

Section 2: Further control over 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
col.main	Specifies the color of the title
col.lab	Specifies the color of the axes titles
cex.main	Font size of the title
cex.lab	Font size of the axes titles
font.main	Font of the title
lwd	Specifies width of the line
lty	Specifies line type
Supplementary functions:	
box	Adds a box around the current plot
axis	Adds axes around the current plot
mtext	Adds text in the margin of the current plot
par	Sets graphical parameters
layout	Helps specify the layout of a complex plot
lines	Adds a line in the current plot
hist	Draws Histogram

Exercise

We'll again work with Freeny's data on quarterly revenue and explanatory variables (use `freeny.df` that was created in the previous section's exercise as follows:

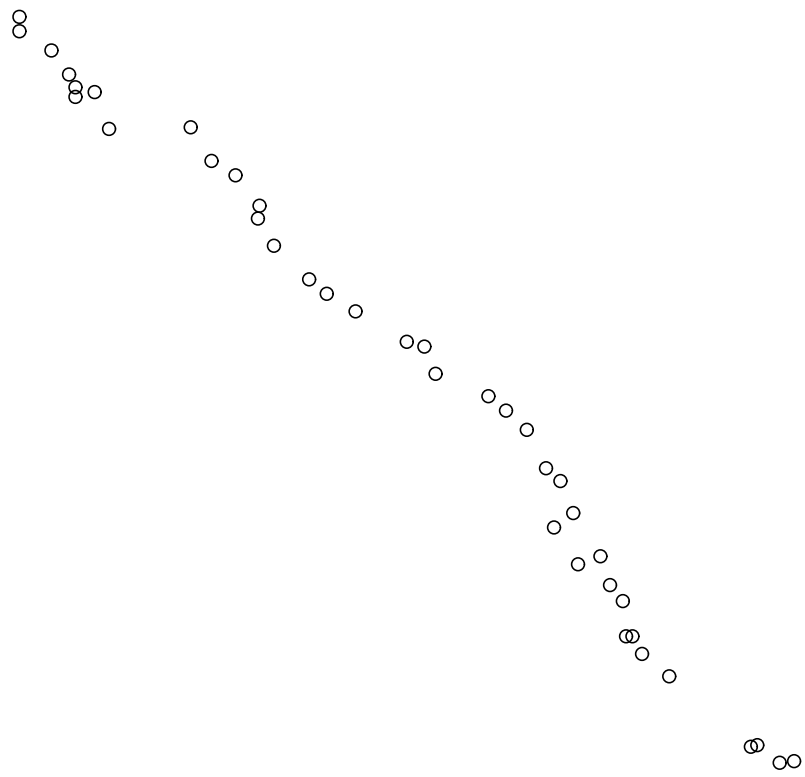
```
freeny
```

```
freeny.mat = as.matrix(freeny)
```

```
freeny.df = as.data.frame(freeny.mat)
```

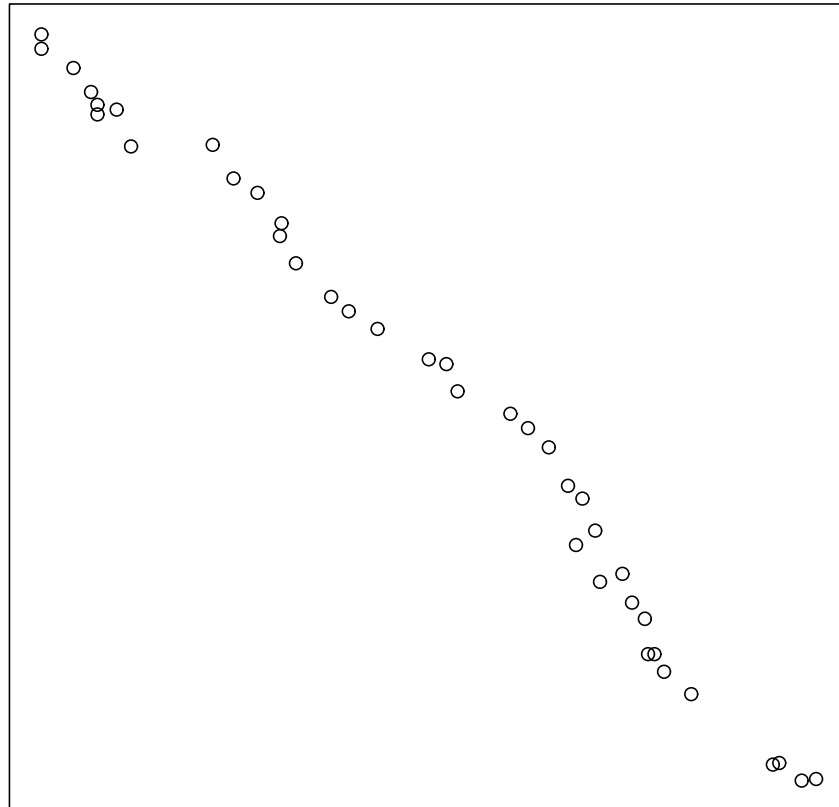
- price.index and y are two variables in the dataset `freeny.df`. Plot a scatterplot with price.index in the horizontal axis and y in the vertical axis, without any axes labels or box around it.

```
plot(y ~ price.index, data = freeny.df, axes = F, xlab = NA,  
ylab = NA)
```



