Working with Copenicus

Victor Nyabuti Ong'era

2022-10-28

How to download data from Copenicus

First you should have your area that you are intrested in.To selct an area go to Google earth.

Turn on the grids by going to the view > check grids. i have highlighted in yellow the grids section and also a place where you can read in real time the coordinates of the area we are intrested in

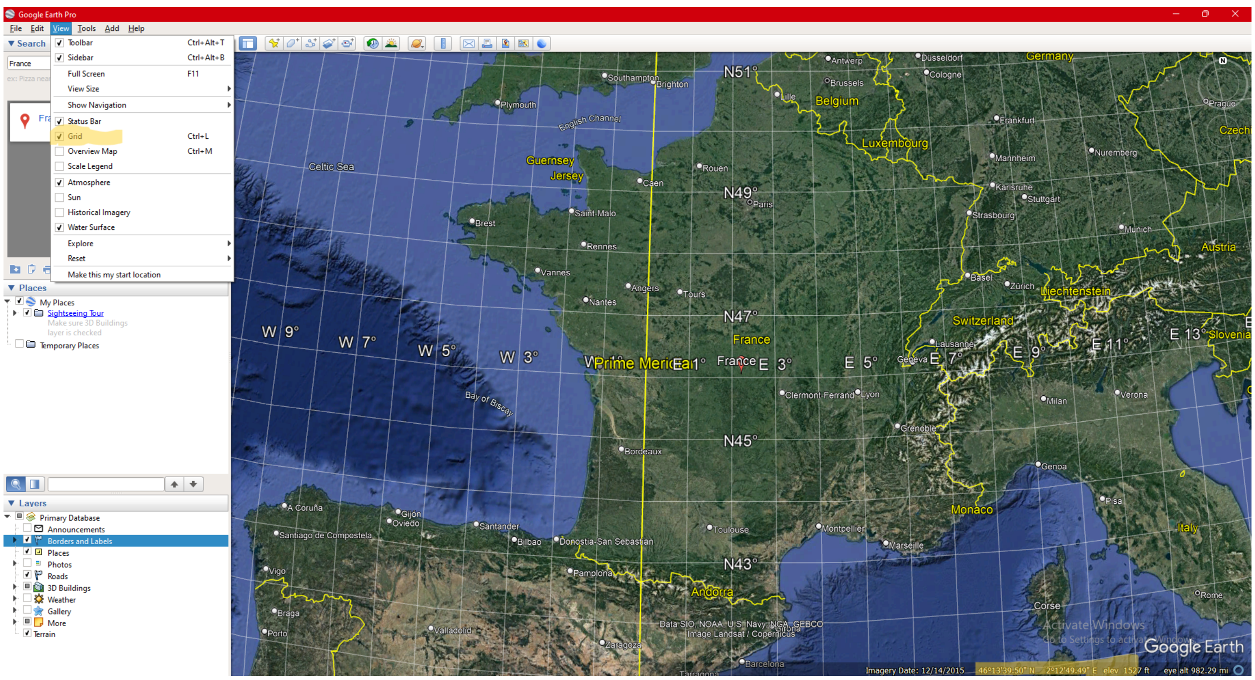
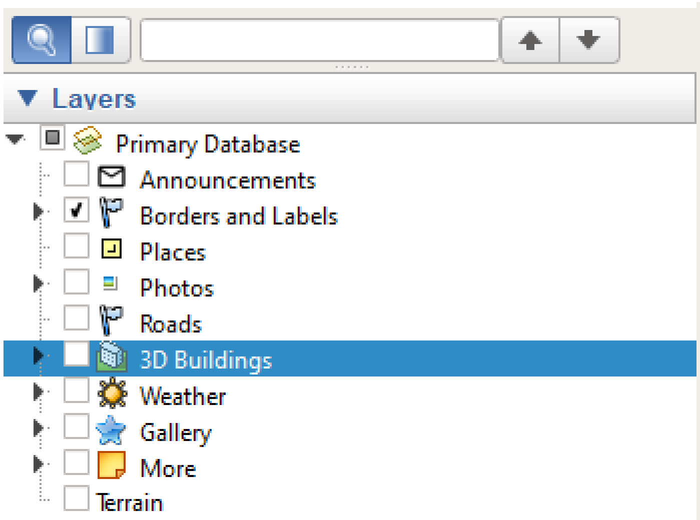


Figure 1: A caption

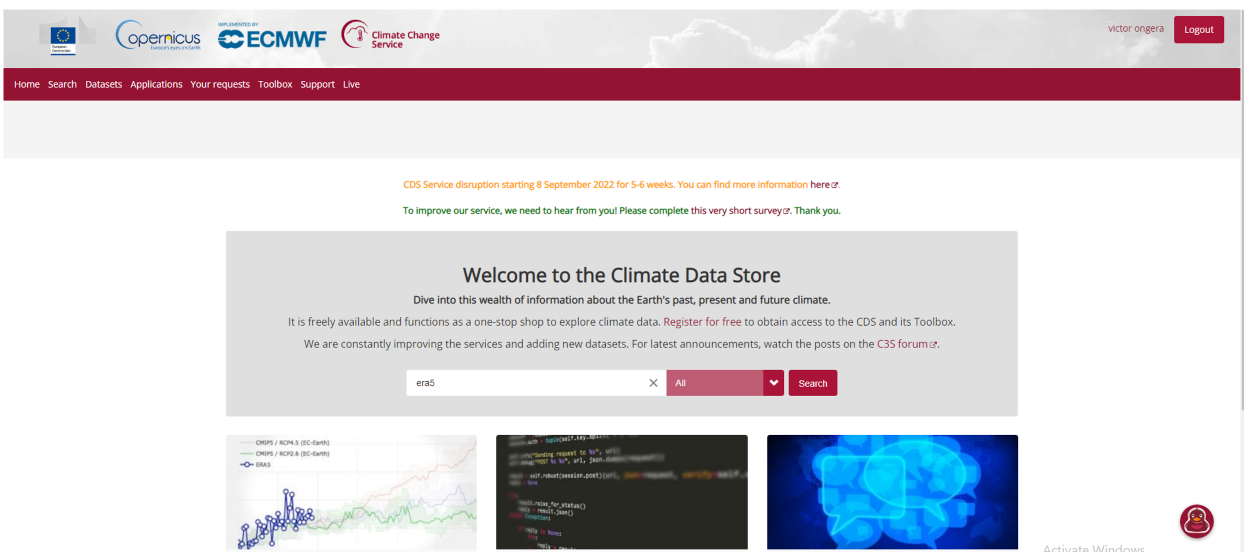
sometimes when zooming in you might end up not see the grids clearlz so it can be beneficial to turn off other layers at the lower left part of the screen



when you have your coordinates,Head to their website, sign up.

