Your full name: Ajay Kanubhai Patel

Student number: 8828978

Course number: STAT8040-23F-Sec1-Statistical Forecasting

1. Data

What is the motive behind choosing this data set and its possible implementation in real life situations?

The requirements of the project suggest that data has a time dimension so that I can perform time-series analysis. Moreover, In order to create a forecasting model and acquire a better understanding of the dynamics of the labor market, we examine a time series data set of unemployment rates in this study. The selection of unemployment data is driven by its importance to the economy and society. Making informed judgments can benefit companies, job seekers, and policymakers by having a thorough understanding of unemployment trends and future rate forecasts. Our goal is to provide answers to inquiries like: What are the long-term patterns in unemployment? Can we forecast future unemployment rates to help with labor force planning and look for seasonal patterns?

Data Source: The data set was downloaded from FRED. Federal Reserve Economic Data (FRED) is an online database consisting of hundreds of thousands of economic data time series from scores of national, international, public, and private sources (Federal Reserve Economic Data, 2023).

Link to download data set: <https://fred.stlouisfed.org/series/UNRATE>

Note: More information related to this data set is given further after analyzing it in R.

library(fpp3)  
library(tidyr)  
library(lubridate)  
library(readr)  
  
  
setwd("C:/PREDICTIVE ANALYTICS/SEM-2/STAT FORECASTING/INDIVIDUAL PROJECT")  
Project <- read\_xls("C:/PREDICTIVE ANALYTICS/SEM-2/STAT FORECASTING/INDIVIDUAL PROJECT/Unemployment.xls")  
summary(Project)

|  | observation\_date | UNRATE |
| --- | --- | --- |
|  | Min. :1948-01-01 00:00:00.000 | Min. : 2.500 |
|  | 1st Qu.:1966-11-23 12:00:00.000 | 1st Qu.: 4.400 |
|  | Median :1985-10-16 12:00:00.000 | Median : 5.500 |
|  | Mean :1985-10-16 02:00:31.717 | Mean : 5.714 |
|  | 3rd Qu.:2004-09-08 12:00:00.000 | 3rd Qu.: 6.725 |
|  | Max. :2023-08-01 00:00:00.000 | Max. :14.700 |

dim(Project)

## [1] 908 2

str(Project)

## tibble [908 × 2] (S3: tbl\_df/tbl/data.frame)  
## $ observation\_date: POSIXct[1:908], format: "1948-01-01" "1948-02-01" ...  
## $ UNRATE : num [1:908] 3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...

class(Project$observation\_date)

## [1] "POSIXct" "POSIXt"

head(Project$observation\_date)

## [1] "1948-01-01 UTC" "1948-02-01 UTC" "1948-03-01 UTC" "1948-04-01 UTC"  
## [5] "1948-05-01 UTC" "1948-06-01 UTC"

Project$observation\_date <- as.Date(Project$observation\_date, format="%m/%d/%Y")

We examine a time series data set of unemployment rates covering the period from January 1948 to August 2023 in this study. The unemployment rate of those who currently reside in 1 of the 50 states or the District of Columbia is observed monthly in the data set. Our goal is to learn more about the unemployment rate’s seasonality, historical trends, and possibilities for forecasting.

There are 908 observations in all in the data set and observation\_date and UNRATE two variables given, with a maximum unemployment rate of 14.7% and a minimum of 2.5%. An average level of unemployment over time is indicated by the 5.714% mean unemployment rate for the entire period. In addition, There is no Null value present in our data set.

Datatype: observation\_date is in POSIXct and POSIXt format while UNRATE is in numeric format.

Project <- Project[order(Project$observation\_date),]  
min\_d <- min(Project$observation\_date)  
max\_d <- max(Project$observation\_date)  
Project\_ts <- ts(Project$UNRATE, start = c(1948,01), end = c(2023,08) , frequency = 12)  
summary(Project\_ts)

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| --- | --- | --- | --- | --- | --- |
| 2.5 | 4.4 | 5.5 | 5.713767 | 6.725 | 14.7 |

head(Project\_ts)

## Jan Feb Mar Apr May Jun  
## 1948 3.4 3.8 4.0 3.9 3.5 3.6

str(Project\_ts)

## Time-Series [1:908] from 1948 to 2024: 3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...

#Let's check whether we have tranformed data in time-series or not?  
class(Project\_ts)

## [1] "ts"

It is good idea to import data in Chronological order before transforming into time-series. From above functions, we can see the observation date is starting from 1948-01-01 to 2023-08-01. Moreover, we are declaring monthly frequency.

