Needed to start R in Colabs:

```
cat(system('python3 -c "from google.colab import drive\ndrive.mount()"', intern=TRUE),
```

```
Warning message in system("python3 -c \"from google.colab import drive\ndrive.mou
“running command 'python3 -c "from google.colab import drive
drive.mount()"' had status 1”

TRUE
```

```
?system
```

```
install.packages("rgl", repos = "http://cran.rstudio.com/")
install.packages("ConsRank", repos = "http://cran.rstudio.com/")
library("ConsRank")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘httpuv’, ‘xtable’, ‘sourcetools’, ‘fastmap’, ‘m


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘XML’, ‘data.table’, ‘rlist’, ‘proxy’, ‘gtools’


Loading required package: rgl

Warning message in rgl.init(initValue, onlyNULL):
“RGL: unable to open X11 display”
Warning message:
“'rgl.init' failed, running with 'rgl.useNULL = TRUE'.”

Attaching package: ‘ConsRank’


The following object is masked from ‘package:base’:

    labels
```

```
system("add-apt-repository -y ppa:marutter/rrutter")
system("add-apt-repository -y ppa:marutter/c2d4u")
system("apt-get update")
system("apt install -y r-cran-rstan")
```

Installing forecasting package:

```
install.packages('forecast', dependencies = TRUE)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘xts’, ‘TTR’, ‘quadprog’, ‘quantmod’, ‘fracdiff’
```

```
install.packages('ggmap')
install.packages('codetools')
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘sp’, ‘RgoogleMaps’, ‘png’, ‘rjson’, ‘jpeg’, ‘bi


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library("ggmap")
```

```
Loading required package: ggplot2

Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

Please cite ggmap if you use it! See citation("ggmap") for details.
```

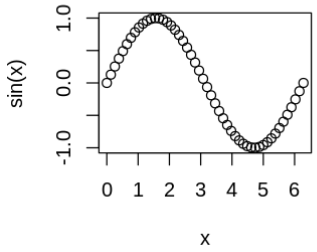
```
options(repr.plot.width  = 3,
        repr.plot.height = 3)
Sys.setenv(USE_CXX14 = 1)
library("rstan") # observe startup messages
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

```
Loading required package: StanHeaders

rstan (Version 2.19.3, GitRev: 2e1f913d3ca3)

For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
```

Testing that R works:

```
x <- seq(0, 2*pi, length.out=50)
```

```
plot(x, sin(x))
```



Exercise 2.3

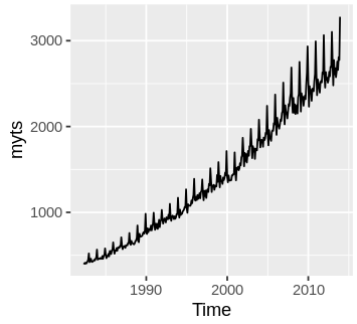
```
retaildata <- readxl::read_excel("/content/retail.xlsx", skip=1)
```

```
library(forecast)
```

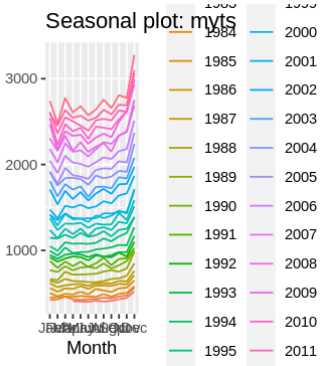
```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
myts <- ts(retaildata[,"A3349398A"],
  frequency=12, start=c(1982,4))
```

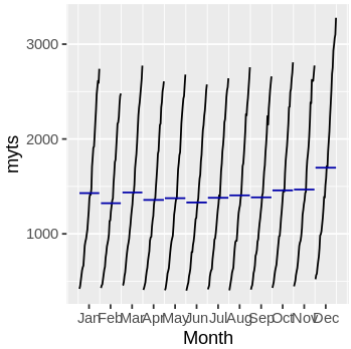
```
autoplot(myts)
```



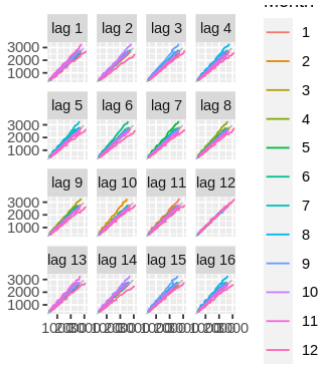

