Time series Analysis - Assignment 3 - Predicting Sales for Superstore

Jinping Bai, Joshua Dalphy, Choongil Kim and Gouri Kulkarni

Thursday, November 12th 2020

Contents


```
################## Time Series Analysis ############################
```

0.0.1 Time Series Analysis

```
# Load in the raw data
data=read.csv(file.choose(), header = TRUE, na.strings = c("NA","","#NA"))
# Create a copy of the data
copy_data = data
```

```
# Delete Columns
raw_data2 = copy_data[ , -c(6,7,12,15,18)]

# Specify columns as dates
raw_data2$Order.Date = as.Date(raw_data2$Order.Date,"%d/%m/%Y")
raw_data2$Ship.Date = as.Date(raw_data2$Ship.Date,"%d/%m/%Y")
```

Number of rows and columns

```
#check # of rows
ncol(raw_data2)
```

```
## [1] 19
```

```
nrow(raw_data2)
```

```
## [1] 51290
```

Plot of US sales

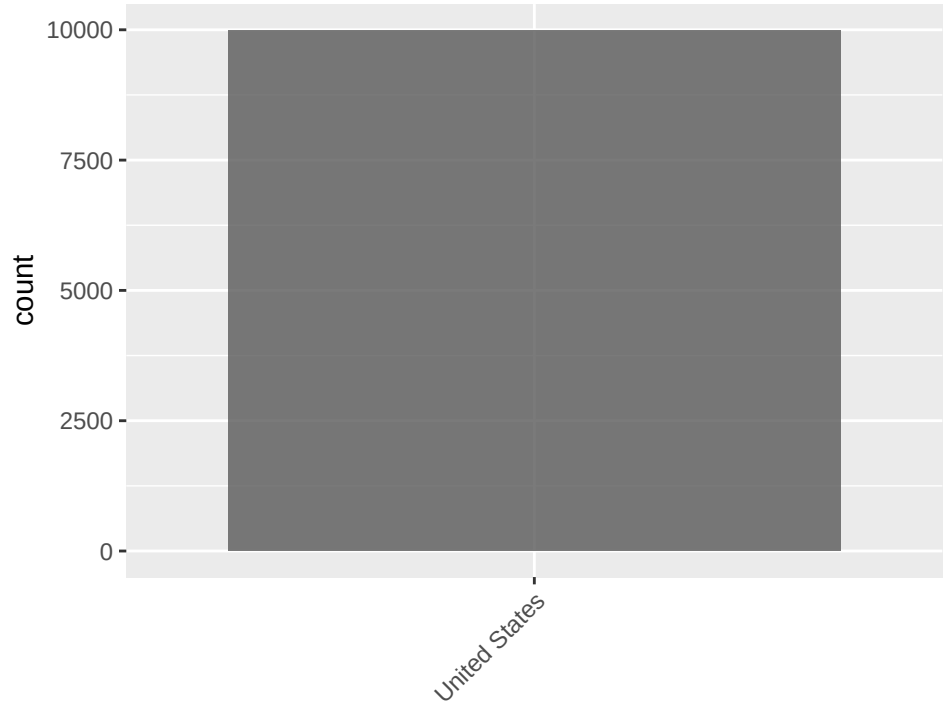
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

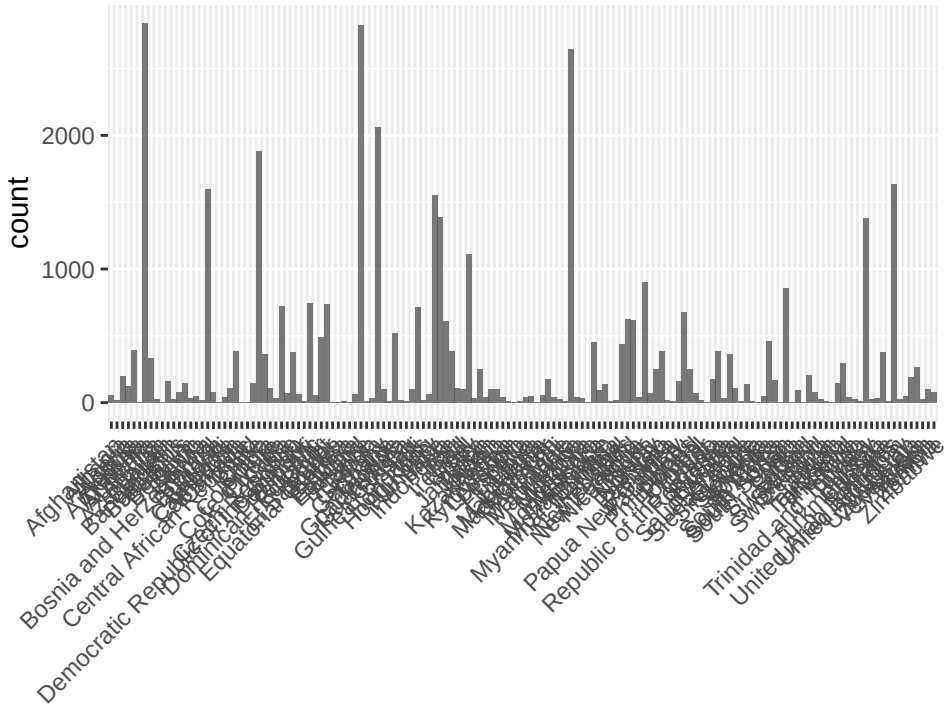
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

p1 <- raw_data2 %>%
  filter(Country == "United States") %>%
  ggplot(aes(x = "United States", fill = Sales)) +
    geom_bar(alpha = 0.8) +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
    guides(fill = FALSE) +
    labs(x = "")
p1
```



Sales of countries other than the US

```
p2 <- raw_data2 %>%
  filter(Country != "United States") %>%
  ggplot(aes(x = Country, fill = Sales)) +
    geom_bar(alpha = 0.8) +
    theme(legend.position = "right") +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
    labs(x = "",
         fill = "")

p2
```



The main customer base is the US

We will remove the countries under a certain level of sales.

The range of sales is :

```
range(raw_data2$Sales)
```

```
## [1]     0.444 22638.480
```

Remove the rows which have Sales below 5000

```
select_raw_data2<-subset(raw_data2, Sales >= 5000)
head(select_raw_data2)
```