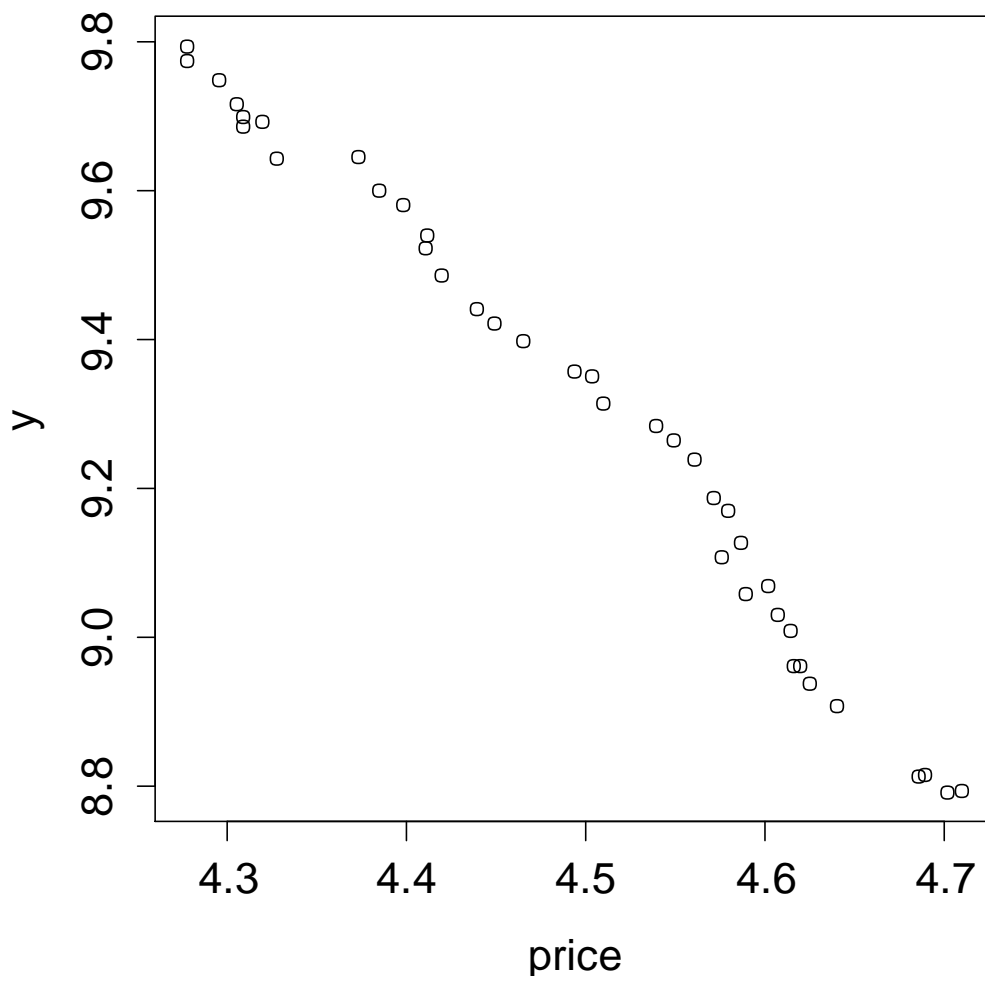
b. Put a box around the above plot.

box()



c. Insert axis labels and axis titles in the above plot.

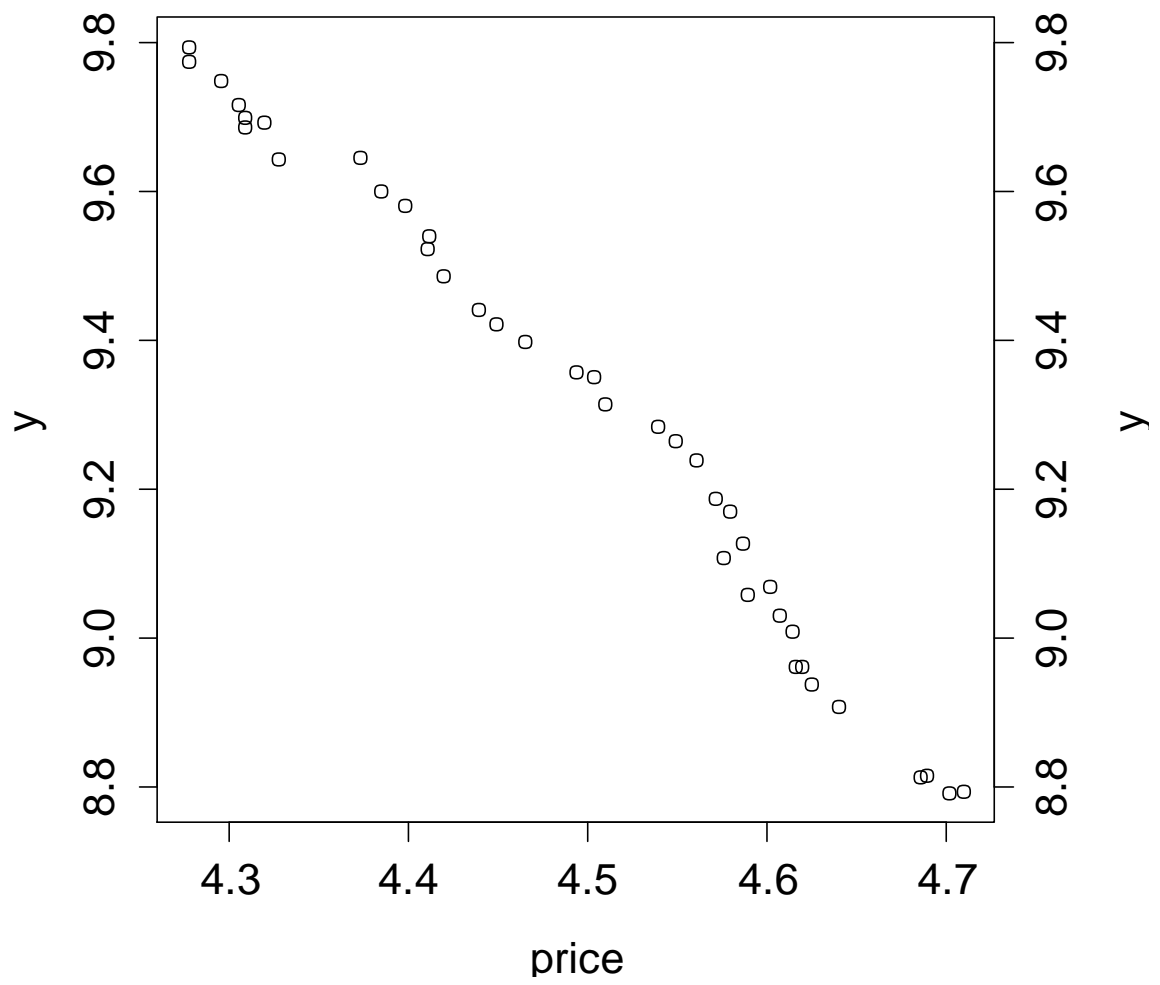
```
axis(side = 1)  
mtext(side = 1, "price", line = 3)  
axis(side = 2)  
mtext(side = 2, "y", line = 3)
```



d. Put the axis label and title in the right-hand side of the plot as well.

