Project 3

Kaushal Khatiwada

2024-04-28

WORK 1

Locate Median of "Temp" variable graphically

Find the interval size and class interval using sturge's formula.

```
aq_temp <- airquality$Temp
High <- max(aq_temp)  # max value is 97
Low <- min(aq_temp)  # min value is 56
N = length(aq_temp)  # number of observation is 153

# Calculate number of class intervals Using Sturges' formula
class_intervals <-  ceiling(log2(N) + 1) # 9 class intervals
breaks <- seq(Low,High,length.out = class_intervals)

#Frequency
f <- table(cut(aq_temp,breaks = breaks,include.lowest = TRUE))
f
```

```
##
##   [56,61.1] (61.1,66.2] (66.2,71.4] (71.4,76.5] (76.5,81.6] (81.6,86.8]
##          11          10          15          25          35          30
## (86.8,91.9]   (91.9,97]
##          15          12
```

Get frequency for less than ogive

```
less_than_ogive <- cumsum(f)
less_than_ogive
```

```
##   [56,61.1] (61.1,66.2] (66.2,71.4] (71.4,76.5] (76.5,81.6] (81.6,86.8]
##          11          21          36          61          96         126
## (86.8,91.9]   (91.9,97]
##         141         153
```

Get frequency for more than ogive

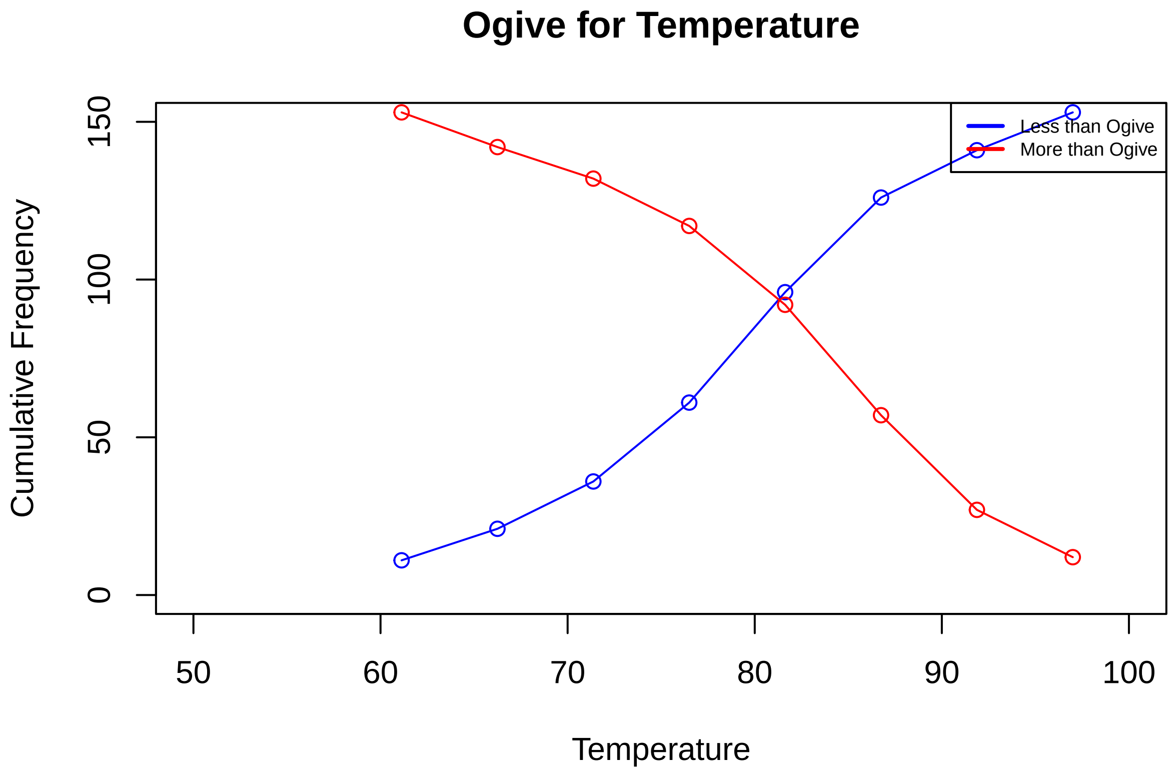
```
more_than_ogive <- rev(cumsum(rev(f)))
more_than_ogive
```

```
##   [56,61.1] (61.1,66.2] (66.2,71.4] (71.4,76.5] (76.5,81.6] (81.6,86.8]
##         153         142         132         117          92          57
## (86.8,91.9]   (91.9,97]
##          27          12
```