```
##      Row.ID        Order.ID Order.Date  Ship.Date      Ship.Mode     Segment
## 2256  31462  CA-2011-139892 2011-09-08 2011-09-12 Standard Class    Consumer
## 2274  13560 ES-2011-3248922 2011-09-08 2011-09-11   Second Class   Corporate
## 2757  27720   ID-2011-64599 2011-02-10       <NA> Standard Class   Corporate
## 2858  12051 ES-2011-2257437 2011-08-10       <NA> Standard Class    Consumer
## 2859  49997    IV-2011-9090 2011-08-10       <NA> Standard Class    Consumer
## 3189  27407   IN-2011-35178 2011-11-11       <NA> Standard Class Home Office
##             City    State        Country Market     Region        Category
## 2256 San Antonio    Texas  United States     US    Central      Technology
## 2274        Lugo  Galicia          Spain     EU      South Office Supplies
## 2757        Fuji Shizuoka          Japan   APAC North Asia      Technology
## 2858      London  England United Kingdom     EU      North      Technology
## 2859     Abidjan  Lagunes  Cote d'Ivoire Africa     Africa      Technology
## 3189      Suzhou    Gansu          China   APAC North Asia      Technology
##      Sub.Category    Sales Quantity Discount    Profit Shipping.Cost
## 2256     Machines 8159.952        8      0.4 -1359.992        342.11
## 2274   Appliances 6517.080       12      0.0  2476.440         28.74
```

```
## 2757       Phones 6998.640       11      0.0  2939.310        413.80
## 2858       Phones 5276.988        9      0.1  1758.888        454.81
## 2859       Phones 5100.000        8      0.0  1428.000         85.22
## 3189       Phones 5725.350        9      0.0  1602.990        302.61
##      Order.Priority
## 2256         Medium
## 2274       Critical
## 2757         Medium
## 2858         Medium
## 2859         Medium
## 3189         Medium
```

The new range of Sales and countries that have sales above 5000

```
range(select_raw_data2$Sales)
```

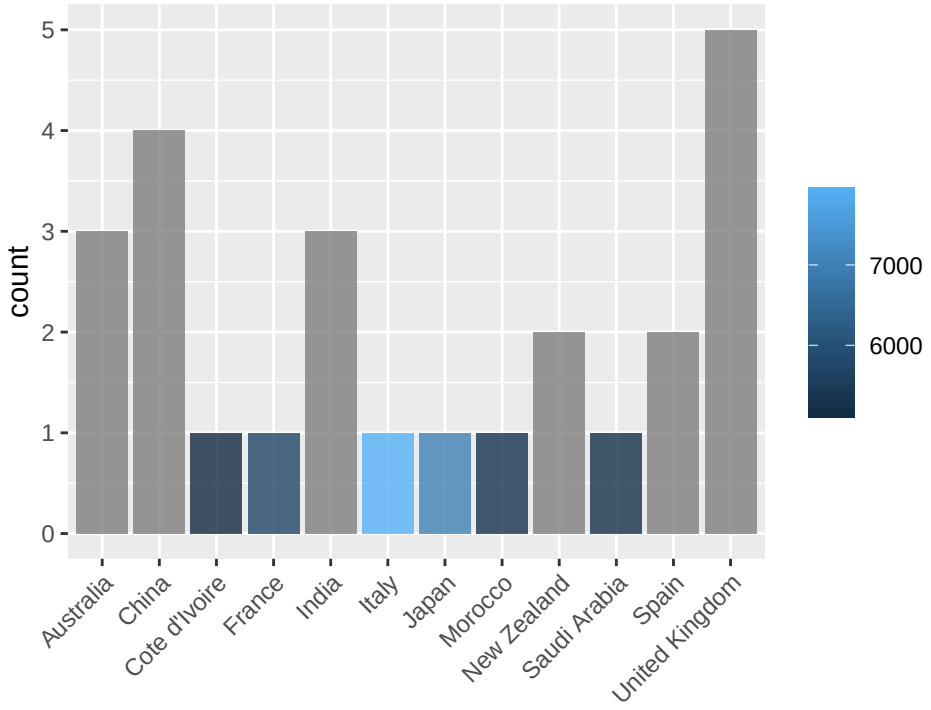
```
## [1]  5049.00 22638.48
```

```
#unique countries
unique(select_raw_data2$Country)
```

```
##  [1] "United States"  "Spain"          "Japan"          "United Kingdom"
##  [5] "Cote d'Ivoire"  "China"          "Australia"      "India"
##  [9] "Italy"          "New Zealand"    "France"         "Saudi Arabia"
## [13] "Morocco"
```

Sales in countries other than the US with sales >= 5000

```
p2 <- select_raw_data2 %>%
  filter(Country != "United States") %>%
  ggplot(aes(x = Country, fill = Sales)) +
    geom_bar(alpha = 0.8) +
    #scale_fill_manual(values = palette_light()) +
    #theme_tq() +
    theme(legend.position = "right") +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
    labs(x = "",
         fill = "")

p2
```



Generate Time series plot of the entire dataset ‘

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

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

```
select_raw_data2$Order.Date <- as.yearmon(select_raw_data2$Order.Date)

head(select_raw_data2)
```

```
##      Row.ID        Order.ID Order.Date  Ship.Date      Ship.Mode     Segment
## 2256  31462  CA-2011-139892   Sep 2011 2011-09-12 Standard Class    Consumer
## 2274  13560 ES-2011-3248922   Sep 2011 2011-09-11   Second Class   Corporate
## 2757  27720   ID-2011-64599   Feb 2011       <NA> Standard Class   Corporate
## 2858  12051 ES-2011-2257437   Aug 2011       <NA> Standard Class    Consumer
## 2859  49997    IV-2011-9090   Aug 2011       <NA> Standard Class    Consumer
## 3189  27407   IN-2011-35178   Nov 2011       <NA> Standard Class Home Office
##             City    State        Country Market     Region        Category
## 2256 San Antonio    Texas  United States     US    Central      Technology
## 2274        Lugo  Galicia          Spain     EU      South Office Supplies
## 2757        Fuji Shizuoka          Japan   APAC North Asia      Technology
## 2858      London  England United Kingdom     EU      North      Technology
## 2859     Abidjan  Lagunes  Cote d'Ivoire Africa     Africa      Technology
## 3189      Suzhou    Gansu          China   APAC North Asia      Technology
##      Sub.Category    Sales Quantity Discount    Profit Shipping.Cost
## 2256     Machines 8159.952        8      0.4 -1359.992        342.11
```

```
## 2274   Appliances 6517.080       12      0.0  2476.440         28.74
## 2757       Phones 6998.640       11      0.0  2939.310        413.80
## 2858       Phones 5276.988        9      0.1  1758.888        454.81
## 2859       Phones 5100.000        8      0.0  1428.000         85.22
## 3189       Phones 5725.350        9      0.0  1602.990        302.61
##      Order.Priority
## 2256         Medium
## 2274       Critical
## 2757         Medium
## 2858         Medium
## 2859         Medium
## 3189         Medium
```

Filter the Order.Date and Sales to used for the timeseries