1. Visualization

library(TSstudio)

## Warning: package 'TSstudio' was built under R version 4.3.1

library(plotly)

## Warning: package 'plotly' was built under R version 4.3.1

##   
## Attaching package: 'plotly'

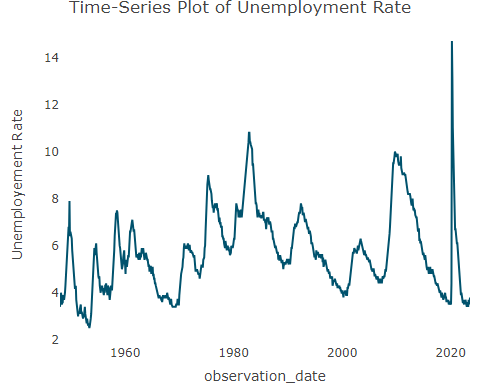
## The following objects are masked from 'package:plyr':  
##   
## arrange, mutate, rename, summarise

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

ts\_plot(Project\_ts,  
 title = "Time-Series Plot of Unemployment Rate",  
 Xtitle = "observation\_date",  
 Ytitle = "Unemployement Rate")



Here, we can observe cyclic pattern in time-series plot.Moreover, this is an unemployment data so seasonality is not important. There is an outlier clearly observed from above plot. Additionally, there is no trend in our time-series data, however, if we carefully observe time series then we can see upward trend from 1950 to 1983.

About stationarity: Here, mean and variance is constant over time and no seasonality Present in our time-series so we can consider it as stationary time-series.

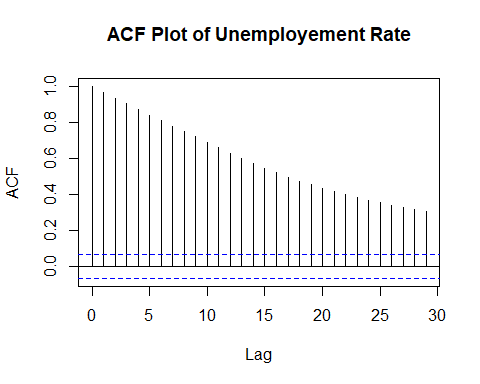
install.packages("forecast")

## Warning: package 'forecast' is in use and will not be installed

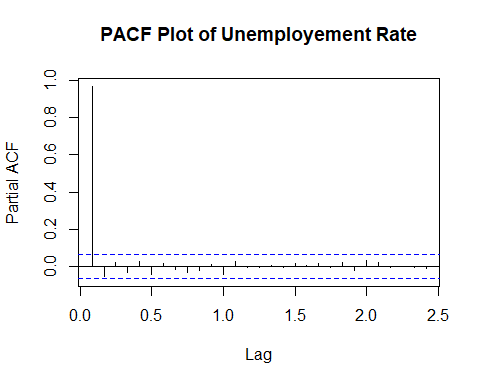
Project\_ts %>% ggAcf(plot=FALSE)

##   
## Autocorrelations of series '.', by lag  
##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12   
## 1.000 0.968 0.935 0.903 0.870 0.840 0.809 0.779 0.750 0.719 0.689 0.659 0.629   
## 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 0.601 0.574 0.547 0.522 0.497 0.475 0.454 0.434 0.415 0.399 0.382 0.367 0.354   
## 26 27 28 29   
## 0.342 0.330 0.318 0.306

acf(Project$UNRATE, main = "ACF Plot of Unemployement Rate")



pacf(Project\_ts, main = "PACF Plot of Unemployement Rate")



I have created Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to understand autocorrelation in our data and to find patterns (Udit, 2022).

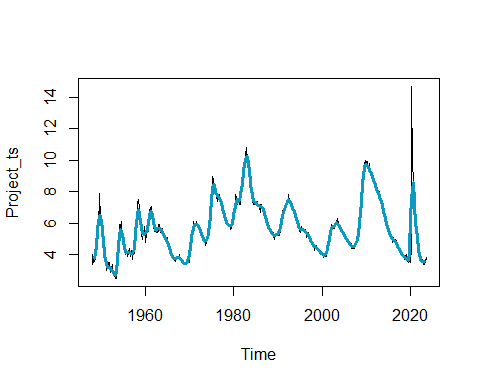
From ACF, we can observe there isn’t any seasonality. In addition, there is a trend as autocorrelations are strong and positive. When we observe lag=0 which compare with itself which is 1 that means the re is no concern regarding our data. blue dotted line represents level of significant and we can clearly see that all the lines are above blue dotted line so all the lines are significant. we can conclude that there is autocorrelation in our data. Apart from this, there is autocorrelation so we can conclude that there isn’t white noise in our series.

It is good idea to ignore seasonality in data set like unemployment. let’s plot graph and observe.

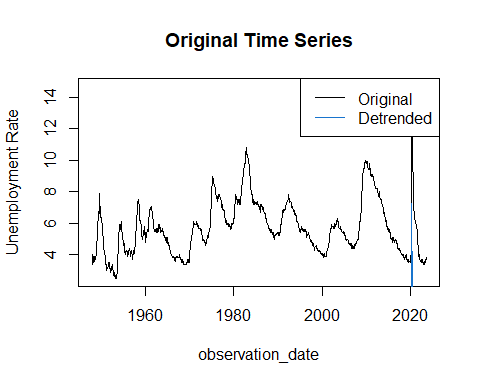
1. Transformations

Moving Average

MA\_12 <- forecast::ma(Project\_ts, order = 12, centre = TRUE)  
plot(Project\_ts)  
lines(MA\_12, col=5, lwd=3)

 Above we have calculated moving average of order 12 and then plotted moving average line on time-series plot.We can clearly see that our moving average line is completely overlapping on our time-series plot which indicated that there is no trend and seasonality in the data. Moreover, it represents pattern is relatively stable.

