

## Section 1: Introducing plot function

Review of important functions covered in this section:

### Arguments of plot function

type	Specifies the type of plot to be drawn.
main	Title
sub	Sub-title
xlab	Horizontal axis title
ylab	Vertical axis title
<b>Supplementary functions:</b>	
legend	adds legend in an existing plot.
par	Helps set graphical parameters.
title	adds title, sub-title or axis titles in an existing plot.
text	adds text in an existing plot.
pdf	saves the graphical output as a pdf file.
dev.off()	shuts down the active graphics device.

### Exercise

In R, type `library(help="datasets")` to take a look at the base R dataset. We'll work with Freeny's data on quarterly revenue and explanatory variables.

- Print this data in R console first.

**freeny**

- Take a look at the help file associated with this dataset.

**?freeny**

- Check the mode of this data.

**mode(freeny)**

- Convert the list as a matrix and name it `freeny.mat`. Then convert the created matrix `freeny.mat` as a data frame and name it `freeny.df`.

**freeny.mat <- as.matrix(freeny)**

**freeny.df <- as.data.frame(freeny.mat)**

- Check the names of the column names of `freeny.df`.

**names(freeny.df)**

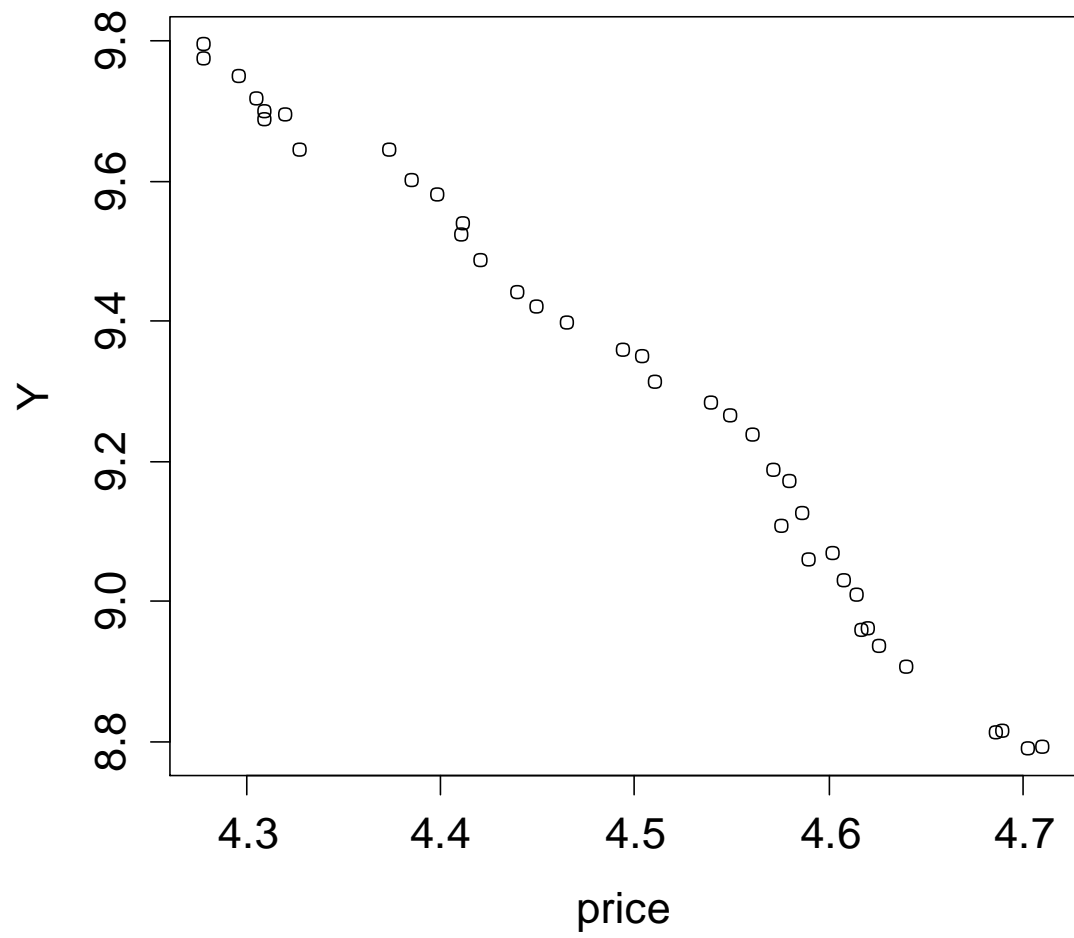
- Save the column "y" as an independent R object in R and name it `Y`. Also, save the column "price.index" as an independent R object in R and name it `price`.