```
summary <- select_raw_data2 %>%
  select(Order.Date, Sales) %>%
  filter(Order.Date >= "2011-01-01") %>%
  filter(Order.Date <= "2014-12-31")

head(summary)
```

```
##   Order.Date    Sales
## 1   Sep 2011 8159.952
## 2   Sep 2011 6517.080
## 3   Feb 2011 6998.640
## 4   Aug 2011 5276.988
## 5   Aug 2011 5100.000
## 6   Nov 2011 5725.350
```

```
tseries <- ts(summary, start = c(2011, 1), end = c(2014,12), frequency = 12)
```

Check the start, end , frequency of the time series

```
## [1] 2011    1
```

```
## [1] 2014   12
```

```
## [1] 0.08333333
```

```
## [1] 12
```

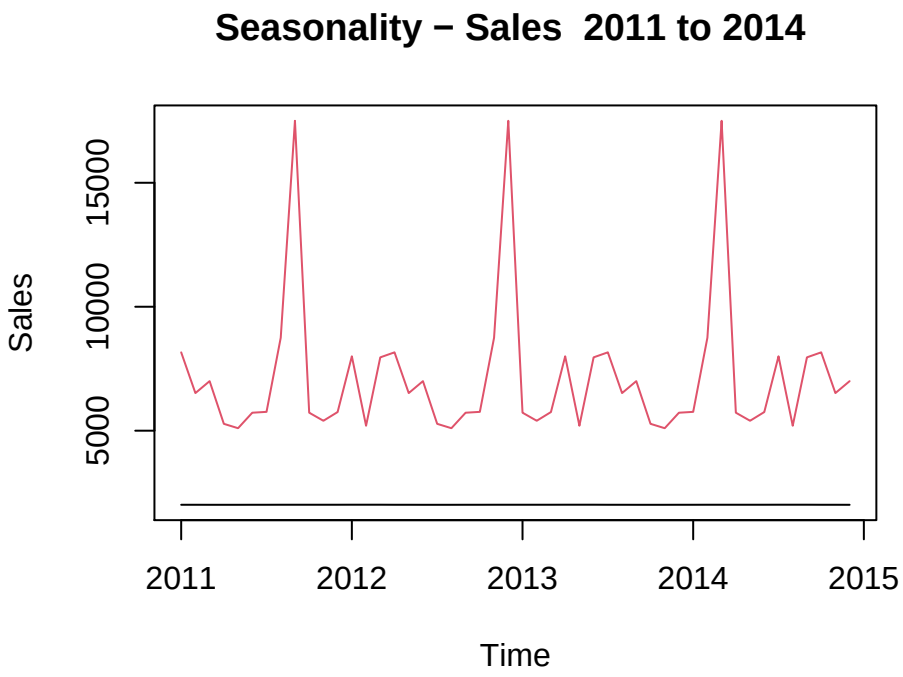
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2011 2011.000 2011.083 2011.167 2011.250 2011.333 2011.417 2011.500 2011.583
## 2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583
## 2013 2013.000 2013.083 2013.167 2013.250 2013.333 2013.417 2013.500 2013.583
## 2014 2014.000 2014.083 2014.167 2014.250 2014.333 2014.417 2014.500 2014.583
##           Sep      Oct      Nov      Dec
## 2011 2011.667 2011.750 2011.833 2011.917
## 2012 2012.667 2012.750 2012.833 2012.917
## 2013 2013.667 2013.750 2013.833 2013.917
## 2014 2014.667 2014.750 2014.833 2014.917
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 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
```

```
##          Order.Date     Sales
## Jan 2011   2011.667  8159.952
## Feb 2011   2011.667  6517.080
## Mar 2011   2011.083  6998.640
## Apr 2011   2011.583  5276.988
## May 2011   2011.583  5100.000
## Jun 2011   2011.833  5725.350
## Jul 2011   2012.583  5759.964
## Aug 2011   2013.083  8749.950
## Sep 2011   2013.750 17499.950
## Oct 2011   2013.417  5726.160
## Nov 2011   2013.750  5399.910
## Dec 2011   2013.417  5751.540
## Jan 2012   2014.833  7999.980
## Feb 2012   2014.750  5199.960
## Mar 2012   2014.667  7958.580
## Apr 2012   2011.667  8159.952
## May 2012   2011.667  6517.080
## Jun 2012   2011.083  6998.640
## Jul 2012   2011.583  5276.988
## Aug 2012   2011.583  5100.000
## Sep 2012   2011.833  5725.350
## Oct 2012   2012.583  5759.964
## Nov 2012   2013.083  8749.950
## Dec 2012   2013.750 17499.950
## Jan 2013   2013.417  5726.160
## Feb 2013   2013.750  5399.910
## Mar 2013   2013.417  5751.540
## Apr 2013   2014.833  7999.980
## May 2013   2014.750  5199.960
## Jun 2013   2014.667  7958.580
## Jul 2013   2011.667  8159.952
## Aug 2013   2011.667  6517.080
## Sep 2013   2011.083  6998.640
## Oct 2013   2011.583  5276.988
## Nov 2013   2011.583  5100.000
## Dec 2013   2011.833  5725.350
## Jan 2014   2012.583  5759.964
## Feb 2014   2013.083  8749.950
## Mar 2014   2013.750 17499.950
## Apr 2014   2013.417  5726.160
## May 2014   2013.750  5399.910
## Jun 2014   2013.417  5751.540
## Jul 2014   2014.833  7999.980
## Aug 2014   2014.750  5199.960
## Sep 2014   2014.667  7958.580
## Oct 2014   2011.667  8159.952
## Nov 2014   2011.667  6517.080
## Dec 2014   2011.083  6998.640
```

Plotting the time series for 2011-2014 to know the Seasonality

```
# Use ts.plot
ts.plot(tseries, col = 1:4, xlab = "Time", ylab = "Sales", main = "Seasonality - Sales  2011 to 2014 ")
```