detrended\_ts <- Project\_ts - MA\_12  
  
plot(Project\_ts, main = "Original Time Series", ylab = "Unemployment Rate", xlab = "observation\_date")  
lines(detrended\_ts, col = 4)  
legend("topright", legend = c("Original", "Detrended"), col = c(1, 4), lty = 1)

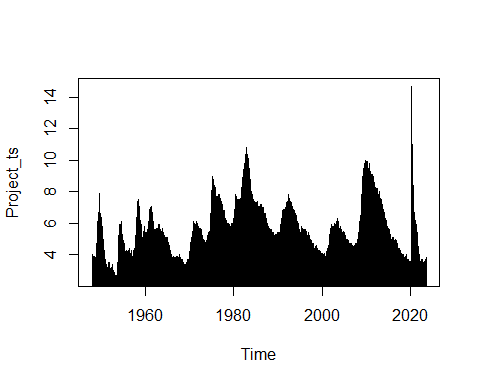


Decomposition of Time-Series using classical Method.

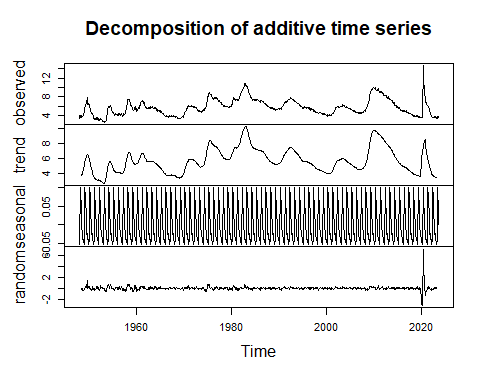
Do we need to transform series using Box-Cox Transformation? Here, we can observe that there is no presence of heteroskedasticity which means variance is neither increases or decreases over time so there is no need of transformation (Labrinos, 2023c).

What kind of decomposition and why? here, additive decomposition would be more suitable option as magnitude of fluctuation remain constant over time (Labrinos, 2023c).

library(ggplot2)  
library(forecast)  
  
plot(Project\_ts, type="h")

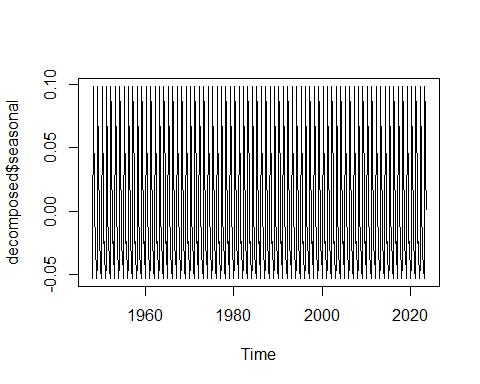


# Decompose the time series  
  
decomposed <- decompose(Project\_ts)  
  
# Plot the decomposed components  
plot(decomposed)



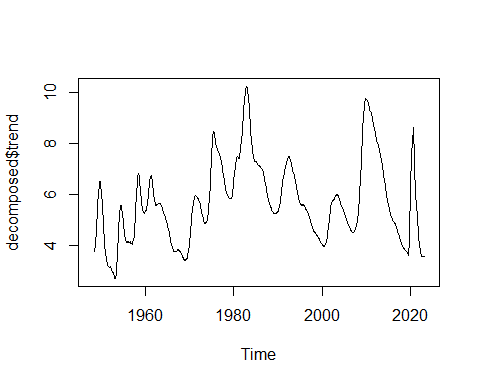
Seasonal component of time-series.

plot(decomposed$seasonal)

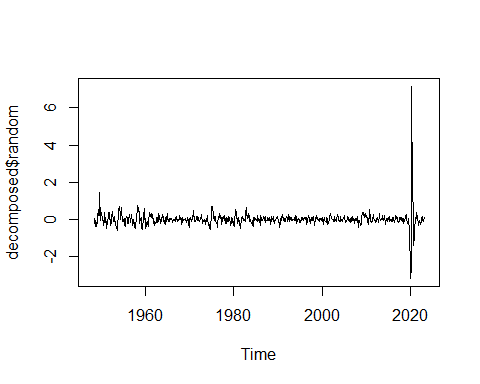
 When we present season part of time series via graph when can see it represented on the scale of -0.05 to 0.10. Moreover, it is good idea to ignore seasonality in data set like unemployment.

Trend component.

plot(decomposed$trend)

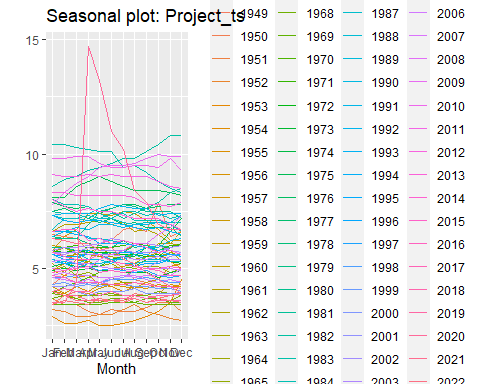
 When we observe graph of trend of our time-series we can see it is almost similar to our original series. It is represented on the scale between 4 to 10.

plot(decomposed$random)

 We can observes errors from above graphs and it is quite stable, however, there is large spike can be observe at around 2020.