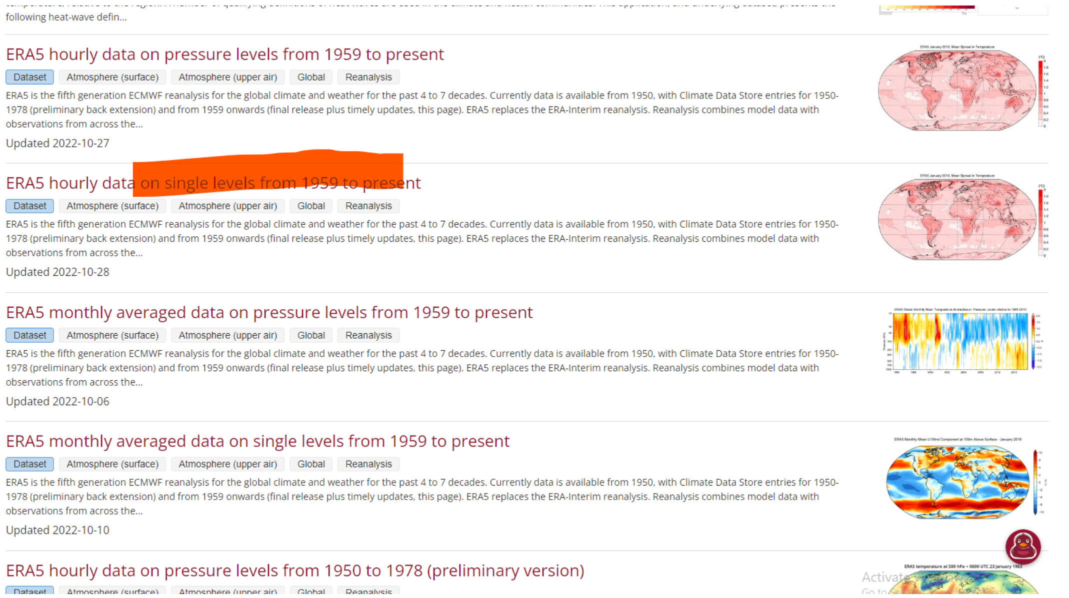
Look for era5

```
path6 <- "C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/data/Copenicus/copenicus.png
path7 <- "C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/data/Copenicus/copenicus1.pn
path8 <-  "C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/data/Copenicus/copenicus3.p
path9 <- "C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/data/Copenicus/copenicus4.pn
knitr::include_graphics(path6)
```



we use *ERA5 hourly data on single levels from 1959 to present*

```
knitr::include_graphics(path7)
```



Select the data set

you will land at the overview tab

Next to it we have the *Download data*

```
knitr::include_graphics(path8)
```

Check the following:

Product type :Monthly averaged reanalysis

variable : 2m tenoerature

year: All except 2022 because the last year is not complete so the matrix has 4 deimesnsion instead of 3.

Month : All

Day : All

Day: All

Time : 00:00

Region : Subregion. put the cordinates of the are szthat zu wangt to download the data from

Format: NetCDF (experimental)

Then finally jut download. It wll take a few minutes depending on your area

```
knitr::include_graphics(path9)
```

Submission date	End date	Duration	Size	Status	
2022-10-28 15:33:12	2022-10-28 15:36:20	0:03:08	13.0 MB	Download	

Now we have the data. Lets have a look

```
library(ncdf4) #Package that is necessary to install and then to upload
#in order to read the netCDF files
library(lubridate) #Package that is necessary to install and then to upload
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#in order to find the leap years
#path where there are the input files and where I will put the output files
path1 <- "C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/data/Copenicus/"

#nc_open() is the function included in the ncdf4 package necessary
#to use to read a netCDF file
nc <- nc_open(paste(path1,"adaptor.mars.internal-1666964178.6106884-18053-4-9f44f851-2eaf-494c-9472-e2b
#I print in the console all the information included in the netCDF
#file that are now saved in the nc variable
#I can check the variables included and their name. The unit of the variables,
#the size and so on..
nc
```

```
## File C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/data/Copenicus/adaptor.mars.in
##
##      1 variables (excluding dimension variables):
##         short t2m[longitude,latitude,time]
##             scale_factor: 0.000350696255846291
##             add_offset: 295.30390180031
##             _FillValue: -32767
##             missing_value: -32767
```

```
##             units: K
##             long_name: 2 metre temperature
##
##      3 dimensions:
##         longitude  Size:297
##             units: degrees_east
##             long_name: longitude
##         latitude  Size:29
##             units: degrees_north
##             long_name: latitude
##         time  Size:756
##             units: hours since 1900-01-01 00:00:00.0
##             long_name: time
##             calendar: gregorian
##
##     2 global attributes:
##         Conventions: CF-1.6
##         history: 2022-10-28 13:36:20 GMT by grib_to_netcdf-2.25.1: /opt/ecmwf/mars-client/bin/grib_to
```

Lets have a closer look at this data. The values we are intrested in are:

Variables we have a total of 6
Units In kelvins
time in days since 1850-01-01 00:00:00
latitude and long 36 N 72 E
Glbal attributes that give a bit more information on the whole thing

It says that we have 2m air temp over land blended with sea water temparature at a depth of 20cm

Lets extract this data

```r
#ncvar_get() is the function included in the ncdf4 package necessary
#to read a variable saved in the netCDF file
data <- ncvar_get(nc,"t2m") #Matrix three-dimensional
longitude <- ncvar_get(nc,"longitude") #Vector
latitude <- ncvar_get(nc,"latitude") #Vector

time <- ncvar_get(nc,"time") #Vector
nhours <- length(time)
```

Lets do some cleaning

```r
#leap_year() is a function included in the lubridate package that is necessary
#to find if a year is leap
#It get a vector with FALSE if a year is not leap and TRUE if a year is leap
#I consider the first and the last possible year
year_leap <- leap_year(1900:2021)
start_year <- 1900 #defines the start year
end_year <- 2021 #defines the lasz year
years <- c(1900:2021)#prints all the years
n_year <- end_year-start_year+1 #number of years
```

Time

Remember that our time is in hours.
We have 24 hours per day

So how do we calculate our time?
Our first Month is January. At that time we had no data from previous months therefore time is 0.
On our subsequent Months lets say february time is

```
No_of_days(nday)*hours(24)+ time from previous month
```

Lets make a matrix with this information

```
day_sequence <- matrix(ncol=3,nrow=(n_year*12))#empty matrix
day_sequence[,1] <- rep(c(start_year:end_year),each=12)#replicates each year 12 times
day_sequence[,2] <- rep(c(1:12),times=n_year)#replicates 1:12 in all years
```

Notefor the if fucntion we do not need to write a the brackets if the whole fucntion is on the same line but if they are in different lines then we need brackets

```
for(i in 1:length(day_sequence[,1])){
#for all years. found in the first column of the matrix day_sequence
  month <- day_sequence[i,2]
    #moths are in column 2
  year <- day_sequence[i,1]
    #years are in column 1
  if(month==1 | month==3 | month==5 | month==7 | month==8 | month==10 | month==12) nd <- 31
#if the month has 31 days i.e jan, march etc allocate 31 to the variable nd(number of days)
  if(month==4 | month==6 | month==9 | month==11) nd <- 30
#if the month has 30 days i.e jan, march etc allocate 30 to the variable nd(number of days)
  if(month==2 & !leap_year(year)) nd <- 28
#if the month is 2/feb and the year is not leap put 28
  if(month==2 & leap_year(year)) nd <- 29
#if the month is 2/feb and the year  leap put 29
  if(i==1) {
    day_sequence[i,3] <- 0
    nhours <- nd*24
  }
#if i(the first year)then the third column will have a value of 0 remeber why juary has no value?
#otherwise if the value is not 1 then the third column will have an added value from
#nhours(previous month)+ values from the current month
  if(i!=1){
    day_sequence[i,3] <- nhours+day_sequence[i-1,3]
    nhours <- nd*24
  }
}
```