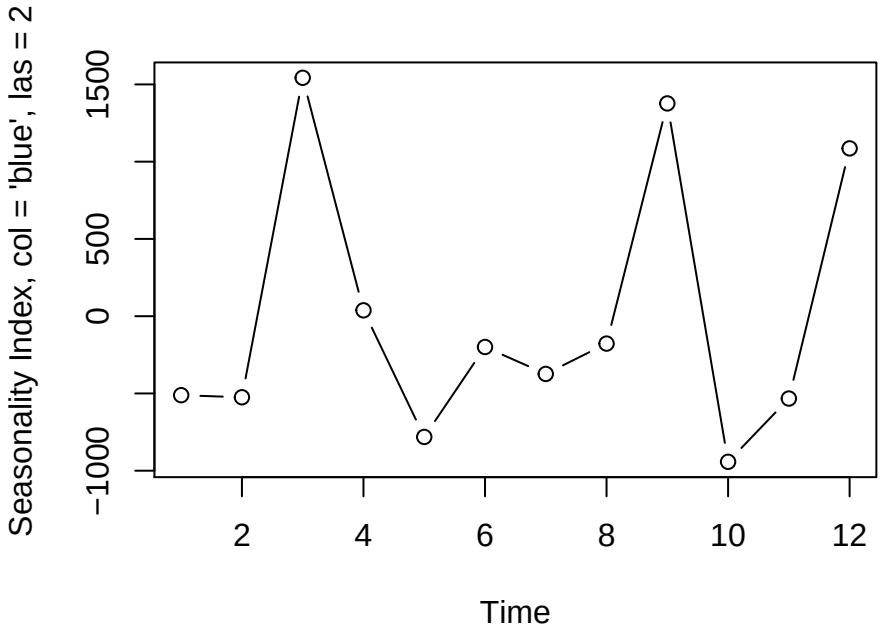


Decompose the series

```
decomp <- decompose(tseries)
decomp$figure
```

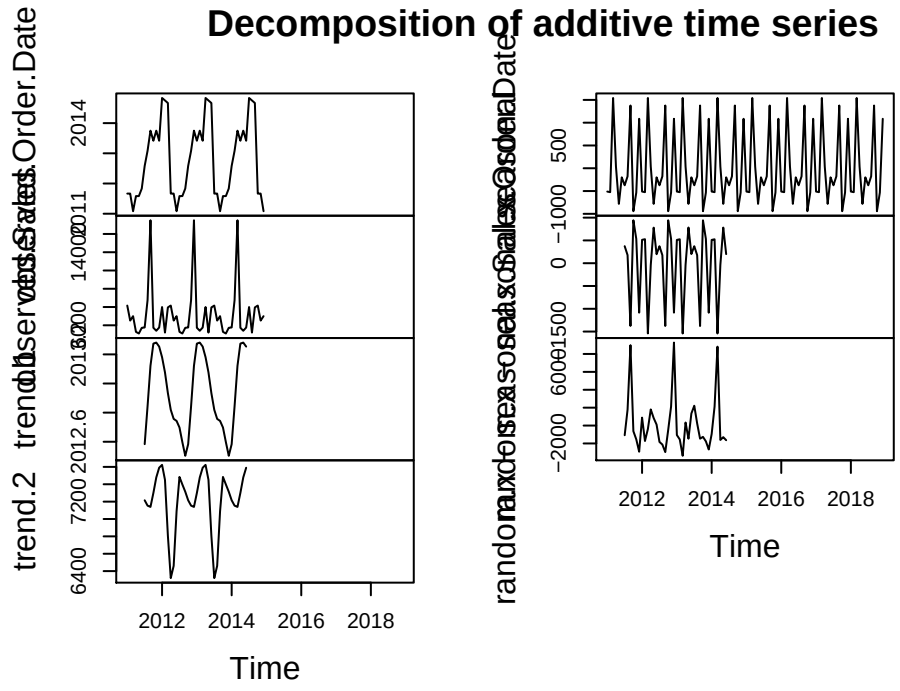
```
##  [1] -511.29898 -524.81814 1542.82615   37.87434 -781.57383 -199.49337
##  [7] -373.65570 -177.40703 1377.00670 -942.38812 -533.02097 1085.94896
```

```
plot(decomp$figure, type = 'b', xlab = 'Time', ylab = "Seasonality Index, col = 'blue', las = 2")
```



Random, seasonal, trend, observed plots

```
plot(decomp)
```



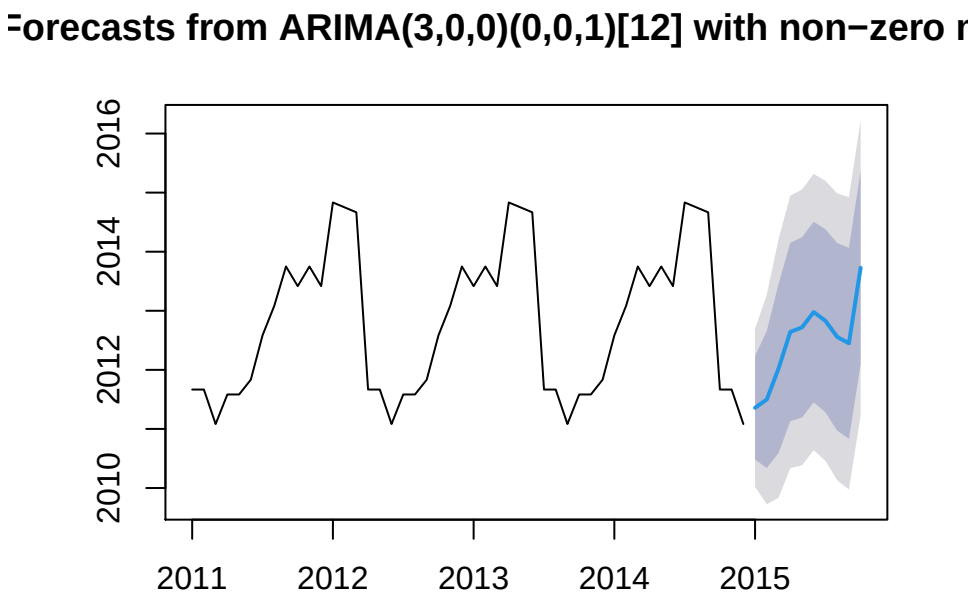
Auto regressive integrated moving average arima()

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
arima_fit = auto.arima(tseries[,1])
arima_forecast = forecast(arima_fit, h = 10)

plot(arima_forecast)
```




```
#acf(tseries, pl=FALSE)
#acf(tseries, lag.max= 12, main=("Corelleogram for 2011 - 2014"))
```

Now we explore time series data for each year separately

Get the rows for sales in 2011

```
summary2011 <- select_raw_data2 %>%
  select(Order.Date, Sales) %>%
  filter(Order.Date >= "2011-01-01") %>%
  filter(Order.Date <= "2011-12-31")


View(summary2011)
```

Get only the year and month from the Order Date

```
library(zoo)
summary2011$Order.Date <- as.yearmon(summary2011$Order.Date)
```

```r
head(summary2011)
```

```
##   Order.Date    Sales
## 1   Sep 2011 8159.952
## 2   Sep 2011 6517.080
## 3   Feb 2011 6998.640
## 4   Aug 2011 5276.988
## 5   Aug 2011 5100.000
## 6   Nov 2011 5725.350
```

```r
# Check whether data_vector and time_series are ts objects
is.ts(summary2011)
```

```
## [1] FALSE
```

Convert to time series object Time series object for 2011 sales