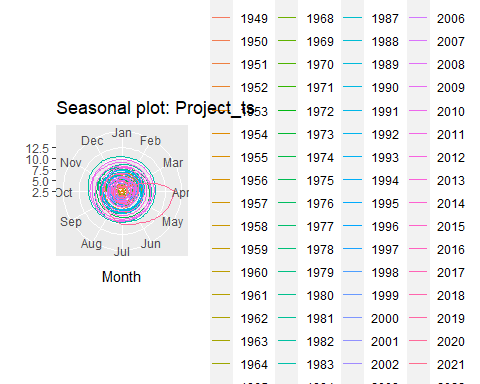
Let’s create different types of plots and observe.

ggseasonplot(Project\_ts)

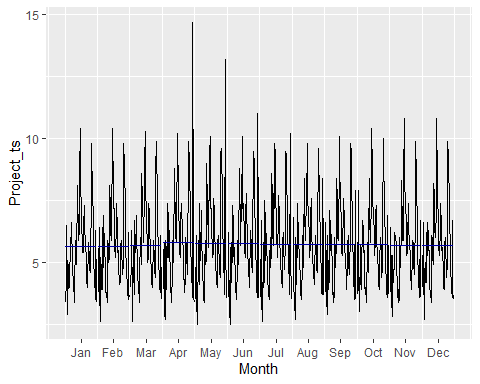


Here, I have created graph which seems really creative and easy to observe. In above graph, we can clearly see that there is one year which is represented is red color has the highest unemployment rate. Moreover, when we observe all other year from January to December there is approximately constant value of unemployment rate throughout year.

ggseasonplot(Project\_ts, polar=TRUE)

 From above polar graph, we can clearly see in one year month of April has the highest unemployment rate.

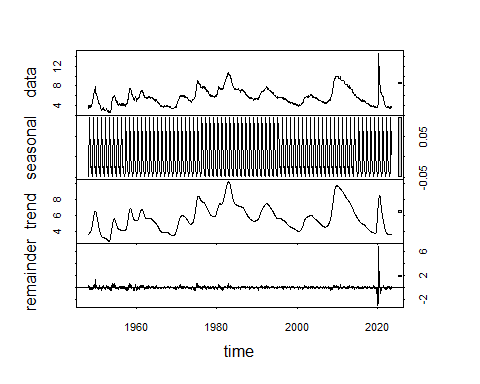
ggsubseriesplot(Project\_ts)



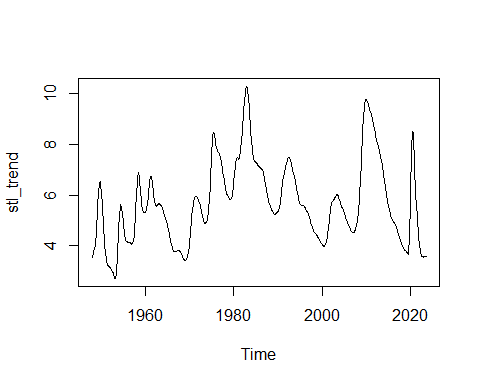
From all the graphs, we have seen value which is extremely large as compared to other values of unemployment rate so we can research what did happen at that time what causes unemployment which was unseen in history. we can also get insight from the events occurred at that time which were responsible for higher unemployment and we can prevent to happening in future.

STL Method for time series decomposition.

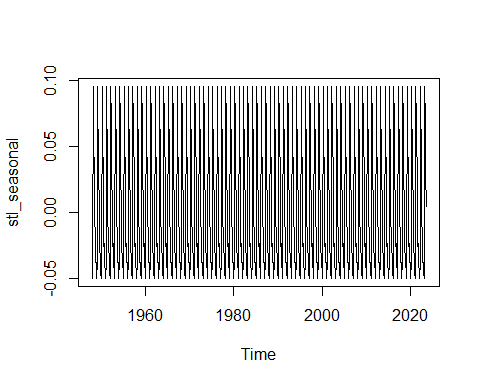
stl\_decomp <- stl(Project\_ts, s.window = "periodic")  
stl\_trend <- stl\_decomp$time.series[, "trend"]  
stl\_seasonal <- stl\_decomp$time.series[, "seasonal"]  
stl\_residual <- stl\_decomp$time.series[, "remainder"]  
  
plot(stl\_decomp)



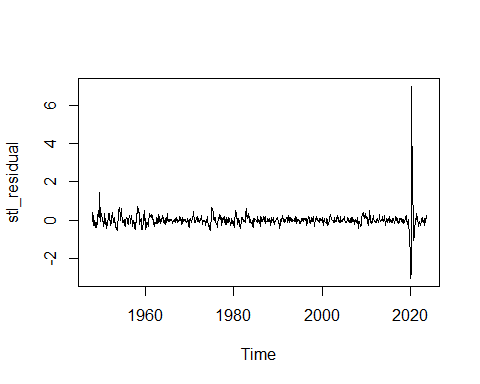
plot(stl\_trend)



plot(stl\_seasonal)



plot(stl\_residual)