**Y <- freeny.df\$y**

**price <- freeny.df\$price.index**

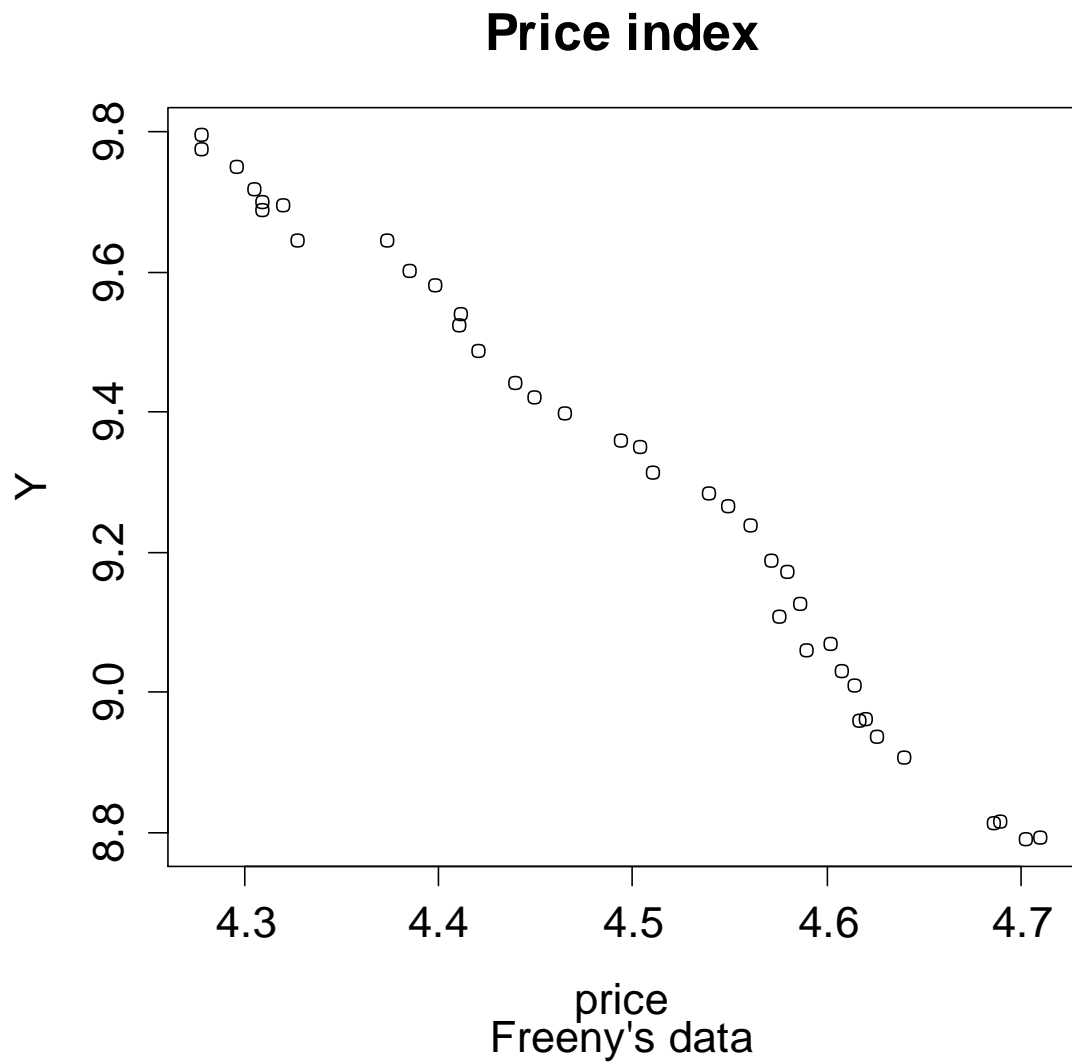
g. Plot a scatterplot with `price` in the horizontal axis and `Y` in the vertical axis.

```
plot(price, Y)
```