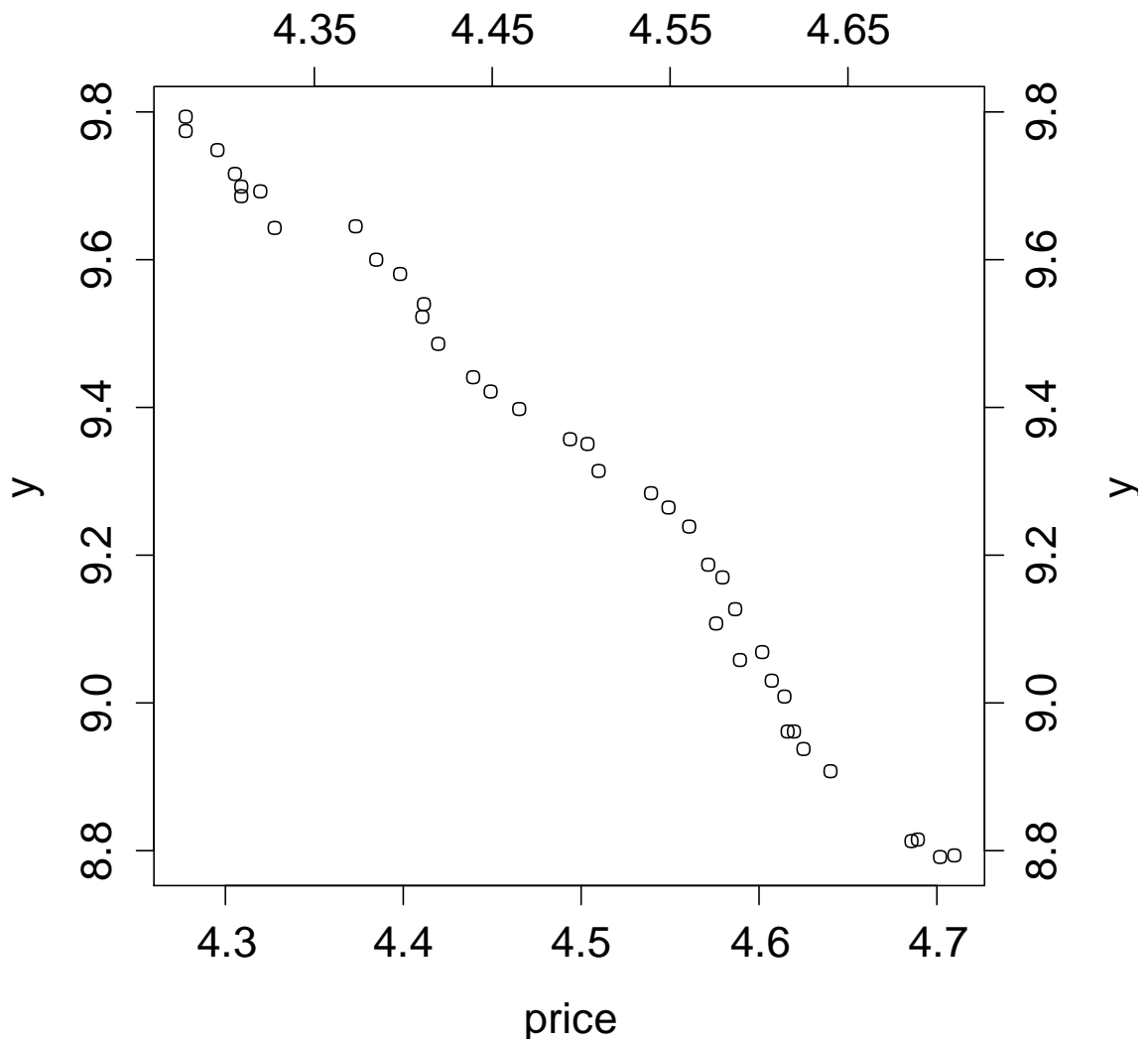
```
axis(side = 4)
```

```
mtext(side = 4, "y", line = 3)
```



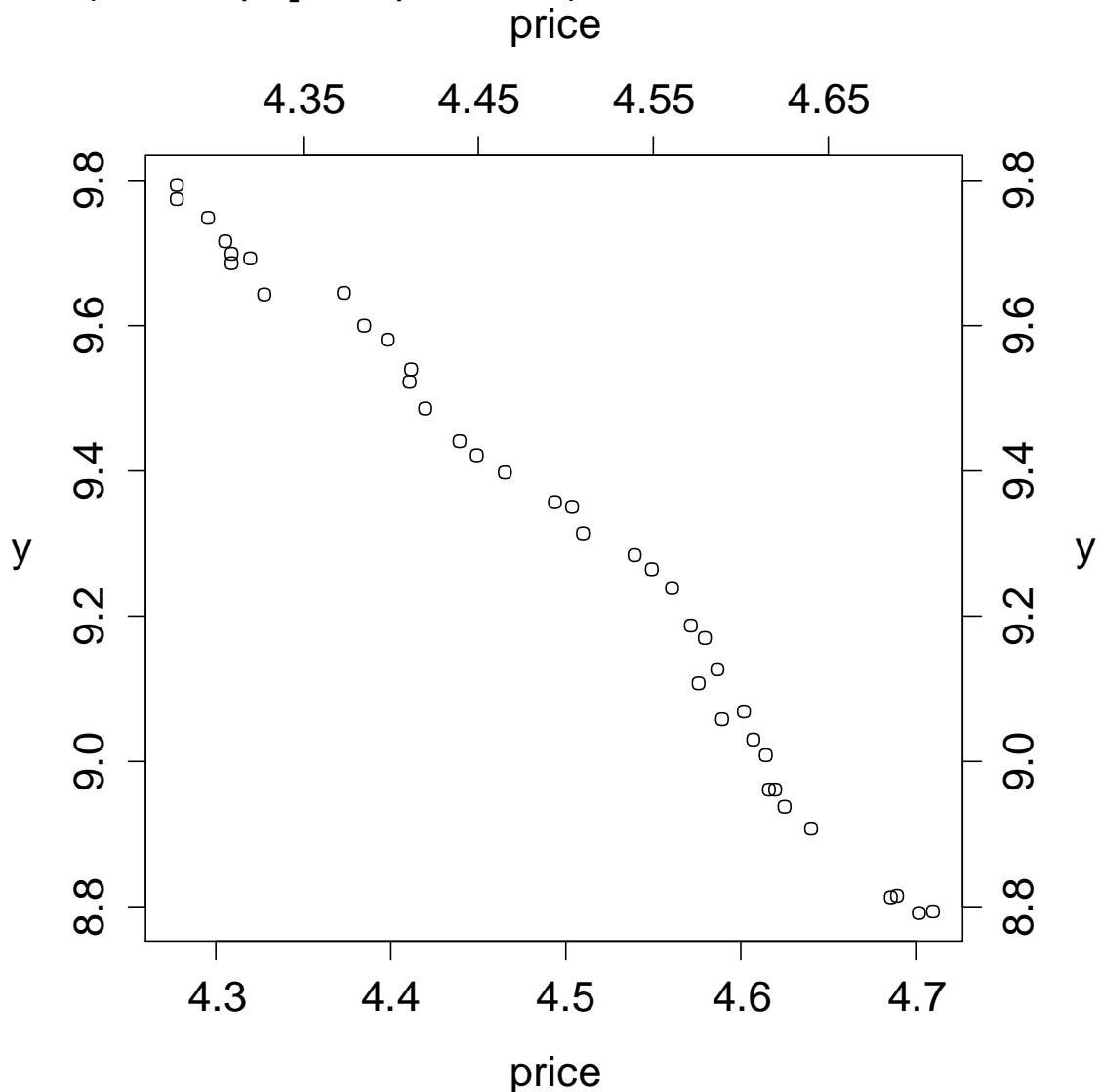
- e. Put the axis label and title at the top of the plot as well. This time, customize the axis labels to 4.35, 4.45, 4.55 and 4.65.

```
new.price = c(4.35, 4.45, 4.55, 4.65)
axis(side = 3, at = new.price, labels = new.price)
mtext(side = 3, "price", line = 3)
```



