Introduction to R

Outline

- Processing Data in R
- Programming in R
- Graphical Analysis in R
- Statistical Analysis in R

RStudio

Select Start → All Programs → RStudio → Rstudio

Select File → New File → R Script

- Script pane
- Console pane
- Workspace / History pane
- Files, Plots, Packages and Help pane

Creating Subsets in Data Frames

```
sales<- read.csv(“yearly_sales.csv”, header = TRUE);
sales<- read.csv(file.choose(), header = TRUE);
```

```
>str(sales)                 ## structure of sales
>head(sales)                   ## Top few records
>sales$sales_total
>sales$sales_total[sales$sales_total >200]
```

```
>sales[sales$sales_total >200,]
>sales[sales$sales_total >200 &
              sales$num_of_orders >5,]
```

Subset(): Creating Subsets in Data Frames

```
>subset(sales$cust_id, sales$sales_total>200)

>subset(sales, sales_total>200)

>subset(sales , sales_total
        >200 & num_of_orders >5)
```

Subset(): Creating Subsets in Data Frames……

```
> subset(sales ,sales_total >200 &
        num_of_orders >5, select = - num_of_orders)

>subset(sales ,sales_total >200 & num_of_orders
                >5, c(num_of_orders, sales_total))

>subset(sales ,sales_total >500 | num_of_orders
                >8, c(num_of_orders, sales_total))
```

Statistical functions

`rnorm, dnorm, pnorm, qnorm`	Normal distribution random sample, density, cdf and quantiles
`lm, glm, anova`	Model fitting
`loess, lowess`	Smooth curve fitting
`sample`	Resampling (bootstrap, permutation)
`.Random.seed`	Random number generation
`mean, median`	Location statistics
`var, cor, cov, mad, range`	Scale statistics
`svd, qr, chol, eigen`	Linear algebra

DESCRIPTIVE STATISTICS

```
summary(sales)

x<-sales$sales_total
y<- sales$num_of_orders

IQR(x)
mean(x)
median(x)
range(x)
```

```
sd(x)
var(x)
apply(sales [,c(1:3)],
MARGIN=2, FUN=sd)

cor(x,y)
cov(x,y)
```

Graphical functions

`plot`	Generic plot eg: scatter
`points`	Add points
`lines, abline`	Add lines
`text, mtext`	Add text
`legend`	Add a legend
`axis`	Add axes
`box`	Add box around all axes
`par`	Plotting parameters (lots!)
`colors, palette`	Use colors

Demo

```
>demo(graphics)
```

Plots for single variable

Histogram

- NormDist<-rnorm(n=500, m=24.2, sd=2.2)
- hist(NormDist)

- histinfo <-hist(NormDist)
- histinfo
  - Lists breaks, counts, density, mids…

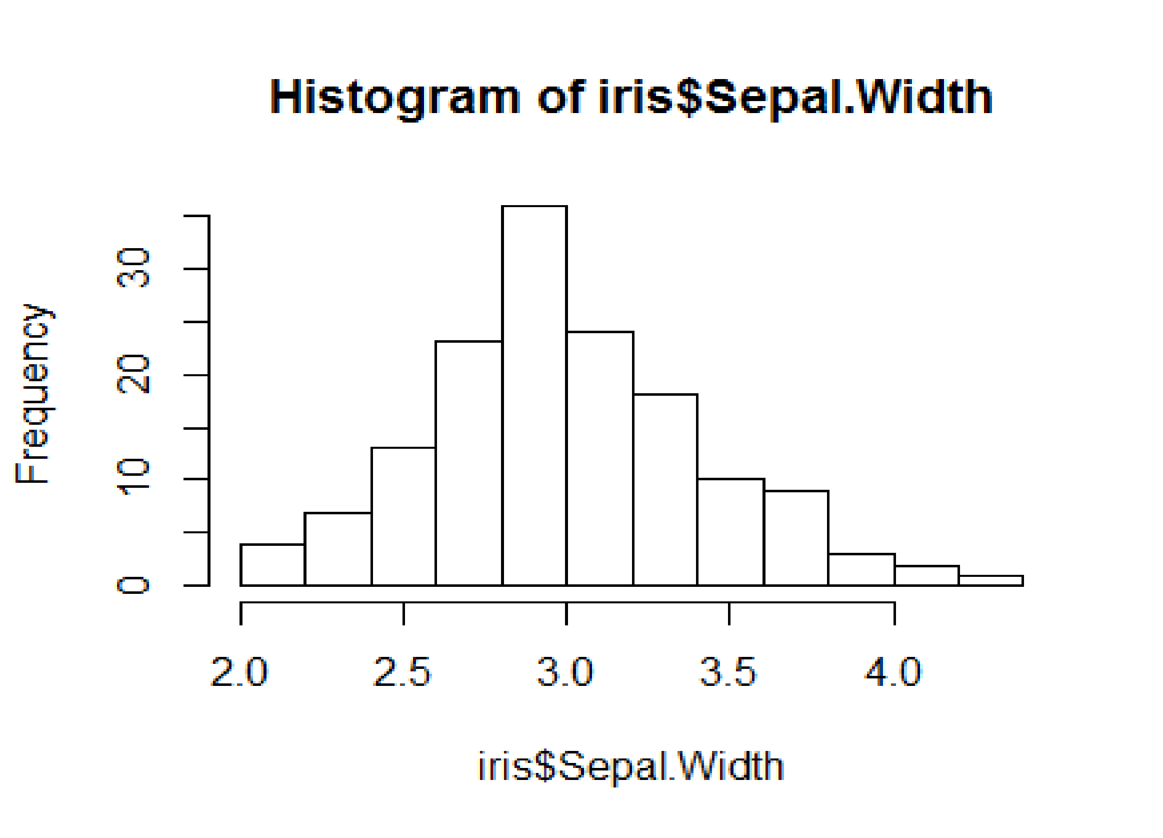
- hist(NormDist, breaks=20)

Iris Data set

- > str(iris) 'data.frame': 150 obs. of 5 variables: $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
- $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
- $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
- $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
- $ Species : Factor w/ 3 levels "setosa", "versicolor", ”verginica”

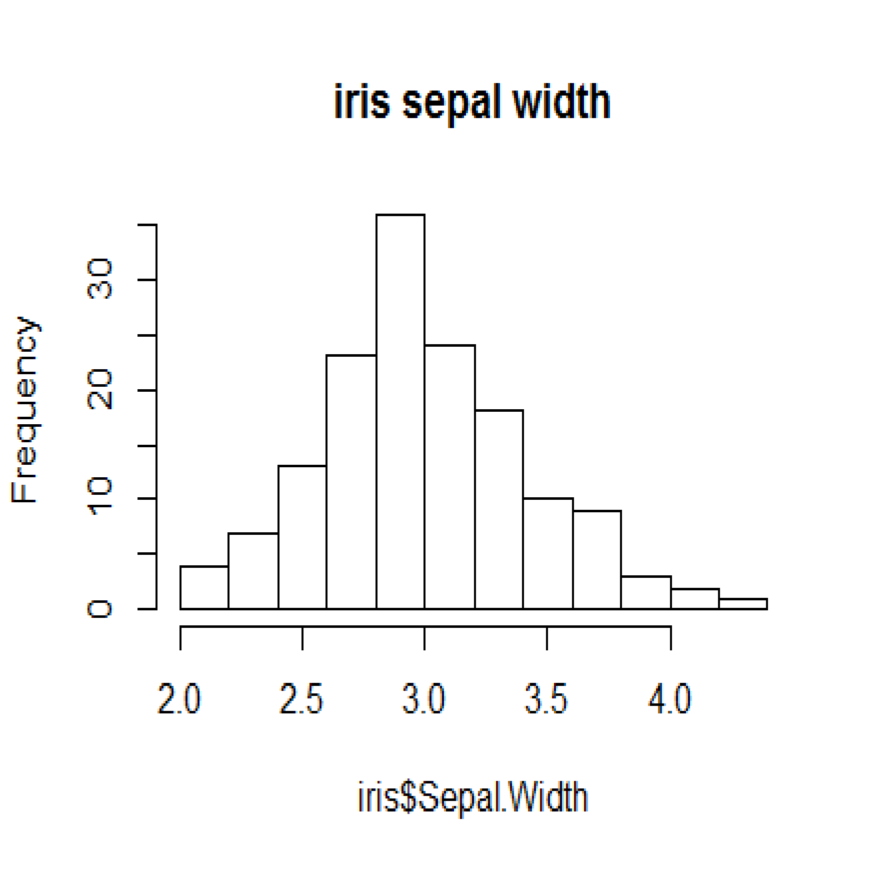
Histogram

- Histogram
  - `hist(iris$sepal.length)`



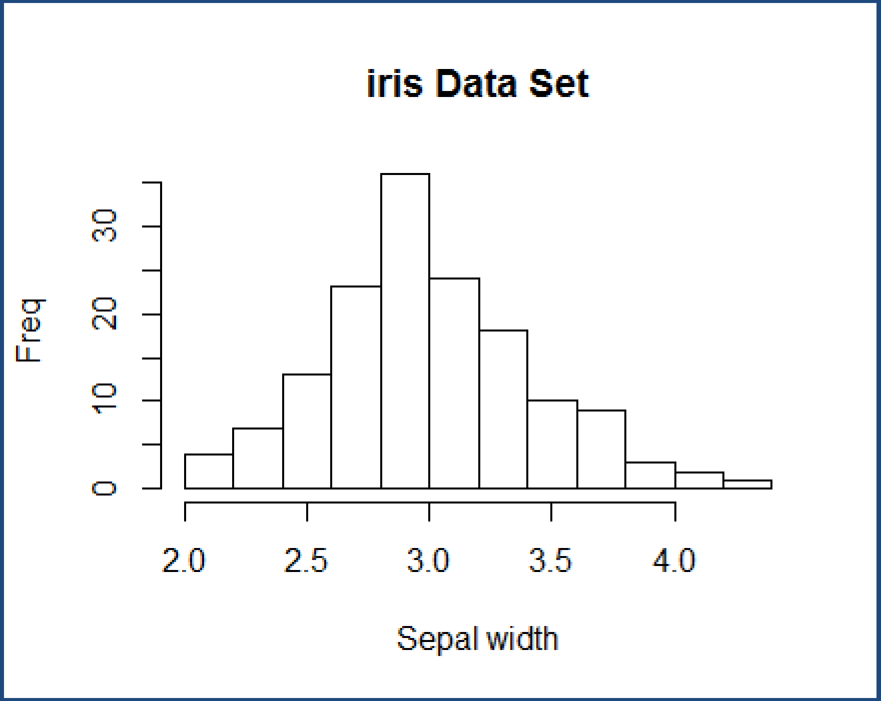
Histogram

- Add a title…
  - The “main” statement will give the plot an overall heading.
  - `hist(iris$sepal.width , main=‘iris: Sepal Width’)`