Now lets filter this data to just our specific area

```
#I check which time element are present in my data. day sequence includes all
# the possible times from the beginning to the end my it is not mandatory that all these times are pres
time_ok <- vector()
for(i in 1:length(time)){
  time_ok[i] <- which(time[i]==day_sequence[,3])
```

```
}
day_sequence <- day_sequence[time_ok,]#new day sequence

#The matrix with the data includes the monthly values for all the available grid-points
#I cut a part and I calculate the monthly mean for the selected area
#to get the are you can diretly write them from the area ypu downloaded the data from
#latitude which return all latitudes
#longitude which returns al the löngitudes
latitude
```

```
##  [1]  3.00  2.75  2.50  2.25  2.00  1.75  1.50  1.25  1.00  0.75  0.50  0.25
## [13]  0.00 -0.25 -0.50 -0.75 -1.00 -1.25 -1.50 -1.75 -2.00 -2.25 -2.50 -2.75
## [25] -3.00 -3.25 -3.50 -3.75 -4.00
```

```
area_lon_min <- -33
area_lon_max <- 41
area_lat_min <- -4
area_lat_max <-  3

#I search the grid-points included in the selected area
lat_selected <- which(latitude>=area_lat_min & latitude<=area_lat_max)
lon_selected <- which(longitude>=area_lon_min & longitude<=area_lon_max)

#Number of point included in the selected area
npoint_selected <- length(lat_selected)*length(lon_selected)
npoint_selected
```

```
## [1] 8613
```

```
#Coordinates of the selected from-points
grid_selected <- matrix(ncol=2,nrow=npoint_selected)#empty matrix
head(grid_selected)
```

```
##      [,1] [,2]
## [1,]   NA   NA
## [2,]   NA   NA
## [3,]   NA   NA
## [4,]   NA   NA
## [5,]   NA   NA
## [6,]   NA   NA
```

```
colnames(grid_selected) <- c("Latitude","Longitude")
head(grid_selected)
```

```
##      Latitude Longitude
## [1,]       NA        NA
## [2,]       NA        NA
## [3,]       NA        NA
## [4,]       NA        NA
## [5,]       NA        NA
## [6,]       NA        NA
```

```r
ii <- 0
for(i in 1:length(lat_selected)){
  for(j in 1:length(lon_selected)){
    ii <- ii+1
    grid_selected[ii,"Latitude"] <- latitude[lat_selected[i]]
    grid_selected[ii,"Longitude"] <- longitude[lon_selected[j]]
  }
}

head(grid_selected)
```

```
##      Latitude Longitude
## [1,]        3    -33.00
## [2,]        3    -32.75
## [3,]        3    -32.50
## [4,]        3    -32.25
## [5,]        3    -32.00
## [6,]        3    -31.75
```

```r
path <- "C:/Users/VICTOR_NYABUTI/Climate/climate_change_global_adaptation/output/Copenicus/ "

#export as table

write.table(grid_selected,paste(path,"Grid_points_",area_lon_min,"_",area_lon_max,"_",area_lat_min,"_",a

#For each month I extract the temperature anomaly data of the selected area and I calculate the area me
temp_selected <- vector()
for(i in 1:length(day_sequence[,1])){
  data_selected <- data[lon_selected,lat_selected,i]
  temp_selected[i] <- mean(data_selected,na.rm=TRUE)-273.15
}

temp_selected <- cbind(day_sequence,temp_selected)
colnames(temp_selected) <- c("Year","Month","Number_of_hours","Mean")

write.table(temp_selected,paste(path,"Temperature_anomaly_mean_",area_lon_min,"_",area_lon_max,"_",area_

number_of_year<- day_sequence[length(day_sequence[,1]),1]-day_sequence[1,1]+1
temp_matrix <- matrix(data=temp_selected[,"Mean"],ncol=12,nrow=number_of_year,byrow=TRUE)
temp_matrix <- cbind(c(day_sequence[1,1]:day_sequence[length(day_sequence[,1]),1]),temp_matrix)
colnames(temp_matrix) <- c("Year","January","February","March","April","May","June","July","August","Se

write.table(temp_matrix,paste(path,"Monthly_temperature_anomaly_mean_matrix_",area_lon_min,"_",area_lon_
```

ASSIGNMENT ATTEMPT

After this I can perform. The analysis which we did in previous data.

```r
temp = as.matrix(temp_matrix)

n_column = length(temp[1,])  # number of columns
```

```r
n_rows   = length(temp[,1]) #  number of rows
n_column
```

```
## [1] 13
```

```r
n_rows
```

```
## [1] 63
```

```r
#Number of years
#
start_year <- temp[1,1]
start_year
```

```
## Year
## 1959
```

```r
end_year <- temp[length(temp[,1]),1]
end_year
```

```
## Year
## 2021
```

```r
n_year <- end_year-start_year+1

#yearly average in a matrix

annual_mean_1 <- matrix(ncol=2,nrow=n_year) #empty ,matrix


annual_mean_1[,1] <- start_year:end_year # defined what is the column, the row by [,1]

head(annual_mean_1) #now the matrix has a column
```

```
##      [,1] [,2]
## [1,] 1959   NA
## [2,] 1960   NA
## [3,] 1961   NA
## [4,] 1962   NA
## [5,] 1963   NA
## [6,] 1964   NA
```

```r
for(i in 1:n_year){
  annual_mean_1[i,2] <- mean(temp[i,2:13],na.rm=FALSE)
}
head(annual_mean_1)
```

```
##      [,1]     [,2]
## [1,] 1959 24.42625
## [2,] 1960 24.42179
```

```
## [3,] 1961 24.27137
## [4,] 1962 24.25011
## [5,] 1963 24.51780
## [6,] 1964 24.27103
```

```
colnames(annual_mean_1) <- c("Year","Annual_mean") #name of columns.
head(annual_mean_1)
```

```
##      Year Annual_mean
## [1,] 1959    24.42625
## [2,] 1960    24.42179
## [3,] 1961    24.27137
## [4,] 1962    24.25011
## [5,] 1963    24.51780
## [6,] 1964    24.27103
```

```
write.table(annual_mean_1,paste(path,"Yearly_average_temperature_anomaly.csv",sep=""),sep=";",col.names=


# so now  we need to do some plotinng

# we are using the package ggplot

#to use the data we need to convert the dat into a data frame. ggplot works witth data frames
annual_mean <- as.data.frame(annual_mean_1)

library(ggplot2)


moving_average <- matrix(ncol=4,nrow=n_year) #an empty matrix where we will put our means

head(moving_average)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   NA   NA   NA   NA
## [2,]   NA   NA   NA   NA
## [3,]   NA   NA   NA   NA
## [4,]   NA   NA   NA   NA
## [5,]   NA   NA   NA   NA
## [6,]   NA   NA   NA   NA
```

```
colnames(moving_average) <- c("Year","Window_1","Window_2","Window_3")#inserting column names

head(moving_average)
```

```
##      Year Window_1 Window_2 Window_3
## [1,]   NA       NA       NA       NA
## [2,]   NA       NA       NA       NA
## [3,]   NA       NA       NA       NA
## [4,]   NA       NA       NA       NA
## [5,]   NA       NA       NA       NA
## [6,]   NA       NA       NA       NA
```

```
moving_average[,1] <- start_year:end_year #years for the first column
head(moving_average)
```

```
##      Year Window_1 Window_2 Window_3
## [1,] 1959       NA       NA       NA
## [2,] 1960       NA       NA       NA
## [3,] 1961       NA       NA       NA
## [4,] 1962       NA       NA       NA
## [5,] 1963       NA       NA       NA
## [6,] 1964       NA       NA       NA
```

```
#1 window
window_1 <- 5


start_1 <- ((window_1-1)/2)+1 #First row where I can calculate the moving average

start_1
```

```
## [1] 3
```

```
end_1 <- n_year-((window_1-1)/2) #Last row where I can calculate the moving average

end_1
```

```
## Year
##   61
```

```
for(i in start_1:end_1){
  moving_average[i,"Window_1"] <- mean(annual_mean[(i-((window_1-1)/2)):(i+((window_1-1)/2)),"Annual_mea
}

head(moving_average)
```

```
##      Year Window_1 Window_2 Window_3
## [1,] 1959       NA       NA       NA
## [2,] 1960       NA       NA       NA
## [3,] 1961 24.37746       NA       NA
## [4,] 1962 24.34642       NA       NA
## [5,] 1963 24.31662       NA       NA
## [6,] 1964 24.38978       NA       NA
```

```
#2 window
window_2 <- 11
start_2 <- ((window_2-1)/2)+1 #First row where I can calculate the moving average
end_2 <- n_year-((window_2-1)/2) #Last row where I can calculate the moving average
start_2
```

```
## [1] 6
```

```r
end_2
```

```
## Year
##   58
```

```r
for(i in start_2:end_2){
  moving_average[i,"Window_2"] <- mean(annual_mean[(i-((window_2-1)/2)):(i+((window_2-1)/2)),"Annual_mea

}

head(moving_average)
```

```
##      Year Window_1 Window_2 Window_3
## [1,] 1959       NA       NA       NA
## [2,] 1960       NA       NA       NA
## [3,] 1961 24.37746       NA       NA
## [4,] 1962 24.34642       NA       NA
## [5,] 1963 24.31662       NA       NA
## [6,] 1964 24.38978 24.36785       NA
```

```r
#3 window
window_3 <- 21
start_3 <- ((window_3-1)/2)+1 #First row where I can calculate the moving average
end_3 <- n_year-((window_3-1)/2) #Last row where I can calculate the moving average
start_3
```

```
## [1] 11
```

```r
end_3
```

```
## Year
##   53
```