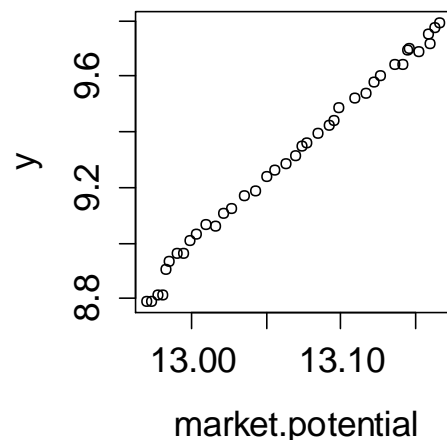
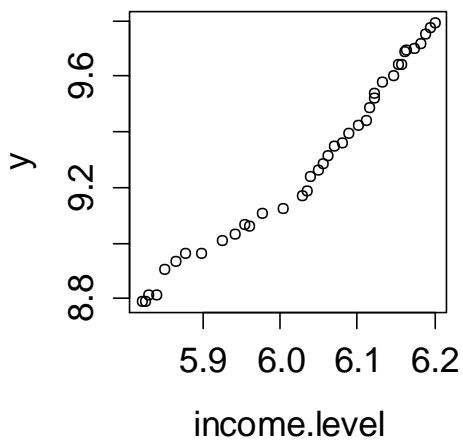
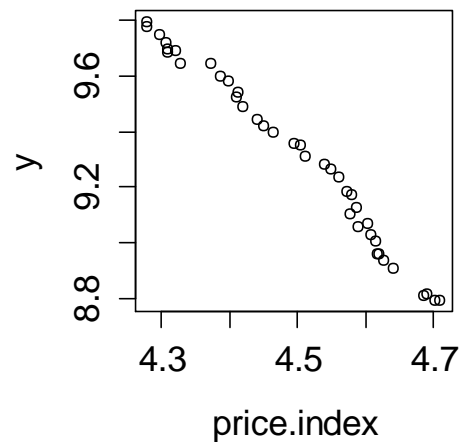
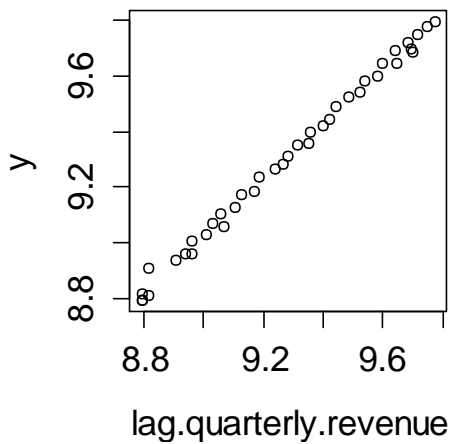
- f. Recreate the above plot, but this time, put the vertical axes title in a horizontal manner.

```
plot(y ~ price.index, data = freeny.df, axes = F, xlab = NA,
     ylab = NA)
box()
axis(side = 1)
mtext(side = 1, "price", line = 3)
axis(side = 2)
mtext(side = 2, "y", line = 3, las = 1)
axis(side = 4)
mtext(side = 4, "y", line = 3, las = 1)
new.price = c(4.35, 4.45, 4.55, 4.65)
axis(side = 3, at = new.price, labels = new.price)
mtext(side = 3, "price", line = 3)
```



