```
require(pacman)
```

```
## Loading required package: pacman
```

```
## Warning: package 'pacman' was built under R version 4.1.3
```

```
#Load packages
pacman::p_load(datasets, tseries)
pacman::p_load(rio)
pacman::p_load(MASS)
```

```
#Membaca Data
masterdata <- import("___.xlsx") head(masterdata)
```

```
##   Penumpang
## 1     45980
## 2     36094
## 3     44646
## 4     39799
## 6     48620
```

```
tail(masterdata)
```

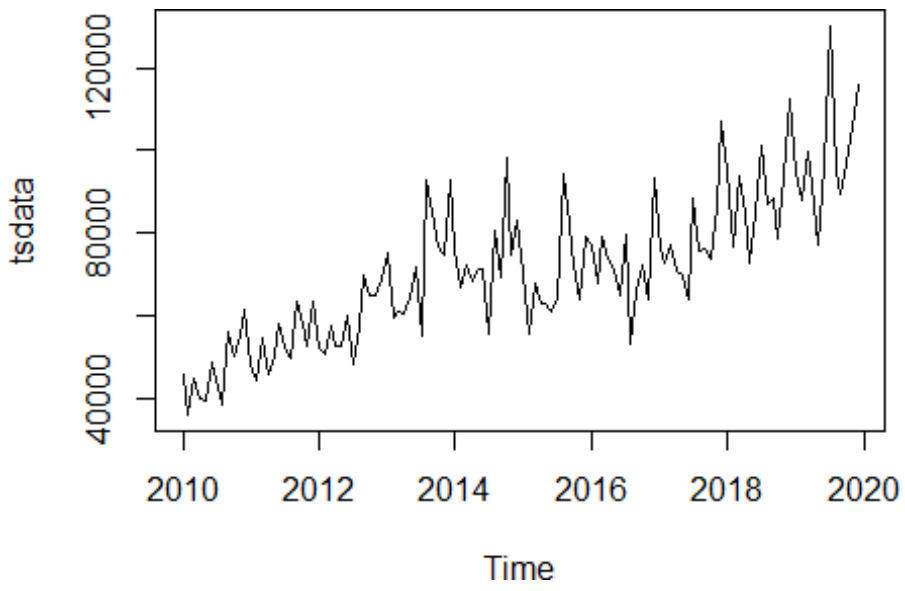
```
##     Penumpang
## 115    130138
## 116     95313
## 117     89222
## 118     96924
## 119    105504
## 120    116099
```

```
summary(masterdata)
```

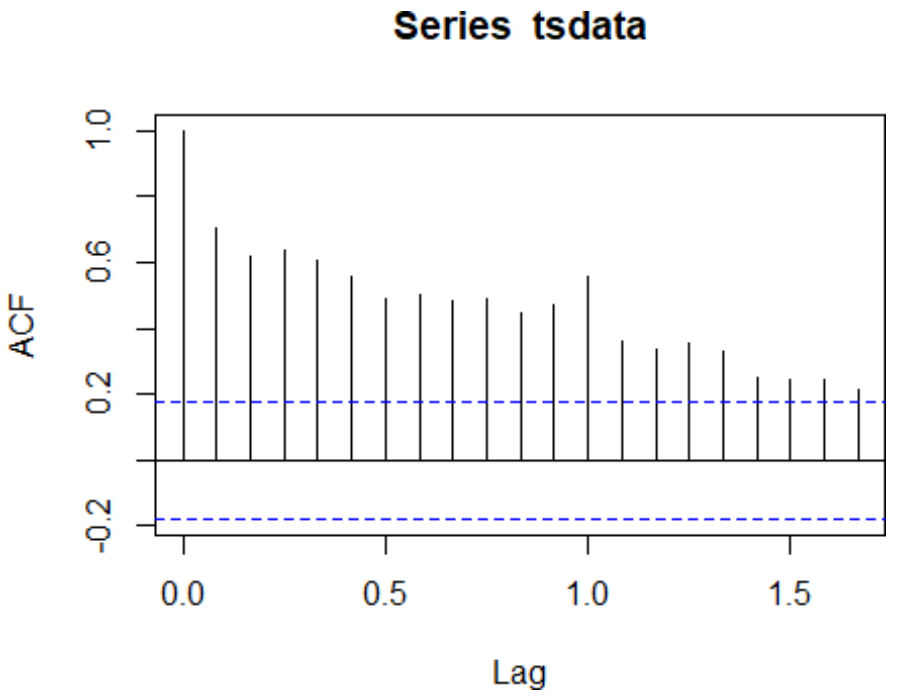
```
##    Penumpang
## Min.    : 36094
##  1st Qu.: 57875
## Median :  70551
##  Mean   :  71218
## 3rd Qu.: 82537
## Max.   : 130138
```

```
#Plot time series
tsdata <- ts(masterdata$Penumpang, frequency = 12, start = c(2010, 1))
tsdata
```

