US Minimum Wage by State

Time Series Modeling of US Wages by States From 2010 Through 2020

Project Report

DSCI 5260 - Fall 2021

<u>Group 10</u>

Architbhai Vaghani
Saja Alogaidi
Venkata Eswar Vemuri
Yamini Mangarai

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Abstract:

Time series forecasting plots are often used in domains such as economics, public health, education policy, and social services research when examining the influence of an interruption on a result. The data set used in this research consists of 41 states covering the years 2010 to 2020, with 1271 observations throughout the whole period. The three-time series models (ARIMA, Holt's Method, and Simple Exponential Method) for two variables (state minimum wages and state minimum wages-2020) were explored for the two variables in question. When compared to other models, the ARIMA model performed exceptionally well with the data. The results of this study indicate that delving deeper into time series plots is a valuable tool for future evaluations to inform policy choices.

Keywords: ADF, KPSS, ARIMA, and Forecasting.

1. Background/Introduction

The numerous states have minimum salaries that are higher than the federal level, even though the federal minimum wage in the United States had not increased since 2009, when it was $7.25 an hour, according to Kuroki (2021). In places where the minimum wage is raised, policymakers are mainly concerned with the negative effects on the labor market; nevertheless, a growing number of studies are looking at the health repercussions of minimum wage rises (Leigh et al., 2019). An essential motivation for establishing minimum wage legislation is the desire to increase the incomes of individuals who live at or below the poverty line. This is one of the most persuasive reasons in favor of enacting the minimum wage law. Several political leaders and officials feel that the minimum wage is a very effective strategy in the battle against severe poverty (Sotomayor, 2021).

2. Methodology

I. Data Set

From 1968 to 2020, this dataset contains statistics on the minimum wage established by each state and the federal government (The data was scraped from https://www.kaggle.com/lislejoem/us-minimum-wage-by-state-from-1968-to-2017 depicting the United States Department of Labor's minimum wage by state). However, after extensive cleaning, the data set included in this study consisted of 41 states spanning 2010 to 2020, with 1271 observations. The univariate time series data comprise a time series composed of a single (scalar) observation gathered sequentially throughout equal time increments. The following time series data analysis techniques were used with the data:

II. ACF and PACF plots

After tautomerizing the time series data, the next step is to determine whether any AR or MA terms are required to correct any autocorrelation that remains in the differenced series after being stationarized before moving on to the next step of the fitting process. Using the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the differenced series, it is possible to make an educated guess as to how many AR and/or MA terms will be required in the differenced series by examining the autocorrelation function and partial autocorrelation function plots of the differenced series. Using the ACF plot and the closely related PACF plot, it is sometimes possible to identify the ideal values for p and q in certain situations.

III. Unit Root Test

If the series has a unit root, it does not move into the deterministic route over the long term. A shocking series that occurs during the present period has long-lasting consequences on the long-term values of the series. According to Rath & Akram (2021), this is critical for modeling the time series (p. 60). The ADF, PP, and KPSS tests are often used to determine whether or not a time series is stationary or nonstationary.

a) Augmented Dickey–Fuller (ADF) Test

The ADF test is the most accurate method of detecting whether or not a time series is stationary. Whenever the test statistic is smaller than the critical value, the null hypothesis is rejected, and the time series is interpreted to be stationary.

b) Kwiatkowski–Phillips–Schmidt–Shin (KPSS)

After this test, the observed series will be cleansed of its deterministic tendency and become stationary due to its purification. Unlike the null hypothesis, which proves that the series is not stationary, the alternative hypothesis demonstrates that the series is stationary but not stationary in all directions.

IV. Time Series Models

I. ARIMA modeling

The notation ARIMA (p, d, q) denotes an ARIMA model in general. The letters p, d, and q signify the orders of auto-regression, integration (difference), and moving average, respectively.

II. Simple exponential smoothing (Holt)

Single Exponential Smoothing (SES) is a time series forecasting approach for univariate data without a trend or seasonality used for univariate data.

III. Holt's linear trend method

Using Holt's linear trend approach, an extension of exponential smoothing, it is possible to deal with trending data more effectively.

3. Results

There are two variables: State Minimum Wages, the actual State's minimum wage on January 1st of the year in question, and the State Minimum Wages in 2020 (Dollars in). Before moving on to time series analysis, let's take a closer look at some of the fundamental characteristics of these two variables.

As of January 1st of the year in question, the minimum state pay is 1.600, with a mean of 5.977 and a maximum of 14.00, with5.150 being the median (Table 1). However, in 2020, the minimum state wage was 2.34, with a mean of 7.768 and a high of 14.16, with a median of 7.80. The fundamental knowledge about the variables is intended to prepare the researcher for a more in-depth examination of time series plots. Except for Georgia, Kansas, and Wyoming, a growing tendency can be seen in most states (Figure 1 & 2). Similarly, for minimum wages through 2020, most states are showing linear increases. In contrast, Georgia, Texas, and Wyoming show declining trends, and Indiana, Indiana, Kentucky, North Carolina, and Wisconsin remain stagnant (Figure 3 & 4). Figure 5 depicts a comparison visualization of the trends in both variables, which clearly shows their relationship.

Note: Both line graphs and bar charts have been given to look deeply into data with ease.

I. Time series plots

The time series plots are typical in public health, education policy, and social services research. They are used to examine the impact of an interruption on a result. When a new health policy goes into force, the disruption may be intentional; but when an environmental exposure occurs accidentally. Therefore, delving deeply into time series plots is helpful for future evaluations that influence policy choices. Both variables' time series charts are shown in Figure 6. There is no evidence of a trend or seasonal trends, in either case, proving that time series modeling fulfills the one-basic requirement.

II. ACF and PACF

Before beginning time series modeling, it is necessary to examine the ACF and PACF of both of the variables. Since these plots exhibit very slight variations, we do not need to differentiate the variables, confirming that the data set is stationary (Figure 7 & 8). In addition, partial ACF has been provided side by side.

III. Augmented Dickey–Fuller and Kwiatkowski–Phillips–Schmidt–Shin Test statistics

Scientists have developed several empirical tests for determining the stationarity of time series data sets. These tests include the Augmented Dickey–Fuller (ADF) Test and the Kwiatkowski–Phillips–Schmidt–Shin Test (KPSS). A significant test is necessary for the data to be declared stationary in the case of ADF. In contrast, a non-significant test is required for the data to be stated stationary in the case of KPSS. According to the results of this research, the Dickey-Fuller test has a p-value of 0.01, and so the null hypothesis is rejected while the alternative hypothesis that the data has reached a point of steady-state is accepted. For example, in contrast to the null hypothesis, which reveals that the series is non-stationary, the alternative hypothesis demonstrates that the series is stationary, so a non-significant test will validate the stationarity of the data set when using the KPSS test method. In this research, the KPSS tests were non-significant, confirming the critical necessity of time series data before advancing to the modeling stage.

IV. Histogram and Normal Q-Q Plot

Outliers are unusually high or low values in a dataset that misinterprets the findings or distorts actual results. Finding outliers depends on the in-depth subject-knowledge of a researcher as there is no thumb rule of what method/test needs to be used to identify outliers. All these methods have advantages and disadvantages for finding unusual values compared to the rest of the dataset.

A QQ-plot and histograms seem better for looking at normality. When the data do not have an outlier, the Q-Q plot lies approximately on a straight line. The in-hand data is free from any type of outlier. As evident from the data displayed in Figures 9 & 10, the extreme valuation theorem does not hold in this case. It is possible to see the non-stationarity of the variables even more clearly by looking at the histogram and standard Q-Q plots of the variables (Figure 9 & 10).

V. Time Series Predictive Modelling

Time Series is a technique for evaluating time series data and obtaining statistical information and characteristics relevant to a specific group of persons. It is one of the essential objectives of the research to forecast the asset's future value under consideration.

VI. Model Diagnostic Statistics: State Minimum Wages

AIC and BIC are two of the most often utilized model selection criteria. AIC is an abbreviation for Akaike's Information Criteria, while BIC is an abbreviation for Bayesian Information Criteria. Even while BIC is more sensitive than AIC, it becomes less tolerant when the numbers are raised to a higher degree of sensitivity. Comparatively speaking, the BIC penalizes free parameters significantly more significant than the AIC. How much smoothing is used for the level is decided by the value of the parameter referred to as 'alpha.' The abbreviation for AICC is “Information score of the model” (the lower-case 'c' indicates that the value was derived using the AIC test with adjustments for small sample sizes), expressed as a percentage of total information. We'll use AIC to compare all of the models. In this example, the AIC for simple exponential smoothing forecasting of State Minimum Wages is 9630.121 (Table 4). Figure 11 shows the related prediction plot. Holt's approach is a helpful extension of exponential smoothing that may be used to cope with trending data. The AIC in this situation is 9634.304. (Table 5). Figure 12 shows the related prediction plot. In terms of the ARIMA model, the AIC value is 3992.41. (Table 6). Figure 13 shows the corresponding prediction plot. Based on aic criterion and measuring errors, we may infer that the ARIMA model has a higher perdition potential than other models. Its more detailed foresting chart also confirms this.

VII. Model Diagnostic statistics: State MinimumWages-2020

To compare all the models, here too, we'll utilize AIC. The AIC for simple exponential smoothing forecasting of State Minimum Wages in this scenario is 8750.595. (Table 7). The corresponding prediction plot is shown in Figure 14. Holt's method is an extension of exponential smoothing that may be used to deal with trending data. In this case, the AIC is 8755.318. (Table 8). The corresponding prediction plot is shown in Figure 15. The AIC value for the ARIMA model is 3157.18. (See Table 9) The corresponding prediction plot is shown in Figure 16. We deduce that the ARIMA model has a more significant perdition potential than other models based on AIC criteria and measuring mistakes. Its more accurate foresting chart backs this up.

4. CONCLUSION

We investigated three-time series models for two variables, state minimum wages and state minimum wages-2020 (ARIMA, Holt's Method, and Simple Exponential Method). In comparison to other models, the ARIMA model suited the data well.

LIST OF TABLES

Table 1 Some basic Statics of the variables

```
 State.Min.Wage    State.Min.Wage.2020
 Min.   : 1.600    Min.   : 2.340
 1st Qu.: 4.250    1st Qu.: 7.050
 Median : 5.150    Median : 7.800
 Mean   : 5.977    Mean   : 7.768
 3rd Qu.: 7.250    3rd Qu.: 8.600
 Max.   :14.000    Max.   :14.160
```

Table 2 ADF and KPSS of Minimum state wages

```
> adf

        Augmented Dickey-Fuller Test

data:  State.Min.Wage
Dickey-Fuller = -11.533, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

> kpss

        KPSS Test for Level Stationarity

data:  State.Min.Wage
KPSS Level = 0.056142, Truncation lag parameter = 7, p-value = 0.1
```

Table 3 ADF and KPSS Minimum state wages-2020

```
> adf

        Augmented Dickey-Fuller Test

data:  State.Min.Wage.2020
Dickey-Fuller = -7.5266, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

> kpss

        KPSS Test for Level Stationarity

data:  State.Min.Wage.2020
KPSS Level = 0.19828, Truncation lag parameter = 7, p-value = 0.1
```

Table 4 Output by Simple Exponential Method: State Minimum Wages

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = mwage$State.Min.Wage, h = 50)

  Smoothing parameters:
    alpha = 0.9624

  Initial states:
    l = 3.8741

  sigma:  1.2374

     AIC     AICc      BIC
9630.121 9630.140 9645.563

Error measures:
                      ME     RMSE       MAE       MPE     MAPE     MASE        ACF1
Training set 0.001043092 1.236473 0.3945067 -3.510031 9.681802 1.013807 0.003967548
```

Table 5 Output by Holt's Method: State Minimum Wages

```
Forecasts:

Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = mwage$State.Min.Wage, h = 50)

  Smoothing parameters:
    alpha = 0.9636
    beta  = 1e-04

  Initial states:
    l = 3.5887
    b = 0.001

  sigma:  1.2385

     AIC     AICc      BIC
9634.304 9634.351 9660.041
```

Table 6 Output by ARIMA's Method: State Minimum Wages

```
Forecasts:
Series: mwage$State.Min.Wage
ARIMA(3,0,2) with non-zero mean

Coefficients:
         ar1     ar2      ar3     ma1      ma2    mean
      0.7968  0.8768  -0.7396  0.0817  -0.8935  5.9791
s.e.  0.0282  0.0230   0.0235  0.0257   0.0250  0.0929

sigma^2 estimated as 1.344:  log likelihood=-1989.21
AIC=3992.41   AICc=3992.5   BIC=4028.45

Training set error measures:
                       ME     RMSE       MAE       MPE    MAPE     MASE       ACF1
Training set 0.0007386431 1.156591 0.5867064 -4.967018 12.8956 1.507724 -0.0137811
```

Table 7 Output by Simple Exponential Method: State Minimum Wages-2020

```
Forecasts:

Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = mwage$State.Min.Wage.2020, h = 50)

  Smoothing parameters:
    alpha = 0.9451

  Initial states:
    l = 7.6506

  sigma:  0.8755

     AIC     AICc      BIC
8750.595 8750.614 8766.038

Error measures:
                       ME      RMSE       MAE        MPE     MAPE     MASE        ACF1
Training set -0.002078634 0.8748225 0.4319005 -0.8467628 5.948704 1.002859 0.008138103
```

Table 8 Output by Holt's Method: State Minimum Wages-2020

```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = mwage$State.Min.Wage.2020, h = 50)

  Smoothing parameters:
    alpha = 0.9451
    beta  = 1e-04

  Initial states:
    l = 8.3287
    b = -0.0023

  sigma:  0.8765

     AIC     AICc      BIC
8755.318 8755.366 8781.056

Error measures:
                        ME      RMSE      MAE        MPE     MAPE     MASE        ACF1
Training set -0.0002892966 0.8750714 0.431416 -0.8218996 5.939393 1.001734 0.007770453
```

Table 9 Output by ARIMA's Method: State Minimum Wages-2020

```
Forecasts:
Series: mwage$State.Min.Wage.2020
ARIMA(2,0,1) with non-zero mean

Coefficients:
          ar1     ar2    ma1    mean
      -0.1118  0.7791  0.975  7.7573
s.e.   0.0493  0.0461  0.035  0.1385

sigma^2 estimated as 0.6985:  log likelihood=-1574.09
AIC=3158.17   AICc=3158.22   BIC=3183.91