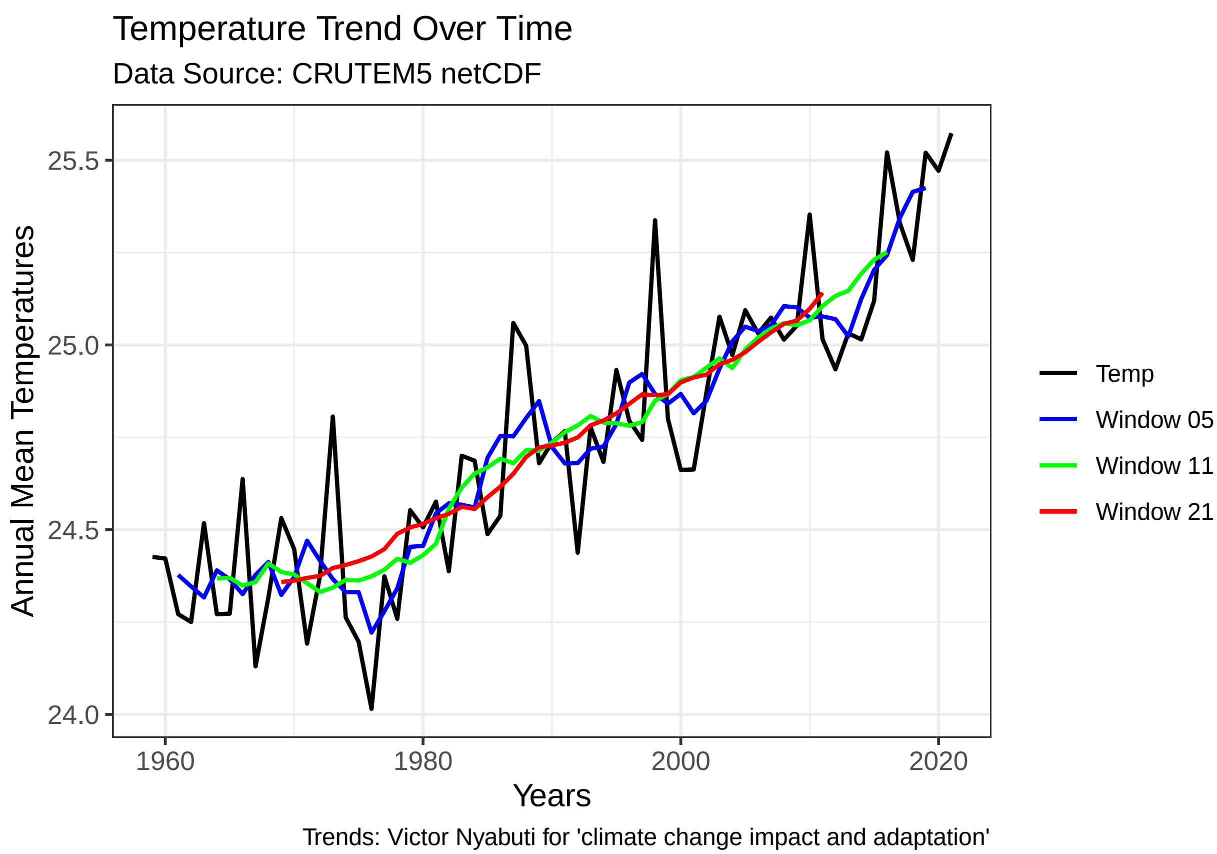
```r
for(i in start_3:end_3){
  moving_average[i,"Window_3"] <- mean(annual_mean[(i-((window_3-1)/2)):(i+((window_3-1)/2)),"Annual_mea
}

tail(moving_average)
```

```
##       Year Window_1 Window_2 Window_3
## [58,] 2016 25.24304   25.251       NA
## [59,] 2017 25.34417       NA       NA
## [60,] 2018 25.41450       NA       NA
## [61,] 2019 25.42490       NA       NA
## [62,] 2020       NA       NA       NA
## [63,] 2021       NA       NA       NA
```

```r
write.table(moving_average,paste(path,"Moving_averages.csv",sep=""),sep=";",col.names=TRUE,row.names=FAL


#some plotting
```

```
moving_average <- as.data.frame(moving_average)

#I plot the yearly average temperature anomaly and the three different moving averages

ggplot()+ #I create a grey paper
  geom_line(data=annual_mean,aes(x=Year,y=Annual_mean,colour="Temp"),size=.75)+ #I add a line.
  #The data are included in the dataframe annual_mean. On the X-axis I put the Years while on
  #the Y-axis I put the annual mean. I decide the size of the line and also the colour
  geom_line(data=moving_average,aes(x=Year,y=Window_1,colour="Window 05"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_2,colour="Window 11"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_3,colour="Window 21"),size=.75)+
  xlab("Year")+ #I change the title of the X-axis
  ylab("Annual mean temperature [?C]")+ #I change the title of the Y-axis
  theme_bw()+ #I change the background from gray that is the default to white
  theme(axis.title.x = element_text(size=12), # I increase the size of the X-axis title
        #(You have to set hese numbers manually)
        axis.title.y = element_text(size=12, angle=90), # I increase the size of the
        #Y-axis title and I rotate it of an angle equal to 90?
        axis.text.x = element_text(size=10),
        axis.text.y = element_text(size=10),
        legend.text=element_text(size=9))+
  labs(title = "Temperature Trend Over Time") +
  labs(subtitle = "Data Source: CRUTEM5 netCDF") +
  labs(col = "") +
  labs(x = "Years") +
  labs(y = "Annual Mean Temperatures") +
  labs(caption = "Trends: Victor Nyabuti for 'climate change impact and adaptation'")+
  scale_colour_manual(values=c("black","blue","green","red"))+ #I add the legend
  labs(colour="") # I remuve the title of the legend
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```