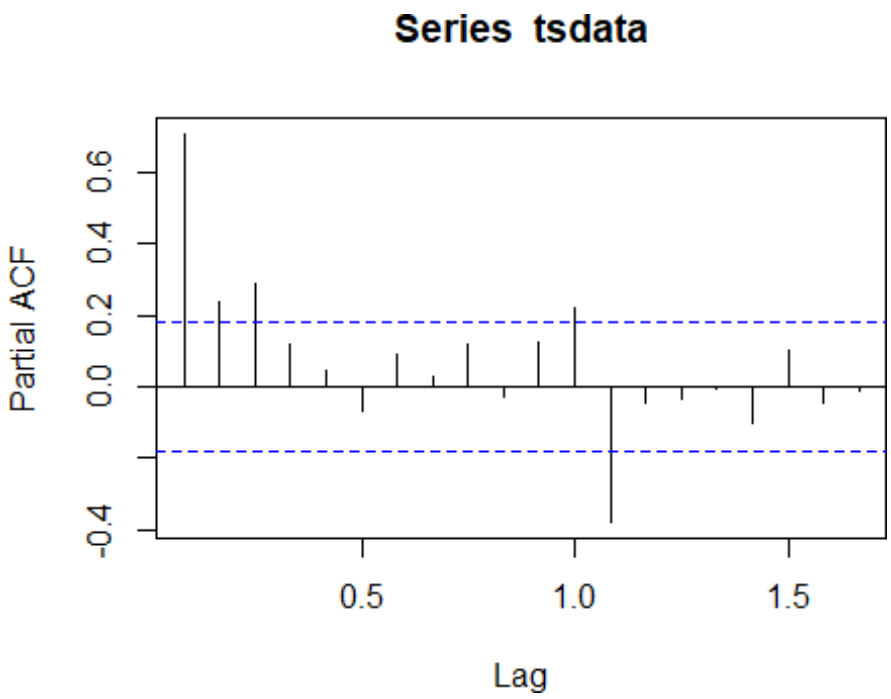
```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2010  45980  36094  44646  39799  39363  48620  45032  38335  55997  50353
## 2011  48350  44405  54560  45742  50127  58210  52658  49643  63390  58000
## 2012  52029  50785  57386  52720  52604  59982  48206  57501  69700  64800
## 2013  75029  59691  61132  60353  64384  71637  55163  92622  84801  76940
## 2014  76063  66733  72501  68315  71341  71210  55537  80665  69489  98232
## 2015  70408  55608  67923  63127  62931  60985  63755  94383  82329  72086
## 2016  76998  67737  78994  73520  71163  64703  79797  53219  66131  72468
## 2017  76162  72852  77381  70694  69750  64102  88245  75571  76091  73603
## 2018  95096  76779  94044  84798  72560  85175 101424  86919  88481  78428
## 2019  95023  87859  99910  91459  76965  96241 130138  95313  89222  96924
##         Nov    Dec
## 2010  55043  61665
## 2011  52940  63347
## 2012  65061  68807
## 2013  74757  92658
## 2014  74874  83161
## 2015  63856  78991
## 2016  63731  93487
## 2017  86459 107000
## 2018  93862 112532
## 2019 105504 116099

plot(tsdata)
```




```
acf(tsdata)
```




```
pacf(tsdata)
```




```
#Cek kestasioneritasan Data
adf.test(tsdata)

##
##  Augmented Dickey-Fuller Test
##
## data:  tsdata
## Dickey-Fuller = -3.5351, Lag order = 4, p-value = 0.0421
## alternative hypothesis: stationary

#transformasi box-cox

library(TSA)

## Warning: package 'TSA' was built under R version 4.1.3

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

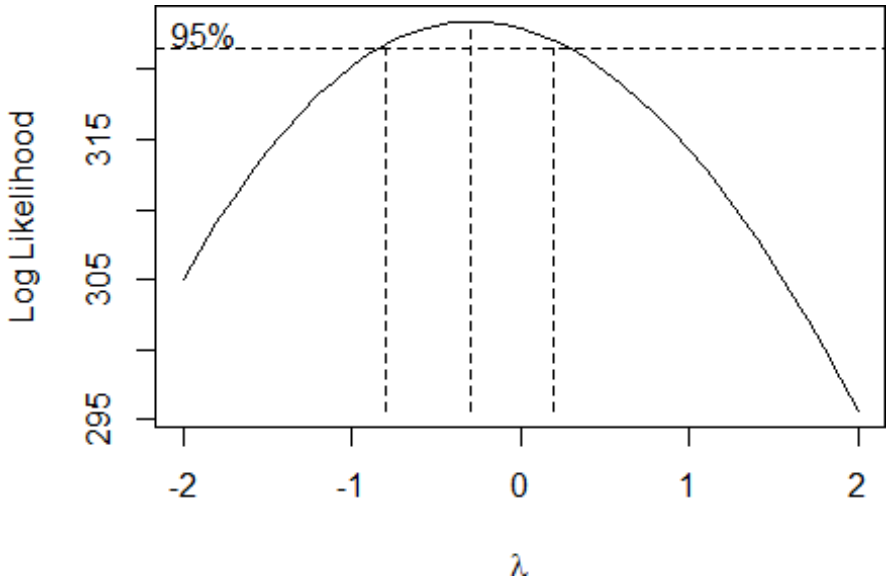
```
library(MASS)
library(car)

