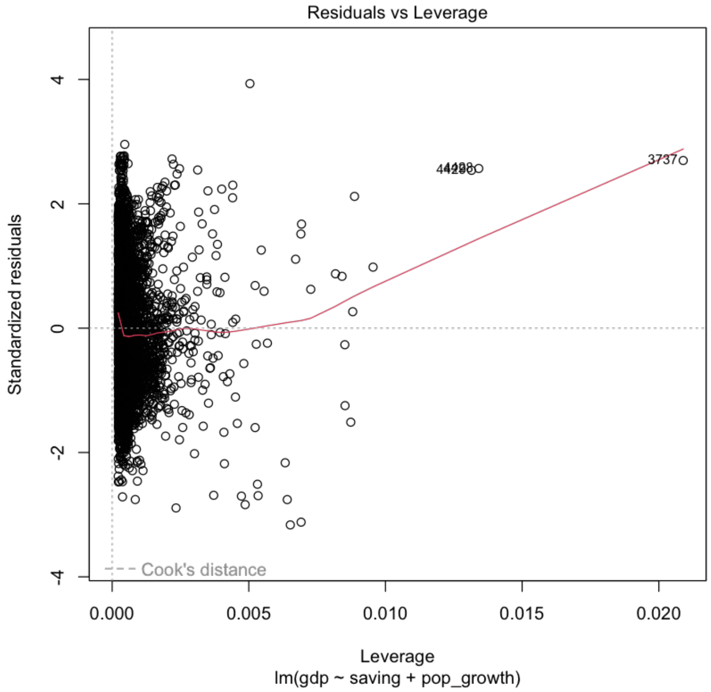
As the null hypothesis is rejected, we plot the model to understand what may be causing the problem

[33]:
```
plot(model)
```




Q-Q Residuals
1806
2080
4158
Standardized residuals
4
2
0
-2
-2
0
2
Theoretical Quantiles
lm(gdp ~ saving + pop_growth)


Scale-Location
1806
4158
2080
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
4
6
8
10
12
Fitted values
lm(gdp ~ saving + pop_growth)



The spread observed in both the `Residuals vs Fitted`plot and `Scale Location`plot follow what we observed in the scatter plot of `GDP and Savings`.

For this reason we change the model to

$$\log(gdp) = \log(co2\ emissions) + \log(fertility) + political\ stability + \log(research) + \log(population\ growth)$$

Accounting for the dummy variables created

[34]:
```
model <- lm(gdp ~ co2_emission + fertility + pol_stability + research +␣
 ↪pop_growth + dummy_F + dummy_PF + dummy_60_plus + dummy_30_60, data =␣
 ↪data_regression)
```

Now we check the summary of the model and run again the BP Test

[35]:
```
summary(model)
bptest(model)
```

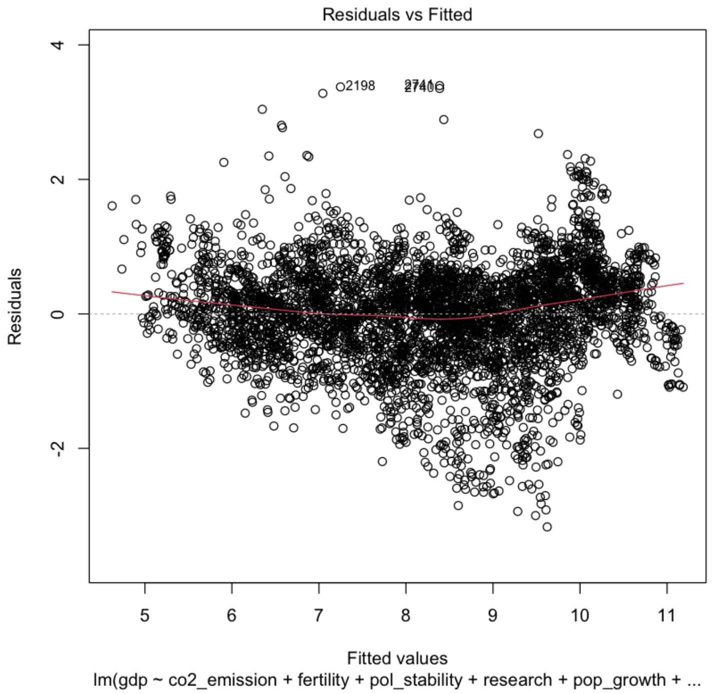
```
Call:
lm(formula = gdp ~ co2_emission + fertility + pol_stability +
    research + pop_growth + dummy_F + dummy_PF + dummy_60_plus +
    dummy_30_60, data = data_regression)

Residuals:
    Min      1Q  Median      3Q     Max
-3.1681 -0.4047  0.0662  0.4601  3.3950
```

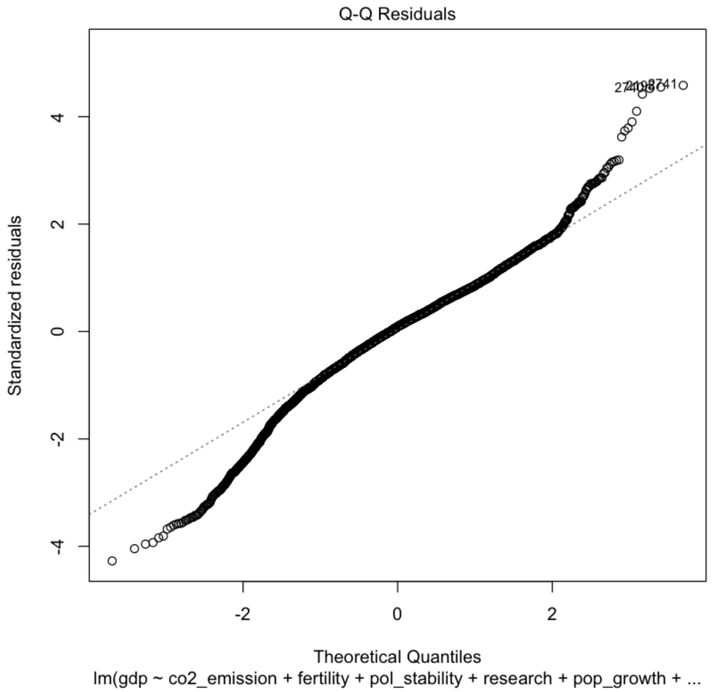
```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    8.3503332  0.0677577 123.238  < 2e-16 ***
co2_emission   0.5379689  0.0119790  44.909  < 2e-16 ***
fertility     -0.6604054  0.0450969 -14.644  < 2e-16 ***
pol_stability  0.0085936  0.0006652  12.919  < 2e-16 ***
research       0.2851603  0.0149724  19.046  < 2e-16 ***
pop_growth     0.1404303  0.0143011   9.820  < 2e-16 ***
dummy_F        0.3218108  0.0360009   8.939  < 2e-16 ***
dummy_PF       0.1571861  0.0322934   4.867 1.17e-06 ***
dummy_60_plus  0.3663217  0.0718106   5.101 3.51e-07 ***
dummy_30_60   -0.1595034  0.0309982  -5.146 2.78e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7425 on 4498 degrees of freedom
Multiple R-squared:  0.7886,	Adjusted R-squared:  0.7882 
F-statistic:  1864 on 9 and 4498 DF,  p-value: < 2.2e-16



	studentized Breusch-Pagan test

data:  model
BP = 283.92, df = 9, p-value < 2.2e-16
```