```r
tseries2011 <- ts(summary2011, start = c(2011, 1), end = c(2011,12), frequency = 12)

print(tseries2011)
```

```
##          Order.Date    Sales
## Jan 2011   2011.667 8159.952
## Feb 2011   2011.667 6517.080
## Mar 2011   2011.083 6998.640
## Apr 2011   2011.583 5276.988
## May 2011   2011.583 5100.000
## Jun 2011   2011.833 5725.350
## Jul 2011   2011.667 8159.952
## Aug 2011   2011.667 6517.080
## Sep 2011   2011.083 6998.640
## Oct 2011   2011.583 5276.988
## Nov 2011   2011.583 5100.000
## Dec 2011   2011.833 5725.350
```

```r
is.ts(tseries2011)
```

```
## [1] TRUE
```

Check the start, end , frequency of the time series

```r
start(tseries2011)
```

```
## [1] 2011    1
```

```r
end(tseries2011)
```

```
## [1] 2011   12
```

```r
deltat(tseries2011)
```

```
## [1] 0.08333333
```

```r
frequency(tseries2011)
```

```
## [1] 12
```

```r
time(tseries2011)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2011 2011.000 2011.083 2011.167 2011.250 2011.333 2011.417 2011.500 2011.583
```

```
##           Sep      Oct      Nov      Dec
## 2011 2011.667 2011.750 2011.833 2011.917
```

```
cycle(tseries2011)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2011   1   2   3   4   5   6   7   8   9  10  11  12
```

```
print(tseries2011)
```

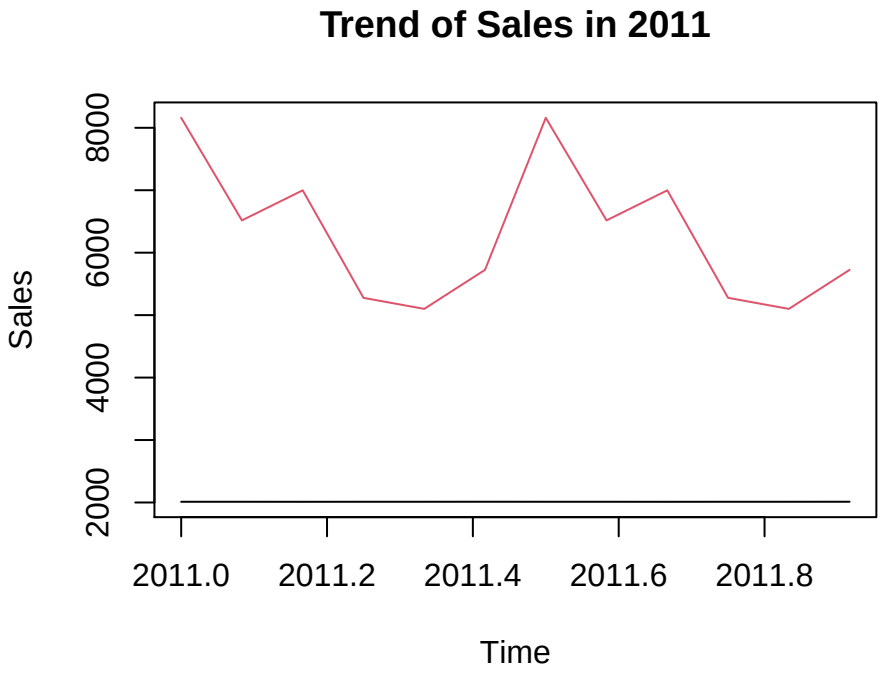
```
##          Order.Date    Sales
## Jan 2011   2011.667 8159.952
## Feb 2011   2011.667 6517.080
## Mar 2011   2011.083 6998.640
## Apr 2011   2011.583 5276.988
## May 2011   2011.583 5100.000
## Jun 2011   2011.833 5725.350
## Jul 2011   2011.667 8159.952
## Aug 2011   2011.667 6517.080
## Sep 2011   2011.083 6998.640
## Oct 2011   2011.583 5276.988
## Nov 2011   2011.583 5100.000
## Dec 2011   2011.833 5725.350
```

```
# Check whether is a ts object
is.ts(tseries2011)
```

```
## [1] TRUE
```

Plotting the time series for 2011 to show the Trend in 2011

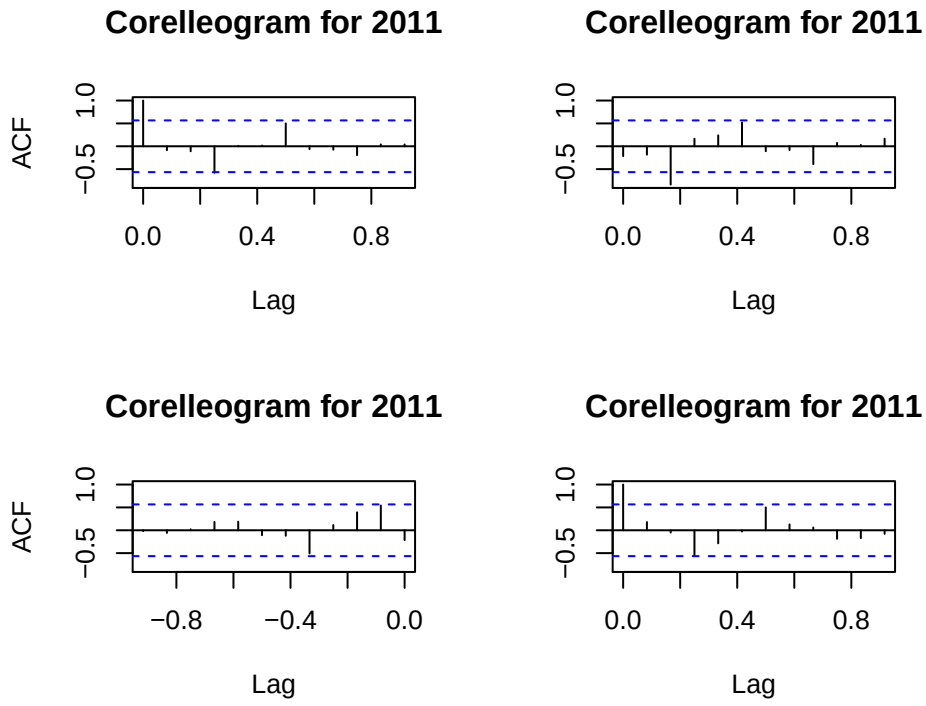
```
# Use ts.plot
ts.plot(tseries2011, col = 1:4, xlab = "Time", ylab = "Sales", main = "Trend of Sales in 2011")
```




