Using R

Forecasting: principles and practice 2

book by Rob Hyndman and George Athanasopoulos
slides by Peter Fuleky and modified by Joseph Alba

Installing R and RStudio

1. Download and install the latest version of R at http://cran.r-project.org

   Some packages may not work on the latest version of R. Older version 5.3 may be downloaded at:

   For windows (https://cran.r-project.org/bin/windows/base/old/)

   For Mac (https://cran.r-project.org/bin/macosx/old/index-old.html)

2. Download and install RStudio Desktop : http://www.rstudio.com/products/rstudio/#Desk

3. Run RStudio. On the 'Tools' menu, click on "Install packages" and install the package "fpp2"" (make sure "install dependencies" is checked).

That's it! You should now be ready to go.

Getting started with R

- Load the `fpp2` package using the Packages tab (lower right window of RStudio). 'forecast', 'fma', 'expsmooth', 'lmtest' and 'zoo' packages should also have a check mark in the box. This needs to be done at the start of every R session. It can also be done by typing `library(fpp2)` in the Console panel or in your script.

- Download the file http://robjhyndman.com/data/tute1.csv.

- Set as your working directory the directory where you downloaded tute1.csv. Use the 'session' menu, choose "set working directory", "choose directory"

- Using the **file** menu, choose "Import Dataset" and "From Text(base)" and import the data `tute1.csv` file.

Getting started with R (continued)

- The data is now saved as an object in your Global Environment workspace. Clicking the name of the object will cause it to be viewed. Typing the name of the object in the Console tab will cause it to be printed to the console.

Use R

See what the following commands do:

```
head(tute1, n=10)
tail(tute1, n=6)
tute1[,2]
tute1[,"Sales"]
tute1[5,]
tute1[1:10,3:4]
tute1[1:10,2] <- 0
tute1[1:20,]
```

Notice that <- means to assign the value on the right to the object on the left.

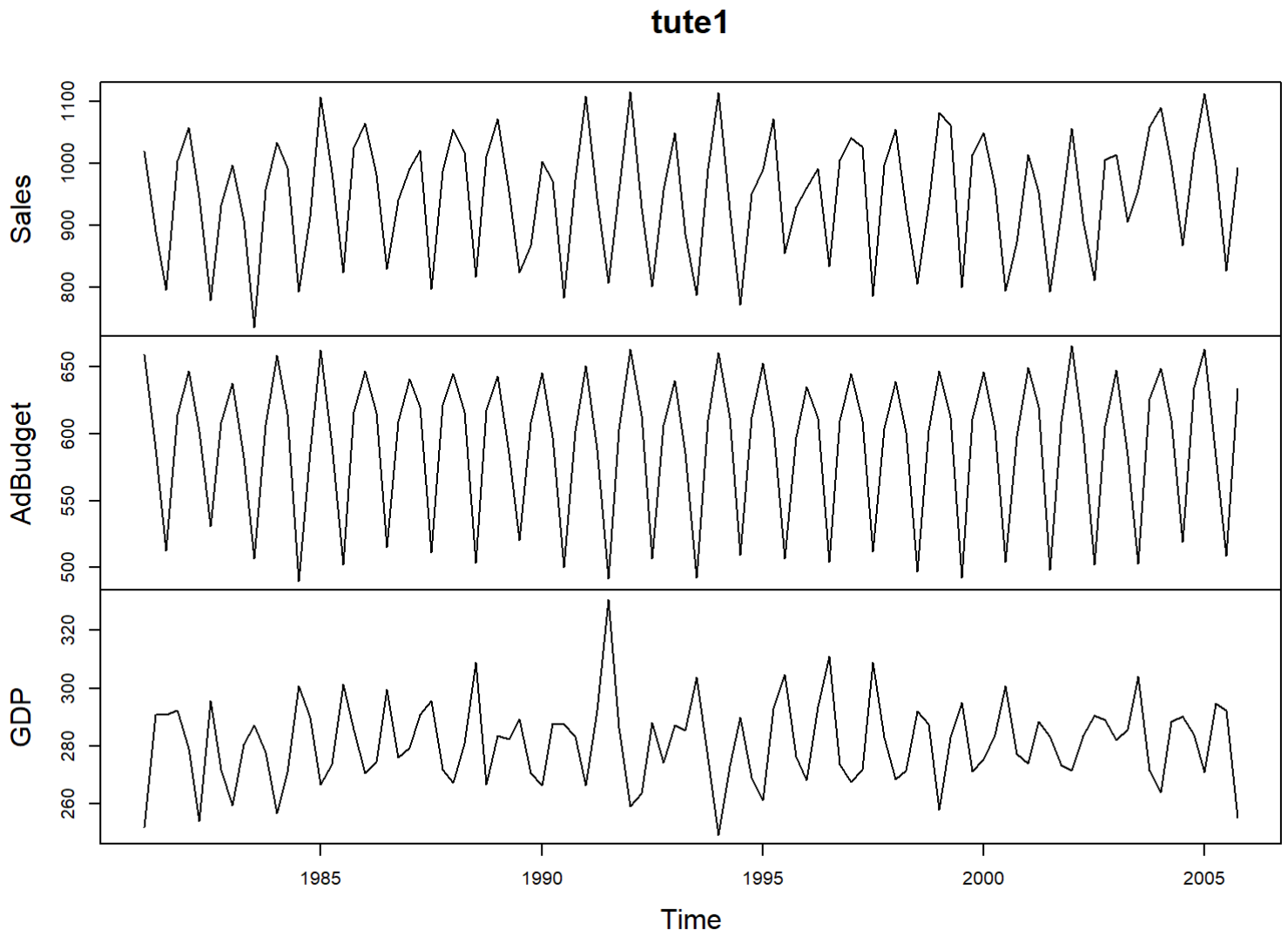
Convert the data to time series

```
tute1 <- read.csv("tute1.csv")
tute1 <- ts(tute1[,-1], start=1981, frequency=4)
```