Training set error measures:
                       ME      RMSE       MAE       MPE     MAPE     MASE       ACF1
Training set 0.0001356481 0.8344191 0.4762708 -1.472612 6.877775 1.105885 0.02068917
```

LIST OF FIGURES

Figure 1 Line Graph of State Minimum Wages

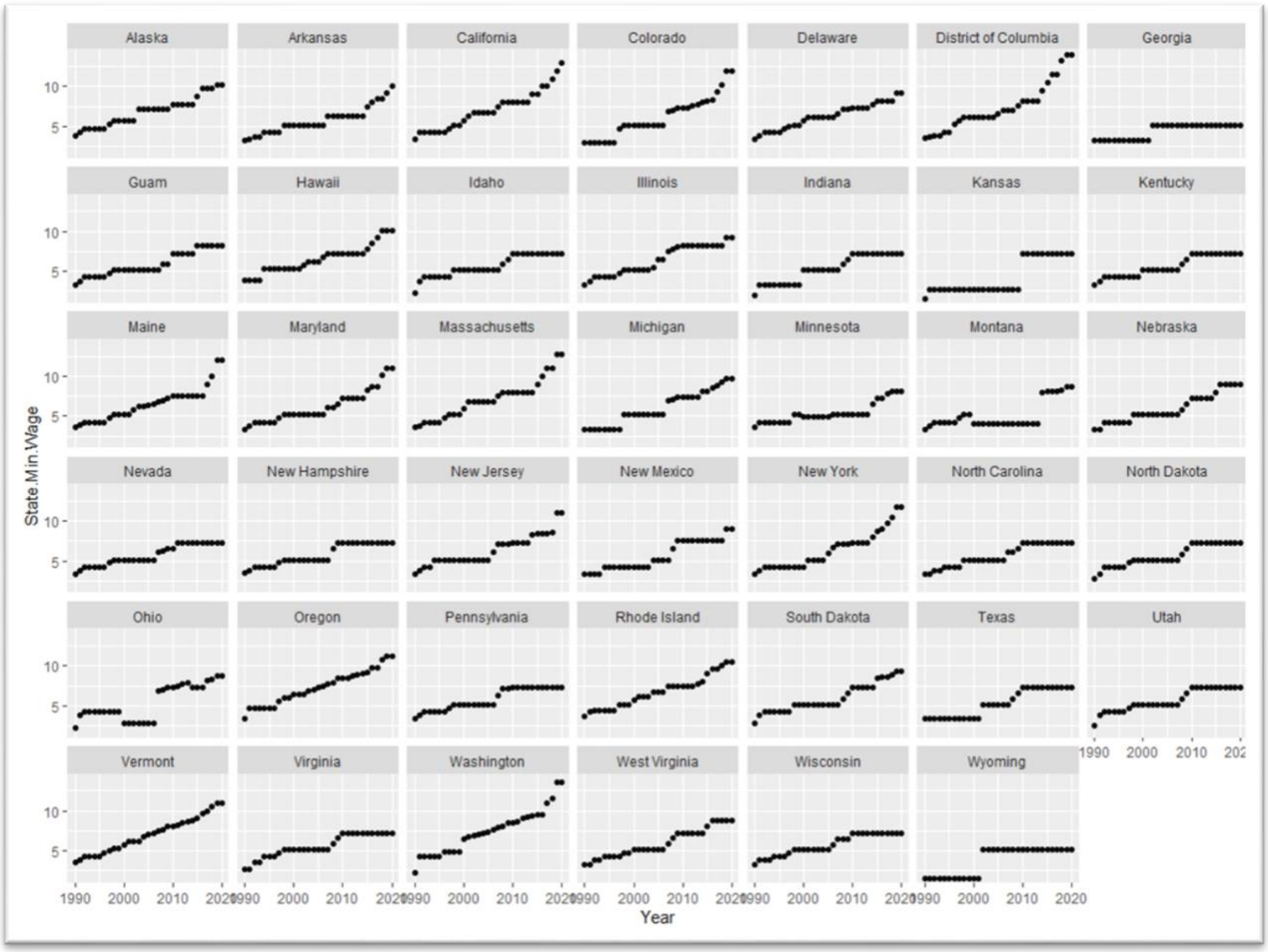


Figure 2 Bar Graph of State Minimum Wages

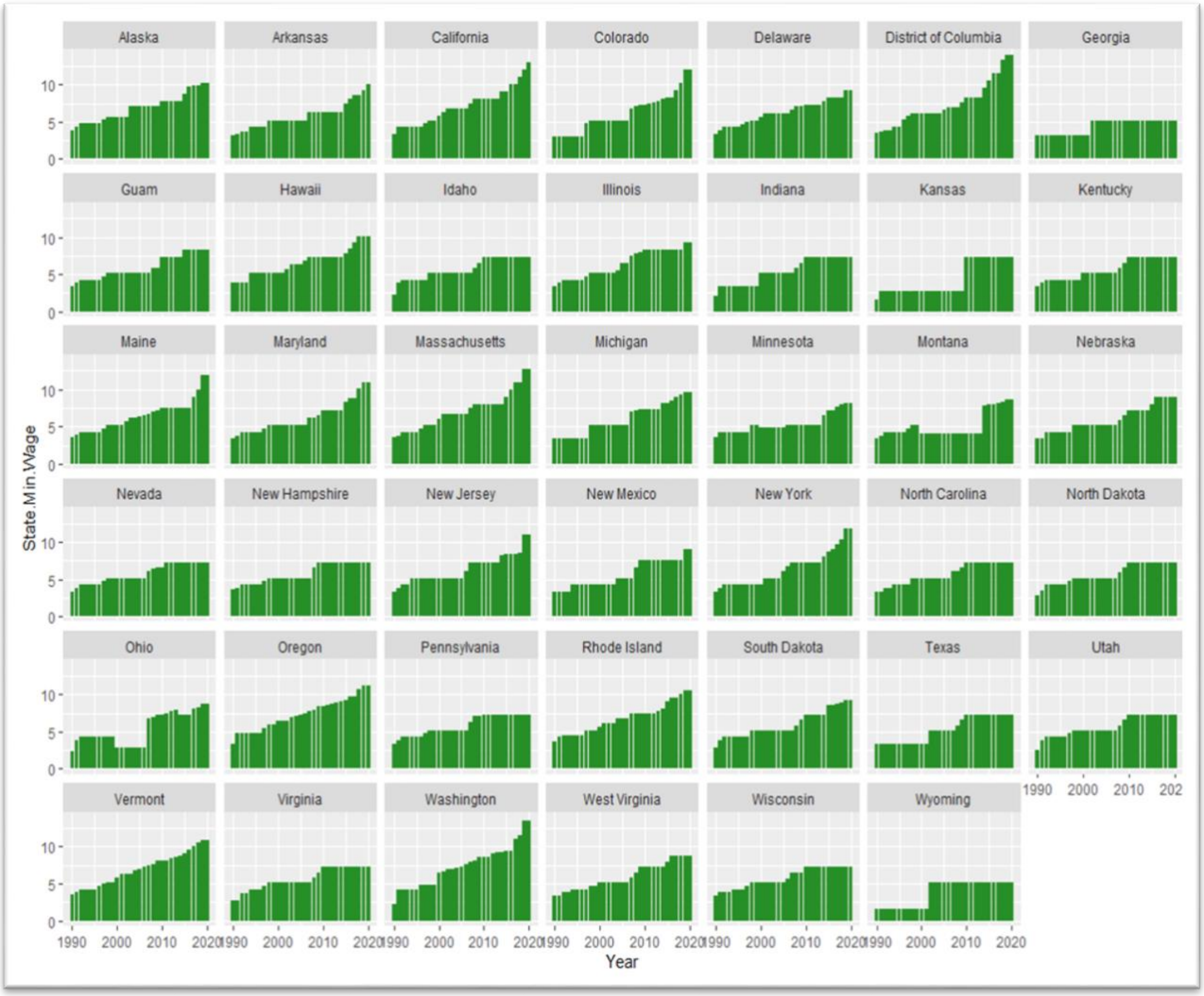


Figure 3 Line graph State Minimum Wages-2020

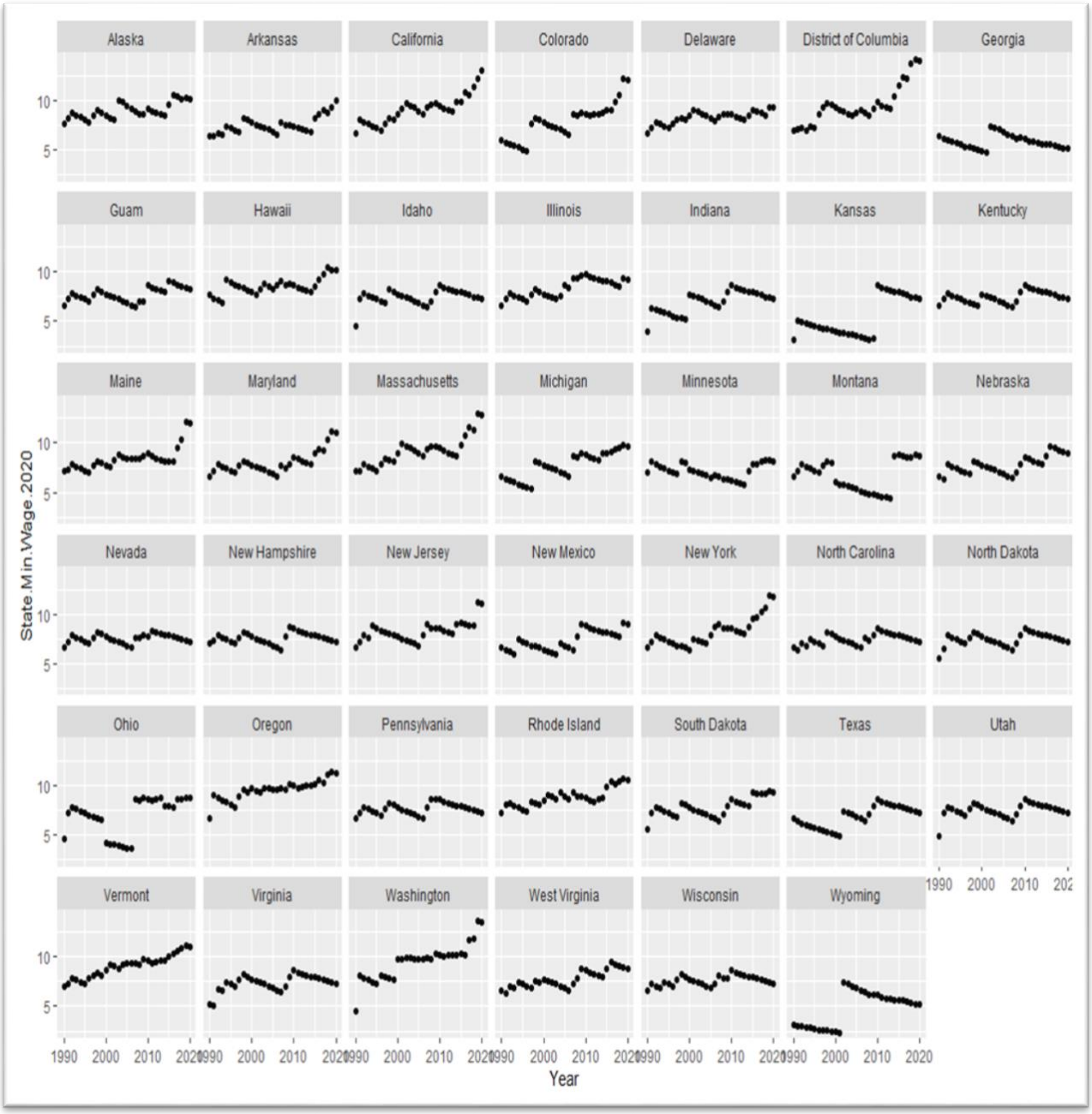


Figure 4 Bar graph State Minimum Wages-2020

Figure 5 Line Plot of Mean (State. Min. Wage vs. State. Min. Wage. 2020)

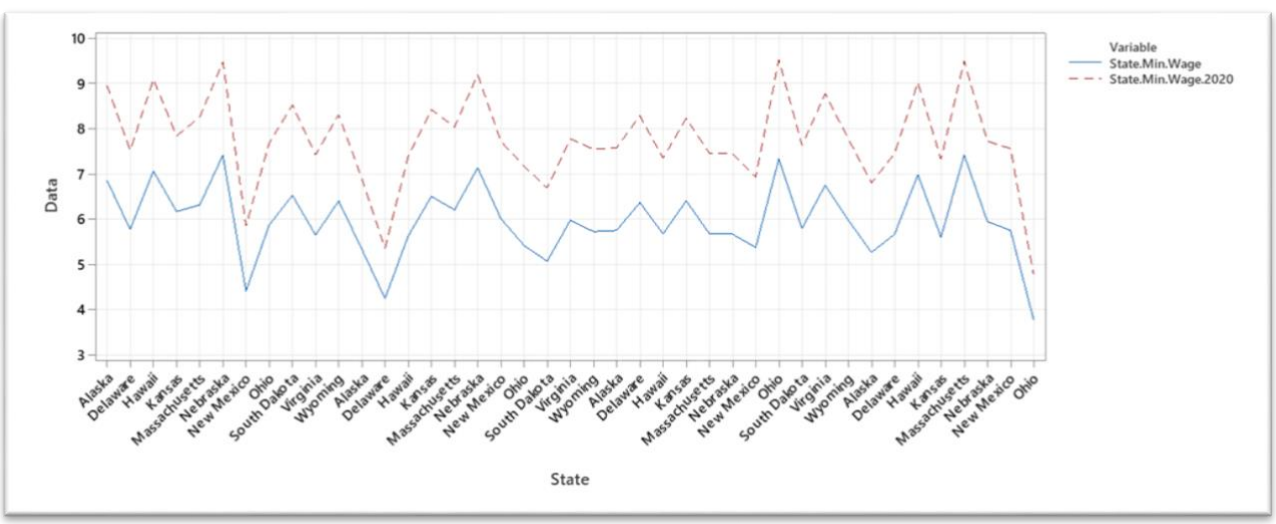


Figure 6 Time Series plots of State minimum Wages and State minimum Wages-2020

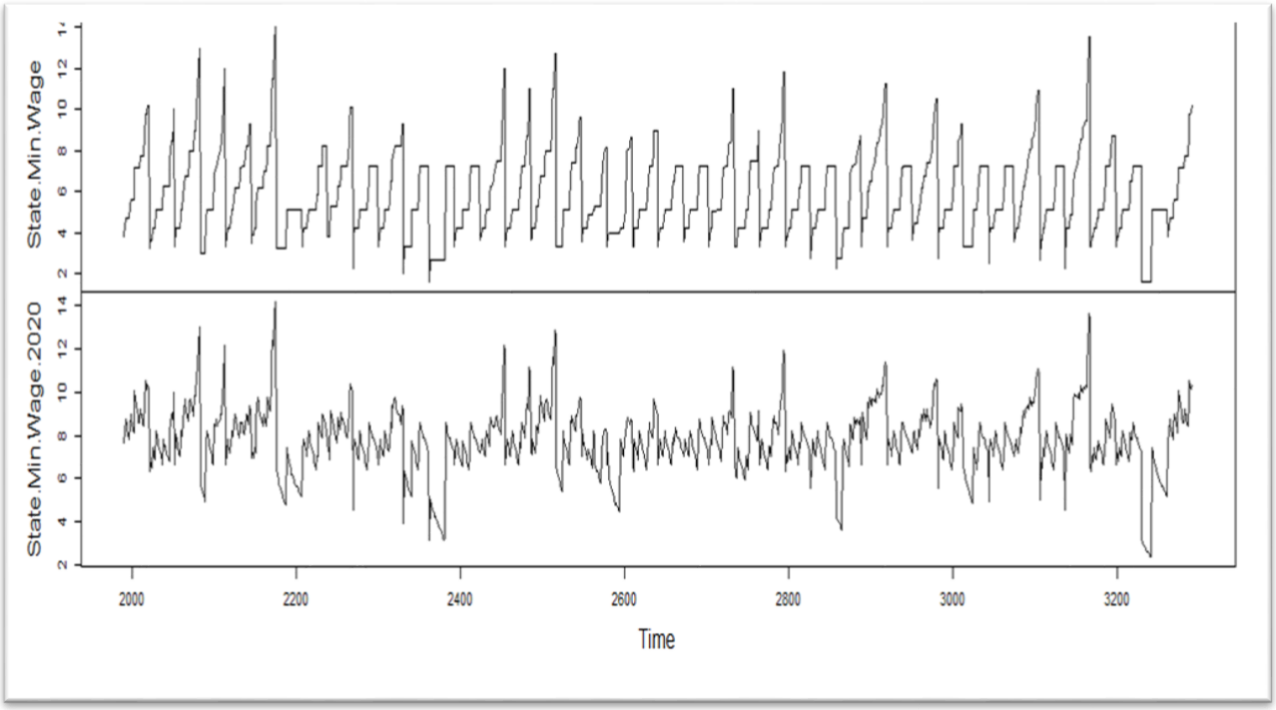


Figure 7 ACF and Partial ACF of State Minimum Wages

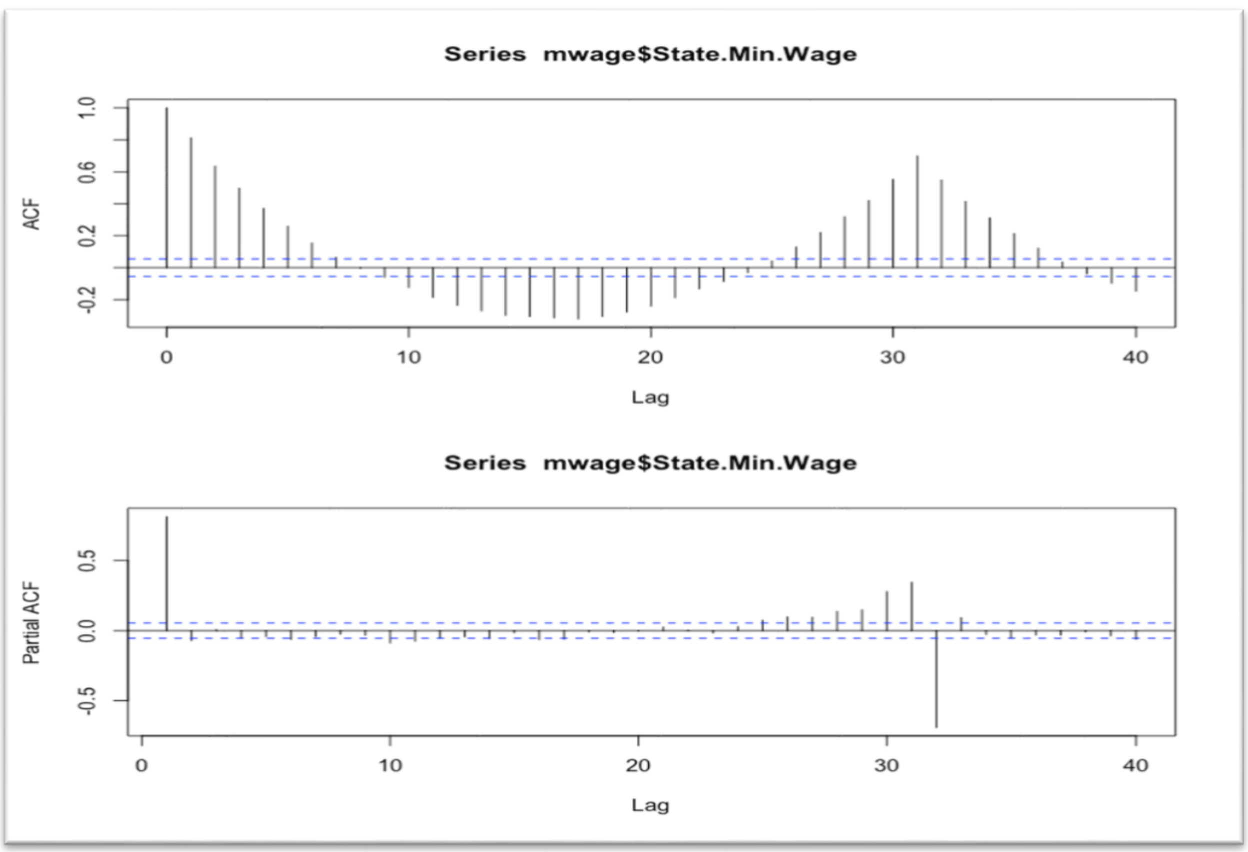


Figure 8 ACF and Partial ACF of State Minimum Wages-2020

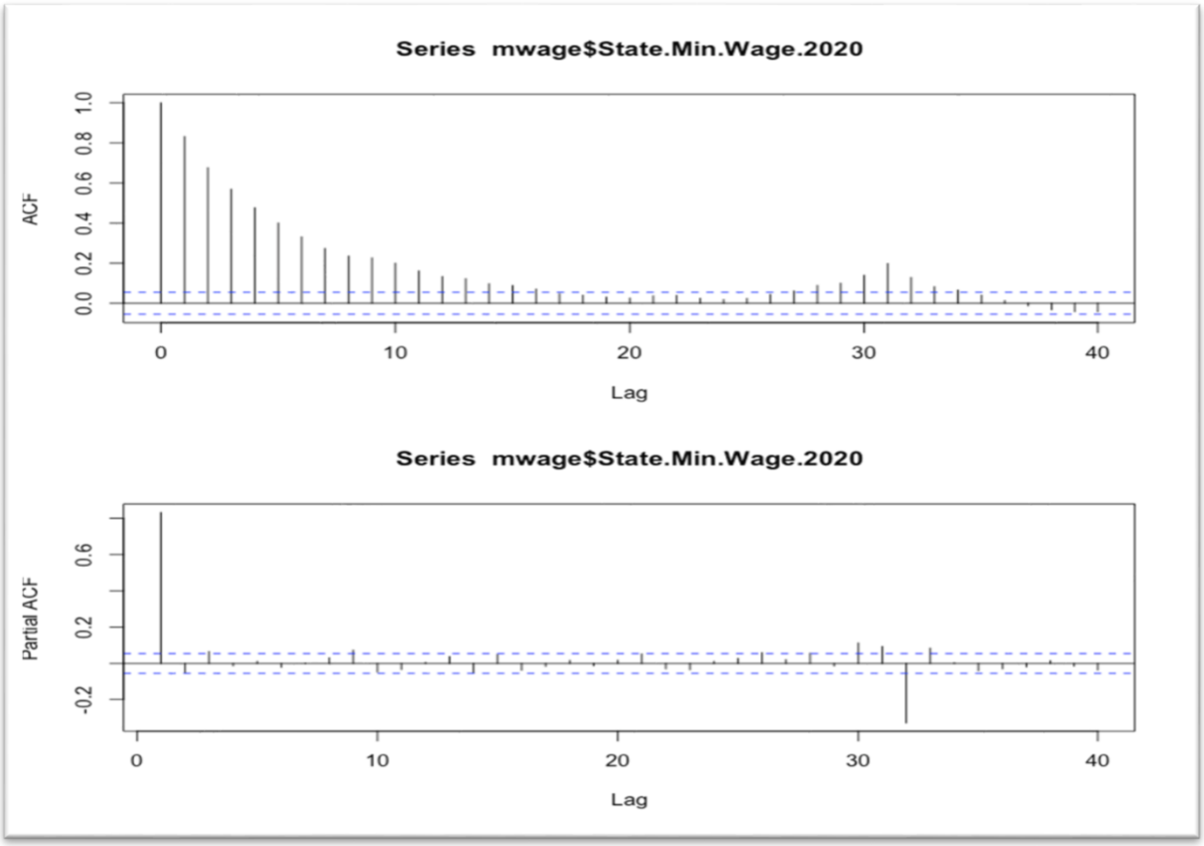


Figure 9 Histogram and Normal Q-Q Plots of State Minim Wages

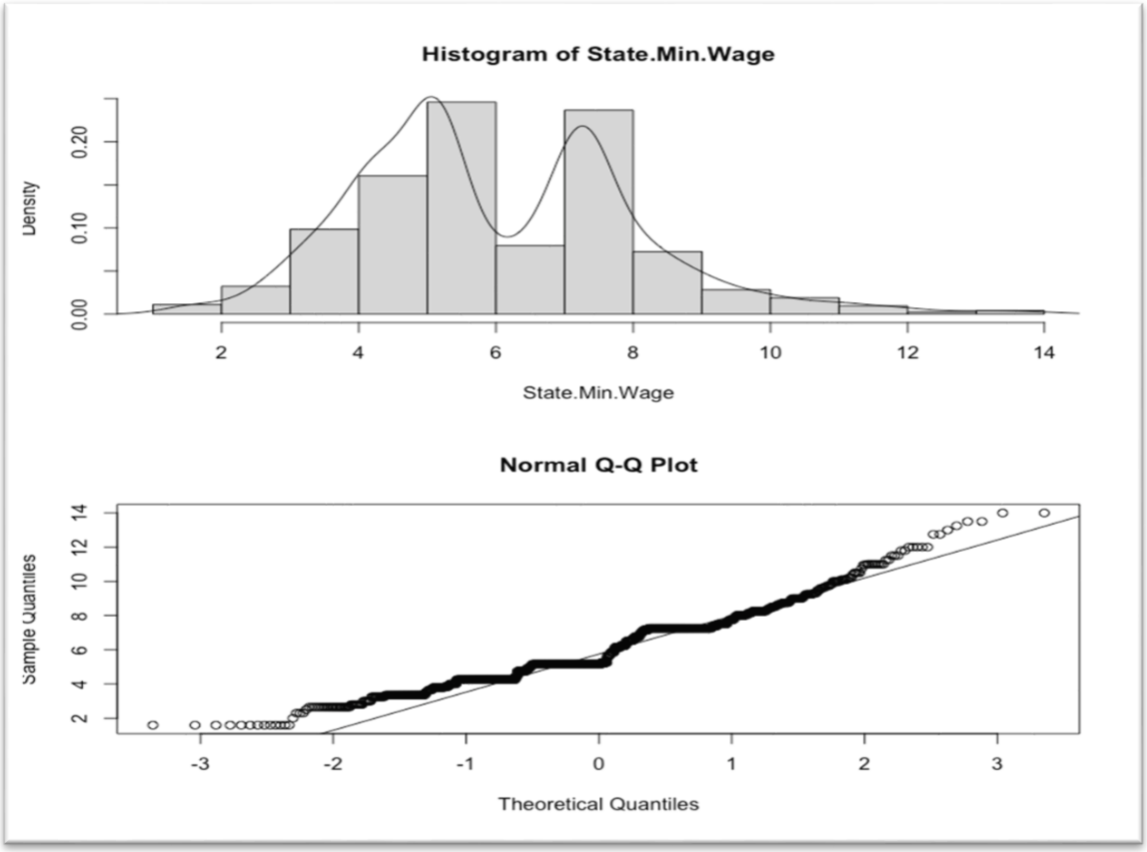


Figure 10 Histogram and Normal Q-Q Plots of State Minim Wages

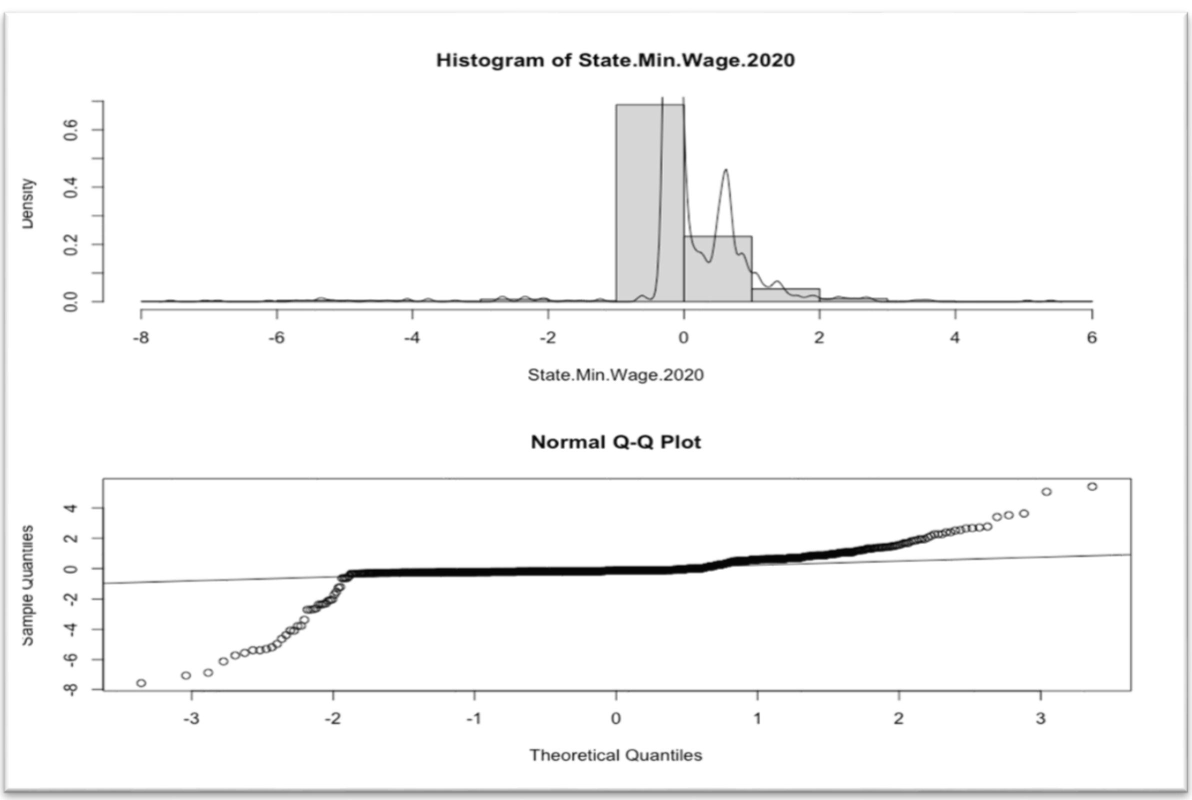