```
acf(tseries2011, pl=FALSE)
```

```
##
## Autocorrelations of series 'tseries2011', by lag
##
## , , Order.Date
##
##  Order.Date       Sales
##   1.000 ( 0.0000) -0.215 ( 0.0000)
##  -0.086 ( 0.0833)  0.542 (-0.0833)
##  -0.109 ( 0.1667)  0.393 (-0.1667)
##  -0.579 ( 0.2500)  0.114 (-0.2500)
##   0.008 ( 0.3333) -0.506 (-0.3333)
##   0.016 ( 0.4167) -0.120 (-0.4167)
##   0.500 ( 0.5000) -0.108 (-0.5000)
##  -0.063 ( 0.5833)  0.188 (-0.5833)
##
## , , Sales
##
##  Order.Date       Sales
##  -0.215 ( 0.0000)  1.000 ( 0.0000)
##  -0.184 ( 0.0833)  0.179 ( 0.0833)
##  -0.839 ( 0.1667) -0.052 ( 0.1667)
##   0.165 ( 0.2500) -0.565 ( 0.2500)
##   0.239 ( 0.3333) -0.285 ( 0.3333)
##   0.519 ( 0.4167) -0.028 ( 0.4167)
##  -0.108 ( 0.5000)  0.500 ( 0.5000)
##  -0.083 ( 0.5833)  0.129 ( 0.5833)
```

```
acf(tseries2011, lag.max= 12, main=("Corelleogram for 2011"))
```



Autocorrelation measures the relationship between a variable's current values and its historical values.

Trend refers to the increasing or decreasing values in a given time series.

```
summary2012 <- raw_data2 %>%
  select(Order.Date, Sales) %>%
  filter(Order.Date >= "2011-12-31") %>%
  filter(Order.Date <= "2012-12-31")

View(summary2012)
```

```
tseries2012 <- ts(summary2012, start = c(2012, 1), end = c(2012,12), frequency = 12)

print(tseries2012)
```

```
##          Order.Date    Sales
## Jan 2012      15371 593.9895
## Feb 2012      15371 151.9200
## Mar 2012      15371 200.1600
## Apr 2012      15371 192.8800
## May 2012      15371  94.0200
## Jun 2012      15371  57.0000
## Jul 2012      15371  95.5170
## Aug 2012      15371  69.5544
## Sep 2012      15371 139.5900
## Oct 2012      15371  33.1200
## Nov 2012      15371   8.9640
## Dec 2012      15371  18.5400
```

Check the start, end of the time series

```
start(tseries2012)
```

```
## [1] 2012    1
```

```
end(tseries2012)
```

```
## [1] 2012   12
```

```
deltat(tseries2012)
```

```
## [1] 0.08333333
```

```
frequency(tseries2012)
```

```
## [1] 12
```

```
time(tseries2012)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583
##           Sep      Oct      Nov      Dec
## 2012 2012.667 2012.750 2012.833 2012.917
```

```
cycle(tseries2012)
```

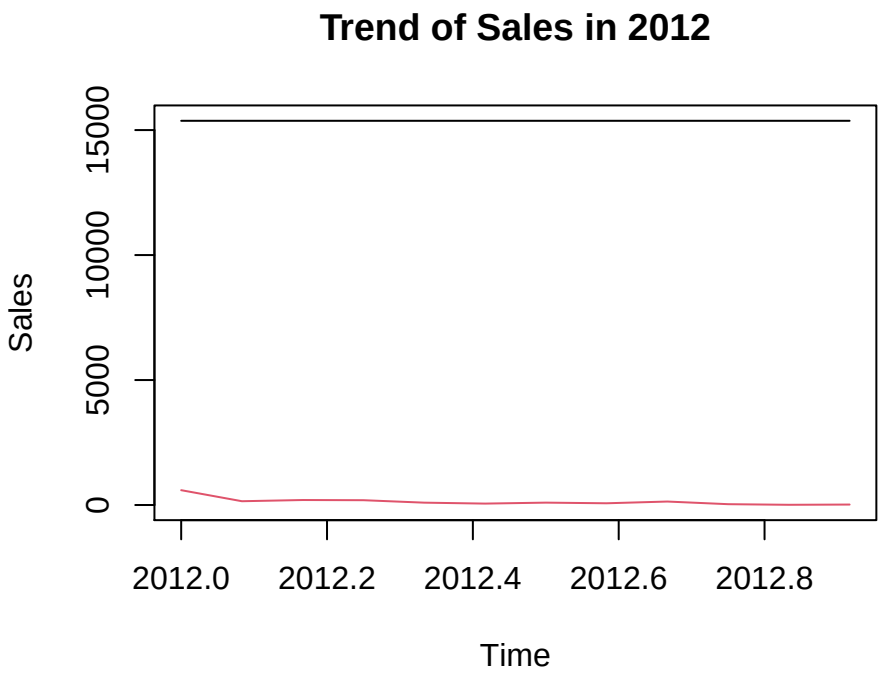
```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2012   1   2   3   4   5   6   7   8   9  10  11  12
```

```
print(tseries2012)
```

```
##          Order.Date    Sales
## Jan 2012      15371 593.9895
## Feb 2012      15371 151.9200
## Mar 2012      15371 200.1600
## Apr 2012      15371 192.8800
## May 2012      15371  94.0200
## Jun 2012      15371  57.0000
## Jul 2012      15371  95.5170
## Aug 2012      15371  69.5544
## Sep 2012      15371 139.5900
## Oct 2012      15371  33.1200
## Nov 2012      15371   8.9640
## Dec 2012      15371  18.5400
```

Plotting the time series for 2011 to show the Trend in 2011

```
# Use ts.plot
ts.plot(tseries2012, col = 1:4, xlab = "Time", ylab = "Sales", main = "Trend of Sales in 2012")
```



2013

Get the rows for sales in 2013

```
summary2013 <- raw_data2 %>%
  select(Order.Date, Sales) %>%
  filter(Order.Date >= "2012-12-31") %>%
  filter(Order.Date <= "2013-12-31")

View(summary2013)
```

Get only the year and month form the Order Date

```
library(zoo)
summary2013$Order.Date <- as.yearmon(summary2013$Order.Date)

head(summary2013)
```

```
##   Order.Date    Sales
## 1   Jan 2013 1649.214
## 2   Jan 2013 1358.280
## 3   Jan 2013  728.568
## 4   Jan 2013 2189.520
## 5   Jan 2013 1362.060
## 6   Jan 2013  299.052
```

```
# Check whether data_vector and time_series are ts objects
is.ts(summary2013)
```

```
## [1] FALSE
```

Convert to time series object Time series object for 2013 sales

```
tseries2013 <- ts(summary2013, start = c(2013, 1), end = c(2013,12), frequency = 12)

print(tseries2013)
```

```
##          Order.Date    Sales
## Jan 2013       2013 1649.214
## Feb 2013       2013 1358.280
## Mar 2013       2013  728.568
## Apr 2013       2013 2189.520
## May 2013       2013 1362.060
## Jun 2013       2013  299.052
## Jul 2013       2013  246.120
## Aug 2013       2013  155.034
## Sep 2013       2013  242.250
## Oct 2013       2013  416.664
## Nov 2013       2013   85.428
## Dec 2013       2013   94.980
```