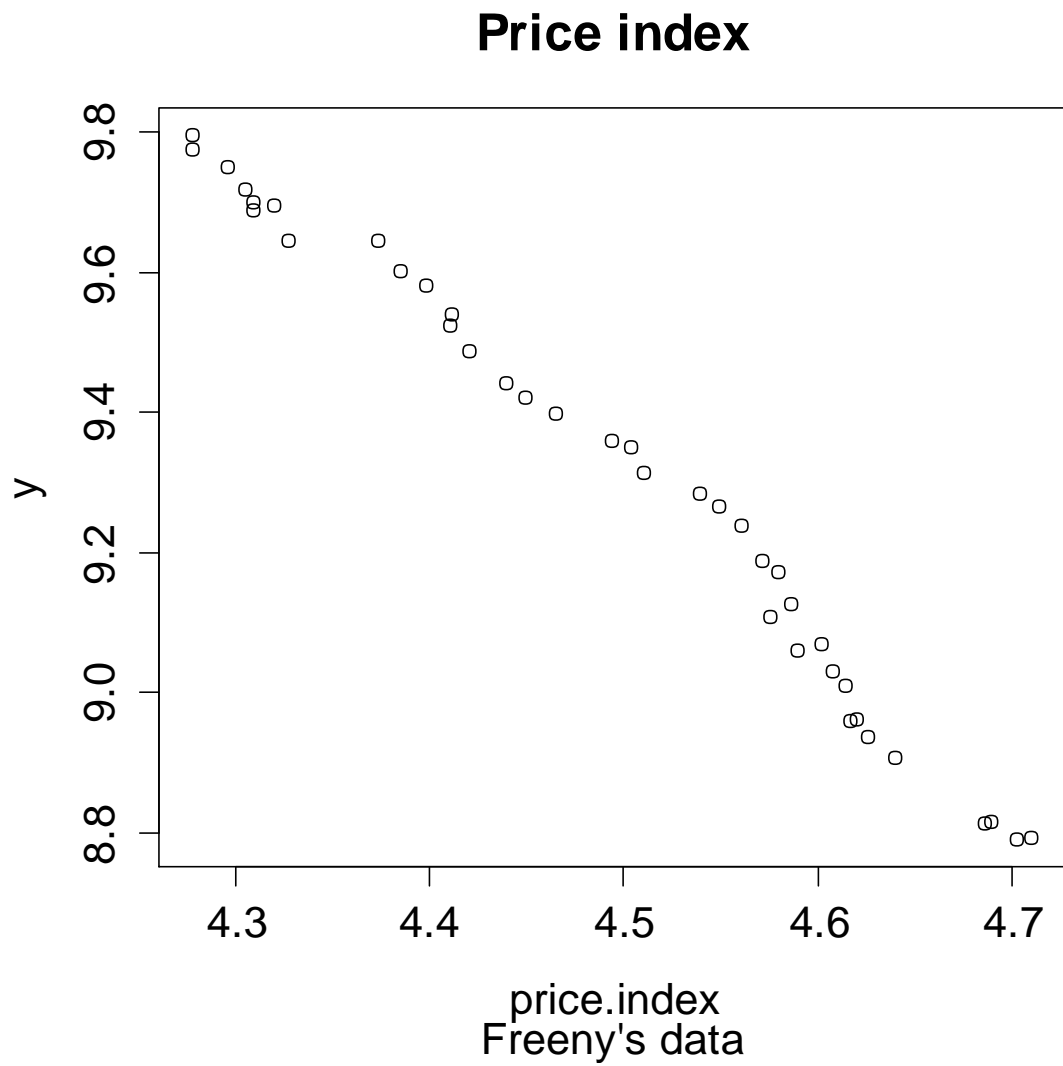
- h. After plotting the above scatterplot, add a title of the plot "Price index" and a subtitle "Freeny's data"

```
title(main="Price index", sub = "Freeny's data")
```



- i. Use data argument in the `plot` function to get the same scatterplot using the column names directly (not using the created independent objects such as `Y` and `price`).