Plot less than and more than ogives

```
plot(breaks[-1],
     less_than_ogive,
     type = "o",
     col = "blue",
     xlab = "Temperature",
     xlim = c(50,100),
     ylim = c(0,150),
     ylab = "Cumulative Frequency",
     main = "Ogive for Temperature")
points(breaks[-1],
       more_than_ogive,
       type = "o",
       col = "red",
       ylim = c(0,150),
       xlim = c(55,100))

legend("topright", legend = c("Less than Ogive", "More than Ogive"), col = c("blue", "red"),
       lty = c(1, 1),lwd = 2,cex = 0.6)
```



Draw a line from the intersection breaks (intersection_lower, intersection_upper)

```
intersection_lower <- breaks[which(less_than_ogive >= more_than_ogive)[1]]
intersection_upper <- breaks[which(less_than_ogive >= more_than_ogive)[1]+1]
intersection_lower
```

```
## [1] 76.5
```

```
intersection_upper
```

```
## [1] 81.625
```

```
#Intersection line
```

Find the median in R

```
median <- median(aq_temp)
cat("Median is ",median)
```

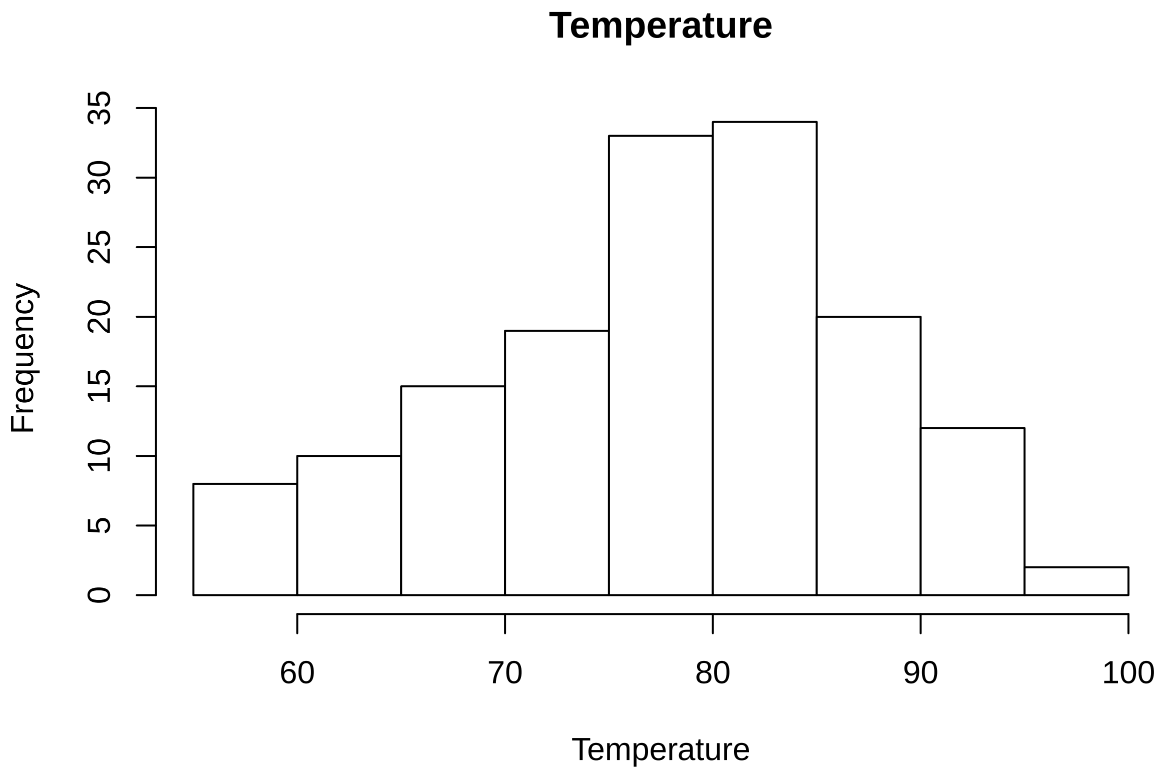
```
## Median is  79
```

WORK 2

Locate Mode of "Temp" variable graphically

Histogram of Temp variable in airquality dataset

```
aq_temp <- airquality$Temp
hist(aq_temp, main = "Temperature",col="white",xlab = "Temperature")
```



```
h <- hist(aq_temp,plot=F)
```

