TSAR Project Assignment Part 3 (10 points)

Times Series Forecasting

gwendolin.wilke@fhnw.ch

Prelimiaries

- Go to Moodle, download `series_IDs.pdf` and find your `Series_ID` in the list.
- Go to R Studio, load the library `fpp3` and briefly inspect the tsibble `us_employment`. It contains 148 monthly time series of US employment data from January 1939 to June 2019. Each `Series_ID` represents a different job sector.
- Filter `us_employment` for your `Series_ID`. Inspect the result using `View()` and `autoplot()`.
- If you find that the filtered series has leading NAs, remove them.
- Use `saveRDS()` to save the resulting time series on disk. Use it to solve the following task.

Task (10 points)

- Create a new Quarto document for Project 3. Call it "`P3-<your TSAR-ID>-<your last name>.qmd`". (E.g., "P3-12-Burri.qmd").
- In your new Quarto document, load your time series (the one you just saved to disk) and store it under the variable name `mySeries`.
- Now train 2 different benchmark models on your series. To do that, you can choose 2 different of the following 5 methods:

  1. the mean method (see Part 2, Lecture 3),
  2. the naive method (see Part 2, Lecture 3),
  3. the seasonal naive method (see Part 2, Lecture 3),
  4. the drift method (see Part 2, Lecture 3),
  5. the seasonal naive method with drift (see Part 2, Self-Study 3, Task 2).

- Evaluate both of your models and interpret all evaluation results.
- Compare the evaluation results of the two models and decide for the better model.
- **Note:** Please find step-by-step instructions for this task below.

Step-By-Step Instructions

1. **(0.5 points)** Extract the Training Set

   - Use `nrow()` to count the number of rows in your tsibble, and multiply it by 0.8.
   - Use `floor()` to round the resulting number to an integer value.
   - Now use the dplyr verb `slice()` to extract the first 80% of the rows of your series.
   - The resulting time series is your training set. Store it in the variable `train`.

     Important: Don't random sample the training data![1]

     Note: You don't need to extract the test set. All you need do is check how long it is, so that you can later make a forecast of the same length.

2. **(0.5 points)** Train your 1st Benchmark Model

   - Train your first benchmark model on the training data.
     To do that, use the function `model()`.
   - Within `model()`, specify the benchmark model you want to use.
   - Use `augment()` to get a first impression of the fitted values.

     Remark: Depending on the benchmark model you apply, the first few fitted values may be NAs. This is because the model needs a few observations to make the first forecast.

3. **(2 points)** Evaluate the Model Fit of Your 1st Benchmark Model[2]

   a. Create a time plot '*Actual vs. Fitted':*
      Plot the actual values and the fitted values of your training set in the same time plot. Use `augment()`, `autoplot()` and `autolayer()` for that. *Then answer the following questions in max. 1 sentence each:*

      - Are the fitted values close to the actual values?
      - Are they systematically too high or too low?
      - If yes, where does it come from? (Consider how your benchmark model works!)

[1] Random sampling is important when the data is not temporal: If non-temporal data happens to be pre-sorted, drawing the first 80% of the rows as training data would introduce data bias. Time series on the other hand are *always* pre-sorted (namely by time index). Time series methods explicitly use this as a feature. If you would random sample, you would distroy the temporal order.

[2] In this step you evaluate the **model fit**. That is, you evaluate how well your model captures the patterns present in the **training data**. Different diagnostic plots and statistical tests are commonly used for that. **Note!** All diagnostic plots and tests for model fit will come out very badly in this case. This is because the patterns of your time series are too complex for a simple benchmark model to capture them. **Your task** is to interpret the diagnostic plots and the Ljung-Box test only as far as we discussed them in the lecture. There may be more things you could read from them, but you are not expected to do that.

b. Create a scatter plot '*Actual vs. Fitted':*
Plot the actual values against the fitted values in a scatter plot. Use `ggplot()` and `geom_point()` for that. Add the identity line (the line with slope 1) using `geom_abline(slope = 1, intercept = 0, linetype=2)`. *Answer the following questions in max. 1 sentence each:*

- Are the points close to the identity line?
- Are they systematically too high or too low?
- If yes, where does it come frome? (Relate your answer to what you saw in the time plot.)

c. Perform residual diagnostics to inspect the model fit:
Feed your model into `gg_tsresiduals()` to produce residual plots. Visually inspect them. *Answer the following questions in max. 1 sentence each:*

- Are the residuals auto-correlated? How do you decide that based on the plots?
- Do the residuals have zero mean? How do you decide that based on the plots?

Hint: Check slide 27 of Part 2, lecture 3 ("essential properties of a good model"), and Tasks 3-6 of Part 2, Exercise 3 ("Residual Diagnostics").

d. Double-check your results from (c).

