R Notebook

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
unempData =read.csv(url("https://data.ny.gov/api/views/5hyu-bdh8/rows.csv?accessType=D
OWNLOAD"),header=TRUE, stringsAsFactors = TRUE)
# Fixing the Date into a YYYY-MM format
unempData$Date <- as.Date(paste(unempData$Year, unempData$Month, sep="-"), "%Y-%M")
# Droping the redundant Year and month
unempData <- subset(unempData, select=-c(Year,Month))
# odering the Data
unempData <- unempData[with(unempData,order(Area,Date)),]
summary(unempData)
```

```
                                  Area        Labor.Force         Employed
Unemployed
 New York City                      :  869   Min.   :   2100   Min.   :   1900   Mi
n.   :   100
 BALANCE OF STATE                   :  519   1st Qu.:  19400   1st Qu.:  18400   1s
t Qu.:  1000
 New York State                     :  519   Median :  43900   Median :  41500   Me
dian :  2300
 New York City Region               :  518   Mean   : 292496   Mean   : 274298   Me
an   : 18200
 Albany-Rensselaer-Schenectady Counties:  350   3rd Qu.: 140700   3rd Qu.: 131900   3r
d Qu.:  7400
 Albany-Schenectady-Troy Metro Area :  350   Max.   :9824400   Max.   :9309000   Ma
x.   :913400
 (Other)                            :7347
0
 Unemployment.Rate      Date
 Min.   : 1.300    Min.   :1976-04-20
 1st Qu.: 4.100    1st Qu.:1997-04-20
 Median : 5.200    Median :2004-04-20
 Mean   : 5.594    Mean   :2004-05-05
 3rd Qu.: 6.800    3rd Qu.:2011-04-20
 Max.   :18.300    Max.   :2019-04-20
```

	Area	Labor.Force	Emplo…	Unemplo…	Unemployment.Rate	
	<fctr>	<int>	<int>	<int>	<dbl>	<d
76256	Yorktown Town	18200	17500	700	3.6	2018-0
76257	Yorktown Town	18100	17400	700	3.9	2018-0
76258	Yorktown Town	18000	17200	800	4.5	2018-0
76259	Yorktown Town	17800	17000	800	4.4	2018-0
76246	Yorktown Town	18100	17500	600	3.5	2019-0
76247	Yorktown Town	18200	17500	600	3.6	2019-0

6 rows

Creating Time Series Data for Albany City

Hide

```
AC_DATA <- subset(unempData, Area=='Albany City')
AC_DATA_UER <- select(AC_DATA,Unemployment.Rate)
AC_DATA_U <- select(AC_DATA,Unemployed)
AC_DATA_E <- select(AC_DATA,Employed)
AC_DATA_LF <- select(AC_DATA,Labor.Force)
ts_AC<-ts(AC_DATA,start=c(1990,1),end=c(2019,2),frequency=12)
ts_AC_UER<-ts(AC_DATA_UER,start=c(1990,1),end=c(2019,2),frequency=12)
ts_AC_U<-ts(AC_DATA_U,start=c(1990,1),end=c(2019,2),frequency=12)
ts_AC_E<-ts(AC_DATA_E,start=c(1990,1),end=c(2019,2),frequency=12)
ts_AC_LF<-ts(AC_DATA_LF,start=c(1990,1),end=c(2019,2),frequency=12)
library(ggplot2)
head(ts_AC_UER)
```

```
[1] 5.0 4.7 4.2 4.3 3.8 3.9
```

Les Up Understnad the Albany City Row Data ie AC_DATA and corresponding time series data ts_AC

Hide