Get the tallest bar, left bar and right bar

```
highest_freq <- max(h$counts)       #34
highest_freq
```

```
## [1] 34
```

```
highest_bar <- which.max(h$counts)  # 6th bar
highest_bar
```

```
## [1] 6
```

```
left_bar <- highest_bar - 1         # 5th bar
left_bar
```

```
## [1] 5
```

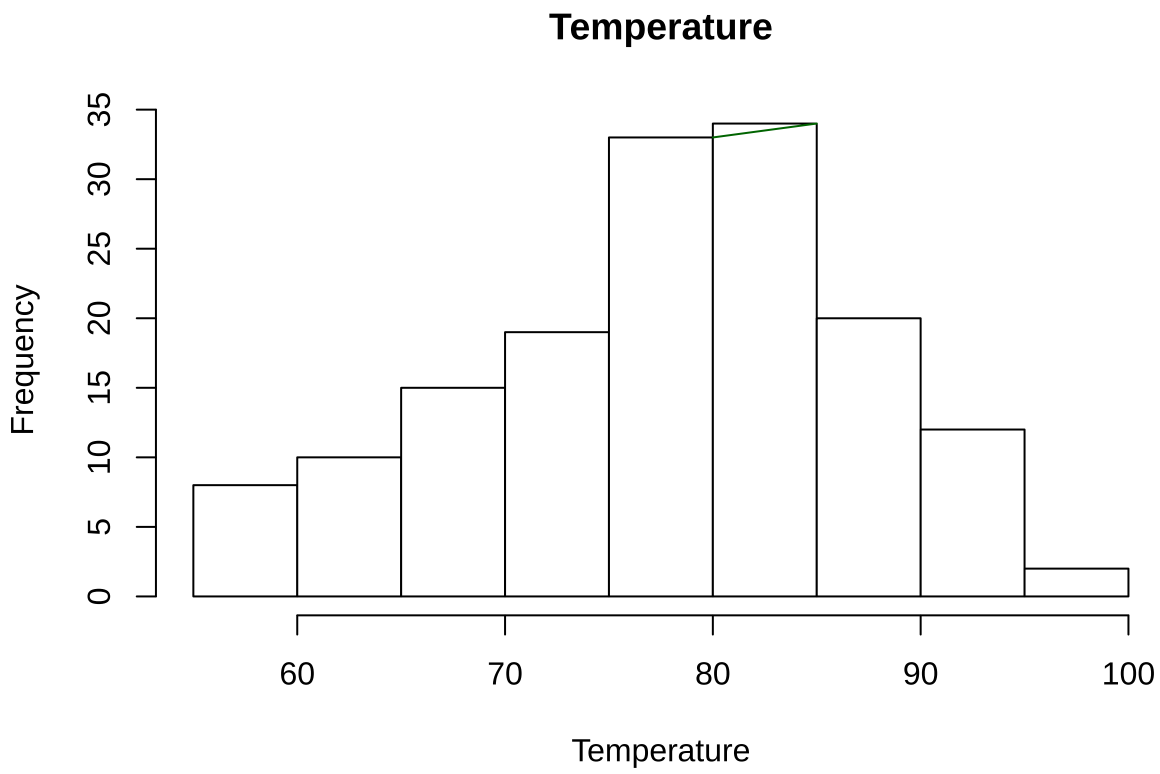
```
right_bar <- highest_bar + 1        # 7th bar
right_bar
```

```
## [1] 7
```

