

Untitled

Prashant Jha

2024-05-31

##question 7

```
library(tidyverse)

aq <- airquality

wind_by_month <- sapply(split(aq, aq$Month), FUN = function(x) c(mean(x$Wind), sd(x$Wind)))

wind_by_month_df <- as.data.frame(t(wind_by_month))

colnames(wind_by_month_df) <- c('Mean', 'SD')

print(wind_by_month_df)
```

```
##           Mean      SD
## 5 11.622581 3.531450
## 6 10.266667 3.769234
## 7  8.941935 3.035981
## 8  8.793548 3.225930
## 9 10.180000 3.461254
```

This table is showing the Mean and Standard Deviation of the speed of wind of everyday from month May to September (represented by 5 to 9) in the year 1973.

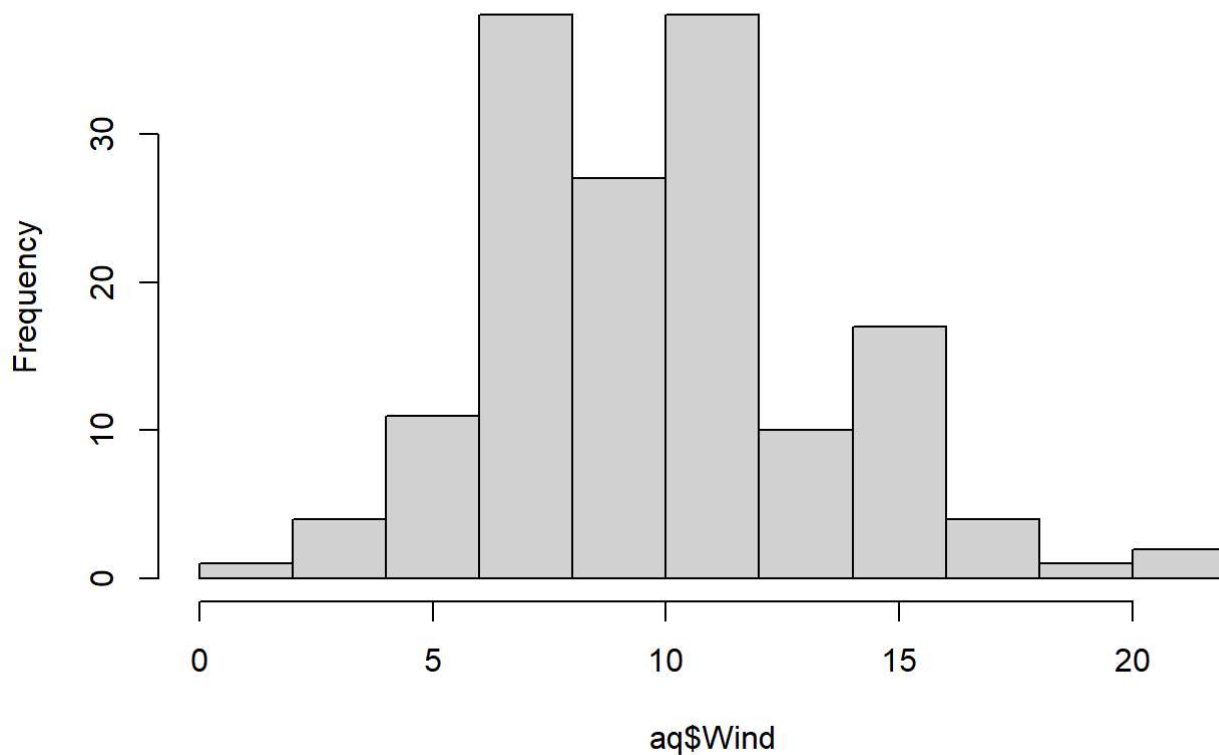
a.

