nkc2xtcz2

December 5, 2024

0.0.1 Hypothesis Test

[2]:
```
x=rnorm(100)
t.test(x,mu=5)
```

```
        One Sample t-test

data:  x
t = -54.004, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 -0.34924643  0.02990576
sample estimates:
 mean of x
-0.1596703
```

[3]:
```
x1=rnorm(100)
y1=rnorm(100)
t.test(x1,y1)
```

```
        Welch Two Sample t-test

data:  x1 and y1
t = -0.26366, df = 197.93, p-value = 0.7923
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2933917  0.2241911
sample estimates:
 mean of x  mean of y
0.04516715 0.07976747
```

[4]:
```
x2=rnorm(100)
t.test(x2,mu=2,alternative = 'greater')
```

```
        One Sample t-test

data:  x2
t = -22.032, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 2
95 percent confidence interval:
 -0.2368293        Inf
sample estimates:
  mean of x
-0.08006741
```

[6]:
```
x3=rnorm(100)
wilcox.test(x3,exact=FALSE)
```

```
        Wilcoxon signed rank test with continuity correction

data:  x3
V = 2875, p-value = 0.2295
alternative hypothesis: true location is not equal to 0
```

[8]:
```
cor.test(mtcars$mpg,mtcars$hp)
```

```
        Pearson's product-moment correlation

data:  mtcars$mpg and mtcars$hp
t = -6.7424, df = 30, p-value = 1.788e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8852686 -0.5860994
sample estimates:
       cor
-0.7761684
```

0.0.2 K Means clustering

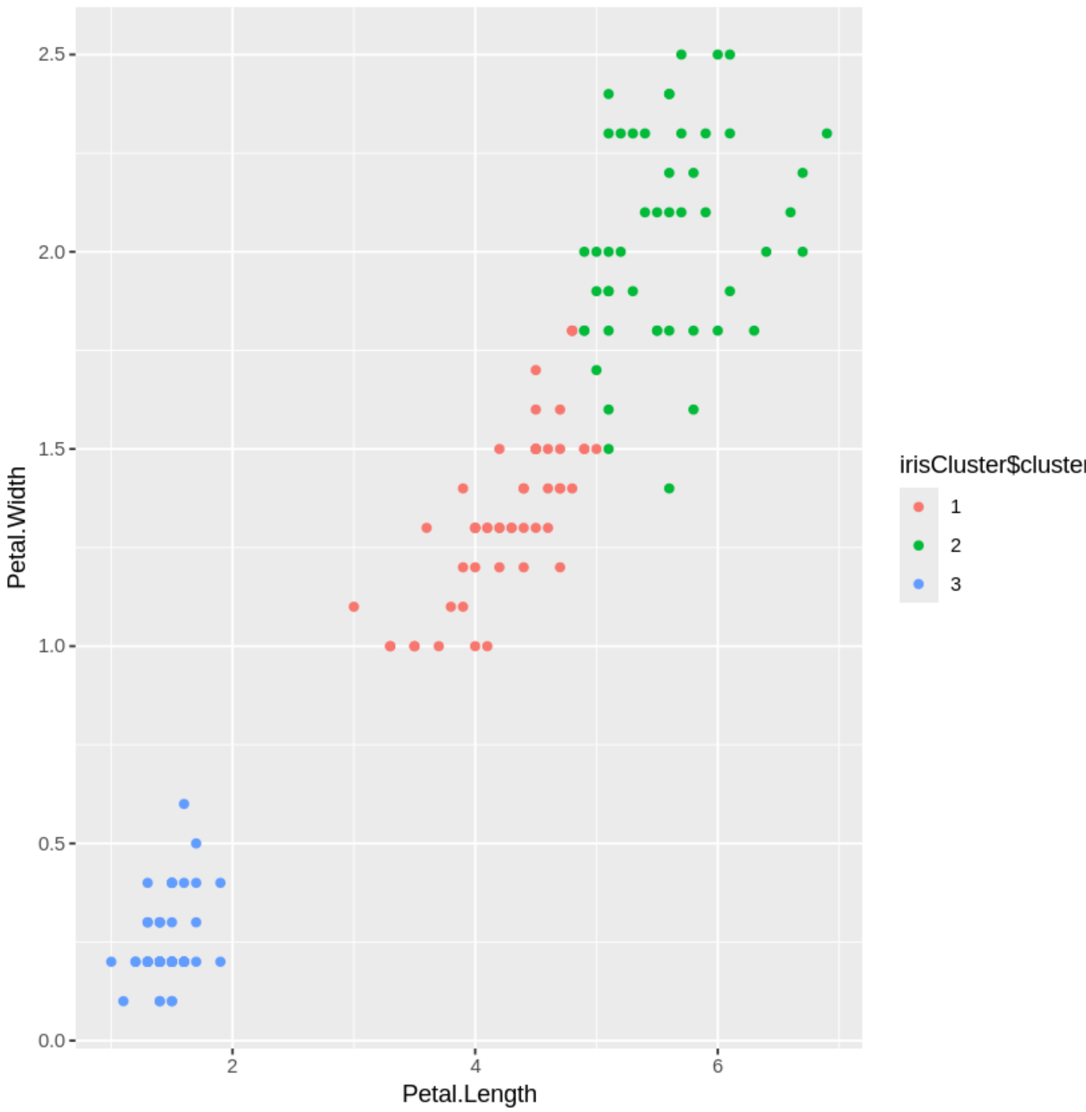
[9]:
```
library(cluster)
library(ggplot2)
```

[10]:
```
set.seed(20)
irisCluster<-kmeans(iris[,3:4],3,nstart=20)
irisCluster
```

