Final_Assignment

Manuel Bottino

2024-10-22

Contents

1. and 2. Introduction

Scientists agree on the fact that climate change is a major issue. We, as data scientists, can help those who do not feel this urge to intervene rapidly and effectively. More than often these are people in power who could make a change. Our tools are simple but powerful. Through my assignment, I try to explore more in deep the major causes of global warming using simple tools.

Data explanation

My idea for tackling the issue is to gather data on max temperature, various indicators of pollution, and environmental disasters for various countries. The reasoning I have followed is that of selecting 17 representative countries (sample). I included both countries mostly affected (now or in the future, Myanmar for instance) by global warnings and its effects, and those who pollute the most (China for instance).

The methodology used to measure the different variables are:

- **Temperature and precipitation** -> CRU TS (Climatic Research Unit gridded Time Series) is the most widely used observational climate dataset. Data is presented on a 0.5° latitude by 0.5° longitude grid over all land domains except Antarctica. It is derived by the interpolation of monthly climate anomalies from extensive networks of weather station observations. The CRU TS version 4.05 gridded dataset is derived from observational data and provides quality-controlled temperature and rainfall values from thousands of weather stations worldwide, as well as derivative products including monthly climatologies and long-term historical climatologies. The dataset is produced by the Climatic Research Unit (CRU) of the University of East Anglia (UEA).
- **Pollution indexes** -> The data series has been compiled using a combination of primary official sources, and third-party data. Traditionally, in bp's Statistical Review of World Energy, the primary energy of non-fossil based electricity (nuclear, hydro, wind, solar, geothermal, biomass in power and other renewables sources) has been calculated on an 'input-equivalent' basis – i.e. based on the equivalent amount of fossil fuel input required to generate that amount of electricity in a standard thermal power plant. For example, if nuclear power output for a country was 100 TWh, and the efficiency of a standard thermal power plant was 38%, the input- equivalent primary energy would be 100/0.38 = 263 TWh or about 0.95 EJ. Oil production data includes crude oil, shale oil, oil sands, condensates (lease condensate or gas condensates that require further refining) and NGLs (natural gas liquids – ethane, LPG and naphtha separated from the production of natural gas). Excludes liquid fuels from other sources such as biofuels and synthetic derivatives of coal and natural gas. This also excludes liquid fuel adjustment factors such as refinery processing gain. Excludes oil shales/kerogen extracted in solid form.
- **Natural disasters** -> EM-DAT publishes comprehensive, global data on each disaster event – estimating the number of deaths; people affected; and economic damages, from UN reports; government records; expert opinion; and additional sources.

Code spectrum

I organized my code spectrum in stages:

1. The easiest part was to find data about CO2 yearly per country which I found on GitHub, a collection of key metrics maintained by Our World in Data and uses data from bp. The only change I made is to select some of the many variables. The hard part was to do the same for temperature and precipitation.

I used data from the World Bank Group, the problem is that downloads are only available per country and type (temperature and precipitation) separately. My job was to put it all together, I managed to do it with under 300 lines of code the big chunk was just repetition of this process as many times as countries chosen, both for precipitation and temperature.

```
df_pr <- read.csv("data/Italia/pr_timeseries_annual_cru_1901-2021_ITA.csv",sep=",",header=FALSE) %>%
  as_tibble()

df_pr <- df_pr[-c(1,2),-c(3:22)]

df_degrees <- read.csv("data/Italia/tas_timeseries_annual_cru_1901-2021_ITA.csv",sep=",",header=FALSE) %
  as_tibble()
df_degrees_pr <-df_degrees[ -c(1,2), -c(3:22)]%>%
  left_join(df_pr, by="V1")%>%
  mutate(Country="Italy")

df <- df %>%
  union_all(df_degrees_pr)
```

2. Then I set up the part on natural disasters:

```
df_nd<-read.csv("data/natural-disasters.csv",sep=",",header=TRUE)
df_nd <- df_nd %>%
  select( Year, Country.name, Total.economic.damages.from.disasters, Number.of.total.people.affected.by
  rename(Country=Country.name)
```

3. Now all that is left to do is to merge the two datasets together. I consciously set up the data to make this part easier. In fact, I just merged them by year and country.

```
df <- left_join(df, df_CO2, by = c("Year" = "Year", "Country" = "Country"))
```

4. I reviewed my data and created three new variables methane_world, nitrous_oxide_world and co2_world (*Notice that with "world" I mean the set of countries selected*), based on the intuition that methane, co2 and nitrous oxide in each country do not increase the temperature of itself, it is the global emission of these greenhouse gases.