Figure 11 Forecasting by Simple Exponential Method: State Minimum Wages

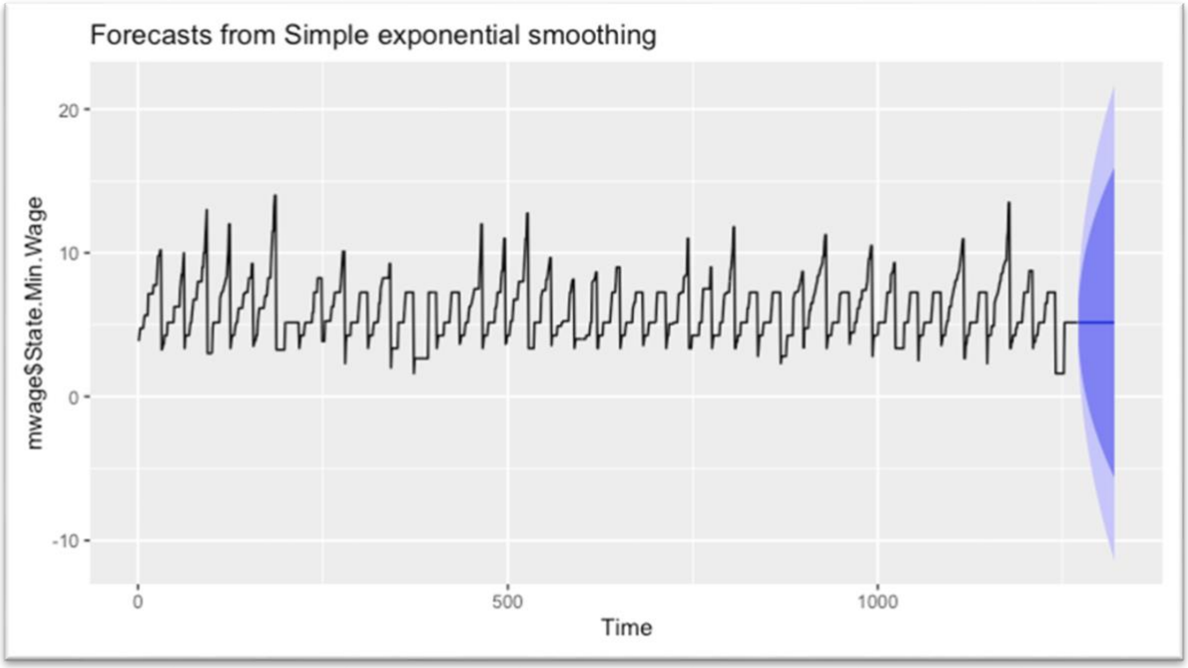


Figure 12 Forecasting by Holt's Method: State Minimum Wages

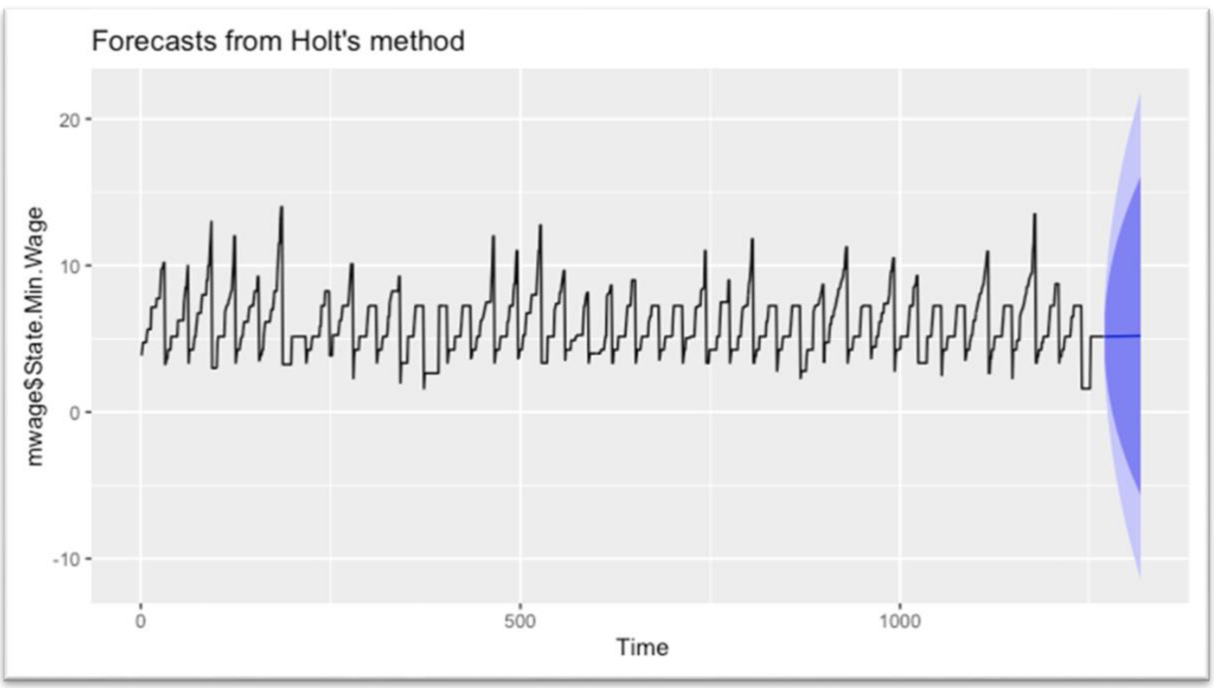


Figure 13 Forecasting by ARIMA's Method: State Minimum Wages

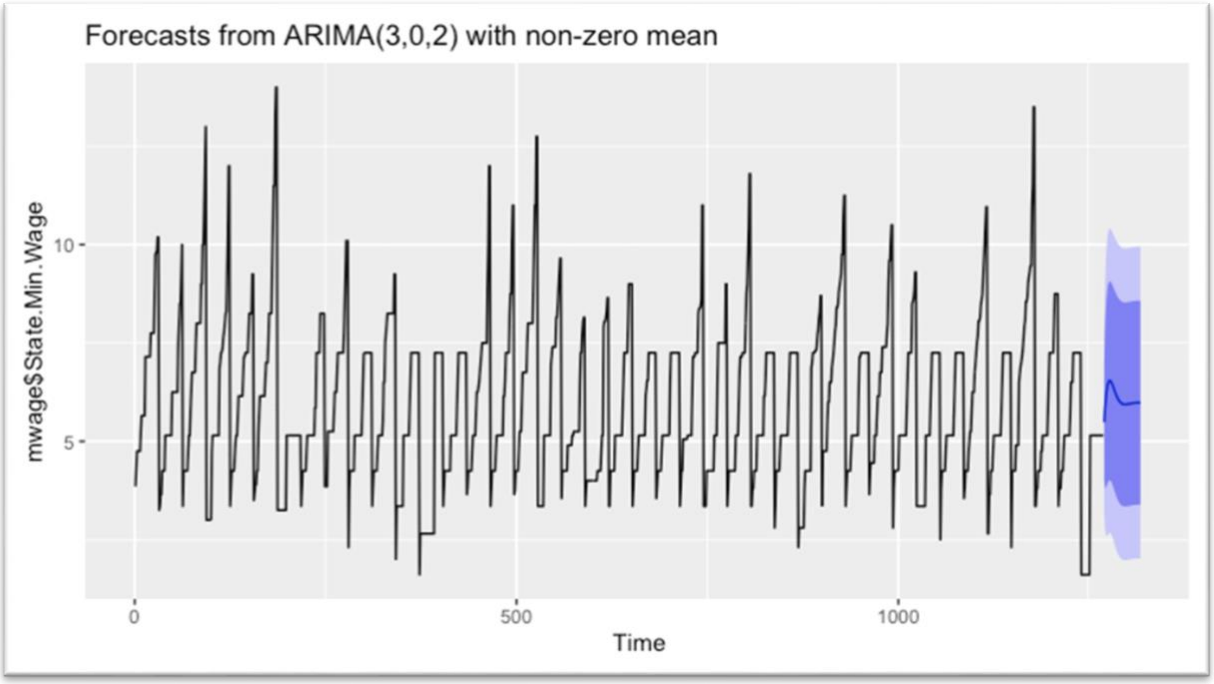


Figure 14 Forecasting by Simple Exponential Method: State Minimum Wages-2020

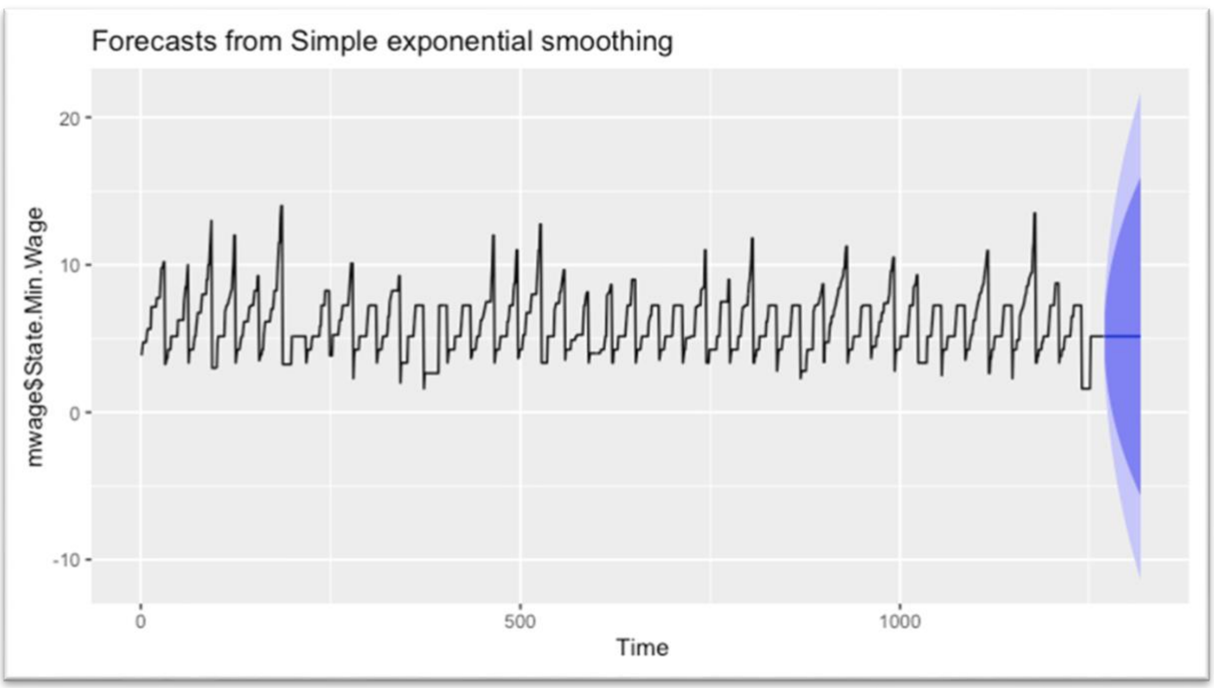


Figure 15 Forecasting by Holt's Method: State Minimum Wages-2020

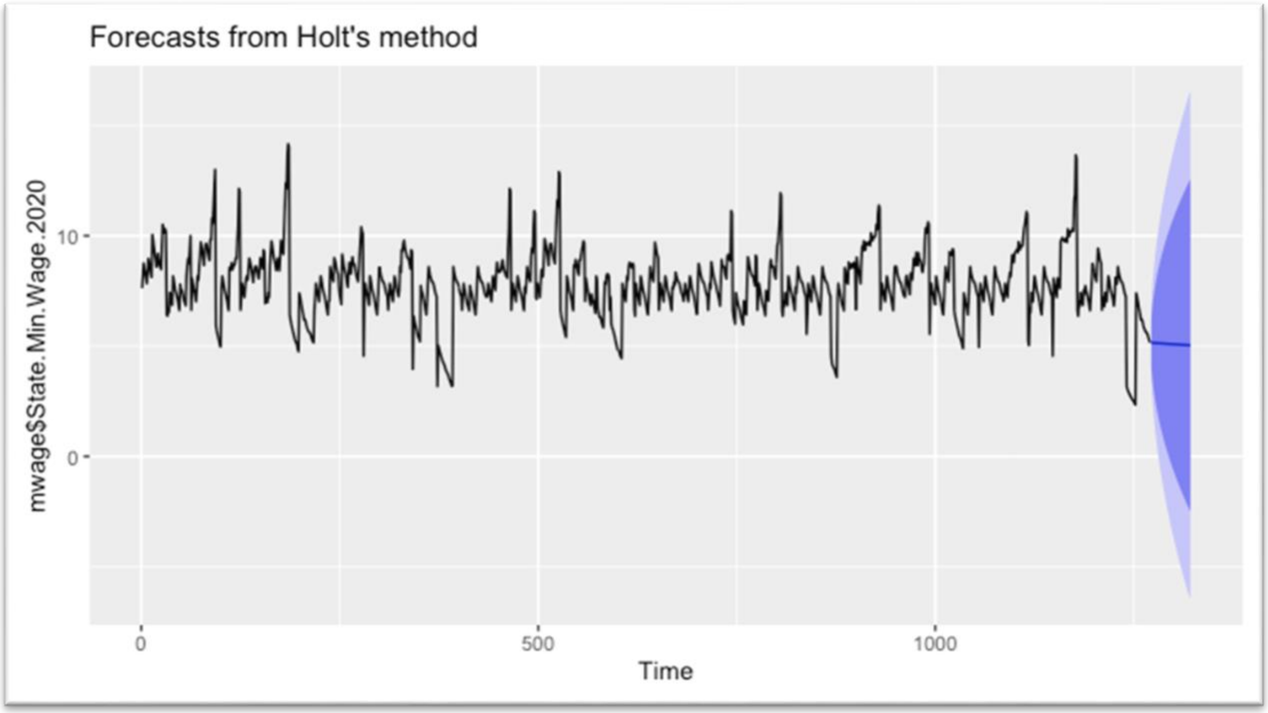
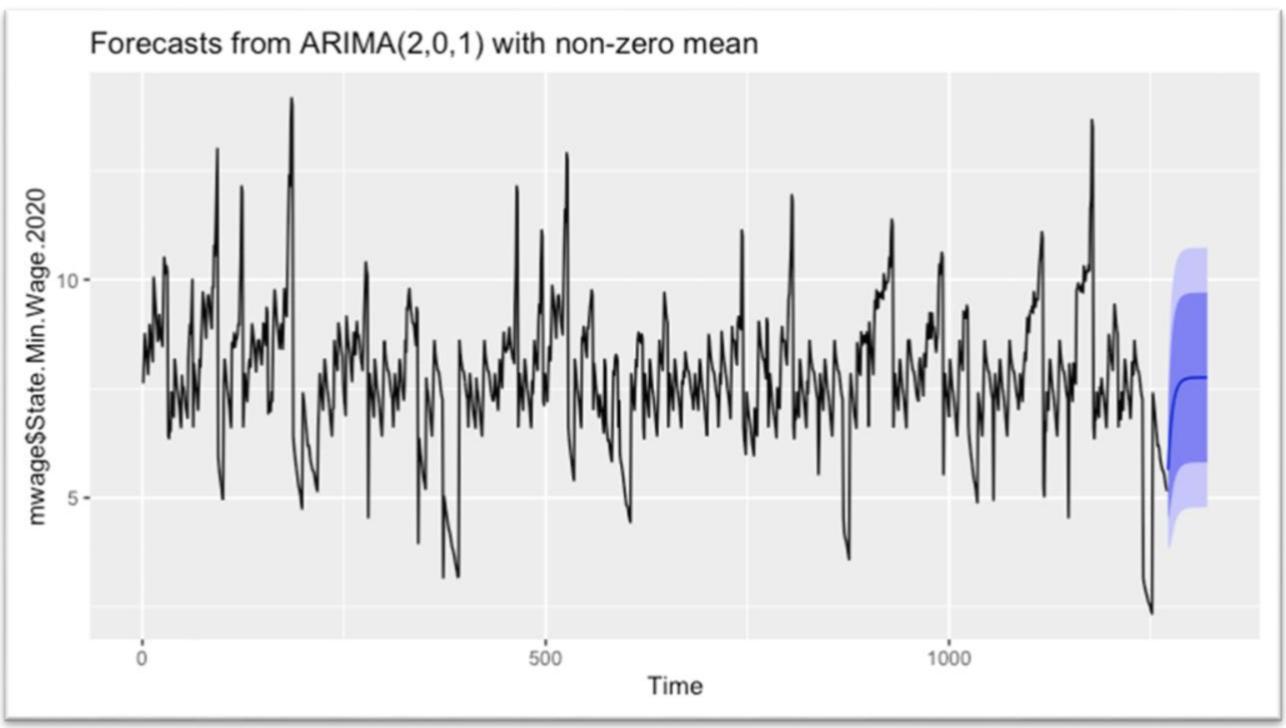


Figure 16 Forecasting by ARIMA's Method: State Minimum Wages-2020



5. REFERENCES

1) Kuroki, M. (2021). State minimum wages and health insurance coverage in the United States: 2008–2018. International Journal of Health Economics and Management, 1-18.
2) Leigh, J. P., Du Leigh, W. A., & J. (2019). Minimum wages and public health: A literature review. Preventive Medicine, 118, 122–134.
3) Rath, B. N., & Akram, V. (2021). The popularity of Unit Root Tests: A Review. Asian Economics Letters, 2(4), 24141.

title: "Group-10-Project" output: pdf_document: default html_document: default

word_document: default

