Online Resource 01 - Supplementary appendix F
Selection process and data description

How does the temperature vary over time? Evidence on the Stationary and Fractal nature of Temperature Fluctuations

John Dagsvik, Mariachiara Fortuna, Sigmund H. Moen

Affiliations:

John K. Dagsvik, Statistics Norway, Research Department;

Mariachiara Fortuna, freelance statistician, Turin;

Sigmund Hov Moen, Westerdals Oslo School of Arts, Communication and Technology.

Corresponding author:

John K. Dagsvik, E-mail: john.dagsvik@ssb.no

Mariachiara Fortuna, E-mail: mariachiara.fortuna1@gmail.com (reference for code and analysis)

RAW DATA EXPLORATION

```
# LOADING PACKAGES

require(ggplot2)
require(dplyr)
require(knitr)
require(tempFGN)
require(tidyr)
require(lubridate)
```

```
# DATA PATH
data_final_path <- file.path("data", "final")
data_raw_path <- file.path("data", "raw")
data_supporting_path <- file.path("data", "supporting")
output_table_path <- file.path("output", "table")
```

```
print_acceptance <- function(data_before, data_after,
                             starting_data, step = 1){

  accepted_before <- dim(data_before)[1]
  accepted_after <- dim(data_after)[1]
  dim_starting_data <- dim(starting_data)[1]

  cat(" Cleaning data - Step :", step, "\n",
      "********************************** \n",
      accepted_after, "accepted time series over", accepted_before,
      "\n Current step acceptance rate :",
      round(accepted_after/accepted_before*100,2), "%",
      "\n Loss from step 1:",
      round((dim_starting_data-accepted_after)/dim_starting_data*100, 2), "%")
}
```

Raw data organization

The data used in this project were collected by **Sigmund Hov Moen**, and are available in the **Rimfrost system**, www.rimfrost.no.

They consist of a large amount of monthly and annual temperature time series from all around the world.

The raw data, as organized in the Rimfrost system, are collected in the folder "data/raw".

The "data/raw" folder contains 101 subfolders named with the English or the Norwegian name of the countries included in the Rimfrost system.

Each country folder contains the temperature time series for each weather station included in the Rimfrost system.

Data structure

Each time series is collected in a separate txt file, usually named with the Norwegian name of the weather station.

Each file is structured as follow:

- Column 1: **Year**
- Columns 2-13: **Monthly temperatures** in that year, from January to December
- Column 14: **Average annual temperature**, measured as mean of the monthly temperarures for that year

There are no column names, and the missing data are usually recorded with the string *99* (but several exceptions are present).

As an example, these are the first six rows of the *Paris.txt* file.

```
Paris_path <- file.path(data_raw_path, "frankrike", "paris.txt")
Paris_data <- read.delim(Paris_path, header=F, na.strings=99)

kable(head(Paris_data))
```

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
1757	-0.33	3.73	6.03	11.23	14.53	19.03	24.63	19.63	16.23	8.23	9.03	0.43	11.0
1758	1.63	3.93	7.93	10.03	18.53	20.33	17.93	20.83	15.43	9.83	5.63	3.13	11.3
1759	4.53	6.03	7.33	11.73	15.33	19.43	23.93	20.43	17.73	12.63	3.73	3.13	12.2
1760	0.23	3.63	6.63	12.33	16.03	20.03	21.63	19.73	18.73	11.83	7.53	1.43	11.6
1761	1.83	6.33	8.73	10.83	16.33	19.53	21.43	21.93	18.33	10.43	5.63	6.43	12.3
1762	4.93	4.03	4.03	13.83	17.33	19.93	23.33	19.63	16.83	9.83	5.83	2.63	11.8

Main features of the raw data

In order to select the set of suitable time series and explore their main features, we first provide a preliminary table with useful information (*T0_Information*). This table contains general information about all the available time series, namely:

- **Country** and **Station** name for each time series
- Its **Status**. The Status variable provides the results of some validation checks about the data format. "OK" means that the time series passed the checks, "ERROR" means that it did not pass the checks (eg the file has a wrong number of columns)
- **From** and **To** show the first and the last year of the recorded time series
- **Length** shows the length in years of the time series
- **Missing** shows the number of missing annual average temperature. NA means *Not Available*

```
#--- STEP 1 - TIME SERIES INFORMATION MATRIX

# ACCESS TO THE FOLDERS
country <- dir(data_raw_path)
ncc <- length(country)

# LIST OF INFORMATION
cc <- 1
j <- 1
information <- list()

for (cc in 1:ncc){

  country_path <- file.path(data_raw_path, country[cc])
```

```
  station0 <- dir(country_path)
  station <- station0[grep(".txt",station0)]
  stationame <- sub(".txt", "", station)
  nss <- length(station)

  for (ss in 1:nss){

  station_path <- file.path(country_path, station[ss])
  data00 <- read.delim(station_path, header=F,
                       na.strings=99)
  n <- dim(data00)[[1]]
  if (is.numeric(data00[,1]) == FALSE | dim(data00)[[2]] != 14) {
    information[[j]] <- paste0(country[cc], ";",
                               stationame[ss],
                               ";ERROR;NA;NA;NA;NA")
  } else {
    nmiss <- sum(is.na(data00[,14]))
    information[[j]] <- paste0(country[cc],";",
                               stationame[ss],";",
                               "OK",";",
                               data00[1,1],";",
                               data00[n,1],";",
                               n,";",
                               nmiss)}
  j <- j+1
  }
}
# Ignore the warnings!

# CONVERSION TO A MATRIX
n_info <- length(information)
info <- NULL
for (i in 1:n_info){
  station_info <- unlist(strsplit(information[[i]], split=";"))
  info <- rbind(info, station_info)}
T0_Information <- as.data.frame(info)
colnames(T0_Information) <- c("Country","Station","Status","From",
                              "To","Length","Missing")


write.csv(T0_Information, file.path(output_table_path, "T0_Information.csv"),
          row.names = F)
```

As an example, consider the first 6 rows of the *T0_Information* table:

```
T0_Information <- read.csv(file.path(output_table_path, "T0_Information.csv"))

kable(head(T0_Information, 6))
```

Country	Station	Status	From	To	Length	Missing
afganistan	herat	OK	1963	1990	28	16
afganistan	kabul	OK	1961	1992	32	12
afganistan	mazar_i_sharif	OK	1964	1992	29	11
algerie	adrar	OK	1965	2011	47	4

Country	Station	Status	From	To	Length	Missing
algerie	alger_dar_elbeida	OK	1923	2011	89	3
algerie	beni_abbes	OK	1931	2011	81	25

Raw data basic exploration

The total number of availbale time series is **1260**.

The table below provides a summary of the information given about each recorded variable:

```
s0 <- T0_Information %>%
  select(-Country, -Station) %>%
  summary()

kable(s0)
```

Status	From	To	Length	Missing
ERROR: 225	Min. :1701	Min. :1869	Min. : 5.00	Min. : 0.00
OK :1035	1st Qu.:1890	1st Qu.:2009	1st Qu.: 61.00	1st Qu.: 1.00
NA	Median :1925	Median :2011	Median : 84.00	Median : 2.00
NA	Mean :1917	Mean :2008	Mean : 92.28	Mean : 5.57
NA	3rd Qu.:1949	3rd Qu.:2012	3rd Qu.:122.00	3rd Qu.: 6.00
NA	Max. :2002	Max. :2012	Max. :312.00	Max. :104.00
NA	NA's :225	NA's :228	NA's :225	NA's :225

The following graph shows all the available time series, sorted by first recorded year. The green dot represents the first recorded year, while the purple dot represents the last recorded year. The grey segment represents the length of the time series.

```
T0_Information %>%
  arrange(desc(From), To) %>%
  ggplot() +
  geom_segment(aes(x = From, y = 1:1260, xend = To, yend = 1:1260),
               size = 0.03, color = "grey") +
  geom_point(aes(x = From, y = 1:1260), size = 0.1,
             color = "darkgreen") +
  geom_point(aes(x = To, y = 1:1260), size = 0.1,
             color = "purple") +
  theme_minimal() +
  theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
```