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.3

BoxCox.ar(tsdata)

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
```

```
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
```

```
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1
```

```
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1
```

```
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possibl
e
## convergence problem: optim gave code = 1
```



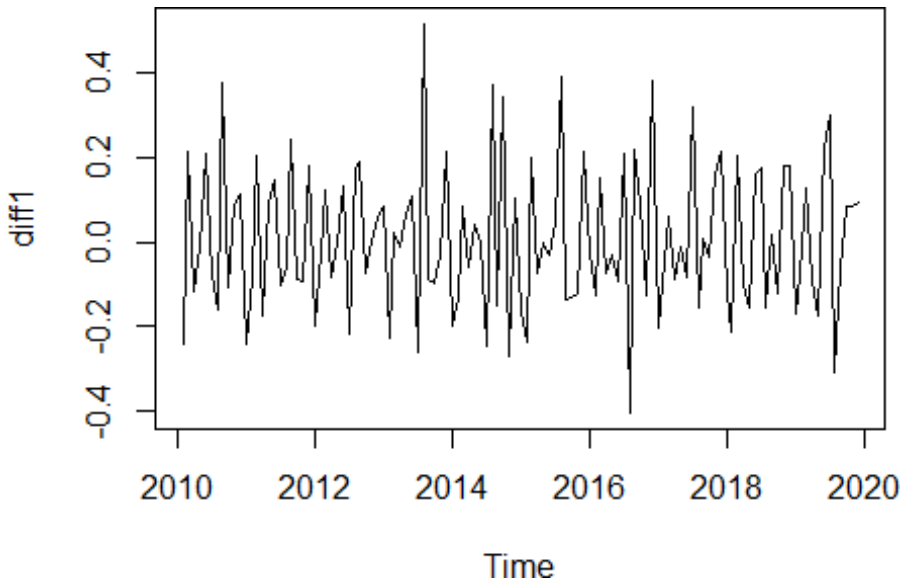

```
#Print data transformasi boxcox
boxcox <- bcPower(tsdata, 0)
boxcox
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      A
ug
## 2010 10.73596 10.49388 10.70652 10.59160 10.58058 10.79179 10.71513 10.554
12
## 2011 10.78622 10.70111 10.90706 10.73077 10.82232 10.97181 10.87157 10.812
61
## 2012 10.85956 10.83536 10.95756 10.87275 10.87055 11.00180 10.78324 10.959
56
## 2013 11.22563 10.99694 11.02079 11.00797 11.07262 11.17937 10.91805 11.436
28
## 2014 11.23932 11.10845 11.19136 11.13188 11.17523 11.17339 10.92480 11.298
06
## 2015 11.16206 10.92608 11.12613 11.05290 11.04979 11.01838 11.06280 11.455
12
## 2016 11.25153 11.12339 11.27713 11.20531 11.17273 11.07756 11.28724 10.882
17
## 2017 11.24062 11.19619 11.25650 11.16612 11.15267 11.06823 11.38787 11.232
83
## 2018 11.46264 11.24869 11.45152 11.34803 11.19217 11.35246 11.52707 11.372
```

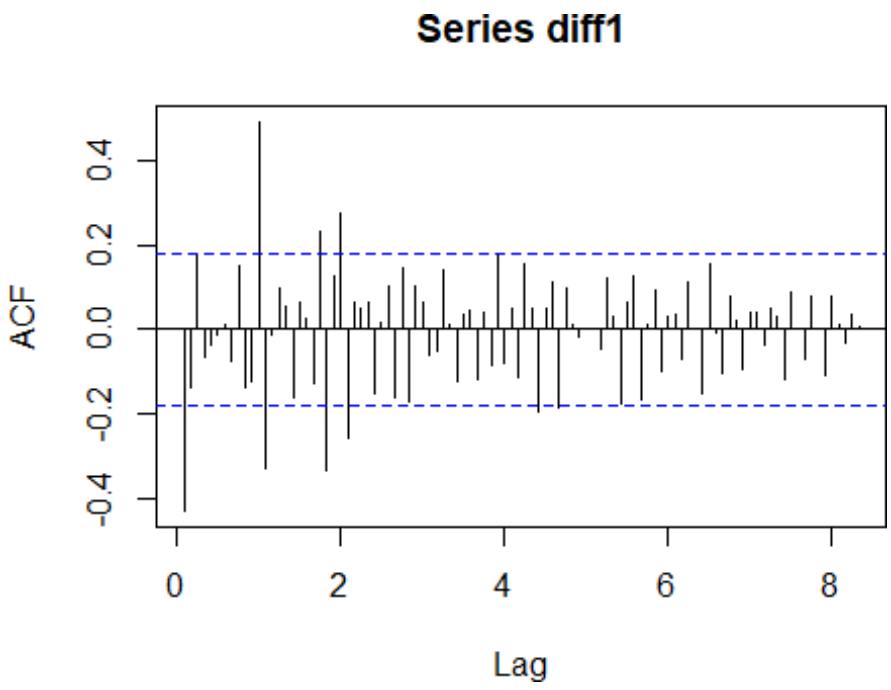
```
73
## 2019 11.46187 11.38349 11.51203 11.42365 11.25111 11.47461 11.77635 11.464
92
##           Sep      Oct      Nov      Dec
## 2010 10.93305 10.82681 10.91587 11.02947
## 2011 11.05706 10.96820 10.87691 11.05638
## 2012 11.15196 11.07906 11.08308 11.13906
## 2013 11.34806 11.25078 11.22200 11.43667
## 2014 11.14892 11.49509 11.22356 11.32853
## 2015 11.31848 11.18562 11.06439 11.27709
## 2016 11.09939 11.19090 11.06243 11.44558
## 2017 11.23969 11.20644 11.36743 11.58058
## 2018 11.39054 11.26994 11.44958 11.63099
## 2019 11.39888 11.48168 11.56650 11.66220

