Statistical Forecasting Individual Project1

Nihar Amitkumar Dave

Student id: 8832838

**STAT8040**

2023

# Data

Time series data contains data points which are collected over regular time intervals. These data points can be collected from official websites, social media or financial stock rates. It is graphically represented as a line graph with time on axis. Time series data is used in finances, economics and weather forecasting. It can provide valuable insights into complex systems and processes.

After a lot of research and discussion with my professor I finally selected data *Unemployment rate of Canada aged of 15 and more* for my report. I took last 40 years (1982-2022) of monthly unemployment data. Since its an unemployment rate of Canada, it doesn’t have a similar trend or cycle. Overall, the trend is decreasing which is economically a good sign. Since the unemployment rate decreased which also suggests that employment rate increased.

Time series data analysis often have many challenges like dealing with missing data or noisy data etc. However, data cleaning can deal with this challenge easily. I manually deleted some of the informative rows from the csv file. Luckily this dataset did not have any missing value or non-stationary values. We do have extreme spike during covid (2020) which is important yet it can influence the data. I had to mutate date using as.date() & kept rate numeric by using as.numeric().

In these uncertain times, economy seems to be bit unpredictable. Researchers are predicting recession in upcoming years. A lot of big tech-companies are firing big number of people. In small jobs, often we see a trend of unemployment rate increasing after holiday season and decreasing holiday season. I want to keep these observations in mind while forecasting on this dataset. This can help economists and policy makers to be prepare a plan accordingly.

library(fpp3)

## ── Attaching packages ──────────────────────────────────────────── fpp3 0.4.0 ──

## ✔ tibble 3.1.8 ✔ tsibble 1.1.3   
## ✔ dplyr 1.0.10 ✔ tsibbledata 0.4.1   
## ✔ tidyr 1.2.1 ✔ feasts 0.3.0   
## ✔ lubridate 1.8.0 ✔ fable 0.3.2   
## ✔ ggplot2 3.4.0

## ── Conflicts ───────────────────────────────────────────────── fpp3\_conflicts ──  
## ✖ lubridate::date() masks base::date()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ tsibble::intersect() masks base::intersect()  
## ✖ tsibble::interval() masks lubridate::interval()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ tsibble::setdiff() masks base::setdiff()  
## ✖ tsibble::union() masks base::union()

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ readr 2.1.2 ✔ stringr 1.5.0  
## ✔ purrr 0.3.4 ✔ forcats 0.5.2  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ lubridate::as.difftime() masks base::as.difftime()  
## ✖ lubridate::date() masks base::date()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ tsibble::intersect() masks lubridate::intersect(), base::intersect()  
## ✖ tsibble::interval() masks lubridate::interval()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ tsibble::setdiff() masks lubridate::setdiff(), base::setdiff()  
## ✖ tsibble::union() masks lubridate::union(), base::union()

library(readxl)   
library(readr)   
library(dplyr)   
library(ggplot2)  
library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo   
##   
## Attaching package: 'forecast'  
##   
## The following objects are masked from 'package:fabletools':  
##   
## accuracy, forecast

library(readr)   
library(stringr)   
library(knitr)   
library(seasonal)

##   
## Attaching package: 'seasonal'  
##   
## The following object is masked from 'package:tibble':  
##   
## view

library(forecast)

setwd("/Users/nihardave/Desktop/Sem2/Statistical Forecasting")  
library(readxl)  
  
Project1\_data <- read\_excel("Project1\_data.xls")%>%  
 mutate(Date = as.Date(Date),   
 Unemployment\_Rate = as.numeric(Unemployment\_Rate)) %>%  
 as\_tsibble(index = Date)  
Project1\_data

## # A tsibble: 481 x 2 [1D]  
## Date Unemployment\_Rate  
## <date> <dbl>  
## 1 1982-12-01 13.1  
## 2 1983-01-01 12.7  
## 3 1983-02-01 12.7  
## 4 1983-03-01 12.5  
## 5 1983-04-01 12.4  
## 6 1983-05-01 12.4  
## 7 1983-06-01 12.4  
## 8 1983-07-01 11.9  
## 9 1983-08-01 11.7  
## 10 1983-09-01 11.4  
## # … with 471 more rows