Draw a diagonal line from upper bound of largest bar to upper bound of left bar at top of the bar

```
left_upper <- h$breaks[left_bar + 1]
left_bar_count <- h$counts[left_bar]

highest_bar_upper <- h$breaks[highest_bar + 1]
highest_bar_count <- h$counts[highest_bar]
hist(aq_temp, main = "Temperature",col="white",xlab = "Temperature")
segments(left_upper, left_bar_count, highest_bar_upper, highest_bar_count, col = "darkgreen")
```

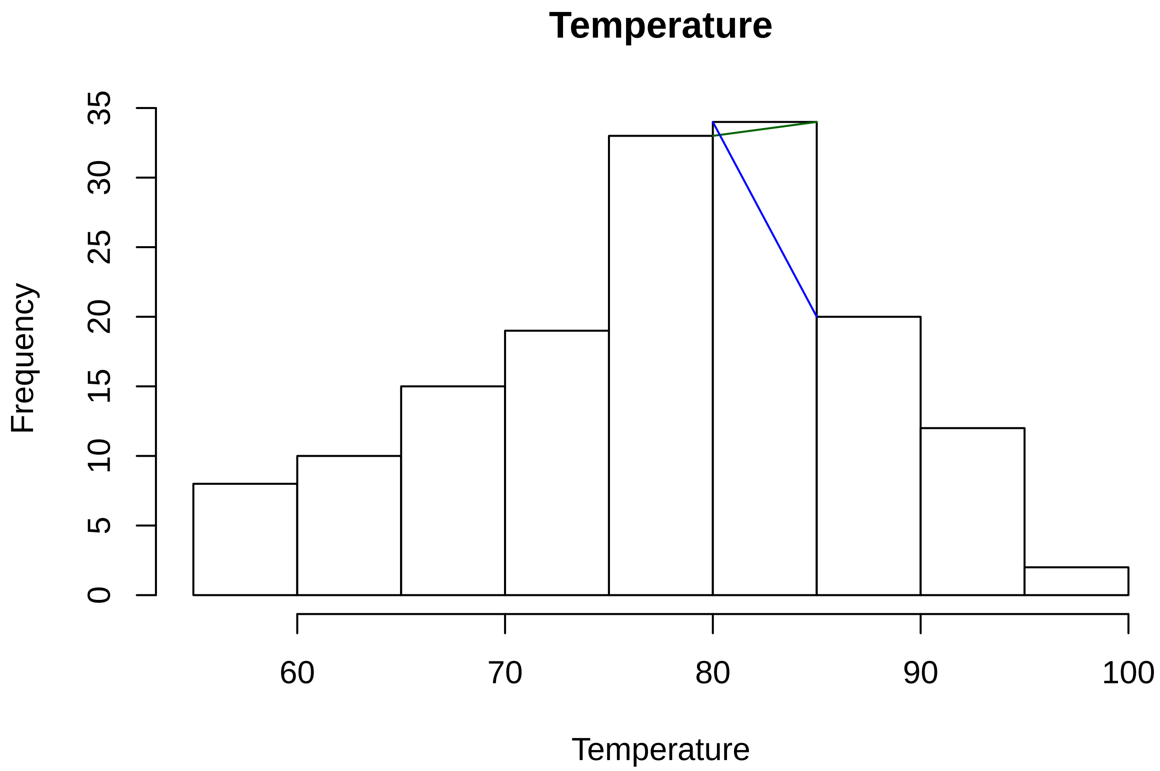


Draw a diagonal line from lower bound of largest bar to lower bound of right bar at top of the bar

```
right_lower <- h$breaks[highest_bar]
right_bar_count <- h$counts[right_bar]

highest_bar_lower <- h$breaks[right_bar]
highest_bar_count <- h$counts[highest_bar]

hist(aq_temp, main = "Temperature",col="white",xlab = "Temperature")
segments(left_upper, left_bar_count, highest_bar_upper, highest_bar_count, col = "darkgreen")
segments(right_lower, highest_bar_count, highest_bar_lower, right_bar_count, col = "blue")
```