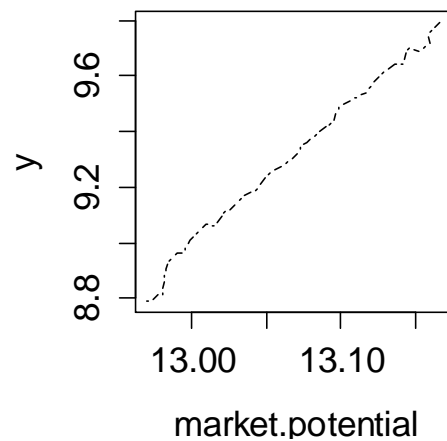
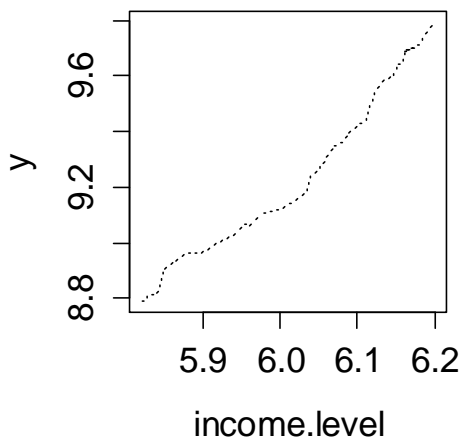
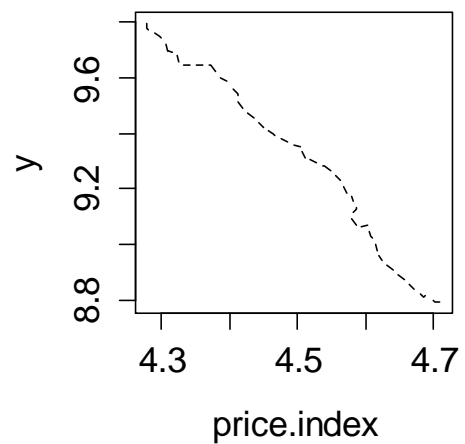
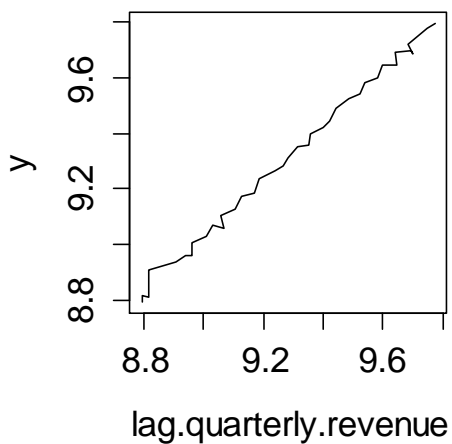
- g. Split the plot window 4 ways and draw 4 scatterplots of y in the vertical axis and each one of variables `lag.quarterly.revenue`, `price.index`, `income.level`, `market.potential` in the horizontal axes separately.

```
par(mfrow = c(2,2))  
plot(y ~ lag.quarterly.revenue, data = freeny.df)  
plot(y ~ price.index, data = freeny.df)  
plot(y ~ income.level, data = freeny.df)  
plot(y ~ market.potential, data = freeny.df)
```



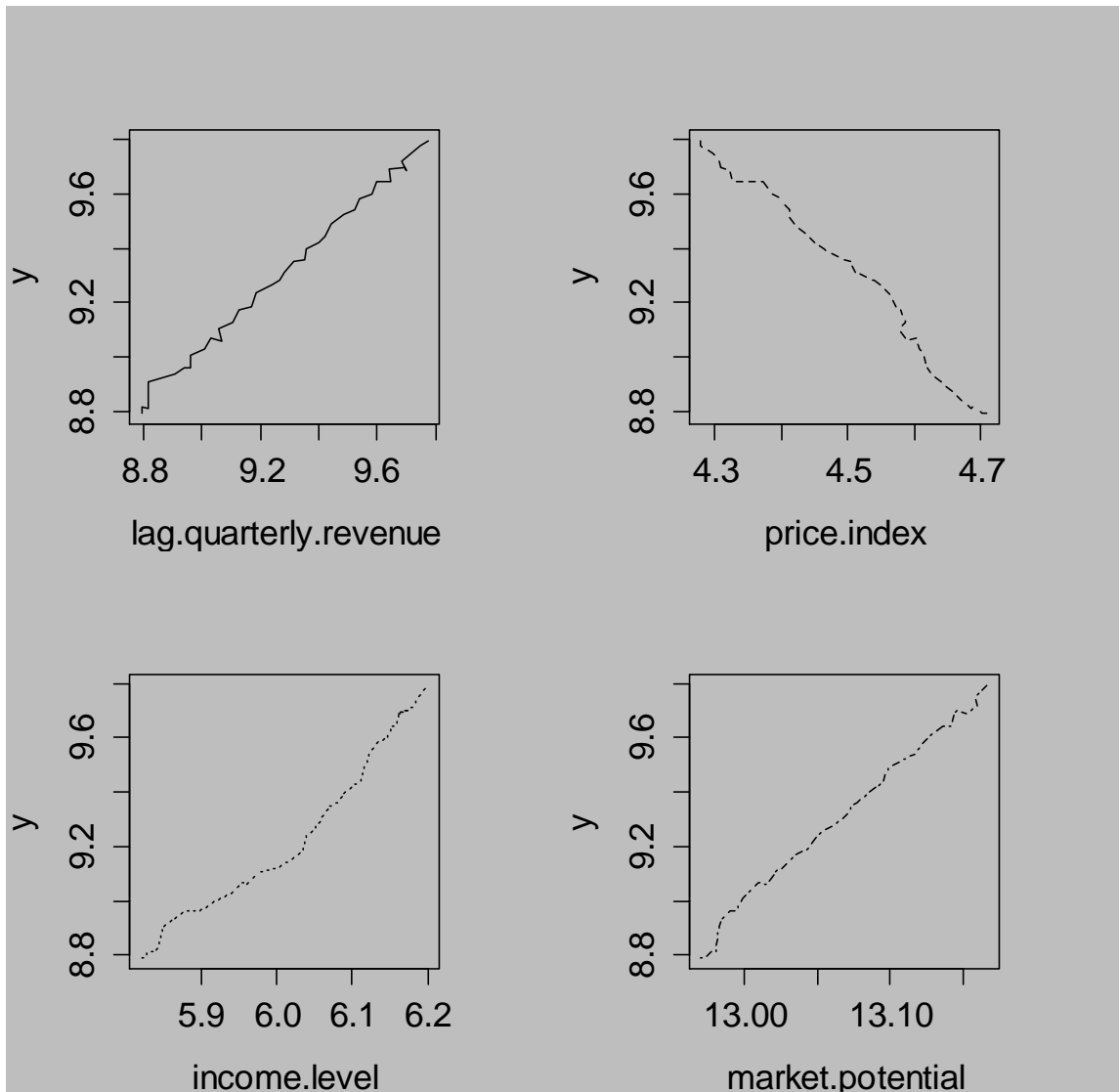
- h. Re-draw the above plots again, but this time instead of points, use solid, dashed, dotted, dotdash lines respectively.

```
par(mfrow = c(2,2))  
plot(y ~ lag.quarterly.revenue, data = freeny.df, type = "l",  
lty = "solid")  
plot(y ~ price.index, data = freeny.df, type = "l", lty =  
"dashed")  
plot(y ~ income.level, data = freeny.df, type = "l", lty =  
"dotted")  
plot(y ~ market.potential, data = freeny.df, type = "l", lty =  
"dotdash")
```



i. Re-draw the above plot with grey background. Hint: `?par`

```
par(mfrow = c(2,2), bg = "grey")
plot(y ~ lag.quarterly.revenue, data = freeny.df, type = "l",
lty = "solid")
plot(y ~ price.index, data = freeny.df, type = "l", lty =
"dashed")
plot(y ~ income.level, data = freeny.df, type = "l", lty =
"dotted")
plot(y ~ market.potential, data = freeny.df, type = "l", lty =
"dotdash")
```