```
ggsave(file="Ex_Annual_mean_temperature_anomaly.png",path=path,dpi=500,width=40,height=40,units="cm")
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```

```
#->


#I calculate the trend over the whole period with the function lm
y <- annual_mean[,"Annual_mean"]
x <- annual_mean[,"Year"]
trend <- lm(y ~ x)
summary(trend)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##       Min       1Q   Median       3Q      Max
## -0.47207 -0.13093 -0.02039  0.09529  0.44883
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.588343   2.671023  -4.339 5.48e-05 ***
## x             0.018257   0.001342  13.602  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1937 on 61 degrees of freedom
## Multiple R-squared:  0.7521, Adjusted R-squared:  0.748
## F-statistic:   185 on 1 and 61 DF,  p-value: < 2.2e-16
```

```
#extract this values
slope <- as.numeric(coef(summary(trend))[, "Estimate"][2])
intercept <- as.numeric(coef(summary(trend))[, "Estimate"][1])
slope_error <- as.numeric(coef(summary(trend))[, "Std. Error"][2])
intercept_error <- as.numeric(coef(summary(trend))[, "Std. Error"][1])
significance <- as.numeric(coef(summary(trend))[, "Pr(>|t|)"][1])

#I put the results in a matrix
trend_results <- matrix(ncol=2,nrow=5)
trend_results[1,1] <- "Slope" #name of row
trend_results[1,2] <- slope
trend_results[2,1] <- "Intercept"
trend_results[2,2] <- intercept
trend_results[3,1] <- "Slope_error"
trend_results[3,2] <- slope_error
trend_results[4,1] <- "Intercept_error"
trend_results[4,2] <- intercept_error
trend_results[5,1] <- "Significance"
trend_results[5,2] <- significance
trend_results
```

```
##      [,1]              [,2]
## [1,] "Slope"           "0.0182567153545399"
## [2,] "Intercept"       "-11.5883430646969"
## [3,] "Slope_error"     "0.0013421667769696"
## [4,] "Intercept_error" "2.67102339384543"
## [5,] "Significance"    "5.48035568287919e-05"
```

```
write.table(trend_results,paste(path,"Trend_results.csv",sep=""),sep=";",col.names=FALSE,row.names=FALSE




#I calculate the residuals of the anomaly from the low-pass-filter
#I calculate the standard deviation of the yearly temperature anomaly

standard_deviation_anomaly <- sd(annual_mean[,2])


#I calculate the residuals of the anomaly from the low-pass-filter
```

```r
residuals <- matrix(ncol=2,nrow=n_year)#empty matrix

head(residuals)
```

```
##      [,1] [,2]
## [1,]   NA   NA
## [2,]   NA   NA
## [3,]   NA   NA
## [4,]   NA   NA
## [5,]   NA   NA
## [6,]   NA   NA
```

```r
colnames(residuals) <- c("Year","Residuals")

head(residuals)
```

```
##      Year Residuals
## [1,]   NA        NA
## [2,]   NA        NA
## [3,]   NA        NA
## [4,]   NA        NA
## [5,]   NA        NA
## [6,]   NA        NA
```

```r
residuals[,"Year"] <- start_year:end_year

head(residuals)
```

```
##      Year Residuals
## [1,] 1959        NA
## [2,] 1960        NA
## [3,] 1961        NA
## [4,] 1962        NA
## [5,] 1963        NA
## [6,] 1964        NA
```

```r
residuals[,"Residuals"] <- annual_mean[,"Annual_mean"]-moving_average[,"Window_2"]

print(residuals)
```

```
##       Year     Residuals
##  [1,] 1959            NA
##  [2,] 1960            NA
##  [3,] 1961            NA
##  [4,] 1962            NA
##  [5,] 1963            NA
##  [6,] 1964 -0.0968126411
##  [7,] 1965 -0.0969929953
##  [8,] 1966  0.2883685371
##  [9,] 1967 -0.2282149500
```

```
## [10,] 1968 -0.0919149890
## [11,] 1969  0.1457568455
## [12,] 1970  0.0686165488
## [13,] 1971 -0.1637439619
## [14,] 1972  0.0424478421
## [15,] 1973  0.4632086737
## [16,] 1974 -0.1021180101
## [17,] 1975 -0.1654112189
## [18,] 1976 -0.3590548602
## [19,] 1977 -0.0176764037
## [20,] 1978 -0.1626289017
## [21,] 1979  0.1421328769
## [22,] 1980  0.0751641939
## [23,] 1981  0.1137644019
## [24,] 1982 -0.1696290206
## [25,] 1983  0.0863012927
## [26,] 1984  0.0348323545
## [27,] 1985 -0.1808698391
## [28,] 1986 -0.1545127959
## [29,] 1987  0.3798341884
## [30,] 1988  0.2828300397
## [31,] 1989 -0.0339338506
## [32,] 1990  0.0001215559
## [33,] 1991  0.0027700672
## [34,] 1992 -0.3450068261
## [35,] 1993 -0.0308666973
## [36,] 1994 -0.1065119592
## [37,] 1995  0.1439117718
## [38,] 1996  0.0135220185
## [39,] 1997 -0.0483919977
## [40,] 1998  0.4880825474
## [41,] 1999 -0.0667549260
## [42,] 2000 -0.2427362715
## [43,] 2001 -0.2505924098
## [44,] 2002 -0.0652329035
## [45,] 2003  0.1129205595
## [46,] 2004  0.0350962444
## [47,] 2005  0.1061476909
## [48,] 2006  0.0112673981
## [49,] 2007  0.0295408025
## [50,] 2008 -0.0446921515
## [51,] 2009 -0.0013644002
## [52,] 2010  0.2863690074
## [53,] 2011 -0.0903737727
## [54,] 2012 -0.1986839563
## [55,] 2013 -0.1151557143
## [56,] 2014 -0.1780413830
## [57,] 2015 -0.1113629511
## [58,] 2016  0.2701279985
## [59,] 2017            NA
## [60,] 2018            NA
## [61,] 2019            NA
## [62,] 2020            NA
## [63,] 2021            NA
```

```
standard_deviation_residuals <- sd(residuals[,"Residuals"],na.rm=TRUE)
standard_deviation_anomaly
```

```
## [1] 0.3858926
```

Trial and Error

Area 2

```
#area 2
print(74*2)
```

```
## [1] 148
```

```
print(297-148)
```

```
## [1] 149
```

```
longitude[148]
```

```
## [1] 3.75
```

```
longitude[149]
```

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

```
area_lon_min <- -3.75
area_lon_max <- 4
area_lat_min <- -3
area_lat_max <-  2

temp = as.matrix(temp_matrix)
n_column = length(temp[1,])
n_rows   = length(temp[,1])
start_year <- temp[1,1]
start_year
```

```
## Year
## 1959
```