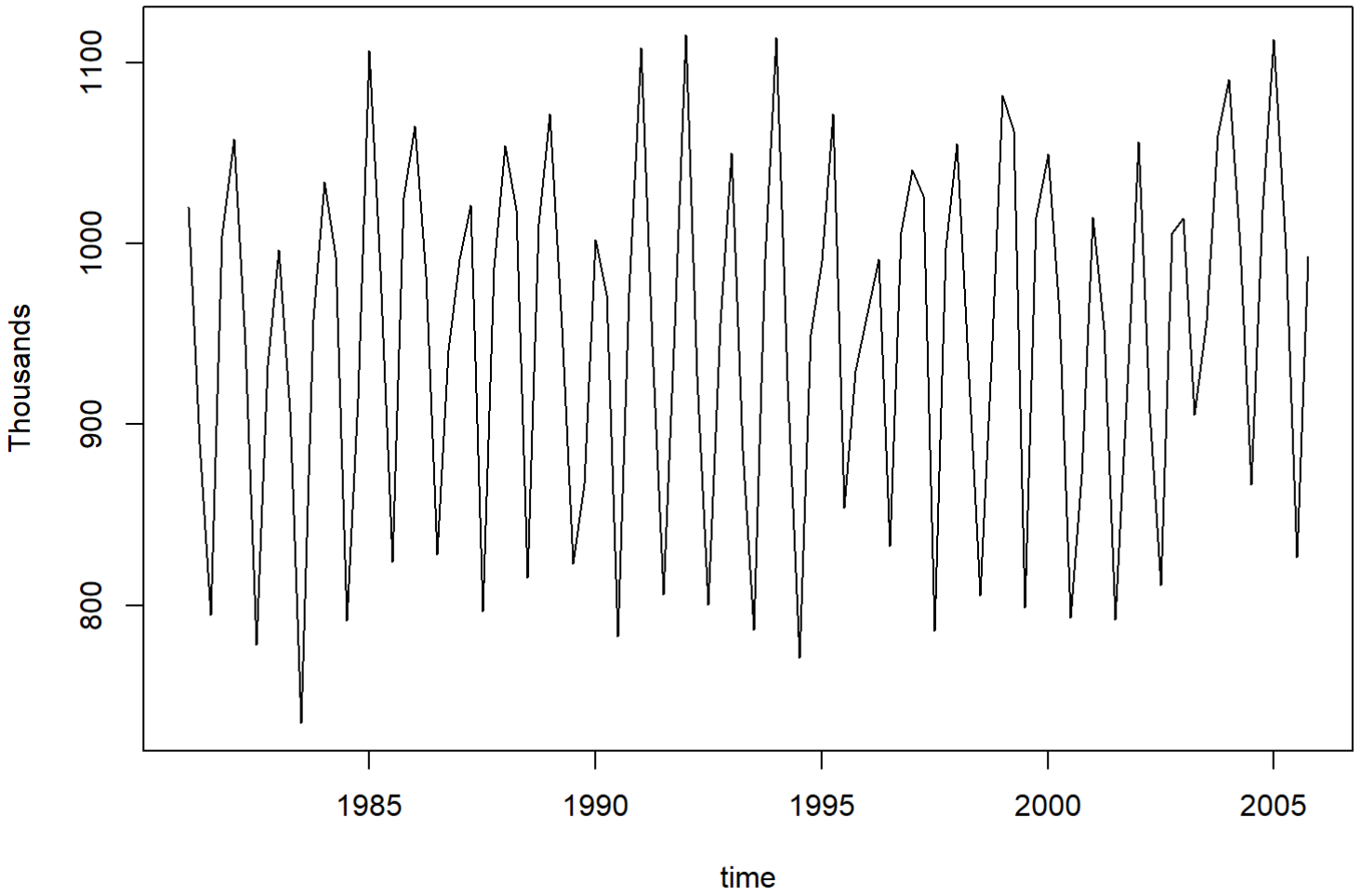
(The [,-1] removes the first column which contains the quarters as we don't need them now.)