```{r setup, include=FALSE}

# install.packages("readxl")

library(readxl)

# libraries

library(tidyquant) library(tseries) library(forecast) library(FitAR) library(tidyverse) library(TSstudio) library(forecast) library(dplyr) library(colortools) library(rmeta) library(eurostat) # package used for querrying EUROSTAT library(dplyr) # package used for data-manipulation library(ggplot2) # package used for plotting install.packages("fpp2") library(fpp2) # package used for forecasting library(xts) library(astsa) library(lubridate) library(timetk) library(readxl) library(tidyquant) library(scales) library(sweep) # Broom tidiers for forecast pkg library(broom) library(tibble) library(stringr) library(ggthemes) library(tseries) library(forecast) library(plm) library(modelr) library(gridExtra) library(grid) library(foreign) library (nlme) library(tseries) library(lmtest) library(ggthemes) library(viridis)

knitr::opts_chunk$set(cache = TRUE)

```
Importing data

```{r loading data, include=FALSE}
## load data
Wages<- read.csv("FFF_Data.csv")
View(Wages)
str(Wages)
names(Wages)
attach(Wages)
```

```{r} TS <- ts(Wages, start=c(1990, 1), end=c(2020, 1271), frequency=1) plot(TS)

summary(Wages) summary(Wages$State.Min.Wage,Wages$State.Min.Wage.2020)

summary(State.Min.Wage) summary(State.Min.Wage.2020) par(mfrow=c(2,1)) # set up the graphics acf(Wages$State.Min.Wage, lag.max = 40) pacf(Wages$State.Min.Wage, lag.max = 40)

acf(Wages$State.Min.Wage.2020, lag.max = 40) pacf(Wages$State.Min.Wage.2020, lag.max = 40)

acf(Wages) pacf(Wages)

lambda <- BoxCox.lambda(Wages$State.Min.Wage,lower=0) lambda lambda1 <- BoxCox.lambda(Wages$State.Min.Wage.2020,lower=0) lambda1

tsdata2 = BoxCox(Wages$State.Min.Wage, lambda=lambda) tsdata3 = BoxCox(Wages$State.Min.Wage.2020, lambda=lambda)
```

```{R}
adf = adf.test(State.Min.Wage)
kpss = kpss.test(State.Min.Wage)
adf
kpss

adf = adf.test(State.Min.Wage.2020)
kpss = kpss.test(State.Min.Wage.2020)
adf
kpss

adf.test(diff(log(Wages$State.Min.Wage)), alternative="stationary", k=0)
adf.test(diff(log(Wages$State.Min.Wage.2020)), alternative="stationary", k=0)
adf.test(diff(log(Wages$State.Min.Wage.2020)), alternative="stationary", k=5)
```

```{r} par(mfrow=c(2,1)) # set up the graphics for State.Min.wage hist(State.Min.Wage, prob=TRUE, 12) # histogram lines(density(State.Min.Wage)) # smooth it - ?density for details qqnorm(State.Min.Wage) # normal Q-Q plot qqline(State.Min.Wage)

par(mfrow=c(2,1)) # set up the graphics for State.Min.Wage.2020 hist(State.Min.Wage.2020, prob=TRUE, 12) # histogram lines(density(State.Min.Wage.2020)) # smooth it - ?density for details qqnorm(State.Min.Wage.2020) # normal Q-Q plot qqline(State.Min.Wage.2020)

```
```{r}
par(mfrow=c(2,1))        # set up the graphics
Min.Wage = diff(State.Min.Wage, differences = 1)
plot.ts(Min.Wage)
State.Min.Wage.2020 = diff(State.Min.Wage.2020, differences = 1)
plot.ts(State.Min.Wage.2020)
# plot  data, with a smoothed trend line
par(mfrow=c(2,1))
ggplot(Wages) + geom_line(mapping = aes(x=ï..Year,y=State.Min.Wage),color="red") +
  geom_smooth(mapping=aes(x=ï..Year,y=State.Min.Wage),color="blue")+
  labs(title = "Time series plot of State minimum wages",
       subtitle="visualization in R",
       x = "ï..Year",
       y= "State.Min.Wage")
base <- ggplot(Wages, aes(x = ï..Year)) +
  geom_line(aes(y = State.Min.Wage), color = "blue") +
  labs(x = "Size", y = "Survival")

base + geom_point(aes(y = State.Min.Wage), alpha = 0.2)
base + stat_summary_bin(geom = "point", fun = mean, aes(y = State.Min.Wage))

# plot  with a smoothed trend line
ggplot(Wages) + geom_line(mapping = aes(x=ï..Year,y=State.Min.Wage.2020),color="black") +
  geom_smooth(mapping=aes(x=ï..Year,y=State.Min.Wage.2020),color="green")+
  labs(title = "Time series plot of State.Min.Wage.2020",
       subtitle="visualization in R",
       x = "ï..Year",
       y= "State.Min.Wage.2020")

# declare the data as time series object ; this will ease forecasting_State.Min.Wage
timeseries_df <- ts(Wages$State.Min.Wage,
                    start=c(1990,1),
                    frequency = 1) #ts declares something as a time series
```

```{r}

# seasonal naive forecasting model

fit <- snaive(State.Min.Wage) # model calibration fit1 <- snaive(State.Min.Wage.2020) # model calibration fit fit1 Box.test (State.Min.Wage, lag = 2, type = "Ljung") Box.test (State.Min.Wage.2020, lag = 2, type = "Ljung")
```

exponential smoothing forecasting model

fit_ets <- ets(timeseries_df) # model calibration checkresiduals(fit_ets) # visualize model performance checkresiduals(fit) # visualize model performance

```
```{r}
# ARIMA forecasting model
fit_arima <- auto.arima(timeseries_df,
                        d=1,
                        D=1,
                        stepwise=FALSE,
                        approximation = FALSE,
                        trace=TRUE) # model calibration

checkresiduals(fit_arima) # visualize model performance

# let us forecast using the ETS model; it outperforms ARIMA
FC <- forecast(fit_ets,
               h=60)

autoplot(FC) +
  labs(title="Forecast",
       subtitle="",
       x = "ï..Year",
       y = "SMin.Wage")
FC
acf(Wages$State.Min.Wage, lag.max=20)
acf(Wages$State.Min.Wage, lag.max=10)
diff1 <- diff(State.Min.Wage, differences=1)
plot.ts(diff1)
pacf(diff1, lag.max=40) # plot a partial correlogram
diff2 <- diff(State.Min.Wage.2020, differences=1)
plot.ts(diff2)
pacf(diff2, lag.max=40) # plot a partial correlogram
pacf(diff1, lag.max=30) # plot a partial correlogram
train_series=State.Min.Wage[1:200]
test_series=State.Min.Wage[291:1271]
```

```{r}

## make arima models

arimaModel_1=arima(train_series, order=c(0,1,2)) arimaModel_2=arima(train_series, order=c(1,1,0)) arimaModel_3=arima(train_series, order=c(1,1,2)) print(arimaModel_1);print(arimaModel_2);print(arimaModel_3)

forecast1=predict(arimaModel_1, 50) forecast2=predict(arimaModel_2, 50) forecast3=predict(arimaModel_3, 50)

forecast1 forecast2 forecast3

AutoArimaModel=auto.arima(train_series) AutoArimaModel autoplot(forecast(AutoArimaModel))

lambda <- BoxCox.lambda(Wages$State.Min.Wage) lambda

```{r}
#  Predictions-Forecasts---State.Min.Wage--State.Min.Wage--State.Min.Wage

####  Naive Forecasting Method
naive_mod <- naive(Wages$State.Min.Wage, h = 50)
summary(naive_mod)

fcast <- forecast(naive_mod, h=50)
autoplot(fcast)
###   Simple Exponential Smoothing
se_model <- ses(Wages$State.Min.Wage, h = 50)
summary(se_model)
fcast1 <- forecast(se_model, h=50)
autoplot(fcast1)

#######   Holt's Trend Method
holt_model <- holt(Wages$State.Min.Wage, h = 50)
summary(holt_model)

fcast2 <- forecast(holt_model, h=50)
autoplot(fcast2)

####   arima  Method
arima_model <- auto.arima(Wages$State.Min.Wage)
summary(arima_model)
fcast3 <- forecast(arima_model, h=50)
fcast3
autoplot(fcast3)

#### linear Regression
# Fit linear regression
fit_lr = lm(as.vector(State.Min.Wage) ~ ï..Year)
fit_lr
par(mfrow = c(3, 2))
plot(fit, ask=FALSE)
plot(fit_lr)
```

```{r}

# Predictions-Forecasts—State.Min.Wage.2020–State.Min.Wage.2020

#### Naive Forecasting Method

naive_mod <- naive(Wages$State.Min.Wage.2020, h = 50) summary(naive_mod)

fcast4 <- forecast(naive_mod, h=50) autoplot(fcast4)

