Lab 9: Forecast Accuracy

Luana Lima

03/24/2021

Objectives

1. Answer questions on M9
2. Answer questions on A6
3. Finish exercise from "08_Lab_Forecasting.Rmd" (repeated on this file)
4. Calculate accuracy metrics.

Setting R code chunk options

First R code chunk is used for setting the options for all R code chunks. The choice echo=TRUE means both code and output will appear on report, include = FALSE neither code nor output is printed.

Loading packages and initializing

Second R code chunk is for loading packages. By setting message = FALSE, the code will appear but not the output.

```
library(lubridate)
library(ggplot2)
library(forecast)
#library(Kendall)
library(tseries)
#library(outliers)
library(tidyverse)
library(smooth)

#New package for M9 to assist with tables
#install.packages("kableExtra")
library(kableExtra)
```

Understanding accuracy metrics

Compute metrics using the formulas below and then compare your results with accuracy().

$ME = \sum(Actual - Forecast)/n$ $MSE = \sum(Actual - Forecast)^2/n$ $RMSE = sqrt(MSE)$ $MAD = MAE = \sum|Actual - Forecast|/n$ $MPE = \sum((Actual - Forecast)/Actual)/n * 100$ $MAPE = \sum|Actual - Forecast|/Actual/n * 100$

```r
actual=c(107,125,115,118,108)
forecast=c(110,121,112,120,109)

#Exercise 1: compute MAD, MSE and MAPE
n=length(actual)

ME = (sum(actual-forecast))/n
MSE = (sum((actual-forecast)^2))/n
RMSE = sqrt(MSE)
MAE = (sum(abs(actual-forecast)))/n
MPE = (sum((actual-forecast)/actual))/n*100
MAPE = (sum(abs(actual-forecast)/actual))/n*100



#Exercise 2: Compute the forecast accuracy metrics you calculated in Ex 1 with the results from accurac
```

Importing data

Today we continue working with data from the Climate Change Knowledge Portal from the World Bank Group. More specifically historical rainfall and temperature averages for Brazil. You will find two new data files on folder "/Data/Raw/". One with rainfall named "pr_1901_2016_BRA.csv" and another with temperature named "tas_1901_2016_BRA.csv". The data span the period from 1901 to 2016 in monthly steps. You can download the data [here][https://climateknowledgeportal.worldbank.org/download-data]

Research question: Can you forecast temperature for the next two months?

```r
# Import both datasets using the read.csv function.
Rainfall_BR <- read.csv("../Data/pr_1901_2016_BRA.csv", stringsAsFactors = TRUE)

Temp_BR <- read.csv("../Data/tas_1901_2016_BRA.csv", stringsAsFactors = TRUE)

# Tidy the rainfall data sets.
#a Rename the column with Rainfall to get rid of the dots.
#b Note that on both data sets that is a column with the month name and average. Convert it to a Month
#c Now you should have a column with Month. Use the paste0() function to paste month and year together
#d Select only the columns of interest: Date and rainfall

Rainfall_BR_processed <-
  Rainfall_BR %>%
  rename( Rainfall = Rainfall....MM.) %>%
  separate(Statistics,c("Null","Month","Null2")," ") %>%
  mutate( Date = my(paste0(Month,"-",Year))) %>%
  select(Date,Rainfall)

#Repeat for the temperature dataset.
Temp_BR_processed <-
  Temp_BR %>%
  rename( Temperature_C = Temperature....Celsius.) %>%
  separate(Statistics,c("Null","Month","Null2")," ") %>%
  mutate( Date = my(paste0(Month,"-",Year))) %>%
  select(Date,Temperature_C)
```

```r
#Join the temperature and rainfall into one tidy data frame with 3 columns: Date, Rainfall and Temperat
BR_complete <- inner_join(Rainfall_BR_processed,Temp_BR_processed)
```