```
print(" Length ")
```

```
[1] " Length "
```

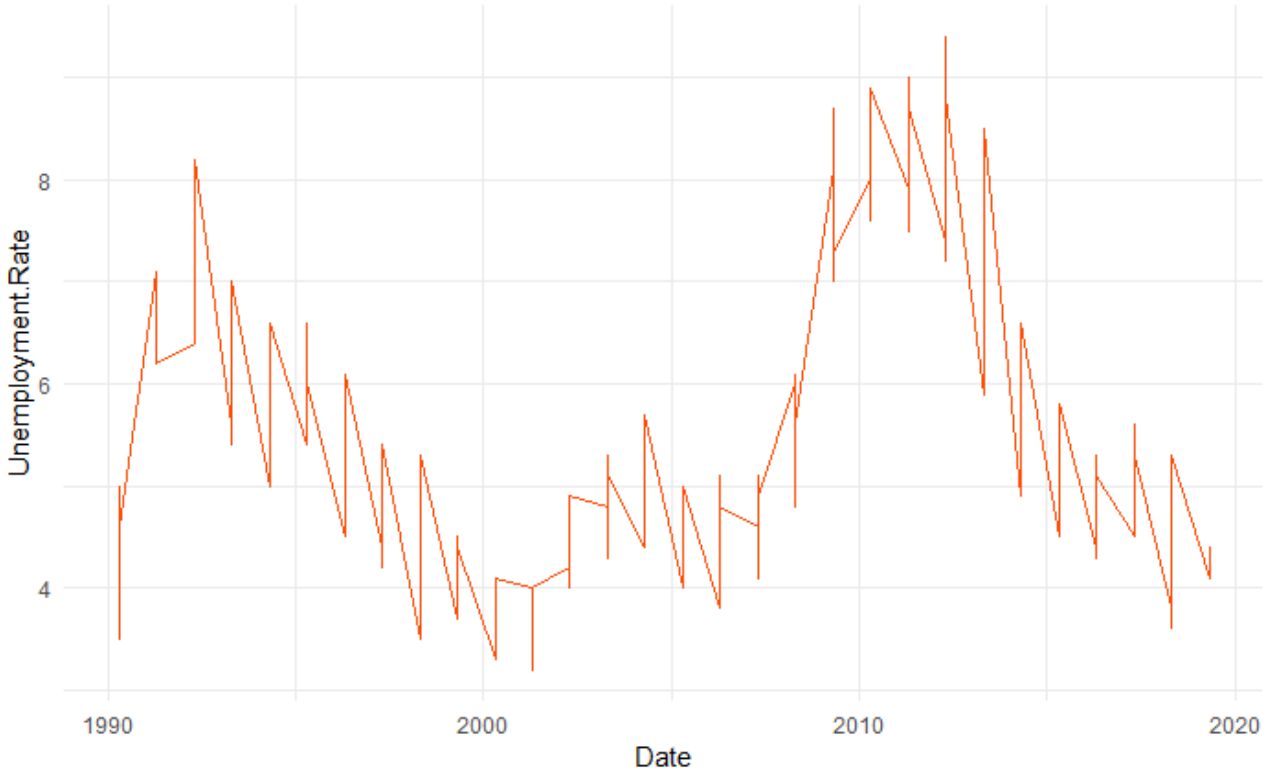
Hide

```
length(ts_AC)
```

```
[1] 2100
```

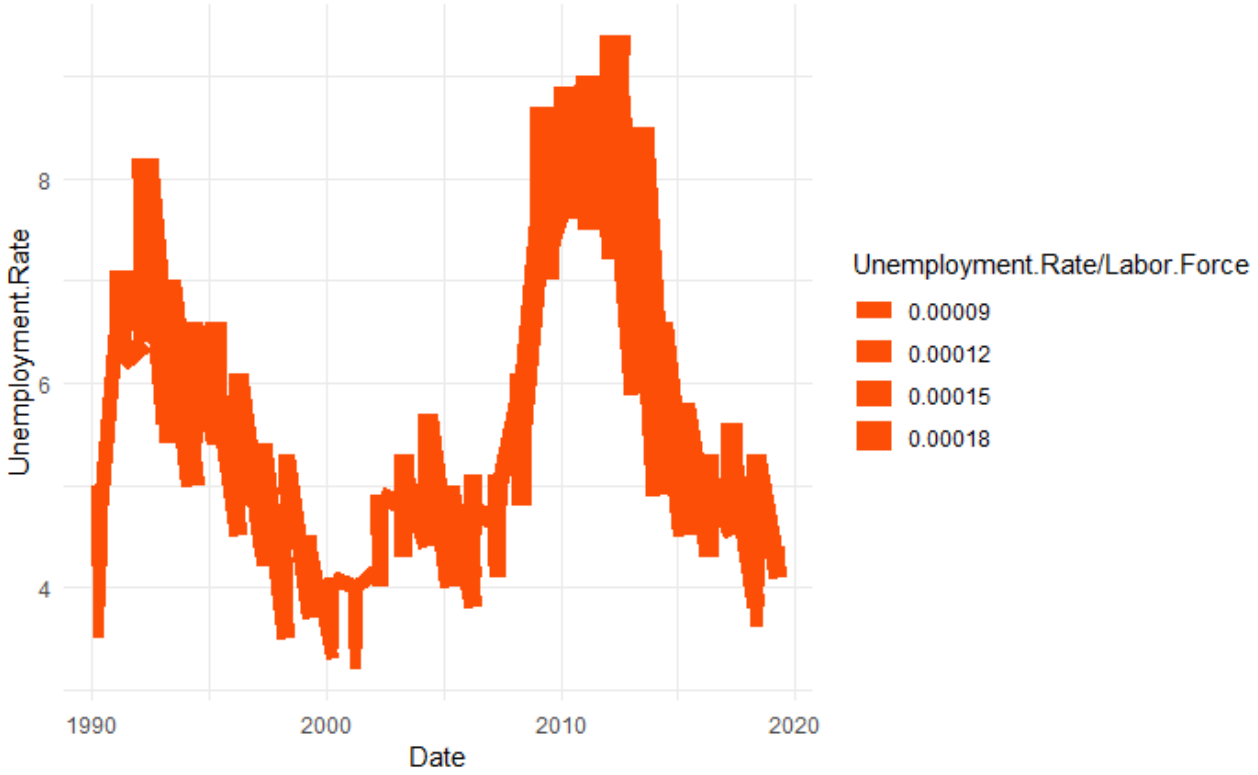
Hide

```
ggplot(data = AC_DATA, aes(x = Date, y = Unemployment.Rate)) +geom_line( color = "#FC4
E07")
```



Hide

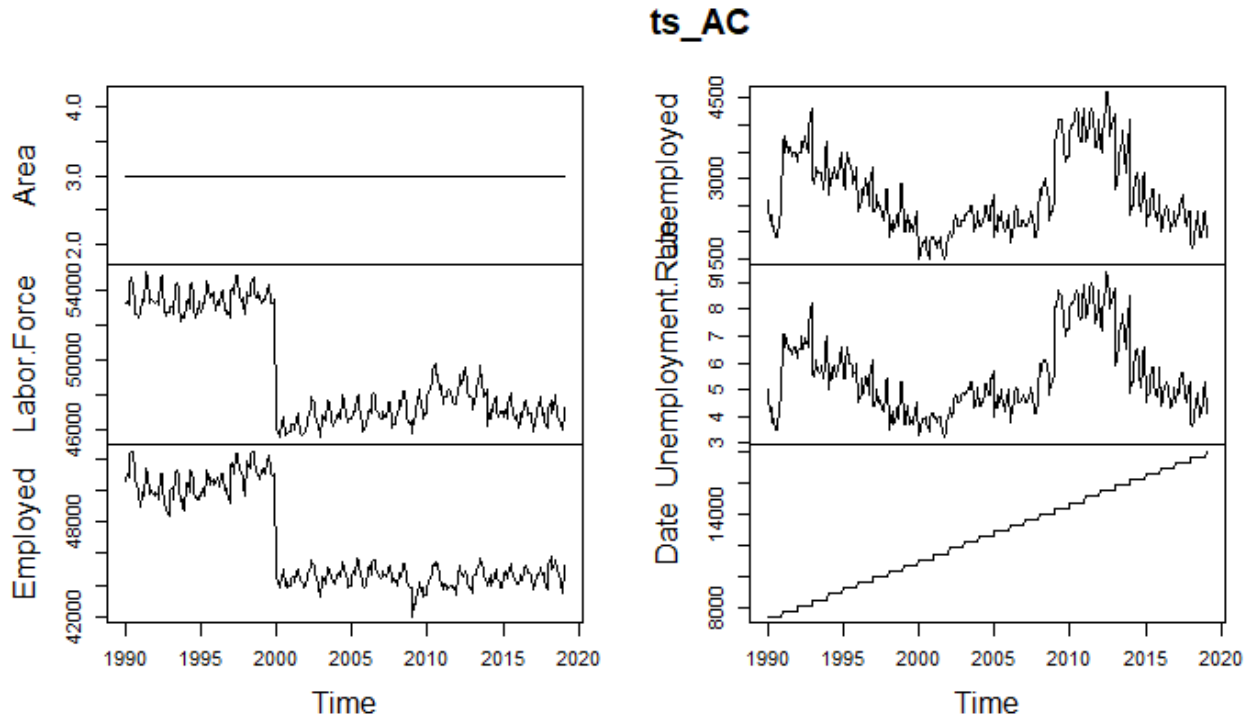
```
ggplot(data = AC_DATA, aes(x = Date, y = Unemployment.Rate)) +geom_line(aes(size = Une
mployment.Rate/Labor.Force), color = "#FC4E07")
```



We could see some trems in Row Data Lets See what best a Time Series Object can offer

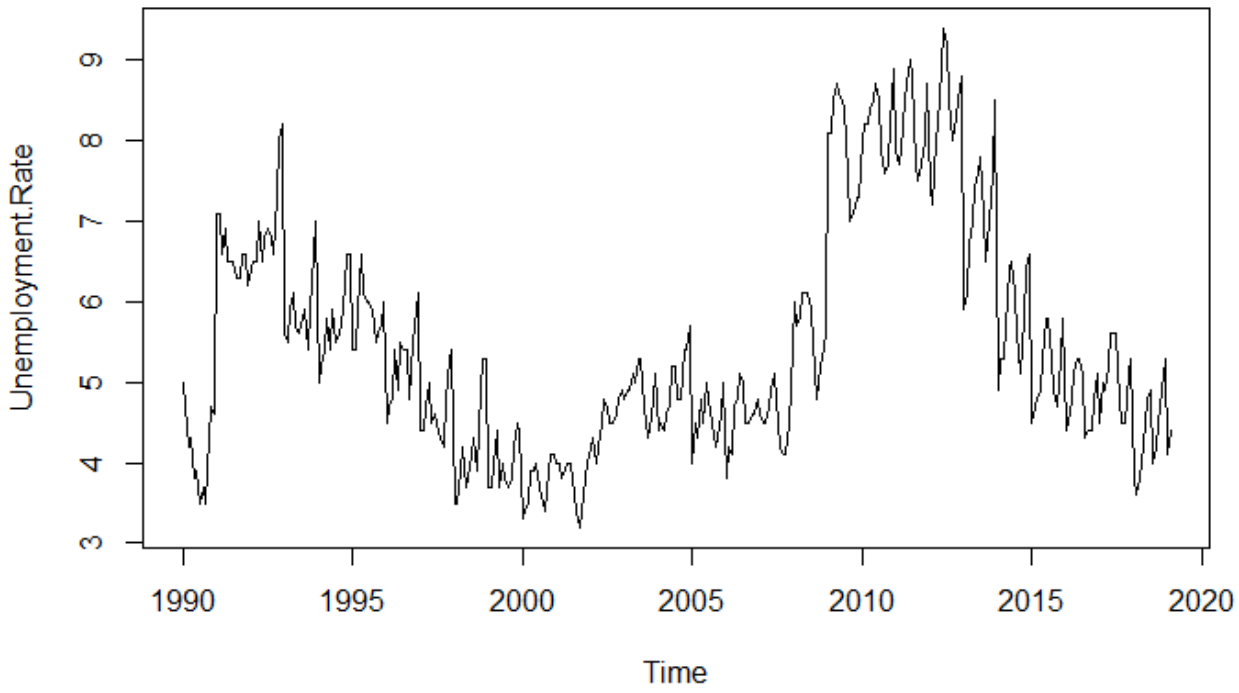
Hide

```
plot(ts_AC)
```



Hide

```
plot(ts_AC_UER)
```



This data is Equlay spaced with out any missing value That could also be established from code

Hide

```
print("Start")
```

```
[1] "Start"
```

Hide

```
start(ts_AC)
```

```
[1] 1990    1
```

Hide

```
print("end")
```

```
[1] "end"
```

Hide

```
end(ts_AC)
```

```
[1] 2019    2
```

Hide

```
print("deltat")
```

```
[1] "deltat"
```

Hide

```
# function returns the fixed time interval between observations
deltat(ts_AC)
```

```
[1] 0.08333333
```

Hide

```
print("frequency")
```

```
[1] "frequency"
```

Hide

```
#function returns the number of observations per unit time
frequency(ts_AC)
```

```
[1] 12
```

Hide

```
print("time")
```

```
[1] "time"
```

Hide

```
time(ts_AC)
```

```
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep
Oct      Nov
1990 1990.000 1990.083 1990.167 1990.250 1990.333 1990.417 1990.500 1990.583 1990.667
1990.750 1990.833
1991 1991.000 1991.083 1991.167 1991.250 1991.333 1991.417 1991.500 1991.583 1991.667
1991.750 1991.833
1992 1992.000 1992.083 1992.167 1992.250 1992.333 1992.417 1992.500 1992.583 1992.667
1992.750 1992.833
1993 1993.000 1993.083 1993.167 1993.250 1993.333 1993.417 1993.500 1993.583 1993.667
1993.750 1993.833
1994 1994.000 1994.083 1994.167 1994.250 1994.333 1994.417 1994.500 1994.583 1994.667