Construct time series plots of each of the three series

```
plot(tute1)
```

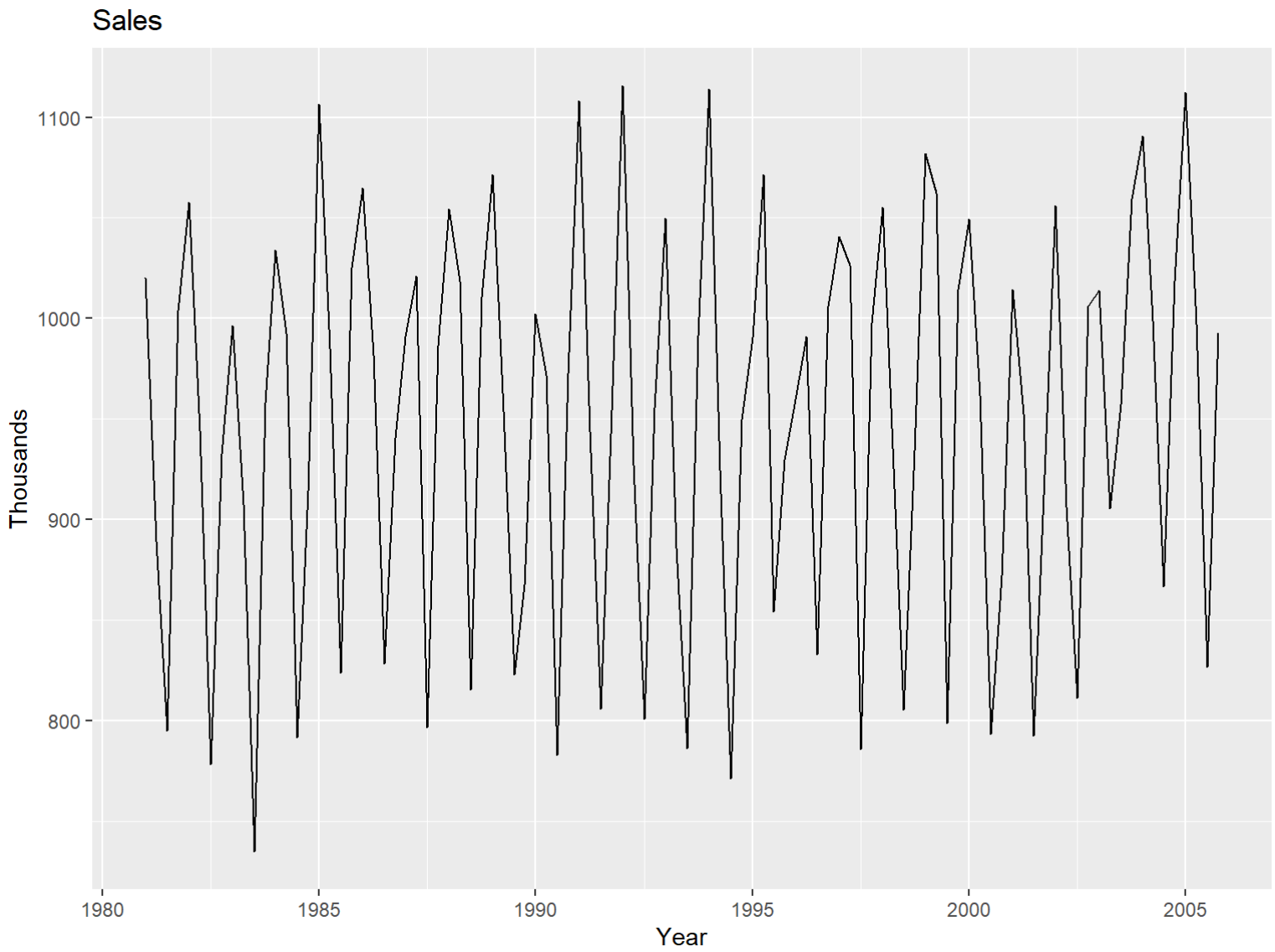

tute1
Sales
AdBudget
GDP
800
900
1000
1100
500
550
600
650
260
280
300
320
1985
1990
1995
2000
2005
Time