Find the intersction point using point of intersection formula

```
# Calculate slopes (m) and y-intercepts (b) of the two lines
slope1 <- (highest_bar_count - left_bar_count) / (highest_bar_upper - left_upper)
slope2 <- (right_bar_count - highest_bar_count) / (highest_bar_lower - right_lower)

intercept1 <- left_bar_count - slope1 * left_upper
intercept2 <- highest_bar_count - slope2 * right_lower

cat("Slope of grren line",slope1)
```

```
## Slope of grren line 0.2
```

```
cat("\nSlope of blue line",slope2)
```

```
##
## Slope of blue line -2.8
```

```
cat("\nIntercept of green line",intercept1)
```

```
##
## Intercept of green line 17
```

```
cat("\nIntercept of blue line",intercept2)
```

```
##
## Intercept of blue line 258
```

Using point of intersection formula

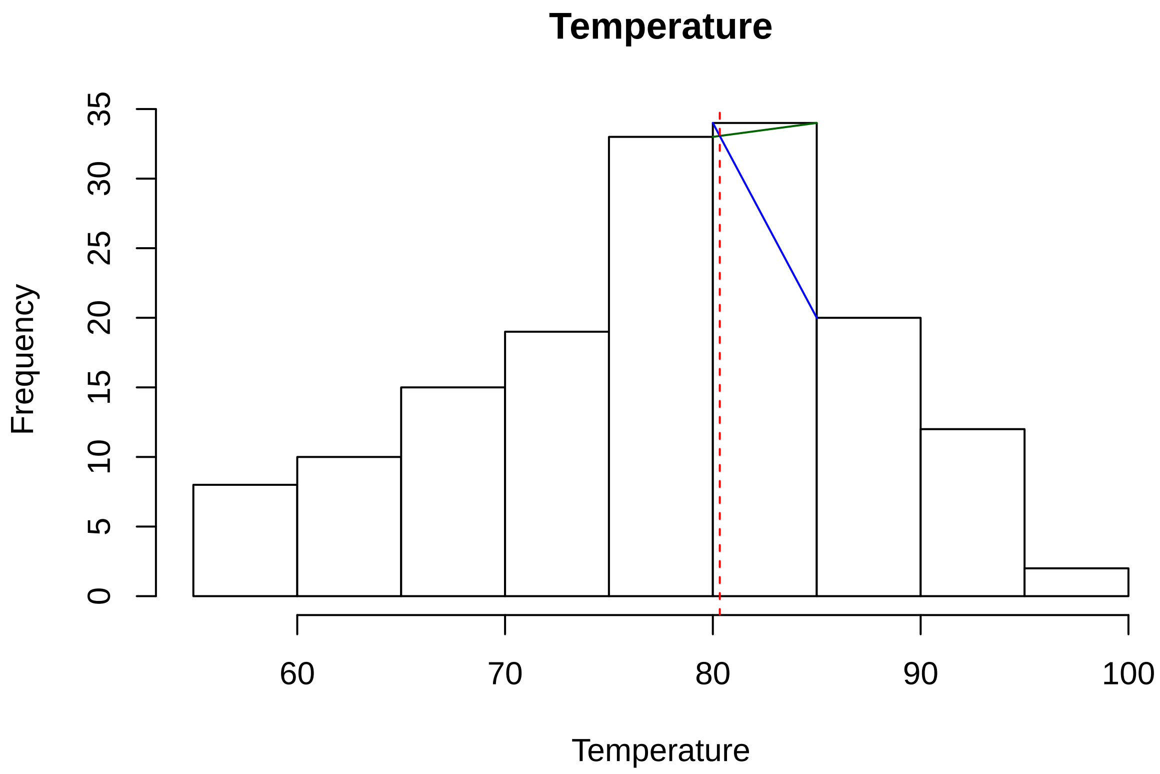
```
intersection_x <- (intercept2 - intercept1) / (slope1 - slope2)
cat("Temperature value at intesection point is ",intersection_x)  # 80.33333
```

```
## Temperature value at intesection point is  80.33333
```

So the mode is 80.333 (approx 81)

Draw a line

```
hist(aq_temp, main = "Temperature",col="white",xlab = "Temperature")
segments(left_upper, left_bar_count, highest_bar_upper, highest_bar_count, col = "darkgreen")
segments(right_lower, highest_bar_count, highest_bar_lower, right_bar_count, col = "blue")
abline(v = intersection_x, col = "red", lty = 2)
```



Mode Using R

```
f <- table(aq_temp)
max_freq <- max(f)
mode <- names(f)[f == max_freq]
cat("Mode is", mode)
```

```
## Mode is 81
```

