ST5209/X Assignment 2

Due 24 Feb, 11.59pm

Set up

1. Make sure you have the following installed on your system: $\LaTeX$, R4.2.2+, RStudio 2023.12+, and Quarto 1.3.450+.
2. Clone the course repo.
3. Create a separate folder in the root directory of the repo, label it with your name, e.g. `yanshuo-assignments`
4. Copy the assignment2.qmd file over to this directory.
5. Modify the duplicated document with your solutions, writing all R code as code chunks.
6. When running code, make sure your working directory is set to be the folder with your assignment .qmd file, e.g. `yanshuo-assignments`. This is to ensure that all file paths are valid.[1]

Submission

1. Render the document to get a .pdf printout.
2. Submit both the .qmd and .pdf files to Canvas.

Question 1 (Forecast evaluation, Q5.12 in FPP)

`tourism` contains quarterly visitor nights (in thousands) from 1998 to 2017 for 76 regions of Australia.

a. Extract data from the Gold Coast region using `filter()` and aggregate total overnight trips (sum over `Purpose`) using `summarise()`. Call this new dataset `gc_tourism`.

b. Using `slice()` or `filter()`, create three training sets for this data excluding the last 1, 2 and 3 years. For example, `gc_train_1 <- gc_tourism |> slice(1:(n()-4))`.

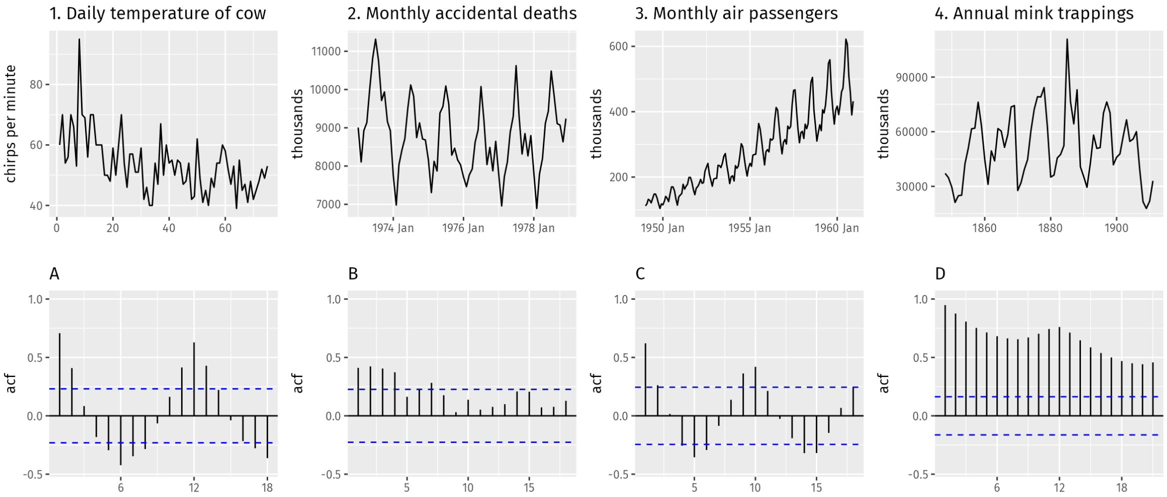
[1] You may view and set the working directory using `getwd()` and `setwd()`.

c. Compute and plot one year of forecasts for each training set using the seasonal naive (`SNAIVE()`) method. Call these `gc_fc_1`, `gc_fc_2` and `gc_fc_3`, respectively. (Hint: You may combine the mable objects from each model into a single mable using `bind_cols`)

d. Use `accuracy()` to compare the test set forecast accuracy using MASE. What can you conclude about the relative performance of the different models? Explain your answer.

Question 2 (ACF plots, Q2.9 in FPP)

The following time plots and ACF plots correspond to four different time series. Match each time plot in the first row with one of the ACF plots in the second row.



Question 3 (Time series classification)

We will investigate this dataset, which contains 600 synthetic control charts for an industrial process. There are 6 types of control charts, and our goal is to build a model to classify them correctly.

We have already processed the raw data into a convenient, labeled form. Run the following code snippet to load it and create train and test sets.

```
ccharts <- read_rds("../_data/CLEANED/ccharts.rds")
ccharts_train <- ccharts[["train"]]
ccharts_test <- ccharts[["test"]]
```

a. Make a time plot of one time series from each category in the training set. Note that the category is recorded under the `Type` column.

b. Compute all time series features for both the training and test set using the snippet. What is the difference between `acf1` and `stl_e_acf1` for Increasing, Decreasing, Upward, and Downward types? Why is there a difference?

```r
# Change to eval: TRUE in order to run
train_feats <- ccharts_train |> features(value, feature_set(pkgs = "feasts"))
test_feats <- ccharts_test |> features(value, feature_set(pkgs = "feasts"))
```

c. Investigate the relationship between the following features and `Type`. Pick two features whose scatter plot gives a good separation between all 6 chart types.

    i. `linearity`

    ii. `trend_strength`

    iii. `acf1`

    iv. `stl_e_acf1`

    v. `shift_level_max`

    vi. `shift_var_max`

    vii. `n_crossing_points`

    Make the scatter plot and explain why these features are able to separate the different chart types.

d. Install the `caret` package and use the following snippet to fit a k-nearest neighbors model on the two features you have selected (substitute `X` and `Y` for the two features you selected in b), and then predict on the test set. What percentage of the test examples are correctly classified? Write code to compute this value. (You should get > 90% accuracy)

```r
# Change to eval: TRUE in order to run
library(caret)
knn_fit <- train(Type ~ X + Y, data = train_feats, method = "knn")
predict(knn_fit, newdata = test_feats)
```

Question 4 (Periodograms)

Consider the `vic_elec` dataset from the `tsibbledata` package.

a. Compute and plot the periodogram for `Demand`. What frequencies and periods do the 7 highest peaks correspond to? *Hint: You may use the* `periodogram` *function provided in* `plot_util.py` *and change the* `max_freq` *setting to see at different resolutions.*

b. Perform an STL decomposition for `Demand`. Plot the periodogram for `season_year`, `season_week`, and `season_day`. Which of the original periodogram peaks appear in each of these periodograms?

c. Based on the peak heights, which is the strongest seasonal component? Does it agree with what you see in a time series decomposition plot?