1. Forecasting and Analysis

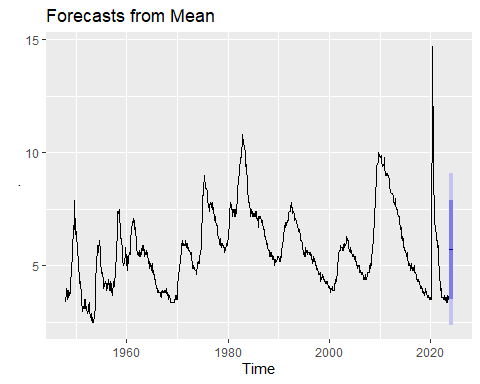
Forecasting with benchmark methods

1.Average Method

average\_method <-Project\_ts %>% meanf(h=10)  
average\_method

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 2023 5.713767 3.522224 7.905309 2.360058 9.067475  
## Oct 2023 5.713767 3.522224 7.905309 2.360058 9.067475  
## Nov 2023 5.713767 3.522224 7.905309 2.360058 9.067475  
## Dec 2023 5.713767 3.522224 7.905309 2.360058 9.067475  
## Jan 2024 5.713767 3.522224 7.905309 2.360058 9.067475  
## Feb 2024 5.713767 3.522224 7.905309 2.360058 9.067475  
## Mar 2024 5.713767 3.522224 7.905309 2.360058 9.067475  
## Apr 2024 5.713767 3.522224 7.905309 2.360058 9.067475  
## May 2024 5.713767 3.522224 7.905309 2.360058 9.067475  
## Jun 2024 5.713767 3.522224 7.905309 2.360058 9.067475

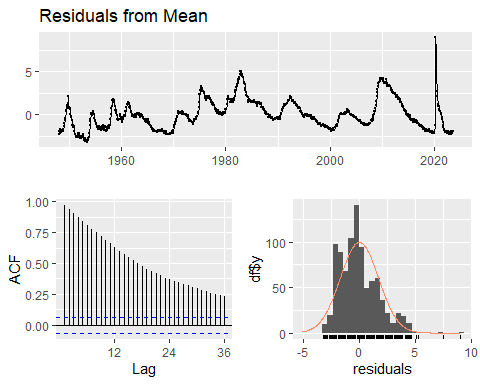
autoplot(average\_method)



avg\_res <- Project\_ts %>% meanf(h=10) %>% residuals()  
  
Box.test(avg\_res, fitdf = 0, type = "Lj")

##   
## Box-Ljung test  
##   
## data: avg\_res  
## X-squared = 854.39, df = 1, p-value < 2.2e-16

checkresiduals(Project\_ts %>% meanf(h=10))



##   
## Ljung-Box test  
##   
## data: Residuals from Mean  
## Q\* = 9625.4, df = 24, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 24

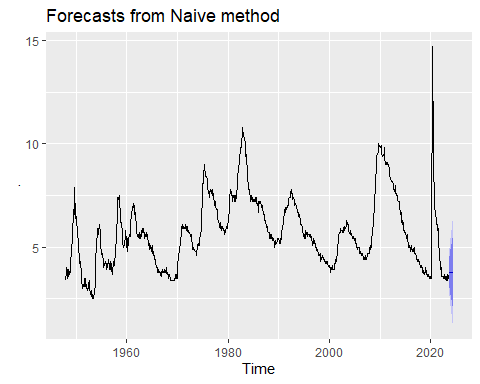
When we observe above graphs, we clearly see that mean is not zero and there is presence of outlier. Moreover, variance is fluctuating. residuals is slightly right skewed. Additionally, p-value of Ljung-Box test is extremely small which suggest that series is not white noise and this is clearly seen in ACF plot that all lags are statistically significant.

1. Naive Method

naive\_method <-Project\_ts %>% naive(h=10)  
naive\_method

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 2023 3.8 3.265080 4.334920 2.981911 4.618089  
## Oct 2023 3.8 3.043510 4.556490 2.643048 4.956952  
## Nov 2023 3.8 2.873492 4.726508 2.383029 5.216971  
## Dec 2023 3.8 2.730161 4.869839 2.163822 5.436178  
## Jan 2024 3.8 2.603884 4.996116 1.970698 5.629302  
## Feb 2024 3.8 2.489720 5.110280 1.796100 5.803900  
## Mar 2024 3.8 2.384736 5.215264 1.635540 5.964460  
## Apr 2024 3.8 2.287019 5.312981 1.486095 6.113905  
## May 2024 3.8 2.195241 5.404759 1.345733 6.254267  
## Jun 2024 3.8 2.108436 5.491564 1.212976 6.387024

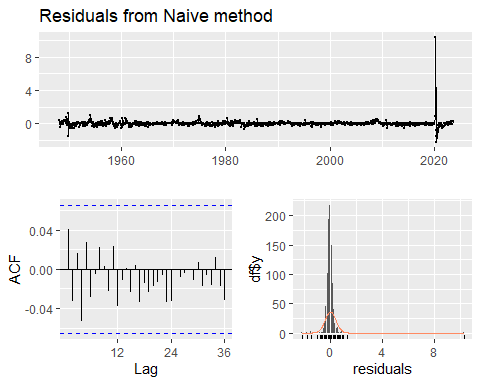
autoplot(naive\_method)



naive\_res <- Project\_ts %>% naive(h=10) %>% residuals()  
  
  
Box.test(naive\_res, fitdf = 0, type = "Lj")