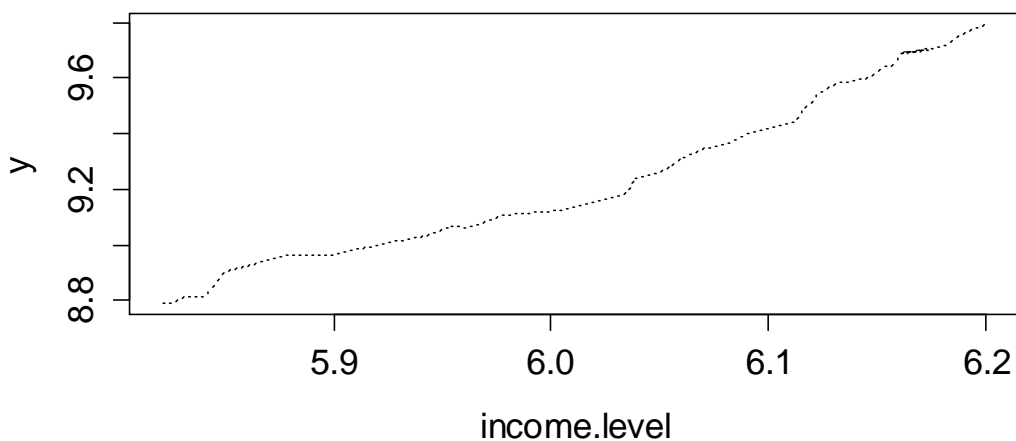
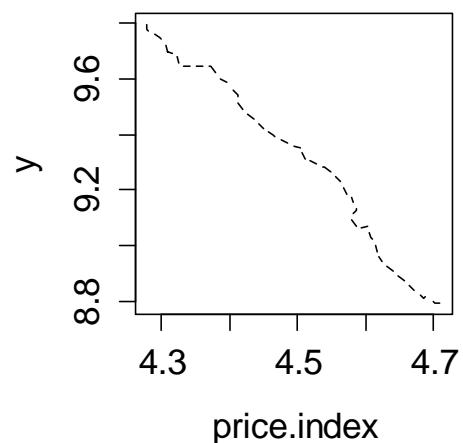
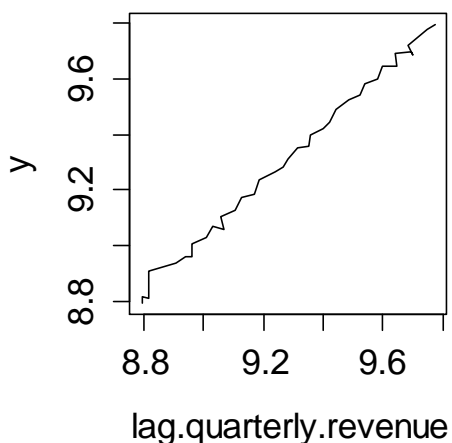


- j. Bring back the plot window in default setting.

```
par(mfrow = c(1,1), bg = "white")
```

- k. Split the plot window 3 ways such that 2 graphs fit in the above, and one larger graph fit in the slot below. Then draw 3 scatterplots of y in the vertical axis and each one of variables `lag.quarterly.revenue`, `price.index` and `income.level` in the horizontal axes separately.

```
lay.mat = matrix(c(1,2,3,3), nrow= 2, ncol = 2, byrow = T)
lay.mat
layout(lay.mat)
plot(y ~ lag.quarterly.revenue, data = freeny.df, type = "l",
lty = "solid")
plot(y ~ price.index, data = freeny.df, type = "l", lty =
"dashed")
plot(y ~ income.level, data = freeny.df, type = "l", lty =
"dotted")
```