5. The dataset is ready, here is the tail.

```
df <- read.csv("data/final_df.csv",sep=",",header=TRUE)
tail(df)
```

```
##      Year Temperature Precipitation     Country population    co2
## 2052 2016       21.40        373.99 Afghanistan   34636212  9.068
## 2053 2017       21.12        298.38 Afghanistan   35643420  9.868
## 2054 2018       21.46        272.78 Afghanistan   36686788 10.818
## 2055 2019       20.61        389.83 Afghanistan   37769496 11.082
## 2056 2020       19.98        398.68 Afghanistan   38972236 11.682
## 2057 2021       21.90        217.90 Afghanistan   40099460 11.874
##      share_global_co2_including_luc total_ghg methane nitrous_oxide
## 2052                          0.024     27.05   15.83          5.05
## 2053                          0.025     26.68   15.61          5.27
```

```
## 2054                          0.027     27.84   15.91          4.69
## 2055                          0.027     28.79   16.37          5.01
## 2056                          0.031        NA      NA            NA
## 2057                          0.030        NA      NA            NA
##      primary_energy_consumption co2_world methane_world nitrous_oxide_world
## 2052                     34.458  230973.2      32405.73            12178.14
## 2053                     36.617  234500.5      32843.70            12421.13
## 2054                     41.989  238741.4      33484.96            12361.10
## 2055                     35.974  239925.7      33980.52            12457.14
## 2056                         NA  228766.5            NA                  NA
## 2057                         NA  240609.6            NA                  NA
##      Total.economic.damages.from.disasters
## 2052                                     0
## 2053                                     0
## 2054                                     0
## 2055                                     0
## 2056                                     0
## 2057                                     0
##      Number.of.total.people.affected.by.disasters Death.rates.from.disasters
## 2052                                            0                 0.2425208
## 2053                                        11240                 0.6340581
## 2054                                     13504000                 0.3080128
## 2055                                       130942                 0.5586519
## 2056                                        51817                 0.8416248
## 2057                                     11035033                 0.9875445
##      above_below_average dummy_above_below_average
## 2052       above average                         1
## 2053       below average                         0
## 2054       below average                         0
## 2055       above average                         1
## 2056       above average                         1
## 2057       below average                         0
```

```r
# Create dataframe for 2021 used for the first graph
df_2021<- df%>%
  filter(Year == "2021")
```

And the explanation of the variables:

- **co2** -> Annual total production-based emissions of carbon dioxide (CO2), excluding land-use change, measured in million tonnes. This is based on territorial emissions, which do not account for emissions embedded in traded goods.
- **share_global_co2_including_luc** -> Annual total production-based emissions of carbon dioxide (CO2), including land-use change, measured as a percentage of global total production-based emissions of CO2 in the same year. This is based on territorial emissions, which do not account for emissions embedded in traded goods. Each country's share of global CO2 emissions has been calculated by Our World in Data using global CO2 emissions provided in the Global Carbon Budget dataset. Global emissions include all country emissions as well as emissions from international aviation and shipping.
- **total_ghg** -> Total greenhouse gas emissions including land-use change and forestry, measured in million tonnes of carbon dioxide-equivalents.
- **methane** -> Total methane emissions including land-use change and forestry, measured in million tonnes of carbon dioxide-equivalents.
- **nitrous_oxide** -> Total nitrous oxide emissions including land-use change and forestry, measured in million tonnes of carbon dioxide-equivalents.

- **primary_energy_consumption** -> Primary energy consumption, measured in terawatt-hours per year.
- The **total number of people affected** is the sum of injured, requiring assistance and homeless
- **'All disasters'** includes all geophysical, meteorological and climate events including earthquakes, volcanic activity, landslides, drought, wildfires, storms, and flooding.

3. Some graphic Illustrations

A general picture can be given by a bar graph that shows the global impact of some countries as a percentage of total emissions per year. In fact, as it is known, developed countries and highly populated ones tend to pollute more. On the other hand, the poorest are the ones who suffer more from global warming, yet they need more resources (electricity, infrastructures, etc. . .) to improve their condition.

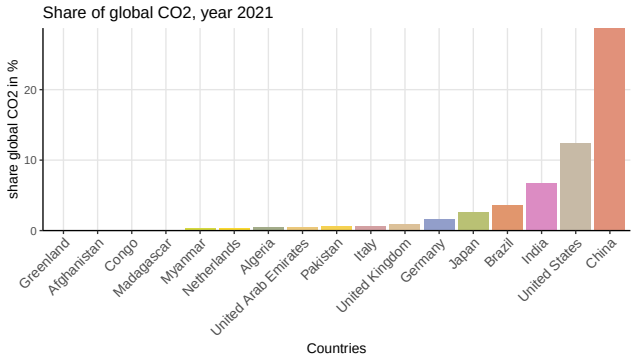


Figure 1: Share of global CO2

In this last graph, I wanted to give another interesting view of the issue. There seems to be a positive correlation between pollution and temperature, even if subtle.