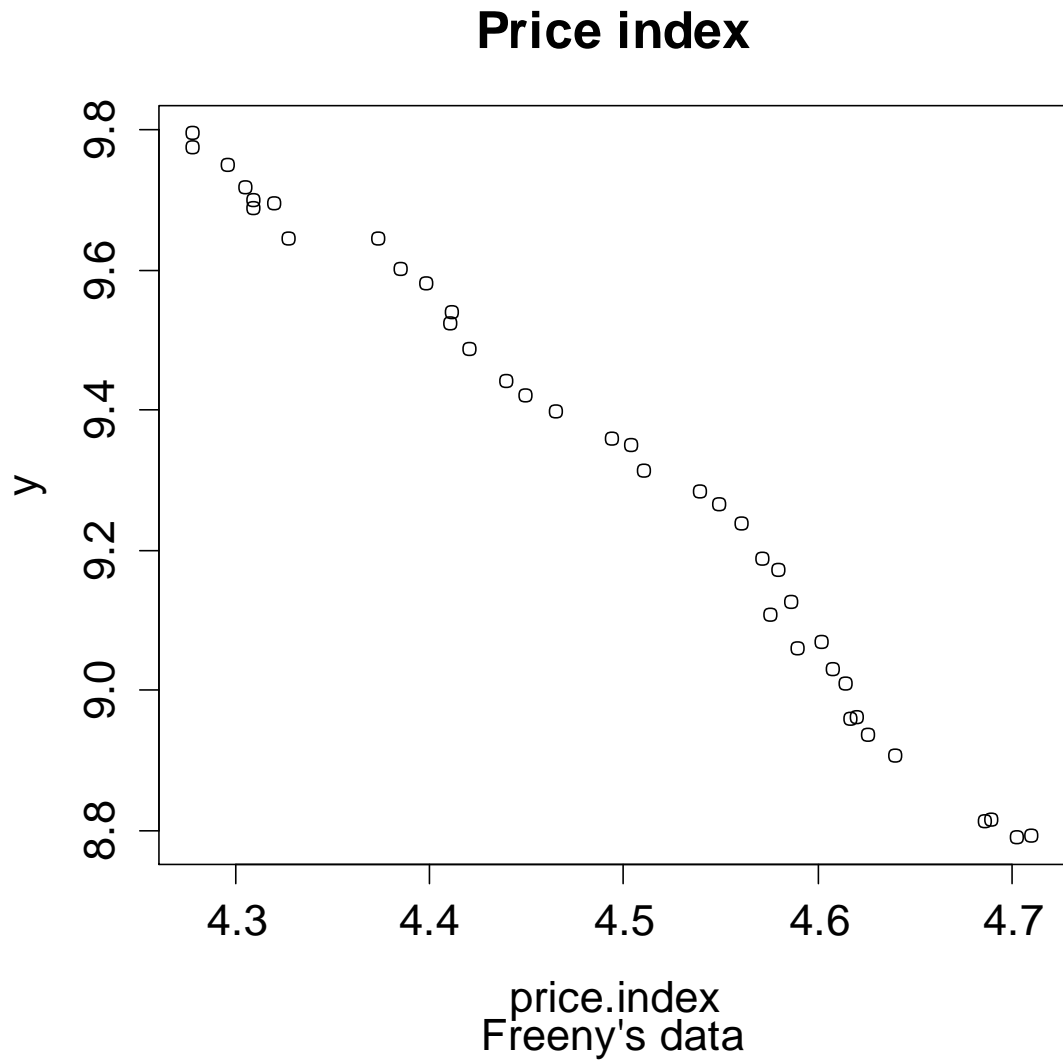
```
plot(y ~ price.index, data = freeny.df, main="Price index",  
sub = "Freeny's data")
```



j. Recreate the same plot without using the `data` argument in the `plot` function.

