

ΑΣΚΗΣΗ 1

1.

```
> mean(airquality$Ozone) # εδω βλεπουμε οτι ορισμενες τιμες στην μεταβλητη Ozone
ειναι αγνωστες
[1] NA
> mean(airquality$Ozone,na.rm=T) #χρησιμοποιουμε το na.rm=T για να μην υπολογισουμε
αυτες τις τιμες
[1] 42.12931
var(airquality$Ozone,na.rm=T) # ετοιμη συναρτηση για τον υπολογισμο της διασπορας
του δειγματος
[1] 1088.201
sd(airquality$Ozone,na.rm=T)^2 # η τυπικη αποκλιση στο τετραγωνο που ισουται με
την διασπορα
[1] 1088.201
```

2.

```
complete.cases(airquality) # βλεπουμε οτι υπαρχουν πολλα FALSE στην μεταβλητη μας
airquality[complete.cases(airquality), ] # διωχνουμε τις τιμες αυτες
airquality<- airquality[complete.cases(airquality), ] # ονομαζουμε την μεταβλητη
για διευκολυνση
> dim(airquality) # βλεπουμε αλλαξε η διασταση αρα εφυγαν οι αγνωστες τιμες
[1] 111 6
> nrow(airquality) # λιγοτερες σειρες οσες και η διασταση της μεταβλητης
[1] 111
```

3.

```
TempC=5/9 * (airquality$Temp-32) # μετατροπη νεας μεταβλητης Temp απο βαθμους
Φαρεναιτ σε Κελσιου
ή
convert_fahr_to_kelvin <- function(temp) {
+   kelvin <- ((temp - 32) * (5 / 9))
+   return(kelvin)
+ } # εναλλακτικος τροπος
```

4.

```
attach(airquality) # επισυναπτουμε την μεταβλητη μας
> mean(airquality$Solar.R) # μεση τιμη για τις μεταβλητες
Solar.R, Wind, Day, TempC, Month, Ozone
[1] 184.8018
> mean(airquality$Wind)
[1] 9.93964
> mean(airquality$Day)
[1] 15.94595
> mean(airquality$Month)
[1] 7.216216
> mean(airquality$Ozone)
[1] 42.0991
> mean(TempC)
[1] 25.44044
> sd(airquality$Ozone) # τυπικη αποκλιση Solar.R, Wind, Day, TempC, Month, Ozone
```

```

[1] 33.27597
> sd(TempC)
[1] 5.294427
> sd(airquality$Solar.R)
[1] 91.1523
> sd(airquality$Wind)
[1] 3.557713
> sd(airquality$Day)
[1] 8.707194
> sd(airquality$Month)
[1] 1.473434
quantile(airquality$Ozone,0.25)      # εκατοστημοριο 25%
Soalr.R,Wind,Day,TempC,Month,Ozone
25%
18
> quantile(airquality$Solar.R,0.25)
25%
113.5
> quantile(airquality$Wind,0.25)
25%
7.4
> quantile(TempC,0.25)
25%
22.22222
> quantile(airquality$Day,0.25)
25%
9
> quantile(airquality$Month,0.25)
25%
6
median(airquality$Ozone) # διαμεσος Soalr.R,Wind,Day,TempC,Month,Ozone
[1] 31
> median(airquality$Solar.R)
[1] 207
> median(airquality$Wind)
[1] 9.7
> median(airquality$Day)
[1] 16
> median(airquality$Month)
[1] 7
> median(TempC)
[1] 26.11111
> quantile(airquality$Ozone,0.75) # εκατοστημοριο 75%
Soalr.R,Wind,Day,TempC,Month,Ozone
75%
62
> quantile(airquality$Solar.R,0.75)
75%
255.5
> quantile(airquality$Wind,0.75)
75%
11.5

```

```

> quantile(airquality$Day, 0.75)
75%
22.5
> quantile(airquality$Month, 0.75)
75%
9
> quantile(TempC, 0.75)
75%
29.44444
> IQR(airquality$Ozone) # ενδοτεταρτημοριακο ευρος
Soalr.R, Wind, Day, TempC, Month, Ozone
[1] 44
> IQR(airquality$Solar.R)
[1] 142
> IQR(airquality$Wind)
[1] 4.1
> IQR(airquality$Day)
[1] 13.5
> IQR(airquality$Month)
[1] 3
> IQR(TempC)
[1] 7.5

> quantile(airquality$Day, 0.05) # 0.05-ποσοστιαιο σημειο
Soalr.R, Wind, Day, TempC, Month, Ozone
5%
2
> quantile(airquality$Month, 0.05)
5%
5
> quantile(airquality$Wind, 0.05)
5%
4.6
> quantile(airquality$Solar.R, 0.05)
5%
22
> quantile(airquality$Ozone, 0.05)
5%
8.5
> quantile(TempC, 0.05)
5%
16.11111
> quantile(TempC, 0.95) # 0.95-ποσοστιαιο σημειο Soalr.R, Wind, Day, TempC, Month, Ozone
95%
33.61111
> quantile(airquality$Ozone, 0.95)
95%
109
> quantile(airquality$Wind, 0.95)
95%
15.5
> quantile(airquality$Day, 0.95)