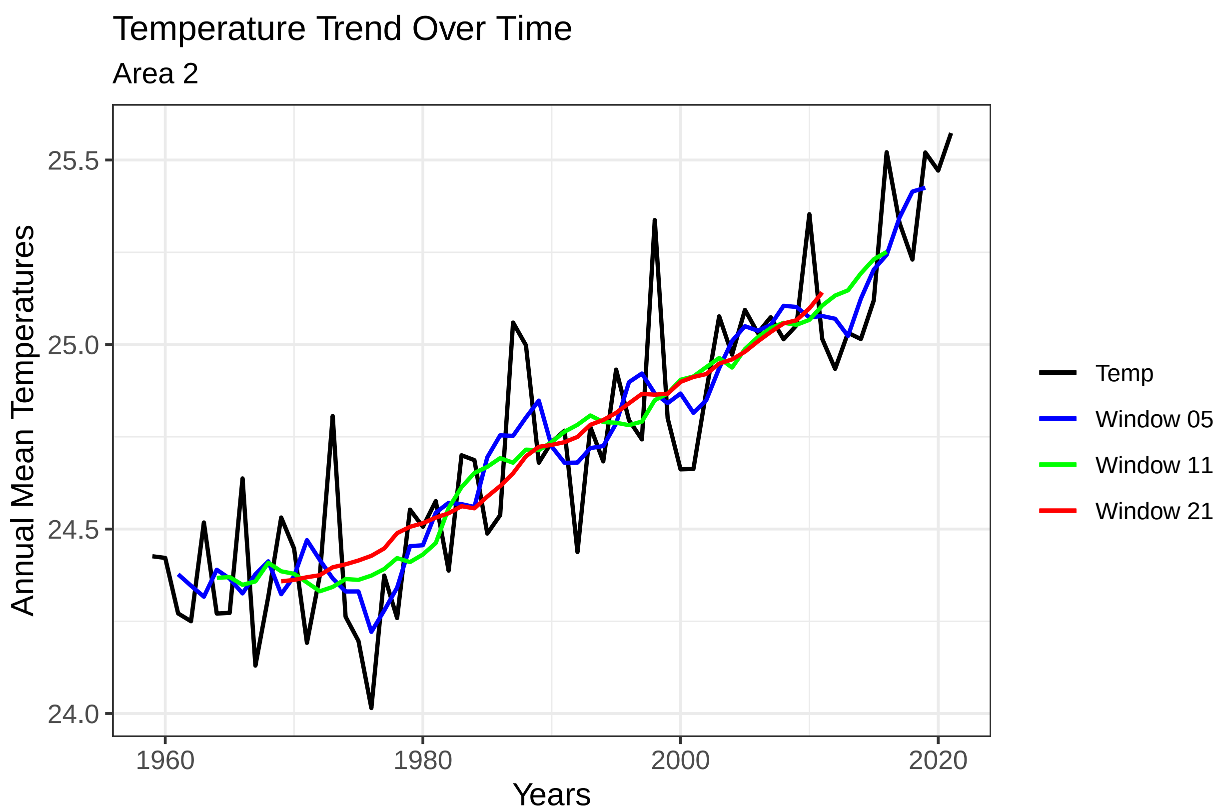
```
end_year <- temp[length(temp[,1]),1]
end_year
```

```
## Year
## 2021
```

```
n_year <- end_year-start_year+1

#yearly average in a matrix

annual_mean_1 <- matrix(ncol=2,nrow=n_year) #empty ,matrix


annual_mean_1[,1] <- start_year:end_year # defined what is the column, the row by [,1]


for(i in 1:n_year){
  annual_mean_1[i,2] <- mean(temp[i,2:13],na.rm=FALSE)
}

colnames(annual_mean_1) <- c("Year","Annual_mean") #name of columns.

write.table(annual_mean_1,paste(path,"Yearly_average_temperature_anomaly.csv",sep=""),sep=";",col.names=




annual_mean <- as.data.frame(annual_mean_1)
moving_average <- matrix(ncol=4,nrow=n_year) #an empty matrix where we will put our means

colnames(moving_average) <- c("Year","Window_1","Window_2","Window_3")#inserting column names

moving_average[,1] <- start_year:end_year #years for the first column

#1 window
window_1 <- 5


start_1 <- ((window_1-1)/2)+1 #First row where I can calculate the moving average

end_1 <- n_year-((window_1-1)/2) #Last row where I can calculate the moving average

for(i in start_1:end_1){
  moving_average[i,"Window_1"] <- mean(annual_mean[(i-((window_1-1)/2)):(i+((window_1-1)/2)),"Annual_mea
}



#2 window
window_2 <- 11
start_2 <- ((window_2-1)/2)+1 #First row where I can calculate the moving average
end_2 <- n_year-((window_2-1)/2) #Last row where I can calculate the moving average

for(i in start_2:end_2){
  moving_average[i,"Window_2"] <- mean(annual_mean[(i-((window_2-1)/2)):(i+((window_2-1)/2)),"Annual_mea
}


#3 window
```

```
window_3 <- 21
start_3 <- ((window_3-1)/2)+1 #First row where I can calculate the moving average
end_3 <- n_year-((window_3-1)/2) #Last row where I can calculate the moving average

for(i in start_3:end_3){
  moving_average[i,"Window_3"] <- mean(annual_mean[(i-((window_3-1)/2)):(i+((window_3-1)/2)),"Annual_mea
}



#some plotting

moving_average <- as.data.frame(moving_average)

library(ggplot2)
ggplot()+ #I create a grey paper
  geom_line(data=annual_mean,aes(x=Year,y=Annual_mean,colour="Temp"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_1,colour="Window 05"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_2,colour="Window 11"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_3,colour="Window 21"),size=.75)+
  xlab("Year")+ #I change the title of the X-axis
  ylab("Annual mean temperature [?C]")+ #I change the title of the Y-axis
  theme_bw()+ #I change the background from gray that is the default to white
  theme(axis.title.x = element_text(size=12), # I increase the size of the X-axis title
        #(You have to set hese numbers manually)
        axis.title.y = element_text(size=12, angle=90), # I increase the size of the
        #Y-axis title and I rotate it of an angle equal to 90?
        axis.text.x = element_text(size=10),
        axis.text.y = element_text(size=10),
        legend.text=element_text(size=9))+
  labs(title = "Temperature Trend Over Time") +
  labs(subtitle = "Area 2") +
  labs(col = "") +
  labs(x = "Years") +
  labs(y = "Annual Mean Temperatures") +
  labs(caption = "Trends: Victor Nyabuti for 'climate change impact and adaptation'")+
  scale_colour_manual(values=c("black","blue","green","red"))+ #I add the legend
  labs(colour="") # I remuve the title of the legend
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```



Trends: Victor Nyabuti for 'climate change impact and adaptation'

```
ggsave(file="area2_Annual_mean_temperature_anomaly.png",path=path,dpi=500,width=40,height=40,units="cm"
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```