1994.750 1994.833
1995 1995.000 1995.083 1995.167 1995.250 1995.333 1995.417 1995.500 1995.583 1995.667
1995.750 1995.833
1996 1996.000 1996.083 1996.167 1996.250 1996.333 1996.417 1996.500 1996.583 1996.667
1996.750 1996.833
1997 1997.000 1997.083 1997.167 1997.250 1997.333 1997.417 1997.500 1997.583 1997.667
1997.750 1997.833
1998 1998.000 1998.083 1998.167 1998.250 1998.333 1998.417 1998.500 1998.583 1998.667
1998.750 1998.833
1999 1999.000 1999.083 1999.167 1999.250 1999.333 1999.417 1999.500 1999.583 1999.667
1999.750 1999.833
2000 2000.000 2000.083 2000.167 2000.250 2000.333 2000.417 2000.500 2000.583 2000.667
2000.750 2000.833
2001 2001.000 2001.083 2001.167 2001.250 2001.333 2001.417 2001.500 2001.583 2001.667
2001.750 2001.833
2002 2002.000 2002.083 2002.167 2002.250 2002.333 2002.417 2002.500 2002.583 2002.667
2002.750 2002.833
2003 2003.000 2003.083 2003.167 2003.250 2003.333 2003.417 2003.500 2003.583 2003.667
2003.750 2003.833
2004 2004.000 2004.083 2004.167 2004.250 2004.333 2004.417 2004.500 2004.583 2004.667
2004.750 2004.833
2005 2005.000 2005.083 2005.167 2005.250 2005.333 2005.417 2005.500 2005.583 2005.667
2005.750 2005.833
2006 2006.000 2006.083 2006.167 2006.250 2006.333 2006.417 2006.500 2006.583 2006.667
2006.750 2006.833
2007 2007.000 2007.083 2007.167 2007.250 2007.333 2007.417 2007.500 2007.583 2007.667
2007.750 2007.833
2008 2008.000 2008.083 2008.167 2008.250 2008.333 2008.417 2008.500 2008.583 2008.667
2008.750 2008.833
2009 2009.000 2009.083 2009.167 2009.250 2009.333 2009.417 2009.500 2009.583 2009.667
2009.750 2009.833
2010 2010.000 2010.083 2010.167 2010.250 2010.333 2010.417 2010.500 2010.583 2010.667
2010.750 2010.833
2011 2011.000 2011.083 2011.167 2011.250 2011.333 2011.417 2011.500 2011.583 2011.667
2011.750 2011.833
2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583 2012.667
2012.750 2012.833
```