```

```

95%
30
> quantile(airquality$Month,0.95)
95%
9
> quantile(airquality$Solar.R,0.95)
95%
310
mean(((airquality$Ozone-mean(airquality$Ozone))/sd(airquality$Ozone))^3) #
συντελεστής ασυμμετρίας
[1] 1.231275
> mean(((airquality$Ozone-mean(airquality$Ozone))/sd(airquality$Ozone))^4) #
συντελεστής κυρτώσης
[1] 4.128994
> mean(((airquality$Solar.R-mean(airquality$Solar.R))/sd(airquality$Solar.R))^3)
[1] -0.4796906
> mean(((airquality$Solar.R-mean(airquality$Solar.R))/sd(airquality$Solar.R))^4)
[1] 2.033632
> mean(((airquality$Wind-mean(airquality$Wind))/sd(airquality$Wind))^3)
[1] 0.449498
> mean(((airquality$Wind-mean(airquality$Wind))/sd(airquality$Wind))^4)
[1] 3.222443
> mean(((airquality$Month-mean(airquality$Month))/sd(airquality$Month))^4)
[1] 1.722203
> mean(((airquality$Month-mean(airquality$Month))/sd(airquality$Month))^3)
[1] -0.2873407
> mean(((airquality$Day-mean(airquality$Day))/sd(airquality$Day))^3)
[1] -0.01265914
> mean(((airquality$Day-mean(airquality$Day))/sd(airquality$Day))^4)
[1] 1.919051
> mean(((TempC-mean(TempC))/sd(TempC))^3)
[1] -0.2220609
> mean(((TempC-mean(TempC))/sd(TempC))^4)
[1] 2.290127
> mean<-c(42.0991,184.8018,9.9396,25.4404,7.2162,15.9459) # δημιουργούμε
διανύσματα
> sd<-c(33.2759,91.1523,3.5577,5.2944,1.4734,8.7071) # χρησιμοποιώντας τις τιμές
των μεταβλητών
> Q1<-c(18,113.5,7.4,21.6666,9,6)
> median<-c(31,207,9.7,26.1111,7,16)
> Q3<-c(62,255.5,11.5,22.5,9,29.4444)
> IQR<-c(44,142,4.1,7.5,3,13.5)
> quantile0.05<-c(8.5,22,4.6,16.1111,5,2)
> quantile0.95<-c(109,310,15.5,33.6111,9,30)
> b1<-c(1.2312,-0.4796,0.4494,-0.2220,-0.2873,-0.0126)
> b2<-c(4.1289,2.0336,3.2224,2.2901,1.7222,1.9190)
> x<-
matrix(c(mean,sd,Q1,median,Q3,IQR,quantile0.05,quantile0.95,b1,b2),ncol=6,byrow=TRUE)
# δημιουργούμε έναν πίνακα με τα διανύσματα των βασικών μετρών περιγραφικής
στατιστικής
> x

```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	42.0991	184.8018	9.9396	25.4404	7.2162	15.9459
[2,]	33.2759	91.1523	3.5577	5.2944	1.4734	8.7071
[3,]	18.0000	113.5000	7.4000	21.6666	9.0000	6.0000
[4,]	31.0000	207.0000	9.7000	26.1111	7.0000	16.0000
[5,]	62.0000	255.5000	11.5000	22.5000	9.0000	29.4444
[6,]	44.0000	142.0000	4.1000	7.5000	3.0000	13.5000
[7,]	8.5000	22.0000	4.6000	16.1111	5.0000	2.0000
[8,]	109.0000	310.0000	15.5000	33.6111	9.0000	30.0000
[9,]	1.2312	-0.4796	0.4494	-0.2220	-0.2873	-0.0126
[10,]	4.1289	2.0336	3.2224	2.2901	1.7222	1.9190

```
> x<-as.data.frame(x) # μετατρέπουμε τον πίνακα σε dataframe
```

```
> x
```

	V1	V2	V3	V4	V5	V6
1	42.0991	184.8018	9.9396	25.4404	7.2162	15.9459
2	33.2759	91.1523	3.5577	5.2944	1.4734	8.7071
3	18.0000	113.5000	7.4000	21.6666	9.0000	6.0000
4	31.0000	207.0000	9.7000	26.1111	7.0000	16.0000
5	62.0000	255.5000	11.5000	22.5000	9.0000	29.4444
6	44.0000	142.0000	4.1000	7.5000	3.0000	13.5000
7	8.5000	22.0000	4.6000	16.1111	5.0000	2.0000
8	109.0000	310.0000	15.5000	33.6111	9.0000	30.0000
9	1.2312	-0.4796	0.4494	-0.2220	-0.2873	-0.0126
10	4.1289	2.0336	3.2224	2.2901	1.7222	1.9190

```
> names(x)
```

```
[1] "V1" "V2" "V3" "V4" "V5" "V6"
```

```
> names(x)<-c('Ozone','Solar.R','Wind','TempC','Month','Day') # ονομάζουμε τις
στήλες του
```

```
> x
```

	Ozone	Solar.R	Wind	TempC	Month	Day
1	42.0991	184.8018	9.9396	25.4404	7.2162	15.9459
2	33.2759	91.1523	3.5577	5.2944	1.4734	8.7071
3	18.0000	113.5000	7.4000	21.6666	9.0000	6.0000
4	31.0000	207.0000	9.7000	26.1111	7.0000	16.0000
5	62.0000	255.5000	11.5000	22.5000	9.0000	29.4444
6	44.0000	142.0000	4.1000	7.5000	3.0000	13.5000
7	8.5000	22.0000	4.6000	16.1111	5.0000	2.0000
8	109.0000	310.0000	15.5000	33.6111	9.0000	30.0000
9	1.2312	-0.4796	0.4494	-0.2220	-0.2873	-0.0126
10	4.1289	2.0336	3.2224	2.2901	1.7222	1.9190

```
> x<-
```

```
data.frame(x,row.names=c('mean','sd','Q1','median','Q3','IQR','quantile0.05','quan
tile0.95','b1','b2')) #ονομάζουμε τις γραμμές του
```

```
> x
```

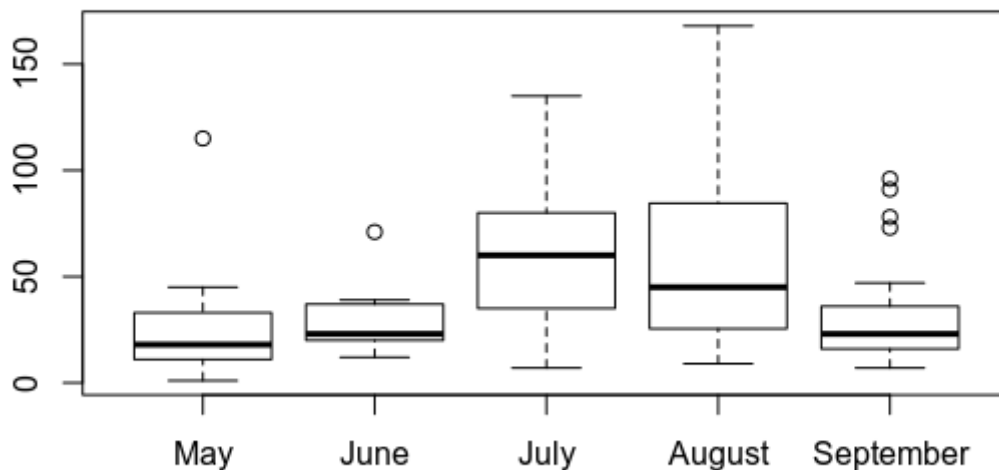
	Ozone	Solar.R	Wind	TempC	Month	Day
mean	42.0991	184.8018	9.9396	25.4404	7.2162	15.9459
sd	33.2759	91.1523	3.5577	5.2944	1.4734	8.7071
Q1	18.0000	113.5000	7.4000	21.6666	9.0000	6.0000