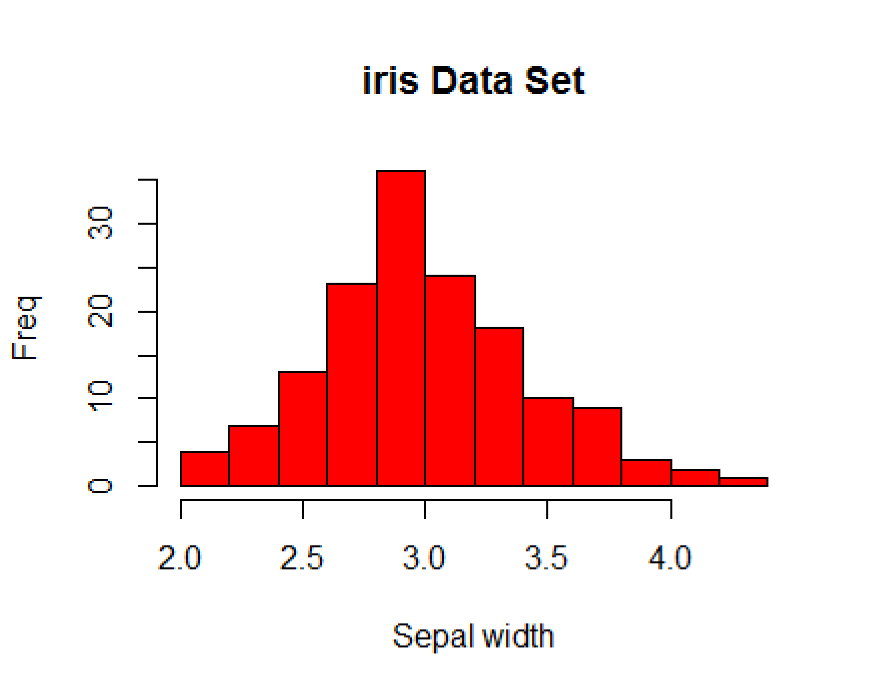
Histogram

- Adding axis labels…
- Use “xlab” and “ylab” to label the X and Y axes, respectively.
- **hist(iris$Sepal.Width, main="iris Data Set", xlab="Sepal width", ylab="Freq")**

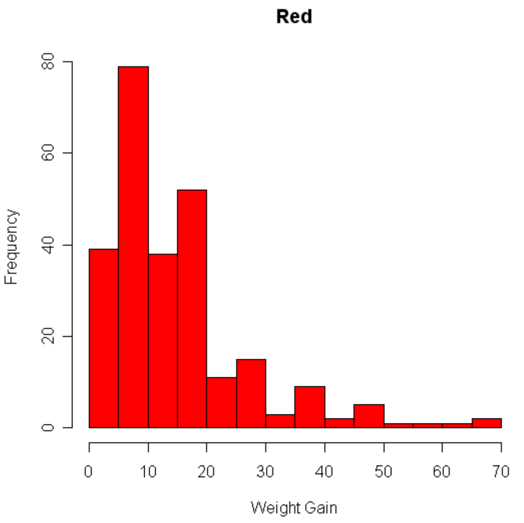


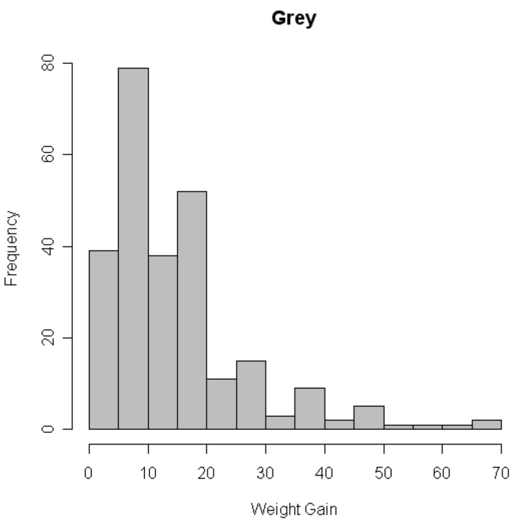
Histogram

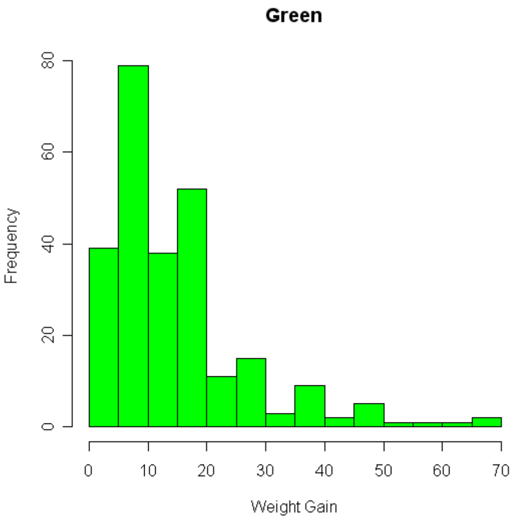
- Changing colors…
- Use the col statement.
  - ?colors will give you help on the colors.
  - Common colors may simply put in using the name.
  - **hist(iris$Sepal.Width, main="iris Data Set", xlab="Sepal width", ylab="Freq", col="red")**