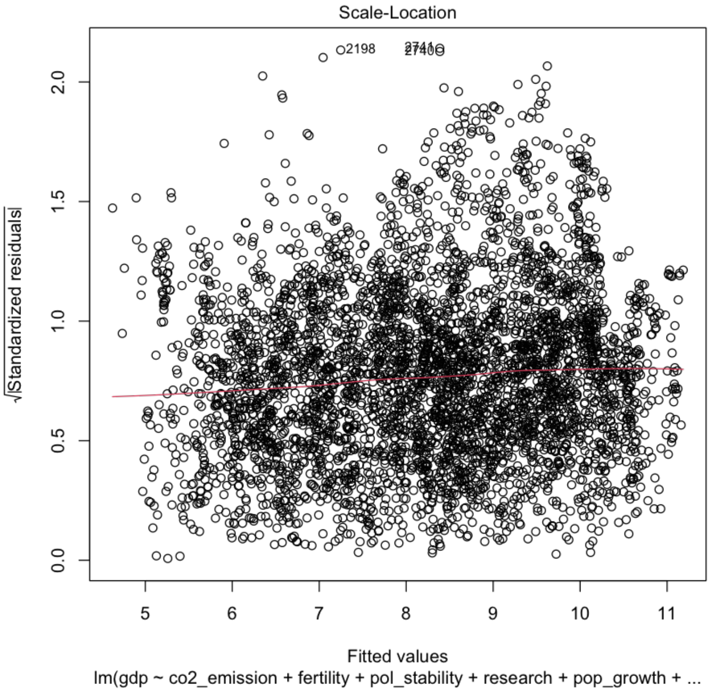
The null hypothesis is still rejected, however all chosen parameters are statistically significant, and the residuals are less spread out. We once again plot the residuals to check what the problem might be

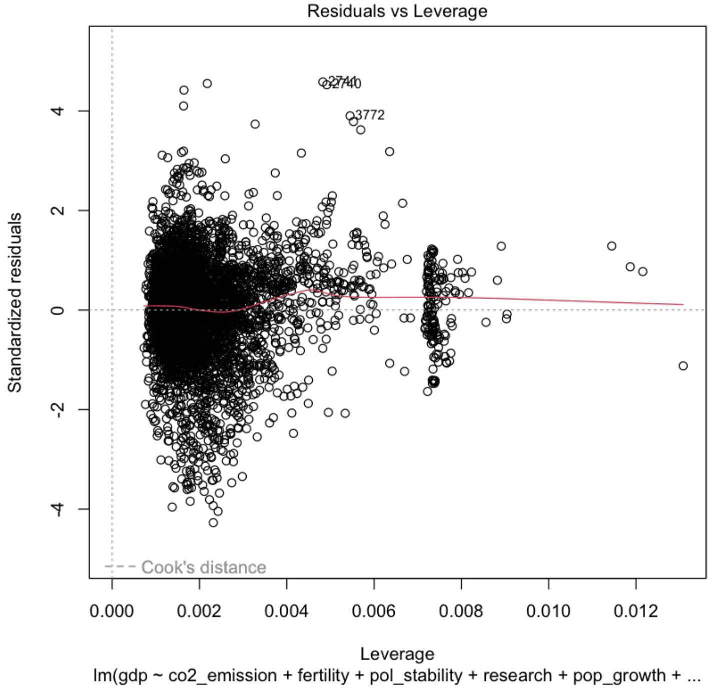
[36]:

```
plot(model)
```


Residuals vs Fitted
Residuals
4
2
0
-2
2198
2741
2740
5
6
7
8
9
10
11
Fitted values
lm(gdp ~ co2_emission + fertility + pol_stability + research + pop_growth + ...


Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
2740
2198
2741
-2
0
2
Theoretical Quantiles
lm(gdp ~ co2_emission + fertility + pol_stability + research + pop_growth + ...



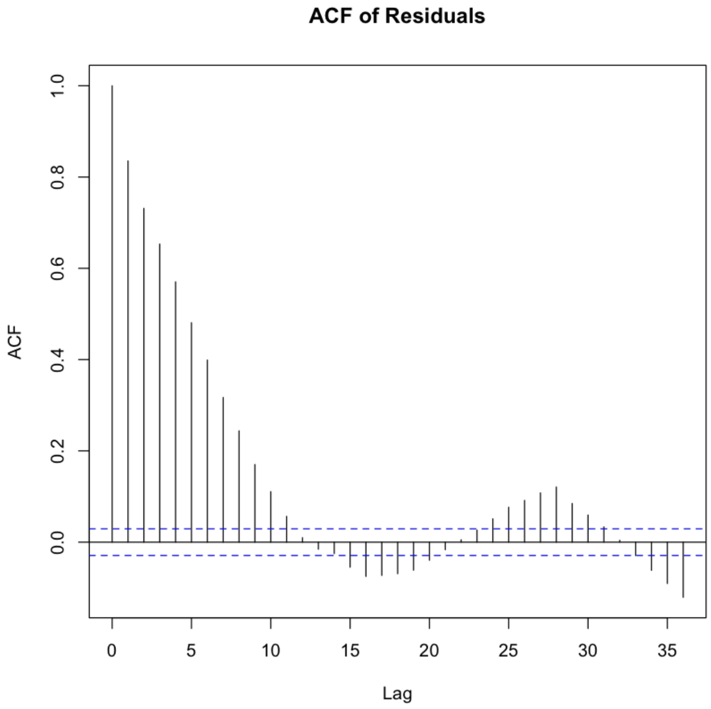


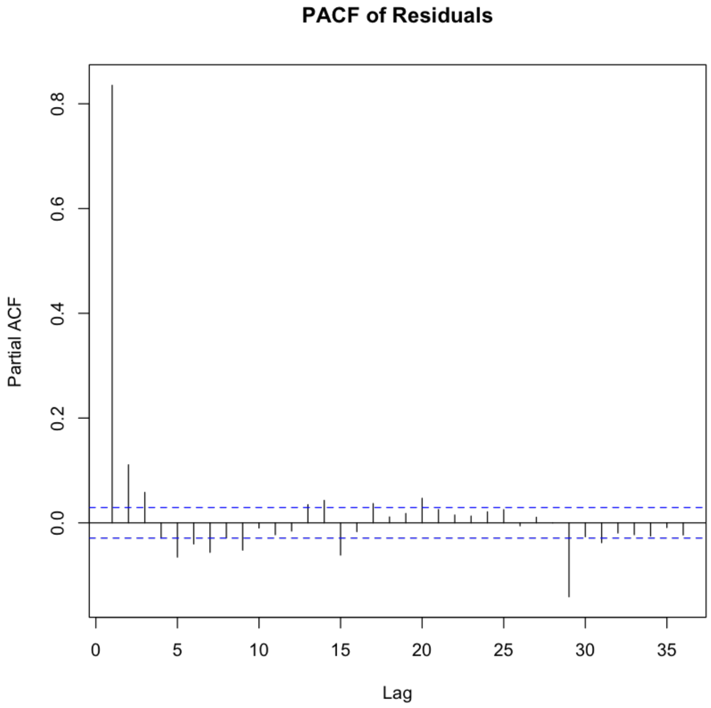
From the `Residuals vs Fitted` we can see that there's a slight correlation and from the `Q-Q plot` we can see that the residuals diverge from a normal distribution.

As we fear that there may be some autocorrelation between the errors, we check the plot of the ACF (autocorrelation function) and PACF (partial autocorrelation function)

[37]:

```
residuals <- residuals(model)
acf(residuals, main="ACF of Residuals")
pacf(residuals, main="PACF of Residuals")
```





As there seems to be significant correlation, we now move to use the Ljung-Box Q-test to check for autocorrelation in the residuals at different lags collectively, not just individually. The Ljung-Box test's primary purpose is to test the null hypothesis that the autocorrelations of the series up to lag k are all zero (i.e., there is no autocorrelation in the series). The test statistic for the Ljung-Box test is defined as:

$$Q = n(n+2)\sum_{k=1}^{h} \frac{\hat{\rho}_k^2}{n-k} \tag{3}$$

where:

- n is the sample size (number of observations in the time series),
- h is the number of lags being tested, - $\hat{\rho}_k$ is the estimated autocorrelation at lag k
- Q is the test statistic.

The test is applied to residuals of a model to check whether the model has left any autocorrelation structure unmodeled. If the null hypothesis is not rejected, it suggests that the model has successfully captured the time-dependent structure of the data.

[38]:

```r
Box.test(residuals, type = "Ljung-Box")
```

```
	Box-Ljung test

data:  residuals
X-squared = 3148, df = 1, p-value < 2.2e-16
```

As the p-value is close to 0, we reject the hypothesis that there's no autocorrelation. For this reason we decided it was best to move away from an OLS model.