median	31.0000	207.0000	9.7000	26.1111	7.0000	16.0000
Q3	62.0000	255.5000	11.5000	22.5000	9.0000	29.4444
IQR	44.0000	142.0000	4.1000	7.5000	3.0000	13.5000
quantile0.05	8.5000	22.0000	4.6000	16.1111	5.0000	2.0000
quantile0.95	109.0000	310.0000	15.5000	33.6111	9.0000	30.0000
b1	1.2312	-0.4796	0.4494	-0.2220	-0.2873	-0.0126
b2	4.1289	2.0336	3.2224	2.2901	1.7222	1.9190

5.

>

```
boxplot(Ozone~Month,dara=airquality,names=c("May","June","July","August","September"))
```



boxplot για την μεταβλητη Ozone αναλογα με το μηνια μετρησεων

6.

```
> TempCat<-function(x){ # φτιαχνουμε συναρτηση για να κατηγοριοποιησουμε τη
μεταβλητη Temp
+ sapply(TempC,function(TempC)
+ if (TempC>25) # αν η τιμες ειναι πανω απο 25 θεωρουνται υψηλες
+ "High"
+ else if (TempC<25) # αν η τιμες ειναι κατω απο 25 θεωρουνται χαμηλες
+ "Low"
+ else
+ "Middle" # αν η τιμες ειναι 25 τοτε ειναι στη μεση
+ ) } # βαλαμε και μια τριτη κατηγορια γιατι ειχαμε προβλημα παρακατω για τις
τιμες που ειναι 25
> TempCat<-TempCat(TempC)
> TempCat
[1] "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low"
"Low" "Low"
```

```

[11] "Low"      "Low"      "Low"      "Low"      "Low"      "Low"      "Low"      "Low"      "Low"
"Low"      "Low"
[21] "Low"      "High"     "High"     "Low"      "High"     "High"     "High"     "High"     "High"
"Middle"   "Low"
[31] "Low"      "Low"      "Low"      "High"     "High"     "High"     "High"     "High"     "High"
"High"     "High"
[41] "High"     "High"     "Low"      "High"     "High"     "High"     "High"     "High"     "High"
"High"     "High"
[51] "Low"      "High"     "High"     "High"     "High"     "High"     "High"     "High"     "High"
"High"     "High"
[61] "High"     "High"     "High"     "High"     "High"     "High"     "High"     "High"     "High"
"Middle"   "High"
[91] "Low"      "Low"      "High"     "Low"      "Middle"   "Low"      "Low"      "Low"      "High"
"Low"      "Low"
[101] "Low"     "High"     "Low"      "Low"      "High"     "Low"      "Low"      "Low"      "Low"
"Low"      "Low"
[111] "Low"

```

```

> Monthhh<-function(x){ # καναμε μια συναρτηση για να δωσουμε ονοματα στους μηνες
+ sapply(Month,function(Month)
+ if (Month==5)
+ "May"
+ else if (Month==6)
+ "June"
+ else if (Month==7)
+ "Jully"
+ else if (Month==8)
+ "August"
+ else if (Month==9)
+ "September"
+ )}
> Month<-Monthhh(Month) # ονομασαμε την συναρτη για ευκολια
> mytable<-table(Month,TempCat) # πινακας διπλης εισοδου της TempCat και της Month
> mytable

```

	TempCat		
Month	High	Low	Middle
August	19	2	2
Jully	24	2	0
June	4	4	1
May	2	22	0
September	12	16	1

```

> prop.table(mytable) # πινακας σχετικων συχνοτητων διπλης εισοδου της TempCat
και της Month

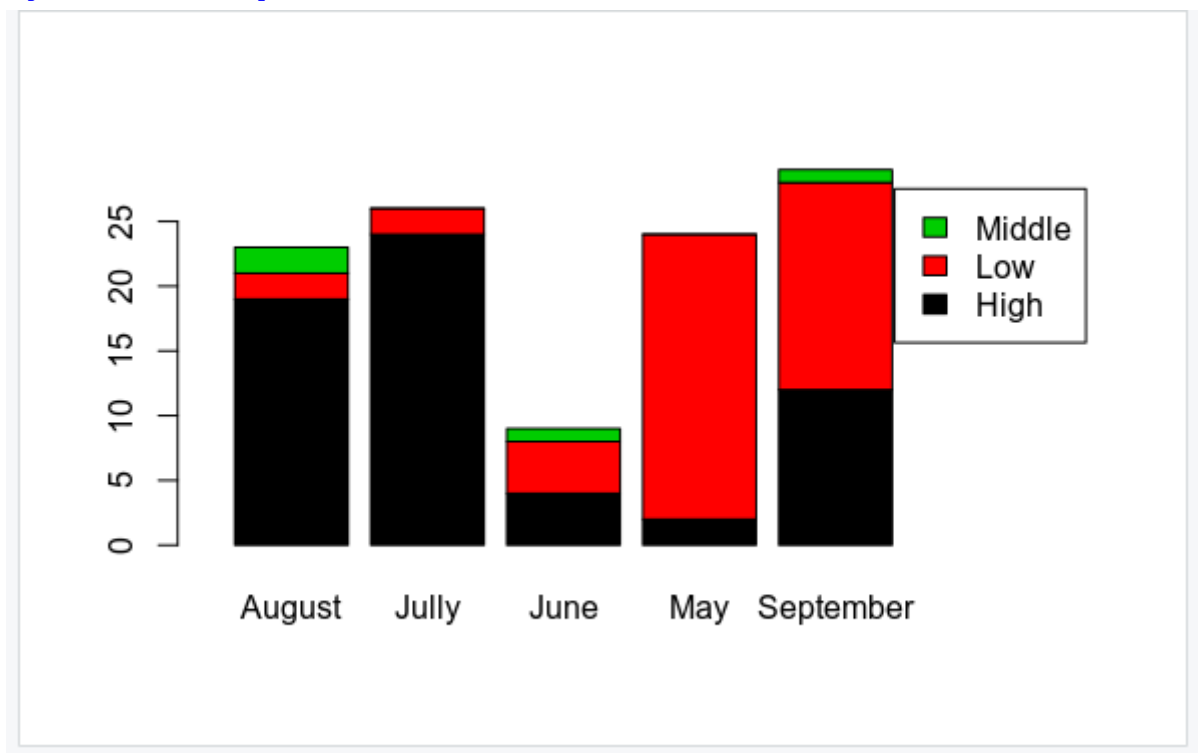
```