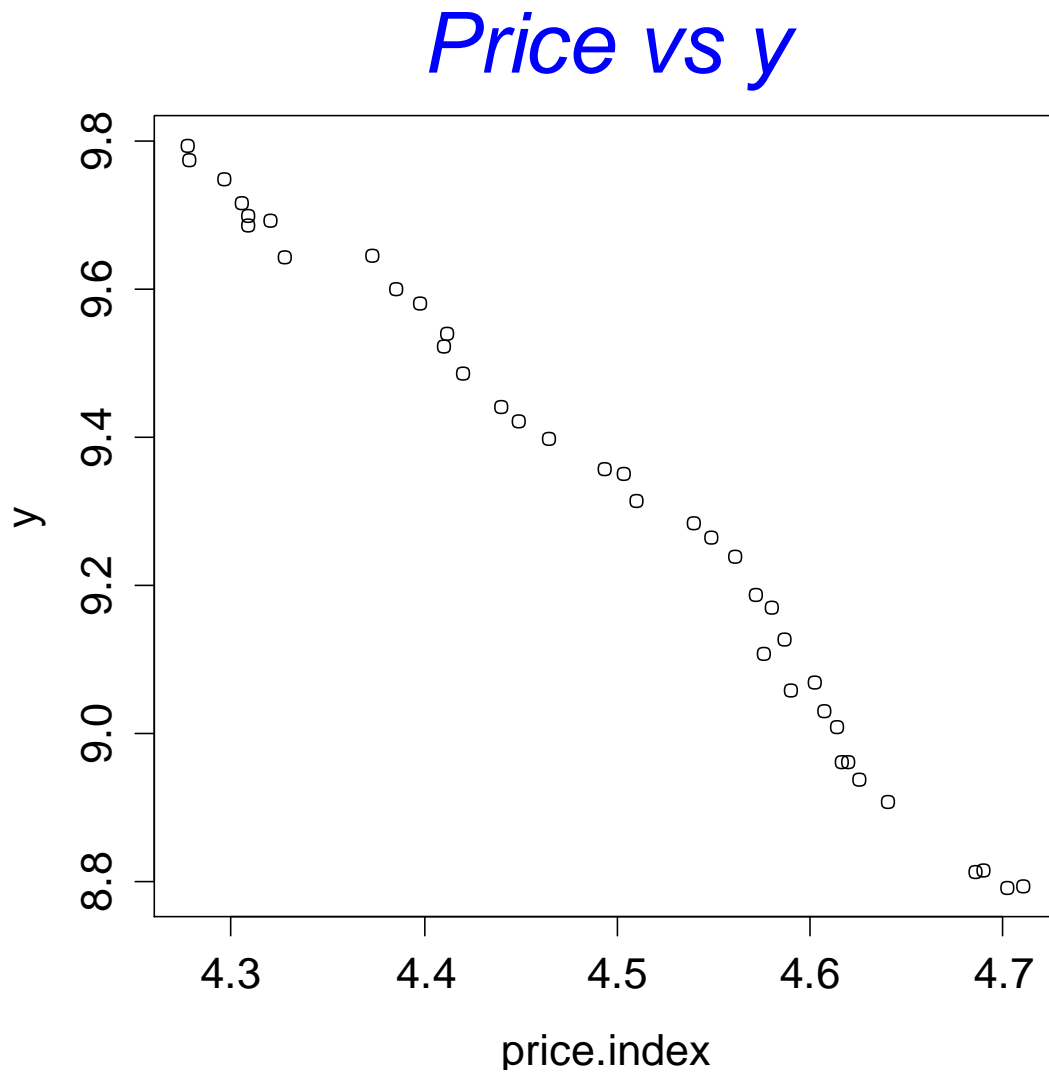


- l. Bring back the plot window in default setting.

```
layout(1)
```

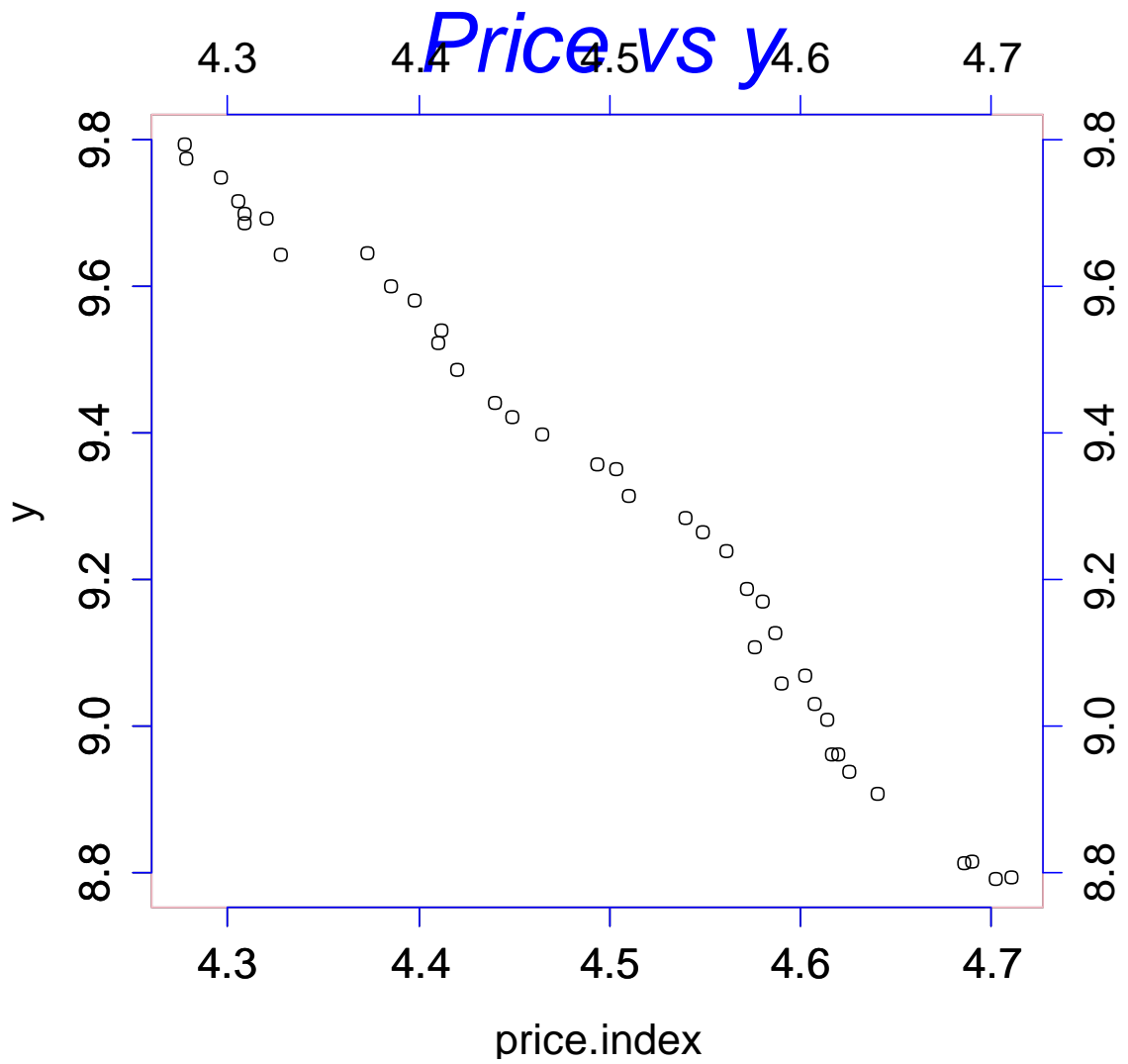
- m. Create a scatterplots of y in the vertical axis and price.index in the horizontal axis with a blue italic title “price vs y ”, double the usual title size.

```
plot(y ~ price.index, data = freeny.df, main = "Price vs y",  
col.main = "blue", font.main = 3, cex.main = 2)
```