```
plot(tute1[,"Sales"], xlab="time",
     ylab="Thousands")
```



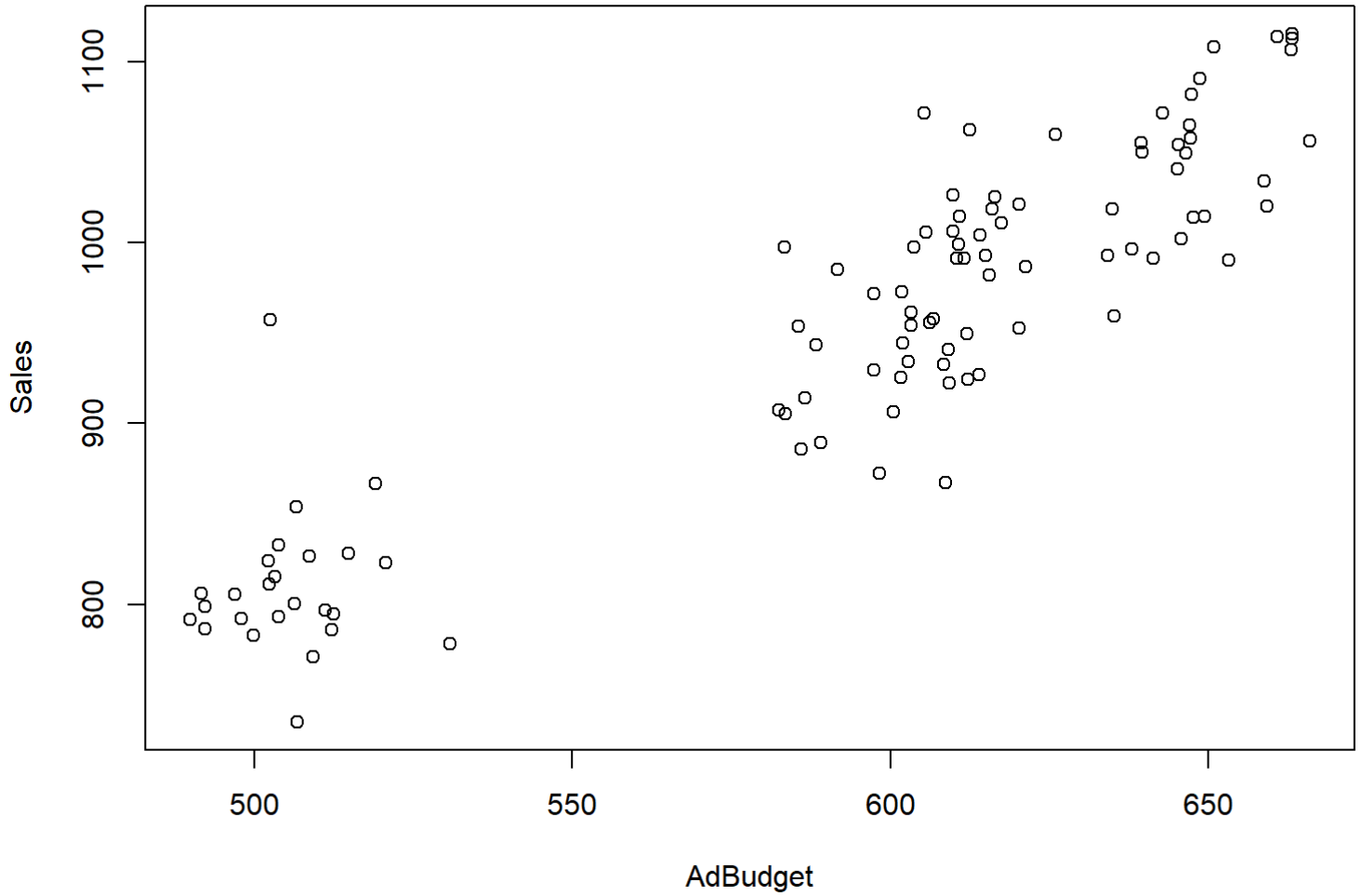
Sales

```
library(fpp2)
S1<- tute1[,"Sales"]
autoplot(S1) +
ggtitle("Sales") +
xlab("Year") + ylab("Thousands")
```