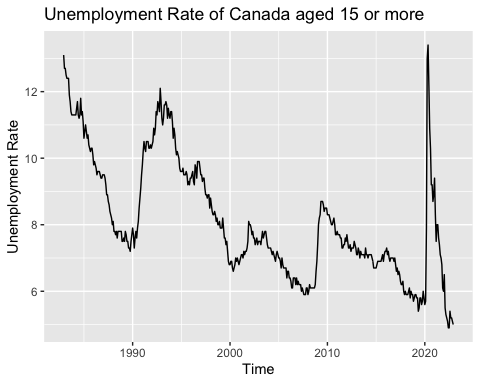
1. Visualization

Creating time series object

data\_ts <- ts(Project1\_data$Unemployment\_Rate, start = c(1982,12,01), frequency =12)

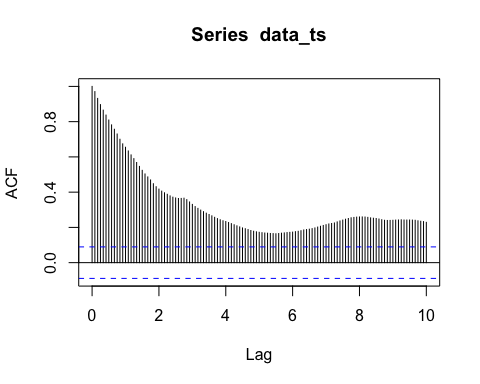
Plotting Time series:

autoplot(data\_ts) + xlab("Time") + ylab("Unemployment Rate") + ggtitle("Unemployment Rate of Canada aged 15 or more")

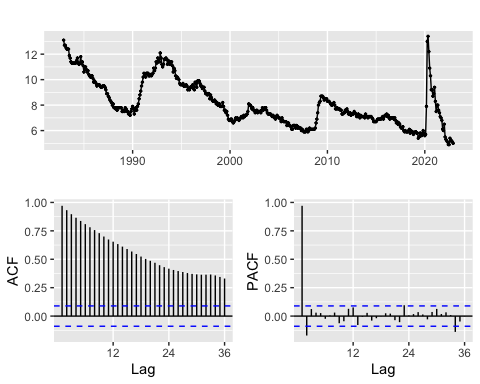


ACF plot:

acf(data\_ts,lag.max = 120)



ggtsdisplay(data\_ts)



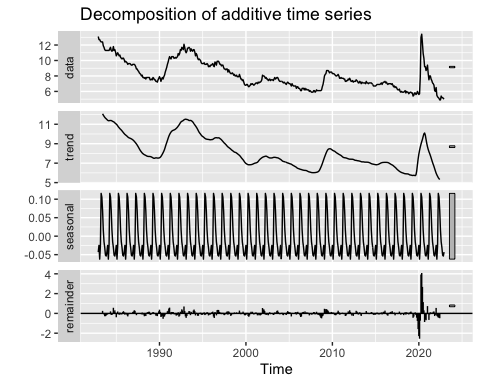
I created time series object and then plotted the data. The underlying trend of the *unemployment rate of Canada aged 15 or more* is negative. Which means over 40 years (1982-2022), unemployment rate is decreasing. There is huge spike in rate when country was hit with recession. But the biggest spike in the unemployment rate is noticed in 2020 but it also decreased in upcoming years.

From the ACF plot, we can notice a positive correlation. When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in value. So, the ACF of a trended time series tends to have positive values that slowly decrease as the lags increase. Whereas, when data is seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal period) than for other lags. We can see combination of trend and seasonality here.

1. Transformation

#### Decompose time series:

data\_decomp <- decompose(data\_ts)  
autoplot(data\_decomp)



Decomposing time series data helps us understand different components that contributed to the overall behavior. I used most basic method of decomposing, which is additive. It breaks down my time series into trend, seasonal and remainder. The trend component represented the overall long-term behavior of the data. Seasonal component captures regular and repeating patterns within a specific time-period. Remainder represents the random fluctuations in the data which we can’t explain in seasonal or trend components. Which can affect the trend of the forecast.

Since, unemployment rates are not affected by season, it can be seasonally adjusted.

Once the time-series is decomposed, we can give importance to components which are important in contributing the overall trend. We can figure that out by forecasting and analysis.