#diff 1
diff1 <- diff(boxcox, 1)
diff1

##               Jan          Feb          Mar          Apr          May
## 2010              -0.242079873  0.212638071 -0.114922931 -0.011015498
## 2011 -0.243250287 -0.085114146  0.205948938 -0.176284100  0.091542872
## 2012 -0.196826294 -0.024200220  0.122199336 -0.084805481 -0.002202728
## 2013  0.086569222 -0.228693450  0.023854205 -0.012824804  0.064654499
## 2014 -0.197353351 -0.130862361  0.082900772 -0.059470993  0.043341835
## 2015 -0.166471595 -0.235979817  0.200047635 -0.073226140 -0.003109682
## 2016 -0.025554474 -0.128146888  0.153739341 -0.071814422 -0.032584458
## 2017 -0.204959739 -0.044432665  0.060311286 -0.090380569 -0.013443283
## 2018 -0.117941927 -0.213955742  0.202831593 -0.103490800 -0.155858152
## 2019 -0.169118658 -0.078385711  0.128536524 -0.088378996 -0.172540011
##               Jun          Jul          Aug          Sep          Oct
## 2010  0.211208680 -0.076661621 -0.161010031  0.378934801 -0.106239918
## 2011  0.149497376 -0.100238987 -0.058960780  0.244448727 -0.088863110
## 2012  0.131252355 -0.218561023  0.176318844  0.192397979 -0.072894714
## 2013  0.106746545 -0.261319262  0.518234256 -0.088219359 -0.097281438
## 2014 -0.001837939 -0.248583793  0.373255311 -0.149136309  0.346163561
## 2015 -0.031410956  0.044419680  0.392313360 -0.136637557 -0.132863564
## 2016 -0.095165452  0.209678341 -0.405070434  0.217222148  0.091507462
## 2017 -0.084441856  0.319641472 -0.155044425  0.006857381 -0.033244207
## 2018  0.160294158  0.174601786 -0.154333099  0.017811189 -0.120606833
## 2019  0.223504689  0.301739964 -0.311429213 -0.066038567  0.082799520
##               Nov          Dec
## 2010  0.089056498  0.113601811
## 2011 -0.091283814  0.179468353
## 2012  0.004019688  0.055980193
## 2013 -0.028783044  0.214672442
## 2014 -0.271525327  0.104971787
## 2015 -0.121229302  0.212703374
## 2016 -0.128473985  0.383151289
## 2017  0.160984527  0.213158521
```

```
## 2018  0.179644612  0.181412007
## 2019  0.084821701  0.095694409
```

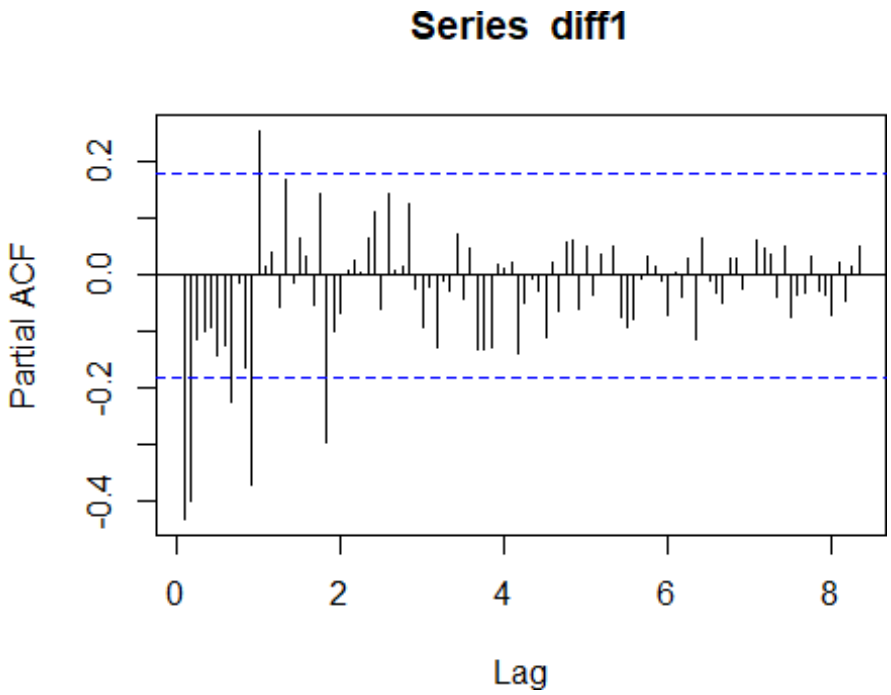
```
plot(diff1)
```