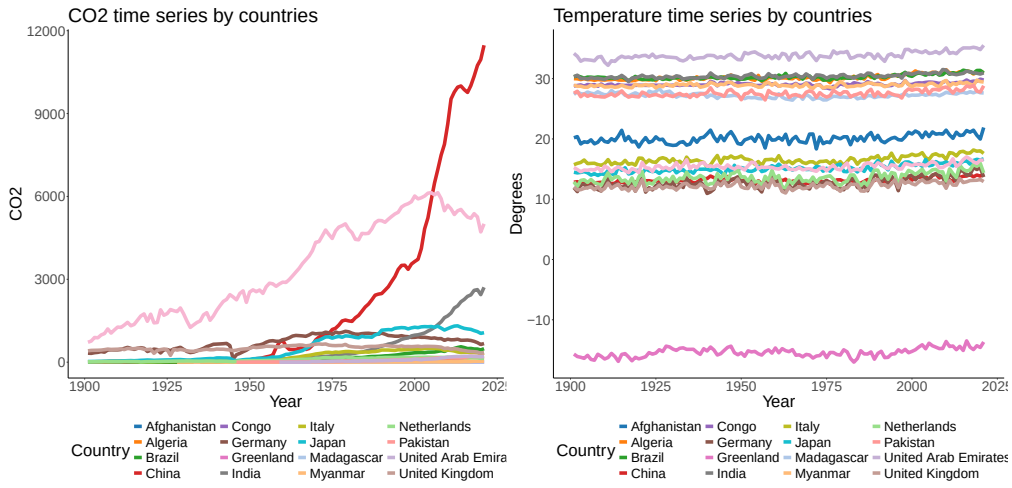


Figure 2: Pollution (Co2) and temperature side by side from 1900 to 2020

Possible Improvements

One idea is to include more countries, it might be the case because my sample has been selected through personal decision (not random), thus it might not be independent and identically distributed. Even though

I tried to exclude personal opinions and include meaningful countries.

4. Multiple regression model

First of all, I chose Italy as the country of interest for the rest of the analysis, the reasoning behind is to draw conclusions on a single country. Still, the possibility of replicate the same reasoning to any other is easily achievable.

In my opinion, it seems logical to use in my multiple regression model temperature (in Celsius) as the response variable and global co2, methane, nitrous oxide and three variables on natural disasters as the explanatory variables. The scope is to quantify the impact of these greenhouse gases on degrees. In other words, how much should we worry if we keep on polluting following the same trend as we did in the last decades?

```
df = df%>%
  filter(Country=="Italy")%>%
  arrange(Year)%>%
  subset(select = -c(Country, above_below_average, dummy_above_below_average, Year,
                     share_global_co2_including_luc, co2, nitrous_oxide, total_ghg, methane))


ols_ML <- lm(Temperature ~ ., data =df)

ols_subset<-regsubsets(Temperature  ~ ., data=df)
summ<-summary(ols_subset)
```

4. and 5. Subset best selection

I have used the best subset selection, instead of the backward or forward methods, which fits all possible linear models (2^p models to fit) and lets us analyze which are the best, on the basis of different criteria, through graphs.

```
# Coefficients of best
format(coef(ols_subset,4), scientific = TRUE, digits = 3)
```

```
##                          (Intercept)                        Precipitation
##                          " 2.51e+01"                          "-1.76e-03"
##                           population                        methane_world
##                          "-2.57e-07"                          " 2.87e-04"
## Total.economic.damages.from.disasters
##                          " 3.22e-11"
```

Cross-validation

The cross-validation method is a method that split the dataset into k folds and computes the Mean Square Error, another information criterion. I had many problems running it, and the only way to fix it was to get rid of all the NAs in the models, which I managed to achieve through a combination of the functions subset and is.finite.

```
df_cross= subset(df, is.finite(as.numeric(Total.economic.damages.from.disasters)))
df_cross= subset(df_cross, is.finite(as.numeric(methane_world)))
p <- 4
k <- 3
folds <- sample (1:k,nrow(df_cross),replace =TRUE)
cv.errors <- matrix (NA ,k, p, dimnames =list(NULL , paste (1:p) ))

for(j in 1:k){
  best.fit =regsubsets (Temperature  ~ ., data=df_cross[folds!=j,])
  mat <- model.matrix(as.formula(best.fit$call[[2]]), df_cross[folds==j,])
  for(i in 1:p) {
    coefi <- coef(best.fit ,id = i)
    xvars <- names(coefi )
    pred <- mat[,xvars ]%*% coefi
    cv.errors[j,i] <- as.numeric(mean((df_cross$Temperature[folds==j] - pred)^2, na.rm=TRUE))
  }
}

cv.mean <- colMeans(cv.errors)
cv.mean
```

```
##         1         2         3         4
## 0.2232631 0.1908163 0.2675283 0.2444006
```