```
2013 2013.000 2013.083 2013.167 2013.250 2013.333 2013.417 2013.500 2013.583 2013.667
2013.750 2013.833
2014 2014.000 2014.083 2014.167 2014.250 2014.333 2014.417 2014.500 2014.583 2014.667
2014.750 2014.833
2015 2015.000 2015.083 2015.167 2015.250 2015.333 2015.417 2015.500 2015.583 2015.667
2015.750 2015.833
2016 2016.000 2016.083 2016.167 2016.250 2016.333 2016.417 2016.500 2016.583 2016.667
2016.750 2016.833
2017 2017.000 2017.083 2017.167 2017.250 2017.333 2017.417 2017.500 2017.583 2017.667
2017.750 2017.833
2018 2018.000 2018.083 2018.167 2018.250 2018.333 2018.417 2018.500 2018.583 2018.667
2018.750 2018.833
2019 2019.000 2019.08
3
          Dec
1990 1990.917
1991 1991.917
1992 1992.917
1993 1993.917
1994 1994.917
1995 1995.917
1996 1996.917
1997 1997.917
1998 1998.917
1999 1999.917
2000 2000.917
2001 2001.917
2002 2002.917
2003 2003.917
2004 2004.917
2005 2005.917
2006 2006.917
2007 2007.917
2008 2008.917
2009 2009.917
2010 2010.917
2011 2011.917
2012 2012.917
2013 2013.917
2014 2014.917
2015 2015.917
2016 2016.917
2017 2017.917
2018 2018.917
2019
```

Hide

```
print("cycle")
```

```
[1] "cycle"
```

Hide

```
#cycle() function returns the position in the cycle of each observation
cycle(ts_AC)
```

```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1990   1   2   3   4   5   6   7   8   9  10  11  12
1991   1   2   3   4   5   6   7   8   9  10  11  12
1992   1   2   3   4   5   6   7   8   9  10  11  12
1993   1   2   3   4   5   6   7   8   9  10  11  12
1994   1   2   3   4   5   6   7   8   9  10  11  12
1995   1   2   3   4   5   6   7   8   9  10  11  12
1996   1   2   3   4   5   6   7   8   9  10  11  12
1997   1   2   3   4   5   6   7   8   9  10  11  12
1998   1   2   3   4   5   6   7   8   9  10  11  12
1999   1   2   3   4   5   6   7   8   9  10  11  12
2000   1   2   3   4   5   6   7   8   9  10  11  12
2001   1   2   3   4   5   6   7   8   9  10  11  12
2002   1   2   3   4   5   6   7   8   9  10  11  12
2003   1   2   3   4   5   6   7   8   9  10  11  12
2004   1   2   3   4   5   6   7   8   9  10  11  12
2005   1   2   3   4   5   6   7   8   9  10  11  12
2006   1   2   3   4   5   6   7   8   9  10  11  12
2007   1   2   3   4   5   6   7   8   9  10  11  12
2008   1   2   3   4   5   6   7   8   9  10  11  12
2009   1   2   3   4   5   6   7   8   9  10  11  12
2010   1   2   3   4   5   6   7   8   9  10  11  12
2011   1   2   3   4   5   6   7   8   9  10  11  12
2012   1   2   3   4   5   6   7   8   9  10  11  12
2013   1   2   3   4   5   6   7   8   9  10  11  12
2014   1   2   3   4   5   6   7   8   9  10  11  12
2015   1   2   3   4   5   6   7   8   9  10  11  12
2016   1   2   3   4   5   6   7   8   9  10  11  12
2017   1   2   3   4   5   6   7   8   9  10  11  12
2018   1   2   3   4   5   6   7   8   9  10  11  12
2019   1   2
```

Hide

```
print("The Total Missing Values are")
```

```
[1] "The Total Missing Values are"
```

Hide

```
sum( is.na( ts_AC ) )
```

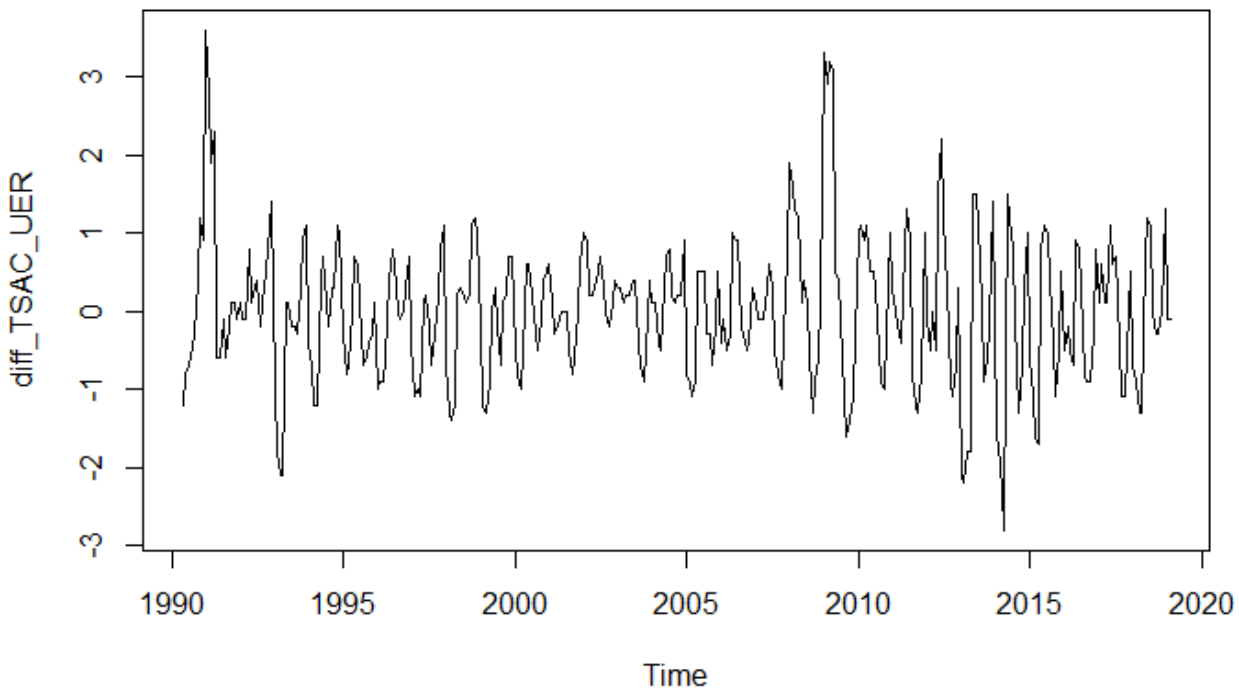
```
[1] 0
```

As We See there are No Missing value.. We need not impute it

As we have seen some trend, in the time sesies, some seasonal factor also. We can now try out diff to remove the trend

Hide