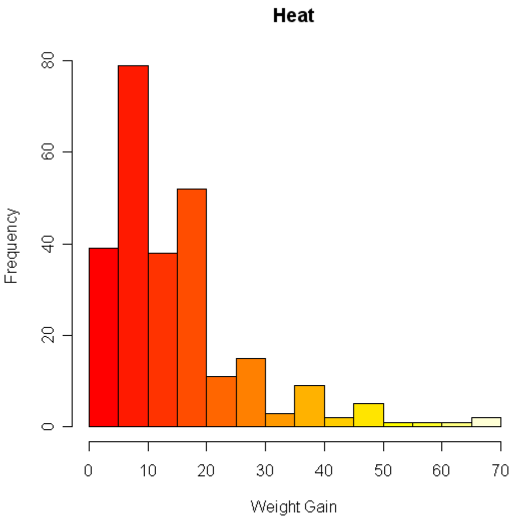


Basic Graphics – Colors





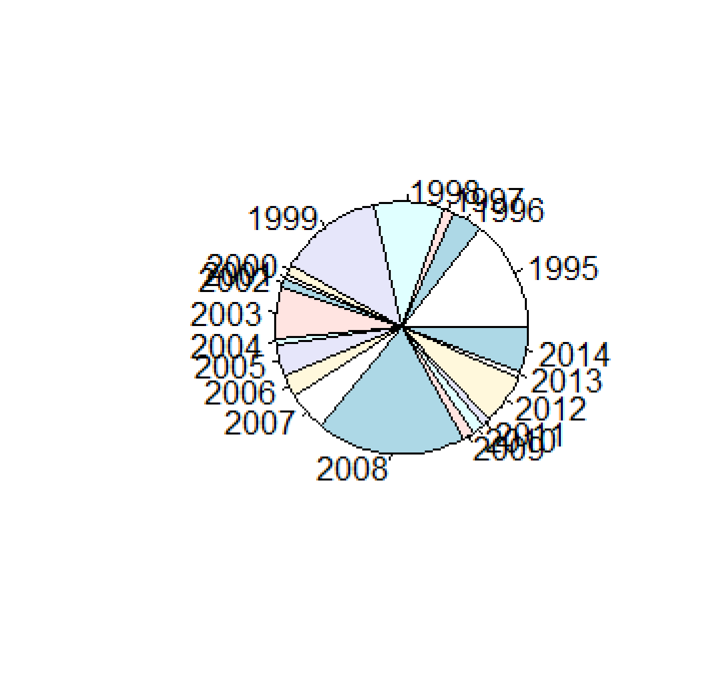


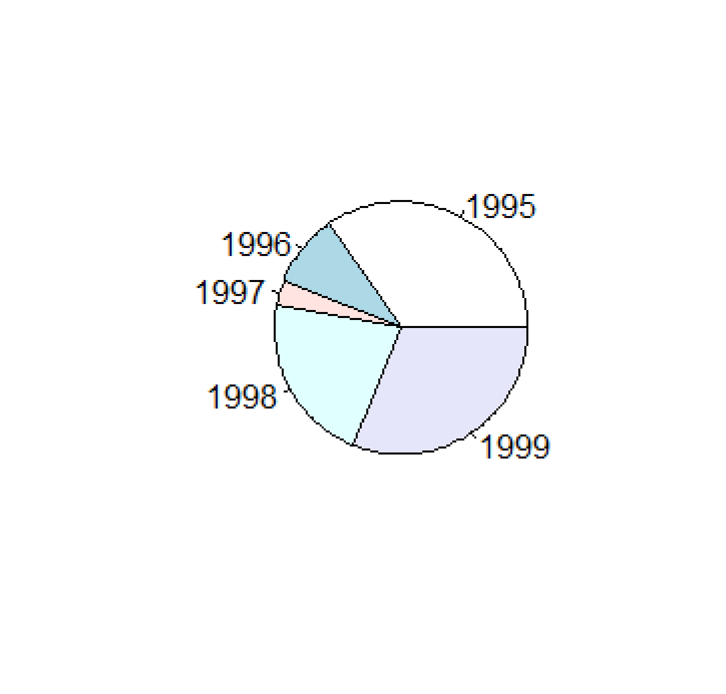


Pie Charts

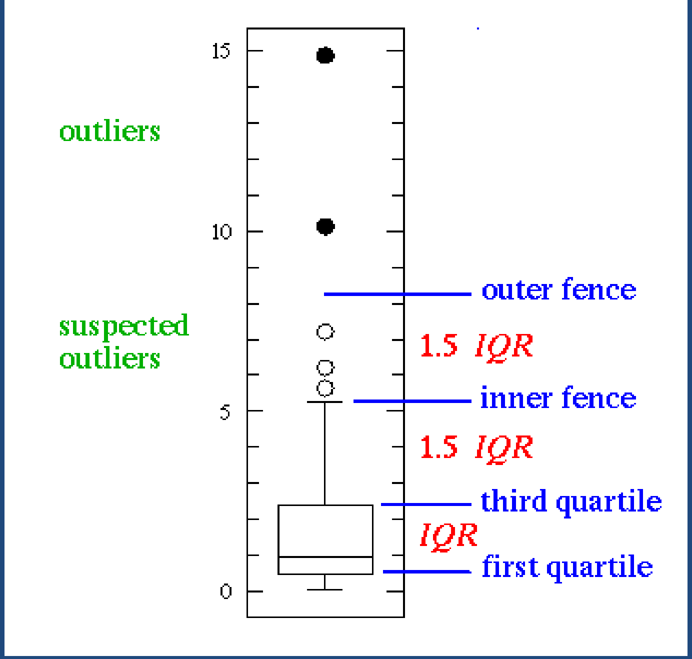
totalSales1<-sales$sales_total[1:20]
time<-1995:2014

- pie(totalSales1, labels=as.character(time))
- pie(totalSales1[1:5], labels=as.character(time[1:5]))

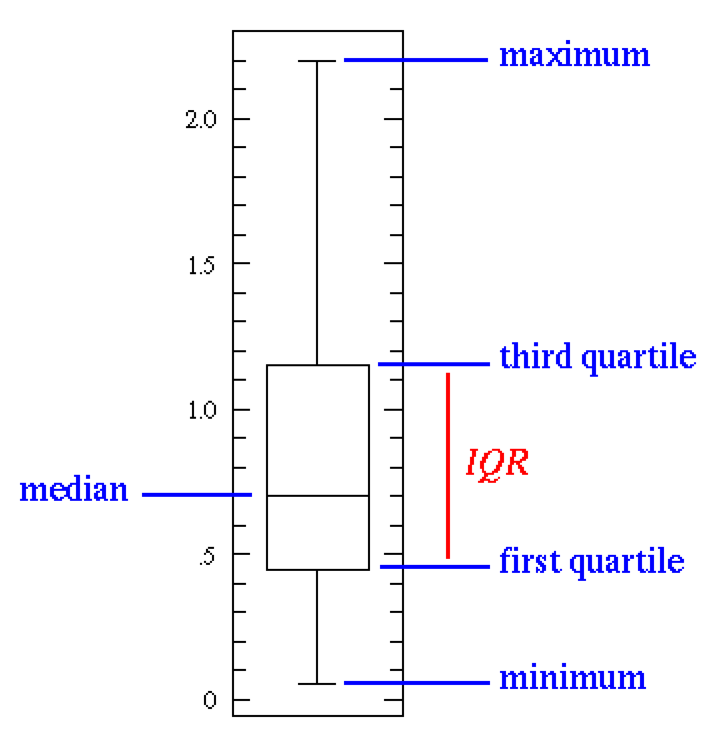


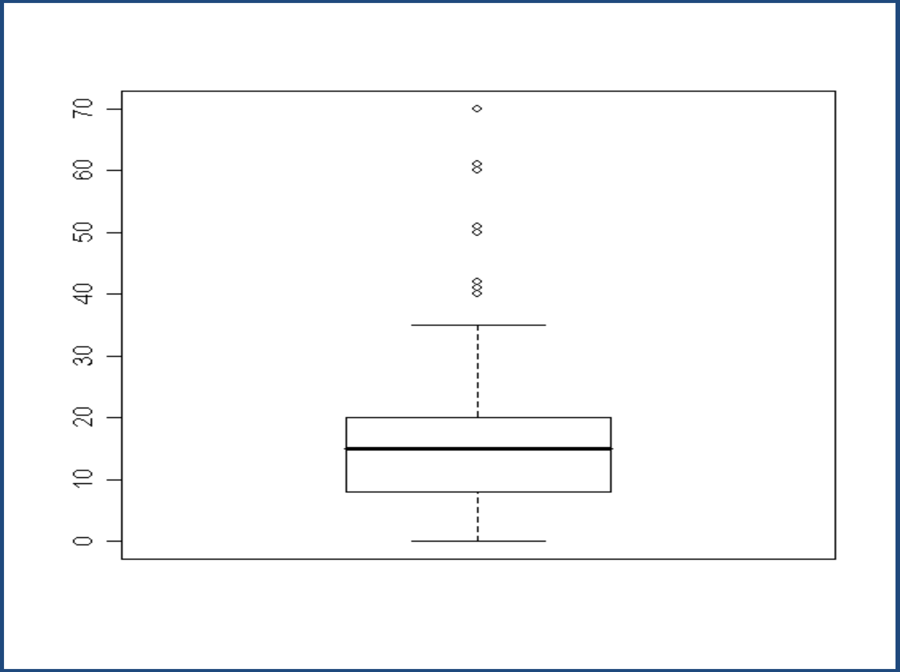


Box Plots



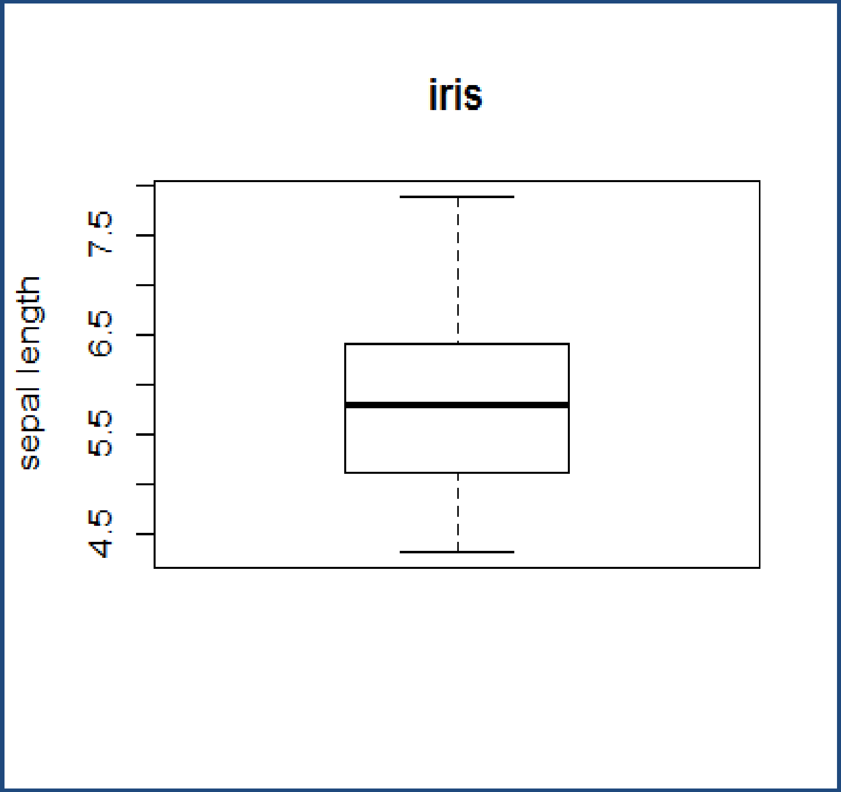
- boxplot(iris$Sepal.Length)