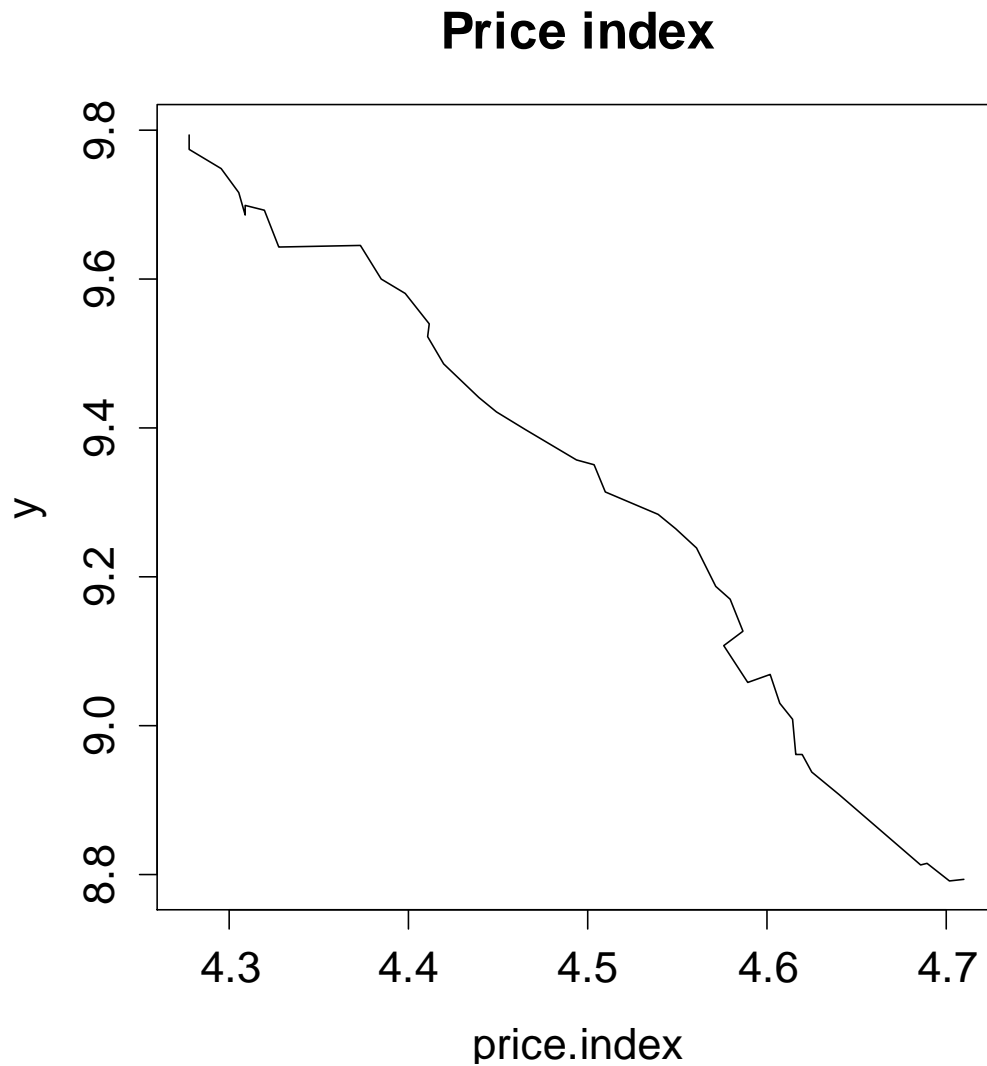
This time use `with` function.

```
with(freeny.df, plot(y ~ price.index, main="Price index", sub  
= "Freeny's data"))
```