```
diff_TSAC_UER <- diff(ts_AC_UER,lag = 4)

# Plot dz
ts.plot(diff_TSAC_UER)
```



Hide

```
# View the length of z and dz, respectively
length(diff_TSAC)
```

```
[1] 2076
```

Hide

```
length(ts_AC)
```

```
[1] 2100
```

We have lost some 24 rows on applying Diff since we have a Lag of 4. Now all what we see is some white noise.

Lets now fit the Time series data to Arima model and find the Mean and variance

Hide

```
diff_TSAC_UER
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1990                    -1.2 -0.8 -0.7 -0.6 -0.3  0.4  1.2  0.9
1991  3.6  2.8  1.9  2.3 -0.6 -0.6 -0.1 -0.6 -0.2  0.1  0.1 -0.1
1992  0.1 -0.1 -0.1  0.8  0.1  0.3  0.4 -0.2  0.1  0.5  1.1  1.4
1993 -1.0 -1.8 -2.1 -2.1  0.1  0.1 -0.2 -0.2 -0.3  0.3  0.9  1.1
1994 -0.4 -0.7 -1.2 -1.2  0.4  0.7  0.1 -0.2  0.3  0.3  1.1  1.0
1995 -0.3 -0.8 -0.7  0.0  0.7  0.6  0.1 -0.7 -0.6 -0.4 -0.3  0.1
1996 -1.0 -0.9 -0.9 -0.6  0.4  0.8  0.6  0.0 -0.1  0.0  0.3  0.7
1997 -0.4 -1.1 -1.0 -1.1  0.1  0.2 -0.2 -0.7 -0.3  0.2  0.8  1.1
1998 -0.7 -1.3 -1.4 -1.2  0.2  0.3  0.2  0.1  0.2  1.1  1.2  1.0
1999 -0.2 -1.2 -1.3 -0.9  0.0  0.3 -0.2 -0.7  0.1  0.2  0.7  0.7
2000 -0.5 -0.8 -1.0 -0.5  0.6  0.6  0.2 -0.3 -0.5 -0.1  0.4  0.5
2001  0.6  0.1 -0.3 -0.2  0.0  0.0  0.0 -0.5 -0.8 -0.6  0.0  0.6
2002  1.0  0.9  0.2  0.2  0.3  0.5  0.7  0.3  0.0 -0.2  0.1  0.4
2003  0.3  0.3  0.1  0.2  0.2  0.4  0.4 -0.4 -0.7 -0.9 -0.5  0.4
2004  0.1  0.1 -0.4 -0.5  0.3  0.7  0.8  0.2  0.1  0.2  0.2  0.9
2005 -0.8 -0.9 -1.1 -0.9  0.5  0.5  0.5 -0.3 -0.3 -0.7 -0.2  0.5
2006 -0.4 -0.1 -0.5 -0.3  1.0  0.9  0.9 -0.2 -0.3 -0.5 -0.4  0.3
2007  0.1 -0.1 -0.1 -0.1  0.3  0.6  0.3 -0.5 -0.8 -1.0 -0.3  0.7
2008  1.9  1.6  1.3  1.2  0.1  0.4  0.1 -0.6 -1.3 -0.9 -0.6  0.1
2009  3.3  2.9  3.2  3.1  0.5  0.4 -0.1 -1.1 -1.6 -1.4 -1.1 -0.3
2010  1.0  1.1  0.9  1.1  0.5  0.5  0.3 -0.5 -0.9 -1.0 -0.3  1.0
2011  0.3  0.0 -0.3 -0.4  0.8  1.3  0.9 -0.8 -1.2 -1.3 -0.8  1.0
2012 -0.1 -0.5  0.0 -0.5  1.5  2.2  1.2  0.3 -0.9 -1.1 -0.7  0.3
2013 -2.1 -2.2 -1.8 -1.8  1.5  1.5  1.1  0.1 -0.9 -0.6 -0.2  1.4
2014 -1.6 -1.7 -2.3 -2.8  1.5  1.2  0.8 -0.1 -1.3 -0.7  0.3  1.0
2015 -0.6 -1.1 -1.6 -1.7  0.8  1.1  1.0  0.4 -0.4 -1.1 -0.5  0.5
2016 -0.5 -0.2 -0.5 -0.7  0.9  0.8  0.3 -0.8 -0.9 -0.9 -0.3  0.8
2017  0.1  0.6  0.1  0.1  1.1  0.6  0.7 -0.4 -1.1 -1.1 -0.5  0.5
2018 -0.7 -0.9 -1.3 -1.3  0.7  1.2  1.1  0.0 -0.3 -0.3  0.0  1.3
2019 -0.1 -0.1
```

Hide

```
# Fit the WN model to y using the arima command
UER_model <- arima(diff_TSAC_UER, order = c(0, 0, 0))
mean(diff_TSAC_UER)
```

```
[1] 0.001445087
```

Hide

```
var(diff_TSAC_UER)
```

```
                 Unemployment.Rate
Unemployment.Rate          0.837824
```

Hide