Boxplots

- Change it!
- **boxplot(iris$Sepal.Length, main="iris", ylab="sepal length")**

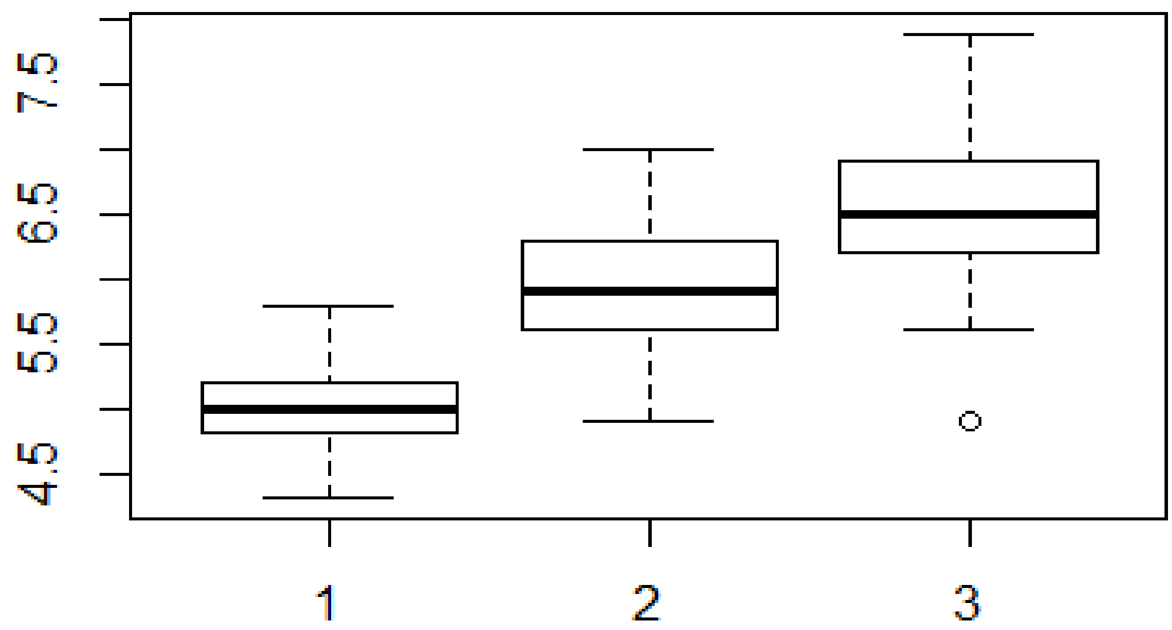


Box-Plots - Groupings

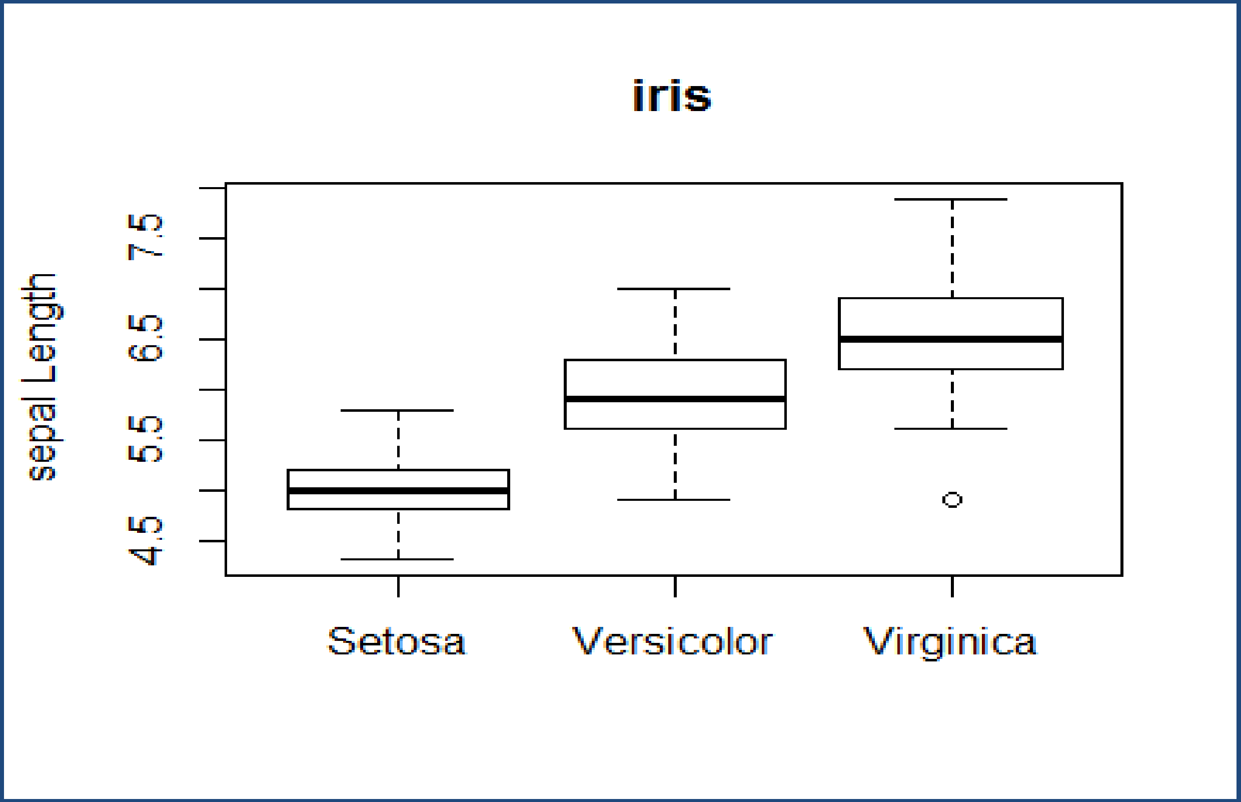
- What if we want several box plots side by side to be able to compare them.
- First Subset the Data into separate variables.
  - irisSetosa<-subset(iris, Species="setosa")
  - irisVersicolor<-subset(iris, Species="versicolor")
  - irisVirginica<-subset(iris, Species="virginica")
- Then Create the box plot.

  -
    ```
    boxplot(irisSetosa$Sepal.Length,
            irisVersicolor$Sepal.Length,
            irisVirginica$Sepal.Length)
    ```

Boxplots – Groupings



Boxplots - Groupings



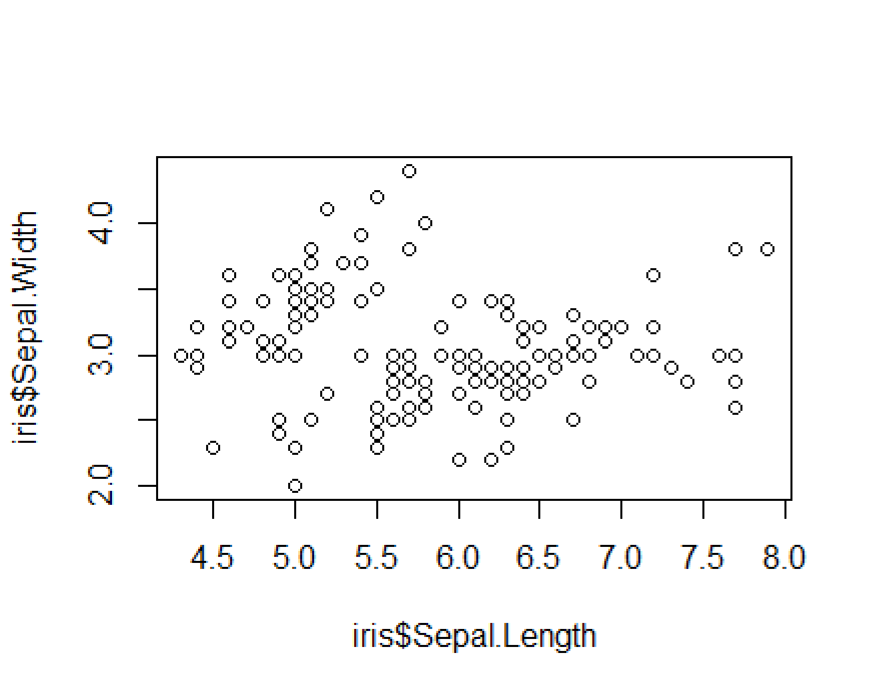
```
boxplot(irisSetosa$Sepal.Length,
irisVersicolor$Sepal.Length, irisVirginica$Sepal.Length,
main="iris", ylab="sepal Length", names=c("Setosa",
"Versicolor", "Virginica"))
```

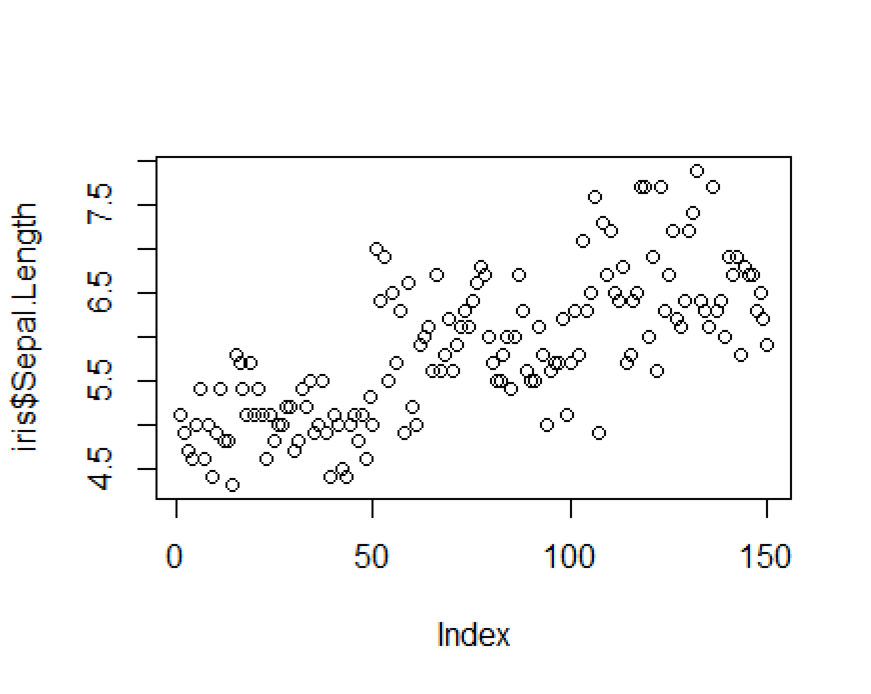
Using Plots for Bivariate

Scatter Plots

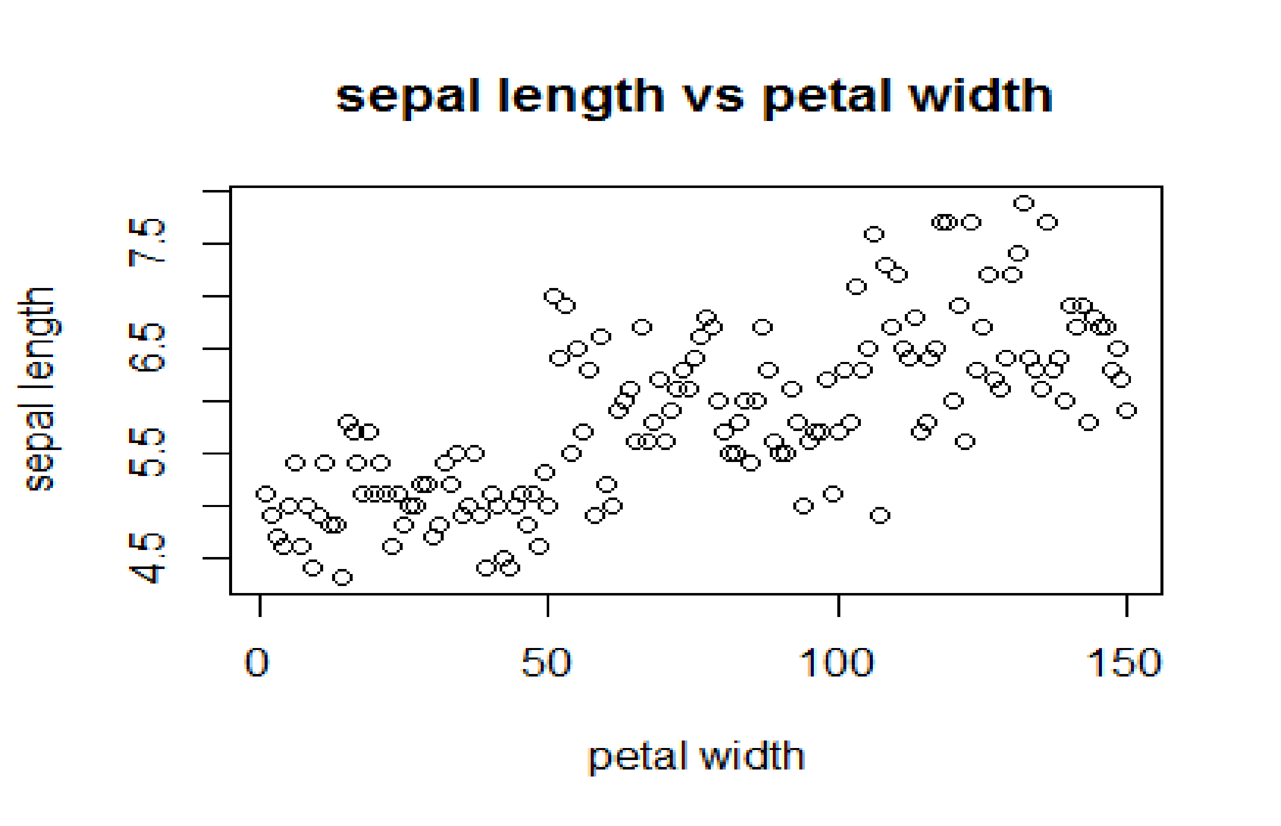