# Simple Exponential Smoothing

se_model <- ses(Wages$State.Min.Wage.2020, h = 50) summary(se_model) fcast5 <- forecast(se_model, h=50) autoplot(fcast5)

## Holt's Trend Method

holt_model <- holt(Wages$State.Min.Wage.2020, h = 50) summary(holt_model)

fcast6 <- forecast(holt_model, h=50) autoplot(fcast6)

#### arima Method

arima_model <- auto.arima(Wages$State.Min.Wage.2020) summary(arima_model) fcast7 <- forecast(arima_model, h=50) fcast7 autoplot(fcast7)

```

```
---
title: "Group-10-Project"
output:
  pdf_document: default
  html_document: default
  word_document:  default
---

```{r setup, include=FALSE}
library(readxl)
# libraries
library(tidyquant)
library(tseries)
library(forecast)
library(FitAR)
library(tidyverse)
library(TSstudio)
library(forecast)
library(dplyr)
library(colortools)
library(rmeta)
library(eurostat) # package used for querrying EUROSTAT
library(dplyr) # package used for data-manipulation
library(ggplot2) # package used for plotting
install.packages("fpp2")
library(fpp2) # package used for forecasting
library(xts)
library(astsa)
library(lubridate)
library(timetk)
library(readxl)
library(tidyquant)
library(scales)
library(sweep)    # Broom tidiers for forecast pkg
library(broom)
library(tibble)
library(stringr)
library(ggthemes)
library(tseries)
library(forecast)
library(plm)
library(modelr)
library(gridExtra)
library(grid)
library(foreign)
library (nlme)
library(tseries)
library(lmtest)
library(ggthemes)
library(viridis)
```
```

```
knitr::opts_chunk$set(cache = TRUE)
```

```



## Importing data

```{r loading data, include=FALSE}
## load data
Wages<- read.csv("FFF_Data.csv")
View(Wages)
str(Wages)
names(Wages)
attach(Wages)








```



```{r}
TS <- ts(Wages, start=c(1990, 1), end=c(2020, 1271), frequency=1)
plot(TS)

summary(Wages)
summary(Wages$State.Min.Wage,Wages$State.Min.Wage.2020)

summary(State.Min.Wage)
summary(State.Min.Wage.2020)
par(mfrow=c(2,1))         # set up the graphics
acf(Wages$State.Min.Wage, lag.max = 40)
pacf(Wages$State.Min.Wage, lag.max =  40)

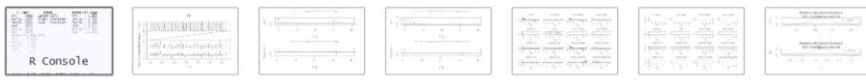
acf(Wages$State.Min.Wage.2020, lag.max = 40)
pacf(Wages$State.Min.Wage.2020, lag.max =  40)


acf(Wages)
pacf(Wages)

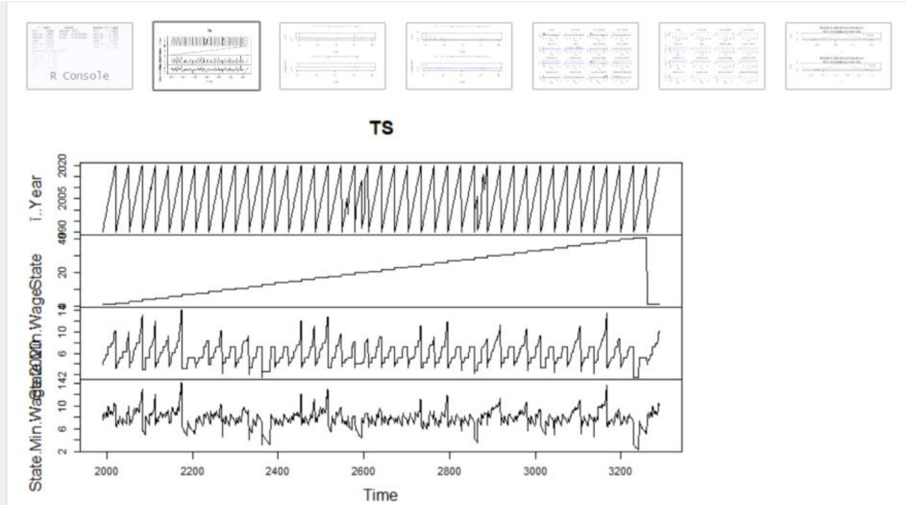

lambda <- BoxCox.lambda(Wages$State.Min.Wage,lower=0)
lambda
lambda1 <- BoxCox.lambda(Wages$State.Min.Wage.2020,lower=0)
lambda1

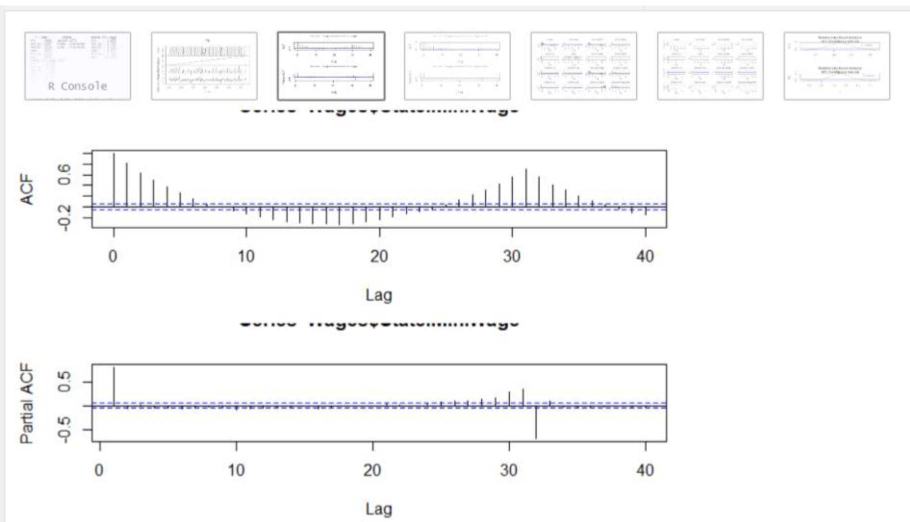
tsdata2 = BoxCox(Wages$State.Min.Wage, lambda=lambda)
tsdata3 = BoxCox(Wages$State.Min.Wage.2020, lambda=lambda)
```
```

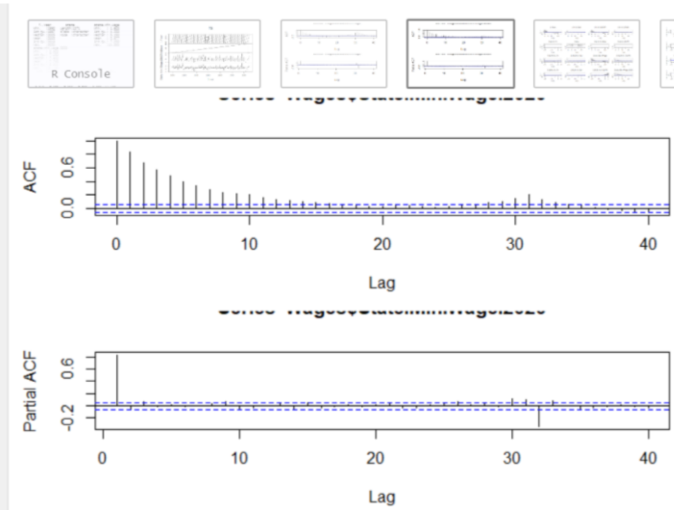
` ` `

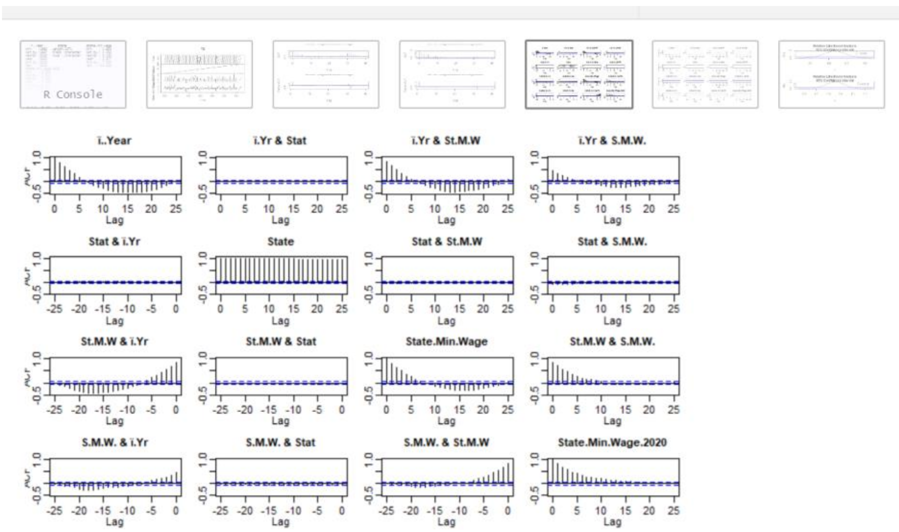


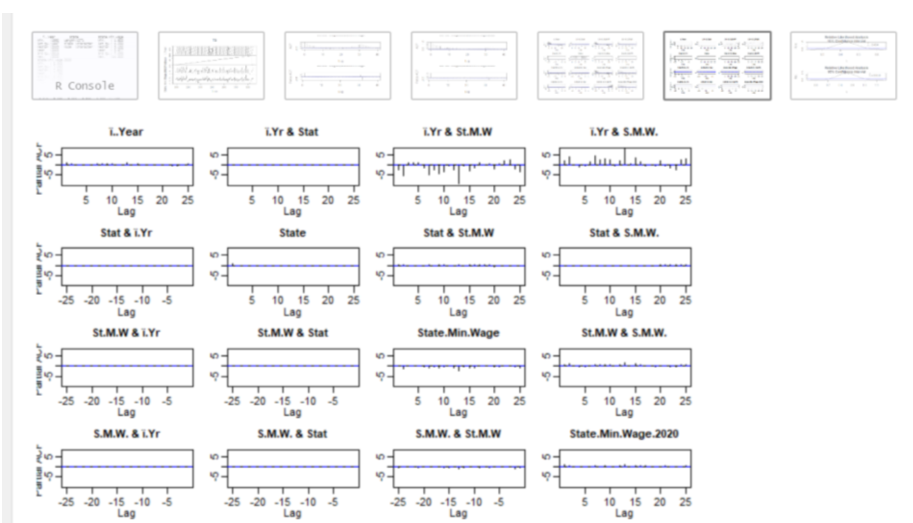

```
     i..Year          State              State.Min.Wage
 Min.   :1990   Length:1271        Min.   : 1.600
 1st Qu.:1997   Class :character   1st Qu.: 4.250
 Median :2005   Mode  :character   Median : 5.150
 Mean   :2005                      Mean   : 5.977
 3rd Qu.:2013                      3rd Qu.: 7.250
 Max.   :2020                      Max.   :14.000
 State.Min.Wage.2020
 Min.   : 2.340
 1st Qu.: 7.050
 Median : 7.800
 Mean   : 7.768
 3rd Qu.: 8.600
 Max.   :14.160
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.600   4.250   5.150   5.977   7.250  14.000
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.600   4.250   5.150   5.977   7.250  14.000
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.340   7.050   7.800   7.768   8.600  14.160
Warning in abbreviate(snames) : abbreviate used with non-ASCII chars
Warning in abbreviate(snames) : abbreviate used with non-ASCII chars
[1] 4.102259e-05
[1] 0.865341
```

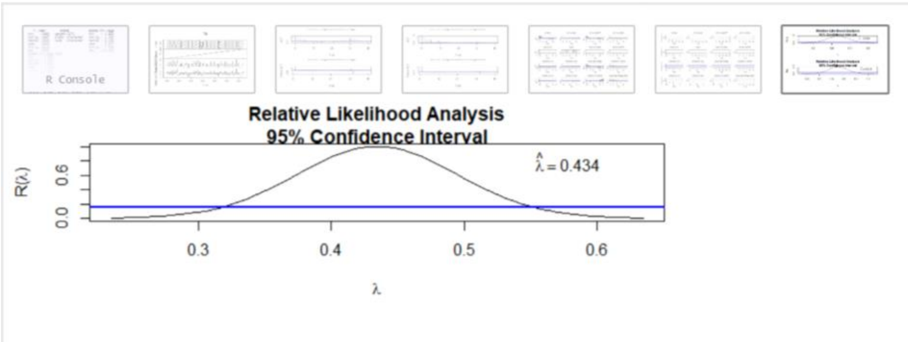





R Console
ACF
Lag
Partial ACF
Lag


R Console
i..Year
i.Yr & Stat
i.Yr & St.M.W
i.Yr & S.M.W.
Stat & i.Yr
State
Stat & St.M.W
Stat & S.M.W.
St.M.W & i.Yr
St.M.W & Stat
State.Min.Wage
St.M.W & S.M.W.
S.M.W. & i.Yr
S.M.W. & Stat
S.M.W. & St.M.W
State.Min.Wage.2020
Lag


R Console
i..Year
i.Yr & Stat
i.Yr & St.M.W
i.Yr & S.M.W.
Stat & i.Yr
State
Stat & St.M.W
Stat & S.M.W.
St.M.W & i.Yr
St.M.W & Stat
State.Min.Wage
St.M.W & S.M.W.
S.M.W. & i.Yr
S.M.W. & Stat
S.M.W. & St.M.W
State.Min.Wage.2020
Lag




```{R}
adf = adf.test(State.Min.Wage)
kpss = kpss.test(State.Min.Wage)
adf
kpss

adf = adf.test(State.Min.Wage.2020)
kpss = kpss.test(State.Min.Wage.2020)
adf
kpss

adf.test(diff(log(Wages$State.Min.Wage)), alternative="stationary", k=0)
adf.test(diff(log(Wages$State.Min.Wage.2020)), alternative="stationary", k=0)
adf.test(diff(log(Wages$State.Min.Wage.2020)), alternative="stationary",  k=5)
```

```
Warning in adf.test(State.Min.Wage) :
  p-value smaller than printed p-value
Warning in kpss.test(State.Min.Wage) :
  p-value greater than printed p-value

        Augmented Dickey-Fuller Test

data:  State.Min.Wage
Dickey-Fuller = -11.533, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary


        KPSS Test for Level Stationarity

data:  State.Min.Wage
KPSS Level = 0.056142, Truncation lag parameter = 7, p-value =
0.1

Warning in adf.test(State.Min.Wage.2020) :
  p-value smaller than printed p-value
Warning in kpss.test(State.Min.Wage.2020) :
  p-value greater than printed p-value

        Augmented Dickey-Fuller Test

data:  State.Min.Wage.2020
Dickey-Fuller = -7.5266, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary


        KPSS Test for Level Stationarity
```

```
data:  State.Min.Wage.2020
KPSS Level = 0.19828, Truncation lag parameter = 7, p-value =
0.1

Warning in adf.test(diff(log(Wages$State.Min.Wage)), alternative = "stationary",  :
  p-value smaller than printed p-value

        Augmented Dickey-Fuller Test

data:  diff(log(Wages$State.Min.Wage))
Dickey-Fuller = -38.895, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary

Warning in adf.test(diff(log(Wages$State.Min.Wage.2020)), alternative = "stationary",  :
  p-value smaller than printed p-value

        Augmented Dickey-Fuller Test

data:  diff(log(Wages$State.Min.Wage.2020))
Dickey-Fuller = -38.517, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary

Warning in adf.test(diff(log(Wages$State.Min.Wage.2020)), alternative = "stationary",  :
  p-value smaller than printed p-value

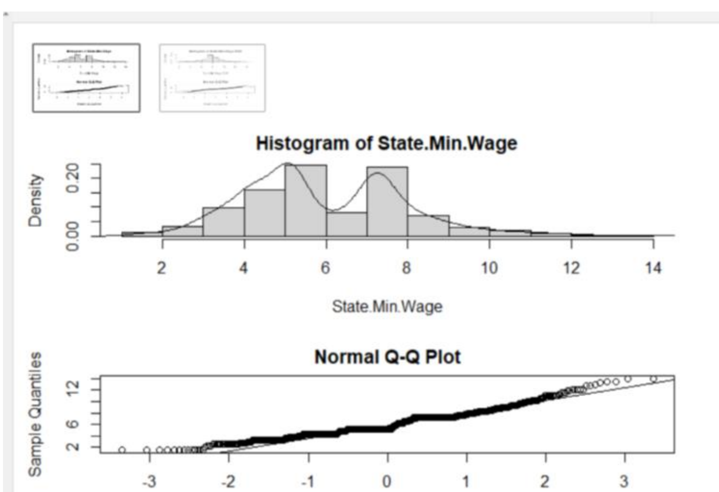
        Augmented Dickey-Fuller Test

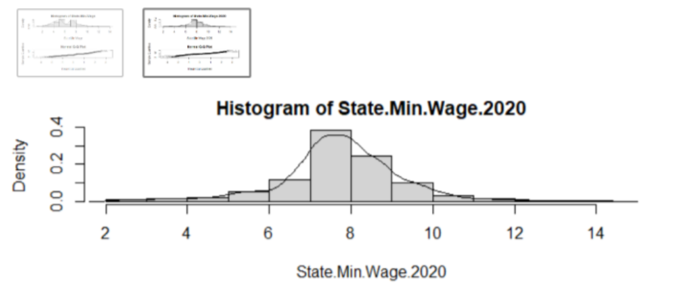
data:  diff(log(Wages$State.Min.Wage.2020))
Dickey-Fuller = -17.143, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

````
```{r}
par(mfrow=c(2,1))          # set up the graphics for State.Min.wage
hist(State.Min.Wage, prob=TRUE, 12) # histogram
lines(density(State.Min.Wage))      # smooth it - ?density for details
qqnorm(State.Min.Wage)              # normal Q-Q plot
qqline(State.Min.Wage)


par(mfrow=c(2,1))          # set up the graphics for State.Min.Wage.2020
hist(State.Min.Wage.2020, prob=TRUE, 12) # histogram
lines(density(State.Min.Wage.2020))      # smooth it - ?density for details
qqnorm(State.Min.Wage.2020)              # normal Q-Q plot
qqline(State.Min.Wage.2020)

```
````