##   
## Box-Ljung test  
##   
## data: naive\_res  
## X-squared = 1.4835, df = 1, p-value = 0.2232

checkresiduals(Project\_ts %>% naive(h=10))



##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 14.432, df = 24, p-value = 0.9363  
##   
## Model df: 0. Total lags used: 24

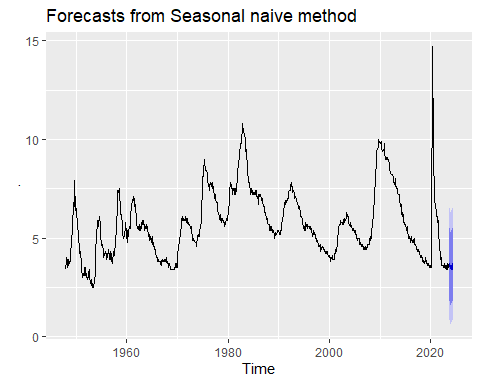
From naive method, mean is zero and variance is constant over time. However, there is an outlier present in the series. Residuals are normally distributed. From Ljung-Box test, p-value is extemely high which suggests that series is white noise and ACF plot also justify this as all lags are below significant line.

Seasonal Naive Method

snaive\_method <-Project\_ts %>% snaive(h=10)  
snaive\_method

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 2023 3.5 1.652156 5.347844 0.6739672 6.326033  
## Oct 2023 3.7 1.852156 5.547844 0.8739672 6.526033  
## Nov 2023 3.6 1.752156 5.447844 0.7739672 6.426033  
## Dec 2023 3.5 1.652156 5.347844 0.6739672 6.326033  
## Jan 2024 3.4 1.552156 5.247844 0.5739672 6.226033  
## Feb 2024 3.6 1.752156 5.447844 0.7739672 6.426033  
## Mar 2024 3.5 1.652156 5.347844 0.6739672 6.326033  
## Apr 2024 3.4 1.552156 5.247844 0.5739672 6.226033  
## May 2024 3.7 1.852156 5.547844 0.8739672 6.526033  
## Jun 2024 3.6 1.752156 5.447844 0.7739672 6.426033

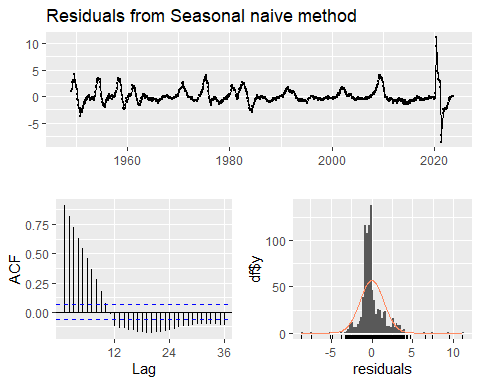
autoplot(snaive\_method)



snaive\_res <- Project\_ts %>% snaive(h=10) %>% residuals()  
  
  
Box.test(snaive\_res, fitdf = 0, type = "Lj")

##   
## Box-Ljung test  
##   
## data: snaive\_res  
## X-squared = 747.68, df = 1, p-value < 2.2e-16

checkresiduals(Project\_ts %>% snaive(h=10))



##   
## Ljung-Box test  
##   
## data: Residuals from Seasonal naive method  
## Q\* = 3183.8, df = 24, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 24

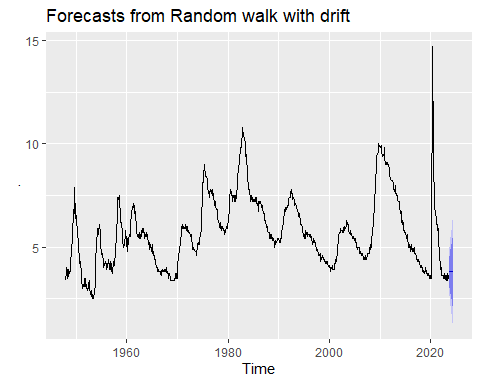
From snaive method, mean is not zero. There is sign of outlier. Moreover, distribution is slightly right skewed. Additionally, Ljung-Box test test value is quite small which indicates absense of white noise and ACF plot also justify this as all lags are statistically significant.

1. Drift Method

drift\_method <-Project\_ts %>% rwf(h=10, drift = TRUE)  
drift\_method

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 2023 3.800441 3.264932 4.335950 2.981450 4.619432  
## Oct 2023 3.800882 3.043141 4.558624 2.642016 4.959748  
## Nov 2023 3.801323 2.872773 4.729873 2.381228 5.221418  
## Dec 2023 3.801764 2.728978 4.874551 2.161079 5.442449  
## Jan 2024 3.802205 2.602135 5.002275 1.966857 5.637554  
## Feb 2024 3.802646 2.487315 5.117977 1.791021 5.814272  
## Mar 2024 3.803087 2.381589 5.224585 1.629094 5.977081  
## Apr 2024 3.803528 2.283052 5.324004 1.478160 6.128896  
## May 2024 3.803969 2.190379 5.417559 1.336197 6.271742  
## Jun 2024 3.804410 2.102609 5.506211 1.201730 6.407090