```
acf(diff1,lag.max = 100)
```




```
pacf(diff1,lag.max=100)
```



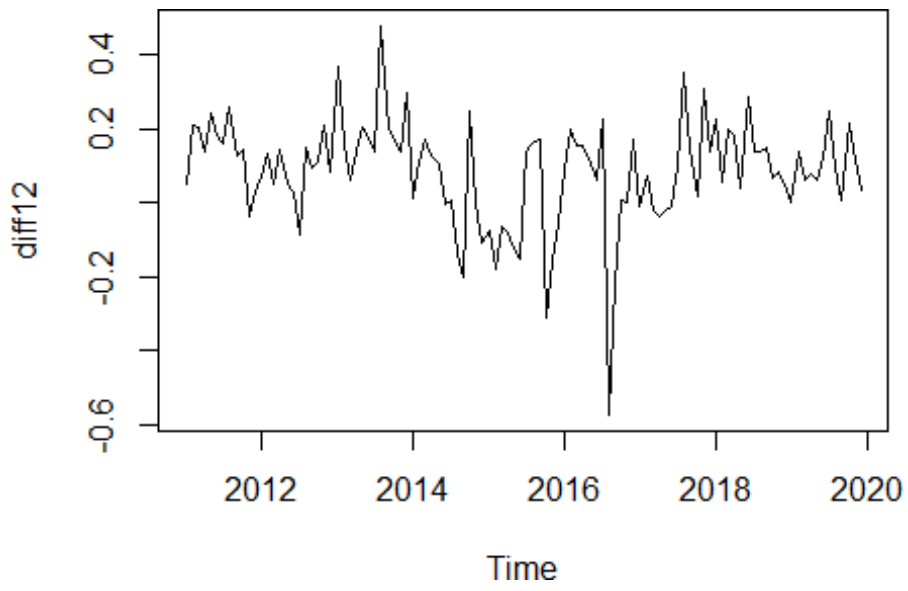

```
adf.test(diff1)
```

```
## Warning in adf.test(diff1): p-value smaller than printed p-value

##
## Augmented Dickey-Fuller Test
##
## data:  diff1
## Dickey-Fuller = -6.6756, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