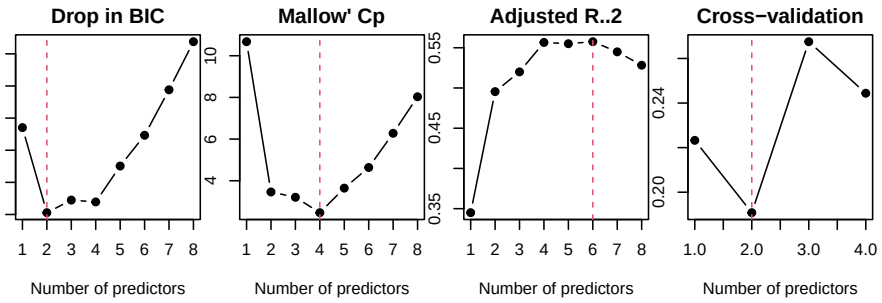


Figure 3: Information criteria

Since I used four criteria for computing the best, I ought to choose which one to depend on for fitting my linear model. It is known that R^2 has many faults, and I would like to have more than two variables (it sounds more realistic for the temperature to be explained by a multitude of variables). These are the reason why I choose Cp as the information criterion. This leads to the four variable selected: Precipitation, population, methane_world, Total.economic.damages.from.disasters.

```
ols_ML <- lm(Temperature ~ Precipitation+population+methane_world+
               Total.economic.damages.from.disasters, data =df)
```

6. Collinearity

The two main ways to look at whether collinearity is present or not are through correlation matrix and VIF method. The latter is generally considered better because it considers the unadjusted coefficient of determination for regressing the ith independent variable on the remaining ones. Let's consider the case above with most regressors:

```
vif(ols_ML)
```

```
##                          Precipitation                             population
##                               1.117835                               11.715349
##                          methane_world Total.economic.damages.from.disasters
##                              11.775756                                1.107293
```

Solutions to collinearity:

1. removing variables, this allows the reduction of linear correlation between two or more variables, trivially because one of the two is not there anymore. Knowing the fact that this would change the model altogether and that the values are slightly above the threshold (10), I still decided to remove the variable population because of the consequences that would come with collinearity.

```
ols_ML <- lm(Temperature ~ Precipitation+methane_world+
               Total.economic.damages.from.disasters, data =df)
vif(ols_ML)
```

```
##                          Precipitation                         methane_world
##                               1.082344                               1.013056
## Total.economic.damages.from.disasters
##                               1.069353
```

2. The second solution is to combine the collinear variables into a single predictor. This solution does not seem adequate in my case.

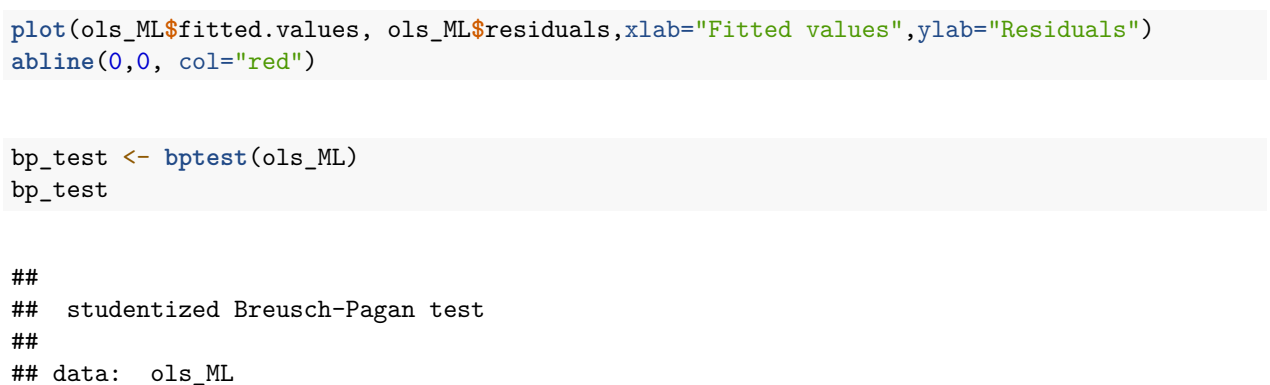
7. Diagnostics

In order to understand better the result it should be pointed out the fact that row 1 corresponds to the year 1900 (first year with methane_world different from NA) and 120 to the year 2019

a) Constant variance assumption

The assumption of constant variance across observations might not be respected, then I would have to intervene with a transformation of the response or use the Weighted Least Squares.

```
plot(ols_ML$fitted.values, ols_ML$residuals,xlab="Fitted values",ylab="Residuals")
abline(0,0, col="red")
```



```
bp_test <- bptest(ols_ML)
bp_test
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ols_ML
## BP = 5.0713, df = 3, p-value = 0.1666
```