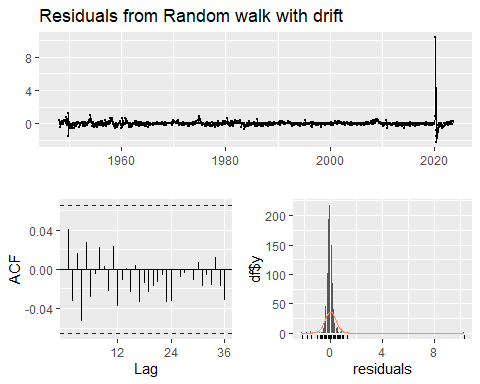
autoplot(drift\_method)



drift\_res <- Project\_ts %>% rwf(h=10, drift = TRUE) %>% residuals()  
  
Box.test(drift\_res, fitdf = 0, type = "Lj")

##   
## Box-Ljung test  
##   
## data: drift\_res  
## X-squared = 1.4835, df = 1, p-value = 0.2232

checkresiduals(Project\_ts %>% rwf(h=10, drift = TRUE))



##   
## Ljung-Box test  
##   
## data: Residuals from Random walk with drift  
## Q\* = 14.432, df = 24, p-value = 0.9363  
##   
## Model df: 0. Total lags used: 24

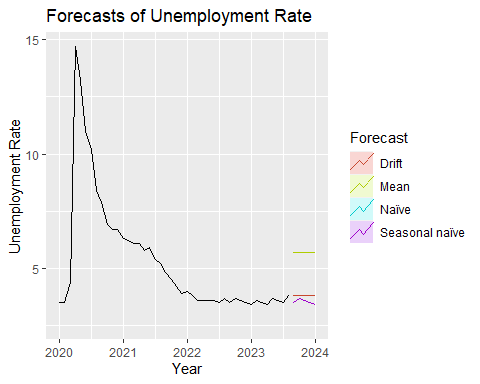
From Drift method, residuals have zero mean and variance is constant over time, however, there is an outlier. Residuals are normally distributed. Apart from this, all lags are below significant level and Ljung-Box test value is quite high which suggests that series has white noise.

Conclusion: From residual perspective, each method violates one or more conditions so we need to do further operations mentioned below. 1. if residual are not normally distributed then use bootstraps and and create prediction intervals (Lambrinos, 2023e).

we cannot say which method is the most suitable method for our timeseries from residual test.

plotting <- autoplot (Project\_ts) +  
 autolayer (meanf (Project\_ts, h = 10),  
 series="Mean", PI = FALSE) +  
 autolayer (naive(Project\_ts, h = 10),  
 series = "Naïve", PI = FALSE) +  
 autolayer (snaive(Project\_ts, h = 10),  
 series = "Seasonal naïve", PI=FALSE) +  
autolayer (rwf(Project\_ts, h = 10, drift = TRUE),  
 series = "Drift", PI=FALSE) +  
 ggtitle("Forecasts of Unemployment Rate") +  
 xlab ("Year") + ylab ("Unemployment Rate") +  
 guides (colour = guide\_legend (title = "Forecast"))  
  
  
plotting + xlim(2020, 2024)

## Scale for x is already present.  
## Adding another scale for x, which will replace the existing scale.

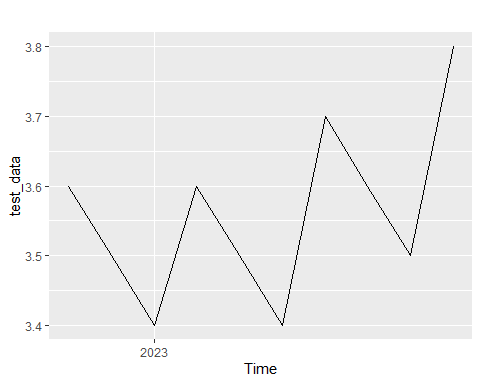


We can divide data set into training and test data set and then we can calculate RMSE to decide which method is the best for our time series forecasting.

Let’s divide Data into train and test.

horizon <- 10  
train\_data <- window(Project\_ts, start = c(1948, 1), end = c(2023, 8 - horizon))  
test\_data <- window(Project\_ts, start = c(2023, 8 - horizon + 1), end = c(2023, 8))  
  
autoplot(test\_data, level=NULL)

## Warning in ggplot2::geom\_line(na.rm = TRUE, ...): Ignoring unknown parameters:  
## `level`



IN train data set we are including data from January 1948 to October 2022. While, test data set contains data from September 2022 to August 2023.

Mean Forecast: All future time periods are forecasted using the historical mean of the data, which is known as the mean forecast. This model makes the assumption that values in the future will resemble the historical mean. It’s a very basic, traditional model. In our data unemployment rate is reasonably stable, it might be appropriate (Labrinos, 2023d).

mean\_model <- meanf(train\_data, h = horizon)  
  
mean\_forecast <- forecast(mean\_model, h = horizon)  
  
forecast\_values\_mean <- mean\_forecast$mean  
actual\_values\_mean <- test\_data  
  
# Calculate MAE  
mae\_mean <- mean(abs(forecast\_values\_mean - actual\_values\_mean))  
  
# Calculate MSE  
mse\_mean <- mean((forecast\_values\_mean - actual\_values\_mean)^2)  
  
# Calculate RMSE  
rmse\_mean <- sqrt(mse\_mean)

Let’s try naive forecasting as in our data no seasonality and trend are present so it is good idea to go with this method to forecast our time-series (Labrinos, 2023d).