	TempCat		
Month	High	Low	Middle
August	0.171171171	0.018018018	0.018018018
Jully	0.216216216	0.018018018	0.000000000
June	0.036036036	0.036036036	0.009009009
May	0.018018018	0.198198198	0.000000000
September	0.108108108	0.144144144	0.009009009

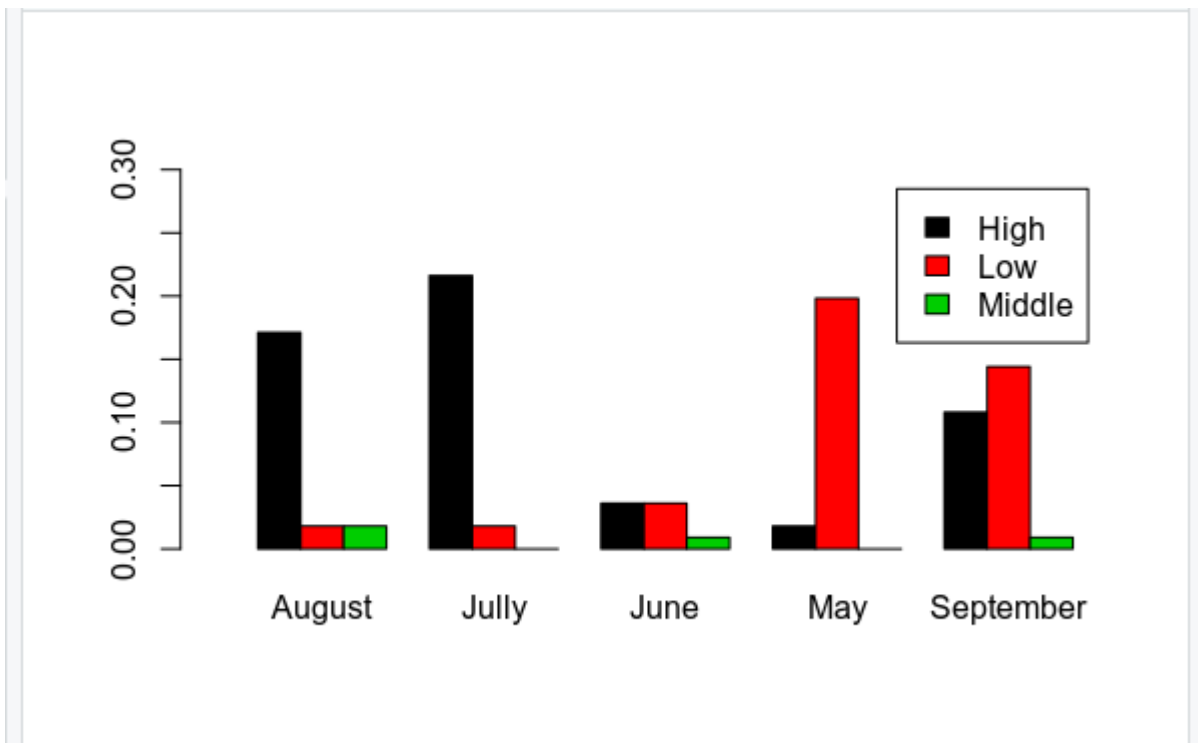
```
> table(TempCat) # πίνακας συχνοτήτων
TempCat
  High    Low Middle
    61    46     4
> prop.table(table(TempCat)) # πίνακας σχετικών συχνοτήτων
TempCat
      High      Low      Middle
0.54954955 0.41441441 0.03603604
```

7.

```
> freq_table<-table(TempCat,Month) #κατασκευή στοιβαγμένου ραβδογραμματος της
TempCat
> levels(TempCat)<-c("High","Low","Middle") # με τις στοιβες ορισμένες ως προς το
μήνα
> barplot(freq_table,width=0.66,xlim=c(0,5),
+ legend=levels(TempCat),col=1:3)
```



```
> barplot(prop.table(freq_table),width = 0.25,xlim=c(0,5),#κατασκευή
ομαδοποιημένου ραβδογραμματος
+ legend=levels(TempCat),ylim=c(0,0.3), # σχετικών συχνοτήτων της TempCat
+ beside=T,col=1:3)
```