- Suppose we have two variables and we wish to see the relationship between them.
- A scatter plot works very well.
- R code:
  - `plot(x,y)`
- Example
  - plot(iris$Sepal.Length, iris$Sepal.Width)
  - plot(iris$Sepal.Length, iris$petal.Width)

Scatterplots





Scatterplots

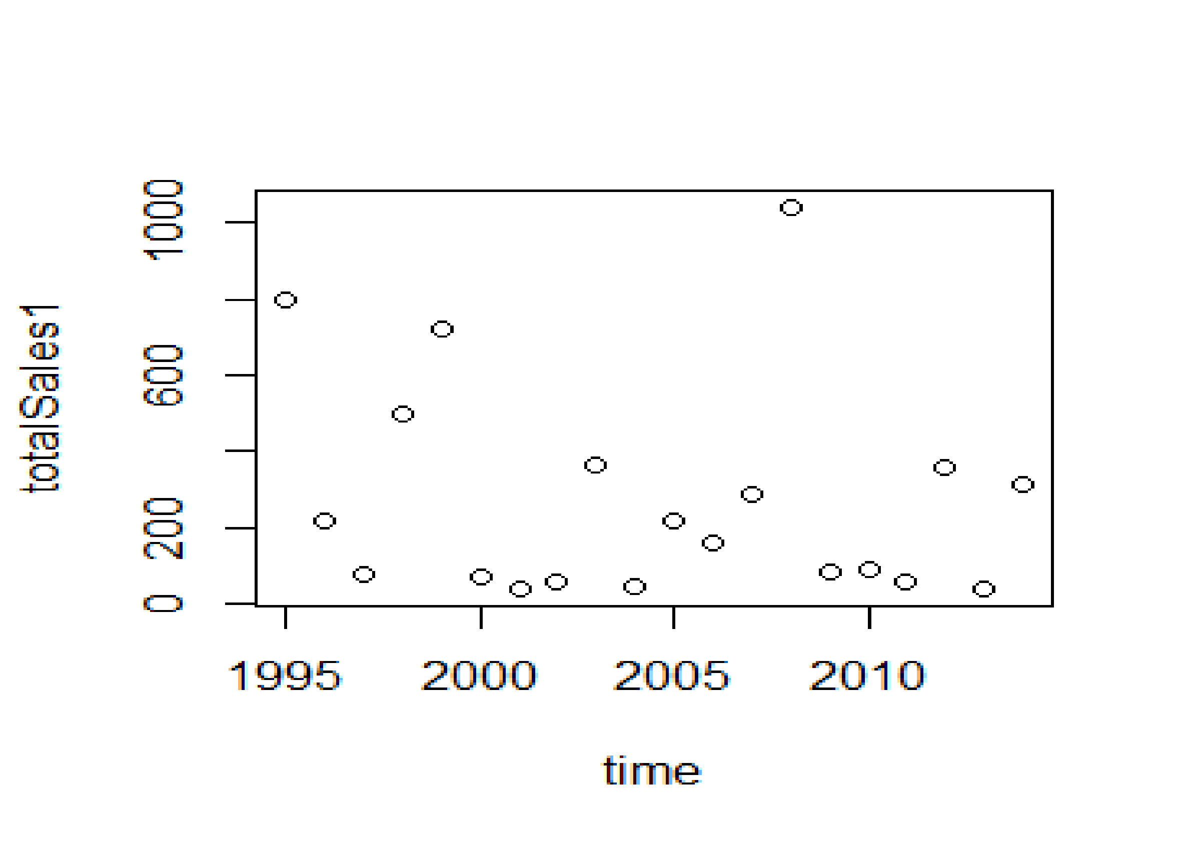


plot(iris$Sepal.Length, iris$petal.Width, main="sepal length vs petal width“, xlab="petal width", ylab= "sepal length")

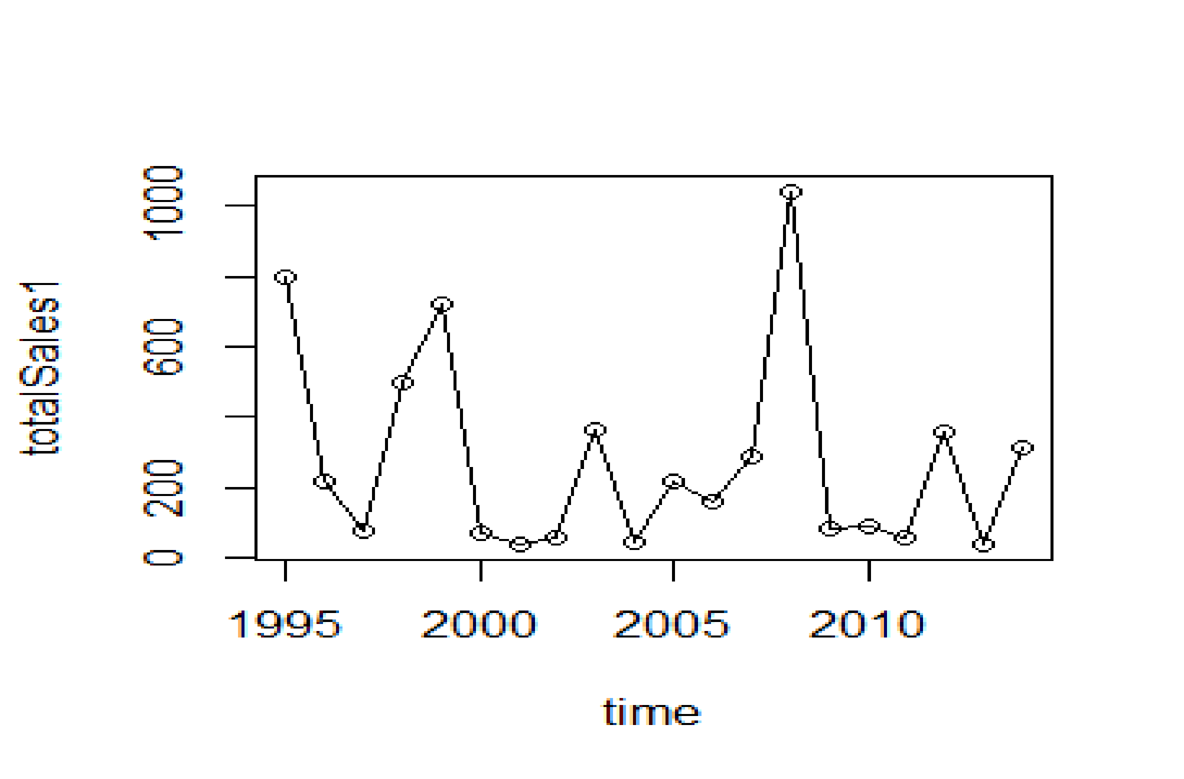
Line Plots

- Often data comes through time.
- Consider Dell stock
  - totalSales1<-sales$sales_total[1:20]
  - time<-1995:2014
  - plot(time,totalSales1)

Line Plots

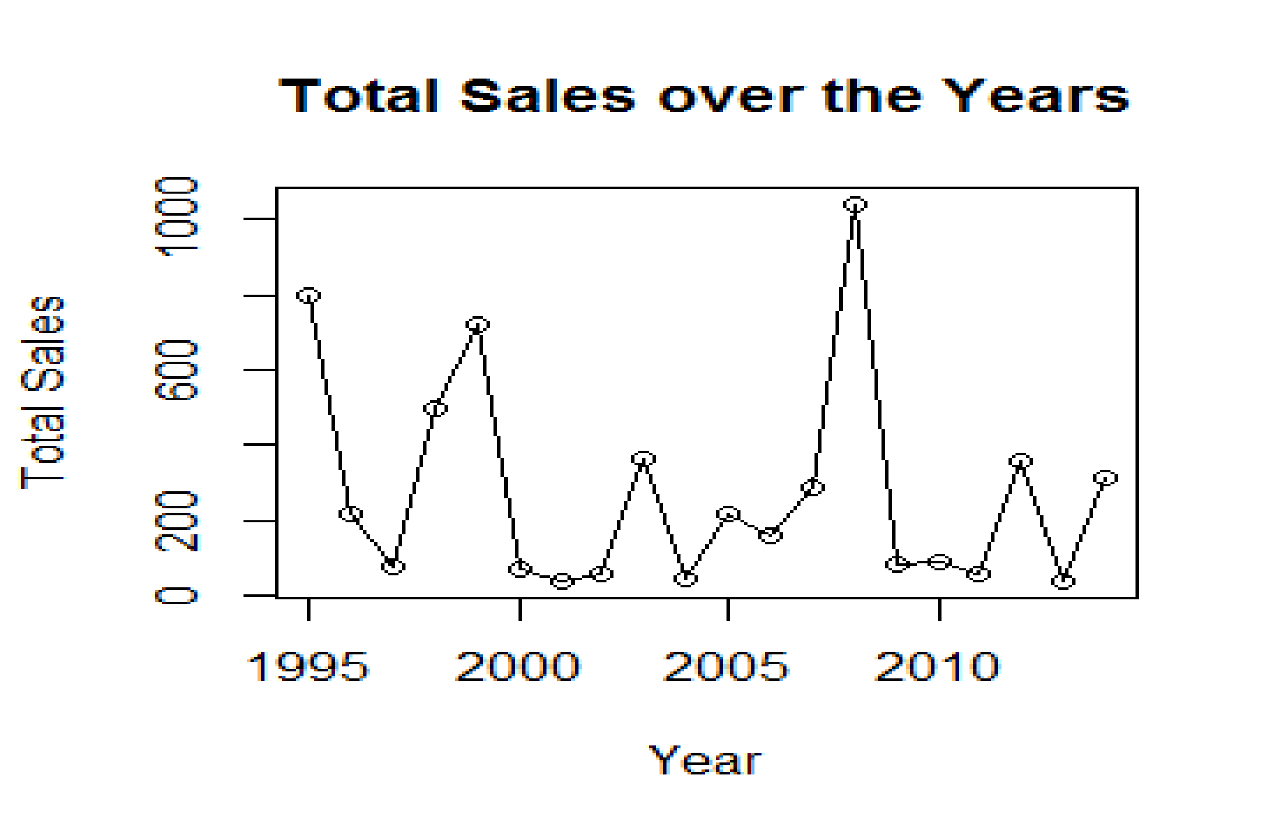


Line Plots



plot(time, totalSales1, type=“o")

Line Plots



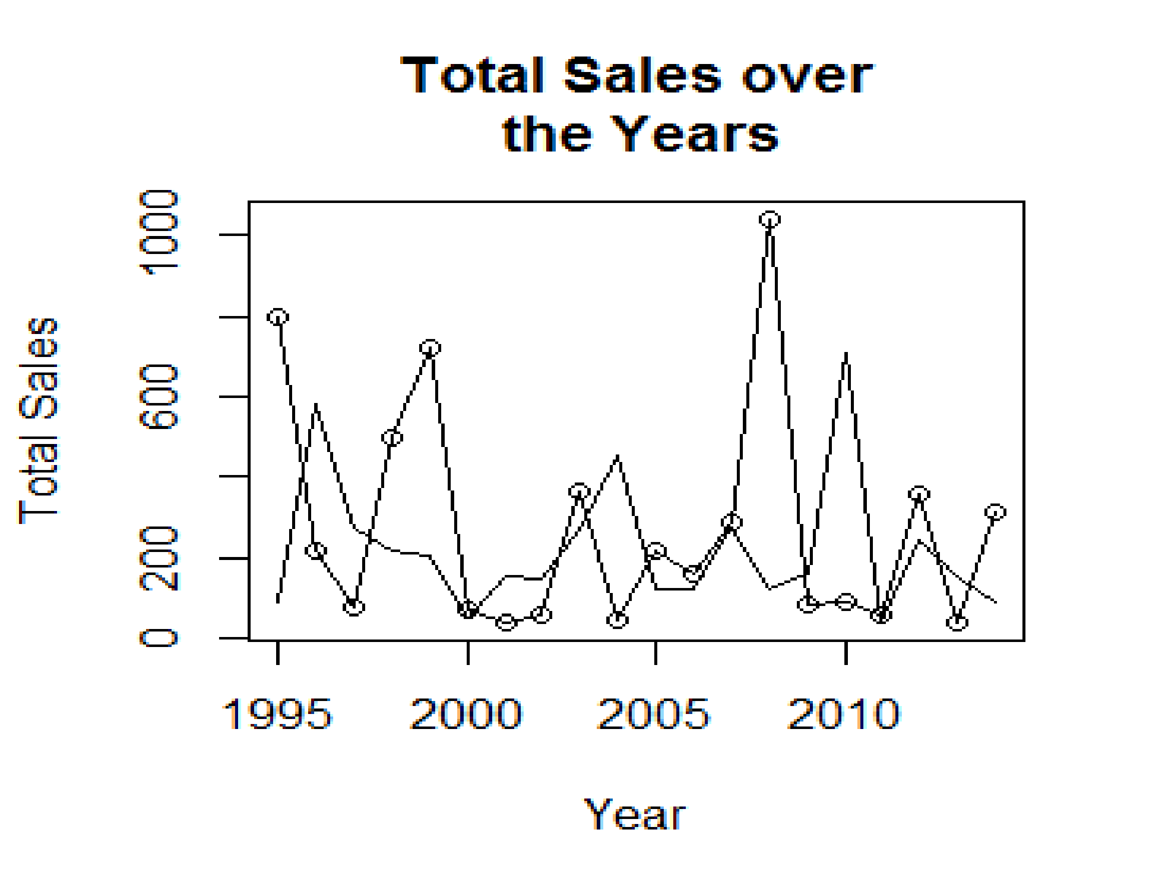
plot(time, totalSales1, type="o", main="Total Sales over the Years", xlab="Year", ylab="Total Sales")

Overlaying Plots

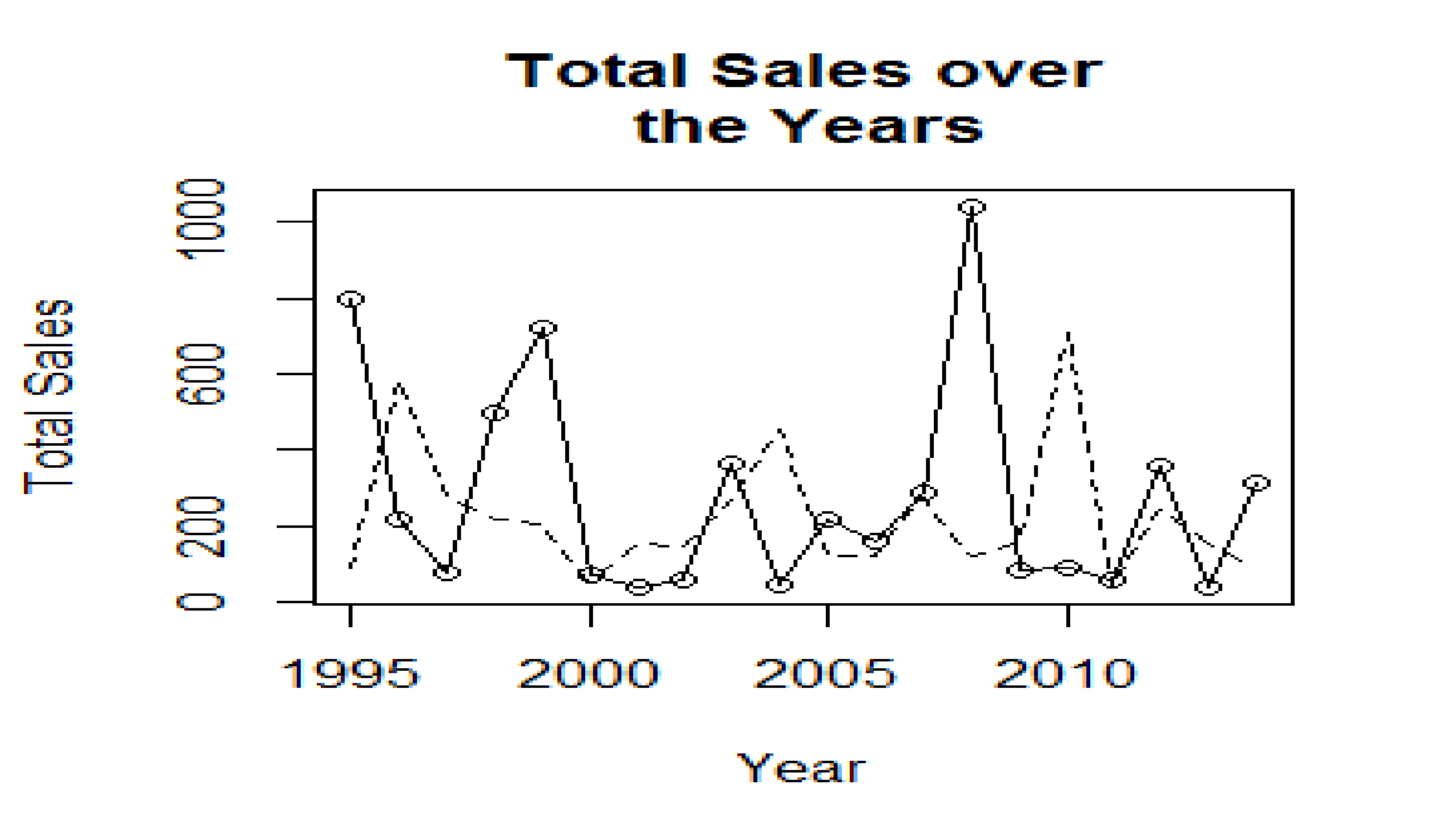
- Often we have more than one variable measured against the same predictor (X).