```
print("Model")
```

```
[1] "Model"
```

Hide

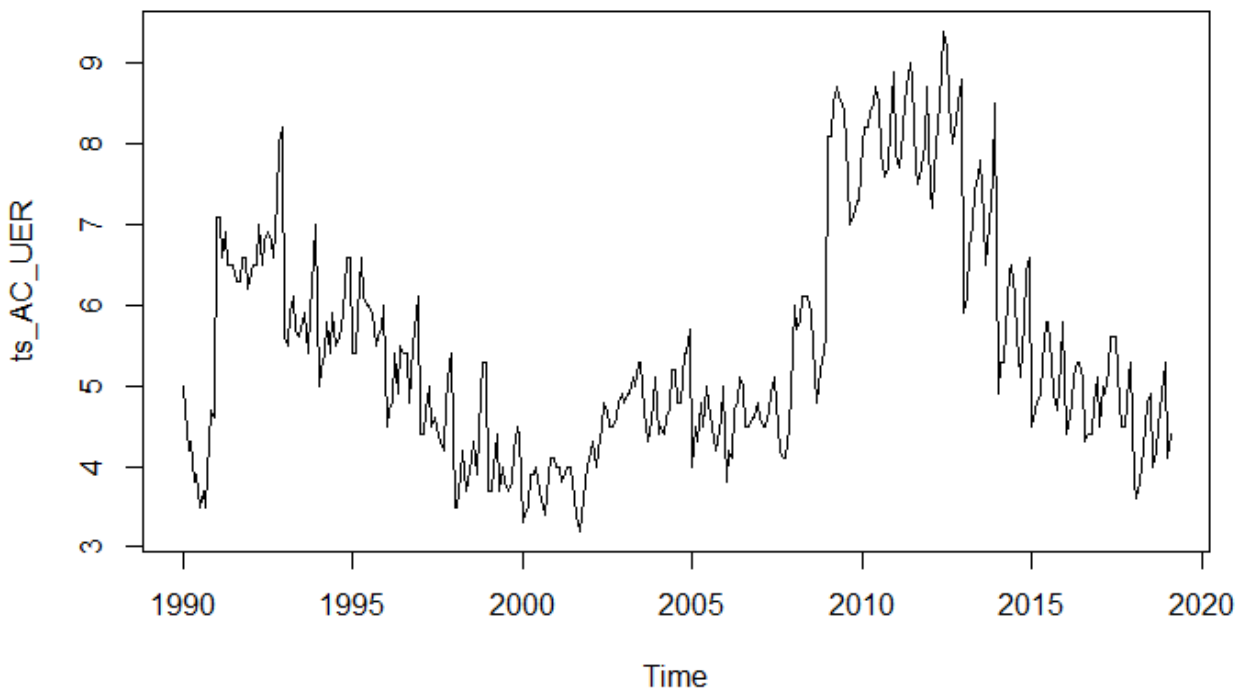
```
UER_model
```

```

Call:
arima(x = diff_TSAC_UER, order = c(0, 0, 0))