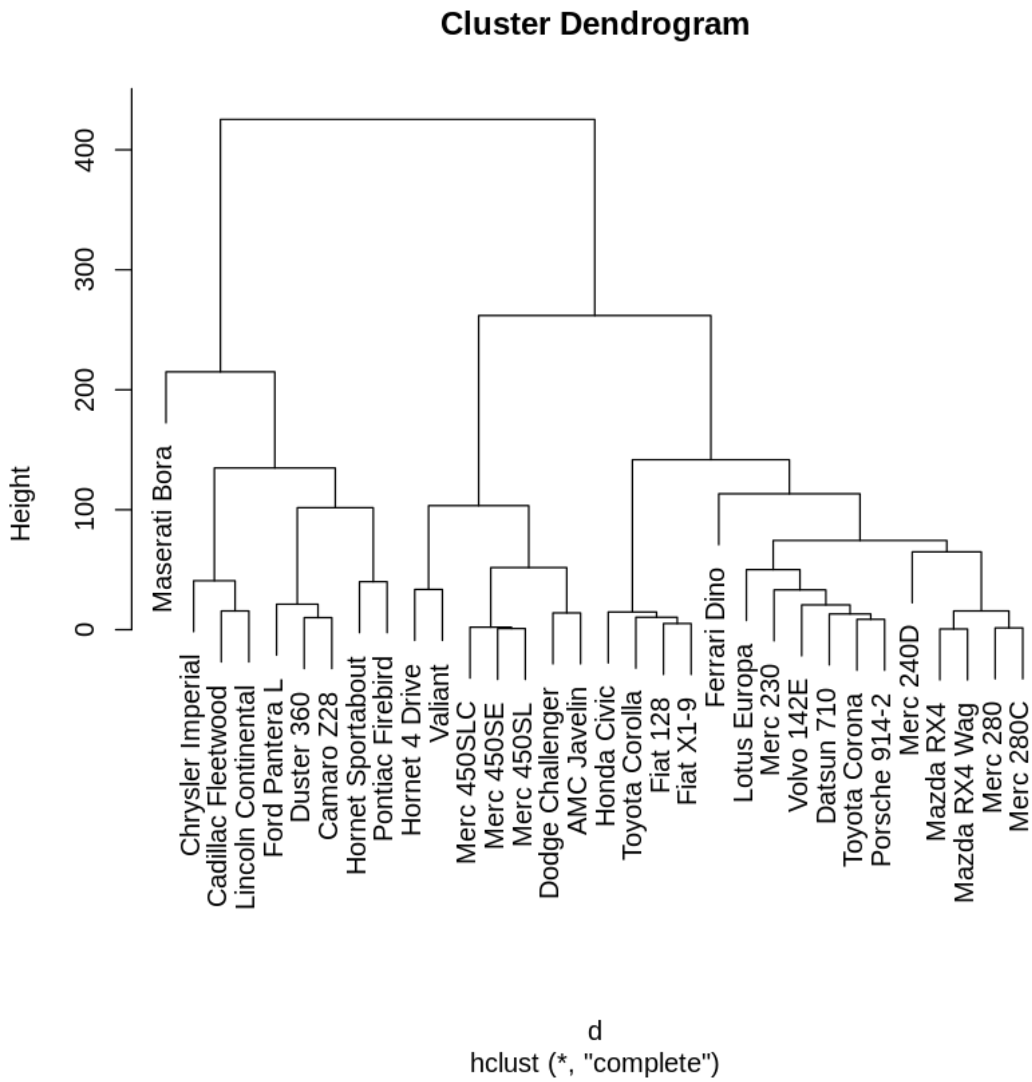
```
K-means clustering with 3 clusters of sizes 52, 48, 50
```

```
Cluster means:
  Petal.Length Petal.Width
1     4.269231    1.342308
2     5.595833    2.037500
3     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [75] 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2
[112] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
[149] 2 2

Within cluster sum of squares by cluster:
[1] 13.05769 16.29167  2.02200
 (between_SS / total_SS =  94.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

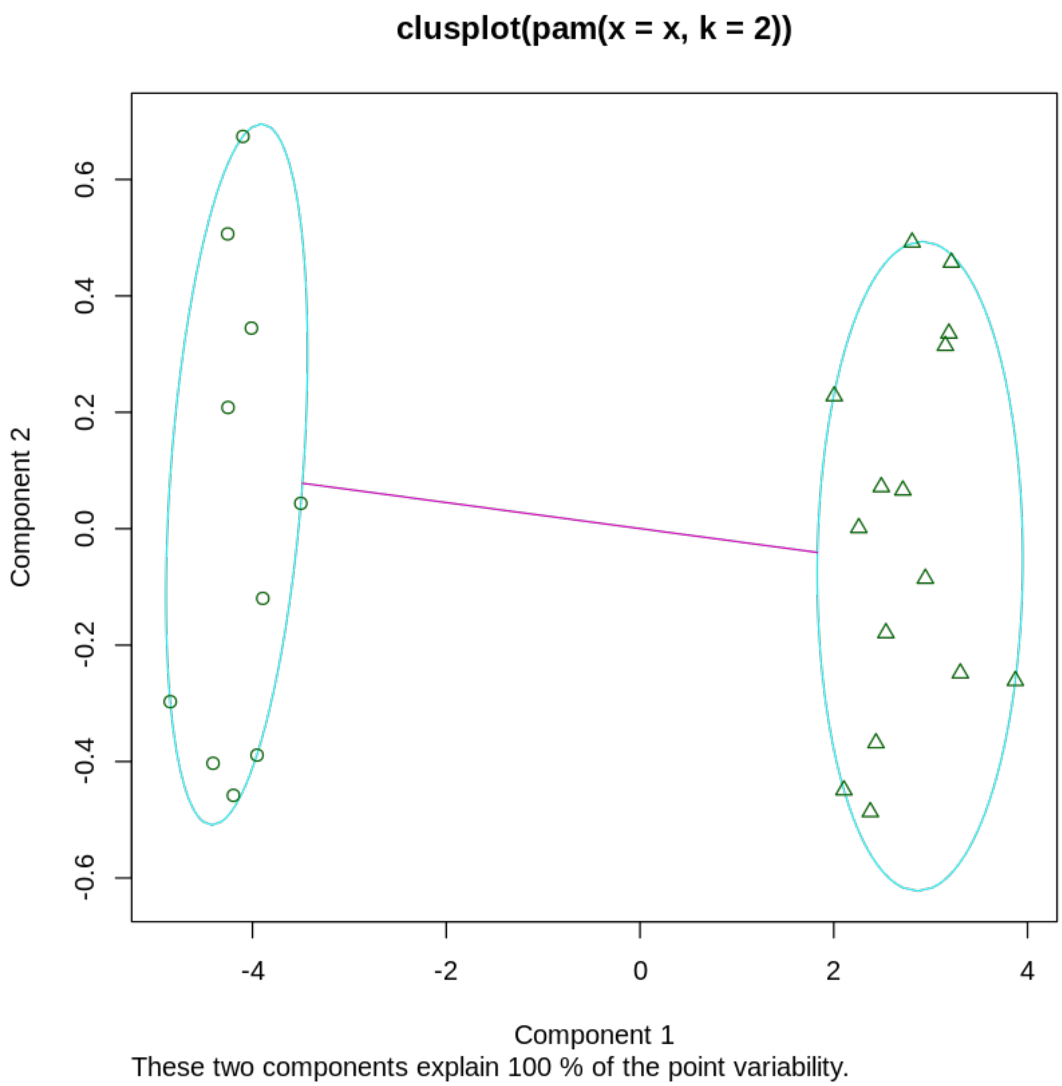
[11]:
```
irisCluster$cluster<-as.factor(irisCluster$cluster)
ggplot(iris,aes(Petal.Length,Petal.
 ↪Width,color=irisCluster$cluster))+geom_point()
