Assignment 2

Group 4

2024-09-16

```
options(contrasts = c("contr.sum", "contr.poly"))
require("ggplot2")
require("dplyr")
require("ppcor")
require("caret")
require("tidyr")
require("stringr")
```

Task 1: Dimension reduction on air quality data

Part A: Get

- Obtain data from https://archive.ics.uci.edu/dataset/360/air+quality.
- Provide a brief description of the data based on the information from the website.

Hints

- See options of `read.table()` for correct import

Part B: Import and Visualize

- Load the data and convert to tsibble.
  - Make sure dates and hours are converted into proper time objects
  - Remove incomplete days at beginning and end of data
- Plot the data as is, preferably as multiple panels in a single plot
- Describe the data. What is most striking?

Part C: PCA of data as is

- Perform PCA on the data as prepared in B
- Create a screeplot and create biplots for 1st and 2nd and for 2nd and 3rd PCs
- Plot the scores for the PCs
- Comment on the results. Can you relate some features to your observations in part B?

Hints

- `ggfortify` provides `autoplot()` for PCA results for ggplot-style biplots
- To plot the scores, you can use the same code as for plotting the original data

Part D: Missing values

- Identify missing values in the time series
- Investigate to which degree missing values occur at the same time for multiple sensors

- Is one or are multiple sensors behaving peculiarly? How would you handle this?
- Discuss options for handling missing values: (a) drop all time points containing any missing value, (b) impute values for missing values. In case of (b) choose a method for imputation. Justify your decisions.
- At the end of this step, you should have a version of the data containing only valid values. Plot these data as in Part B.

Hints

- Remember `imputeTS`
- You can apply its functions to an entire dataframe, will be done column-wise

Part E: PCA of cleaned data

- Perform PCA on the data as prepared in D
- Create a screeplot and biplots for 1st/2nd, 2nd/3rd, 3rd/4th PC
- Compute total variance explained by 1st, 1st and 2nd, 1st to 3rd, ... PCs
- Choose how many PCs to keep and transform data back to original sample space
- Plot the result against the cleaned data, compare and discuss
- Also plot the scores, zoom in to short time intervals and look at periodicity
- Can you interpret certain PCs?

Task 2: STL and correlation on weather data

Part A: Data collection for a single station

Based on material from the lectures, write an R function that can obtain a daily average temperature series for a meteorological station from the Norwegian Met Institute's Frost service. The function shall return a tsibble.

Part B: Data preparation for a single station

- Identify gaps in the time series.
- Assume that gaps up to 31 days are acceptable. Find the earliest date in the time series such that all following data have no gaps longer than 31 days. Limit the time series to this.
- Create a regular time series by filling gaps in the tsibble with n/a-s.
- Impute values for the n/a-s. Justify your choice of imputation method.
- You should now have a regular time series with only numeric values.
- Remove all data for 29 February so all years have data for exactly 365 days.
- Combine all this code into a function for re-use later. The function should receive the original tsibble from part A as input and return a new tsibble.

Hints

- tidyverse provides functions such as has_gaps() and count_gaps()

Part C: Exploratory analysis for a single station

- Plot the temperature data as function of time
- Create density plots of original data and data with imputed values
- Turn the temperature data into a timeseries (ts) object
- Plot the autocorrelation function for lags up to 5.5 years; describe and discuss your observations
- Also plot the ACF only for short lags, up to four weeks
- Select some days distributed throughout the year and plot temperature as function of year for, e.g., 1 October, as a scatter plot. This plot can be useful to choose the seasonality window later (see Figs 7 and 8 in Cleveland et al, 1990)

Part D: STL analysis

- Perform STL on the data. Explore different values for the seasonality and trend windows (remember that we want to look at trends over many years!), the choice between robust STL or not, and possibly the lowpass filter window. Describe your observations. It might be interesting to look at the ACF of the remainder in the STL result.
- Consult the original STL paper by Cleveland et al. (1990) for suggestions on how to choose STL parameters.
- Based on your analysis, can you suggest a set of STL parameters to use for further work?

Part E: Multiple station analysis

- Obtain data from eight more stations. Two should be in the same part of Norway as the station from part A; then choose three stations each from two other parts of Norway. Data should cover several decades at least, so look for stations with long series.
- Preprocess the data as described in Part B. Find the latest starting date of any series and create a multivariate time series with data from all nine stations starting at this date.
- Obtain the cross-correlation matrix between the nine stations. Is there any structure in this 9x9 matrix?
- Perform STL individually on each of the nine stations using the parameters from part D. Compare the resulting trends. Are all STL results of equal quality?

Hints

You can get a list of all available stations from Frost using

```
#.stations_url = str_glue("https://{.client_id}@frost.met.no/sources/v0.jsonld")
#raw_stations <- fromJSON(URLencode(.stations_url), flatten=TRUE)
```

To limit this to stations with actual data, starting at least as early as 1950, coming from only some parts of Norway relevant columns, and limiting to relevant columns, filter the raw data as

```
# COUNTYS = c(fylke1, fylke2, etc)   # replace with names of "fylker" you are interested in
#
# stations <- unnest(raw_stations$data, cols='id') |>
#   select(id, validFrom, country, county, municipality, name, masl, `@type`) |>
#   mutate(validFrom=as.Date(validFrom)) |>
#   filter(`@type` == "SensorSystem" & validFrom <= "1950-01-01" & country == "Norge" & county %in% COU
```

Part F (bonus): PCA

- Perform PCA on the multivariate time series.