Area 3

```
#area 3
print(74*3)
```

```
## [1] 222
```

```
print(297-222)
```

```
## [1] 75
```

```r
longitude[222]
```

```
## [1] 22.25
```

```r
longitude[75]
```

```
## [1] -14.5
```

```r
area_lon_min <- -14.5
area_lon_max <- 22.5
area_lat_min <- -2
area_lat_max <-  1

temp = as.matrix(temp_matrix)
n_column = length(temp[1,])
n_rows   = length(temp[,1])
start_year <- temp[1,1]
end_year <- temp[length(temp[,1]),1]

n_year <- end_year-start_year+1

#yearly average in a matrix

annual_mean_1 <- matrix(ncol=2,nrow=n_year) #empty ,matrix


annual_mean_1[,1] <- start_year:end_year # defined what is the column, the row by [,1]

head(annual_mean_1) #now the matrix has a column
```

```
##      [,1] [,2]
## [1,] 1959   NA
## [2,] 1960   NA
## [3,] 1961   NA
## [4,] 1962   NA
## [5,] 1963   NA
## [6,] 1964   NA
```

```r
for(i in 1:n_year){
  annual_mean_1[i,2] <- mean(temp[i,2:13],na.rm=FALSE)
}
colnames(annual_mean_1) <- c("Year","Annual_mean") #name of columns.
head(annual_mean_1)
```

```
##      Year Annual_mean
## [1,] 1959    24.42625
## [2,] 1960    24.42179
## [3,] 1961    24.27137
## [4,] 1962    24.25011
## [5,] 1963    24.51780
## [6,] 1964    24.27103
```

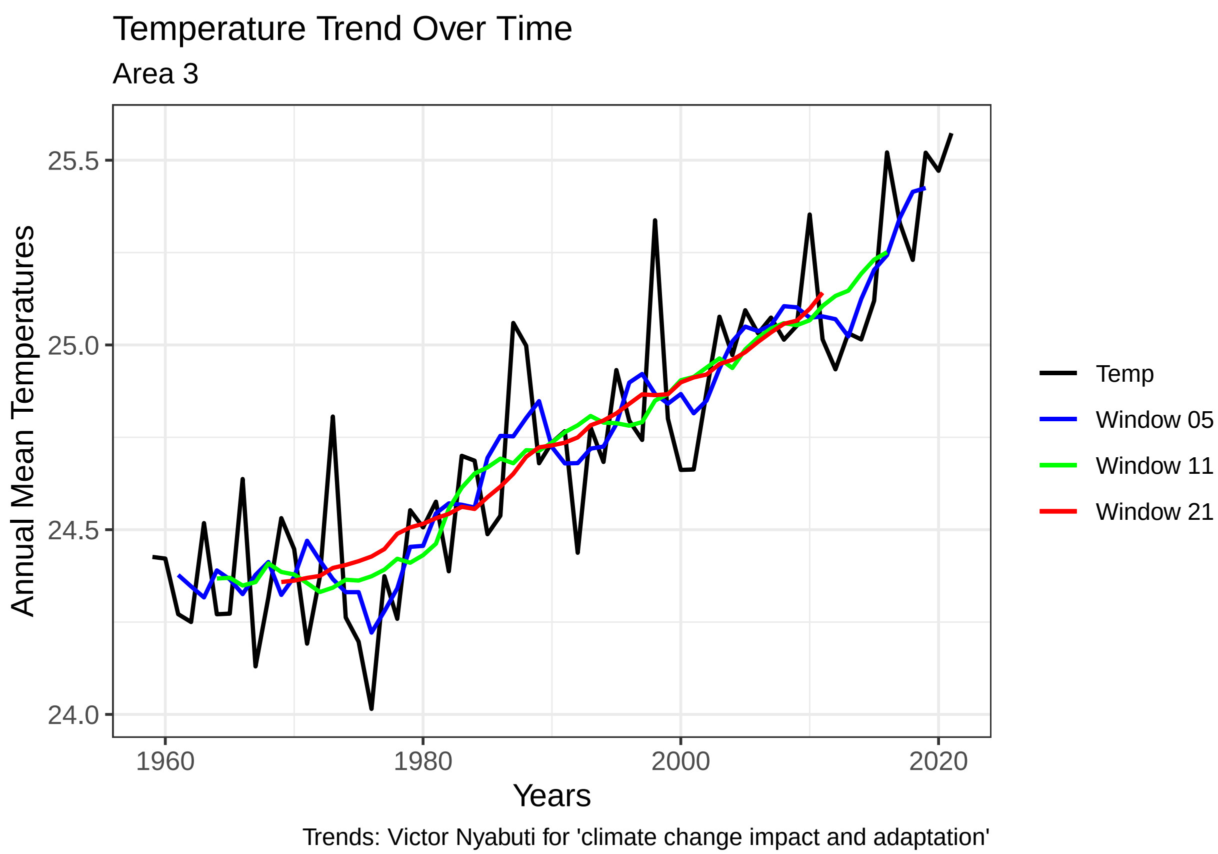
```r
write.table(annual_mean_1,paste(path,"Yearly_average_temperature_anomaly.csv",sep=""),sep=";",col.names=



annual_mean <- as.data.frame(annual_mean_1)
moving_average <- matrix(ncol=4,nrow=n_year) #an empty matrix where we will put our means

colnames(moving_average) <- c("Year","Window_1","Window_2","Window_3")#inserting column names

moving_average[,1] <- start_year:end_year #years for the first column

#1 window
window_1 <- 5


start_1 <- ((window_1-1)/2)+1 #First row where I can calculate the moving average

end_1 <- n_year-((window_1-1)/2) #Last row where I can calculate the moving average

for(i in start_1:end_1){
  moving_average[i,"Window_1"] <- mean(annual_mean[(i-((window_1-1)/2)):(i+((window_1-1)/2)),"Annual_mea
}



#2 window
window_2 <- 11
start_2 <- ((window_2-1)/2)+1 #First row where I can calculate the moving average
end_2 <- n_year-((window_2-1)/2) #Last row where I can calculate the moving average

for(i in start_2:end_2){
  moving_average[i,"Window_2"] <- mean(annual_mean[(i-((window_2-1)/2)):(i+((window_2-1)/2)),"Annual_mea
}


#3 window
window_3 <- 21
start_3 <- ((window_3-1)/2)+1 #First row where I can calculate the moving average
end_3 <- n_year-((window_3-1)/2) #Last row where I can calculate the moving average


for(i in start_3:end_3){
  moving_average[i,"Window_3"] <- mean(annual_mean[(i-((window_3-1)/2)):(i+((window_3-1)/2)),"Annual_mea
}



#some plotting

moving_average <- as.data.frame(moving_average)


library(ggplot2)
```

```r
ggplot()+ #I create a grey paper
  geom_line(data=annual_mean,aes(x=Year,y=Annual_mean,colour="Temp"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_1,colour="Window 05"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_2,colour="Window 11"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_3,colour="Window 21"),size=.75)+
  xlab("Year")+
  ylab("Annual mean temperature [?C]")+
  theme_bw()+
  theme(axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=12, angle=90),
        axis.text.x = element_text(size=10),
        axis.text.y = element_text(size=10),
        legend.text=element_text(size=9))+
  labs(title = "Temperature Trend Over Time") +
  labs(subtitle = "Area 3") +
  labs(col = "") +
  labs(x = "Years") +
  labs(y = "Annual Mean Temperatures") +
  labs(caption = "Trends: Victor Nyabuti for 'climate change impact and adaptation'")+
  scale_colour_manual(values=c("black","blue","green","red"))+ #I add the legend
  labs(colour="") # I remuve the title of the legend
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 10 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```




```
ggsave(file="area3_Annual_mean_temperature_anomaly.png",path=path,dpi=500,width=40,height=40,units="cm"
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```

Area 4

```
#area 4
area_lon_min <- -3.75
area_lon_max <- 4
area_lat_min <-  0
area_lat_max <-  0

temp = as.matrix(temp_matrix)
n_column = length(temp[1,])
n_rows   = length(temp[,1])
start_year <- temp[1,1]

end_year <- temp[length(temp[,1]),1]
```