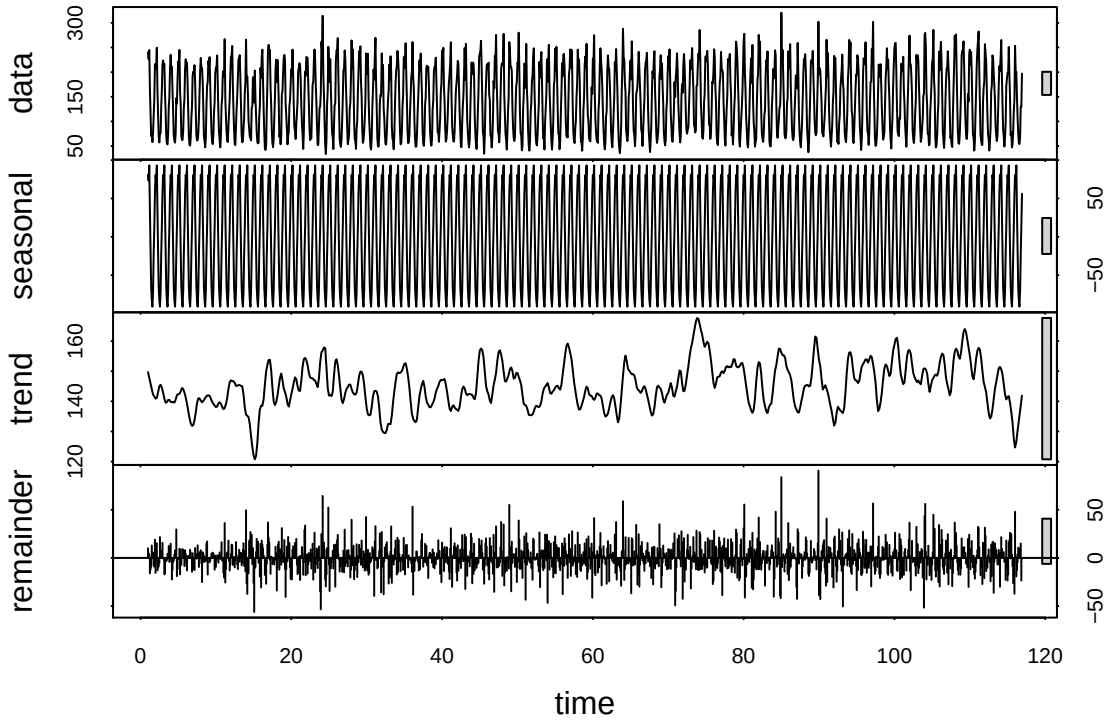
```
## Joining, by = "Date"
```

Transforming data into time series object

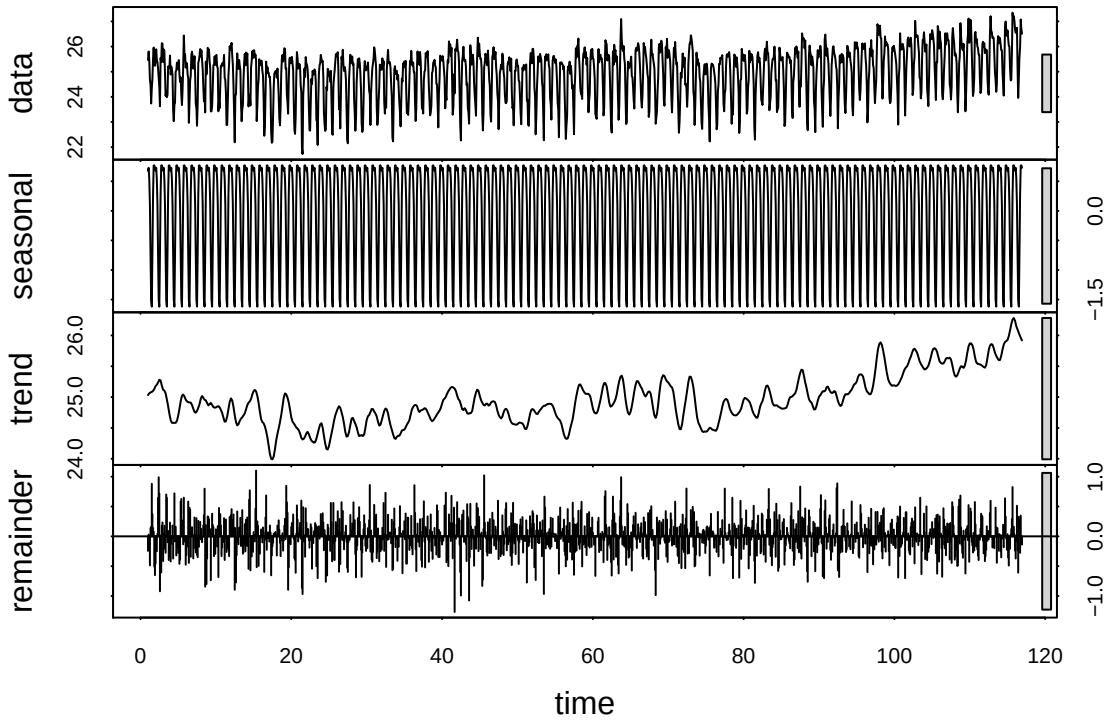
```r
f_month <- month(first(BR_complete$Date))
f_year <- year(first(BR_complete$Date))

BR_Rain_ts <- ts(BR_complete$Rainfall,frequency = 12)
BR_Temp_ts <- ts(BR_complete$Temperature_C,frequency = 12)
```