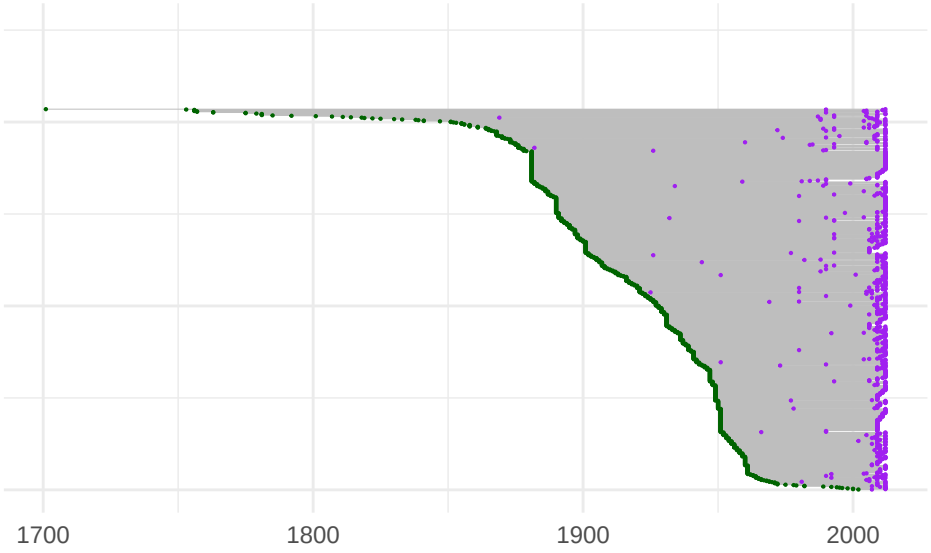


Figure 1: Time series plot

SELECTION PROCESS

Given the wide variety of properties of the available time series, we decided to apply a multi-step automatic selection procedure to obtain a subset of series with specific characteristics.

In any case, the full set of available time series (1260) is contained in the data/raw folder, and can be analyzed with the provided code.

Multi-step automatic process

Briefly, the multi-step process was designed as follow:

1. **Step 1 - Valid structure**: selection of the time series with valid data structure
2. **Step 2 - Length in years**: selection of the time series with more than 105 recorded years
3. **Step 3 - Recorded months**: selection of the time series with more than 1280 recorded months (non missing)
4. **Step 4 - Missing months**: selection of the time series with less than 80 missing months
5. **Step 5 - Length in months**: selection of the time series with full monthly length not inferior to 1290 months

Step 1 - Selection by valid structure

Using the information table previusly built (*T0_Information*), we selected all the time series with the variable *Status* equal to *OK*.

Here the results of the selection procedure for valid structure (step 1):

```
global <- dim(T0_Information)[1]

T0.1_Valid <- T0_Information %>%
  filter(Status=="OK")
```

```
print_acceptance(T0_Information, T0.1_Valid, T0_Information, step = 1)
```

```
##  Cleaning data - Step : 1
##  ***********************************
##  1035 accepted time series over 1260
##  Current step acceptance rate : 82.14 %
##  Loss from step 1: 17.86 %
```

Step 2 - Selection by length (years)

The second step of our selection procedure was to identify all the time series that had more than 106 recorded years.

The results of the second step of the selection procedure follows here:

```
T0.2_Ylength <- T0.1_Valid %>%
  filter(Length > 106)

print_acceptance(T0.1_Valid, T0.2_Ylength, T0_Information, step = 2)
```

```
##  Cleaning data - Step : 2
##  ***********************************
##  376 accepted time series over 1035
##  Current step acceptance rate : 36.33 %
##  Loss from step 1: 70.16 %
```

At this point, our subset consists of 376 weather stations.

Information table 2 - Monthly information

In order to apply step 3-5 of the selection procedure, we need to gather information about the data at a monthly level.

We then built a second information table, with the following fields:

- **Country** and **Station** name for each time series
- **First_Year** and **First_Year**, first and the last year of the recorded time series
- **Years**, length in years of the time series
- **Months**, number of available months (non missing)
- **Full_Length**, number of months between the first recorded month and the last one.
- **Missing_M**, number of missing months

```
station_path <- file.path(data_raw_path, T0.2_Ylength$Country,
                          T0.2_Ylength$Station) %>%
  paste0(".txt")

stationame <- paste(T0.2_Ylength$Country, T0.2_Ylength$Station,
                    sep = ", ")

n_step2 <- dim(T0.2_Ylength)[[1]]

all_info <- NULL
```

```
for (j in 1:n_step2){
  data <- read.delim(station_path[j], header=F, na.strings=99)
  dataM <- try(monthlyAdj(data, scale = T))

  if(is.data.frame(dataM)) {

    station_info <- dataM %>%
      summarize(Stationame = stationame[j],
                Country = T0.2_Ylength$Country[j],
                Station = T0.2_Ylength$Station[j],
                First_Year = min(year(Time)),
                Last_Year  = max(year(Time)),
                Years = Last_Year-First_Year+1,
                Months = n(),
                Full_Lenght = 12*Years + month(max(Time)) - month(min(Time)),
                Missing_M = Full_Lenght - Months # Check it
      )

  } else {

    station_info <- t(c(stationame[j], rep(NA, 8)))
    colnames(station_info) <- c("Stationame", "Country", "Station",
                                "First_Year",
                                "Last_Year", "Years", "Months", "Full_Lenght",
                                "Missing_M")
  }

  # Create the information data.frame
  all_info <- rbind(all_info, station_info)

}

T02_Information2 <- all_info %>%
  mutate_at(.funs = funs(as.numeric(.)),
            .vars =vars(First_Year:Missing_M))

write.csv(T02_Information2, file.path(output_table_path,
                                      "T02_Information2.csv"),
          row.names = F)
```