1.5 Generalised Least Squared Model

As the model with new predictors is still heteroskedastic, due to autocorrelation, we decided it was best to use a Generalized Least Squares model. Unlike OLS, GLS acknowledges that the variance of the error terms may not be constant (heteroscedasticity) or that errors may be correlated over time or across observations (autocorrelation). To address these issues, GLS transforms the original model using a provided variance-covariance matrix of the errors, leading to accurate and unbiased estimators even in the presence of heteroscedasticity. This approach involves pre-multiplying the model by the inverse square root of the variance-covariance matrix to produce homoscedastic and uncorrelated error terms, allowing the application of OLS to this transformed model. The transformed model's estimators, obtained through OLS, are the GLS estimators for the original model, providing more reliable coefficient estimates under the specific violations of OLS assumptions.

As there is autocorrelation within the model, we decided to use an autoregressive of order 1 correlation structure (`corAR1`). The intuition is that each observation is correlated to its preceding observation, implying that the current value can be partly predicted by its immediate past value. Mathematically, the variance-covariance matrix Σ can be seen as:

$$\Sigma = (X^T W^{-1} X)^{-1} X^T W^{-1} V W^{-1} X (X^T W^{-1} X)^{-1} \tag{4}$$

Where:

- X is the matrix of independent variables, so in our case intercept, corruption, co2 emission, research, fertility and population growth
- W is a diagonal weight matrix used to account for heteroscedasticity, in this case $W = \sigma^2 \cdot I$
- V represents the covariance matrix that models the correlation between the observations. For `corAR1`, this correlation is modeled as decaying exponentially with the distance between observations, which is parameterized by the correlation coefficient ρ. Mathematically:

$$\rho(y_i, y_j) = \rho^{|i-j|} \tag{5}$$

[39]:
```
gls_model <- gls(gdp ~ pol_stability + co2_emission + research + fertility +␣
 ↪pop_growth + dummy_F + dummy_PF + dummy_30_60 + dummy_60_plus,
  data = data_regression,
  corr = corAR1(),
  control = glsControl(
    maxIter = 1000,
    msMaxIter = 100,
    returnObject = TRUE,
    tolerance = 1e-6,
    msTol = 1e-6
    ),
  verbose = TRUE
)
```

We now check the summary of the model

[40]:
```
summary(gls_model)
```

```
Generalized least squares fit by REML
  Model: gdp ~ pol_stability + co2_emission + research + fertility + pop_growth + ␣
 ↪     dummy_F + dummy_PF + dummy_30_60 + dummy_60_plus
  Data: data_regression
       AIC      BIC    logLik
  3570.465 3647.401 -1773.232

Correlation Structure: AR(1)
 Formula: ~1
 Parameter estimate(s):
      Phi
0.9144551

Coefficients:
                     Value  Std.Error   t-value p-value
(Intercept)     9.216946 0.09928995  92.82860  0.0000
```

```
pol_stability  0.004938 0.00058547   8.43428  0.0000
co2_emission   0.224587 0.01279419  17.55381  0.0000
research       0.058513 0.01105590   5.29245  0.0000
fertility     -1.329224 0.05232644 -25.40253  0.0000
pop_growth     0.002456 0.01034193   0.23749  0.8123
dummy_F        0.251617 0.04208140   5.97929  0.0000
dummy_PF       0.023927 0.03201914   0.74727  0.4549
dummy_30_60   -0.205493 0.05258678  -3.90769  0.0001
dummy_60_plus  0.789015 0.12016880   6.56589  0.0000

 Correlation:
              (Intr) pl_stb c2_mss resrch frtlty pp_grw dmmy_F dmm_PF d_30_6
pol_stability -0.275
co2_emission  -0.242 -0.117
research       0.120 -0.074 -0.001
fertility     -0.674  0.070  0.430  0.090
pop_growth     0.089  0.037 -0.049  0.000 -0.222
dummy_F       -0.235 -0.236 -0.072 -0.084  0.127 -0.010
dummy_PF      -0.218 -0.085  0.102  0.130  0.097  0.010  0.576
dummy_30_60   -0.419  0.053 -0.054 -0.133  0.402  0.060 -0.073 -0.075
dummy_60_plus -0.133 -0.044 -0.068 -0.136  0.124 -0.022 -0.094 -0.035  0.278

Standardized residuals:
        Min          Q1         Med          Q3         Max
-3.49264276 -0.60845157 -0.03293112  0.58357503  4.15505714

Residual standard error: 0.8810338
Degrees of freedom: 4508 total; 4498 residual
```

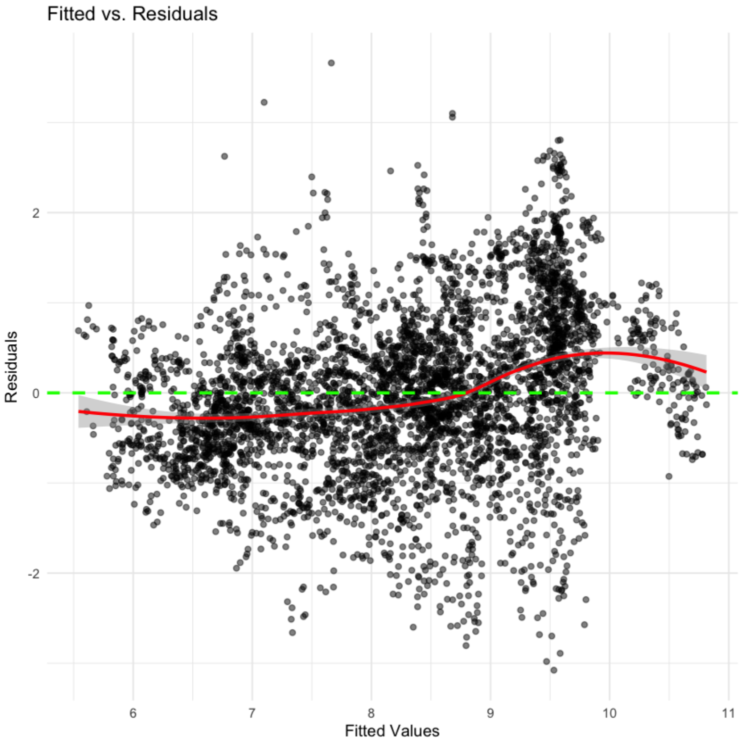
How to interpret the output:

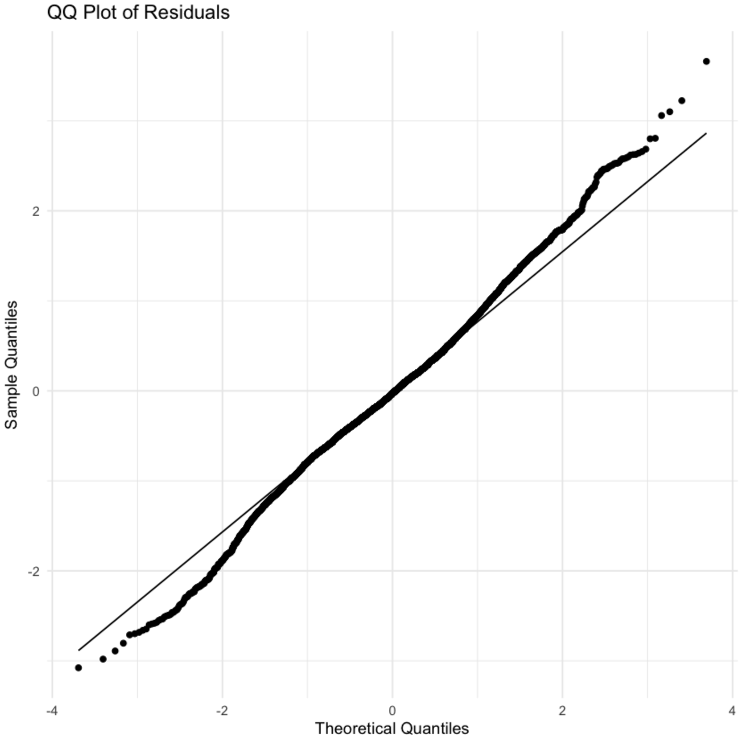
- Generalized least squares fit by REML" - The model was fitted using the Restricted Maximum Likelihood method
- "Model" - The model used
- "AIC" - Akaike Information Criterion, a measure of the model quality that balances model fit and complexity. Lower values are better
- "BIC" - Bayesian Information Criterion, similar to AIC but with a stronger penalty for model complexity. Lower values are better
- "logLik" - The log-likelihood of the model, a measure of how well the model fits the data. Higher values are better
- "Correlation Structure": type of correlation used
- Parameter estimate(s) - range: The estimated range parameter for the correlation structure
- "Coefficients" - Same as simple lm except that the indicators of significance aren't present
- "Correlation" - Shows the correlation between the predictors