```
plot(time, totalSales1, type="o", main="Total Sales over the Years", xlab="Year", ylab="Total Sales")
lines(time, totalSales2)
```

Overlaying Graphs



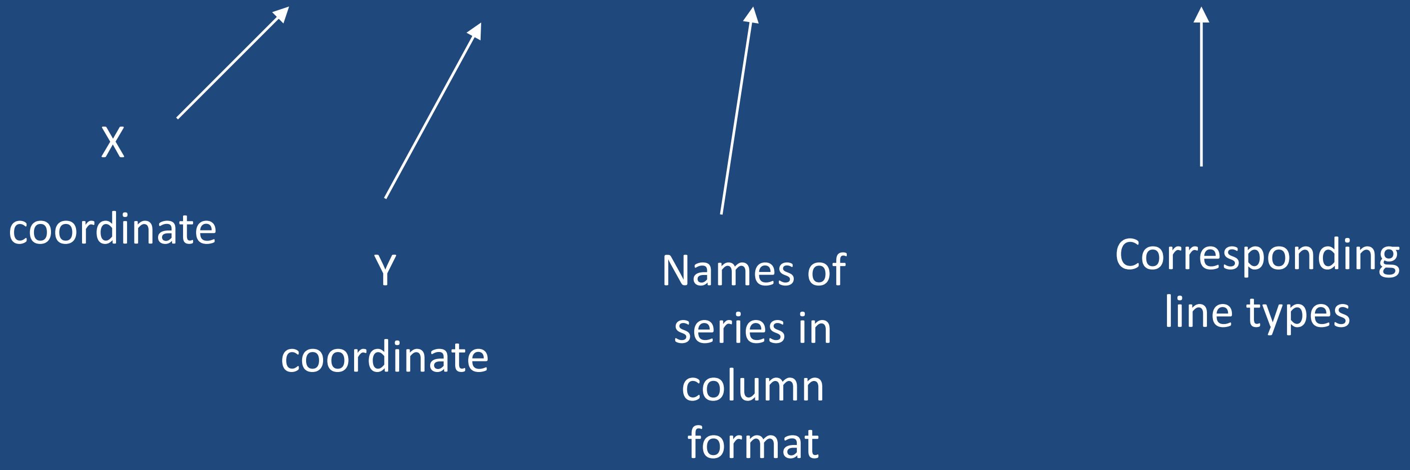
Overlaying Graphs