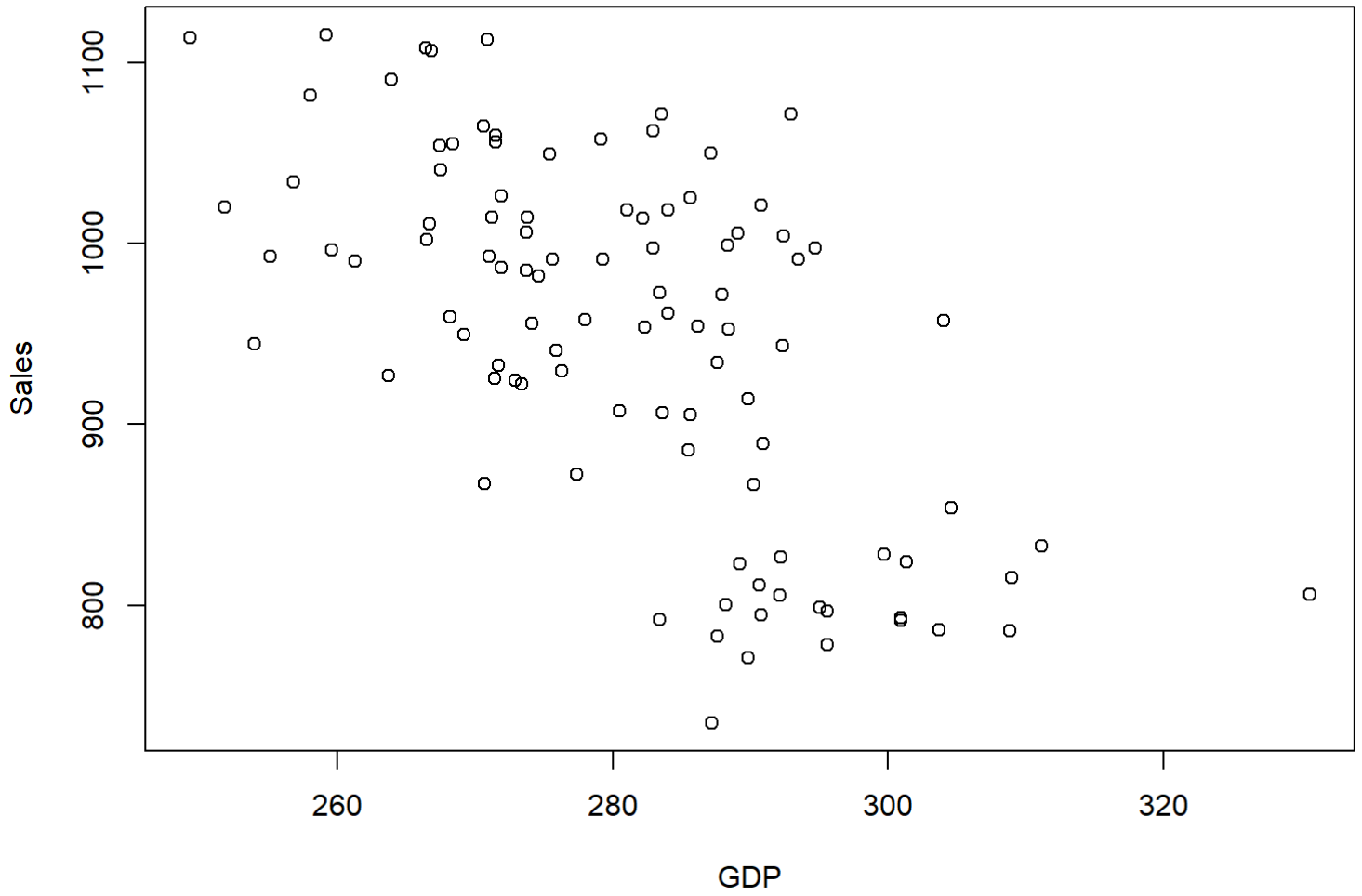

Sales
Thousands
1100
1000
900
800
1980
1985
1990
1995
2000
2005
Year

Construct scatterplots of (AdBudget,Sales) and (GDP,Sales), with Sales on the vertical axes.

```
plot(Sales ~ AdBudget, data=tute1)
```