I am using Shapiro-Wilk test to perform goodness-of-fit test on wind variable to check if it follows normal distribution or not.

```
library(stats)

hist(aq$Wind)
```

Histogram of aq\$Wind



```
shapiro.test(aq$Wind)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  aq$Wind  
## W = 0.98575, p-value = 0.1178
```

Since the W value is closer to 1 and p-value is >0.05 , I conclude that the variable Wind is normally distributed.

b.

Goodness-of-fit test to check if variances among different groups(months) are equal or not.

```
library(car)  
  
aq$Month <- factor(aq$Month)  
  
leveneTest(aq$Wind ~ aq$Month, data = aq, center = mean)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##           Df F value Pr(>F)
## group    4  0.1859 0.9454
##          148
```

Since $p\text{-value} > 0.05$, we accept the null hypothesis that means equal variance among months.

C.

Normality test of each month.

```
with(aq, shapiro.test(Wind[Month == 5]))
```

```
##
## Shapiro-Wilk normality test
##
## data:  Wind[Month == 5]
## W = 0.968, p-value = 0.4659
```

```
with(aq, shapiro.test(Wind[Month == 6]))
```

```
##
## Shapiro-Wilk normality test
##
## data:  Wind[Month == 6]
## W = 0.96858, p-value = 0.501
```

```
with(aq, shapiro.test(Wind[Month == 7]))
```

```
##
## Shapiro-Wilk normality test
##
## data:  Wind[Month == 7]
## W = 0.95003, p-value = 0.1564
```

```
with(aq, shapiro.test(Wind[Month == 8]))
```

```
##
## Shapiro-Wilk normality test
##
## data:  Wind[Month == 8]
## W = 0.98533, p-value = 0.937
```

```
with(aq, shapiro.test(Wind[Month == 9]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Wind[Month == 9]  
## W = 0.97853, p-value = 0.7852
```