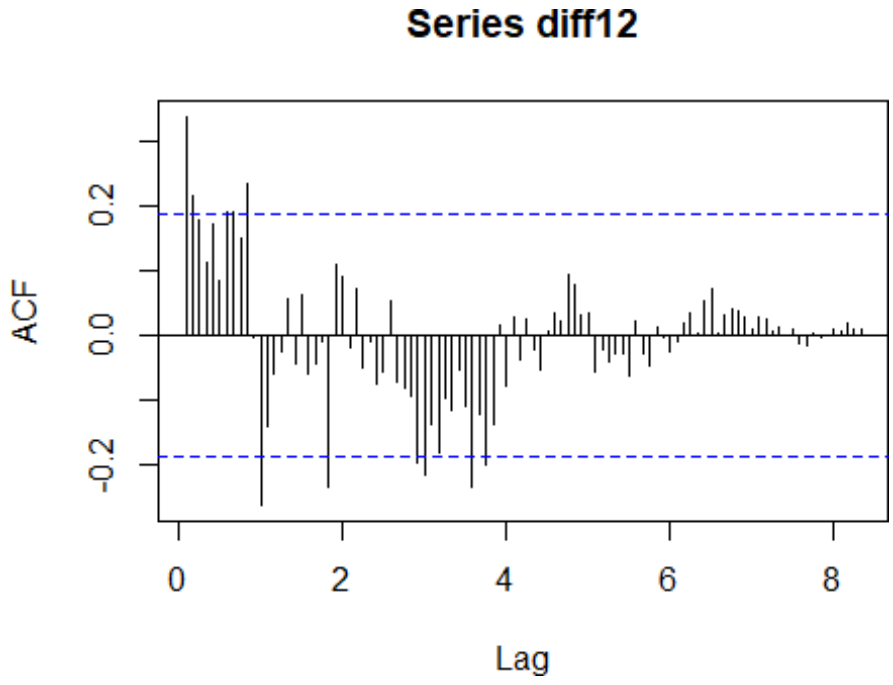
```
#diff 12
diff12 <- diff(boxcox, 12)
diff12
```

```
##                Jan           Feb           Mar           Apr           May
## 2011  0.0502597026  0.2072254292  0.2005362959  0.1391751267  0.2417334963
## 2012  0.0733350335  0.1342489597  0.0504993576  0.1419779772  0.0482323774
## 2013  0.3660734500  0.1615802199  0.0632350895  0.1352157661  0.2020729926
## 2014  0.0136872388  0.1115183279  0.1705648942  0.1239187055  0.1026060422
## 2015 -0.0772550512 -0.1823725074 -0.0652256442 -0.0789807915 -0.1254323091
## 2016  0.0894725544  0.1973054836  0.1509971896  0.1524089078  0.1229341323
## 2017 -0.0109167968  0.0727974265 -0.0206306278 -0.0391967744 -0.0200555997
## 2018  0.2220242570  0.0525011793  0.1950214854  0.1819112539  0.0394963851
## 2019 -0.0007679401  0.1348020915  0.0605070229  0.0756188268  0.0589369673
##                Jun           Jul           Aug           Sep           Oct
## 2011  0.1800221924  0.1564448257  0.2584940766  0.1240080024  0.1413848102
## 2012  0.0299873559 -0.0883346792  0.1469449453  0.0948941975  0.1108625928
## 2013  0.1775671831  0.1348089440  0.4767243555  0.1961070174  0.1717202940
## 2014 -0.0059784423  0.0067570266 -0.1382219183 -0.1991388685  0.2443061305
## 2015 -0.1550053257  0.1379981470  0.1570561961  0.1695549483 -0.3094721772
## 2016  0.0591796360  0.2244382974 -0.5729454967 -0.2190857917  0.0052852344
## 2017 -0.0093320036  0.1006311272  0.3506571363  0.1402923691  0.0155407007
## 2018  0.2842323989  0.1391927126  0.1399040388  0.1508578473  0.0634952207
## 2019  0.1221474988  0.2492856763  0.0921895622  0.0083398062  0.2117461598
##                Nov           Dec
## 2011 -0.0389555012  0.0269110407
## 2012  0.2061660946  0.0826779343
## 2013  0.1389175620  0.2976098115
## 2014  0.0015638474 -0.1081368075
## 2015 -0.1591761524 -0.0514445660
## 2016 -0.0019594479  0.1684884673
## 2017  0.3049992127  0.1350064453
## 2018  0.0821553053  0.0504087912
## 2019  0.1169232484  0.0312056498
```

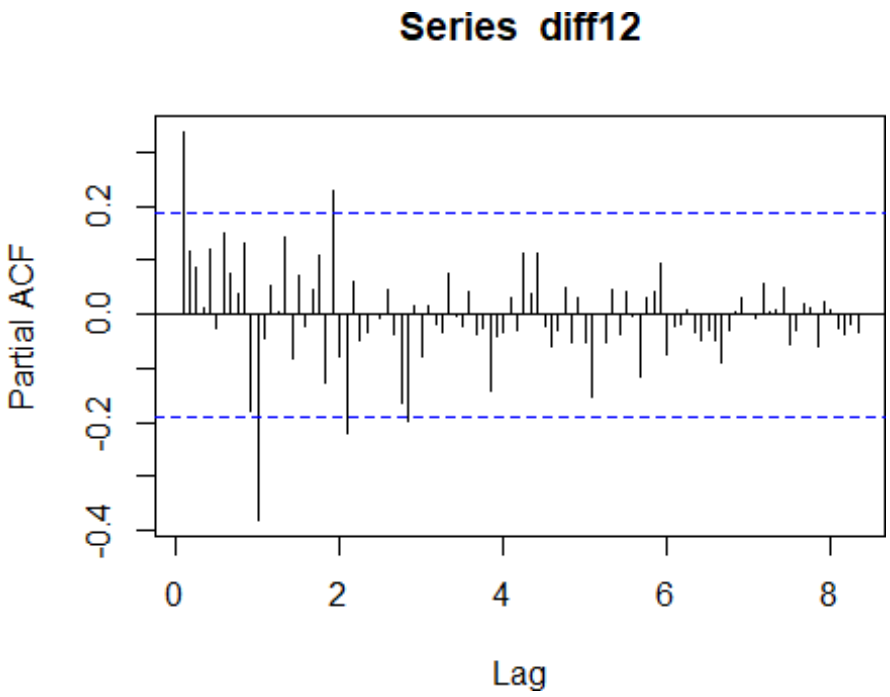
```
plot(diff12)
```




```
acf(diff12,lag.max = 100)
```




```
pacf(diff12,lag.max=100)
```