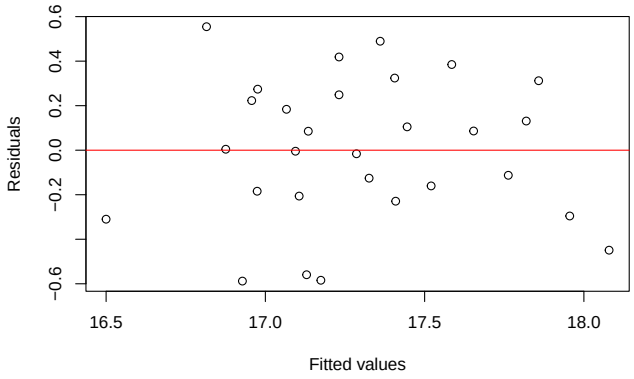


Figure 4: null plot

there is not any obvious sign of non-constant variance or non-linearity, which is also confirmed by the Breusch-Pagan test that I have found by browsing online since just looking did not convince me. This test formally assesses the homoscedasticity assumption by regressing the squared residuals on the independent variables. The null hypothesis is that the variance is constant (homoscedastic).

b) Linearity

A look at residual plots for each variable, the goal is to check linearity issues or subgroups.

```
residualPlots(ols_ML, tests=FALSE)
```

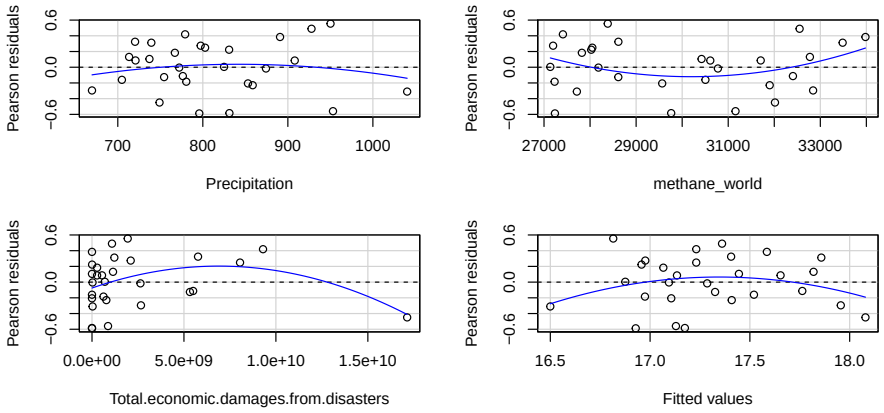


Figure 5: Residuals against coefficient and fitted values

My variables seem to follow the linearity assumption pretty well, except for Total.economic.damages.from.disasters that will be dealt with afterward by taking its squared root.

c) Normality assumption

To address the validity of the normality assumption, we use two tools: qq-plot and shapiro test. The former is a plot that graphs on the x-axis a sample of size n from the normal distribution and on the y-axis another sample of size n from the distribution of the errors. The latter, instead, is a test for the normality distribution of the residuals, which is supposed as the null hypothesis.

```
qqnorm(residuals(ols_ML))
qqline(residuals(ols_ML))
```

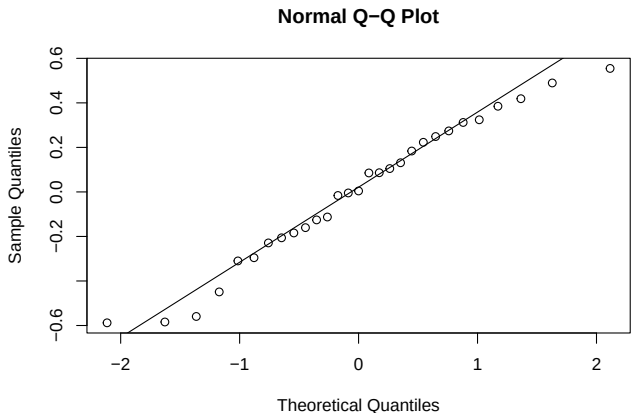


Figure 6: qq-norm

```
shapiro.test(residuals(ols_ML))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ols_ML)
## W = 0.96892, p-value = 0.5307
```

> My plot shows a short-tail situation, which is nothing to worry about. Also, the shapiro test strongly confirms what is shown in the qq-plot.

d) Unusual observations

I) Large leverage points

Leverage points are values far from the variable main domain, where most data is centered. First, we have to compute h_{ii}, then, as a rule of thumb, if its value is above $\frac{2(p+1)}{n}$ it is considered a high leverage point. Differently with outliers, high leverage points influence the estimation of the coefficients.

```
infl <- influence(ols_ML)
hat <- infl$hat
head(hat)
```

```
##         90         91         92         93         94         96
## 0.09358446 0.10854009 0.09829032 0.10850984 0.20536485 0.30016326
```

```
#We verify that the sum of the leverages is indeed 4 - the number of parameters in the model.
sum(hat)
```

```
## [1] 4
```

```
nrow<-sum(!is.na(df$methane_world))# because if we were to count all rows, it would count NA
                                    # which are not included in the fitted model

leverage_points<-hat[which(hat>=(2*4/nrow))] # 4=3(regressors) +1
leverage_points
```

```
##        96       112
## 0.3001633 0.6029722
```

Both the plot and the test show two leverage points, which I will deal with later. In general, it is best to delete them.