Coefficients:
      intercept
         0.0014
s.e.     0.0491

sigma^2 estimated as 0.8354:  log likelihood = -459.84,  aic = 923.68
```

Lets Plot and see how does our Model fits.

Hide

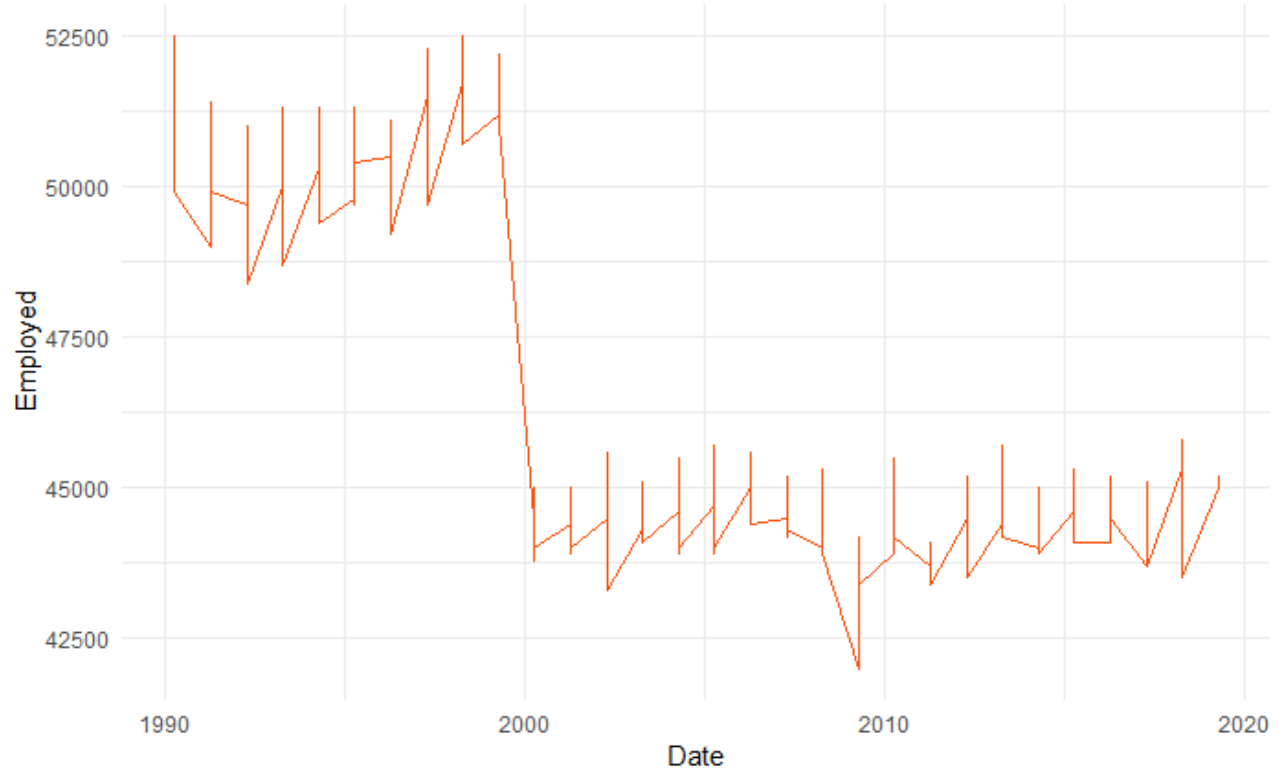
```
ts.plot(ts_AC_UER)
modelCoef <- UER_model$coef
abline(0, modelCoef)
```



Lets Do that with another variable, Employed

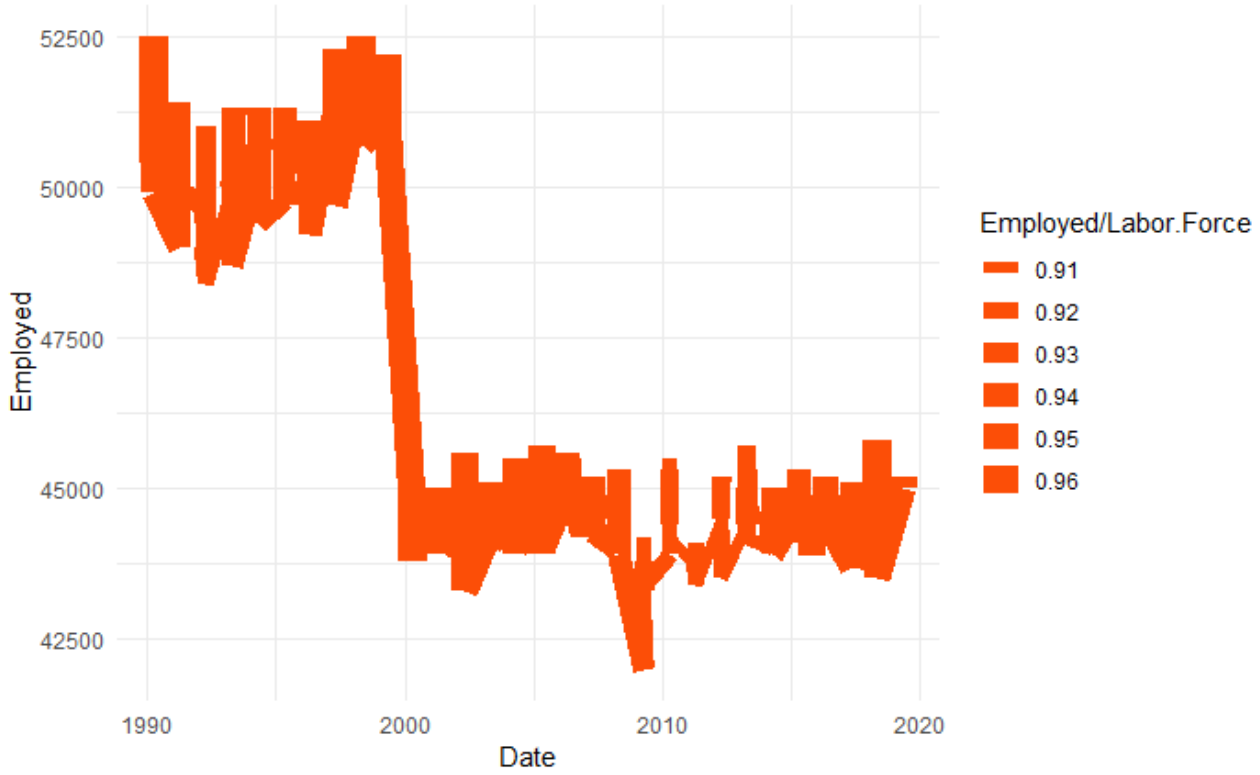
Hide

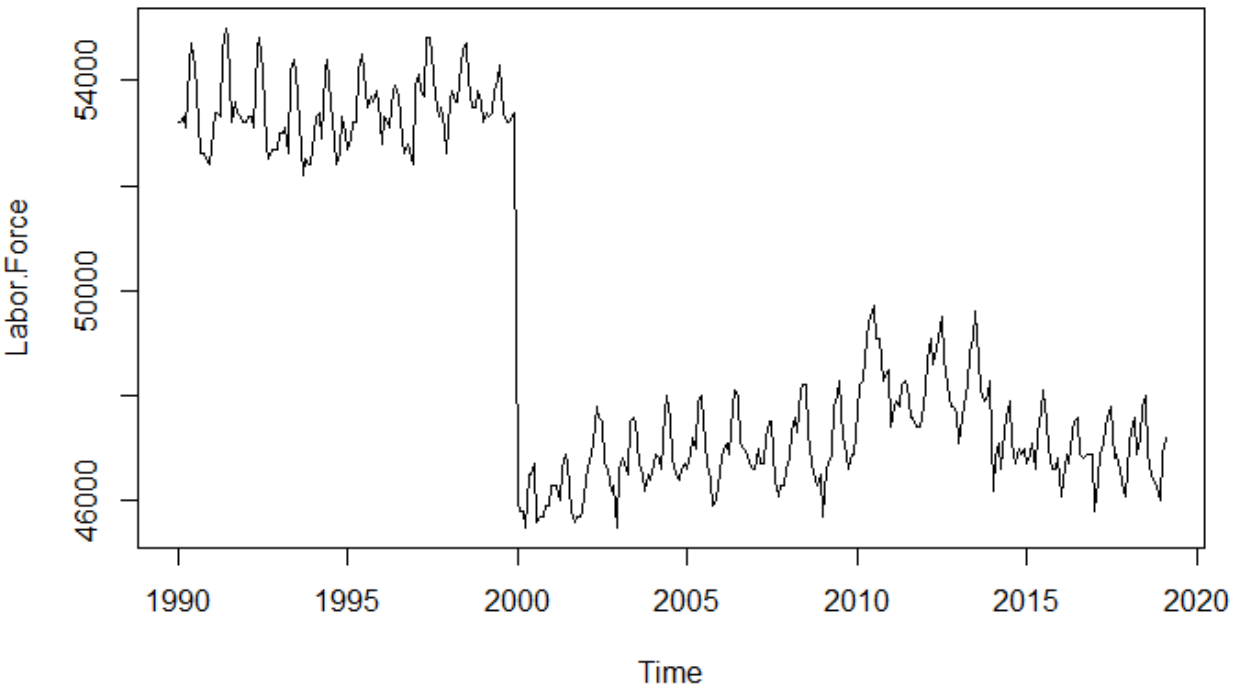
```
ggplot(data = AC_DATA, aes(x = Date, y = Employed)) +geom_line( color = "#FC4E07")
```



Hide