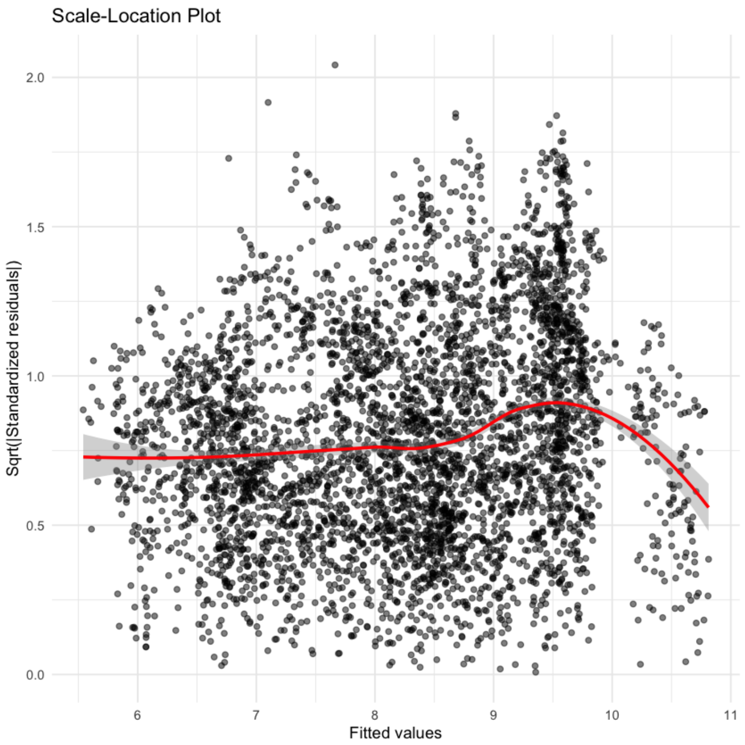
- "Standardized residuals" - The residuals of the model, which are the differences between the observed and predicted values of the dependent variable, standardized by the standard deviation of the residuals
- "Residual standard error" - The standard deviation of the residuals, a measure of the model's accuracy. Lower values are better
- "Degrees of freedom" - The number of observations minus the number of parameters in the model

To check if we have removed the problem of heteroskedasticity, we will plot the residuals of the model, as the Breusch-Pagan Test is not available for GLS models

[41]:
```r
data_for_ggplot <- data.frame(
  Fitted = fitted(gls_model),
  Residuals = residuals(gls_model)
)
ggplot(data_for_ggplot, aes(x = Fitted, y = Residuals)) +
  geom_point(alpha = 0.5) +
  geom_smooth(aes(y = Residuals), method = "loess", formula = 'y ~ x', color =␣
 ↪"red") +
  geom_hline(yintercept = 0, color = "green", linetype = "dashed", size = 1) +
  labs(x = "Fitted Values", y = "Residuals", title = "Fitted vs. Residuals") +
  theme_minimal()
ggplot(mapping = aes(sample = residuals(gls_model))) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantiles", y = "Sample␣
 ↪Quantiles") +
  theme_minimal()
std_resid <- resid(gls_model, type = "pearson") / sd(resid(gls_model, type =␣
 ↪"pearson"))
data_for_plot <- data.frame(
  Fitted = fitted(gls_model),
  SqrtAbsStdResid = sqrt(abs(std_resid))
)
ggplot(data_for_plot, aes(x = Fitted, y = SqrtAbsStdResid)) +
  geom_point(alpha = 0.5) +
  geom_smooth(aes(y = SqrtAbsStdResid), method = "loess", formula = 'y ~ x',␣
 ↪color = "red") +
  labs(x = "Fitted values", y = "Sqrt(|Standardized residuals|)", title =␣
 ↪"Scale-Location Plot") +
  theme_minimal()
```


Fitted vs. Residuals
Residuals
2
0
-2
6
7
8
9
10
11
Fitted Values

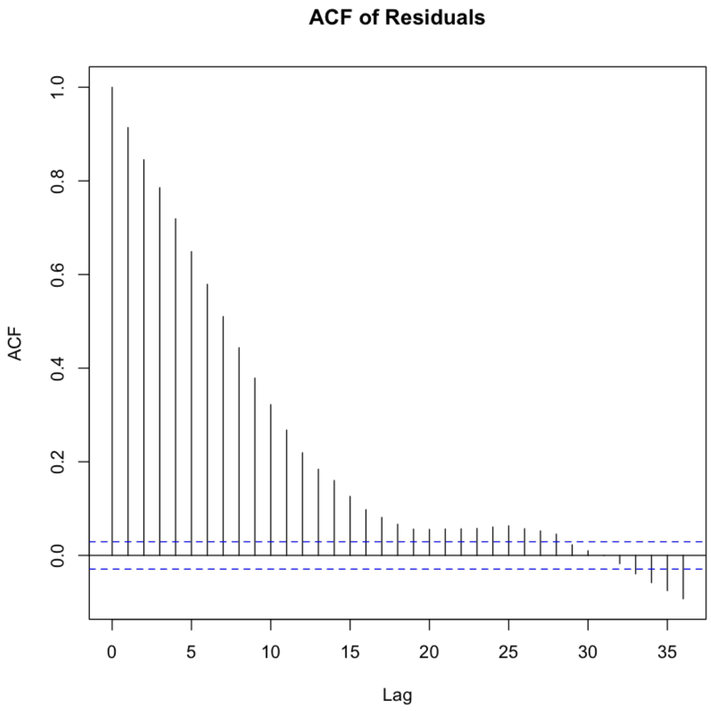

QQ Plot of Residuals
Sample Quantiles
2
0
-2
-4
-2
0
2
4
Theoretical Quantiles

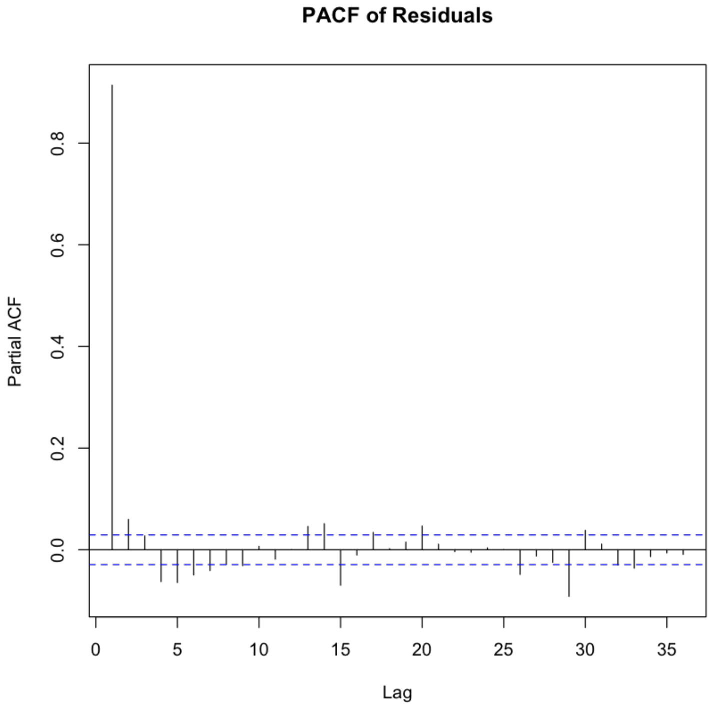


The model is quite better than the OLS one, with quite a lot of improvement on the distribution of the errors, although there's some amount of heteroskedasticity not explained by the current model, with errors following a rather unusual patterns, as demonstrated by the Scale-Location plot and the Fitted against Residual Plot. For this reason, we check whether our model is able to catch all the autocorrelation we observed in the OLS model.

[42]:

```r
residuals <- residuals(gls_model)
acf(residuals, main="ACF of Residuals")
pacf(residuals, main="PACF of Residuals")
```





```
[43]: Box.test(residuals, type = "Ljung-Box")
```

```
        Box-Ljung test
```

```
data:  residuals
X-squared = 3768.5, df = 1, p-value < 2.2e-16
```

From the Ljung-Box Q-test test, it is clear that we need a better model. From the summary, we can see that population growth and the dummy variable for the Partially Free countries are not significant, we will drop them.