```
is.ts(tseries2013)
```

```
## [1] TRUE
```

Check the start, end of the time series

```
start(tseries2013)
```

```
## [1] 2013    1
```

```
end(tseries2013)
```

```
## [1] 2013   12
```

```
deltat(tseries2013)
```

```
## [1] 0.08333333
```

```
frequency(tseries2013)
```

```
## [1] 12
```

```
time(tseries2013)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2013 2013.000 2013.083 2013.167 2013.250 2013.333 2013.417 2013.500 2013.583
##           Sep      Oct      Nov      Dec
## 2013 2013.667 2013.750 2013.833 2013.917
```

```
cycle(tseries2013)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2013   1   2   3   4   5   6   7   8   9  10  11  12
```

```
print(tseries2013)
```

```
##          Order.Date    Sales
## Jan 2013       2013 1649.214
## Feb 2013       2013 1358.280
## Mar 2013       2013  728.568
## Apr 2013       2013 2189.520
## May 2013       2013 1362.060
## Jun 2013       2013  299.052
```

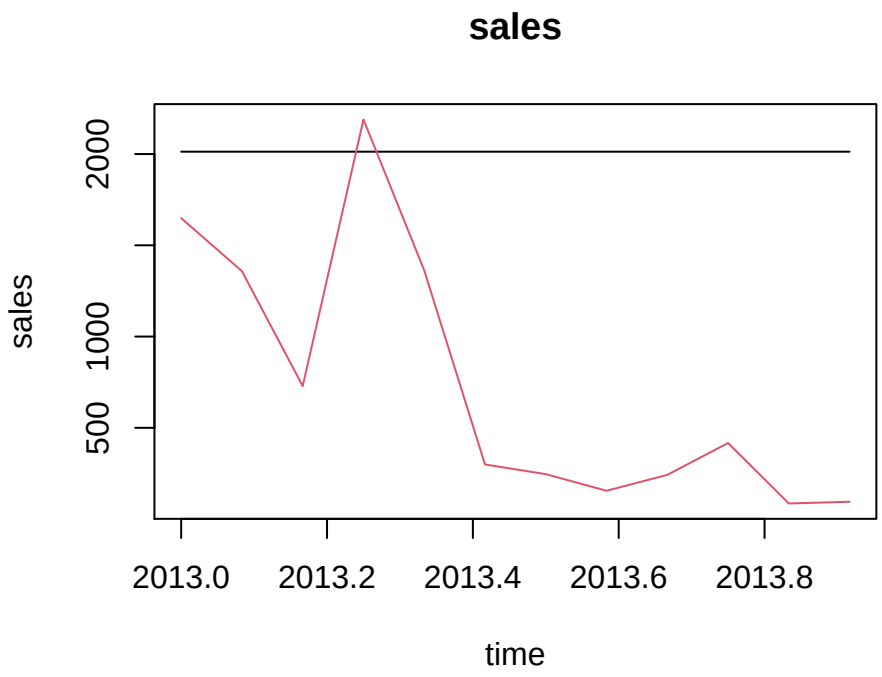
```
## Jul 2013       2013  246.120
## Aug 2013       2013  155.034
## Sep 2013       2013  242.250
## Oct 2013       2013  416.664
## Nov 2013       2013   85.428
## Dec 2013       2013   94.980
```

Check if the time series object was generated

```
# Check whether is a ts object
is.ts(tseries2013)
```

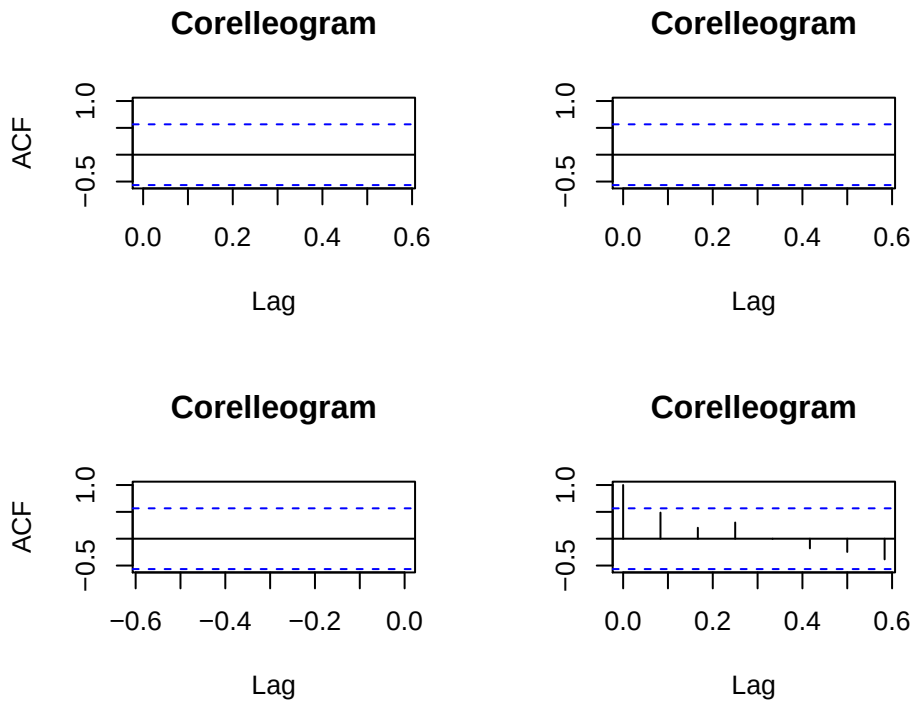
```
## [1] TRUE
```

```
# Use ts.plot
ts.plot(tseries2013, col = 1:4, xlab = "time", ylab = "sales", main = "sales")
```



When the autocorrelation in a time series is high, it becomes easy to predict future values by simply referring to past values.