naive\_model <- naive(train\_data, h = horizon)  
  
# Generate forecasts  
naive\_forecast <- forecast(naive\_model, h = horizon)  
  
  
forecast\_values\_naive <- naive\_forecast$mean  
actual\_values\_naive <- test\_data  
  
# Calculate MAE  
mae\_naive <- mean(abs(forecast\_values\_naive - actual\_values\_naive))  
  
# Calculate MSE  
mse\_naive <- mean((forecast\_values\_naive - actual\_values\_naive)^2)  
  
# Calculate RMSE  
rmse\_naive <- sqrt(mse\_naive)

Seasonal Naive Forecast:A modification of the naïve forecast that takes seasonality into consideration is the seasonal naive forecast. It forecasts the equivalent season of the current year using the observation from the same season the year before. In our data there is no seasonal pattern present so it is not suitable method to forecast time-series with this method using our data, however, I am going to perform and check the results for better understanding (Labrinos, 2023d).

snaive\_model <- snaive(train\_data, h = horizon)  
  
snaive\_forecast <- forecast(snaive\_model, h = horizon)  
  
forecast\_values\_snaive <- snaive\_forecast$mean  
actual\_values\_snaive <- test\_data  
  
# Calculate MAE  
mae\_snaive <- mean(abs(forecast\_values\_snaive - actual\_values\_snaive))  
  
# Calculate MSE  
mse\_snaive <- mean((forecast\_values\_snaive - actual\_values\_snaive)^2)  
  
# Calculate RMSE  
rmse\_snaive <- sqrt(mse\_snaive)

The Drift Forecast (DRIFT) is a basic model that assumes a linear trend in the collected data. By extending the trend from the most recent data points, it predicts future values (Labrinos, 2023d).

drift\_model <- rwf(train\_data, drift = TRUE, h = horizon)  
  
drift\_forecast <- forecast(drift\_model, h = horizon)  
  
forecast\_values\_drift <- drift\_forecast$mean  
actual\_values\_drift <- test\_data  
  
# Calculate MAE  
mae\_drift <- mean(abs(forecast\_values\_drift - actual\_values\_drift))  
  
# Calculate MSE  
mse\_drift <- mean((forecast\_values\_drift - actual\_values\_drift)^2)  
  
# Calculate RMSE  
rmse\_drift <- sqrt(mse\_drift)

We are using same data set for forecasting so it is good idea to find Absolute Error (MAE) and Root Mean Squared Error and select model which has the lowest value (Labrinos, 2023e).

mae\_mean

## [1] 2.177751

mae\_naive

## [1] 0.16

mae\_snaive

## [1] 0.23

mae\_drift

## [1] 0.1611706

rmse\_mean

## [1] 2.181054

rmse\_naive

## [1] 0.1843909

rmse\_snaive

## [1] 0.3146427

rmse\_drift

## [1] 0.1855236

From above Mean Absolute Error (MAE) and Root Mean Squared Error values (RMSE) we can see that naive and drift forecast have lower values compare to other forecast method so from MAE and RMSE perspective naive and drift methods are more suitable (Wee5:BasicForecastingII, 2023e).

References:

1. Lambrinos,P.(2023a).Week 1: Introduction to Forecasting [Powerpoint slide] <https://conestoga.desire2learn.com/d2l/le/content/879796/viewContent> /18588762/View
2. Lambrinos,P.(2023b).Week 2: Time Series Graphics [Powerpoint slide] <https://conestoga.desire2learn.com/d2l/le/content/879796/viewContent> /18588763/View
3. Lambrinos,P.(2023c).Week 3: Time Series Decomposition [Powerpoint slide] <https://conestoga.desire2learn.com/d2l/le/content/879796/viewContent/> 18588764/View
4. Lambrinos,P.(2023d).Week 4: Basic Forecasting [Powerpoint slide] <https://conestoga.desire2learn.com/d2l/le/content/879796/viewContent/> 18588765/View
5. Lambrinos,P.(2023e).Week 5: Basic Forecasting II [Powerpoint slide] <https://conestoga.desire2learn.com/d2l/le/content/879796/viewContent/> 18588766/View
6. Lambrinos,P.(2023f).Week 6: Time Series Regression [Powerpoint slide] <https://conestoga.desire2learn.com/d2l/le/content/879796/viewContent/> 18588767/View
7. Udit.(2022, Dec 30). Deciphering ACF and PACF Plots: A Guide to Time Series Forecasting <https://itsudit.medium.com/deciphering-acf-and-pacf-plots-a-guide-to-> time-series-forecasting-3323948935fb#:~:text=The%20ACF%20plot%20can% 20be,likely%20be%20a%20good%20fit.
8. U.S. Bureau of Labor Statistics. (2023, October 11). Unemployment Rate [UNRATE], retrieved from FRED, Federal Reserve Bank of St. Louis. <https://fred.stlouisfed.org/series/UNRATE>
9. Federal Reserve Economic Data: Your trusted data source since 1991. (n.d.). <https://fredhelp.stlouisfed.org/fred/about/about-fred/what-is-fred/>