This compulsory assignment consists of 4 exercises. Exercises must be solved in groups (assigned in Canvas) and uploaded to Canvas by the submission deadline. Please provide a single `.html` or `.pdf`-file, entitled `dat320_comp1_groupX.html` (or `.pdf`), where `X` should be replaced by your group number (1-16). The file should be structured into sections (one section per exercise) and subsections (one subsection per task).

Exercise 1 (Data processing with R)

The *traffic_volume* volume dataset contains the hourly traffic volume of Interstate 94 between Minneapolis and St Paul, MN, USA, between 2012 and 2018. In addition to the hourly traffic volume, environmental aspects like rain, snow, clouds and temperature are also included.

Material provided:

- `traffic_volume.csv`: dataset

Tasks:

(a) *Load in the dataset from the file traffic_volume.csv and select the columns* `date_time` *and* `traffic_volume`*. Convert the* `date_time` *column to POSIXCT format (use* `lubridate`*). Further, remove all observations from the incomplete calendar years 2012 and 2018 and print the traffic volume in the first and the last year. Remove the 29th of February in leap years and verify that the number of rows for each calendar year is 365. Finally, plot the data. Hint: use only shorter intervals for plotting.*

(b) *Create four data frames by aggregating the data at the following levels:*

- *daily,*
- *weekly,*
- *monthly,*
- *yearly.*

At each level, compute the mean, std. deviations, min and max of the traffic volume. Plot the aggregated data along with the std. deviations and min/max traffic volume in line plots.

(c) *Perform STL decompositions of the original hourly data, as well as the aggregated daily, weekly and monthly data computed in (b). Yearly is not of interest because it only contains 5 data points. Plot the decompositions at different frequencies (you may restrict the data to a sub-interval for better visibility). Hint: Look at decompositions of both shorter time intervals and full time intervals.*

(d) *Comment shortly on the following questions:*

- *What is the interpretation of the seasonal components in the data?*
- *Does it make sense to aggregate by mean? Would median be more reasonable?*
- *A common paradigm is that "more data is better". Does that hold if you aim to investigate monthly trends?*
- *Does simple STL decomposition make sense in the case of multiple seasons?*

Exercise 2 (Missing values)

For this exercise, we use the dataset *romanian_energy* that contains information about power consumption in Romania, measured as hourly samples between 2019 and 2023. The dataset contains three columns where the first represents the date (`DateTime`), the second contains the Consumption (`Consumption.Full`), and the third is the consumption with missing values (`Consumption.Missing`). You will work with the column `Consumption.Missing` and investigate options for missing value imputation. The `Consumption.Full` time series represents the ground truth and will be used to evaluate imputations.

Material provided:

- `romanian_energy.csv`: dataset

Tasks:

(a) *Load the dataset romanian_energy.csv and convert the DateTime column to the appropriate time series object. Plot the time series* `Consumption.Missing`*. Hint: look at various time interval sizes and use the function ggplot_na_distribution() to visualize NA's. Further, plot the ACF and comment on the seasonal components. Use different magnitudes of the hyperparameter* `lag.max`*.*

(b) *Replace the missing values with each of the following metrics:*

(i) global mean,

(ii) last observation carried forward (LOCF)

(iii) Seasonally Decomposed Missing Value Imputation (SeaDec)

Plot all options of the imputed time series and describe the differences. Compute the MSE between your fit and the ground truth, which is given in column `Consumption.Full`*. Do you see any problems with the imputations?*

(c) *Use the function* `msts` *to convert the time series* `Consumption.Full` *into a multiple seasonal time series object, and decompose the time series using the function* `mstl`*. Take into account all reasonable seasonal periods detected in (a). Next, create another decomposition but this time decompose the* `Consumption.Missing`*. R will automatically impute nonsense in the* `Consumption.Missing` *components, so remove the* `NA` *rows on each component that corresponds to the* `NA` *rows in* `Consumption.Missing`*. Plot both decompositions.*

(d) *Impute each component separately. For this purpose, choose an optimal imputation method for each component based on comparing the MSE of the complete component from* `Consumption.Full` *and the corresponding imputed component from* `Consumption.Missing`*. After imputing, transform the components back to the original data. Compute the MSE of the full time series and compare with the results in (b).*

Exercise 3 (Transformations)

The folder *surface_temp* contains data about the daily mean 2-meter air temperature from the NCEP Climate Forecast between 1979 to 2023. The data is given in JSON formats. For reference, all the JSON files have the same format and structure. The goal of this exercise is to explore a different data format and utilizing data transformations.