WORK 3

Perform social network analysis of first and second variables

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.3.3
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

Get first and column column

```
sna <- read.csv("Kaushal Khatiwada - SNA_School.csv")
s <- sna[, 1:2]
```

```
net <- graph_from_data_frame(s, directed = T)
```

Number of vertices

```
V(net)   # 52 vertices
```

```
## + 52/52 vertices, named, from 499b26a:
##  [1] AA AB AF DD CD BA CB CC BC ED AE CA EB BF BB AC DC BD DB CF DF BE EA CE EE
## [26] EF FF FD GB GC GD AD KA KF LC DA EC FA FB DE FC FE GA GE KB KC KD KE LB LA
## [51] LD LE
```

Number of edges

```
E(net)   # 290 Edges
```

```
## + 290/290 edges from 499b26a (vertex names):
##   [1] AA->DD AB->DD AF->BA DD->DA CD->EC DD->CE CD->FA CD->CC BA->AF CB->CA
##  [11] CC->CA CD->CA BC->CA DD->DA ED->AD AE->AC AB->BA CD->EC CA->CC EB->CC
##  [21] BF->CE BB->CD AC->AE CC->FB DC->BB BD->CF DB->DA DD->DA DB->DD BC->AF
##  [31] CF->DE DF->BF CB->CA BE->CA EA->CA CB->CA CB->CA CC->CA CD->CA BC->CA
##  [41] BF->CA CE->CA AC->AD BD->BE AE->DF CB->DF AC->DF AA->DD AA->DD AA->DD
##  [51] CD->DD AA->DD EE->DD CD->DD DB->AA AA->FC BE->CC EF->FD CF->FE BB->DD
##  [61] CD->DD BA->AB CD->EC BE->EE CE->CC CD->CC ED->CC BB->CC BE->CE DD->CE
##  [71] AC->CD ED->CD FF->CD AC->CD DD->CD DD->CD AE->GA AE->GA AE->GA AE->GA
##  [81] BA->ED BE->ED EB->ED CD->ED FD->EF FD->EF CD->BB BF->BB BC->BB BB->CF
##  [91] AE->AC DD->DA BE->CA BE->CA CB->CA CB->CA CC->CA BE->CC BE->CC DB->DD
## + ... omitted several edges
```

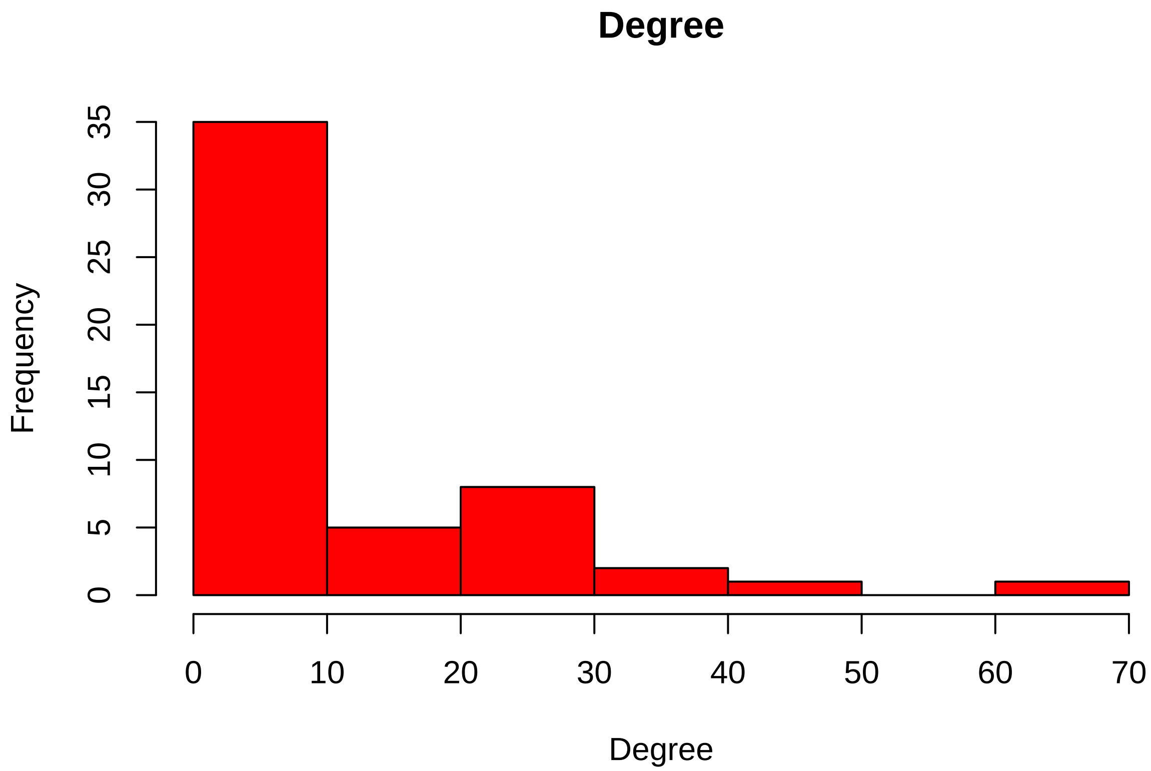
Degree of each vertices

```
degree(net)
```

```
## AA AB AF DD CD BA CB CC BC ED AE CA EB BF BB AC DC BD DB CF DF BE EA CE EE EF
## 18  9 23 36 40 26 24 50 21 27 15 62  7 12 23 27  2  4  8 12 23 20  8 10  6  8
## FF FD GB GC GD AD KA KF LC DA EC FA FB DE FC FE GA GE KB KC KD KE LB LA LD LE
##  1  8  1  1  1  9  3  3  1  7  3  1  1  2  1  2  5  1  1  1  1  1  1  1  1  1
```