8.

```
> Solar.R<-airquality$Solar.R #ονομαζουμε την μεταβλητη Solar.R για ευκολια
> 1+log(length(Solar.R),2) # θελουμε να ομαδοποιησουμε τα δεδομενα σε κλασεις ισου
πλατους
[1] 7.794416
> ceiling(1+log(length(Solar.R),2)) #αρα το πληθος των κλασεων θα ειναι 8
[1] 8

> range(Solar.R) # βρισουμε το ευρος των παρατηρησεων
[1] 7 334
diff(range(Solar.R)) #η διαφορα μεταξυ ζευγων διαδοχικων στοιχειων του ευρους της
Solar.R
[1] 327
> breaks<-seq(7,335,length.out = 9) #χρωιζουμε το ευρος των τιμων σε κλασεις
οριζοντας μια ακολουθια σημειων,τα οποια ισαπεχουν μεταξυ τους.Ξεκινουμε με το 7 ως
το κατω οριο και το 335 το ανω.Επισης βαζουμε μια παραπανω κλαση για να ειναι πιο
καλη οι αριθμοι
> breaks
[1] 7 48 89 130 171 212 253 294 335
> Solar.R.cut<-cut(Solar.R,breaks,right=FALSE) #ταξινομουμε τα δεδομενα στις
κλασεις
> Solar.R.freq<-table(Solar.R.cut) #υπολογιζουμε τις συχνοτητες καθε κλασεις
> Solar.R.freq
Solar.R.cut
 [7,48) [48,89) [89,130) [130,171) [171,212) [212,253) [253,294) [294,335)
      13       10        9         7        17        24        22         9
> cbind(Solar.R.freq) #μια πιο <<κομψη>> παρουσιαση της ομαδοποιησης μας
      Solar.R.freq
 [7,48)      13
 [48,89)      10
 [89,130)       9
```

```

[130,171)          7
[171,212)         17
[212,253)         24
[253,294)         22
[294,335)          9
> A<-cbind(Solar.R.freq) # Τελος η πληρης μορφη του πινακα συχνοτητων για
ομαδοποιημενα δεδομενα
> D<-cbind(A,cumsum(A),A/sum(A),cumsum(A/sum(A)))
> colnames(D)<-c("freq","cum freq","rel freq","cum rel freq")
> D

```

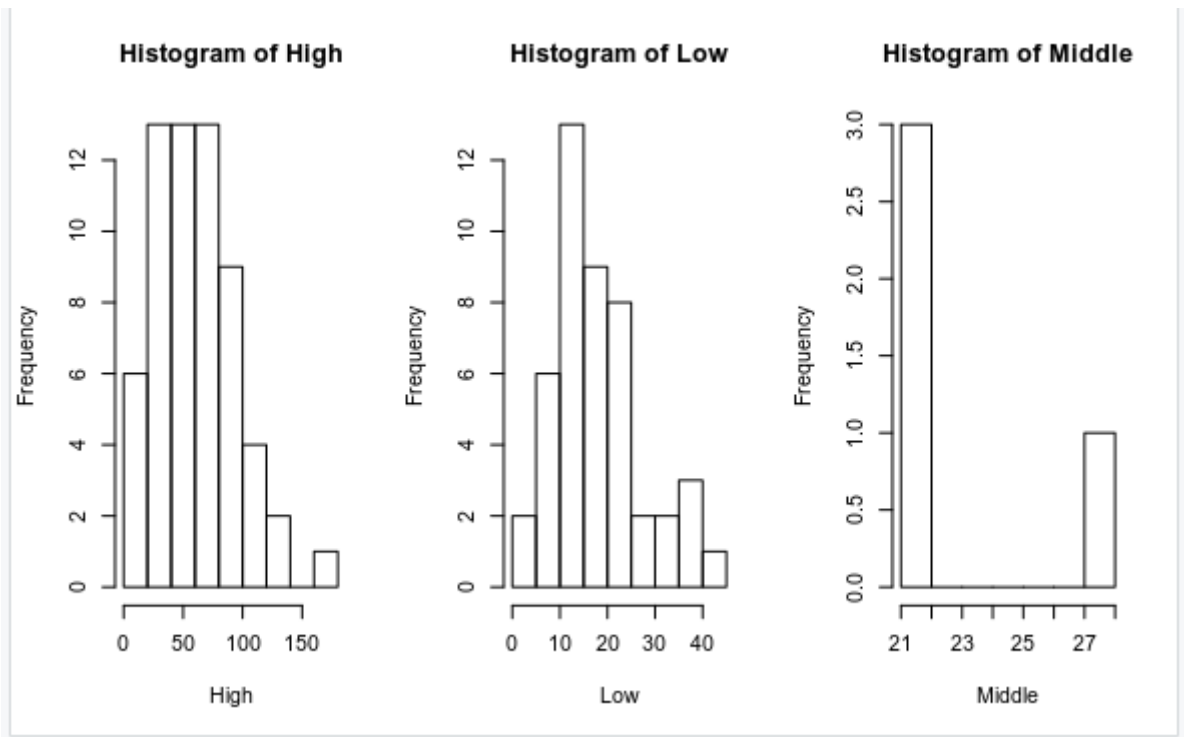
	freq	cum freq	rel freq	cum rel freq
[7,48)	13	13	0.11711712	0.1171171
[48,89)	10	23	0.09009009	0.2072072
[89,130)	9	32	0.08108108	0.2882883
[130,171)	7	39	0.06306306	0.3513514
[171,212)	17	56	0.15315315	0.5045045
[212,253)	24	80	0.21621622	0.7207207
[253,294)	22	102	0.19819820	0.9189189
[294,335)	9	111	0.08108108	1.0000000