Since the W value is closer to 1 and p-value is >0.05 in each month, I conclude that variable Wind in each Month is normally distributed. So classical one-way anova must be used.

```
dataset <- iris  
dataset
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa
## 7	4.6	3.4	1.4	0.3	setosa
## 8	5.0	3.4	1.5	0.2	setosa
## 9	4.4	2.9	1.4	0.2	setosa
## 10	4.9	3.1	1.5	0.1	setosa
## 11	5.4	3.7	1.5	0.2	setosa
## 12	4.8	3.4	1.6	0.2	setosa
## 13	4.8	3.0	1.4	0.1	setosa
## 14	4.3	3.0	1.1	0.1	setosa
## 15	5.8	4.0	1.2	0.2	setosa
## 16	5.7	4.4	1.5	0.4	setosa
## 17	5.4	3.9	1.3	0.4	setosa
## 18	5.1	3.5	1.4	0.3	setosa
## 19	5.7	3.8	1.7	0.3	setosa
## 20	5.1	3.8	1.5	0.3	setosa
## 21	5.4	3.4	1.7	0.2	setosa
## 22	5.1	3.7	1.5	0.4	setosa
## 23	4.6	3.6	1.0	0.2	setosa
## 24	5.1	3.3	1.7	0.5	setosa
## 25	4.8	3.4	1.9	0.2	setosa
## 26	5.0	3.0	1.6	0.2	setosa
## 27	5.0	3.4	1.6	0.4	setosa
## 28	5.2	3.5	1.5	0.2	setosa
## 29	5.2	3.4	1.4	0.2	setosa
## 30	4.7	3.2	1.6	0.2	setosa
## 31	4.8	3.1	1.6	0.2	setosa
## 32	5.4	3.4	1.5	0.4	setosa
## 33	5.2	4.1	1.5	0.1	setosa
## 34	5.5	4.2	1.4	0.2	setosa
## 35	4.9	3.1	1.5	0.2	setosa
## 36	5.0	3.2	1.2	0.2	setosa
## 37	5.5	3.5	1.3	0.2	setosa
## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor

## 52	6.4	3.2	4.5	1.5 versicolor
## 53	6.9	3.1	4.9	1.5 versicolor
## 54	5.5	2.3	4.0	1.3 versicolor
## 55	6.5	2.8	4.6	1.5 versicolor
## 56	5.7	2.8	4.5	1.3 versicolor
## 57	6.3	3.3	4.7	1.6 versicolor
## 58	4.9	2.4	3.3	1.0 versicolor
## 59	6.6	2.9	4.6	1.3 versicolor
## 60	5.2	2.7	3.9	1.4 versicolor
## 61	5.0	2.0	3.5	1.0 versicolor
## 62	5.9	3.0	4.2	1.5 versicolor
## 63	6.0	2.2	4.0	1.0 versicolor
## 64	6.1	2.9	4.7	1.4 versicolor
## 65	5.6	2.9	3.6	1.3 versicolor
## 66	6.7	3.1	4.4	1.4 versicolor
## 67	5.6	3.0	4.5	1.5 versicolor
## 68	5.8	2.7	4.1	1.0 versicolor
## 69	6.2	2.2	4.5	1.5 versicolor
## 70	5.6	2.5	3.9	1.1 versicolor
## 71	5.9	3.2	4.8	1.8 versicolor
## 72	6.1	2.8	4.0	1.3 versicolor
## 73	6.3	2.5	4.9	1.5 versicolor
## 74	6.1	2.8	4.7	1.2 versicolor
## 75	6.4	2.9	4.3	1.3 versicolor
## 76	6.6	3.0	4.4	1.4 versicolor
## 77	6.8	2.8	4.8	1.4 versicolor
## 78	6.7	3.0	5.0	1.7 versicolor
## 79	6.0	2.9	4.5	1.5 versicolor
## 80	5.7	2.6	3.5	1.0 versicolor
## 81	5.5	2.4	3.8	1.1 versicolor
## 82	5.5	2.4	3.7	1.0 versicolor
## 83	5.8	2.7	3.9	1.2 versicolor
## 84	6.0	2.7	5.1	1.6 