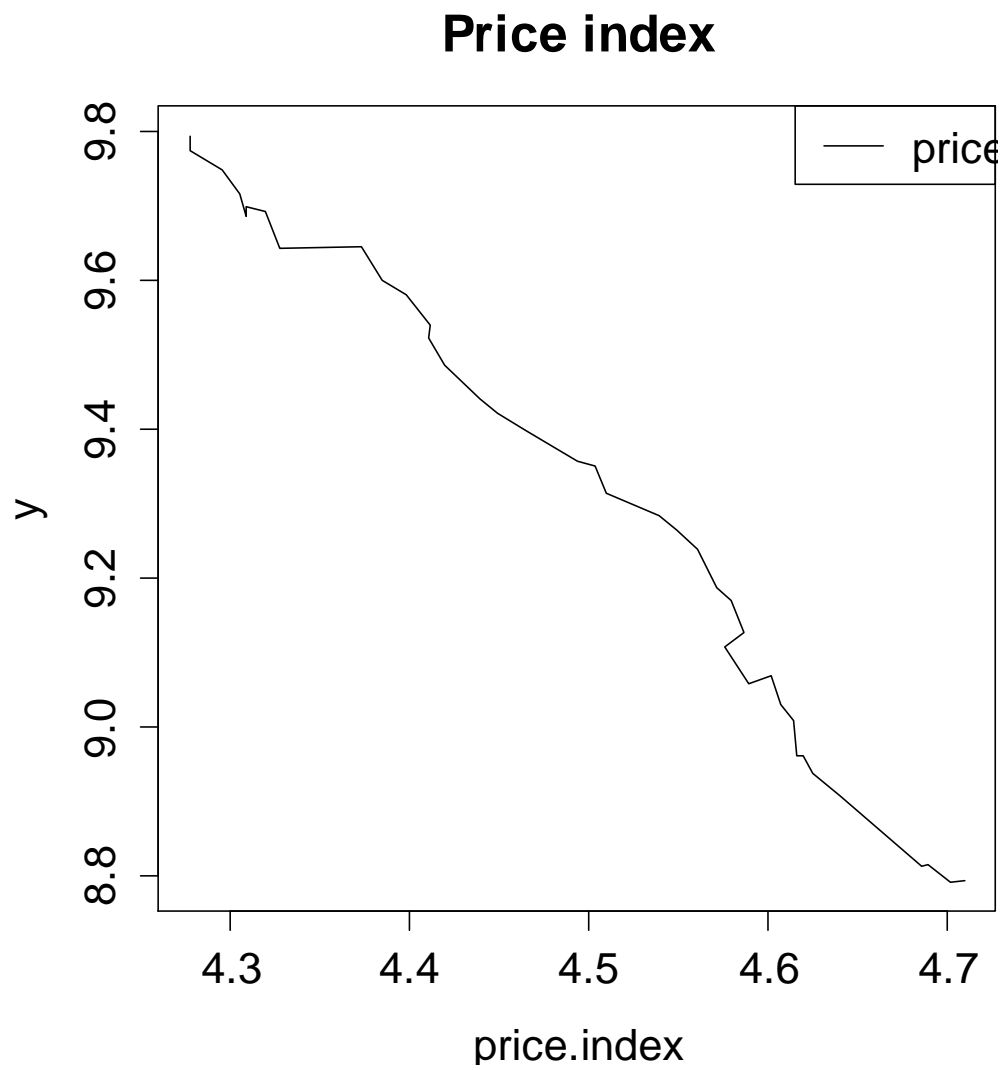


k. Create a margin such that you have 4cm in all the directions.  
`par(mar=c(bottom=4, left=4, top=4, right=4))`

l. Plot the data with `price.index` in the horizontal axis and `y` in the vertical axis as a line. See `?plot` for details of `type` argument.  
`plot(y ~ price.index, data = freeny.df, main="Price index",  
sub = "Freeny's data", type = "l")`