Material provided:

- `surface_temp`: folder
  - `world.json`: dataset
  - `tropics.json`: dataset
  - `sh.json`: dataset
  - `nh.json`: dataset
  - `arctic.json`: dataset
  - `antarctic.json`: dataset

Tasks:

(a) *Create a framework for loading and merging the datasets into one joint data frame. The following preprocessing steps should be implemented:*

i. Create a function import_json(file) that takes an argument `file` *containing the filename and returns a dataframe. Hint: Use* `fromJSON()` *from the package* `jsonlite`*. Investigate the head and tail of the data. Is there anything that can be removed?*

ii. Create another function `preprocess_one(df)` *that takes in the loaded dataframe. Here you want to:*

- *Slice the data to only keep rows between 1979 and 2022.*
- *Remove every 366th observation (leap year is not of interest).*
- *Create a column "date" in yyyy-mm-dd format.*

iii. Finally, create a wrapper-function merge_dataset() that sequentially loads and preprocesses each file and returns one dataframe with columns "date" and one column for each region.

(b) *Visualize the data for each region individually. The x-axis should be a date-of-the-year representation of the time index, the y-axis should denote the temperature of the region. Color by the year. Find fitting sizes and alpha values of the lines/points.*

(c) *Prepare a correlation plot and a PCA to identify the relationship between the regions. Visualize the results from the PCA by plotting explained variances, scores, and loadings.*

(d) *Comment on the relationship between the regions. Use the visualization from each of the tasks to explain the following:*

- *How do temperature patterns in the regions differ for each other?*
- *How can the relationships be identified in the given plots?*

Exercise 4 (ACF and decompositions)

In this exercise, the procedure of a simplified decomposition into trend and seasonality will be implemented and analyzed. This is a simplification of the Loess (local regression) procedure used in STL.

Material provided:

- `austres` dataset from package `datasets`
- `co2` dataset from package `datasets`
- `nottem` dataset from package `datasets`

Tasks:

(a) *Implement the procedure described in the following to estimate seasonality and trend components: Assume a time series $(x_t)_{t\in T}$, a seasonal period of length p, and an odd-numbered integer parameter `s.window` > 0. For a given time index t, consider the subset of indices*

$$W_t = \{(t - w \cdot p), (t - (w-1) \cdot p), \ldots, t, \ldots, (t + (w-1) \cdot p), (t + w \cdot p)\} \cap T,$$

where $w = \frac{\texttt{s.window}-1}{2}$. Compute the seasonal component as $\hat{s}_t = mean(\{x_i : i \in W_t\})$. As a second step, assume an odd-numbered integer parameter `t.window` > 0. Subtract the estimated seasonal component $\hat{s}_t$ from x_t to obtain y_t, i.e. $y_t = x_t - \hat{s}_t$ for each t. Then, use the subset of indices

$$V_t = \{(t - v), (t - (v-1)), \ldots, t, \ldots, (t + (v-1)), (t + v)\} \cap T,$$

where $v = \frac{\texttt{t.window}-1}{2}$. Compute the trend component as $\hat{\tau}_t = mean(\{y_i : i \in V_t\})$. The remainder r_t is given as $r_t = y_t - \hat{\tau}_t$. Which range of values makes sense for the parameters `s.window` and `t.window`?

(b) *Load the datasets* `austres` *(quarterly number of residents in Australia),* `co2` *(monthly measurements of atmospheric CO_2 concentrations) and* `nottem` *(monthly averaged air temperatures) from the R package* `datasets`*. For each dataset, decompose the time series using your implementation in (a). Try to find good values for* `s.window` *and* `t.window`*. Which quantitative metric could be used to optimize the parameters? Plot the decomposition for the chosen parameter setting.*

(c) *After choosing optimal parameters for each dataset in (b), plot the autocorrelation function of the original dataset, as well as the autocorrelation of each component. Which component is most dominant in each dataset? How does the shape of the autocorrelation function reflect the type of the component (seasonality / trend / remainder)? What does the autocorrelations tell us about the quality of the decomposition?*

(d) *In general, which effect does each of the following transformations have on the decomposition, if applied a priori:*

- *smoothing*
- *differencing*
- *standardization*

Explain why! Which type of preprocessing could be useful by default for this procedure?