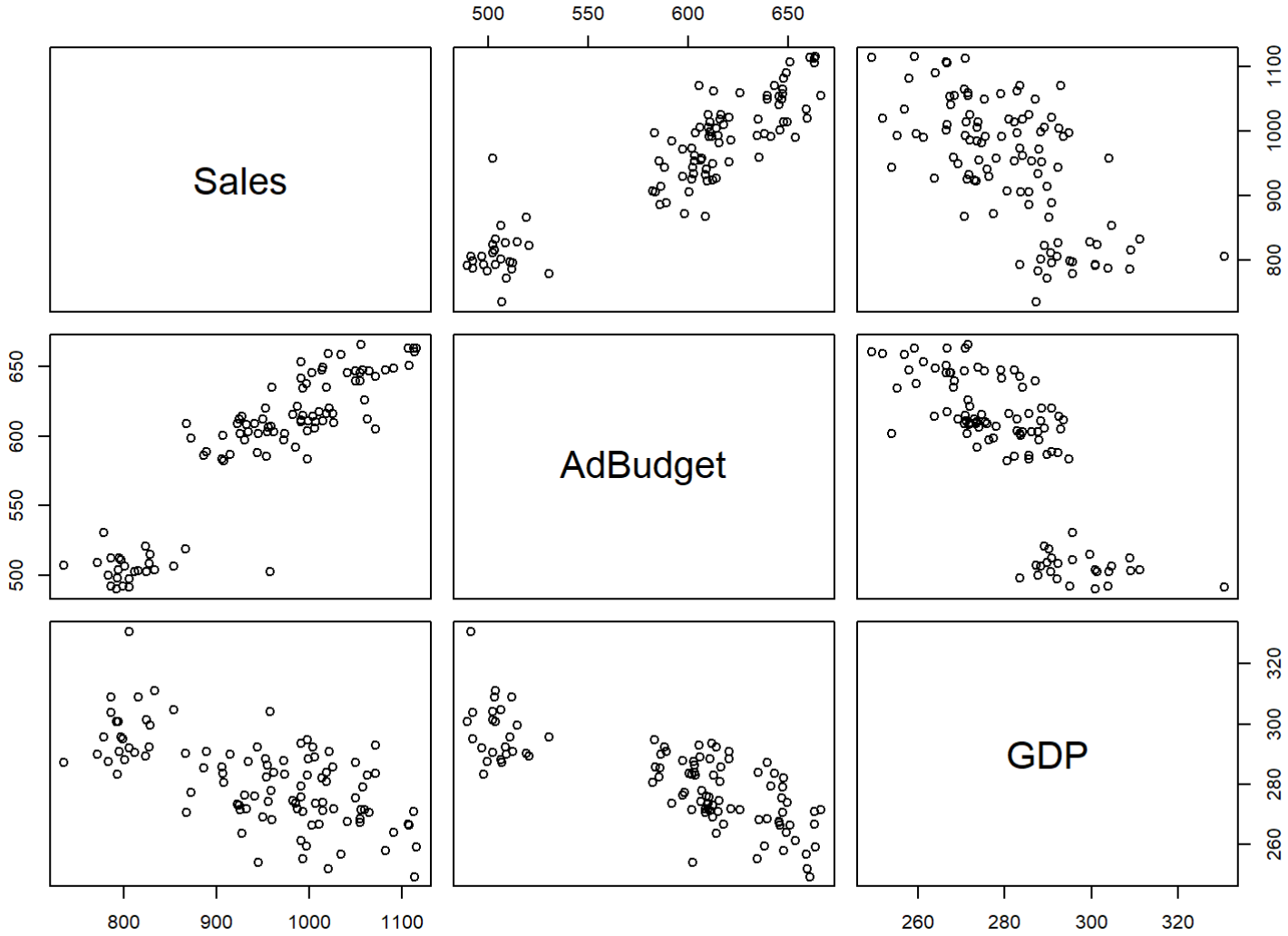

Sales
800
900
1000
1100
500
550
600
650
AdBudget

```
plot(Sales ~ GDP, data=tute1)
```



Do a scatterplot matrix of the three variables.

```
pairs(as.data.frame(tute1))
```



What is plotted in each panel?