```
lines(time, totalSales2, lty=2)
```

Adding a Legend

- Adding a legend is a bit tricky in R.
- Syntax
- `legend(    x,    y,    names,    line types)`



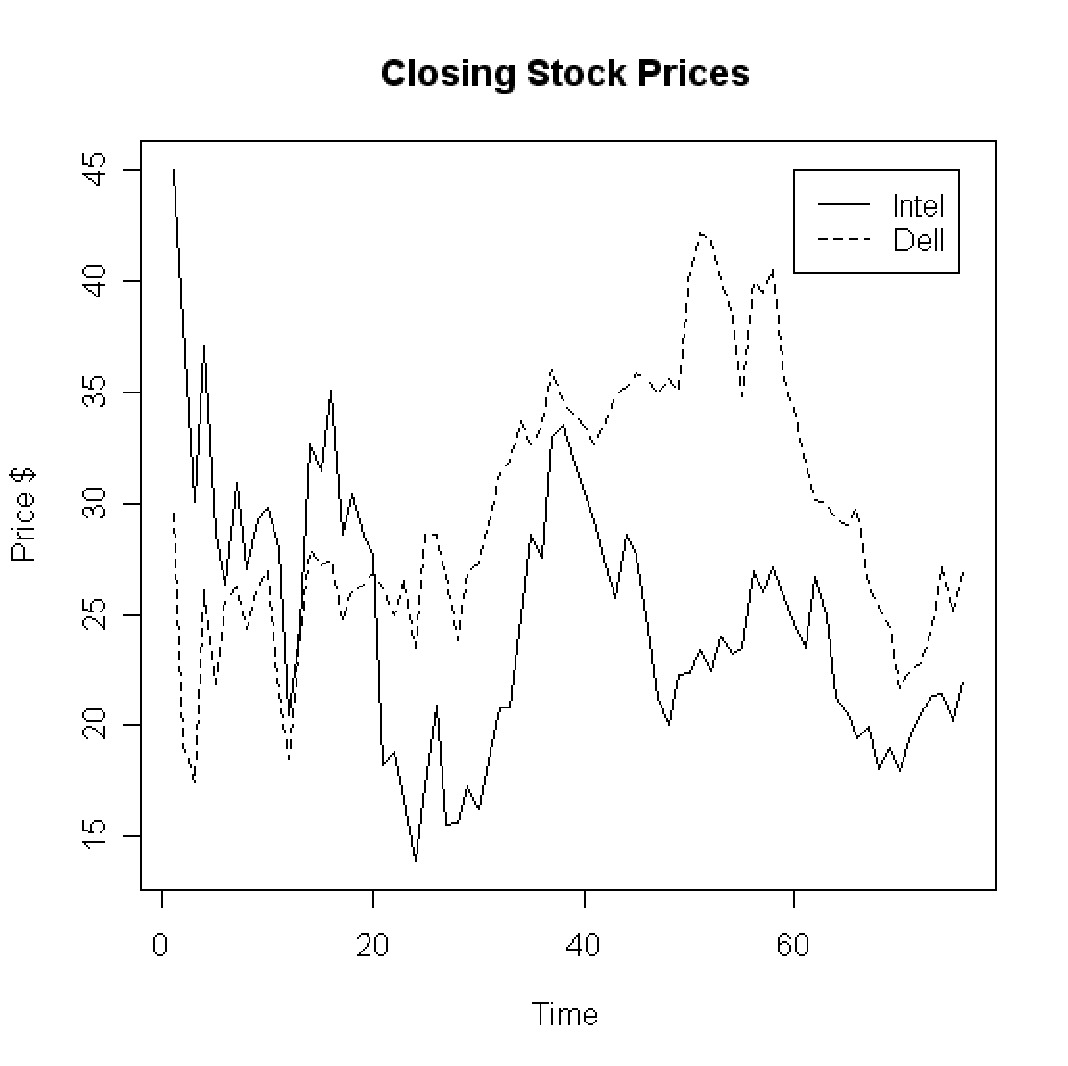
X coordinate

Y coordinate

Names of series in column format

Corresponding line types

Adding a Legend



```
legend(60,45,c('Item1','Item2'),lty=c(1,2))
```

Paneling Graphics

- Suppose we want more than one graphic on a panel.
- We can partition the graphics panel to give us a framework in which to panel our plots.
- `par(mfrow = c( nrow, ncol))`

Number of rows

Number of columns

Paneling Graphics

- Consider the following:

-
  ```
  par(mfrow=c(2,2))
  ```

-
  ```
  hist(iris$Sepal.Width, main="iris Data Set",
  xlab="Sepal width", ylab="Freq", col="red")
  ```

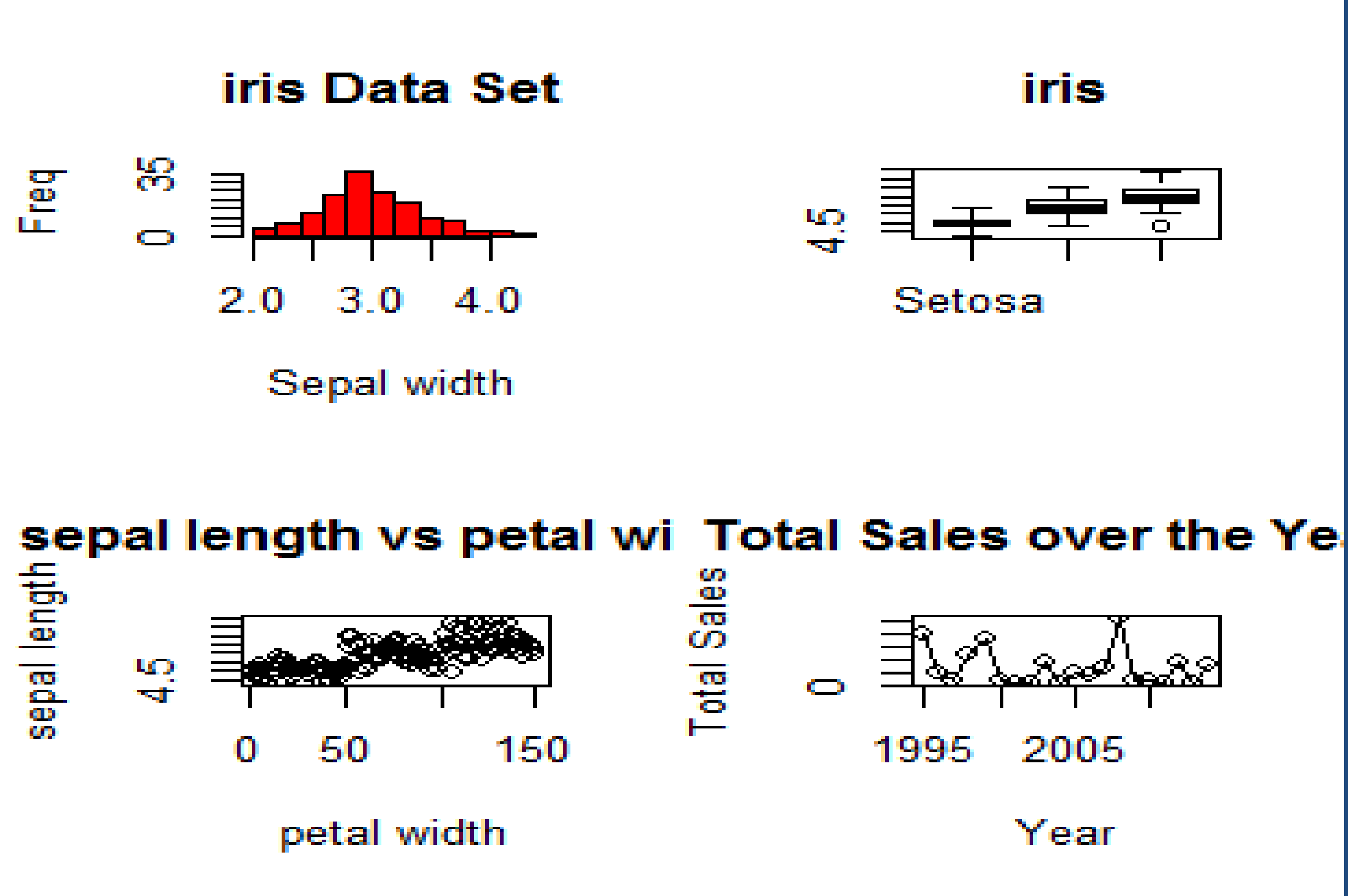
-
  ```
  boxplot(irisSetosa$Sepal.Length,
  irisVersicolor$Sepal.Length,
  irisVirginica$Sepal.Length, main="iris", ylib="sepal
  Length", names=c("Setosa", "Versicolor",
  "Virginica"))
  ```