versicolor
## 85	5.4	3.0	4.5	1.5 versicolor
## 86	6.0	3.4	4.5	1.6 versicolor
## 87	6.7	3.1	4.7	1.5 versicolor
## 88	6.3	2.3	4.4	1.3 versicolor
## 89	5.6	3.0	4.1	1.3 versicolor
## 90	5.5	2.5	4.0	1.3 versicolor
## 91	5.5	2.6	4.4	1.2 versicolor
## 92	6.1	3.0	4.6	1.4 versicolor
## 93	5.8	2.6	4.0	1.2 versicolor
## 94	5.0	2.3	3.3	1.0 versicolor
## 95	5.6	2.7	4.2	1.3 versicolor
## 96	5.7	3.0	4.2	1.2 versicolor
## 97	5.7	2.9	4.2	1.3 versicolor
## 98	6.2	2.9	4.3	1.3 versicolor
## 99	5.1	2.5	3.0	1.1 versicolor
## 100	5.7	2.8	4.1	1.3 versicolor
## 101	6.3	3.3	6.0	2.5 virginica
## 102	5.8	2.7	5.1	1.9 virginica
## 103	7.1	3.0	5.9	2.1 virginica

## 104	6.3	2.9	5.6	1.8	virginica
## 105	6.5	3.0	5.8	2.2	virginica
## 106	7.6	3.0	6.6	2.1	virginica
## 107	4.9	2.5	4.5	1.7	virginica
## 108	7.3	2.9	6.3	1.8	virginica
## 109	6.7	2.5	5.8	1.8	virginica
## 110	7.2	3.6	6.1	2.5	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 112	6.4	2.7	5.3	1.9	virginica
## 113	6.8	3.0	5.5	2.1	virginica
## 114	5.7	2.5	5.0	2.0	virginica
## 115	5.8	2.8	5.1	2.4	virginica
## 116	6.4	3.2	5.3	2.3	virginica
## 117	6.5	3.0	5.5	1.8	virginica
## 118	7.7	3.8	6.7	2.2	virginica
## 119	7.7	2.6	6.9	2.3	virginica
## 120	6.0	2.2	5.0	1.5	virginica
## 121	6.9	3.2	5.7	2.3	virginica
## 122	5.6	2.8	4.9	2.0	virginica
## 123	7.7	2.8	6.7	2.0	virginica
## 124	6.3	2.7	4.9	1.8	virginica
## 125	6.7	3.3	5.7	2.1	virginica
## 126	7.2	3.2	6.0	1.8	virginica
## 127	6.2	2.8	4.8	1.8	virginica
## 128	6.1	3.0	4.9	1.8	virginica
## 129	6.4	2.8	5.6	2.1	virginica
## 130	7.2	3.0	5.8	1.6	virginica
## 131	7.4	2.8	6.1	1.9	virginica
## 132	7.9	3.8	6.4	2.0	virginica
## 133	6.4	2.8	5.6	2.2	virginica
## 134	6.3	2.8	5.1	1.5	virginica
## 135	6.1	2.6	5.6	1.4	virginica
## 136	7.7	3.0	6.1	2.3	virginica
## 137	6.3	3.4	5.6	2.4	virginica
## 138	6.4	3.1	5.5	1.8	virginica
## 139	6.0	3.0	4.8	1.8	virginica
## 140	6.9	3.1	5.4	2.1	virginica
## 141	6.7	3.1	5.6	2.4	virginica
## 142	6.9	3.1	5.1	2.3	virginica
## 143	5.8	2.7	5.1	1.9	virginica
## 144	6.8	3.2	5.9	2.3	virginica
## 145	6.7	3.3	5.7	2.5	virginica
## 146	6.7	3.0	5.2	2.3	virginica
## 147	6.3	2.5	5.0	1.9	virginica
## 148	6.5	3.0	5.2	2.0	virginica
## 149	6.2	3.4	5.4	2.3	virginica
## 150	5.9	3.0	5.1	1.8	virginica

10.a),b),c)

Putting the iris data set and creating a hierarchical clustering using these methods.