```



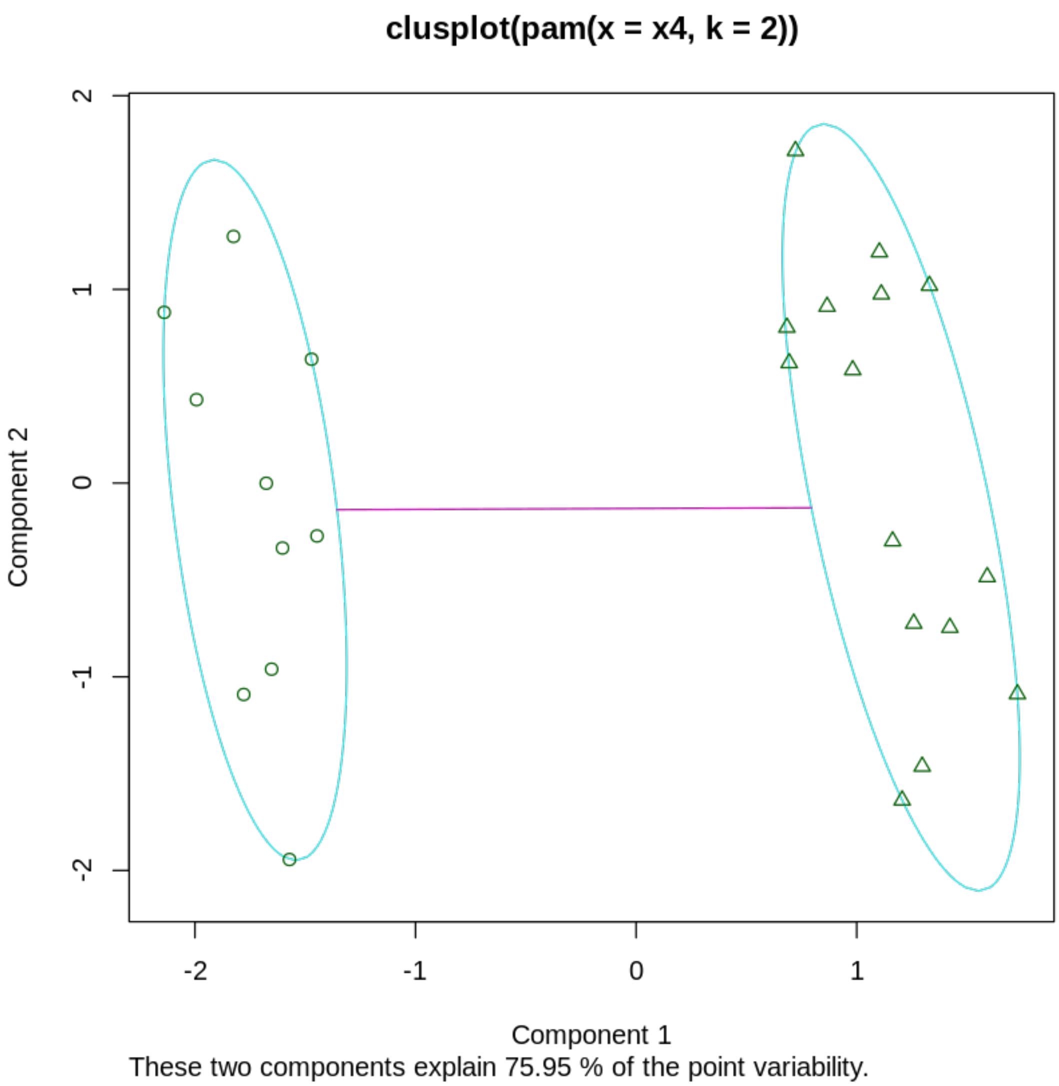
[12]:
```
d<-dist(as.matrix(mtcars))
hc<-hclust(d)
plot(hc)
```



[14]:
```
x<-rbind(cbind(rnorm(10,0,0.5),rnorm(10,0,0.5)),cbind(rnorm(15,5,0.
↪5),rnorm(15,5,0.5)))
clusplot(pam(x,2))
```



[15]:
```
x4<-cbind(x,rnorm(25),rnorm(25))
clusplot(pam(x4,2))
```



0.0.3 Linear and Logistic Regression

[16]:
```
dataset=read.csv("/content/data-marketing-budget-12mo.
↪csv",header=T,colClasses=c("numeric","numeric","numeric"))
head(dataset)
```

A data.frame: 6 × 3

	Month <dbl>	Spend <dbl>	Sales <dbl>
1	1	1000	9914
2	2	4000	40487
3	3	5000	54324
4	4	4500	50044
5	5	3000	34719
6	6	4000	42551

[17]:
```
simple.fit = lm(Sales~Spend,data=dataset)
summary(simple.fit)
```