```
acf(tseries2013, main=("Corelleogram"))
```



2014

Get the rows for sales in 2014

```
summary2014 <- select_raw_data2 %>%
  select(Order.Date, Sales) %>%
  filter(Order.Date >= "2014-01-01") %>%
  filter(Order.Date <= "2014-12-31")


View(summary2014)
```

Get only the year and month form the Order Date

```
library(zoo)
summary2014$Order.Date <- as.yearmon(summary2014$Order.Date)

head(summary2014)
```

```
##   Order.Date   Sales
## 1   Nov 2014 7999.98
## 2   Oct 2014 5199.96
## 3   Sep 2014 7958.58
```

```
# Check whether data_vector and time_series are ts objects
is.ts(summary2014)
```

```
## [1] FALSE
```

Convert to time series object Time series object for 2014 sales

```
tseries2014 <- ts(summary2014, start = c(2014, 1), end = c(2014,12), frequency = 12)
```

```
print(tseries2014)
```

```
##          Order.Date   Sales
## Jan 2014   2014.833 7999.98
## Feb 2014   2014.750 5199.96
## Mar 2014   2014.667 7958.58
## Apr 2014   2014.833 7999.98
## May 2014   2014.750 5199.96
## Jun 2014   2014.667 7958.58
## Jul 2014   2014.833 7999.98
## Aug 2014   2014.750 5199.96
## Sep 2014   2014.667 7958.58
## Oct 2014   2014.833 7999.98
## Nov 2014   2014.750 5199.96
## Dec 2014   2014.667 7958.58
```

```
is.ts(tseries2014)
```

```
## [1] TRUE
```

Check the start, end of the time series

```
start(tseries2014)
```

```
## [1] 2014    1
```

```
end(tseries2014)
```

```
## [1] 2014   12
```

```
deltat(tseries2014)
```

```
## [1] 0.08333333
```

```
frequency(tseries2014)
```

```
## [1] 12
```

```
time(tseries2014)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2014 2014.000 2014.083 2014.167 2014.250 2014.333 2014.417 2014.500 2014.583
##           Sep      Oct      Nov      Dec
## 2014 2014.667 2014.750 2014.833 2014.917
```

```
cycle(tseries2014)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2014   1   2   3   4   5   6   7   8   9  10  11  12
```

```
print(tseries2014)
```

```
##          Order.Date   Sales
## Jan 2014   2014.833 7999.98
## Feb 2014   2014.750 5199.96
## Mar 2014   2014.667 7958.58
## Apr 2014   2014.833 7999.98
## May 2014   2014.750 5199.96
## Jun 2014   2014.667 7958.58
## Jul 2014   2014.833 7999.98
## Aug 2014   2014.750 5199.96
```

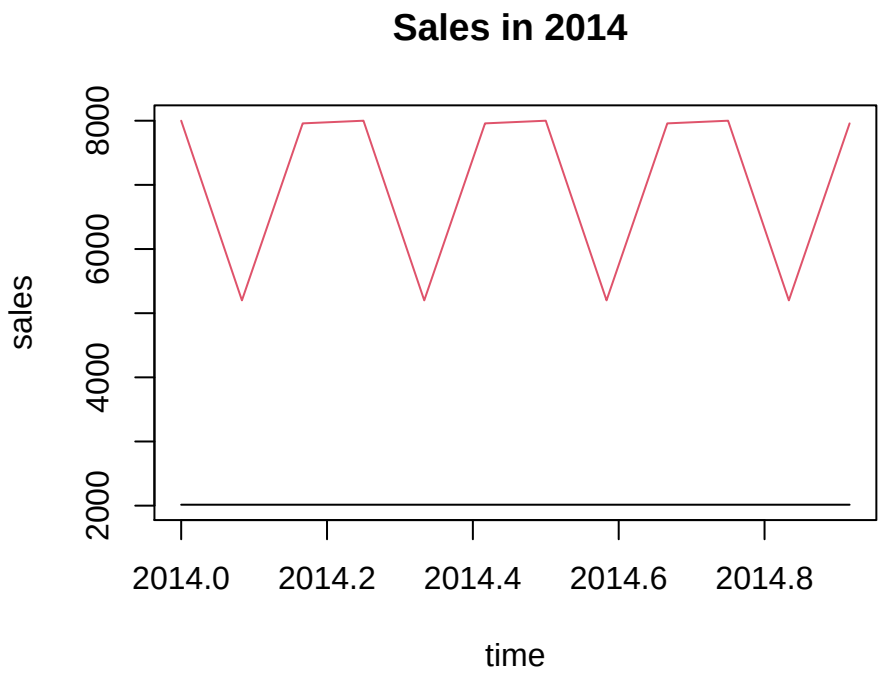
```
## Sep 2014   2014.667 7958.58
## Oct 2014   2014.833 7999.98
## Nov 2014   2014.750 5199.96
## Dec 2014   2014.667 7958.58
```

Check if the time series object was generated

```
# Check whether is a ts object
is.ts(tseries2014)
```

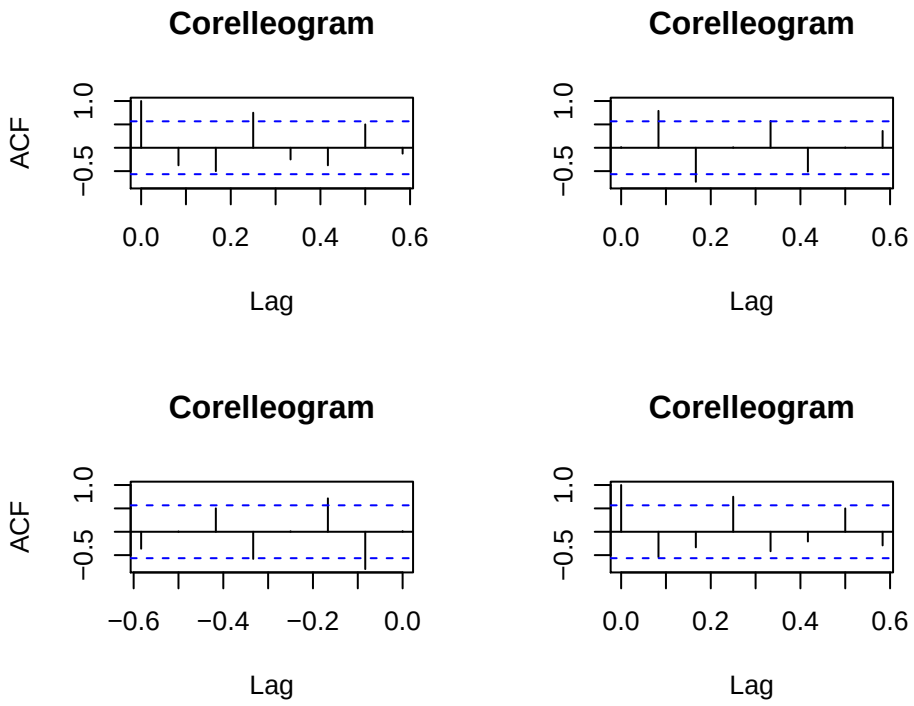
```
## [1] TRUE
```

```
# Use ts.plot with eu_stocks
ts.plot(tseries2014, col = 1:4, xlab = "time", ylab = "sales", main = "Sales in 2014 ")
```



When the autocorrelation in a time series is high, it becomes easy to predict future values by simply referring to past values.

```
acf(tseries2014, main=("Corelleogram"))
```



We have plotted the timeseries to look for Seasonality in the 4 year and trends in each on the 4 years, and forecasted sales for the next year.

Also, we have plotted the autocorrelation function to try and dig more into the data.