Decomposing the time series

```r
BR_Rain_Decomposed <- stl(BR_Rain_ts, s.window = "periodic")
plot(BR_Rain_Decomposed)
```



```
BR_Temp_Decomposed <- stl(BR_Temp_ts, s.window = "periodic")
plot(BR_Temp_Decomposed)
```

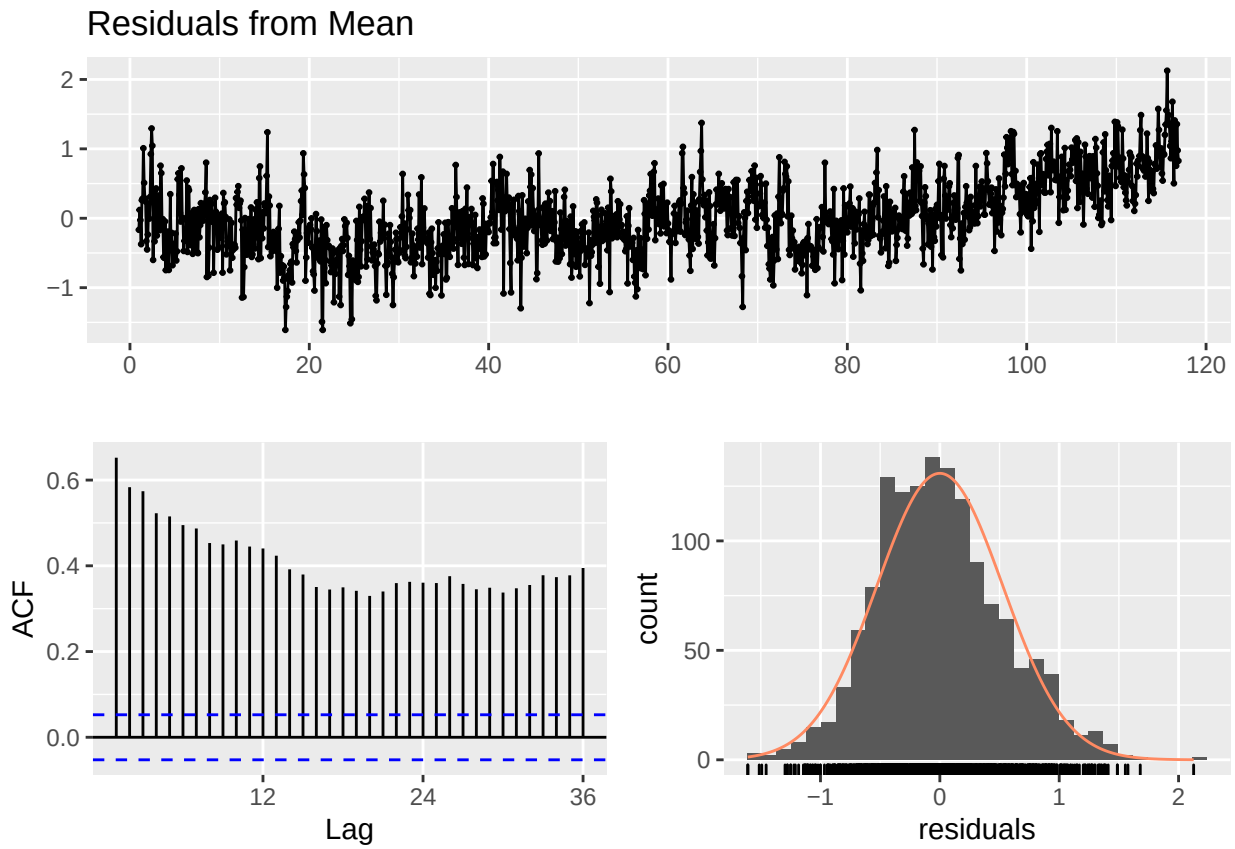



```
BR_Temp_ts_deseas <- seasadj(BR_Temp_Decomposed)
```