II) Outliers

An outlier is a value that does not fit the actual model, there are different ways to treat them. For instance to exclude them or use robust estimators to outliers.

The method of standardized residual is used to identify them

```
rsta <- rstandard(ols_ML)
range(rsta)
```

```
## [1] -2.090997  1.753581
```

```
outlier<-rsta[which.max(rsta)]
outlier
```

```
##      102
## 1.753581
```

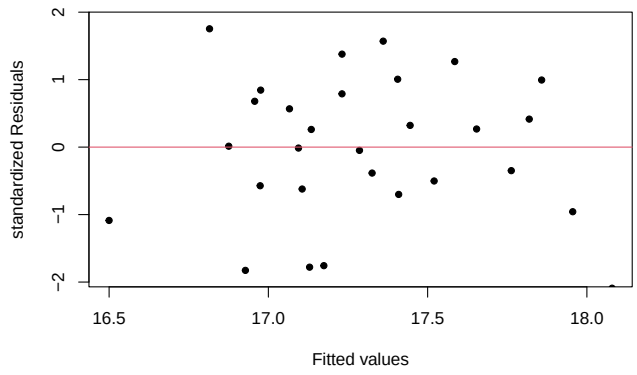


Figure 7: Outliers

As a rule of thumb we consider a data point an outlier if the modulus of its standardized residual is greater than 3. Every point in my dataset has a value represented below this threshold.

III) Influential points

Influential points are a set of all the points that influence the model particularly, of course, both outliers and leverage points could also be influential points.

```
#compute the cook distance
cook <- cooks.distance(ols_ML)

influential<-cook[which.max(cook)]
influential
```

```
##      112
## 1.660058
```

```
#exclude influential observations
ols_cook <- lm(Temperature ~ Precipitation+methane_world+Total.economic.damages.from.disasters, data =d

plot(ols_cook, which=4)
```

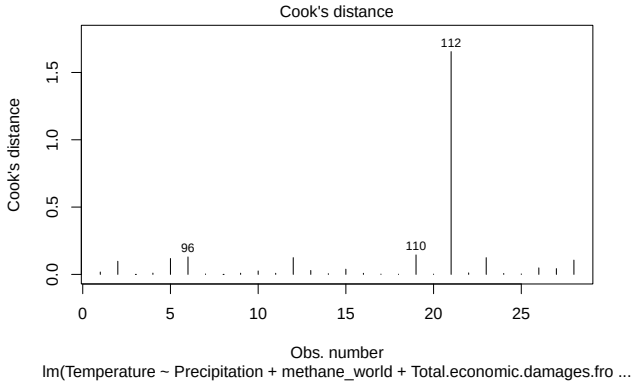


Figure 8: Cook's distance

112 is both an infuential point and a leverage point, it corresponds to the year 2011

8. Improve your model

1. Remove the year 2011 (both an influential point and a leverage point)

```
df<-df %>%  filter(!row_number() %in% c(112))
```

2. Trasformations of Total.economic.damages.from.disasters

```
ols_ML <- lm(Temperature ~ Precipitation + methane_world +sqrt(Total.economic.damages.from.disasters) ,
summary(ols_ML)
```

```
##
## Call:
```

```
## lm(formula = Temperature ~ Precipitation + methane_world + sqrt(Total.economic.damages.from.disasters
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61787 -0.24040  0.06709  0.23353  0.46822
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                               1.462e+01  1.074e+00  13.618
## Precipitation                            -1.865e-03  7.080e-04  -2.634
## methane_world                             1.337e-04  2.769e-05   4.830
## sqrt(Total.economic.damages.from.disasters)  6.082e-06  2.125e-06   2.862
##                                           Pr(>|t|)
## (Intercept)                               8.74e-13 ***
## Precipitation                              0.01454 *
## methane_world                             6.40e-05 ***
## sqrt(Total.economic.damages.from.disasters)  0.00859 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3138 on 24 degrees of freedom
##   (92 observations deleted due to missingness)
## Multiple R-squared:  0.6454, Adjusted R-squared:  0.6011
## F-statistic: 14.56 on 3 and 24 DF,  p-value: 1.305e-05
```

It is interesting, Total.economic.damages.from.disasters was not significant when the variable population was removed, after removing the only influential point and taking the square root, it returns significant. The rest is similar to what we had before.

3. Other options

Also, I could have removed more points (outliers or leverage) but it does not seem wise to do so because the degree of freedom is already pretty low, reducing them more would probably worsen the model.