9.

```

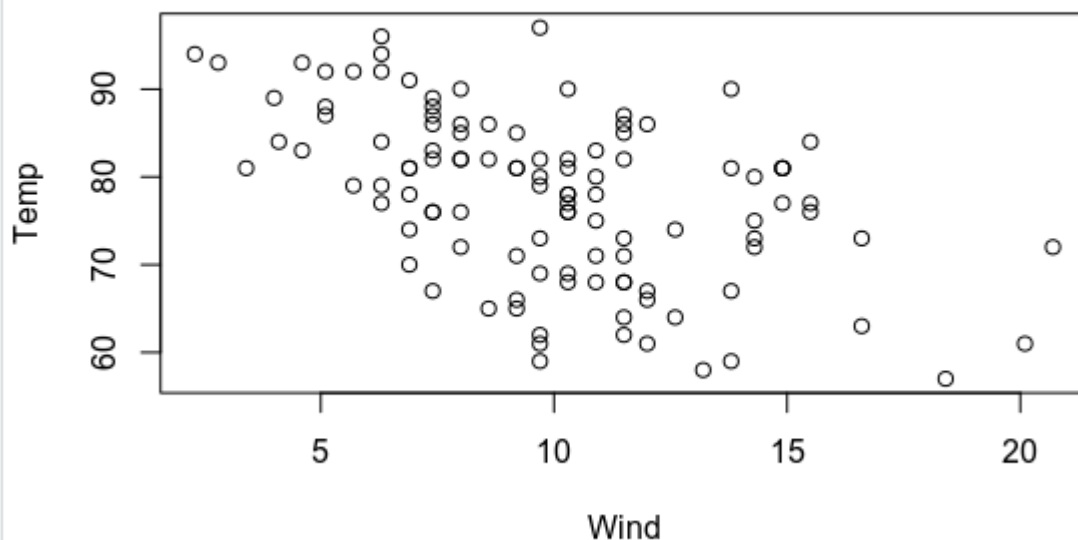
> Ozone<-airquality$Ozone #δινουμε νεο ονομα στην μεταβλητη Ozone
> 1+log(length(Ozone),2)
[1] 7.794416
> ceiling(1+log(length(airquality$Ozone),2)) #βλεπουμε οτι θα εχουμε 8 κλασεις
[1] 8
> par(mfrow=c(1,3))
> hist(Ozone[TempCat=="High"],nclass = 8,xlab="High",main="Histogram of High")
> hist(Ozone[TempCat=="Low"],nclass = 8,xlab="Low",main="Histogram of Low")
> hist(Ozone[TempCat=="Middle"],nclass = 8,xlab="Middle",main="Histogram of
Middle")
# φτιαξαμε τρια διαφορετικα διαγραμματα γιατι ειχαμε κατηγοριοποιησει σε τρεις
διαφορετικες περιπτωσης

```

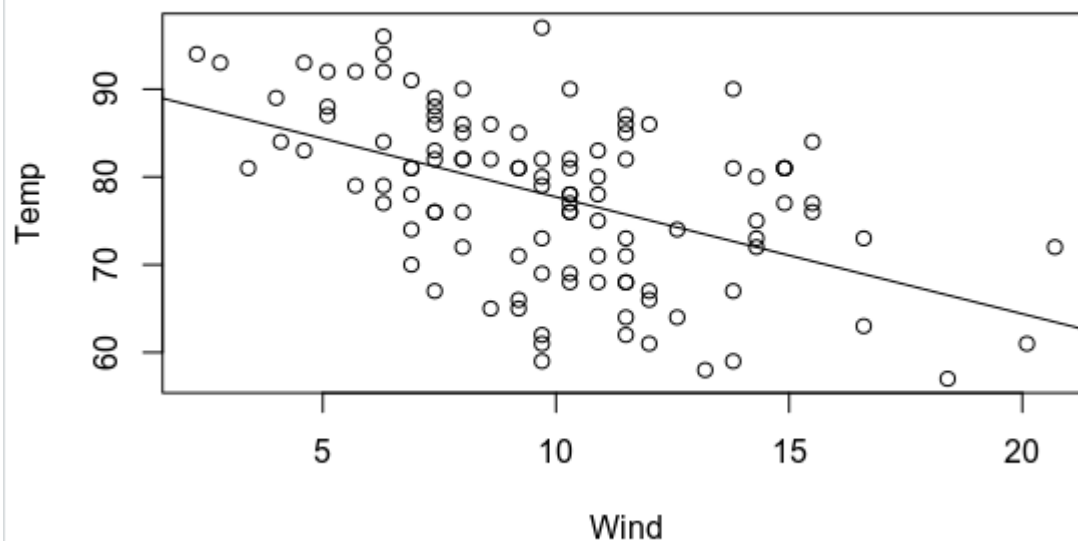


10.

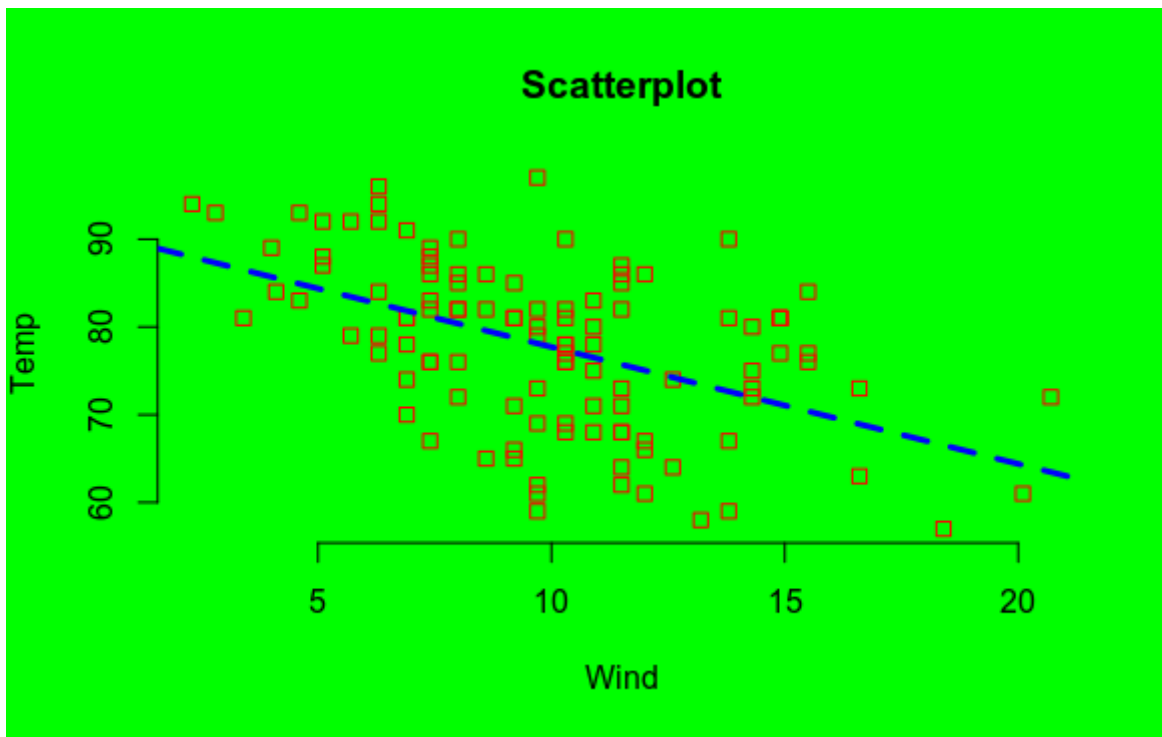
```
> xbar<-sum(airquality$Wind)/length(airquality$Wind) #υπολογισμος xbar
> xbar
[1] 9.93964
> ybar<-sum(airquality$Temp)/length(airquality$Wind) #υπολογισμος ybar
> ybar
[1] 77.79279
> b<-sum((airquality$Wind-xbar)*(airquality$Temp-ybar))/sum((airquality$Wind-
xbar)^2) #υπολογισμος του b
> b
[1] -1.331811
> a<-ybar-b*xbar #υπολογισμος του a
> a
[1] 91.03052
> plot(airquality$Wind,airquality$Temp,
+ xlab="Wind",
+ ylab="Temp",
+ ) # δημιουργια διαγραμματος διασπορας
```



```
> abline(a,b) # προσθηκη της ευθειας  $y=a+bx$ 
```



```
> par(bg="green") # χροματισαμε το διαγραμμα σε πρασινο
> plot(airquality$Wind,airquality$Temp,main = "Scatterplot", #δωσαμε καταλληλο
τιτλο στο διαγραμμα
+ xlab="Wind",ylab="Temp",pch=0,col="red", # δωσαμε χρωμα και σχημα στα σημεια
(x,y)
+ frame=FALSE)
> abline(a,b,col="blue",lty=2,lwd=3) # προσθεσαμε και την ευθεια με μπλε
χρωμα ,διακεκομμενη γραμμη και παχος μεγαλυτερο απο το συνηθισμενο
```



```
install.packages("car") Warning in install.packages : #εναλλακτικός τρόπος που το
Rstudio δεν υποστηρίζει
package 'scatterplot' is not available (for R version 3.4.4)
```