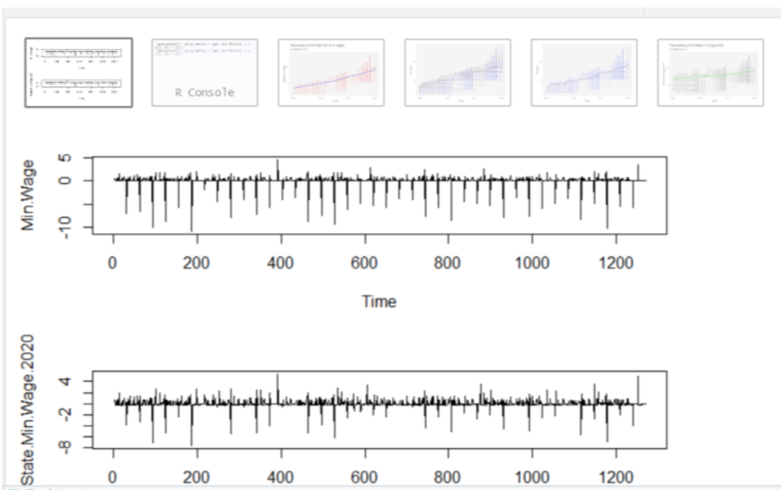

```
```{r}
par(mfrow=c(2,1))         # set up the graphics
Min.Wage = diff(State.Min.Wage, differences = 1)
plot.ts(Min.Wage)
State.Min.Wage.2020 = diff(State.Min.Wage.2020, differences = 1)
plot.ts(State.Min.Wage.2020)
# plot  data, with a smoothed trend line
par(mfrow=c(2,1))
ggplot(Wages) + geom_line(mapping =
aes(x=ï..Year,y=State.Min.Wage),color="red")  +
  geom_smooth(mapping=aes(x=ï..Year,y=State.Min.Wage),color="blue")+
  labs(title = "Time series plot of State minimum wages",
       subtitle="visualization in R",
       x = "ï..Year",
       y= "State.Min.Wage")
base <- ggplot(Wages, aes(x = ï..Year)) +
  geom_line(aes(y = State.Min.Wage), color = "blue") +
  labs(x = "Size", y = "Survival")
```
```

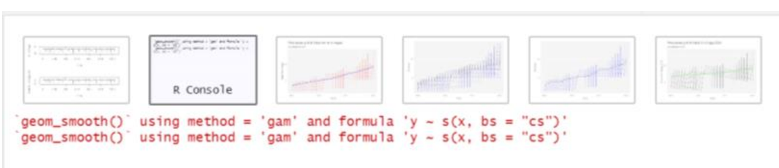
```
base + geom_point(aes(y = State.Min.Wage), alpha = 0.2)
base + stat_summary_bin(geom = "point", fun = mean, aes(y = State.Min.Wage))

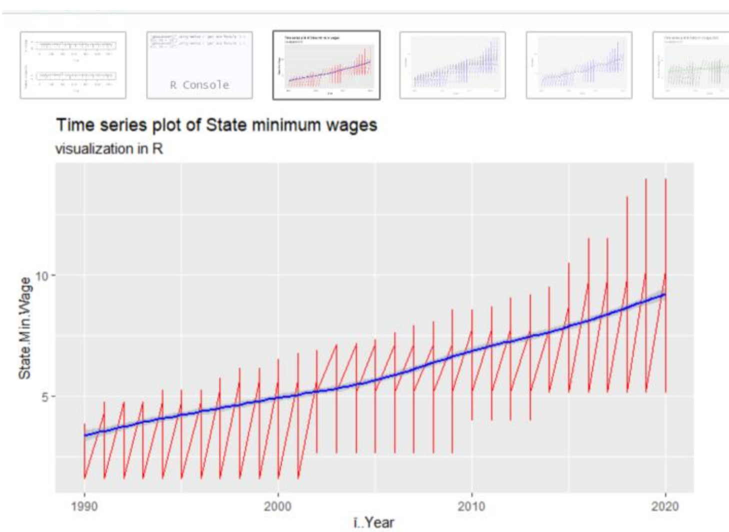
# plot  with a smoothed trend line
ggplot(Wages) + geom_line(mapping = aes(x=ï..Year,y=State.Min.Wage.
2020),color="black") +
  geom_smooth(mapping=aes(x=ï..Year,y=State.Min.Wage.2020),color="green")+
  labs(title = "Time series plot of State.Min.Wage.2020",
       subtitle="visualization in R",
       x = "ï..Year",
       y= "State.Min.Wage.2020")

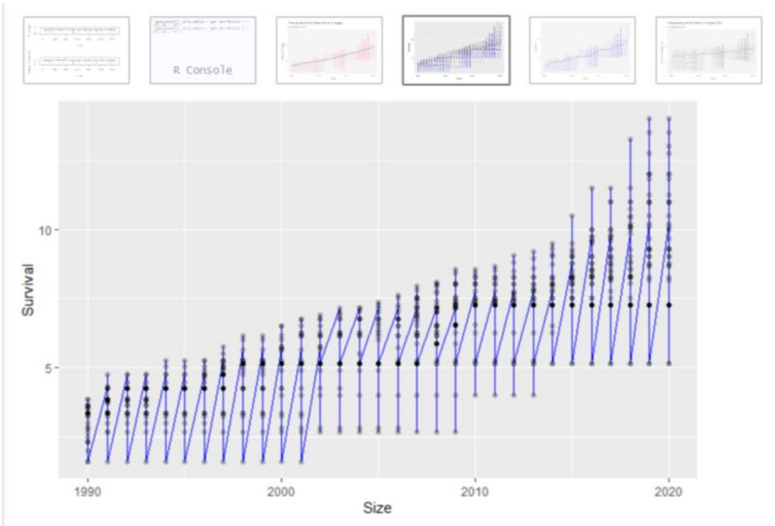
# declare the data as time series object ; this will ease
forecasting_State.Min.Wage
timeseries_df <- ts(Wages$State.Min.Wage,
                    start=c(1990,1),
                    frequency = 1) #ts declares something as a time  series

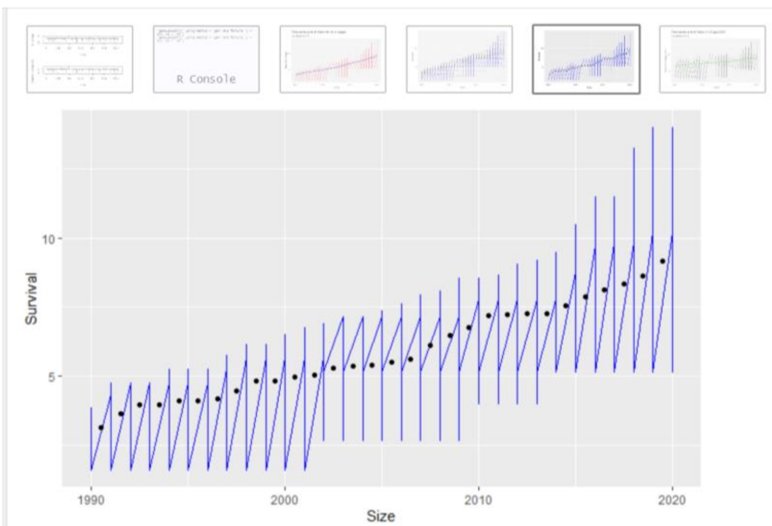
```

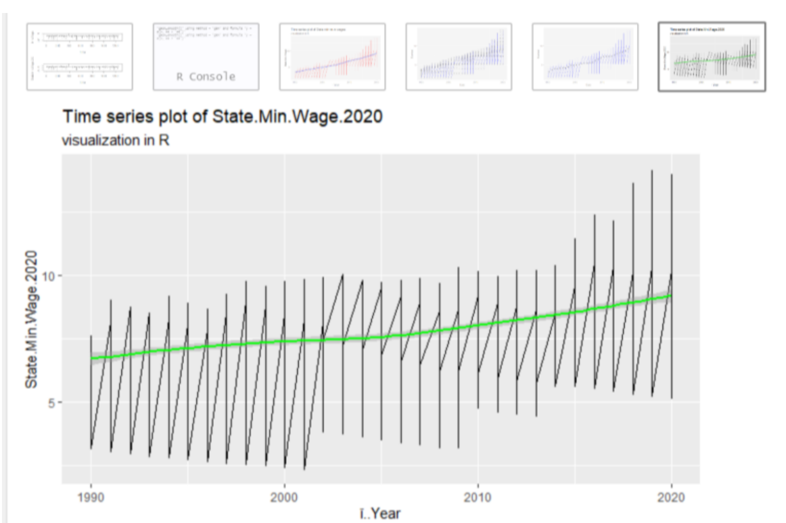













```{r}
# seasonal naive forecasting model
fit <- snaive(State.Min.Wage) # model calibration
fit1 <- snaive(State.Min.Wage.2020) # model calibration
fit
```

```
fit1
Box.test (State.Min.Wage, lag = 2, type = "Ljung")
Box.test (State.Min.Wage.2020, lag = 2, type = "Ljung")

# Exponential smoothing  forecasting  model
fit_ets <- ets(timeseries_df) # model  calibration
checkresiduals(fit_ets) # visualize model performance
checkresiduals(fit) # visualize model  performance

```

data.frame 2 x 5 | data.frame 2 x 5 | R Console

Description: df [2 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.15	3.563886	6.736114	2.724248	7.575752
1273	5.15	2.906896	7.393104	1.719469	8.580531

2 rows

data.frame 2 x 5 | data.frame 2 x 5 | R Console

Description: df [2 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1271	-0.06	-1.678675	1.558675	-2.535550	2.415550
1272	-0.06	-2.349152	2.229152	-3.560956	3.440956

2 rows

data.frame 2 x 5 | data.frame 2 x 5 | R Console

```
        Box-Ljung test

data:  State.Min.Wage
X-squared = 1353.9, df = 2, p-value < 2.2e-16


        Box-Ljung test

data:  State.Min.Wage.2020
X-squared = 28.133, df = 2, p-value = 7.782e-07


        Ljung-Box test

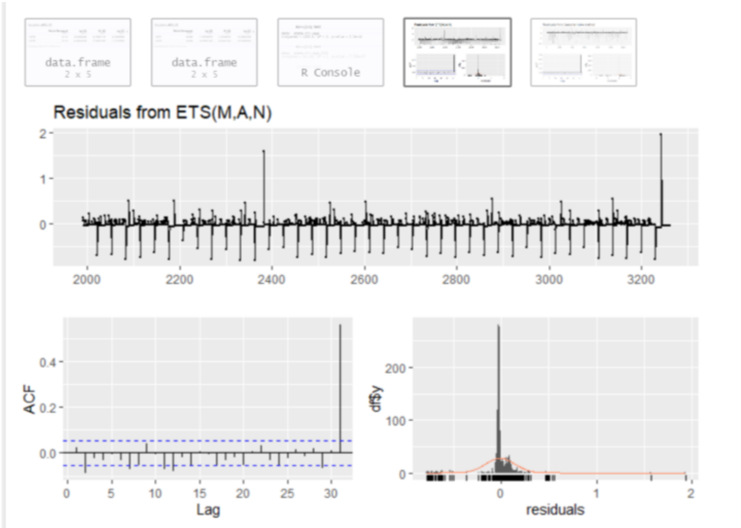
data:  Residuals from ETS(M,A,N)
Q* = 26.36, df = 6, p-value = 0.0001908

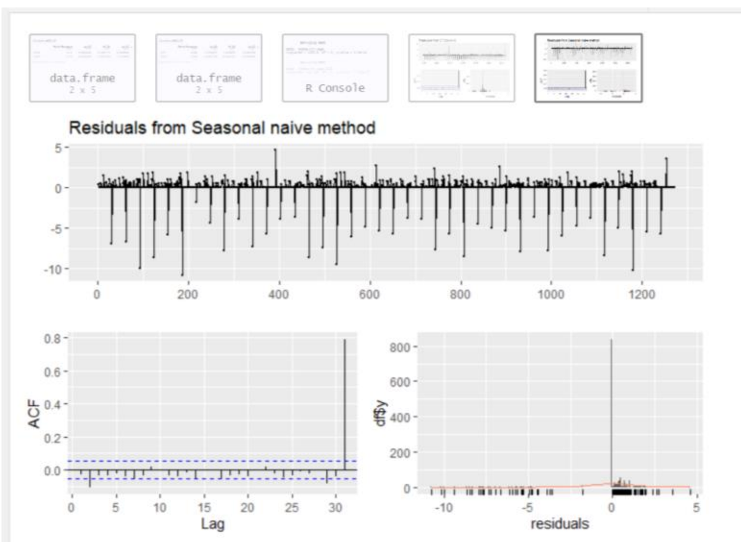
Model df: 4.   Total lags used: 10


        Ljung-Box test

data:  Residuals from Seasonal naive method
Q* = 26.017, df = 10, p-value = 0.003717

Model df: 0.   Total lags used: 10
```






```
```{r}
# ARIMA forecasting model
fit_arima <- auto.arima(timeseries_df,
                        d=1,
                        D=1,
                        stepwise=FALSE,
                        approximation =  FALSE,
                        trace=TRUE) # model calibration

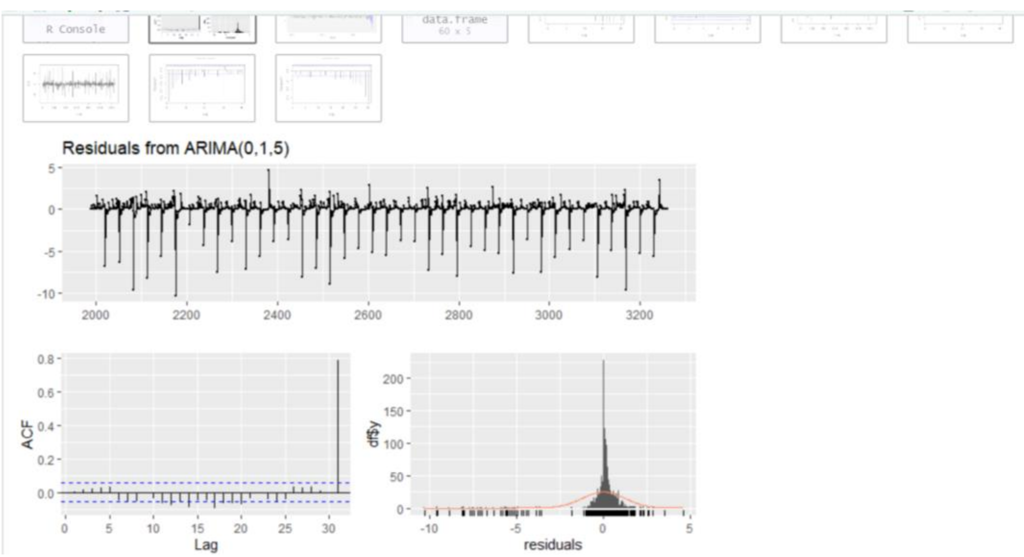
checkresiduals(fit_arima) # visualize model performance

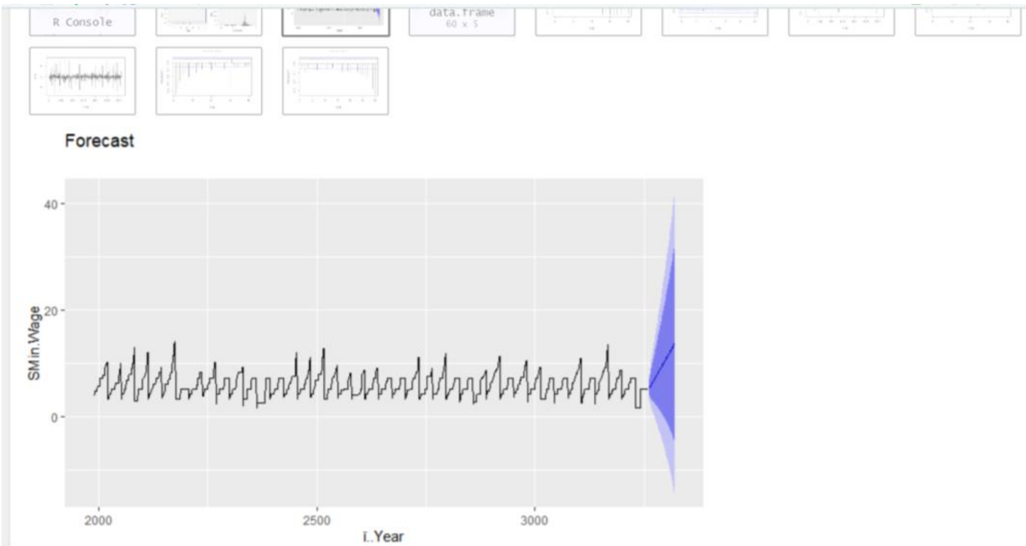
# let us forecast using the ETS model; it outperforms ARIMA
FC <- forecast(fit_ets,
               h=60)