9. Report the coefficients and use graphics

```
format(coef(ols_ML), scientific = TRUE, digits = 3)
```

```
##                                   (Intercept)
##                                    " 1.46e+01"
##                                 Precipitation
##                                    "-1.86e-03"
##                                 methane_world
##                                    " 1.34e-04"
## sqrt(Total.economic.damages.from.disasters)
##                                    " 6.08e-06"
```

Each coefficient reflects the effect that a unit variation has on the response variable, keeping the other ones constant.

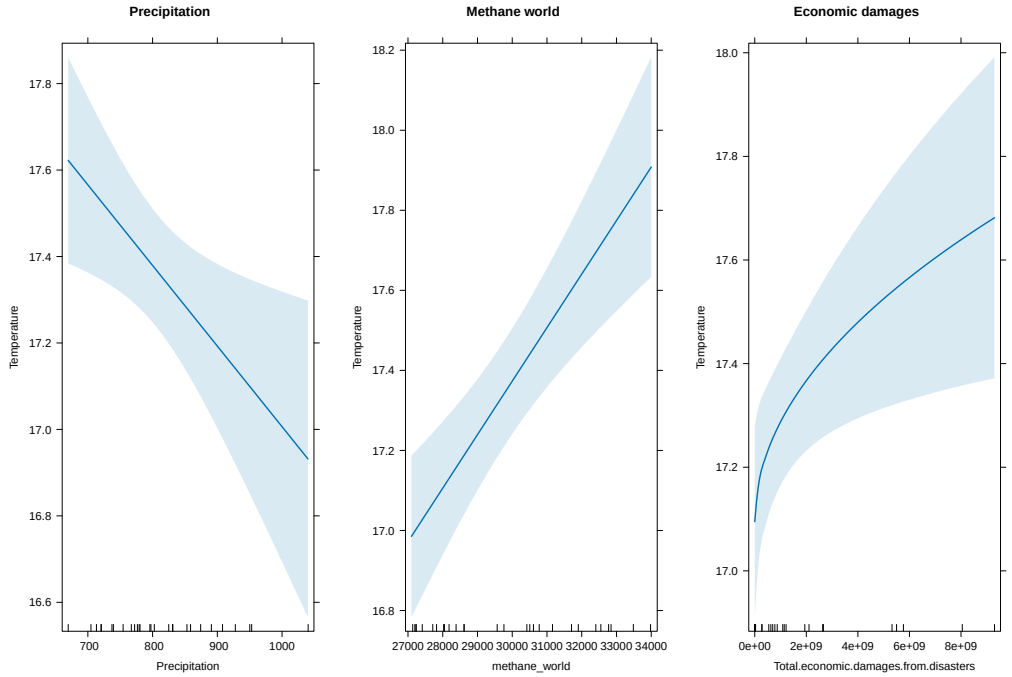


Figure 9: Effect of the coefficients

In my situation, the estimated values are coherent with what happens in the real world. More precipitations make the temperature drop and more methane in the atmosphere makes it rise. It is also noticeable that the coefficient associated with Total.economic.damages.from.disasters has a large shaded area, this reflects a high uncertainty on its real value.

10.,11.,12. Test each regressor, test a group of regressors and all. Discuss the goodness of fit

It is possible to analyse everything with the summary and linearHypothesis, a function from the car library that computes the F-test, among other tests.

```
summary(ols_ML)
```

```
##
## Call:
## lm(formula = Temperature ~ Precipitation + methane_world + sqrt(Total.economic.damages.from.disasters
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61787 -0.24040  0.06709  0.23353  0.46822
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             1.462e+01  1.074e+00  13.618
## Precipitation                          -1.865e-03  7.080e-04  -2.634
## methane_world                           1.337e-04  2.769e-05   4.830
```

```
## sqrt(Total.economic.damages.from.disasters)  6.082e-06  2.125e-06   2.862
##                                                Pr(>|t|)
## (Intercept)                                    8.74e-13 ***
## Precipitation                                   0.01454 *
## methane_world                                  6.40e-05 ***
## sqrt(Total.economic.damages.from.disasters)     0.00859 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3138 on 24 degrees of freedom
##   (92 observations deleted due to missingness)
## Multiple R-squared:  0.6454, Adjusted R-squared:  0.6011
## F-statistic: 14.56 on 3 and 24 DF,  p-value: 1.305e-05
```

All variables are significant but in very different ways. If for the intercept and methane_world there is substantial evidence against the null hypothesis. For the other regressors, Precipitation and Total.economic.damages.from.disasters, this is not the case, they refer to a significant level of 5%, which for many statisticians is not enough.

Concerning goodness of fit, I refer to the adjusted R^2 because it is adjusted for multiple variable models, where the typical R^2 would increase by construction. The model explains 60% of the model variance, which is okay.