ΑΣΚΗΣΗ 2

1.

```
> Pearson<-function(x1,y1){ #κατασκευή συνάρτησης που παίρνει ως ορίσμα τις τιμές
2 διανυσμάτων
+ if (length(x1)==111) { #ελεγχος αν το μήκος είναι ίδιο με το μήκος των
μεταβλητών Wind και Temp
+ if (length(y1)==111) {
+ x1bar<-sum(x1)/length(x1)
+ y1bar<-sum(y1)/length(y1)
+ r<-sum((x1-x1bar)*(y1-y1bar))/(sum(x1-x1bar)*sum(y1-y1bar)) # υπολογισμος του r
+ return(r) } }
+ else { #αν έχουν διαφορετικό μήκος τότε δεν υπολογίζει το r
+ print("error")
+ }}
> Pearson(airquality$Wind,airquality$Temp)
[1] 5.829835e+28
> x2<-c(1,1,1,1,1) # φτιαξαμε εμεις δυο διανυσματα με διαφορετικο μήκος
> y2<-c(2,2,2,2,2)
> Pearson(x2,y2) # ωστες να φαίνεται το μήκυνα λαθους
[1] "error"
```

2.

```
> Spearman<-function(x1,y1){#δημιουργησαμε νεα συνάρτηση χρησιμοποιοντας τα ranks
και τον μέσο όρο
+ if (length(x1)==111){
+ if (length(y1)==111){
+ x1bar<-sum(x1)/length(x1)
```

```

+ y1bar<-sum(y1)/length(y1)
+ rs<-sum((rank(x1)-x1bar)*(rank(y1)-y1bar))/(sum(rank(x1)-x1bar)*sum(rank(y1)-
y1bar))
+ return(rs) } }
+ else {
+ print("error")
+ }}
> Spearman(airquality$Wind,airquality$Temp)
[1] 0.01359787
> x3<-c(1,2,3,4,5,6,7,8,9)# φτιαξαμε εμεις δυο διανυσματα με διαφορετικο μηκος
> y3<-c(9,8,7,6,5,4,3,2,1)
> Spearman(x3,y3)# ωστες να φαίνεται το μήμυνα λαθους
[1] "error"

```

3.

```

> Sinartisi<-function(x1,y1){#φτιαξαμε συναρτηση που συνενωνει τα 2 διανυσματα σε
ενα κοινο δειγμα
+ koinodeigma<-c(rank(x1),rank(y1))
+ length(koinodeigma)==length(x1)+length(y1)
+ print(koinodeigma)# η οποια δινει ranks στις τιμες στο κοινο δειγμα
+ return(sum(rank(x1)))#και επιστεφει το αθροισμα των ranks για τις παρατηρησεις
που ανηκουν στο χ
+ }
> set.seed(2) # δημιουργια 1ης εφαρμογης
> x<-rnorm(30,0,1)
> y<-rnorm(25,2,1)
> Sinartisi(x,y)
[1] 6 15 25 4 11 14 20 9 29 10 17 23 8 5 27 2 22 13 24 18 30 3 26 28 12 1
19 7 21 16
[31] 20 18 21 14 8 9 2 5 11 15 12 1 6 23 19 24 13 17 16 3 7 25 10 22 4
[1] 465
> set.seed(3) # δημιουργια 2ης εφαρμογης
> x1<-rnorm(20,10,2)
> y1<-rnorm(30,10,4)
> Sinartisi(x1,y1)
[1] 4 10 17 2 14 11 12 18 1 20 6 3 7 16 13 9 5 8 19 15 9 5 14 2 10 8
28 26 16 3
[31] 25 24 20 21 12 19 29 17 4 23 22 13 30 7 18 1 15 27 11 6
[1] 210

```

ΑΣΚΗΣΗ 3

1.

```

> fp<-function(x,lambada){#συναρτηση η ποια δεχεται ως παραμετρους εισοδου την τιμη
λ και χ
+ if (x< 0 || !(x== floor(x)) || lambada <0){#ελεγχος αν δεν ειναι φυσικος
+ print ("error") } #τοτε παιρνουμε μνμ λαθους
+ else if (x==0 && lambada>0) {# αν το χ=0 τοτε εχουμε e^λ
+ print(exp(-lambada)) } # εκτυπωση του e^λ

```

```
+ else if (x>0 && lambda>0) {# αν το  $\chi > 0$  τότε υπολογίζουμε αναδρομικά την
συναρτηση
+ z<-c()
+ z[1]<-exp(-lambda)
+ z[x]<-((exp(-lambda)*lambda^(x-1))/factorial(x-1))*(lambda/x)
+ return(z) } }
# βλέπουμε αν η συναρτηση που καναμε υπολογίζει αυτο που θελουμε
```

2.