```
ggplot(data = AC_DATA, aes(x = Date, y = Employed)) +geom_line(aes(size = Employed/Labor.Force), color = "#FC4E07")
```

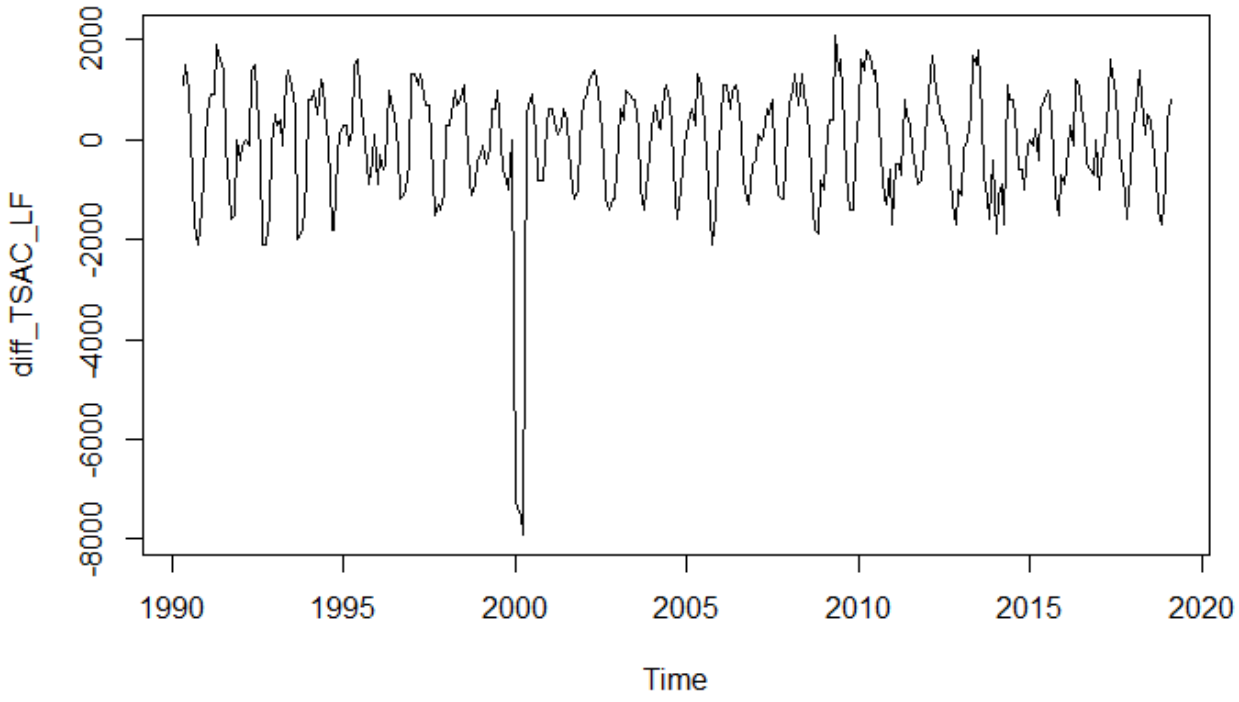




Some thing went wrong in year 2000 in our data set. Lets Look further.

Hide

```
diff_TSAC_LF <- diff(ts_AC_LF,lag = 4)
ts.plot(diff_TSAC_LF)
```



Hide

```
length(diff_TSAC_LF)
```

```
[1] 346
```

Hide

```
length(ts_AC)
```

```
[1] 2100
```

Hide

```
LF_model <- arima(diff_TSAC_LF, order = c(0, 0, 0))
mean(diff_TSAC_LF)
```

```
[1] -76.30058
```

Hide

```
var(diff_TSAC_LF)
```

```
            Labor.Force
Labor.Force     1561582
```

Hide

```
print("Model_LF")
```

```
[1] "Model_LF"
```

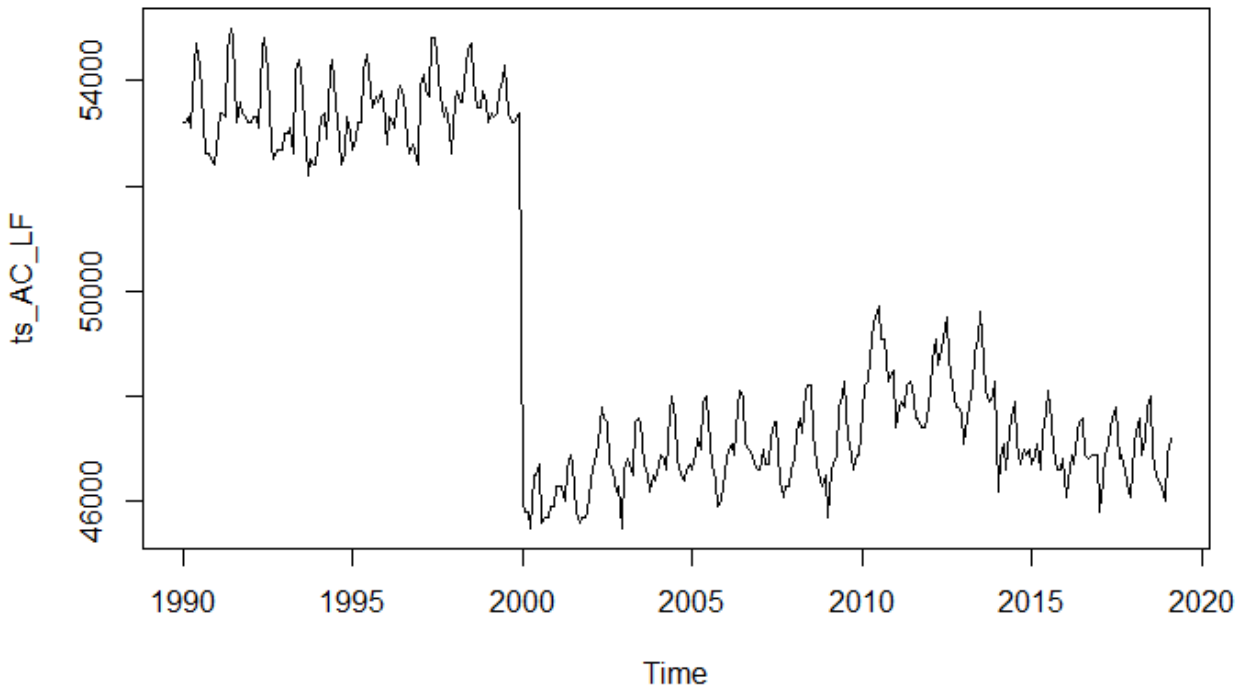
Hide

```
LF_model
```

```

Call:
arima(x = diff_TSAC_LF, order = c(0, 0, 0))

Coefficients:
      intercept
       -76.3006
s.e.    67.0835

sigma^2 estimated as 1557068:  log likelihood = -2957.64,  aic = 5919.28
```

Hide

```
ts.plot(ts_AC_LF)
modelCoef_LF <- LF_model$coef
abline(50678, modelCoef_LF)
```



When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.