Furthermore, as there seems to be a correlation between fertility, CO2 Emissions and the dummy for countries between 30 and 60 degrees of latitude from the equator, we will decorrelate them, using as a predictor the residuals of the following model:

$$fertility = \beta_0 + \beta_1 \cdot Dummy\ for\ distance\ from\ equator + \beta_2 \cdot CO2\ Emission$$

[34]:

```
decorr_model <- lm(fertility ~ dummy_30_60 + co2_emission, data =␣
 ↪data_regression)
data_regression$decorr_fertility <- residuals(decorr_model)
```

We now try to fit another GLS model, this time trying to also model the variance of the error terms changes as a power function of the fitted values. Mathematically:

$$W_{i,i} = |\hat{y}_i|^\theta \tag{6}$$

Where $\hat{y}_i$ is the fitted value of the ith observation and θ is estimated by the model

[45]:

```
gls_model_2 <- gls(gdp ~ pol_stability + co2_emission + research + dummy_F +␣
 ↪dummy_30_60 + dummy_60_plus + decorr_fertility,
  data = data_regression,
  corr = corAR1(),
  weights = varPower(form = ~fitted(.)),
  control = glsControl(
    maxIter = 1000,
    msMaxIter = 100,
    returnObject = TRUE,
    tolerance = 1e-6,
    msTol = 1e-6
  ),
  verbose = TRUE
)
```

We once again check the summary, and plot the residuals to check whether heteroskedasticity is still present

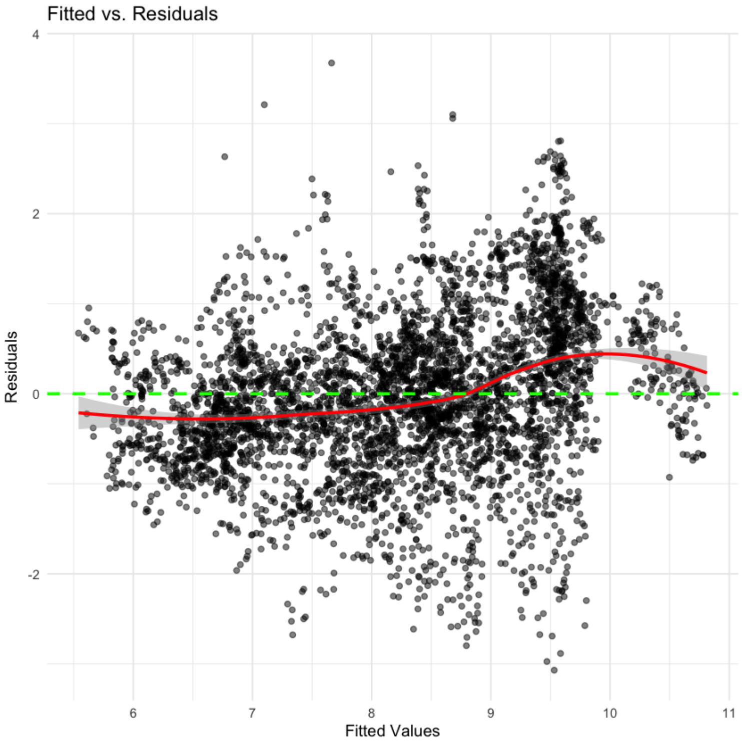
[46]:

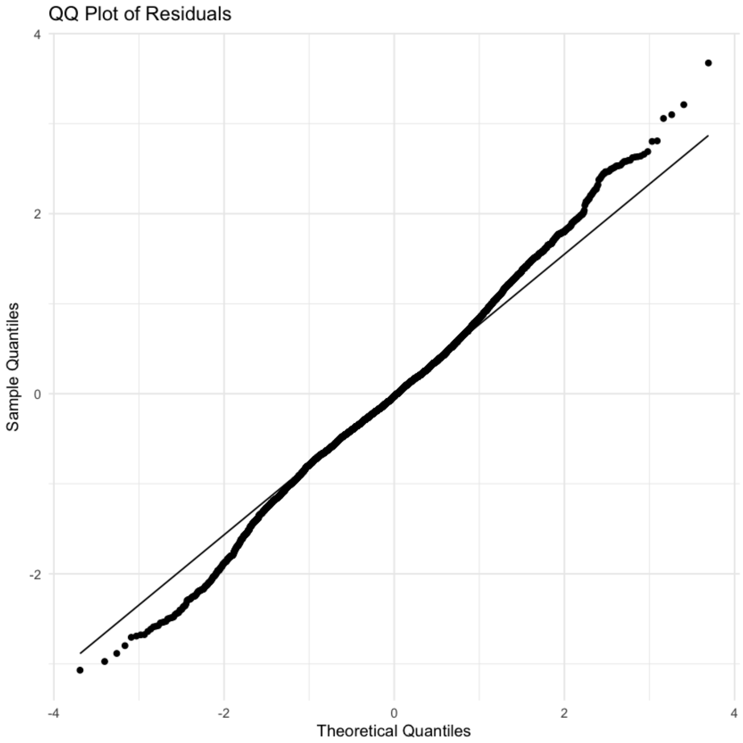
```
summary(gls_model_2)
```

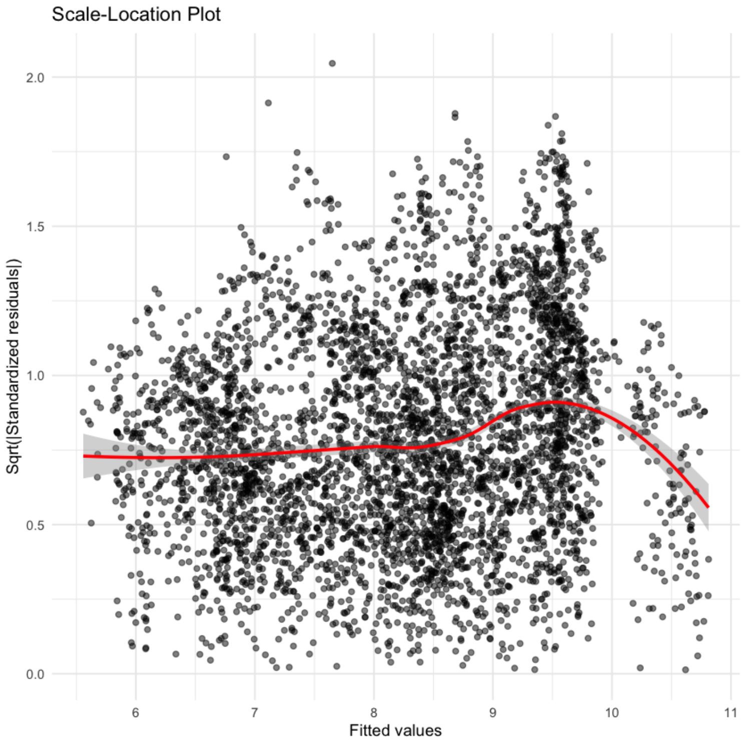
```
Generalized least squares fit by REML
  Model: gdp ~ pol_stability + co2_emission + research + dummy_F + dummy_30_60 +  ␣
↪    dummy_60_plus + decorr_fertility
  Data: data_regression
       AIC      BIC     logLik
  3556.71 3627.24 -1767.355

Correlation Structure: AR(1)
 Formula: ~1
 Parameter estimate(s):
      Phi
0.9144376
Variance function:
 Structure: Power of variance covariate
 Formula: ~fitted(.)
 Parameter estimates:
     power
0.01070032

Coefficients:
                     Value  Std.Error   t-value p-value
(Intercept)       7.637016 0.07155278 106.73262   0e+00
pol_stability     0.004958 0.00058248   8.51209   0e+00
co2_emission      0.488707 0.01242834  39.32202   0e+00
research          0.057313 0.01095593   5.23122   0e+00
dummy_F           0.233378 0.03437766   6.78864   0e+00
dummy_30_60       0.186288 0.04774464   3.90175   1e-04
dummy_60_plus     0.792647 0.12031890   6.58788   0e+00
decorr_fertility -1.331481 0.05073539 -26.24363   0e+00

 Correlation:
                 (Intr) pl_stb c2_mss resrch dmmy_F d_30_6 dm_60_
pol_stability    -0.336
co2_emission      0.110 -0.184
research          0.275 -0.064 -0.080
dummy_F          -0.111 -0.230 -0.235 -0.196
dummy_30_60      -0.248  0.021 -0.321 -0.162 -0.065
dummy_60_plus    -0.084 -0.047 -0.170 -0.133 -0.091  0.266
decorr_fertility -0.057  0.090 -0.378  0.080  0.085  0.169  0.127

Standardized residuals:
        Min          Q1         Med          Q3         Max
-3.48044113 -0.60793526 -0.03313992  0.58655835  4.17418388

Residual standard error: 0.8612087
Degrees of freedom: 4508 total; 4500 residual
```