```
Call:
lm(formula = Sales ~ Spend, data = dataset)

Residuals:
   Min     1Q Median     3Q    Max
 -3385  -2097    258   1726   3034

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1383.4714  1255.2404   1.102    0.296
Spend         10.6222     0.1625  65.378 1.71e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2313 on 10 degrees of freedom
Multiple R-squared:  0.9977,	Adjusted R-squared:  0.9974
F-statistic:  4274 on 1 and 10 DF,  p-value: 1.707e-14
```

[18]:
```
multi.fit = lm(Sales~Spend+Month, data=dataset)
summary(multi.fit)
```

```
Call:
lm(formula = Sales ~ Spend + Month, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-1793.73 -1558.33    -1.73  1374.19  1911.58

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -567.6098  1041.8836  -0.545  0.59913
Spend         10.3825     0.1328  78.159 4.65e-14 ***
Month        541.3736   158.1660   3.423  0.00759 **
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1607 on 9 degrees of freedom
Multiple R-squared:  0.999,	Adjusted R-squared:  0.9988
F-statistic:  4433 on 2 and 9 DF,  p-value: 3.368e-14
```

[19]:
```
input<- mtcars [,c("am","cyl","hp","wt")]
print(head(input))
```

```
                  am cyl  hp    wt
Mazda RX4          1   6 110 2.620
Mazda RX4 Wag      1   6 110 2.875
Datsun 710         1   4  93 2.320
Hornet 4 Drive     0   6 110 3.215
Hornet Sportabout  0   8 175 3.440
Valiant            0   6 105 3.460
```

[20]:
```
input<- mtcars [,c("am","cyl","hp","wt")]
am.data =glm(formula = am~ cyl+hp+wt,data = input,family = binomial)
print(summary(am.data))
```

```
Call:
glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 19.70288    8.11637   2.428   0.0152 *
cyl          0.48760    1.07162   0.455   0.6491
hp           0.03259    0.01886   1.728   0.0840 .
wt          -9.14947    4.15332  -2.203   0.0276 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.2297  on 31  degrees of freedom
Residual deviance:  9.8415  on 28  degrees of freedom
AIC: 17.841