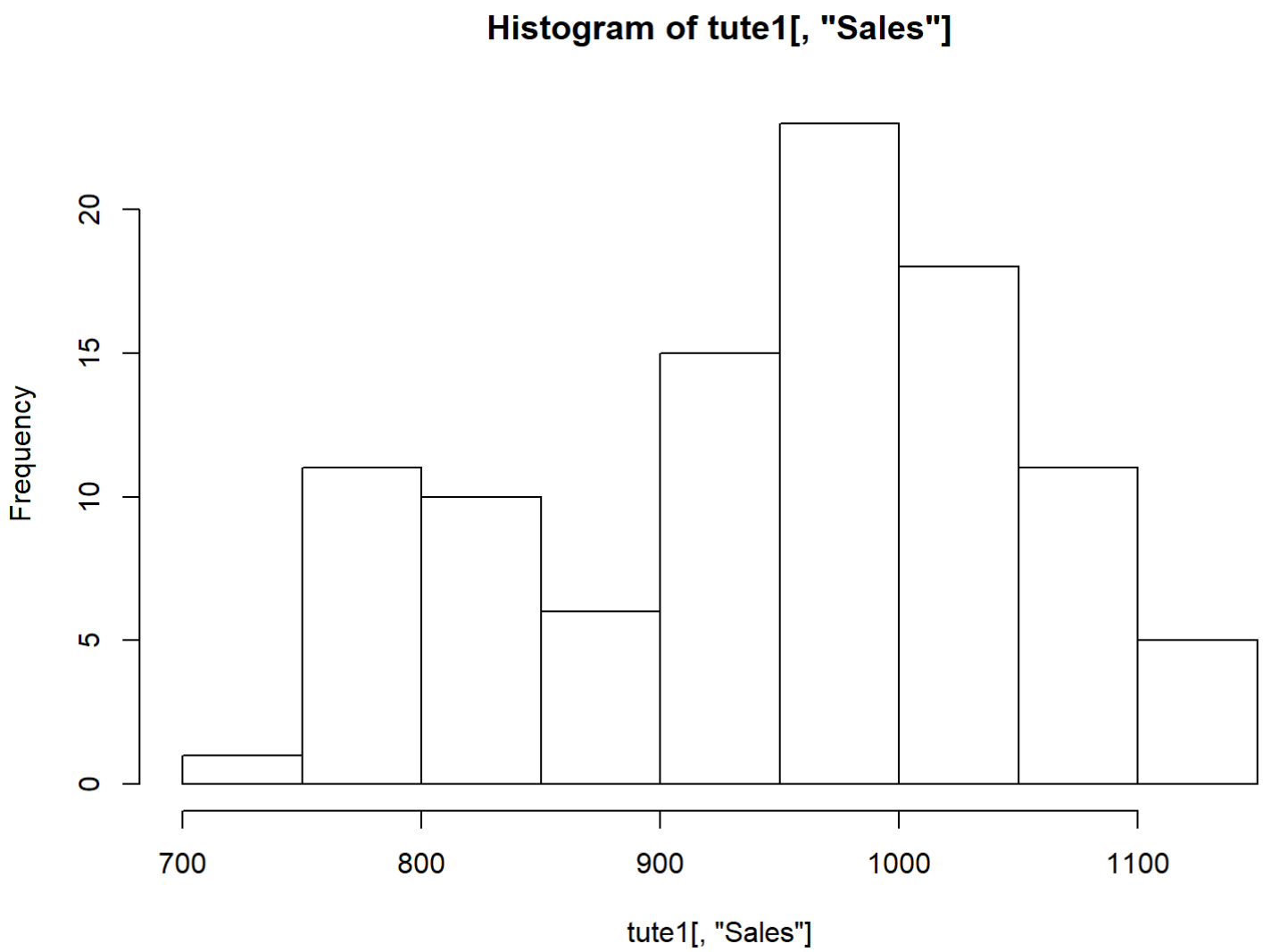
Use the summary command to get summary information about the data:

```
summary(tute1)
```

```
##      Sales          AdBudget         GDP
##  Min.   : 735.1   Min.   :489.9   Min.   :249.3
##  1st Qu.: 871.1   1st Qu.:569.5   1st Qu.:271.4
##  Median : 960.6   Median :608.5   Median :282.6
##  Mean   : 948.7   Mean   :591.9   Mean   :281.2
##  3rd Qu.:1018.7   3rd Qu.:635.0   3rd Qu.:290.3
##  Max.   :1115.5   Max.   :665.9   Max.   :330.6
```

Produce some more plots of the data:

```
hist(tute1[,"Sales"])
hist(tute1[,"AdBudget"])
```


Histogram of tute1[, "Sales"]
Frequency
0
5
10
15
20
700
800
900
1000
1100
tute1[, "Sales"]

Also do a correlation test.

```
cor.test(tute1[,"Sales"],tute1[,"AdBudget"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  tute1[, "Sales"] and tute1[, "AdBudget"]
## t = 21.423, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8657037 0.9370976
## sample estimates:
##       cor
## 0.9077655
```

Now try using RStudio as a calculator. Figure out what each of the following is doing.

```
(100+2)/3
5*10^2
1/0
0/0
(0i-9)^(1/2)
sqrt(2 * max(-10, 0.2, 4.5))
x <- sqrt(2 * max(-10, 0.2, 4.5)) + 100
x
log(100)
log(100, base=10)
```

Save the workfile, and exit RStudio.

More tutorials

There are dozens of R tutorials available on the web. Some of the best of them are listed below:

- Try R Code School
- DataCamp Introduction to R
- R tutorial (Clarkson University)
- Coursera R Programming

And plenty of guides and references such as:

- Using R for Time Series Analysis
- Kickstarting R

Search google for documentation of R commands and packages or for suggestions on how to correct errors (https://www.google.com)

Other references such as

Other references at (https://www.rstudio.com/online-learning/#R)