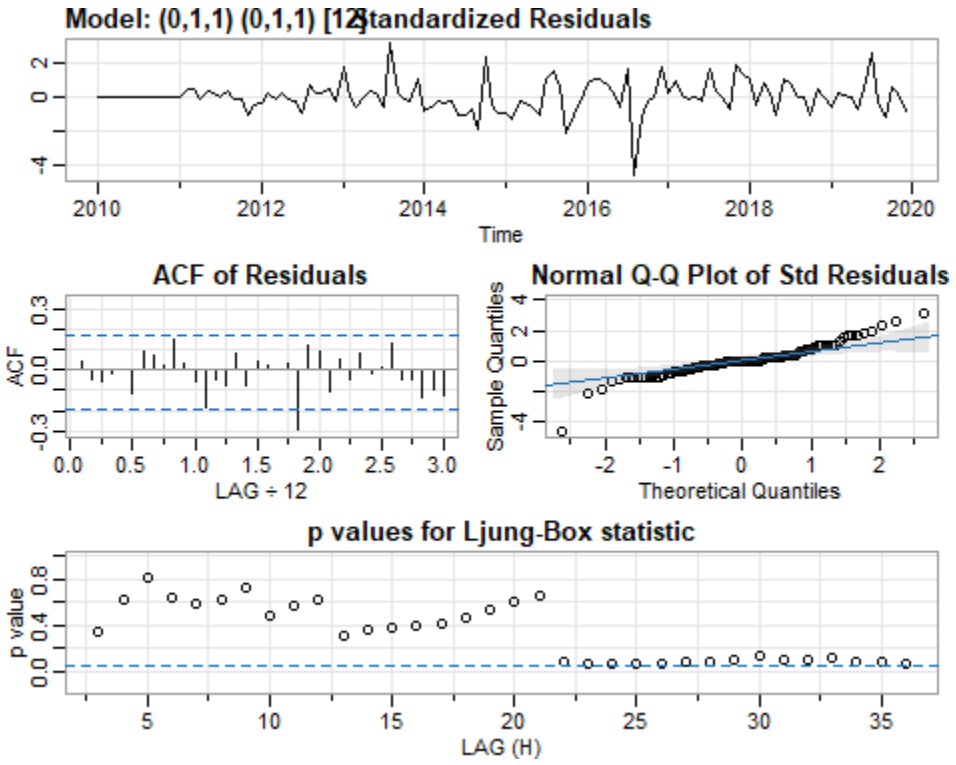



```
adf.test(diff12)

##
##  Augmented Dickey-Fuller Test
##
## data:  diff12
## Dickey-Fuller = -3.1647, Lag order = 4, p-value = 0.0974
## alternative hypothesis: stationary

##Estimasi Model
library(astsa)

## Warning: package 'astsa' was built under R version 4.1.3

Imasima <- arima(boxcox, order = c(0,1,1), seasonal = list(order = c(0,1,1),
period = 12), include.mean = FALSE)
Imasima

##
## Call:
## arima(x = boxcox, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
period = 12),
##     include.mean = FALSE)
##
## Coefficients:
##           ma1     sma1
##       -0.7919  -0.4352
## s.e.   0.0603   0.1064
```

```
##
## sigma^2 estimated as 0.01533:  log likelihood = 69.92,  aic = -135.84
```

```
sarima(tsdata, 0,1,1, 0,1,1, 12)
```

```
## initial  value 9.413054
## iter   2 value 9.218819
## iter   3 value 9.182209
## iter   4 value 9.144682
## iter   5 value 9.133233
## iter   6 value 9.132268
## iter   7 value 9.131821
## iter   8 value 9.131492
## iter   9 value 9.131487
## iter   9 value 9.131487
## iter   9 value 9.131487
## final  value 9.131487
## converged
## initial  value 9.142397
## iter   2 value 9.141501
## iter   3 value 9.141477
## iter   4 value 9.141469
## iter   4 value 9.141469
## iter   4 value 9.141469
## final  value 9.141469
## converged
```



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, op
tim.control = list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.8016  -0.4134
## s.e.   0.0590   0.1003
##
## sigma^2 estimated as 84456877:  log likelihood = -1129.96,  aic = 2265.93
##
## $degrees_of_freedom
## [1] 105
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.8016 0.0590 -13.5807   0e+00
## sma1  -0.4134 0.1003  -4.1212   1e-04
##
## $AIC
## [1] 21.17689
##
## $AICc
## [1] 21.17797
##
## $BIC
## [1] 21.25183

library(lmtest)

## Warning: package 'lmtest' was built under R version 4.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

#tes coef
coeftest(Imasima)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ma1  -0.791906   0.060322 -13.1280 < 2.2e-16 ***
## sma1 -0.435152   0.106372  -4.0908 4.298e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Uji white noise residual
Box.test(Imasima$residuals,type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  Imasima$residuals
## X-squared = 0.039957, df = 1, p-value = 0.8416
```

```
#Uji Normalitas residual
ks.test(Imasima$residuals,"pnorm",mean=0, sd=sd(Imasima$residuals))
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data:  Imasima$residuals
## D = 0.10395, p-value = 0.1495
## alternative hypothesis: two-sided
```