Number of Fisher Scoring iterations: 8
```

0.0.4 Time Series Analysis

[21]:
```
kings <- scan("http://robjhyndman.com/tsdldata/misc/kings.dat",skip=3)
kings
```

1. 60 2. 43 3. 67 4. 50 5. 56 6. 42 7. 50 8. 65 9. 68 10. 43 11. 65 12. 34 13. 47 14. 34 15. 49 16. 41 17. 13 18. 35 19. 53 20. 56 21. 16 22. 43 23. 69 24. 59 25. 48 26. 59 27. 86 28. 55 29. 68 30. 51 31. 33 32. 49 33. 67 34. 77 35. 81 36. 67 37. 71 38. 81 39. 68 40. 70 41. 77 42. 56

[22]:
```
kingstimeseries <- ts(kings)
kingstimeseries
```

A Time Series:
1. 60 2. 43 3. 67 4. 50 5. 56 6. 42 7. 50 8. 65 9. 68 10. 43 11. 65 12. 34 13. 47 14. 34 15. 49 16. 41 17. 13 18. 35 19. 53 20. 56 21. 16 22. 43 23. 69 24. 59 25. 48 26. 59 27. 86 28. 55 29. 68 30. 51 31. 33 32. 49 33. 67 34. 77 35. 81 36. 67 37. 71 38. 81 39. 68 40. 70 41. 77 42. 56

[23]:
```
births<-scan("http://robjhyndman.com/tsdldata/data/nybirths.dat")
birthstimeseries <- ts(births, frequency=12, start=c(1946,1))
birthstimeseries
```

A Time Series: 14 × 12

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oc
1946	26.663	23.598	26.931	24.740	25.806	24.364	24.477	23.901	23.175	23
1947	21.439	21.089	23.709	21.669	21.752	20.761	23.479	23.824	23.105	23