-
  ```
  plot(iris$Sepal.Length, iris$petal.Width, main="sepal
  length vs petal width", xlab="petal width", ylab=
  "sepal length")
  ```

-
  ```
  plot(time1, totalSales1, type="o", main="Total Sales
  over the Years", xlab="Year", ylab="Total Sales")
  ```

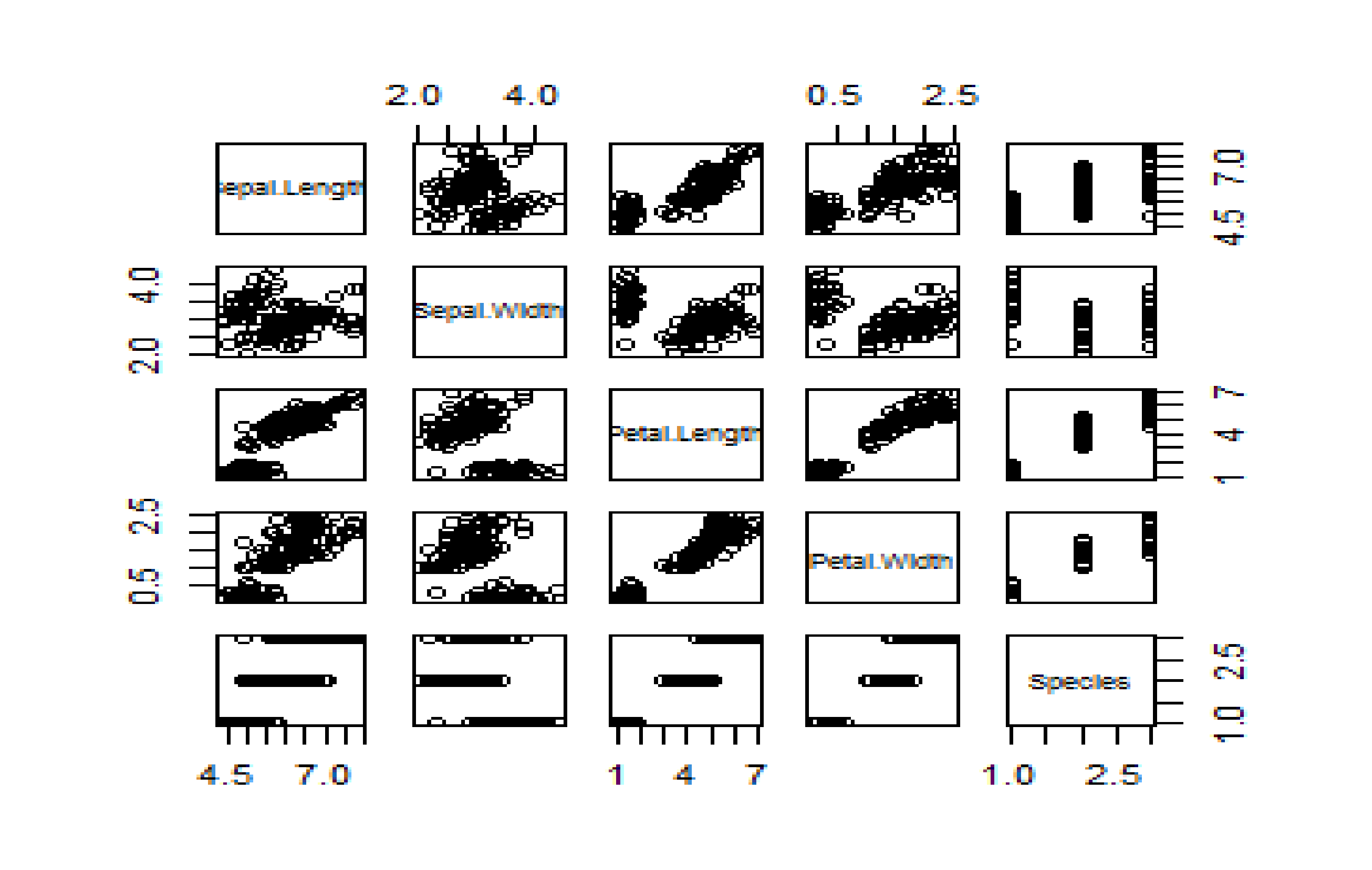
-
  ```
  lines(time, totalSales2, lty=2)
  ```

Paneling Graphics



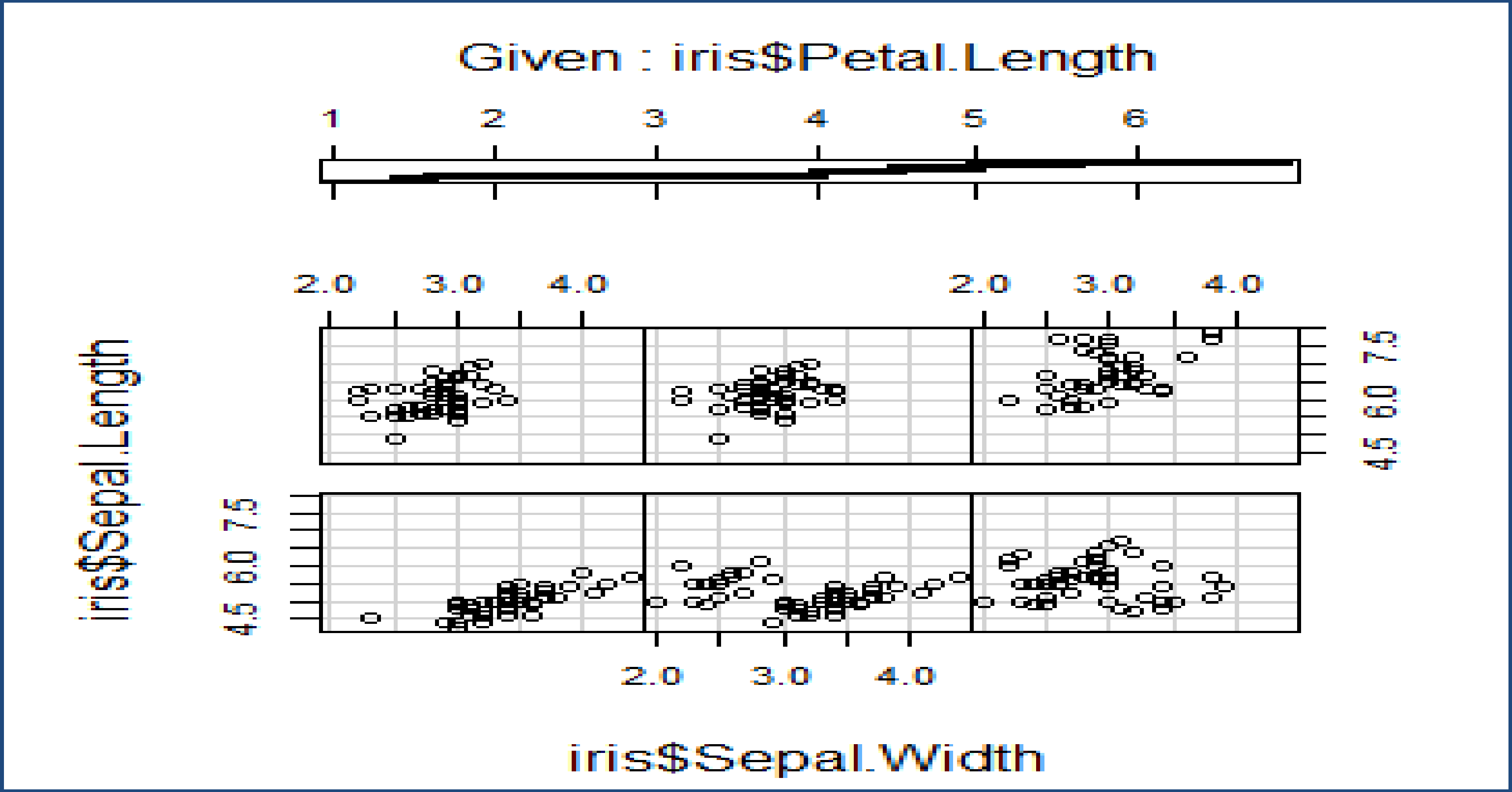
Plots for multiple variables

- pairs(iris)



Plots for multiple variables

- coplot(iris$Sepal.Length~iris$Sepal.Width|iris $Petal.Length)


Given : iris$Petal.Length
iris$Sepal.Length
iris$Sepal.Width