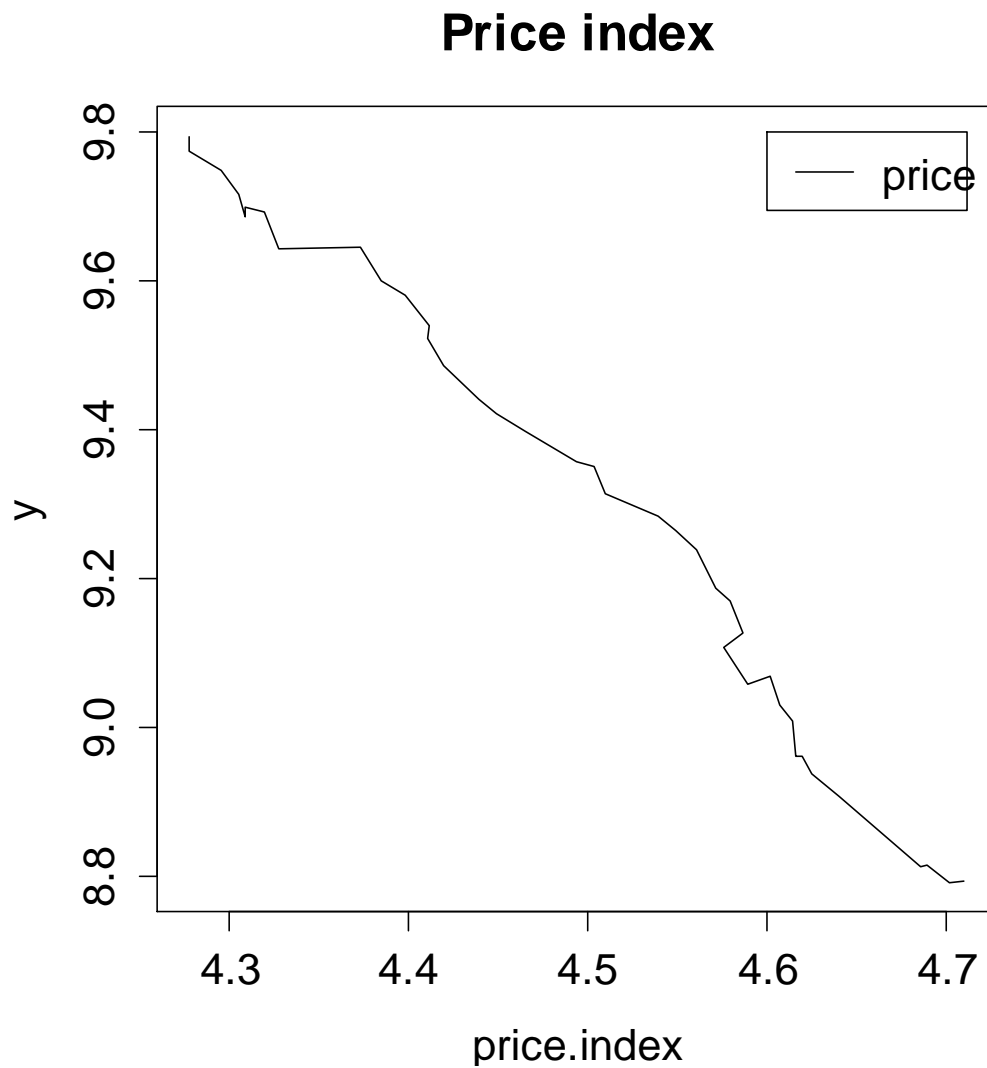


m. Add a legend in the top right corner with legend "price" and line type as 1.  
`legend("topright", legend=c("price"), lty = 1)`



- n. Instead of using "topright" in the legend function, use co-ordinates (4.6, 9.8) to place the legend.

```
plot(y ~ price.index, data = freeny.df, main="Price index",  
sub = "Freeny's data", type = "l")  
legend(4.6, 9.8, legend=c("price"), lty = 1)
```





- o. Find out the working directory of your R session.

```
getwd()
```

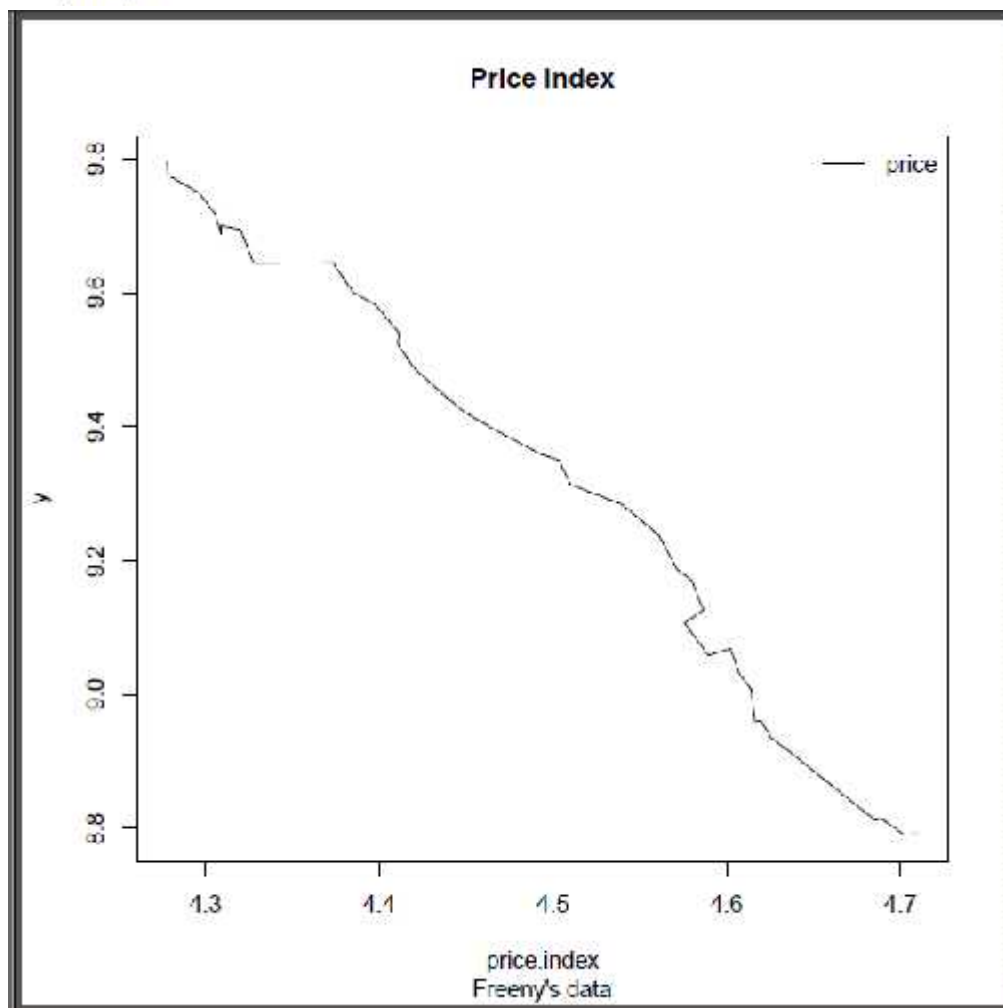
- p. In that working directory, save the above plot as a plotfile.PDF file.

```
pdf("plotfile.pdf")
```

```
plot(y ~ price.index, data = freeny.df, main="Price index",  
sub = "Freeny's data", type = "l")
```

```
legend("topright", legend=c("price"), lty = 1)
```

```
dev.off()
```