1948	21.937	20.035	23.590	21.672	22.222	22.123	23.950	23.504	22.238	23
1949	21.548	20.000	22.424	20.615	21.761	22.874	24.104	23.748	23.262	22
1950	22.604	20.894	24.677	23.673	25.320	23.583	24.671	24.454	24.122	24
1951	23.287	23.049	25.076	24.037	24.430	24.667	26.451	25.618	25.014	25
1952	23.798	22.270	24.775	22.646	23.988	24.737	26.276	25.816	25.210	25
1953	24.364	22.644	25.565	24.062	25.431	24.635	27.009	26.606	26.268	26
1954	24.657	23.304	26.982	26.199	27.210	26.122	26.706	26.878	26.152	26
1955	24.990	24.239	26.721	23.475	24.767	26.219	28.361	28.599	27.914	27
1956	26.217	24.218	27.914	26.975	28.527	27.139	28.982	28.169	28.056	29
1957	26.589	24.848	27.543	26.896	28.878	27.390	28.065	28.141	29.048	28
1958	27.132	24.924	28.963	26.589	27.931	28.009	29.229	28.759	28.405	27
1959	26.076	25.286	27.660	25.951	26.398	25.565	28.865	30.000	29.261	29

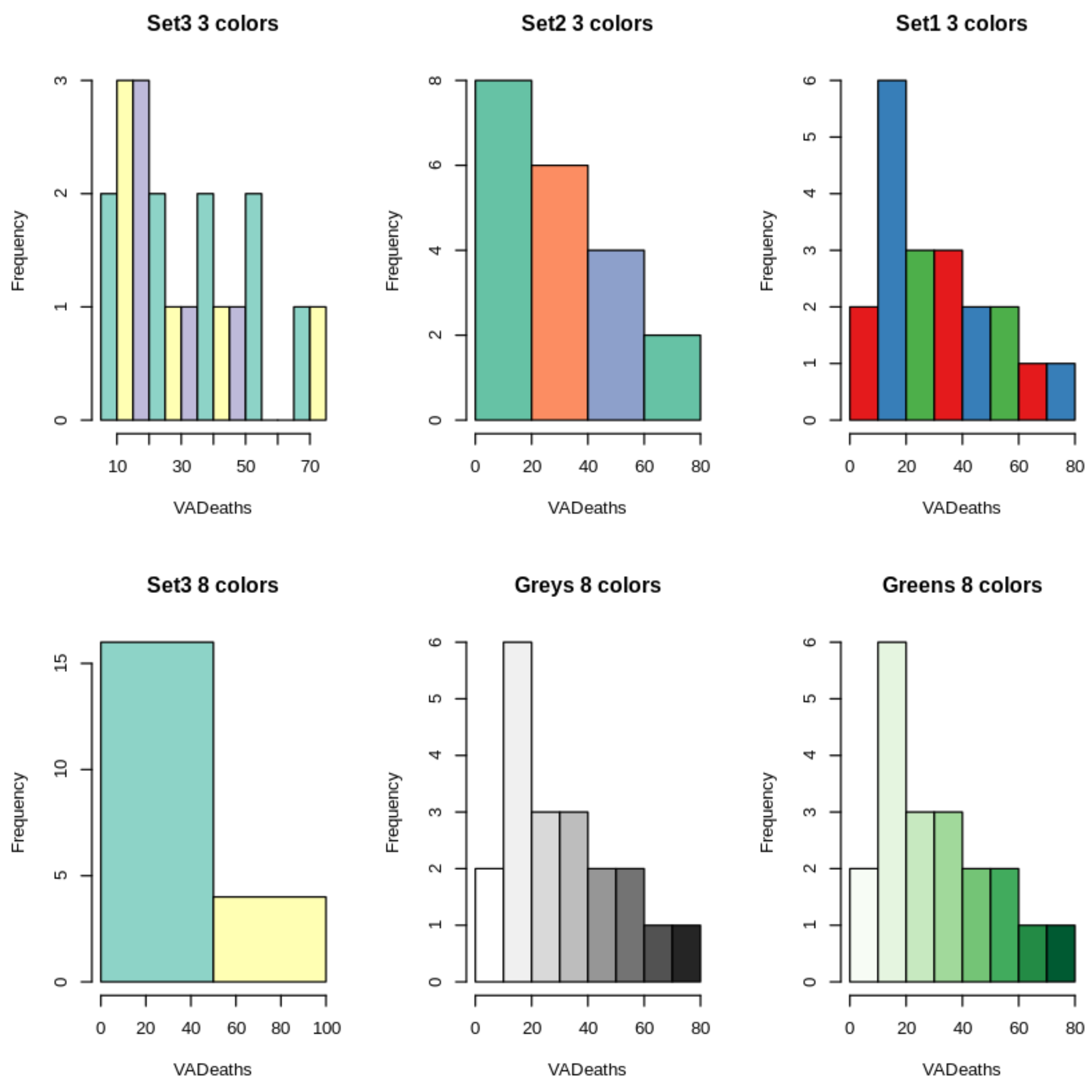
[25]:
```
souvenir<-scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirtimeseries <- ts(souvenir, frequency=12, start=c(1987,1))
souvenirtimeseries
```

A Time Series: 7 × 12

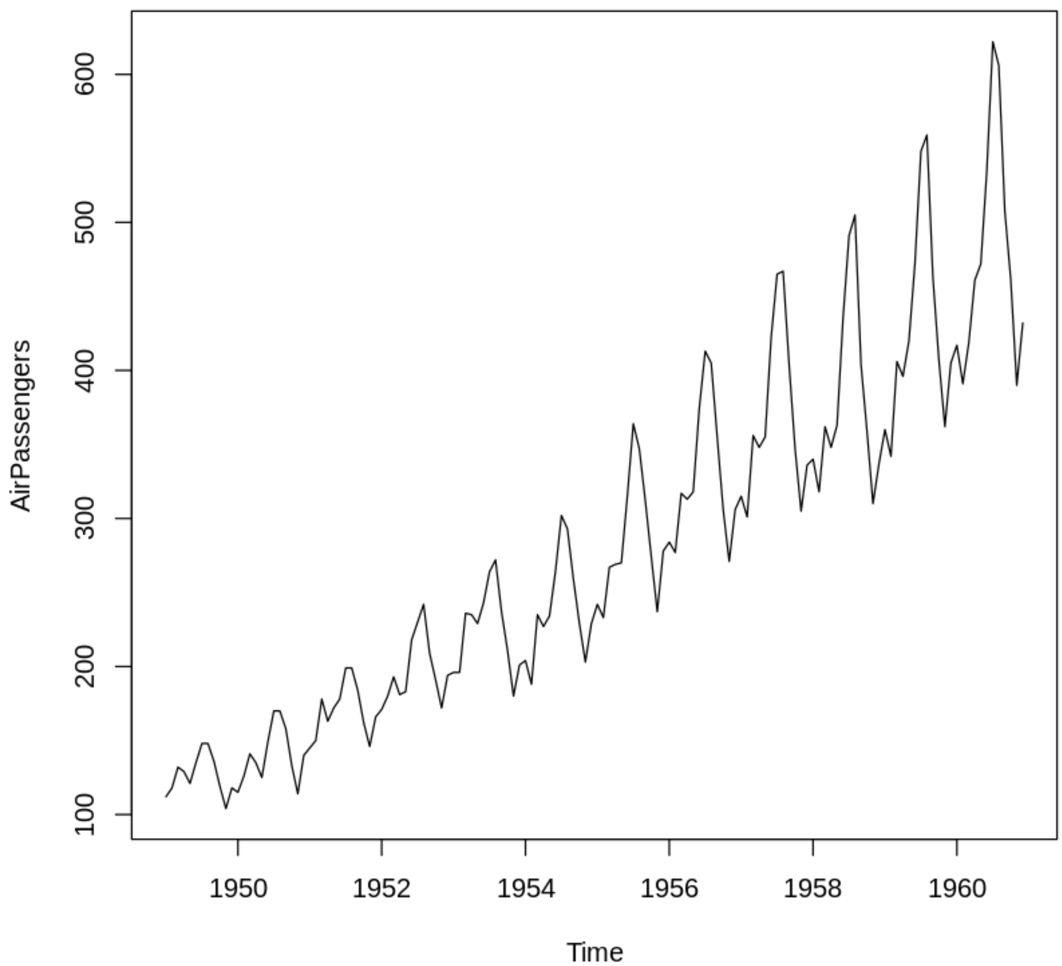
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
1987	1664.81	2397.53	2840.71	3547.29	3752.96	3714.74	4349.61	3566.