1. Forecasting & Analysis

#### Training and testing data set:

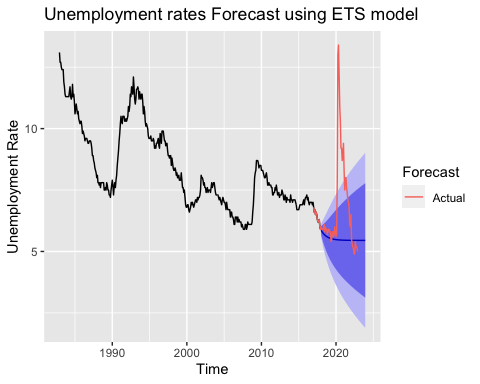
train\_data <- window(data\_ts, end = c(2017,12))  
test\_data <- window(data\_ts, start = c(2017,01))

1. **Fitting a ETS model & generating forecast for testing data:**

data\_ets <- ets(train\_data)  
data\_fc <- forecast::forecast(data\_ets, h = length(test\_data))

Plotting forecast & actual values:

autoplot(data\_fc) + xlab("Time") + ylab("Unemployment Rate") + ggtitle("Unemployment rates Forecast using ETS model") +  
 autolayer(test\_data, series = "Actual") +   
 guides(colour = guide\_legend(title = "Forecast"))

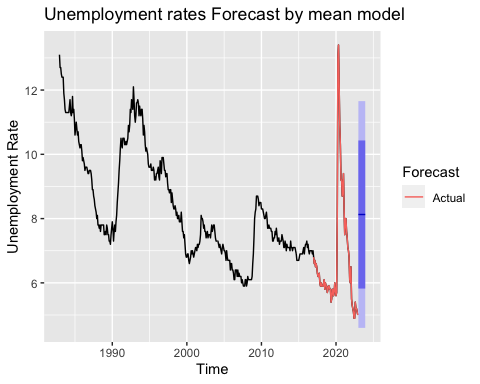


1. **Fitting a Mean model & generating forecast for testing data:**

mean <- meanf(data\_ts, h=12)  
mean\_forecast <- forecast::forecast(mean, h=12)

Plotting forecast & actual values:

autoplot(mean\_forecast) + xlab("Time") + ylab("Unemployment Rate") + ggtitle("Unemployment rates Forecast by mean model") +  
 autolayer(test\_data, series = "Actual") +  
 guides(colour = guide\_legend(title = "Forecast"))

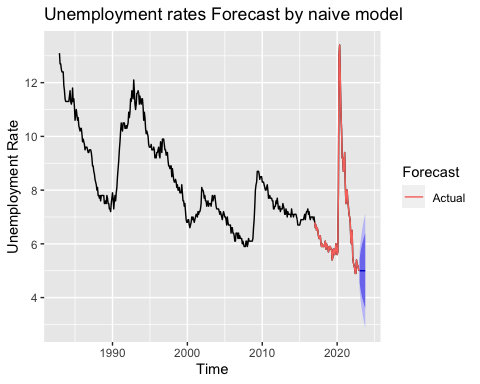


1. **Fitting a Naive model & generating forecast for testing data:**

naive\_data <- naive(data\_ts)  
naive\_forecast <- forecast::forecast(naive\_data)

Plotting forecast & actual values:

autoplot(naive\_forecast) + xlab("Time") + ylab("Unemployment Rate") + ggtitle("Unemployment rates Forecast by naive model") +  
 autolayer(test\_data, series = "Actual") +  
 guides(colour = guide\_legend(title = "Forecast"))

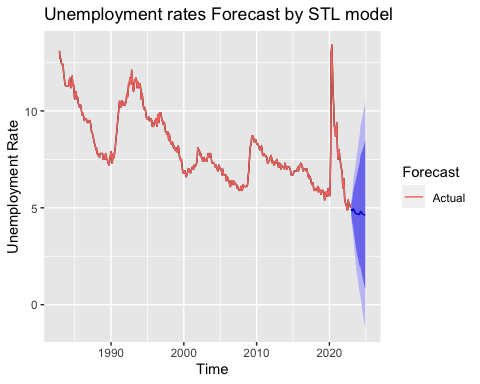


1. **Fitting a STL model & generating forecast for testing data:**

data\_stl <- stlf(data\_ts, s.window = "periodic")

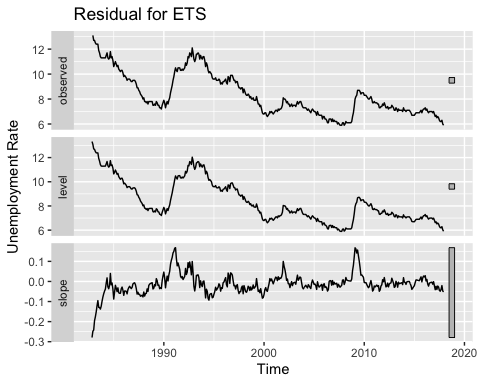
Plotting forecast & actual values:

autoplot(data\_stl) + xlab("Time") + ylab("Unemployment Rate") + ggtitle("Unemployment rates Forecast by STL model") +  
 autolayer(data\_ts, series = "Actual") +   
 guides(colour = guide\_legend(title = "Forecast"))