```
hcasingle <- hclust(dist(dataset),method = "single")
```

```
## Warning in dist(dataset): NAs introduced by coercion
```

```
hcacomplete <- hclust(dist(dataset),method = "complete")
```

```
## Warning in dist(dataset): NAs introduced by coercion
```

```
hcaaverage <- hclust(dist(dataset),method = "average")
```

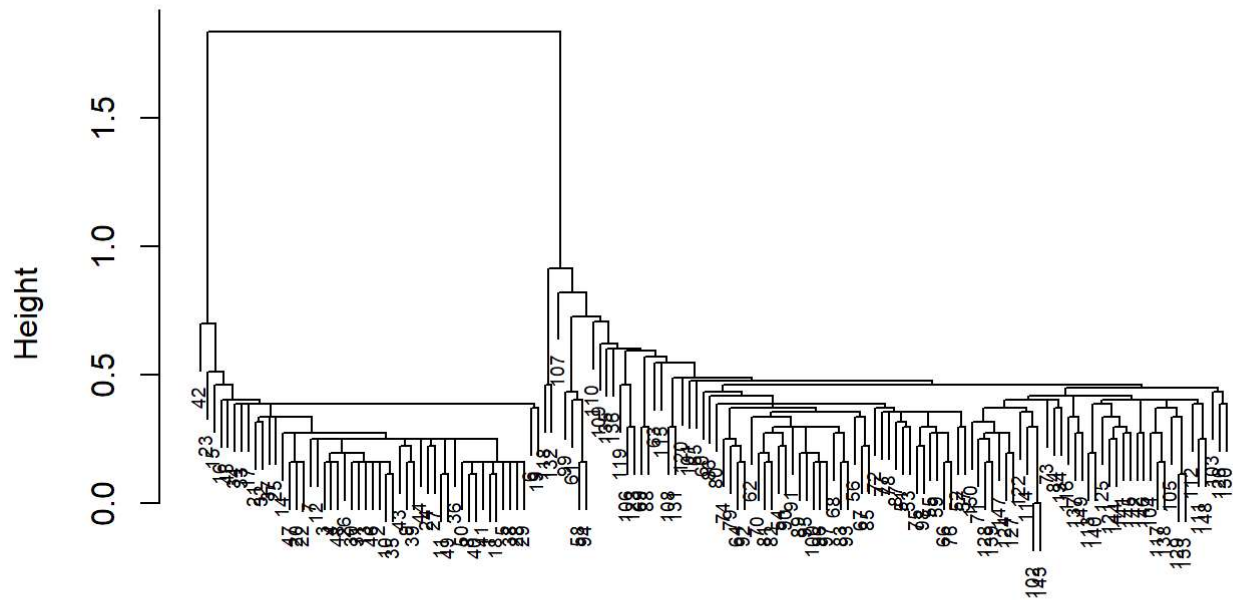
```
## Warning in dist(dataset): NAs introduced by coercion
```

Getting the dendrogram using three different models i.e. single linkage, complete linkage,

average linkage

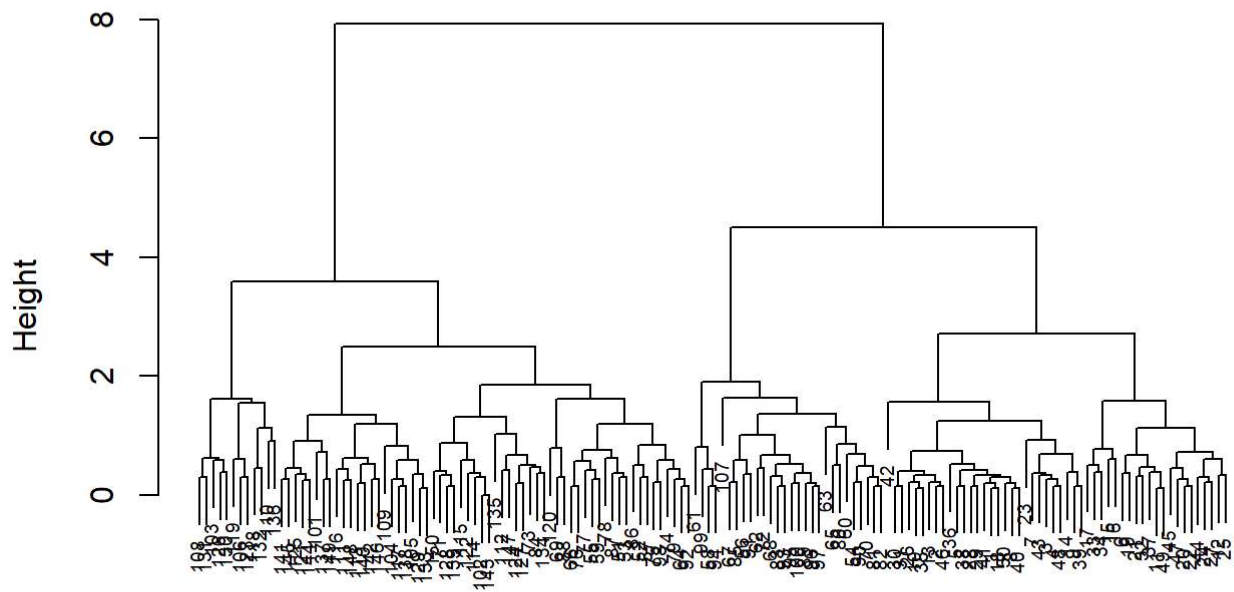
```
plot(hcasingle, main = "Single Linkage",xlab="",sub="",cex = 0.6)
```


Single Linkage



```
plot(hcacomplete, main = "Complete Linkage",xlab="",sub="",cex = 0.6)
```

Complete Linkage



```
plot(hcaaverage, main = "Average Linkage",xlab="",sub="",cex = 0.6)
```