1988	2499.81	5198.24	7225.14	4806.03	5900.88	4951.34	6179.12	4752.
1989	4717.02	5702.63	9957.58	5304.78	6492.43	6630.80	7349.62	8176.
1990	5921.10	5814.58	12421.25	6369.77	7609.12	7224.75	8121.22	7979.
1991	4826.64	6470.23	9638.77	8821.17	8722.37	10209.48	11276.55	1255
1992	7615.03	9849.69	14558.40	11587.33	9332.56	13082.09	16732.78	1988
1993	10243.24	11266.88	21826.84	17357.33	15997.79	18601.53	26155.15	2858

0.0.5 Data Visualization

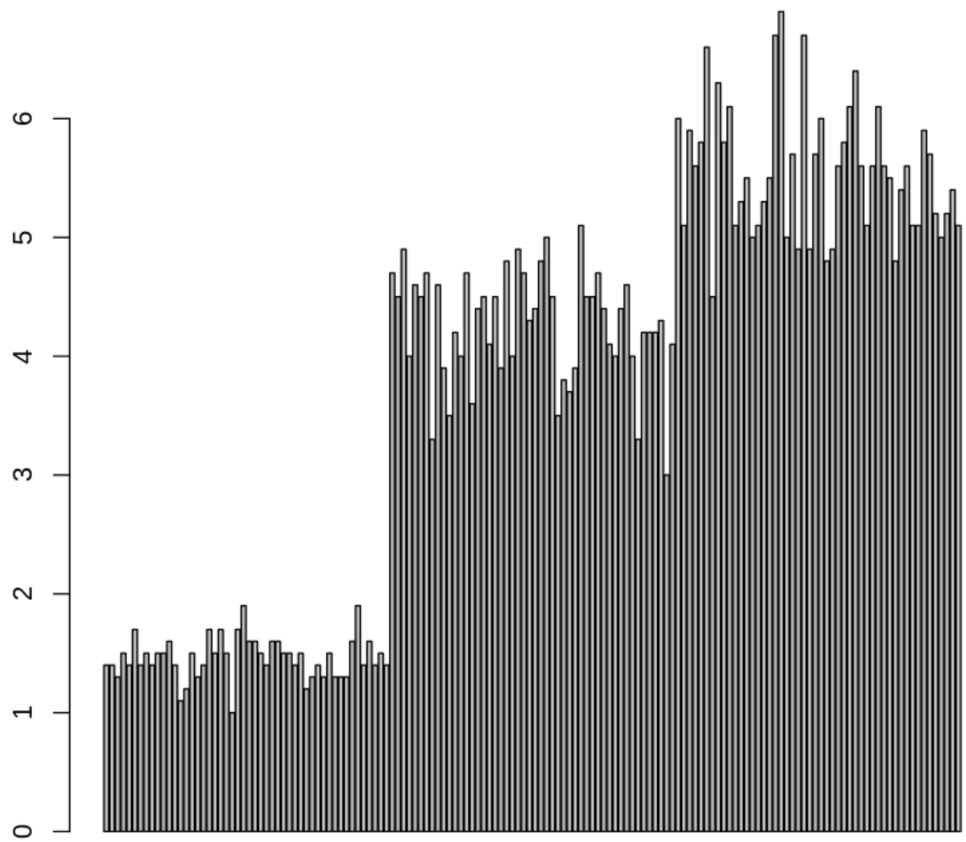
[26]:
```
library(RColorBrewer)
data(VADeaths)
par(mfrow=c(2,3))
hist(VADeaths,breaks=10, col=brewer.pal(3,"Set3"),main="Set3 3 colors")
hist(VADeaths,breaks=3 ,col=brewer.pal(3,"Set2"),main="Set2 3 colors")
hist(VADeaths,breaks=7, col=brewer.pal(3,"Set1"),main="Set1 3 colors")
hist(VADeaths,,breaks= 2, col=brewer.pal(8,"Set3"),main="Set3 8 colors")
hist(VADeaths,col=brewer.pal(8,"Greys"),main="Greys 8 colors")
hist(VADeaths,col=brewer.pal(8,"Greens"),main="Greens 8 colors")
```

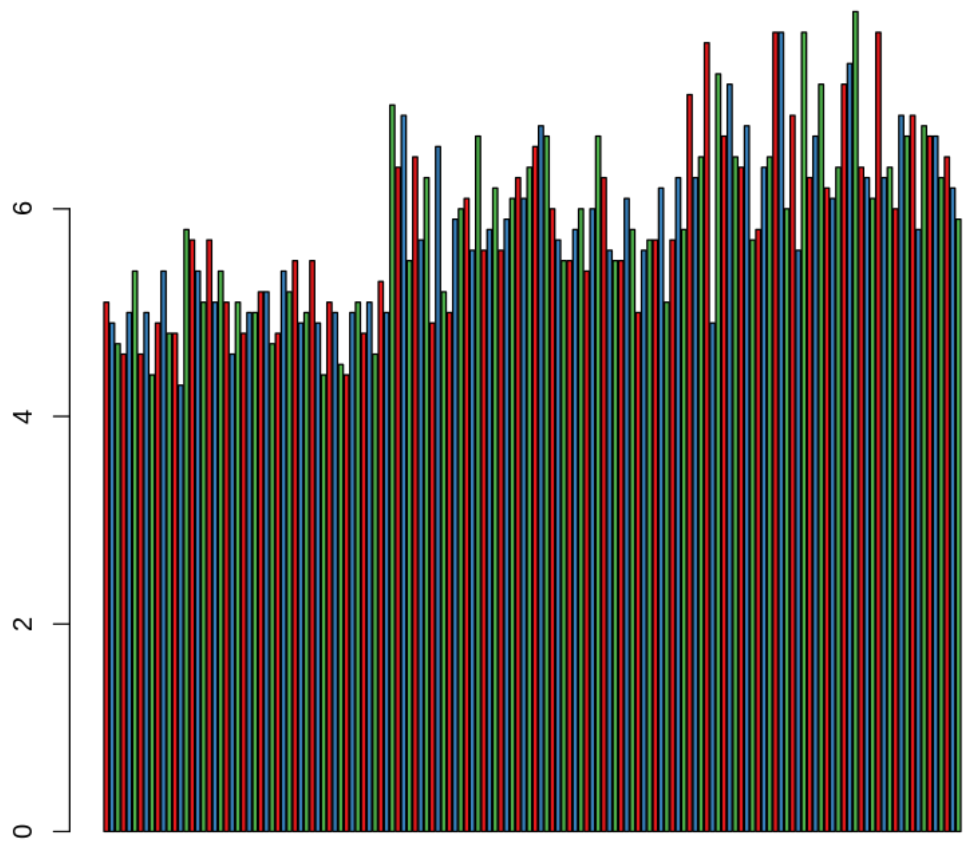


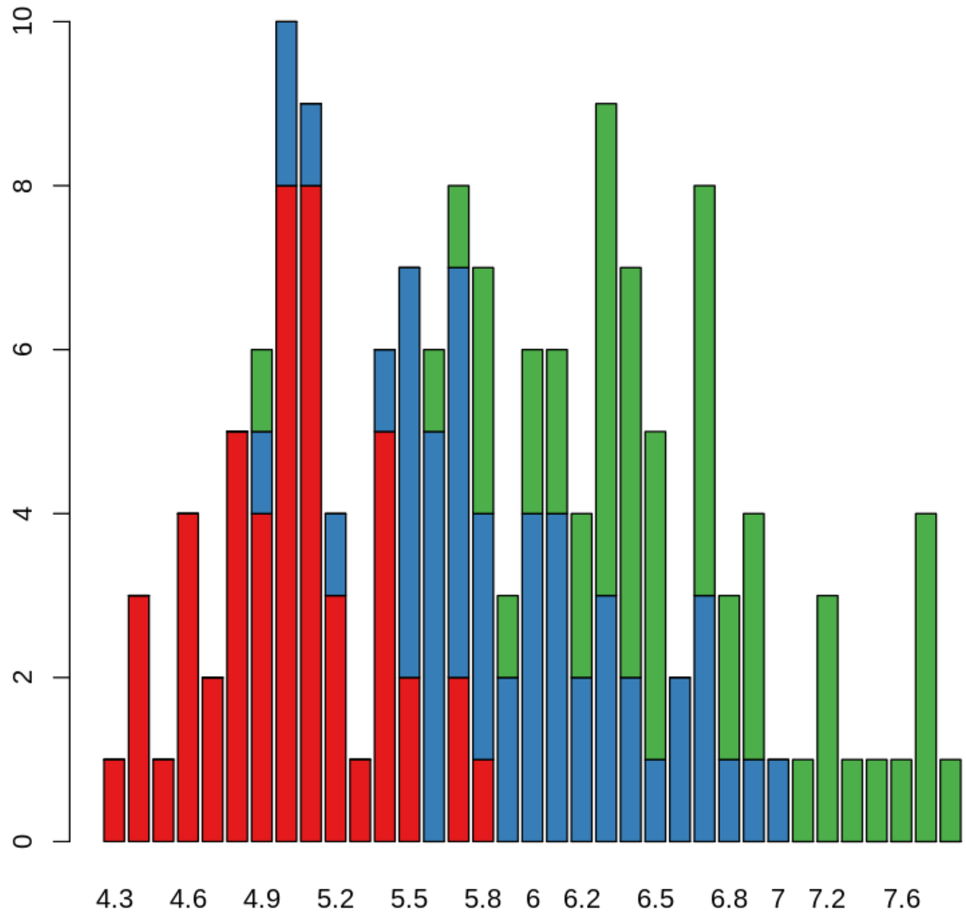
[27]:
```
data(AirPassengers)
plot(AirPassengers,type="l")
```



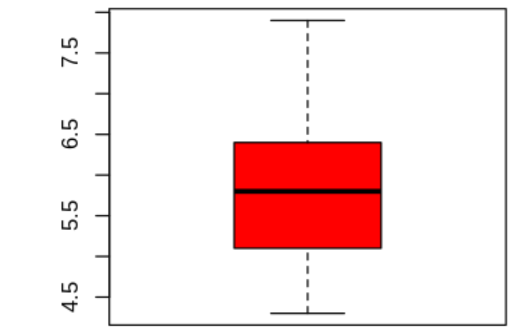
[28]:
```
data("iris")