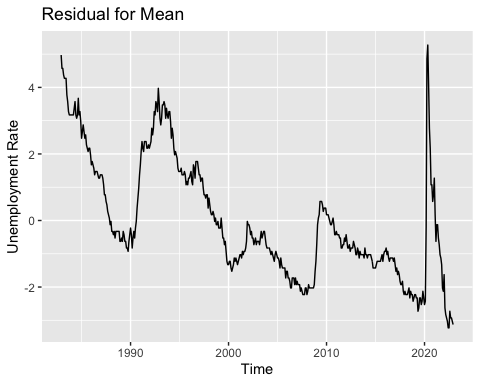
1. **Residual for ETS forecasting:**

ets\_residual <- residuals(data\_ets)  
autoplot(data\_ets)+xlab("Time") + ylab("Unemployment Rate") + ggtitle("Residual for ETS")



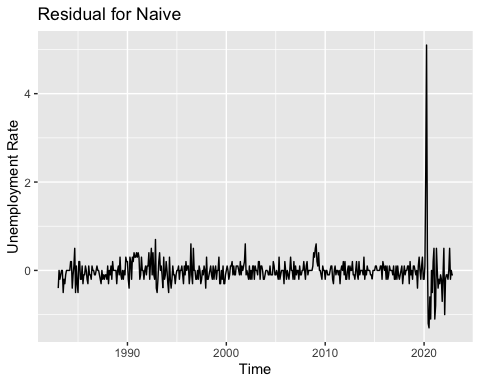
1. **Residual of Mean forecast:**

mean\_residual <- residuals(mean\_forecast)  
autoplot(mean\_residual)+xlab("Time") + ylab("Unemployment Rate") + ggtitle("Residual for Mean")



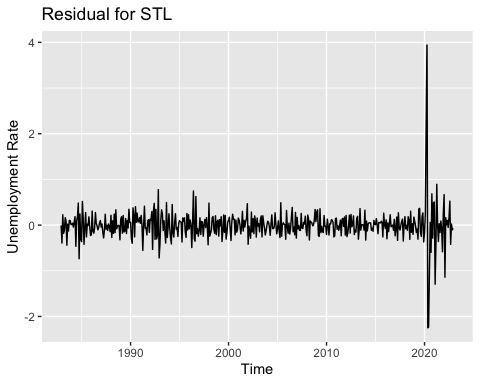
1. **Residual of naive forecast:**

naive\_residual <- residuals(naive\_forecast)  
autoplot(naive\_residual)+xlab("Time") + ylab("Unemployment Rate") + ggtitle("Residual for Naive")



1. **Residual of STL Forecast:**

STL\_residual <- residuals(data\_stl)  
autoplot(STL\_residual)+xlab("Time") + ylab("Unemployment Rate") + ggtitle("Residual for STL")



Forecasting and analysis is necessary for decision-making and planning. I have run four models to forecast future trends from the historical data. Initially I trained the data until Jan-2017 and tested the data from Dec-2017. After training and testing the data, I can apply it on various models.

(1) ETS model gives us a good range of forecast, where the trend consists of. The residuals of ETS model are helpful.

(2) Mean model gives us somewhat a good forecast, but it is not very helpful since it will be the average of overall trends. Even residuals of mean are not so helpful.

(3) Naïve model gives us decreasing trend and a range of forecast, which is helpful. Residuals of Naïve model are helpful as well.

(4) STL based model gives us decreasing trend and a range of forecast, which is very helpful. Residuals of STL based model are helpful as well.

I would personally prefer ETS, Naïve and STL based models to forecast this kind of data. According to me, ETS model gives us best desirable forecast. This model can help us predict the future trends. According to these models it suggests that the unemployment rate will be decreasing in future. This prediction will try to respond to changing market conditions and global economy.

Reference:

Organization for Economic Co-operation and Development, Unemployment Rate: Aged 15 and over: All Persons for Canada [LRUNTTTTCAM156S], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/LRUNTTTTCAM156S, February 23, 2023.