```
ggseasonplot(myts)
```



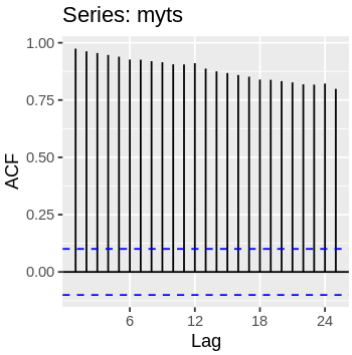

```
ggsubseriesplot(myts)
```




```
gglagplot(myts)
```



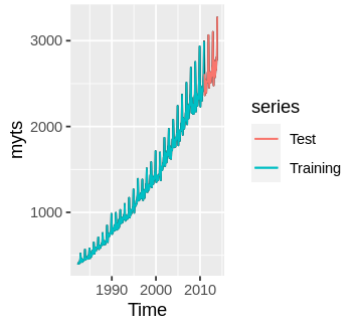

```
ggAcf(myts)
```



Exercise.3.8

```
myts.train <- window(myts, end=c(2010,12))
myts.test <- window(myts, start=2011)
```

```
autoplot(myts) +
  autolayer(myts.train, series="Training") +
  autolayer(myts.test, series="Test")
```




```
fc <- snaive(myts.train)
```

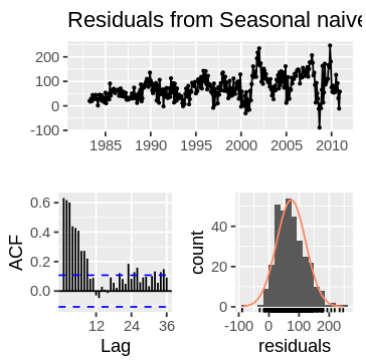
```
accuracy(fc,myts.test)
```

A matrix: 2 × 8 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	73.94114	88.31208	75.13514	6.068915	6.134838	1.000000	0.6312891	NA
Test set	115.00000	127.92727	115.00000	4.459712	4.459712	1.530576	0.2653013	0.7267171

```
checkresiduals(fc)
```

```
	Ljung-Box test

data:  Residuals from Seasonal naive method
Q* = 671.41, df = 24, p-value < 2.2e-16

Model df: 0.   Total lags used: 24
```



Do the residuals appear to be uncorrelated and normally distributed?

Answer: If i am reading the chart above correctly they appear correlated and fairly normally distributed

How sensitive are the accuracy measures to the training/test split?

Answer: With the wide difference in values it appears that they split is very sensative.

Exercise 2.2

```
tute1 <- read.csv("/content/tute1.csv", header=TRUE)
tute1
```

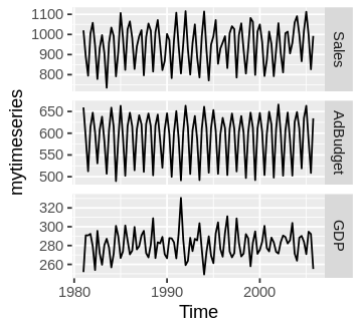
A data.frame: 100 × 4

X	Sales	AdBudget	GDP
<fct>	<dbl>	<dbl>	<dbl>
Mar-81	1020.2	659.2	251.8
Jun-81	889.2	589.0	290.9
Sep-81	795.0	512.5	290.8
Dec-81	1003.9	614.1	292.4
Mar-82	1057.7	647.2	279.1
Jun-82	944.4	602.0	254.0
Sep-82	778.5	530.7	295.6
Dec-82	932.5	608.4	271.7
Mar-83	996.5	637.9	259.6
Jun-83	907.7	582.4	280.5
Sep-83	735.1	506.8	287.2
Dec-83	958.1	606.7	278.0
Mar-84	1034.1	658.7	256.8
Jun-84	992.8	614.9	271.0
Sep-84	791.7	489.9	300.9
Dec-84	914.2	586.5	289.8
Mar-85	1106.5	663.0	266.8
Jun-85	985.1	591.7	273.7
Sep-85	823.9	502.2	301.3
Dec-85	1025.1	616.4	285.6
Mar-86	1064.7	647.1	270.6
Jun-86	981.9	615.5	274.6
Sep-86	828.3	514.8	299.7
Dec-86	940.7	609.1	275.9
Mar-87	991.1	641.3	279.3
Jun-87	1021.2	620.2	290.8
Sep-87	796.7	511.2	295.6
Dec-87	986.6	621.3	271.9
Mar-88	1054.2	645.3	267.4
Jun-88	1018.7	616.0	281.0
⋮	⋮	⋮	⋮
Sep-98	805.6	497.0	292.1
Dec-98	934.1	602.8	287.6
Mar-99	1081.7	647.3	258.0
Jun-99	1062.3	612.5	282.9
Sep-99	798.8	492.2	295.0
Dec-99	1014.3	610.8	271.2
Mar-00	1049.5	646.5	275.4
Jun-00	961.7	603.3	284.0
Sep-00	793.4	503.8	300.9
Dec-00	872.3	598.3	277.4
Mar-01	1014.2	649.4	273.8
Jun-01	952.6	620.2	288.4
Sep-01	792.4	497.9	283.4

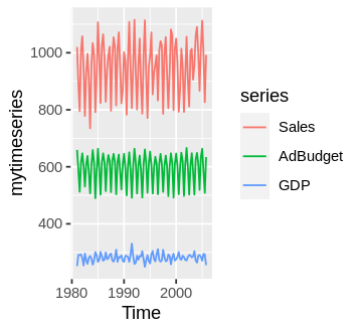
Sep-01	792.4	497.9	283.4
Dec-01	922.3	609.2	273.4
Mar-02	1055.9	665.9	271.5
Jun-02	906.2	600.4	283.6
Sep-02	811.2	502.3	290.6
Dec-02	1005.8	605.6	289.1
Mar-03	1013.8	647.6	282.2
Jun-03	905.6	583.5	285.6
Sep-03	957.3	502.5	304.0
Dec-03	1059.5	625.9	271.5
Mar-04	1090.6	648.7	263.9
Jun-04	998.9	610.7	288.3
Sep-04	866.6	519.1	290.2
Dec-04	1018.7	634.9	284.0
Mar-05	1112.5	663.1	270.9
Jun-05	997.4	583.3	294.7
Sep-05	826.8	508.6	292.2
Dec-05	992.6	634.2	255.1