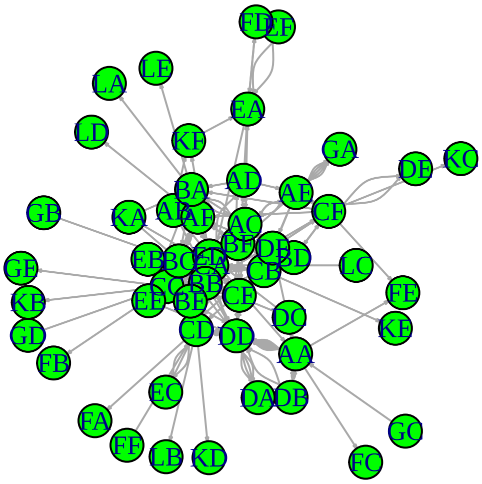
Hisotgram

```
degree_table <- table(degree(net))
set.seed(13)
hist(degree(net),
     main = "Degree",
     xlab = "Degree",
     ylab = "Frequency",
     col= "red")
```



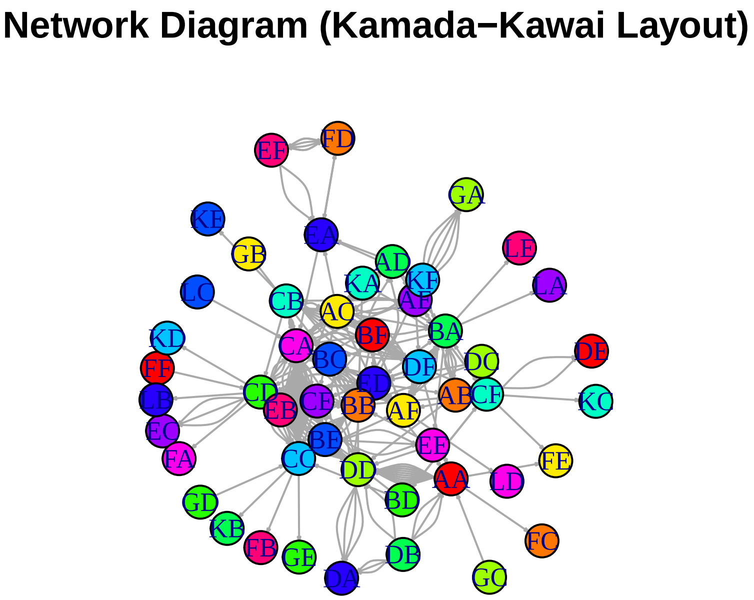
Network Diagram

```
plot(net,
     vertex.color = "green",
     edge.arrow.size = 0.1,
     vertex.label.cex = 0.8)
```