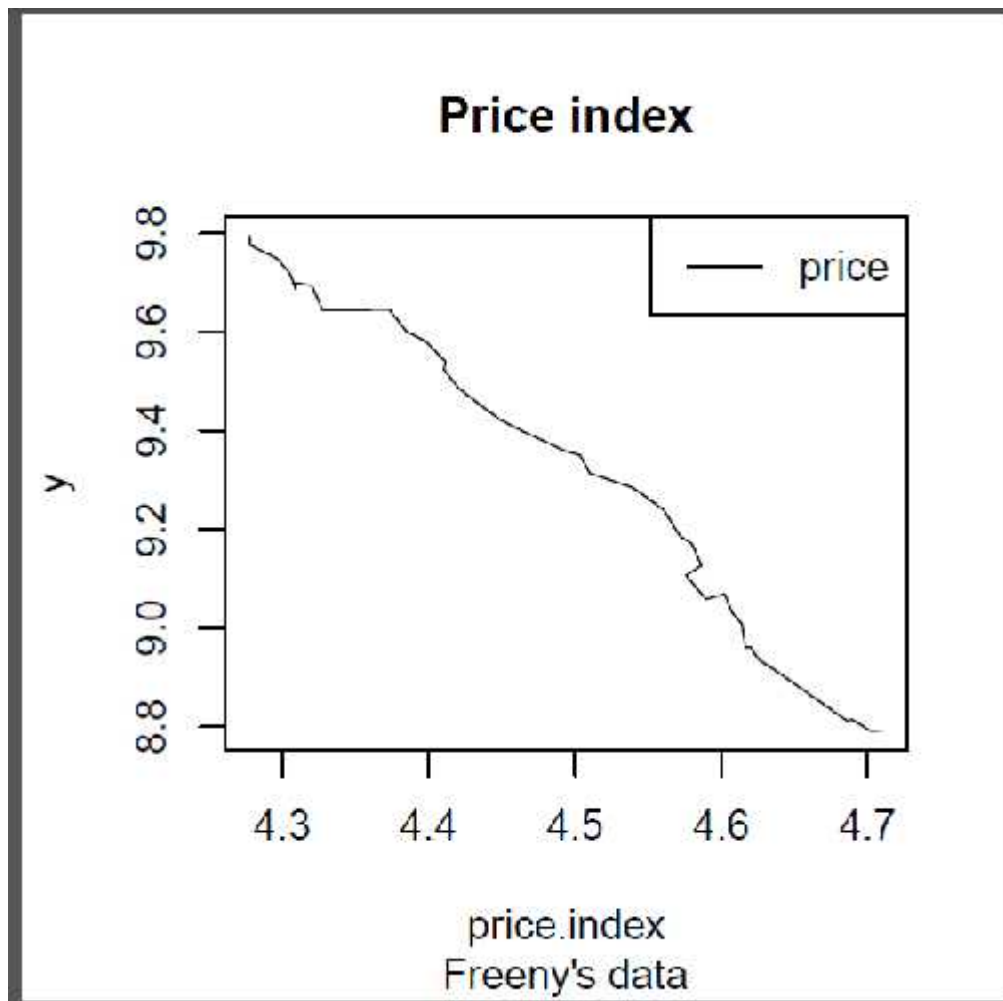


- q. In the working directory, again save the above plot as a plotfile2.PDF file, with both height and width as 4cm. Take a look at ?pdf to find the appropriate arguments.

```
pdf("plotfile2.pdf", width=4, height=4)
plot(y ~ price.index, data = freeny.df, main="Price index",
sub = "Freeny's data", type = "l")
legend("topright", legend=c("price"), lty = 1)
dev.off()
```



plotfile2.pdf



r. In the working directory, save the above plot as a plotfile.PS file.

```
postscript("plotfile.ps")
plot(y ~ price.index, data = freeny.df, main="Price index",
sub = "Freeny's data", type = "l")
legend("topright", legend=c("price"), lty = 1)
dev.off()
```



plotfile.ps