```
mytimeseries <- ts(tute1[,-1], start=1981, frequency=4)
```

```
autoplot(mytimeseries, facets=TRUE)
```




```
autoplot(mytimeseries)
```



Exercise 3.5 (modified)

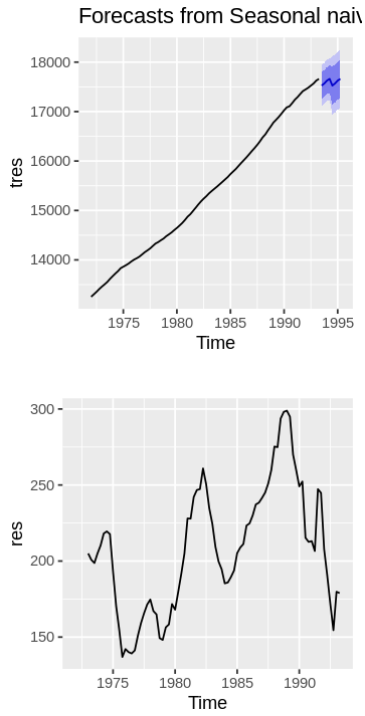
```
data(austres)
austres
```

A Time Series: 23 × 4

	Qtr1	Qtr2	Qtr3	Qtr4
1971		13067.3	13130.5	13198.4
1972	13254.2	13303.7	13353.9	13409.3
1973	13459.2	13504.5	13552.6	13614.3
1974	13669.5	13722.6	13772.1	13832.0
1975	13862.6	13893.0	13926.8	13968.9
1976	14004.7	14033.1	14066.0	14110.1
1977	14155.6	14192.2	14231.7	14281.5
1978	14330.3	14359.3	14396.6	14430.8
1979	14478.4	14515.7	14554.9	14602.5
1980	14646.4	14695.4	14746.6	14807.4
1981	14874.4	14923.3	14988.7	15054.1
1982	15121.7	15184.2	15239.3	15288.9
1983	15346.2	15393.5	15439.0	15483.5
1984	15531.5	15579.4	15628.5	15677.3
1985	15736.7	15788.3	15839.7	15900.6
1986	15961.5	16018.3	16076.9	16139.0
1987	16203.0	16263.3	16327.9	16398.9
1988	16478.3	16538.2	16621.6	16697.0
1989	16777.2	16833.1	16891.6	16956.8
1990	17026.3	17085.4	17106.9	17169.4
1991	17239.4	17292.0	17354.2	17414.2
1992	17447.3	17482.6	17526.0	17568.7
1993	17627.1	17661.5		

```
tres <- window(austres, start=1972)
fc <- snaive(tres)
autoplot(fc)
```

```
res <- residuals(fc)
autoplot(res)
```




```
checkresiduals(fc)
```

```
        Ljung-Box test

data:  Residuals from Seasonal naive method
Q* = 305.7, df = 8, p-value < 2.2e-16

Model df: 0.   Total lags used: 8
```

I had to use a dataset that was in the current R package. That said it looks like the residuals fare wide data. But agian i am comparing data in a way that might not have been inteded to be .