barplot(iris$Petal.Length) #Creating simple Bar Graph
barplot(iris$Sepal.Length,col = brewer.pal(3,"Set1"))
barplot(table(iris$Species,iris$Sepal.Length),col = brewer.pal(3,"Set1"))
```

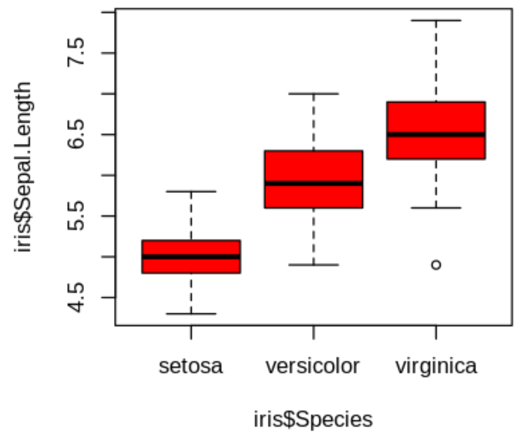

0
1
2
3
4
5
6

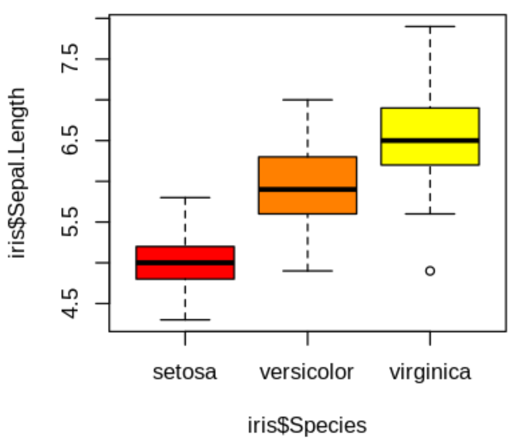

0
2
4
6

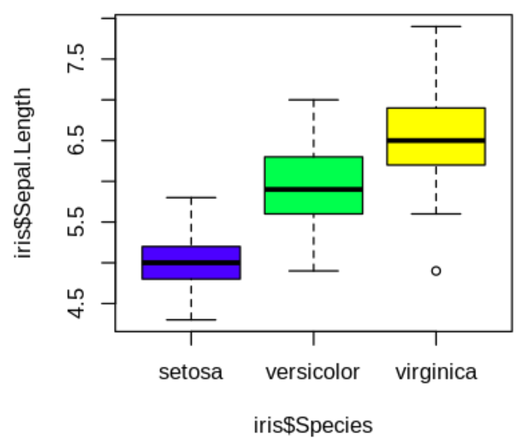


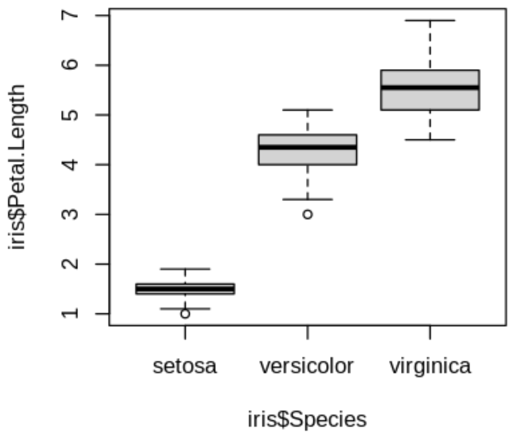
[29]:
```
data(iris)
par(mfrow=c(2,2))
boxplot(iris$Sepal.Length,col="red")
boxplot(iris$Sepal.Length~iris$Species,col="red")
boxplot(iris$Sepal.Length~iris$Species,col=heat.colors(3))
boxplot(iris$Sepal.Length~iris$Species,col=topo.colors(3))
boxplot(iris$Petal.Length~iris$Species)
```


4.5
5.5
6.5
7.5

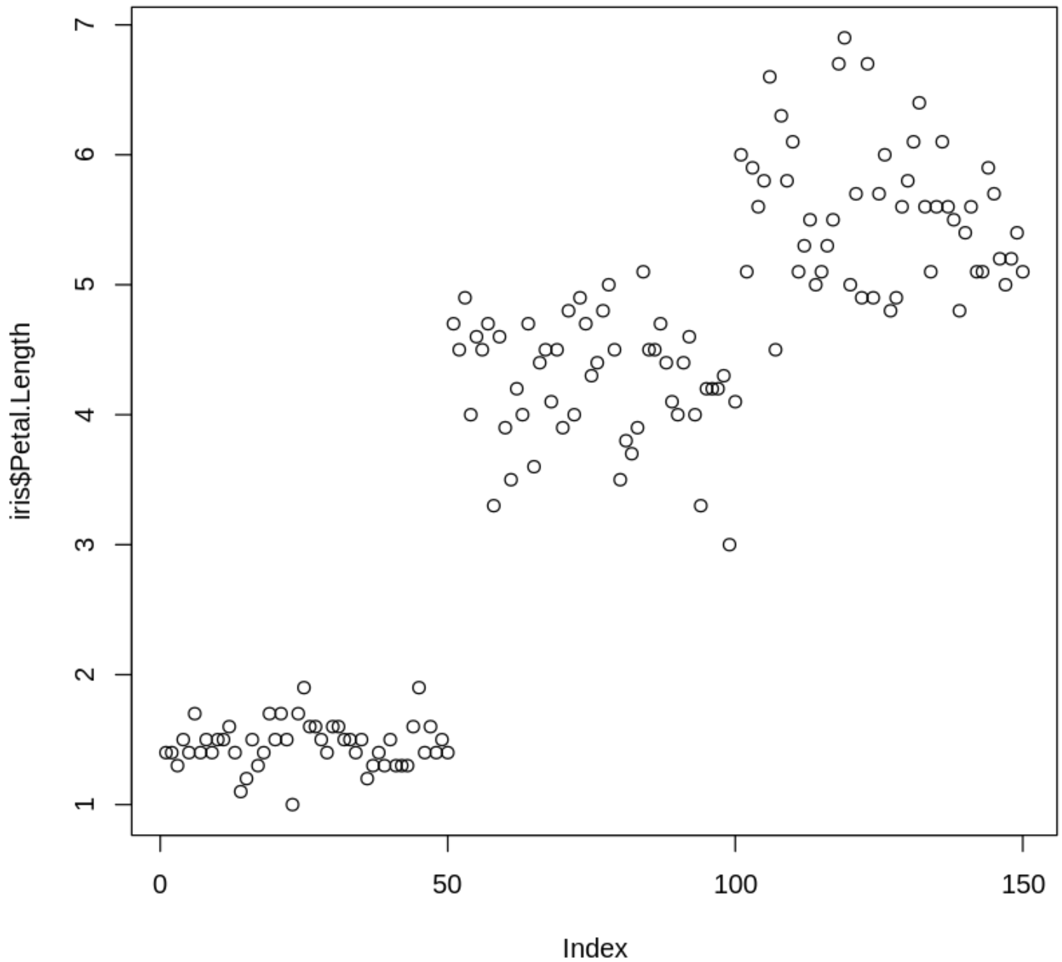

iris$Sepal.Length
4.5
5.5
6.5
7.5
setosa
versicolor
virginica
iris$Species


iris$Sepal.Length
4.5
5.5
6.5
7.5
setosa
versicolor
virginica
iris$Species

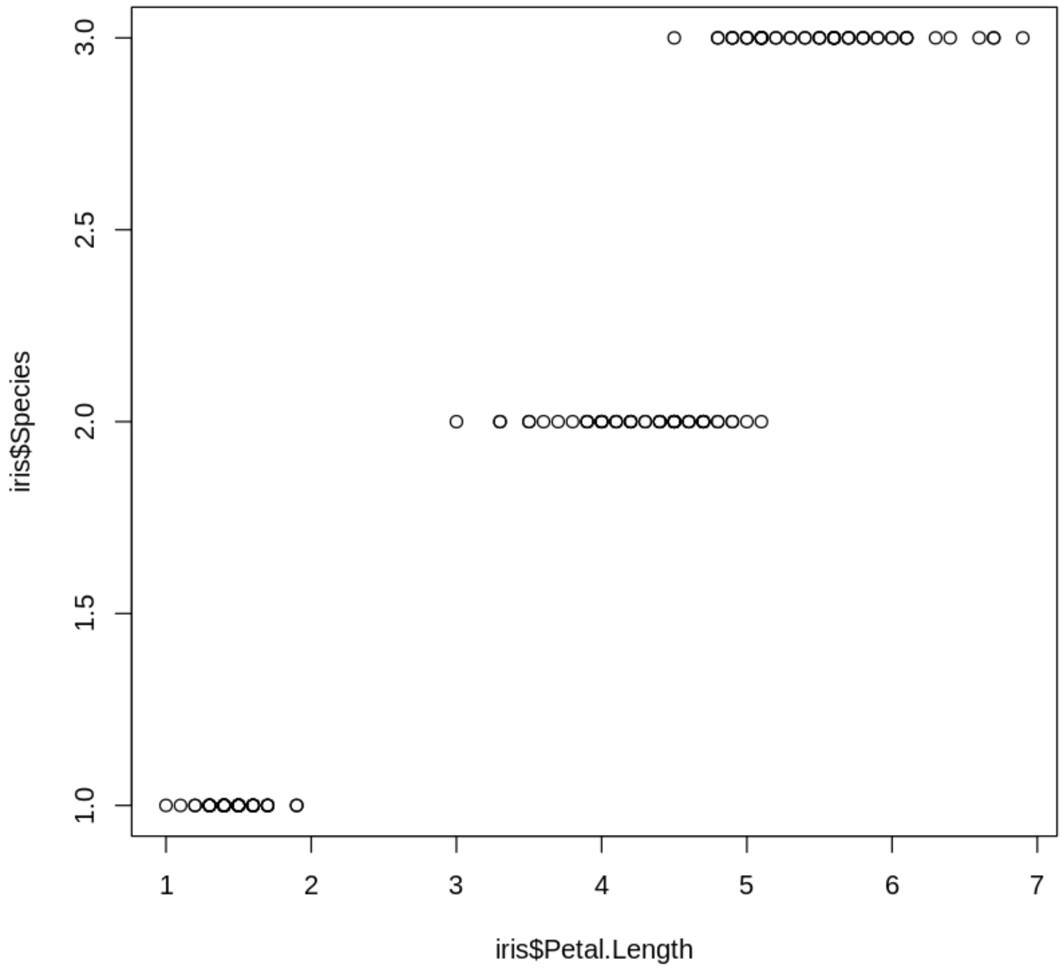

iris$Sepal.Length
4.5
5.5
6.5
7.5
setosa
versicolor
virginica
iris$Species



[30]:
```
plot(x=iris$Petal.Length)
plot(x=iris$Petal.Length,y=iris$Species)
```



iris$Petal.Length
1
2
3
4
5
6
7
0
50
100
150
Index


3.0
2.5
2.0
1.5
1.0
iris$Species
1
2
3
4
5
6
7
iris$Petal.Length