I chose to test the hypothesis that both precipitation and methane_world are equal to 0 to study whether these variables are not so important, but it is precipitations that influence temperature the most.

```
linearHypothesis(ols_ML, c("Precipitation = 0", "methane_world = 0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## Precipitation = 0
## methane_world = 0
##
## Model 1: restricted model
## Model 2: Temperature ~ Precipitation + methane_world + sqrt(Total.economic.damages.from.disasters)
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     26 5.6517
## 2     24 2.3632  2    3.2885 16.698 2.857e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test is highly significant, this implies that there is enough statistical evidence to assert that the two variables are different from 0

13. New observation

From an article about an analysis published by US National Oceanic and Atmospheric Administration (NOAA) I have gathered that during the year 2021, an average of 1,895.7 ppb were emitted globally, 17 ppb more than during the year 2020. Thus, there has been an increase of 0.9049%. Under the assumption that

methane increases yearly at the same rate, and the fact that I have the variable of methane in carbon-dioxide equivalent. I will just multiply the last value at my disposal times 1.009049 as many times as it is needed to predict a future value (2030 in my case, given 2017 as starting year). The other two variables, precipitations and total economic damages from disasters are assumed to stay the same from 2017 to 2030, the former because of its complexity. Global precipitations are expected to be more frequent, given the increase in temperature, but it varies a lot among countries. In fact, some countries are getting dryer.

```
new_methane <- tail(df$methane_world, 3)[1]*(1.009049)^12
new_total.economic.damages=tail(df$Total.economic.damages.from.disasters, 3)[1]
new_precipitation <- tail(df$Precipitation, 3)[1]

newdata <- data.frame(Precipitation=new_precipitation, methane_world=new_methane,
                      Total.economic.damages.from.disasters=new_total.economic.damages)
prediction <- predict(ols_ML, newdata)
prediction
```

```
##        1
## 18.02052
```

```
# %increase of temperature in degrees from 2017 to 2030
paste(round((prediction-tail(df$Temperature, 3)[1])/tail(df$Temperature, 3)[1]*100,2), "%")
```

```
## [1] "0.28 %"
```

From this I conclude that the temperature, according to the model fitted and adjusted as above, will increase of about 0.3%.

14. Simulation

There exists a nice function called simulate, by giving it the model and how many simulations. It outputs simulated data, for each year in my instance, using random covariates.

```
sim_data <- simulate(ols_ML, nsim = 6)
sim_data
```

```
##        sim_1    sim_2    sim_3    sim_4    sim_5    sim_6
## 90  16.82123 17.22346 17.33760 17.27706 17.47079 15.95231
## 91  17.00573 17.29996 17.32683 16.50222 16.15464 16.64815
## 92  16.84577 16.33803 16.60042 17.13437 17.44633 17.44915
## 93  17.01056 17.12912 16.96668 17.07764 17.22449 16.89677
## 94  17.50841 17.37741 17.16352 16.89720 17.36189 16.99617
## 96  16.26166 15.99727 16.65576 16.53369 15.95033 16.46154
## 97  17.96011 17.43457 17.59165 16.67452 17.61547 17.68492
## 98  17.32081 17.44772 16.77692 16.83922 16.60527 16.93317
## 99  17.17773 17.24719 17.11258 16.98067 16.48217 17.08875
## 100 17.17769 17.76216 17.60852 17.00059 17.11302 17.38632
## 101 17.45668 17.20468 16.90254 16.65297 16.72999 17.26750
## 102 17.13637 17.43661 16.82968 16.96594 16.63970 16.92323
## 103 17.66197 17.51086 17.06392 17.88326 17.50088 17.38536
## 104 17.31709 16.61219 16.73176 16.37058 16.95013 17.02962
```

```
## 105 16.95888 17.12408 16.77739 17.31565 17.12345 16.98690
## 106 17.21467 17.91928 17.17287 16.95421 17.31126 17.87766
## 107 17.41641 17.56899 17.10729 16.89913 17.54558 17.37220
## 108 17.19368 16.82497 16.92183 17.56609 17.43539 17.41090
## 109 17.34536 17.50101 17.76866 17.02541 17.53574 17.61490
## 110 17.02147 17.28326 17.37684 16.72398 17.52209 16.98099
## 111 17.02509 17.71610 17.29702 17.55741 17.75017 17.51039
## 112 17.73951 17.42530 17.33663 17.47220 17.53752 17.06930
## 113 17.50164 17.48065 17.26278 17.16623 17.55440 17.27639
## 114 18.23345 17.81553 18.17827 18.09933 17.69942 17.65341
## 115 17.62943 17.47649 18.10245 17.90084 17.99219 17.47669
## 116 18.45818 18.05622 18.45681 18.19722 18.14112 17.95181
## 117 18.21603 17.80589 17.84945 17.59105 18.19029 17.97496
## 118 17.27462 18.30074 17.81092 17.09640 18.16314 17.30171
```