- Perform a Ljung-Box test of residual autocorrelation:
  Use the function `features()` and set the `lag` argument so that it fits your series. *Answer the following questions in 1 sentence:* Does the test result support your conclusions from (c)? How do you conclude that from the test result?

  Hint: The lag should be at least 2 times the seasonal period of your time series. Alternatively, you can use the same number of lags that you see in the ACF plot of `gg_tsresiduals()`*.*

- Calculate the residual mean:
  Apply the function `mean()` to the column `Employed` of `mySeries`. Does the result support your conclusions from (c)? *Answer in 1 sentence.*

4. **(1.5 points)** Evaluate the Point Forecast Accuracy of your 1st Benchmark Model[3]

- Make a forecast:
  Feed your trained model into the function `forecast()`. Set the argument `h` so that the forecast horizon has the same length as the test data. The test data is whatever remained from your time series after you extracted the training data.

[3]In this step you evaluate the **accuracy of the point forecasts**. That is, you evaluate how closely the point forecasts match the true values on the **test set**. Different accuracy metrics are commonly used for that.

- Create a time plot '*Actual vs. Forecast'*:
  Plot the actual values and the forecast values in the same time plot. To do this, simply feed your forecast into `autoplot()` and set the original series as a second argument, e.g., `fc_benchmark1 %>% autoplot(mySeries)`.
- Calculate the forecast accuracy metrics:
  Feed your forecast into `accuracy()` and set the original series as a second argument, e.g., `fc_benchmark1 %>% accuracy(mySeries)`. In the output, interpret the RMSE and the MAPE *in max. 3 sentences each.* Use the time plot '*Actual vs. Forecast*' to aid you in the interpretation. Examples:
  - *"The root mean squared error RMSE is scale dependent and must therefore be interpreted by a domain expert. Looking at the time plot, I see that it is approximately in the magnitude of the seasonal variations, which seems acceptable for a benchmark model. Yet, it also shows that the model is by far not accurate enough to predict within-season employment numbers ."*
  - *"The mean absolute percentage error MAPE compares each error with the corresponding level of the series. This is well suited for my series, since it has a strong trend and cycle (that is, the level is changing strongly, relative to the seasonality). 4% seems reasonable, since it means that the error is only 4% of the level on average."*

5. **(0.5 points)** Evaluate the Point Forecast Uncertainty of Your 1st Benchmark Model[4]
   - Perform residual diagnostics to check the prediction interval reliability:
     Use `gg_tsresiduals()` once more to produce the same residual plots you used in step 3. *Answer the following questions in max. 1 sentence each:*
     - Are the residuals homoscedastic? (That is, do they have constant variance?)
     - Are the residuals normally distributed?

     Hint: Check slide 28 of Part 2, lecture 3 ("properties 3 and 4 of a good model"), and look again at Tasks 3-6 of Part 2, Exercise 3 ("Residual Diagnostics").
   - Look again at the time plot '*Actual vs. Forecast'* from step 4 and pay attention to the 80% and 95% prediction intervals shown there. Apply the function `hilo()` to your forecast to display the prediction intervals' boundaries. Does it make sense to include these prediction intervals in your model evaluation? Why/why not? *Answer in 1 sentence.*

[4] In this step, you are evaluating the **uncertainty of the point forecasts**. That is, you are inspecting the prediction intervals of the point forecasts. A good forecasting model should not only provide accurate point forecasts but also small and reliable prediction intervals. **Note:** The point forecast *accuracy* tells you how well the model performs with the specific sample you are using. The point forecast *uncertainty* (the prediction intervals) provides you with an estimate of how well it performs on new samples on average.

6. Perform Steps 2-5 for your 2nd Benchmark Model
    - **(0.5 points)** Train your 2nd benchmark model on the training data.
    - **(2 points)** Evaluate its model fit.
    - **(1.5 points)** Evaluate its point forecast accuracy.
    - **(0.5 points)** Evaluate its point forecast uncertainty.
7. **(0.5 point)** Compare the Evaluation Results of the Two Models
    - Compare all evaluation results and decide for one model.
    - Argue why you decide for your model of choice. If you cannot decide, argue why. *Answer in max. 4 sentences.*

    Note:

    - *Comparing the evaluation of model fits will likely not help you much in your decision. The reason is that you can only determine **if** the model fits are good or bad. You don't have the tools to see **why** they are good or bad - we did not cover that in the lecture.*
    - *Comparing the prediction intervals of the models will not help you either: Due to the bad model fit, you cannot trust the prediction intervals, and thus you cannot use them for comparison.*
    - *Therefore, the main decision point in this case will be the comparison of point forecast accuracies of the two models.*