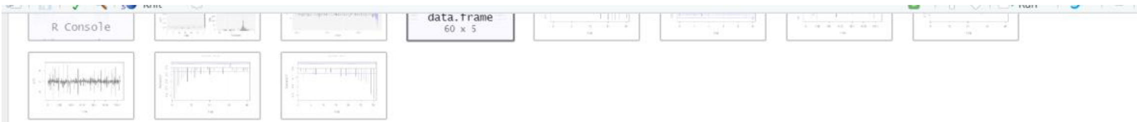
autoplot(FC) +
  labs(title="Forecast",
       subtitle="",
       x = "ï..Year",
```
```

```
        y = "SMin.Wage")
FC
acf(Wages$State.Min.Wage, lag.max=20)
acf(Wages$State.Min.Wage,  lag.max=10)
diff1 <- diff(State.Min.Wage, differences=1)
plot.ts(diff1)
pacf(diff1, lag.max=40) # plot a partial correlogram
diff2 <- diff(State.Min.Wage.2020, differences=1)
plot.ts(diff2)
pacf(diff2, lag.max=40) # plot a partial correlogram
pacf(diff1, lag.max=30) # plot a partial correlogram
train_series=State.Min.Wage[1:200]
test_series=State.Min.Wage[291:1271]
```

```
ARIMA(0,1,0)                    : 4147.674
ARIMA(0,1,0) with drift         : 4149.679
ARIMA(0,1,1)                    : 4148.263
ARIMA(0,1,1) with drift         : 4150.271
ARIMA(0,1,2)                    : 4134.287
ARIMA(0,1,2) with drift         : 4136.299
ARIMA(0,1,3)                    : 4133.288
ARIMA(0,1,3) with drift         : 4135.303
ARIMA(0,1,4)                    : 4130.082
ARIMA(0,1,4) with drift         : 4132.1
ARIMA(0,1,5)                    : 4128.046
ARIMA(0,1,5) with drift         : 4130.067
ARIMA(1,1,0)                    : 4148.56
ARIMA(1,1,0) with drift         : 4150.568
ARIMA(1,1,1)                    : Inf
ARIMA(1,1,1) with drift         : Inf
ARIMA(1,1,2)                    : Inf
ARIMA(1,1,2) with drift         : Inf
ARIMA(1,1,3)                    : Inf
ARIMA(1,1,3) with drift         : Inf
ARIMA(1,1,4)                    : Inf
ARIMA(1,1,4) with drift         : Inf
ARIMA(2,1,0)                    : 4136.793
```





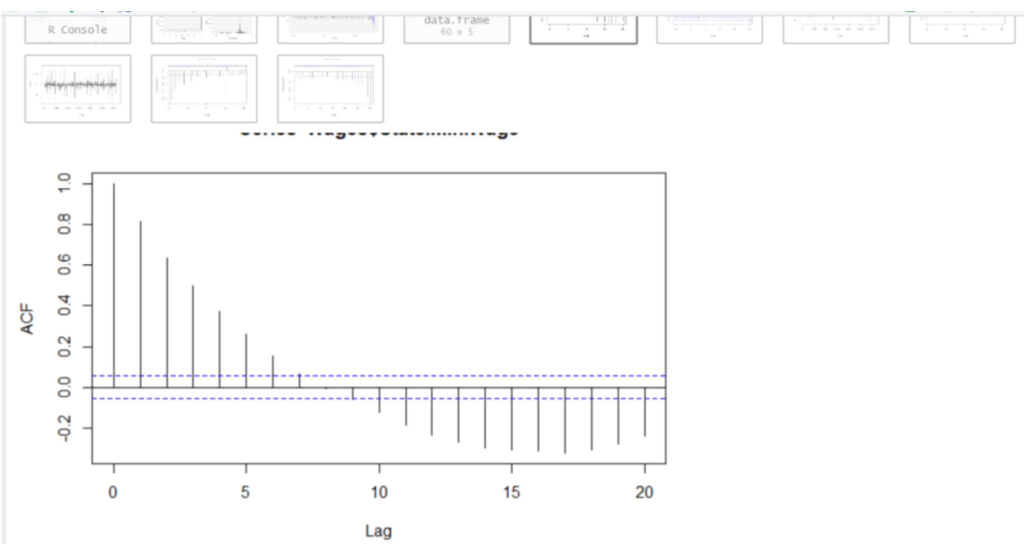


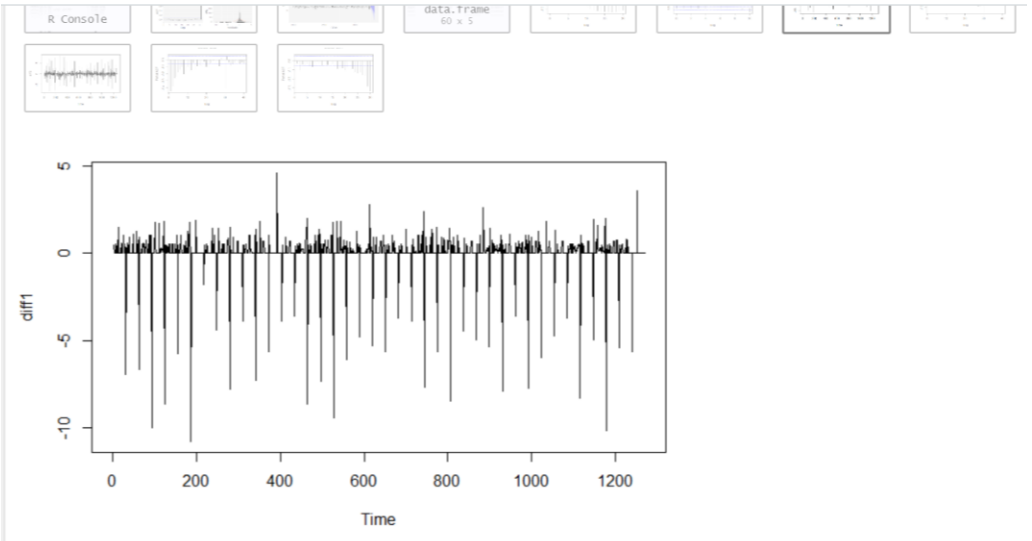
Description: df [60 x 5]

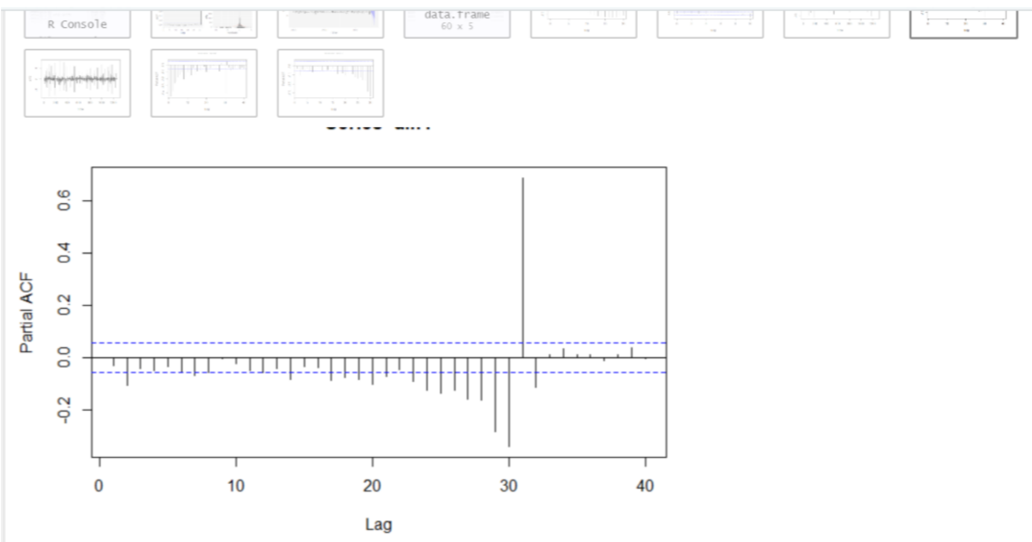
	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
3261	5.302192	4.24919090	6.355193	3.691765705	6.912619
3262	5.444933	3.97240462	6.917462	3.192895055	7.696971
3263	5.587674	3.76962255	7.405726	2.807204079	8.368145
3264	5.730415	3.60372940	7.857101	2.477929837	8.982901
3265	5.873156	3.45987741	8.286436	2.182364633	9.563948
3266	6.015898	3.33040343	8.701392	1.908788700	10.123006
3267	6.158639	3.21076668	9.106511	1.650257523	10.667020
3268	6.301380	3.09802776	9.504732	1.402275656	11.200484
3269	6.444121	2.99016207	9.898080	1.161746749	11.726495
3270	6.586862	2.88570918	10.288015	0.926437268	12.247287

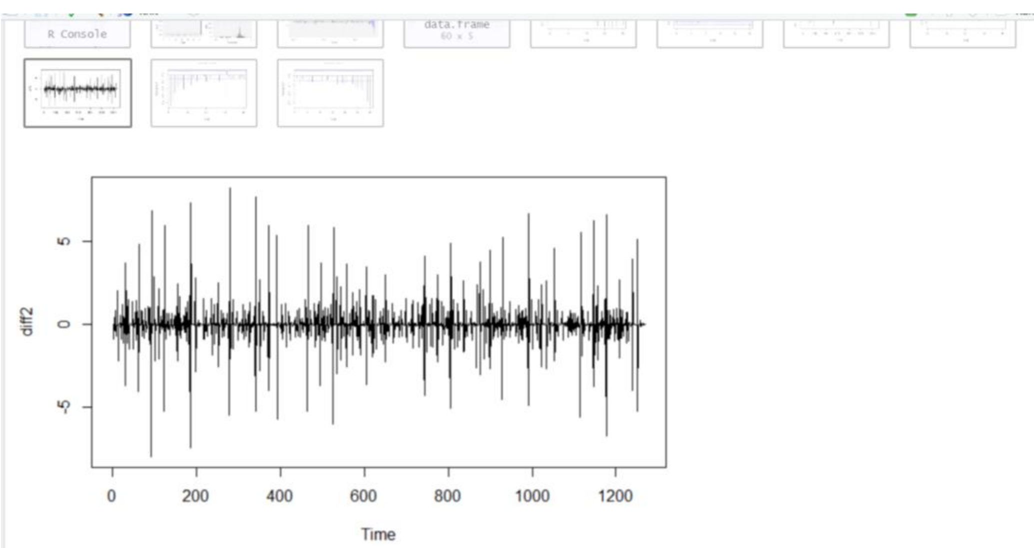
1-10 of 60 rows

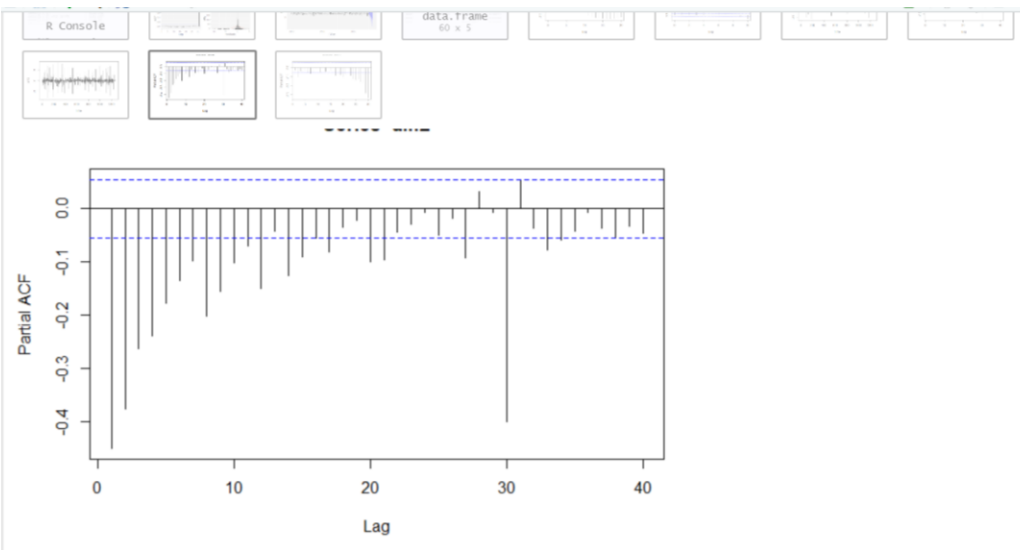
Previous 1 2 3 4 5 6 Next

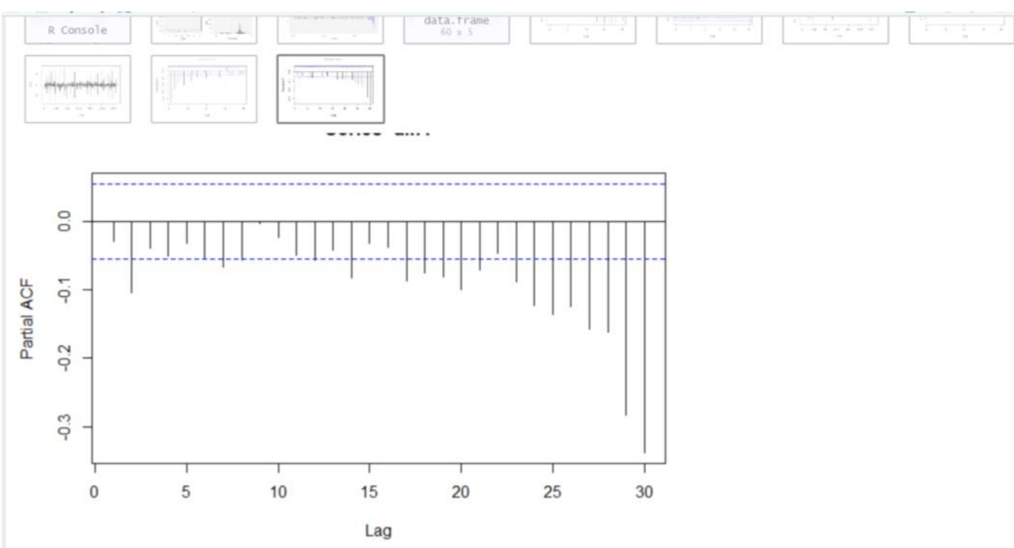



R Console
data.frame
60 x 5
diff1
5
0
-5
-10
0
200
400
600
800
1000
1200
Time


R Console
data.frame
60 x 5
Partial ACF
0.6
0.4
0.2
0.0
-0.2
0
10
20
30
40
Lag


R Console
data.frame
60 x 5
diff2
5
0
-5
0
200
400
600
800
1000
1200
Time





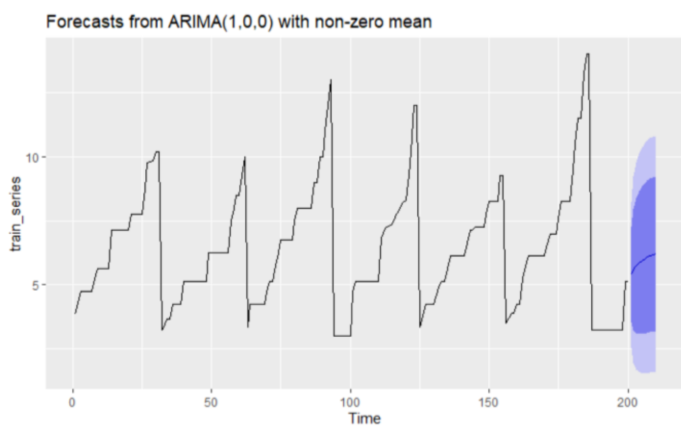

```{r}
## make arima models
arimaModel_1=arima(train_series, order=c(0,1,2))
arimaModel_2=arima(train_series, order=c(1,1,0))
arimaModel_3=arima(train_series,  order=c(1,1,2))
print(arimaModel_1);print(arimaModel_2);print(arimaModel_3)

forecast1=predict(arimaModel_1,  50)
forecast2=predict(arimaModel_2,  50)
forecast3=predict(arimaModel_3,  50)

forecast1
forecast2
forecast3


AutoArimaModel=auto.arima(train_series)
AutoArimaModel
autoplot(forecast(AutoArimaModel))

lambda <- BoxCox.lambda(Wages$State.Min.Wage)
lambda
```

```
```

R Console

```
[50] 6.326091

$se
Time Series:
Start = 201
End = 250
Frequency = 1
 [1] 1.443553 1.910262 2.136837 2.259597 2.329230 2.369788 2.393866
 [8] 2.408399 2.417313 2.422873 2.426404 2.428687 2.430193 2.431204
[15] 2.431897 2.432380 2.432721 2.432965 2.433143 2.433273 2.433369
[22] 2.433440 2.433493 2.433533 2.433563 2.433585 2.433602 2.433615
[29] 2.433625 2.433632 2.433638 2.433642 2.433645 2.433648 2.433649
[36] 2.433651 2.433652 2.433653 2.433653 2.433654 2.433654 2.433654
[43] 2.433655 2.433655 2.433655 2.433655 2.433655 2.433655 2.433655
[50] 2.433655

Series: train_series
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1    mean
      0.7886  6.3179
s.e.  0.0430  0.4740

sigma^2 estimated as 2.101:  log likelihood=-357.49
AIC=720.98   AICc=721.1   BIC=730.87
[1] -0.3750581
```