As an example, consider the the first 10 rows of the *T02_Information2* table:

```
T02_Information2 <- read.csv(file.path(output_table_path,
                                       "T02_Information2.csv")) %>%
  select(-Stationame)

kable(head(T02_Information2))
```

Country	Station	First_Year	Last_Year	Years	Months	Full_Lenght	Missing_M
alpene	geneve_ecad	1901	2009	109	1303	1314	11
alpene	graz	1894	2009	116	1387	1398	11
alpene	hohenpeissenberg	1781	2009	229	2741	2754	13
alpene	innsbruck	1877	2009	133	1490	1597	107
alpene	klagenfurt	1881	2009	129	1513	1554	41

Country	Station	First_Year	Last_Year	Years	Months	Full_Lenght	Missing_M
alpene	kremsmunster	1876	2009	134	1602	1614	12

Step 3 - Selection by available months

We can now refine the selection procedure checking that the number of *non missing months* is superior to *1260*.

Here the results of this selection step:

```
T0.3_Mlength <- T02_Information2 %>%
  mutate(NM_Months = as.numeric(Months)) %>%
  filter(Months >= 1260)

# Acceptance rate
print_acceptance(T0.2_Ylength, T0.3_Mlength, T0_Information, step = 3)
```

```
##  Cleaning data - Step : 3
##  ************************************
##  329 accepted time series over 376
##  Current step acceptance rate : 87.5 %
##  Loss from step 1: 73.89 %
```

Step 4 - Selection by missing months

We can now exclude all the time series with number of *missing months* above *80*.

Recall that the previous step was about non missing months: in this step we are avoiding time series that (although long), contain so many "holes" that they might compromise the data quality.

Here the results of this selection step:

```
T0.4_Missing <- T0.3_Mlength %>%
  filter(Missing_M < 80)

print_acceptance(T0.3_Mlength, T0.4_Missing, T0_Information, step = 3)
```

```
##  Cleaning data - Step : 3
##  ************************************
##  278 accepted time series over 329
##  Current step acceptance rate : 84.5 %
##  Loss from step 1: 77.94 %
```

Step 5 - Length in months

The last step of the selection procedure is to check if the total length of the time series (from the first recorded month to the last one, missing included) is not inferior to 1290.

Above the results:

```
T0.5_LengthM <- T0.4_Missing %>%
  filter(Full_Lenght >= 1290)

# Acceptance rate
print_acceptance(T0.4_Missing, T0.5_LengthM, T0_Information, step = 5)
```

```
##  Cleaning data - Step : 5
##  ***********************************
##  277 accepted time series over 278
##  Current step acceptance rate : 99.64 %
##  Loss from step 1: 78.02 %
```

At this point we have a subset of 277 waether stations, characterized by a valid data structure and a length and presence of missing observation below specific thresholds.

Final selection

In order to select the final set of weather stations we proceded by manual inspection of all the time series.