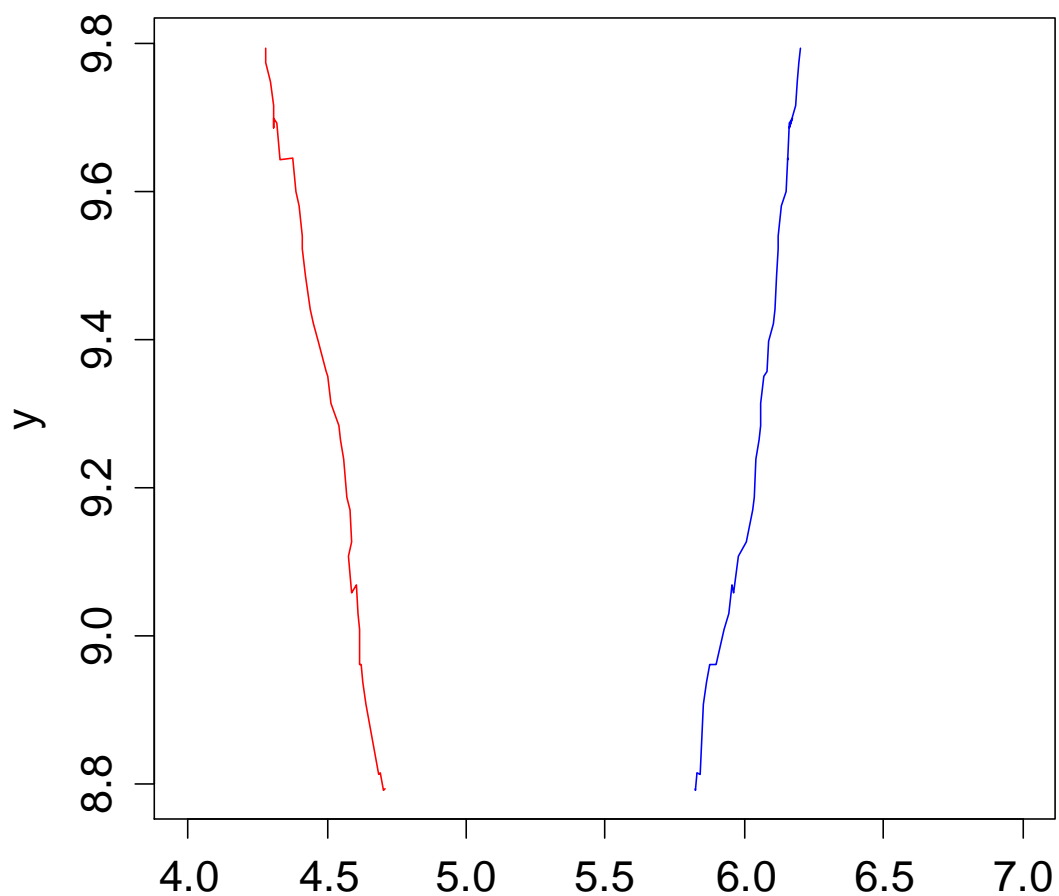
n. In the above plot, add a pink box and color all the axes as blue.

```
box(col = "pink")  
axis(side = 1, col = "blue")  
axis(side = 2, col = "blue")  
axis(side = 3, col = "blue")  
axis(side = 4, col = "blue")
```



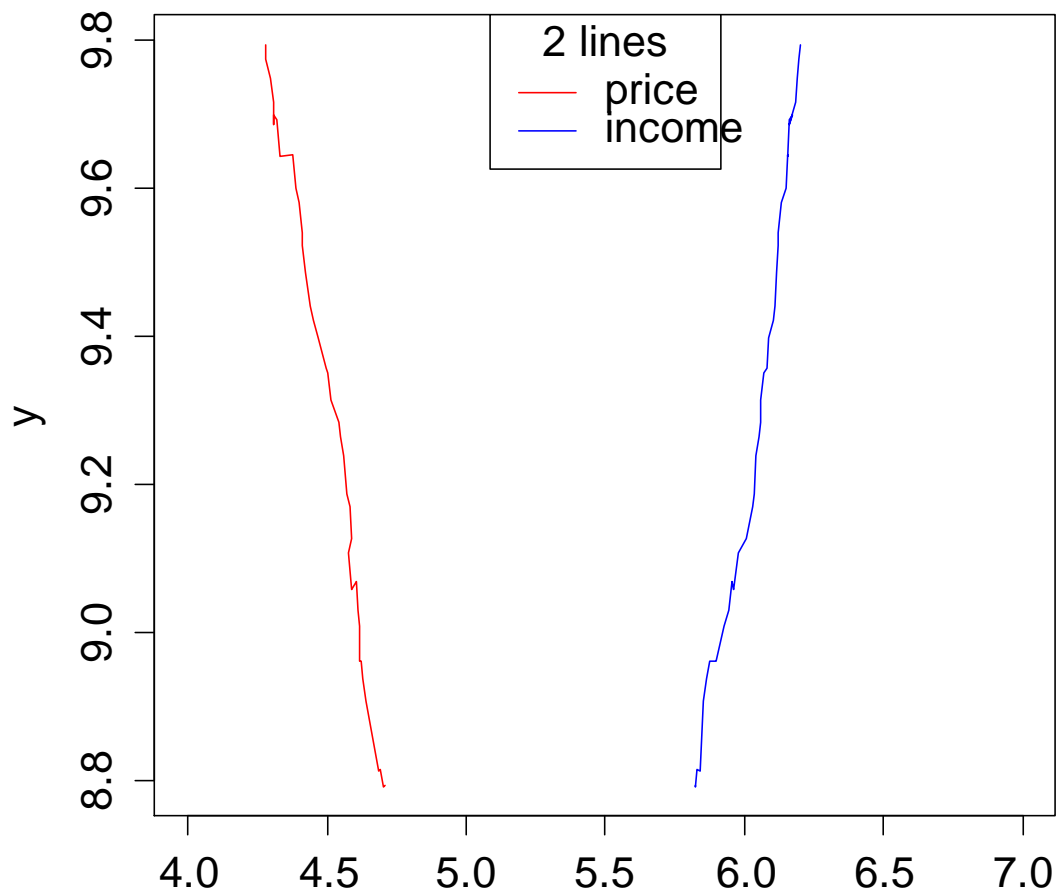
- o. Create a plots of `y` in the vertical axis and `price.index` in the horizontal axis as a red line and another plots of `y` in the vertical axis and `income.level` in the horizontal axis as a blue line. To accommodate both in the same plot, set horizontal axis margin from 4 to 7 and remove the horizontal axis title while plotting.

```
plot(y~price.index, data = freeny.df, type = "l", col = "red",  
xlim = c(4,7), xlab = NA)  
lines(y~income.level, data = freeny.df, type = "l", col =  
"blue")
```



p. Add legends at the top of the above plot specifying which line indicates what.

```
legend("topleft", legend=c("price","income"), title = "2  
lines", lty = 1, col = c("red", "blue"))
```



- q. Add a text in the middle of the plot saying "2 lines are completely separated" in two lines (hint: use "\n" in the text).

```
text(5,9.5, "2 lines are \ncompletely separated")
```