[47]:

```r
data_for_ggplot <- data.frame(
  Fitted = fitted(gls_model),
  Residuals = residuals(gls_model_2)
)
ggplot(data_for_ggplot, aes(x = Fitted, y = Residuals)) +
  geom_point(alpha = 0.5) +
  geom_smooth(aes(y = Residuals), method = "loess", formula = 'y ~ x', color =␣
 ↪"red") +
  geom_hline(yintercept = 0, color = "green", linetype = "dashed", size = 1) +
  labs(x = "Fitted Values", y = "Residuals", title = "Fitted vs. Residuals") +
  theme_minimal()
ggplot(mapping = aes(sample = residuals(gls_model_2))) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantiles", y = "Sample␣
 ↪Quantiles") +
  theme_minimal()
std_resid <- resid(gls_model_2, type = "pearson") / sd(resid(gls_model_2, type =␣
 ↪"pearson"))
data_for_plot <- data.frame(
  Fitted = fitted(gls_model_2),
  SqrtAbsStdResid = sqrt(abs(std_resid))
)
ggplot(data_for_plot, aes(x = Fitted, y = SqrtAbsStdResid)) +
  geom_point(alpha = 0.5) +
  geom_smooth(aes(y = SqrtAbsStdResid), method = "loess", formula = 'y ~ x',␣
 ↪color = "red") +
  labs(x = "Fitted values", y = "Sqrt(|Standardized residuals|)", title =␣
 ↪"Scale-Location Plot") +
  theme_minimal()
```



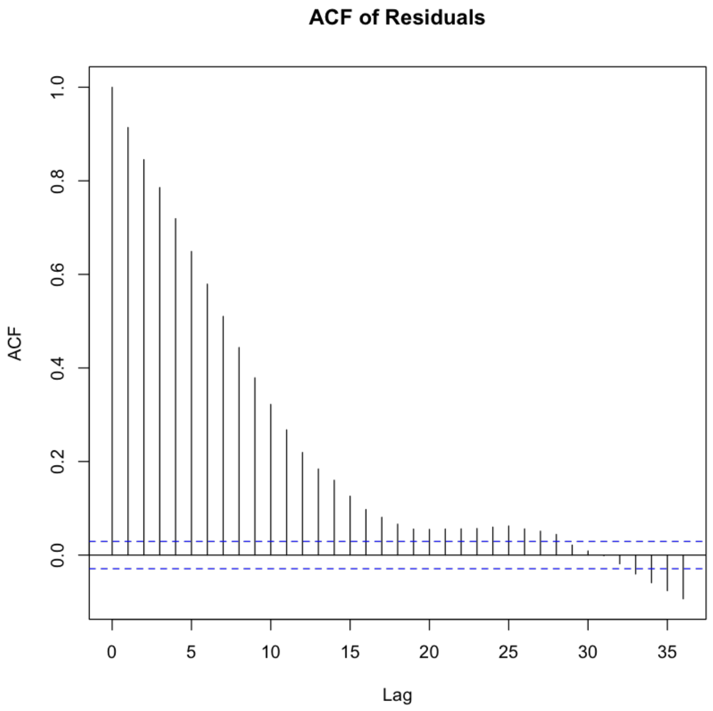


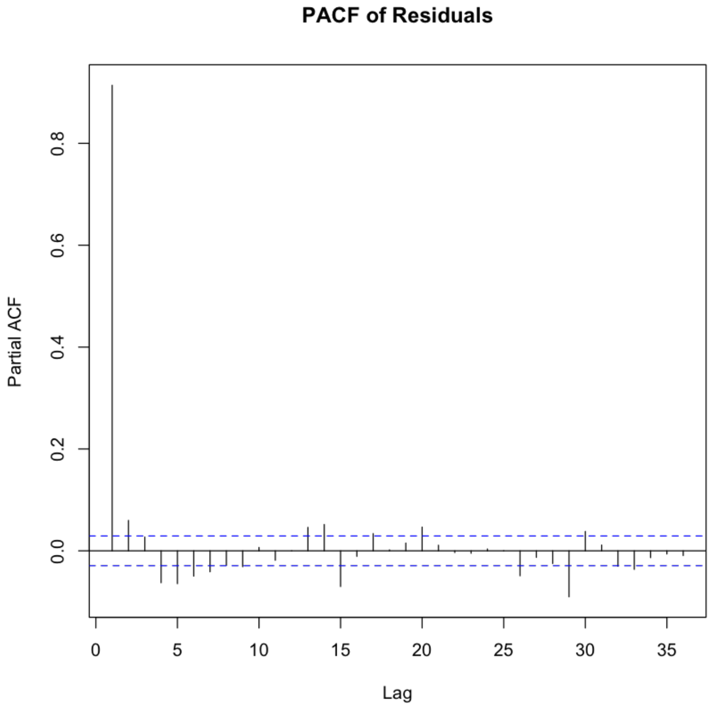


As the model hasn't quite improved, we now move on to test wheter autocorrelation is still present

[48]:

```
residuals <- residuals(gls_model_2)
acf(residuals, main="ACF of Residuals")
pacf(residuals, main="PACF of Residuals")
```





[49]:
```
Box.test(residuals, type = "Ljung-Box")
```

```
	Box-Ljung test
```

```
data:  residuals
X-squared = 3768.9, df = 1, p-value < 2.2e-16
```

As the autocorrelation is still there, we choose to move to another model, built just to handle autocorrelation

1.6 AutoRegressive Integrated Moving Average with eXogenous variables

Given the limitations observed in the GLS approach for addressing all forms of autocorrelation, we try using an AutoRegressive Integrated Moving Average with eXogenous variables (ARIMAX) model. This model extends the ARIMA (AutoRegressive Integrated Moving Average) framework by incorporating external (exogenous) variables into the equation. The ARIMAX model is constructed as follows:

- AR (AutoRegressive): models the current value of the series as a function of its previous values. The autoregressive part indicates that the evolving variable of interest is regressed on its own lagged values.
- I (Integrated): deals with differencing the observational data to achieve stationarity, although in our case, as we will see, the data is already stationary
- MA (Moving Average): captures the dependency between an observation and a residual error from a moving average model applied to lagged observations. It helps in smoothing out the series and addressing short-term correlations.
- X (Exogenous Variables): allows the ARIMAX model to account for the influence of outside factors on the time series of interest. These variables are not modeled within the time series dynamics but are considered additional inputs that impact the series.

Transitioning to an ARIMAX model facilitates a more nuanced understanding of time series data by allowing the incorporation of external factors that influence the series, alongside modeling the intricate autocorrelation patterns not adequately captured by GLS.

But first, we check for stationarity using the Augmented Dick Fuller Test (ADF), which is a statistical test used to determine whether a time series is stationary, specifically whether it has a unit root, which is indicative of non-stationarity. The ADF test is a hypothesis test where the null hypothesis (H_0) says that the time series has a unit root (and is thus non-stationary). The alternative hypothesis (H_1) suggests that the time series does not have a unit root, implying it is stationary or can be made stationary through differencing.