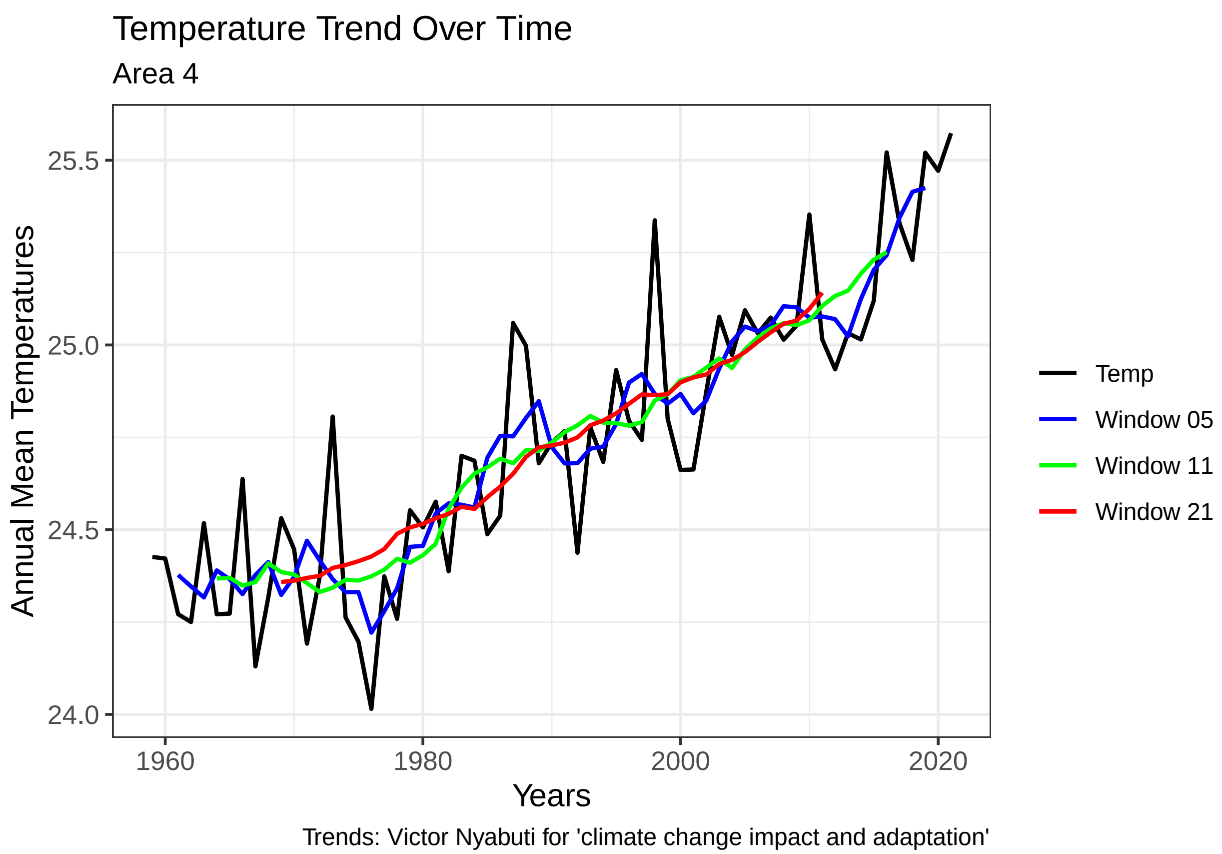
```r
n_year <- end_year-start_year+1

#yearly average in a matrix

annual_mean_1 <- matrix(ncol=2,nrow=n_year) #empty ,matrix


annual_mean_1[,1] <- start_year:end_year


for(i in 1:n_year){
  annual_mean_1[i,2] <- mean(temp[i,2:13],na.rm=FALSE)
}


colnames(annual_mean_1) <- c("Year","Annual_mean")
head(annual_mean_1)
```

```
##      Year Annual_mean
## [1,] 1959    24.42625
## [2,] 1960    24.42179
## [3,] 1961    24.27137
## [4,] 1962    24.25011
## [5,] 1963    24.51780
## [6,] 1964    24.27103
```

```r
write.table(annual_mean_1,paste(path,"Yearly_average_temperature_anomaly.csv",sep=""),sep=";",col.names=




annual_mean <- as.data.frame(annual_mean_1)
moving_average <- matrix(ncol=4,nrow=n_year)
colnames(moving_average) <- c("Year","Window_1","Window_2","Window_3")
moving_average[,1] <- start_year:end_year
#1 window
window_1 <- 5


start_1 <- ((window_1-1)/2)+1
end_1 <- n_year-((window_1-1)/2)
for(i in start_1:end_1){
  moving_average[i,"Window_1"] <- mean(annual_mean[(i-((window_1-1)/2)):(i+((window_1-1)/2)),"Annual_mea
}



#2 window
window_2 <- 11
start_2 <- ((window_2-1)/2)+1
end_2 <- n_year-((window_2-1)/2)
for(i in start_2:end_2){
  moving_average[i,"Window_2"] <- mean(annual_mean[(i-((window_2-1)/2)):(i+((window_2-1)/2)),"Annual_mea
```

```
}


#3 window
window_3 <- 21
start_3 <- ((window_3-1)/2)+1
end_3 <- n_year-((window_3-1)/2)
for(i in start_3:end_3){
  moving_average[i,"Window_3"] <- mean(annual_mean[(i-((window_3-1)/2)):(i+((window_3-1)/2)),"Annual_mea
}



#some plotting

moving_average <- as.data.frame(moving_average)

#I plot the yearly average temperature anomaly and the three different moving averages
library(ggplot2)
ggplot()+ #I create a grey paper
  geom_line(data=annual_mean,aes(x=Year,y=Annual_mean,colour="Temp"),size=.75)+ #I add a line.
  #The data are included in the dataframe annual_mean. On the X-axis I put the Years while on
  #the Y-axis I put the annual mean. I decide the size of the line and also the colour
  geom_line(data=moving_average,aes(x=Year,y=Window_1,colour="Window 05"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_2,colour="Window 11"),size=.75)+
  geom_line(data=moving_average,aes(x=Year,y=Window_3,colour="Window 21"),size=.75)+
  xlab("Year")+ #I change the title of the X-axis
  ylab("Annual mean temperature [?C]")+ #I change the title of the Y-axis
  theme_bw()+ #I change the background from gray that is the default to white
  theme(axis.title.x = element_text(size=12), # I increase the size of the X-axis title
        #(You have to set hese numbers manually)
        axis.title.y = element_text(size=12, angle=90), # I increase the size of the
        #Y-axis title and I rotate it of an angle equal to 90?
        axis.text.x = element_text(size=10),
        axis.text.y = element_text(size=10),
        legend.text=element_text(size=9))+
  labs(title = "Temperature Trend Over Time") +
  labs(subtitle = "Area 4") +
  labs(col = "") +
  labs(x = "Years") +
  labs(y = "Annual Mean Temperatures") +
  labs(caption = "Trends: Victor Nyabuti for 'climate change impact and adaptation'")+
  scale_colour_manual(values=c("black","blue","green","red"))+ #I add the legend
  labs(colour="") # I remuve the title of the legend
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```




```
ggsave(file="area4_Annual_mean_temperature_anomaly.png",path=path,dpi=500,width=40,height=40,units="cm")
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 20 row(s) containing missing values (geom_path).
```