Forecasting non-seasonal Temperature

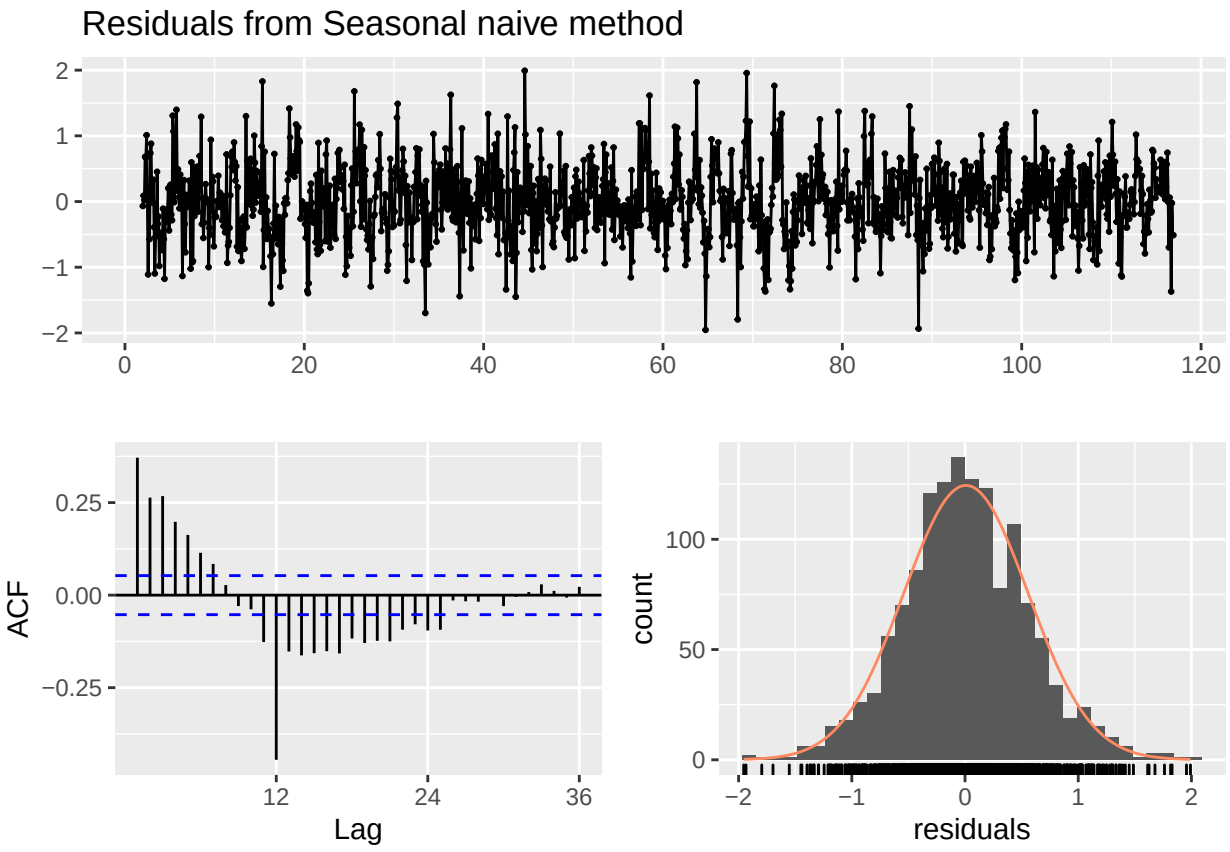
Which models could be used to forecast the non-seasonal temperature series? When fitting a model it's important to do the backward-looking assessment. Check residuals to see if your model is well representing the historical data.

```
#Exercise 3: Try fitting different model to the deseasonal series. Check teh residuals using the checkr

Mean1 <- meanf(BR_Temp_ts_deseas)
checkresiduals(Mean1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 6597.8, df = 23, p-value < 2.2e-16
##
## Model df: 1.   Total lags used: 24
```

```
Naive1 <- snaive(BR_Temp_ts_deseas)
checkresiduals(Naive1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 1097.4, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

Model Performance for forecasting 12 steps ahead

In the lesson we used function `accuracy()` to calculate performance measures like **ME:** Mean Error **RMSE:** Root Mean Squared Error **MAE:** Mean Absolute Error **MPE:** Mean Percentage Error **MAPE:** Mean Absolute Percentage Error **MASE:** Mean Absolute Scaled Error **ACF1:** Autocorrelation of errors at lag 1

```
#Exercise 4: Apply function accuracy() to the forecasts generated with each model you fit for the non-s
```

A better visualization of the results from `accuracy()` can be obtained by creating a data frame where rows correspond to models and columns to metrics. You can choose one metric to help you choose among models.

```
# Exercise 5: create data frame that combines metrics for all the models you created on Ex 3.


# Exercise 6: Decide which model is the best fit by comparing the RMSE metric, i.e, choose model with l
```

Visualization Challenge

```
# Exercise 7: Generate a comparison table for your report using the kbl() function.
```