[50]:

```
adf.test(data_regression$gdp)
adf.test(data_regression$pol_stability)
adf.test(data_regression$co2_emission)
adf.test(data_regression$research)
adf.test(data_regression$decorr_fertility)
```

```
	Augmented Dickey-Fuller Test

data:  data_regression$gdp
Dickey-Fuller = -9.8752, Lag order = 16, p-value < 0.01
```

```
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  data_regression$pol_stability
Dickey-Fuller = -10.723, Lag order = 16, p-value < 0.01
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  data_regression$co2_emission
Dickey-Fuller = -10.474, Lag order = 16, p-value < 0.01
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  data_regression$research
Dickey-Fuller = -9.9602, Lag order = 16, p-value < 0.01
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  data_regression$decorr_fertility
Dickey-Fuller = -10.84, Lag order = 16, p-value < 0.01
alternative hypothesis: stationary
```

As the p-value is <0.01, we reject the null hypothesis and know that the series is stationary, or can be made stationary. Now we move on to implementing the ARIMAX model, using `auto.arima`, which is a function of the forecast package that finds the best hyper-parameters p, q, d for a given model. An ARIMA model is described by three parameters, (p, d, q), where:

- p is the order of the autoregressive (AR) part,
- d is the degree of differencing required to make the series stationary,
- q is the order of the moving average (MA) part

The mathematical representation of an ARIMA model is:

$$\Phi(B)\Delta^d y_t = \delta + \theta(B)\epsilon_t \tag{7}$$

where:

- y_t is the time series at time t
- B is the backshift operator, such that $B^k y_t = y_{t-k}$

- $\Delta^d = (1B)^d$ represents the differencing operator to achieve stationarity,
- $\Phi(B) = 1\phi_1 B\phi_2 B^2 - \cdots - \phi_p B^p$ is the AR polynomial of order p,
- $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$ is the MA polynomial of order q
- δ is a constant (intercept term),
- ϵ_t is the error term, which is assumed to be white noise.

`auto.arima` also estimates the best lambda for the Box-Cox transformation, a transformation that was designed to stabilize the variance and make a dataset more closely conform to the assumption of normality. Mathematically, given a dataset y with $y_i \geq 0\ \forall i$ and parameter λ,

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda > 0 \\ log(y) & \text{if } \lambda = 0 \end{cases}$$

[51]:

```
gdp_ts <- ts(data_regression$gdp)
exog_vars <- as.matrix(data_regression[, c("pol_stability", "co2_emission",␣
 ↪"research", "dummy_F", "dummy_30_60", "dummy_60_plus", "decorr_fertility")])
arimax_model <- auto.arima(gdp_ts, xreg = exog_vars, max.p = 7, max.q = 7, max.P␣
 ↪= 7, max.Q = 7, max.order = 21, max.d = 7, max.D = 7, lambda = "auto")
summary(arimax_model)
```

```
Series: gdp_ts
Regression with ARIMA(1,0,1) errors
Box Cox transformation: lambda= -0.8451986

Coefficients:
         ar1      ma1  intercept  pol_stability  co2_emission  research
      0.9149  -0.0774     0.9657          1e-04        0.0123     8e-04
s.e.  0.0068   0.0166     0.0031          0e+00        0.0003     3e-04
      dummy_F  dummy_30_60  dummy_60_plus  decorr_fertility
       0.0048       0.0018         0.0073           -0.0316
s.e.   0.0008       0.0011         0.0027            0.0012

sigma^2 = 6.507e-05:  log likelihood = 15336.13
AIC=-30650.26   AICc=-30650.21   BIC=-30579.72

Training set error measures:
                     ME     RMSE       MAE       MPE     MAPE    MASE
Training set 0.02038114 0.371539 0.1807438 0.0109281 2.273556 1.05521
                    ACF1
Training set -0.03127374
```

How to interpret the results:

- Series: it is the series the model is predicting, in this case gdp per capita
- Regression with ARIMA(p,d,q) errors: indicates the model being fitted

- Box Cox transformation: lambda = n: indicates the lambda used for the Box-Cox transformation
- Coefficients: Enumeration of coefficients and s.e.
- sigma^2: the variance of the errors
- log likelihood: The log of the likelihood function, a measure of model fit. The higher, the better
- AIC (Akaike Information Criterion), AICc (Corrected Akaike Information Criterion), and BIC (Bayesian Information Criterion) are measures used to compare models. Lower values suggest a better model fit, considering the trade-off between goodness of fit and complexity
- Training error measures: evaluate how well the model has performed on the training dataset

Now we check that all the parameters are statistically significant

[52]:

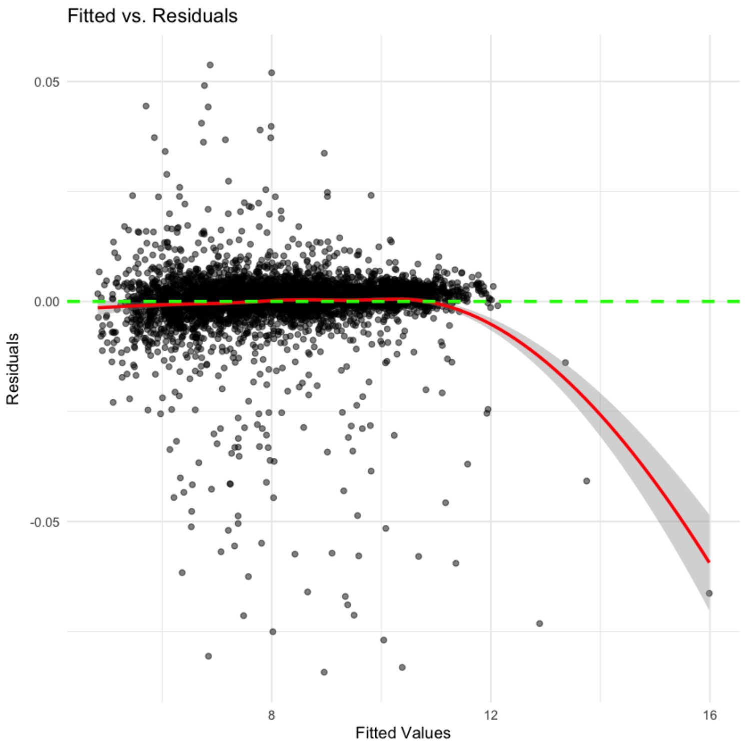
```
coefficients <- coef(arimax_model)
var_coefficients <- summary(arimax_model)$var.coef
std_errors <- sqrt(diag(var_coefficients))
t_stats <- coefficients / std_errors
degrees_of_freedom <- length(residuals(arimax_model)) - length(coefficients) #
 ↪degrees of freedom approximation
p_values <- 2 * pt(-abs(t_stats), df=degrees_of_freedom)
results <- data.frame(Coefficients = coefficients,
                      StdError = std_errors,
                      TStatistic = t_stats,
                      PValue = p_values)
print(results)
```

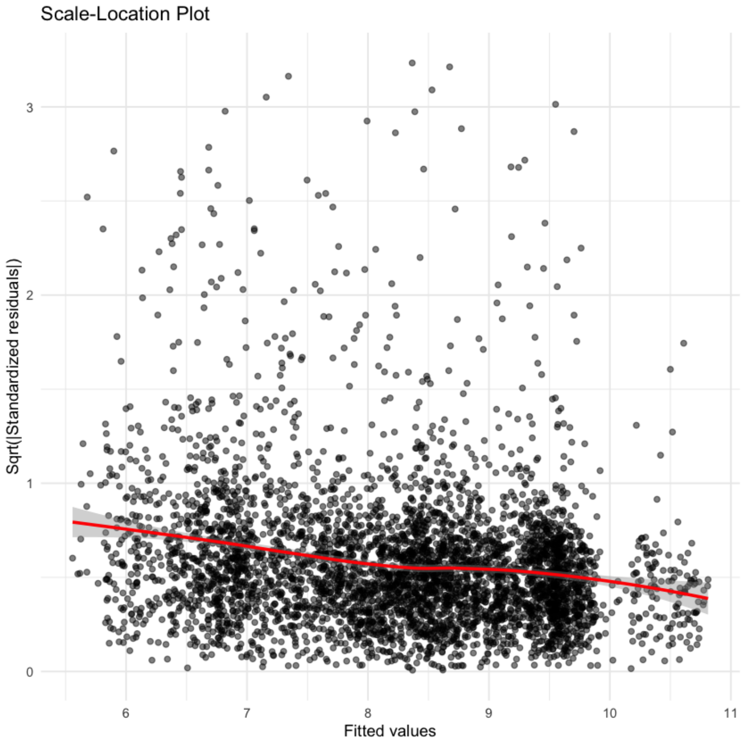
```
                  Coefficients     StdError TStatistic        PValue
ar1               9.148805e-01 6.785058e-03 134.837538  0.000000e+00
ma1              -7.740546e-02 1.660254e-02  -4.662265  3.218045e-06
intercept         9.657357e-01 3.068716e-03 314.703535  0.000000e+00
pol_stability     8.497375e-05 3.037885e-05   2.797135  5.177712e-03
co2_emission      1.228039e-02 2.891518e-04  42.470394  0.000000e+00
research          8.151699e-04 2.541598e-04   3.207313  1.349204e-03
dummy_F           4.820674e-03 7.785880e-04   6.191559  6.488404e-10
dummy_30_60       1.832892e-03 1.075666e-03   1.703960  8.845756e-02
dummy_60_plus     7.312241e-03 2.696310e-03   2.711944  6.714422e-03
decorr_fertility -3.163406e-02 1.152313e-03 -27.452668 1.556455e-153
```