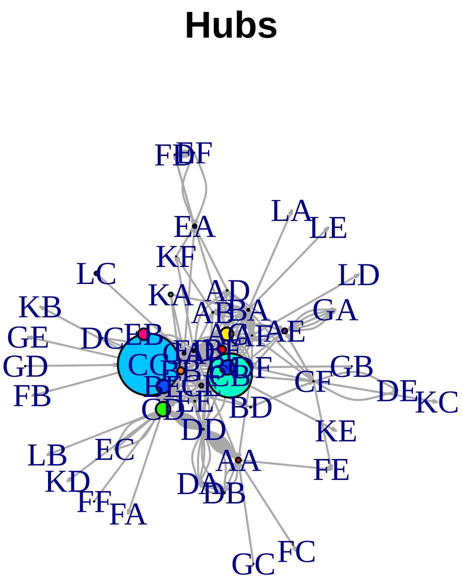
Network Diagram using Kamada-kawai layout

```r
plot(net,
     layout = layout.kamada.kawai,
     main = "Network Diagram (Kamada-Kawai Layout)",
     vertex.color = rainbow(13),
     edge.arrow.size = 0.1,
     vertex.label.cex = 0.8)
```



Hubs is a node with most outer edges

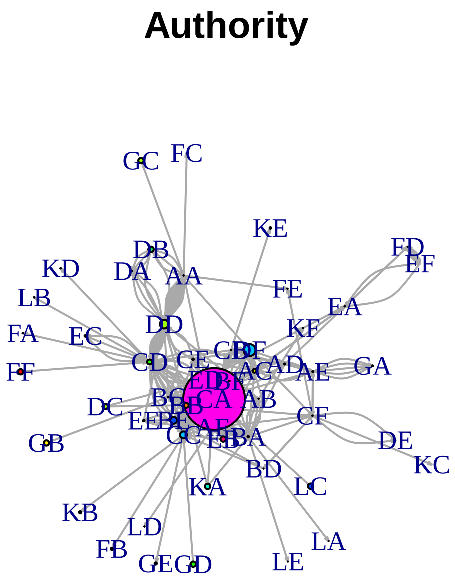
```
hubs <- hub_score(net)$vector
plot(net,
     main = "Hubs",
     vertex.color = rainbow(13),
     vertex.size = hubs * 30,
     edge.arrow.size = 0.1
     )
```



CC and CB are hubs

Authorities is a node with most inner edges

```
authorities <- authority_score(net)$vector
plot(net,
     main = "Authority",
     vertex.color = rainbow(13),
     vertex.size = authorities * 30,
     edge.arrow.size = 0.1,
     vertex.label.cex = 0.8)
```

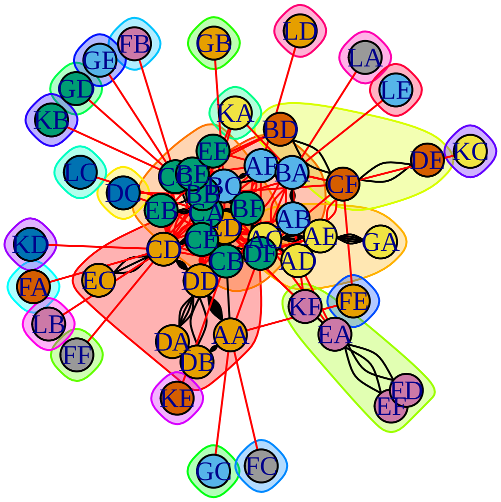


Authority

CA is authority

Community is form with densely connected noede

```
net <- graph_from_data_frame(s, directed = F)
community_cluster <- cluster_edge_betweenness(net)
plot(community_cluster,
     net,
     vetex.size = 10,
     vertex.label.cex = 0.8
)
```


LD
GB
FB
LA
GE
LE
GD
KA
KB
BD
EE
DE
KC
AF
BA
LC
CC
BE
BC
CF
DC
BB
EB
CA
BF
AB
ED
AE
GA
KD
CE
AC
CD
DF
CB
AD
EC
FA
DD
KF
FE
LB
EA
DA
AA
FF
DB
ED
KE
EF
GC
FC