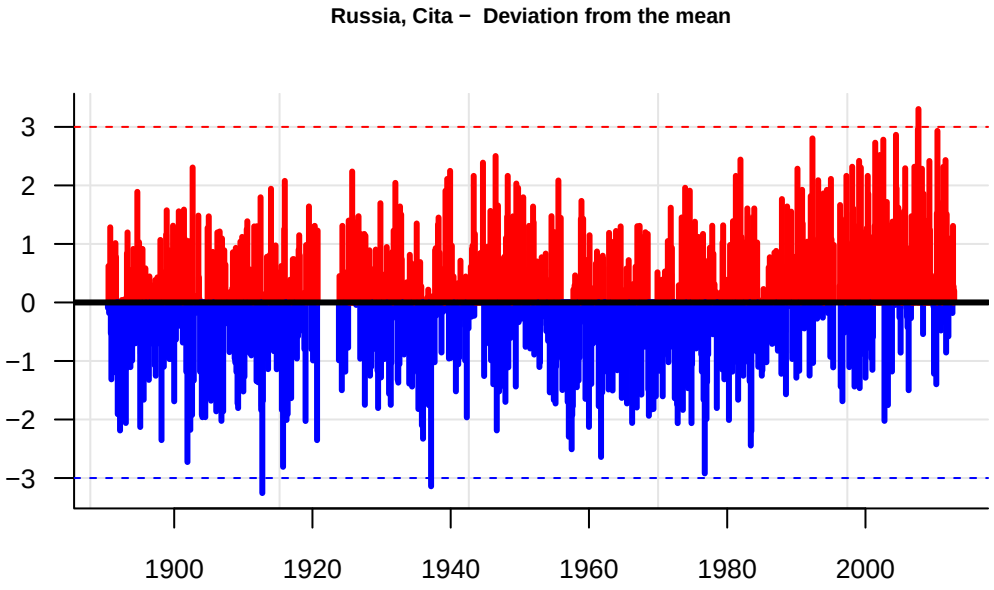
We excluded all the time series with substantial quality problems, such as those with extreme outliers and several consecutive missing over time intervals.

In the final selection we tried as much as possible to select weather stations that were distributed across most parts of the world.

Consecutive missing data example - Cita

```
cita_data <- file.path(data_raw_path, "russland", "cita.txt")

data <- read.delim(cita_data, header=F, na.strings=99)
dataM <- monthlyAdj(data, scale = T)
  blueRedPlot(Zj = dataM$Zm, Time = dataM$Time,
              main = "Russia, Cita - ", cex.main=0.7)
```

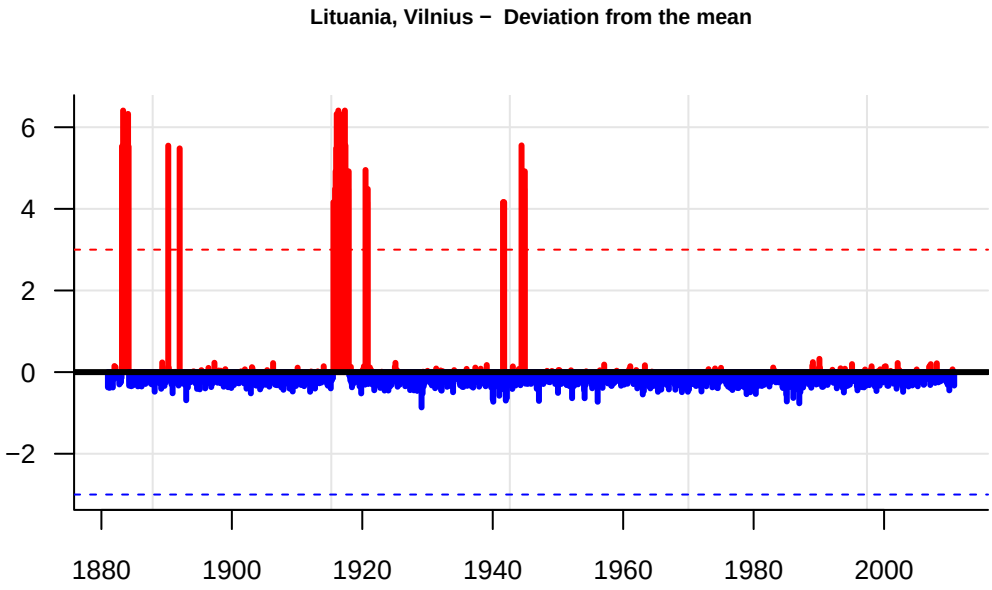


Outliers example - Vilnius

```
vilnius_data <- file.path(data_raw_path, "litauen", "villnius.txt")

data <- read.delim(vilnius_data, header=F, na.strings=99)
```

```
dataM <- monthlyAdj(data, scale = T)
  blueRedPlot(Zj = dataM$Zm, Time = dataM$Time,
              main = "Lituania, Vilnius - ", cex.main=0.7)
```



Final data

Summary statistics of the selected of 96 time series is shown in Appendix B, table B1.

There are two time series that did not pass the tests but we still included them in our analysis. One is from Ivittuut, Greenland, and the second one is from Uppsala, Sweden. The reason is that Greenland is of particular interest in the climate debate, and the temperature series from Uppsala is the longest time series ever recorded.

All the selected time series are available in the data/final folder. All the names (country and stations) have been translated to English.

We have highlighted results from 9 weather stations because they are well known cities with good quality of the temperature data.