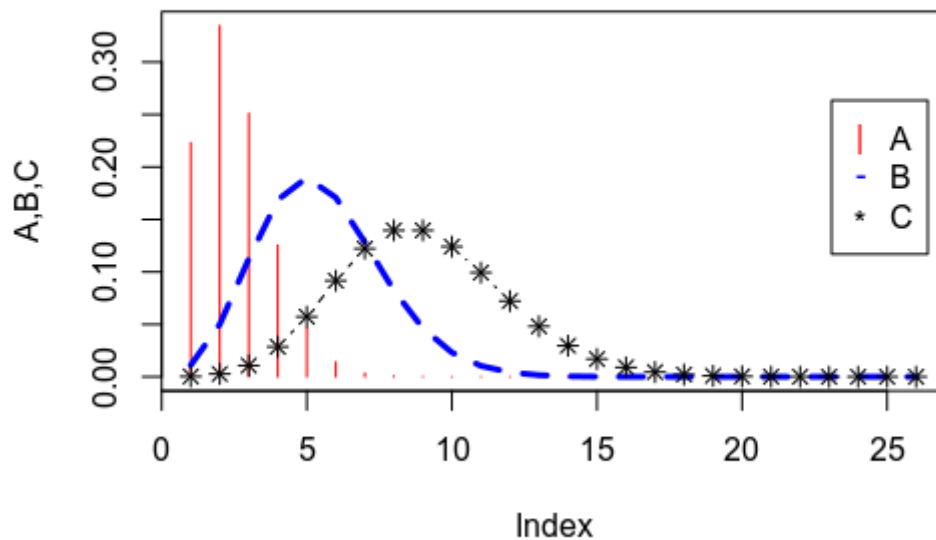
```
> A<-c(exp(-1.5),fp(1:25,1.5))#υπολογίζουμε το A
> A
[1] 2.231302e-01 3.346952e-01 2.510214e-01 1.255107e-01 4.706652e-02 1.411996e-02
3.529989e-03
[8] 7.564262e-04 1.418299e-04 2.363832e-05 3.545748e-06 4.835111e-07 6.043888e-08
6.973717e-09
[15] 7.471840e-10 7.471840e-11 7.004850e-12 6.180750e-13 5.150625e-14 4.066283e-15
3.049712e-16
[22] 2.178366e-17 1.485249e-18 9.686409e-20 6.054006e-21 3.632403e-22
```

```
> B<-c(exp(-4.5),fp(1:25,4.5))#υπολογίζουμε το B
> B
[1] 1.110900e-02 4.999048e-02 1.124786e-01 1.687179e-01 1.898076e-01 1.708269e-01
1.281201e-01
[8] 8.236295e-02 4.632916e-02 2.316458e-02 1.042406e-02 4.264389e-03 1.599146e-03
5.535504e-04
[15] 1.779269e-04 5.337808e-05 1.501258e-05 3.973919e-06 9.934798e-07 2.352979e-07
5.294202e-08
[22] 1.134472e-08 2.320511e-09 4.540129e-10 8.512742e-11 1.532294e-11
```

```
> C<-c(exp(-8),fp(1:25,8))# υπολογίζουμε το C
> C
[1] 3.354626e-04 2.683701e-03 1.073480e-02 2.862614e-02 5.725229e-02 9.160366e-02
1.221382e-01
[8] 1.395865e-01 1.395865e-01 1.240769e-01 9.926153e-02 7.219021e-02 4.812680e-02
2.961649e-02
[15] 1.692371e-02 9.025979e-03 4.512990e-03 2.123760e-03 9.438933e-04 3.974287e-04
1.589715e-04
[22] 6.056057e-05 2.202203e-05 7.659835e-06 2.553278e-06 8.170491e-07
```

```
# κατασκευάζουμε στο ίδιο διαγραμμα την γραφικη παρασταση των A,B,C
```

```
> plot(A,type="h",col="red",ylab = "(A),(B),(C)") #δινουμε κοκκινο χρωμα και
καθετη γραμμη στο A
> lines(B,col="blue",type = "l",lty=2,lwd=3)#δινουμε διακεκομμενη γραμμη πιο παχια
για το B
> lines(C,type = "b",pch=8,col="black")#δινουμε καθε σημειο να αναπαρισταται με
αστερισκο για το C
> legend(locator(1),pch = c("|","-","*"),col = c(2,4,1),legend = c("A","B","C"))
```



3.

```
> alpha<-function(lambda,tv){# συναρτηση για τον υπολογισμου του χ*
+ z<-1
+ repeat {#χρησιμοποιηουμε τον βρογχο repeat
+ sum1<-sum(fp(1:z,lambda))+exp(-lambda)
+ z<-z+1
+ if (sum1>tv) {# η οποια σταματαει οταν ξεπερασει την τιμη threshold value
+ break } }
+ y<-list(z-1,lambda,"alpha",sum1) #δημιουργουμε μια λιστα με τα αποτελεσματα που
θελουμε
+ return(y) } # και ζηταμε στην συναρτηση να μας επιστρεφει την λιστα
```

```
> alpha(8,0.9)
```

```
[[1]]
```

```
[1] 12
```

```
[[2]]
```

```
[1] 8
```

```
[[3]]
```

```
[1] "alpha"
```

```
[[4]]
```

```
[1] 0.9362028
```

```
> alpha<-function(lambda,tv){ #εναλλακτικος τροπος με χρηση της while κα if
+ z<-1
+ while(TRUE) {
```



```
+ sum1<-sum(fp(1:z,lambda))+exp(-lambda)
+ if (isTRUE(sum1>tv)) {
+ y<-list(z,lambda,"aplha",sum1)
+ return(y) }
+ else {
+ z<-z+1 }
+ } }
> alpha(8,0.9)
[[1]]
[1] 12

[[2]]
[1] 8

[[3]]
[1] "aplha"

[[4]]
[1] 0.9362028
```