As all the coefficients are significant at the 0.05 level, we will move on to plot the model

[53]:

```
data_for_ggplot <- data.frame(
  Fitted = fitted(arimax_model),
  Residuals = residuals(arimax_model)
)
ggplot(data_for_ggplot, aes(x = Fitted, y = Residuals)) +
  geom_point(alpha = 0.5) +
  geom_smooth(aes(y = Residuals),
              method = "loess",
              formula = "y ~ x",
              color = "red"
  ) +
  geom_hline(yintercept = 0, color = "green", linetype = "dashed", size = 1) +
  labs(x = "Fitted Values", y = "Residuals", title = "Fitted vs. Residuals") +
  theme_minimal()
std_resid <- resid(arimax_model) / sd(resid(arimax_model))
data_for_plot <- data.frame(
  Fitted = fitted(gls_model_2),
  SqrtAbsStdResid = sqrt(abs(std_resid))
)

ggplot(data_for_plot, aes(x = Fitted, y = SqrtAbsStdResid)) +
  geom_point(alpha = 0.5) +
  geom_smooth(aes(y = SqrtAbsStdResid),
              method = "loess",
              formula = "y ~ x",
              color = "red"
  ) +
  labs(x = "Fitted values",
       y = "Sqrt(|Standardized residuals|)",
       title = "Scale-Location Plot"
  ) +
  theme_minimal()
```

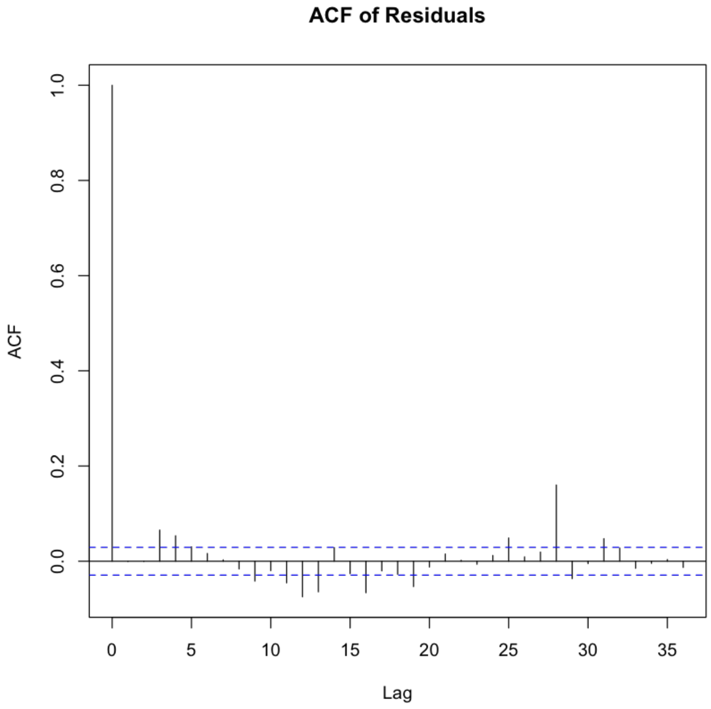


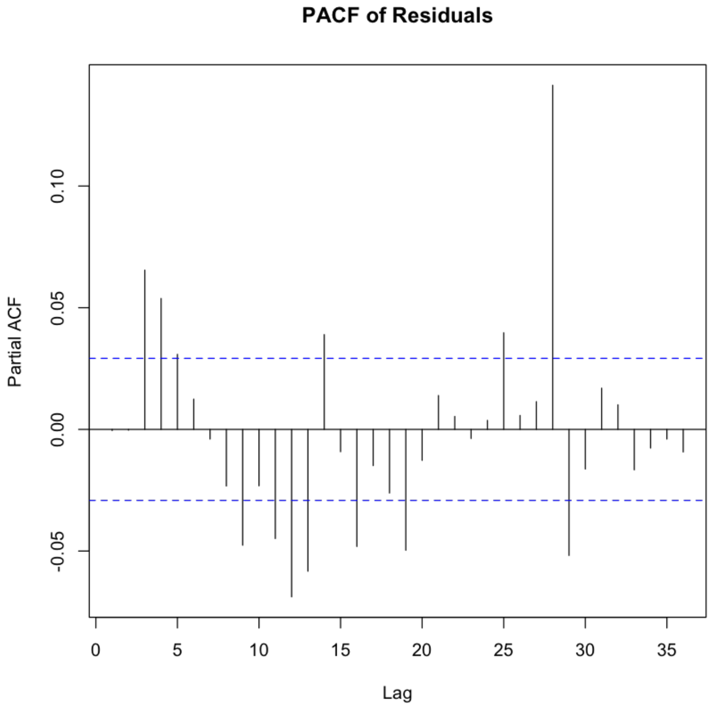


As the model is exceedingly close to having no errors, we plot the ACF and PACF, and run the Ljung-Box Q-test to ensure that no autocorrelation is present

[54]:

```r
residuals <- residuals(arimax_model)
acf(residuals, main = "ACF of Residuals")
pacf(residuals, main = "PACF of Residuals")
```





[55]:
```
Box.test(residuals, type = "Ljung-Box")
```

```
	Box-Ljung test
```

```
data:  residuals
X-squared = 0.0010359, df = 1, p-value = 0.9743
```

The errors of the model are negligible and there isn't any correlation, we can draw some conclusions from the model:

- AR1 coefficient: indicates a strong positive correlation between observations at time t and $t-1$, aligning with expectations. This is consistent with the nature of GDP, where its value at any given moment tends to be significantly influenced by its immediate past;
- MA coefficient: although modest, its negative value indicates that errors from the immediate past tend to slightly decrease the current value. This suggests a corrective mechanism in the model, where past prediction errors are taken into account to adjust the current GDP estimate downwards;
- Intercept: significantly lower compared to all previously estimated intercepts, which we believe more accurately reflects reality, historically, it's evident that an economy consisting solely of a population, without the integration of factors like technology, governance, and infrastructure, tends to be less productive;
- Political stability: small, yet positive influence, aligning with the expectation that a nation with greater stability tends to have a higher GDP per capita on average; the minimal magnitude could stem from the aggregation of upper and lower estimates, however this approach was necessitated by the absence of more refined data, with fewer extreme values
- CO2 Emissions per capita: seems to be slightly positive correlated with gdp, which is to be expected
- Decorrelated Fertility: when accounted for CO2 Emission, fertility is quite negatively correlated, with a decrease in Fertility leading to quite a significant increase in GDP - Research: There is a slight positive correlation between research expenditure and GDP per capita, which is to be expected
- Dummy Free: being classified as a free country shows a slight positive correlation with a higher GDP, although the magnitude of this effect is not as substantial as anticipated. This observation may reflect a narrowing gap between free and less free economies, potentially due to the emergence and success of hybrid authoritarian regimes (such as those in China, Singapore, etc.), which have managed to achieve significant economic growth despite limited political freedoms
- Dummy_30_60 and Dummy_60_plus: indicates a slight economic advantage for countries situated further from the equator, revealing a 'Goldilocks' zone for economic prosperity situated between 30 and 60 degrees latitude, countries within this optimal latitude range tend to demonstrate superior economic performance; this advantage marginally declines for countries located beyond 60 degrees latitude, however this reduction may be influenced by nations in the southern parts of Africa and South America, suggesting geographical location plays a significant role in economic outcomes, with certain latitudes offering more favorable conditions than others