R Console




```{r}
#   Predictions-Forecasts---State.Min.Wage--State.Min.Wage--State.Min.Wage

####   Naive Forecasting Method
naive_mod <- naive(Wages$State.Min.Wage, h = 50)
summary(naive_mod)

fcast <- forecast(naive_mod, h=50)
autoplot(fcast)
###   Simple Exponential Smoothing
se_model <- ses(Wages$State.Min.Wage, h = 50)
summary(se_model)
fcast1 <- forecast(se_model, h=50)
autoplot(fcast1)
```

```
#######   Holt's Trend Method
holt_model <- holt(Wages$State.Min.Wage, h = 50)
summary(holt_model)

fcast2 <- forecast(holt_model, h=50)
autoplot(fcast2)

####   arima  Method
arima_model <- auto.arima(Wages$State.Min.Wage)
summary(arima_model)
fcast3 <- forecast(arima_model, h=50)
fcast3
autoplot(fcast3)

#### linear Regression
# Fit linear  regression
fit_lr = lm(as.vector(State.Min.Wage) ~ ï..Year)
fit_lr
par(mfrow = c(3, 2))
plot(fit, ask=FALSE)
plot(fit_lr)

```

R Console | 50 x 5 | 50 x 5 | 50 x 5 | 50 x 5

```
Training set 1.013568 0.002905258

Forecasts:
Series: Wages$State.Min.Wage
ARIMA(3,0,2) with non-zero mean

Coefficients:
         ar1     ar2      ar3     ma1      ma2    mean
      0.7968  0.8768  -0.7396  0.0817  -0.8935  5.9791
s.e.  0.0282  0.0230   0.0235  0.0257   0.0250  0.0929

sigma^2 estimated as 1.344:  log likelihood=-1989.21
AIC=3992.41   AICc=3992.5   BIC=4028.45

Training set error measures:
                       ME     RMSE       MAE       MPE    MAPE
Training set 0.0007386431 1.156591 0.5867064 -4.967018 12.8956
                 MASE        ACF1
Training set 1.507724 -0.0137811

Call:
lm(formula = as.vector(State.Min.Wage) ~ ï..Year)

Coefficients:
(Intercept)      ï..Year
  -373.7341       0.1894
```

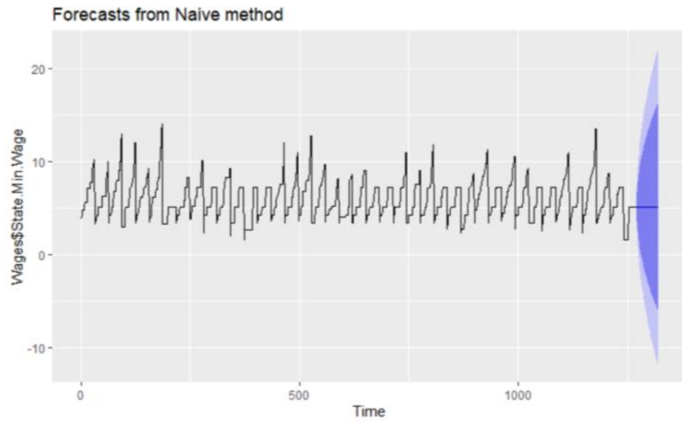
R Console | 50 x 5 | 50 x 5 | 50 x 5

Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.15	3.5638863	6.736114	2.7242484	7.575752
1273	5.15	2.9068964	7.393104	1.7194691	8.580531
1274	5.15	2.4027704	7.897230	0.9484749	9.351525
1275	5.15	1.9777725	8.322227	0.2984967	10.001503
1276	5.15	1.6033419	8.696658	-0.2741456	10.574146
1277	5.15	1.2648307	9.035169	-0.7918538	11.091854
1278	5.15	0.9535375	9.346463	-1.2679356	11.567936
1279	5.15	0.6637929	9.636207	-1.7110617	12.011062
1280	5.15	0.3916588	9.908341	-2.1272549	12.427255
1281	5.15	0.1342680	10.165732	-2.5209002	12.820900

1-10 of 50 rows   Previous 1 2 3 4 5 Next

R Console



R Console

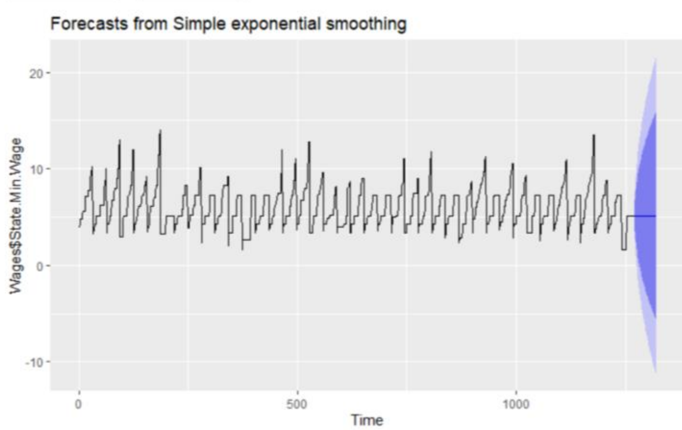
Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.15	3.56414770	6.735852	2.7246482	7.575352
1273	5.15	2.94907432	7.350926	1.7839746	8.516025
1274	5.15	2.47171320	7.828287	1.0539138	9.246086
1275	5.15	2.06740912	8.232591	0.4355841	9.864416
1276	5.15	1.71030338	8.589697	-0.1105620	10.410562
1277	5.15	1.38693487	8.913065	-0.6051115	10.905111
1278	5.15	1.08923585	9.210764	-1.0604028	11.360403
1279	5.15	0.81191842	9.488082	-1.4845233	11.784523
1280	5.15	0.55129386	9.748706	-1.8831141	12.183114
1281	5.15	0.30466776	9.995332	-2.2602962	12.560296

1-10 of 50 rows

Previous 1 2 3 4 5 Next

R Console



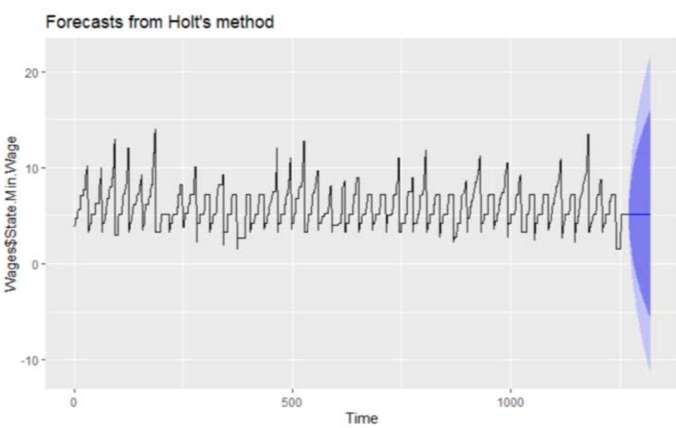
R Console

Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.151008	3.56379064	6.738226	2.7235684	7.578448
1273	5.151980	2.94765149	7.356308	1.7807505	8.523209
1274	5.152952	2.46979045	7.836113	1.0494107	9.256492
1275	5.153923	2.06521327	8.242633	0.4301490	9.877697
1276	5.154895	1.70795556	8.601834	-0.1167439	10.426534
1277	5.155867	1.38450894	8.927224	-0.6119272	10.923660
1278	5.156838	1.08678028	9.226896	-1.0677782	11.381455
1279	5.157810	0.80946639	9.506153	-1.4924076	11.808027
1280	5.158781	0.54886868	9.768694	-1.8914717	12.209035
1281	5.159753	0.30228595	10.017220	-2.2691018	12.588608

1-10 of 50 rows Previous 1 2 3 4 5 Next

R Console

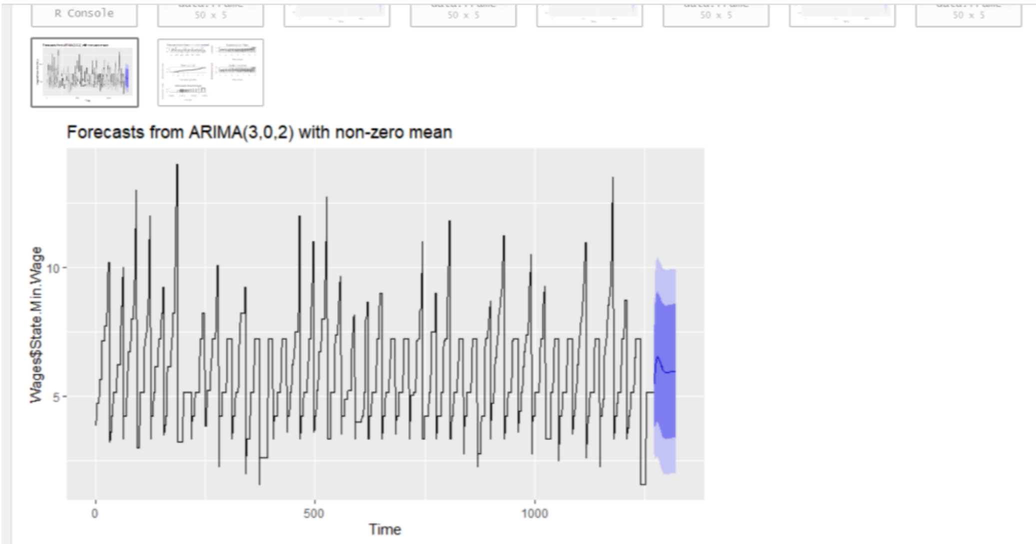


R Console

Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.485487	3.999745	6.971228	3.213241	7.757732
1273	5.813502	3.835832	7.791172	2.788917	8.838087
1274	6.027567	3.804518	8.250616	2.627706	9.427427
1275	6.237613	3.855962	8.619264	2.595192	9.880034
1276	6.350070	3.892595	8.807544	2.591687	10.108452
1277	6.465521	3.961405	8.969637	2.635806	10.295236
1278	6.500763	3.980778	9.020748	2.646778	10.354748
1279	6.546899	4.019086	9.074712	2.680942	10.412855
1280	6.529170	4.000859	9.057480	2.662452	10.395887
1281	6.529430	4.001113	9.057746	2.662703	10.396156

1-10 of 50 rows Previous 1 2 3 4 5 Next



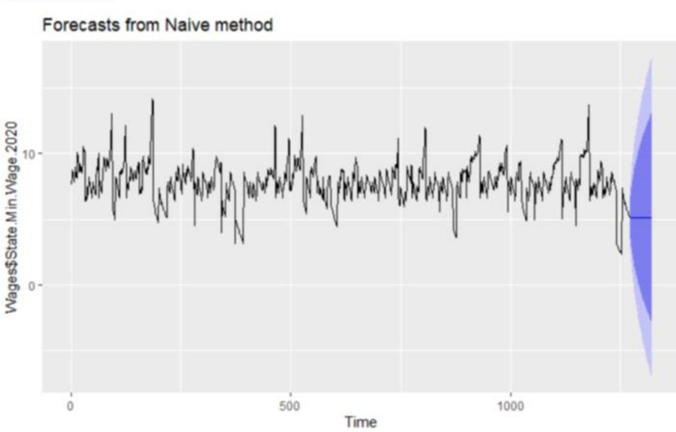

```{r}
#  Predictions-Forecasts---State.Min.Wage.2020--State.Min.Wage.2020

####  Naive Forecasting Method
naive_mod <- naive(Wages$State.Min.Wage.2020, h = 50)
summary(naive_mod)

fcast4 <- forecast(naive_mod, h=50)
autoplot(fcast4)
#Simple Exponential Smoothing
se_model <- ses(Wages$State.Min.Wage.2020, h = 50)
summary(se_model)
fcast5 <- forecast(se_model, h=50)
autoplot(fcast5)

##Holt's Trend Method
holt_model <- holt(Wages$State.Min.Wage.2020, h = 50)
summary(holt_model)

fcast6 <- forecast(holt_model, h=50)
autoplot(fcast6)

####   arima  Method
arima_model <- auto.arima(Wages$State.Min.Wage.2020)
summary(arima_model)
fcast7 <- forecast(arima_model, h=50)
fcast7
autoplot(fcast7)
```

```
     AIC     AICc      BIC 
8755.318 8755.366 8781.056 

Error measures:
                        ME      RMSE      MAE        MPE     MAPE
Training set -0.0002892966 0.8750714 0.431416 -0.8218996 5.939393
                 MASE        ACF1
Training set 1.001734 0.007770453

Forecasts:
Series: Wages$State.Min.Wage.2020 
ARIMA(2,0,1) with non-zero mean 

Coefficients:
          ar1     ar2    ma1    mean
      -0.1118  0.7791  0.975  7.7573
s.e.   0.0493  0.0461  0.035  0.1385

sigma^2 estimated as 0.6985:  log likelihood=-1574.09
AIC=3158.17   AICc=3158.22   BIC=3183.91

Training set error measures:
                       ME      RMSE       MAE       MPE     MAPE
Training set 0.0001356481 0.8344191 0.4762708 -1.472612 6.877775
                 MASE       ACF1
Training set 1.105885 0.02068917
```

Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.15	4.027243234	6.272757	3.432891704	6.867108
1273	5.15	3.562182154	6.737818	2.721642159	7.578358
1274	5.15	3.205328236	7.094672	2.175881188	8.124119
1275	5.15	2.904486467	7.395514	1.715783407	8.584217
1276	5.15	2.639439548	7.660560	1.310429124	8.989571
1277	5.15	2.399818817	7.900181	0.943960841	9.356039
1278	5.15	2.179464813	8.120535	0.606958473	9.693042
1279	5.15	1.974364307	8.325636	0.293284318	10.006716
1280	5.15	1.781729701	8.518270	-0.001324889	10.301325
1281	5.15	1.599531360	8.700469	-0.279973206	10.579973

1-10 of 50 rows   Previous 1 2 3 4 5 Next

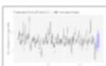


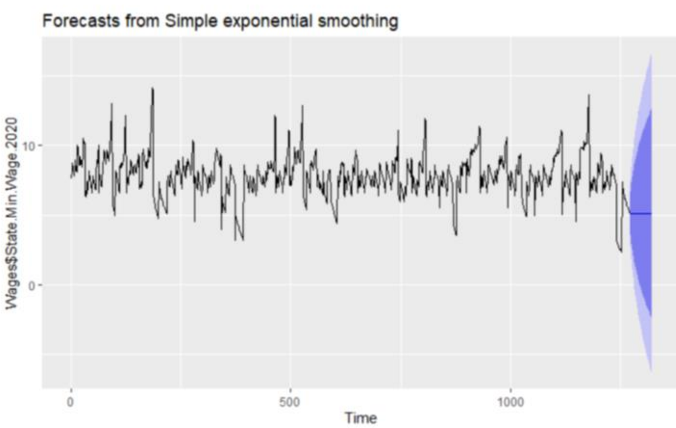
Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.153585	4.03157162	6.275598	3.437613693	6.869556
1273	5.153585	3.60972764	6.697442	2.792458999	7.514711
1274	5.153585	3.28059887	7.026571	2.289099982	8.018070
1275	5.153585	3.00122385	7.305946	1.861832759	8.445337
1276	5.153585	2.75416007	7.553010	1.483981315	8.823188
1277	5.153585	2.53026240	7.776907	1.141559380	9.165610
1278	5.153585	2.32402621	7.983143	0.826148352	9.481021
1279	5.153585	2.13183313	8.175337	0.532214391	9.774955
1280	5.153585	1.95115376	8.356016	0.255889148	10.051281
1281	5.153585	1.78013761	8.527032	-0.005657484	10.312827

1-10 of 50 rows

Previous 1 2 3 4 5 Next



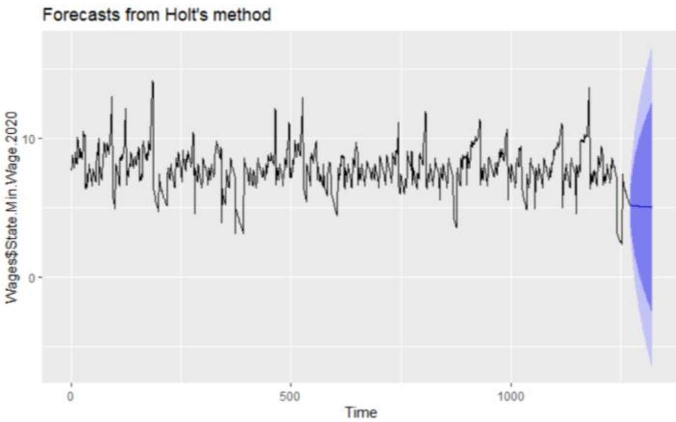


Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.151095	4.02787671	6.274313	3.43328103	6.868908
1273	5.148740	3.60315887	6.694321	2.78497774	7.512502
1274	5.146385	3.27121541	7.021555	2.27856061	8.014209
1275	5.144030	2.98904987	7.299010	1.84827225	8.439788
1276	5.141675	2.73920624	7.544144	1.46741598	8.815934
1277	5.139320	2.51253272	7.766108	1.12199534	9.156645
1278	5.136966	2.30352095	7.970410	0.80358601	9.470345
1279	5.134611	2.10855027	8.160671	0.50665068	9.762571
1280	5.132256	1.92508984	8.339422	0.22731874	10.037193
1281	5.129901	1.75128817	8.508514	-0.03724138	10.297043

1-10 of 50 rows

Previous 1 2 3 4 5 Next



Description: df [50 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1272	5.635898	4.564860	6.706935	3.997887	7.273908
1273	5.962954	4.548042	7.377866	3.799033	8.126875
1274	6.304983	4.712324	7.897643	3.869221	8.740746
1275	6.521581	4.805639	8.237523	3.897275	9.145888
1276	6.763862	4.977034	8.550690	4.031144	9.496579
1277	6.905545	5.064880	8.746210	4.090491	9.720599
1278	7.078480	5.206885	8.950075	4.216123	9.940837
1279	7.169544	5.273028	9.066061	4.269072	10.070016
1280	7.294107	5.383723	9.204491	4.372426	10.215787
1281	7.351138	5.428907	9.273369	4.411339	10.290937

1-10 of 50 rows Previous 1 2 3 4 5 Next