```
#Forecasting

library(forecast)

library(sarima)

forecasting <- forecast(boxcox, model = Imasima, h = 12)

forecasting

plot(forecasting, main = "Plot Forecasting")
```

Description: df [12 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
Jan 2020	11.58865	11.42998	11.74732	11.34599	11.83131
Feb 2020	11.47403	11.31197	11.63610	11.22617	11.72189
Mar 2020	11.61544	11.45004	11.78084	11.36249	11.86839
Apr 2020	11.52468	11.35602	11.69334	11.26674	11.78262
May 2020	11.38659	11.21473	11.55845	11.12375	11.64943
Jun 2020	11.53897	11.36396	11.71397	11.27132	11.80661
Jul 2020	11.79349	11.61540	11.97158	11.52112	12.06585
Aug 2020	11.55762	11.37649	11.73874	11.28061	11.83463
Sep 2020	11.53180	11.34769	11.71592	11.25023	11.81338
Oct 2020	11.54938	11.36233	11.73643	11.26331	11.83544

1-10 of 12 rows

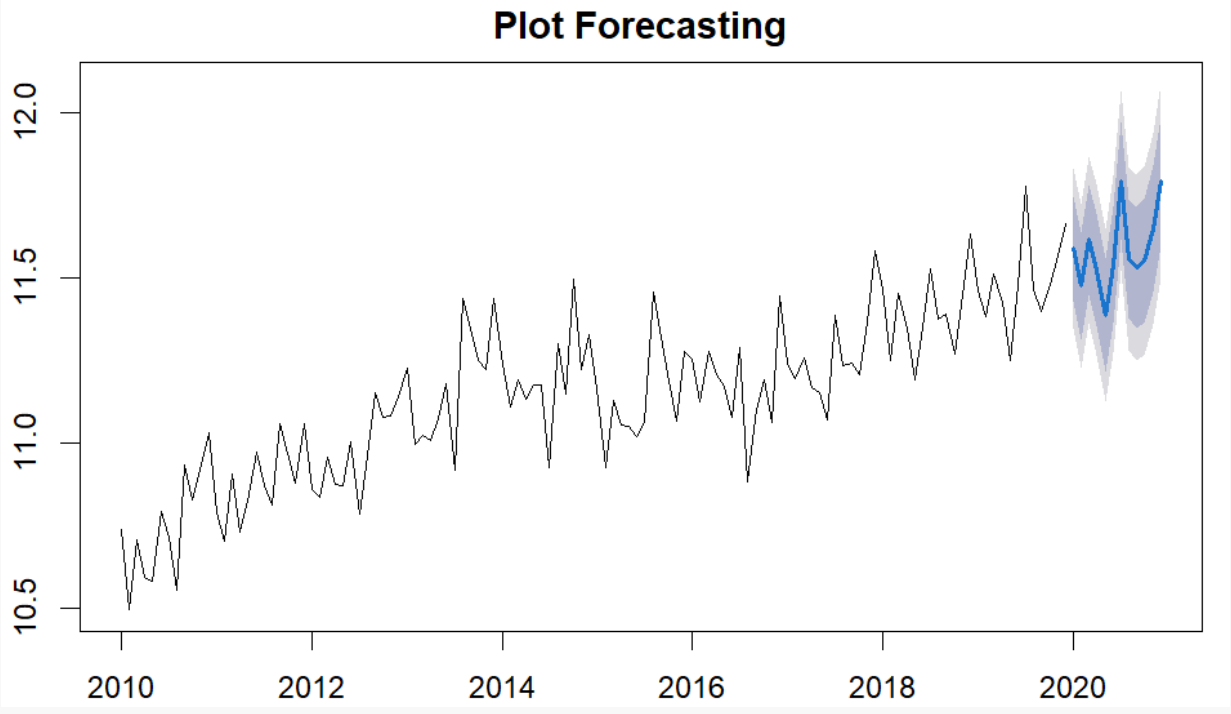
Previous 1 2 Next

Description: df [12 x 5]

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
Nov 2020	11.64770	11.45776	11.83764	11.35721	11.93819
Dec 2020	11.79332	11.60053	11.98611	11.49848	12.08817

11-12 of 12 rows

Previous 1 2 Next



```r
#MAPE
tsdata.fitted <- fitted(Imasima)
nilaimape <- mean(abs(tsdata-(tsdata.fitted))/(tsdata))*100
nilaimape
```

```
## [1] 99.9834
```

```r
#uji residual
t.test(Imasima$residuals, mu = 0, alternative = "two.sided")
```

```
##
## One Sample t-test
##
## data:  Imasima$residuals
## t = -0.22795, df = 119, p-value = 0.8201
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.02366969  0.01878252
## sample estimates:
##    mean of x
## -0.002443587
```

```r
#Kesimpulan
print(Imasima)
```

```
##
## Call:
## arima(x = boxcox, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
period = 12),
##     include.mean = FALSE)
##
## Coefficients:
##           ma1     sma1
##       -0.7919  -0.4352
## s.e.   0.0603   0.1064
##
## sigma^2 estimated as 0.01533:  log likelihood = 69.92,  aic = -135.84
```