trump_explore

Explore Points

Clean and save cleaned data

```
# Configure for enviroment
CLEAN_DATA = TRUE

if(CLEAN_DATA) {
  # Data clean functions
  agree_to_num=function(agree) {
    switch(as.character(agree),
           "Strongly Disagree" = as.numeric(0),
           "Disagree" = as.numeric(1/3),
           "Agree" = as.numeric(2/3),
           "Strongly Agree" = as.numeric(1))
  }
  sex_to_num=function(sex) {
    switch(as.character(sex),
           "Male" = as.numeric(0),
           "Female" = as.numeric(1))
  }
  # Simplification - assumption of average age
  age_to_num=function(age) {
    switch(as.character(age),
           "17 or younger" = as.numeric(15),
           "18-20" = as.numeric(19),
           "21-29" = as.numeric(25),
           "30-39" = as.numeric(35),
           "40-49" = as.numeric(45),
           "50-59" = as.numeric(55),
           "60 or older" = as.numeric(65))
  }

  # resp 37 Cultural Marxism does not exist
  # resp 86 no data

  # Load data
  base_df=read.csv("C:\\dev\\bayes-present\\explore\\poll_data.csv")

  # Clean data
  base_df$Sharia=sapply(base_df$Sharia, agree_to_num, simplify = TRUE)
  base_df$Marx=sapply(base_df$Marx, agree_to_num)
  base_df$SS=sapply(base_df$SS, agree_to_num)
  base_df$Medi=sapply(base_df$Medi, agree_to_num)
  base_df$ChinaWar=sapply(base_df$ChinaWar, agree_to_num)
  base_df$ChinaMoney=sapply(base_df$ChinaMoney, agree_to_num)
  base_df$Trad=sapply(base_df$Trad, agree_to_num)
  base_df$Support=sapply(base_df$Support, agree_to_num)
  base_df$Sex=sapply(base_df$Sex, sex_to_num)
  base_df$Age=sapply(base_df$Age, age_to_num)
```

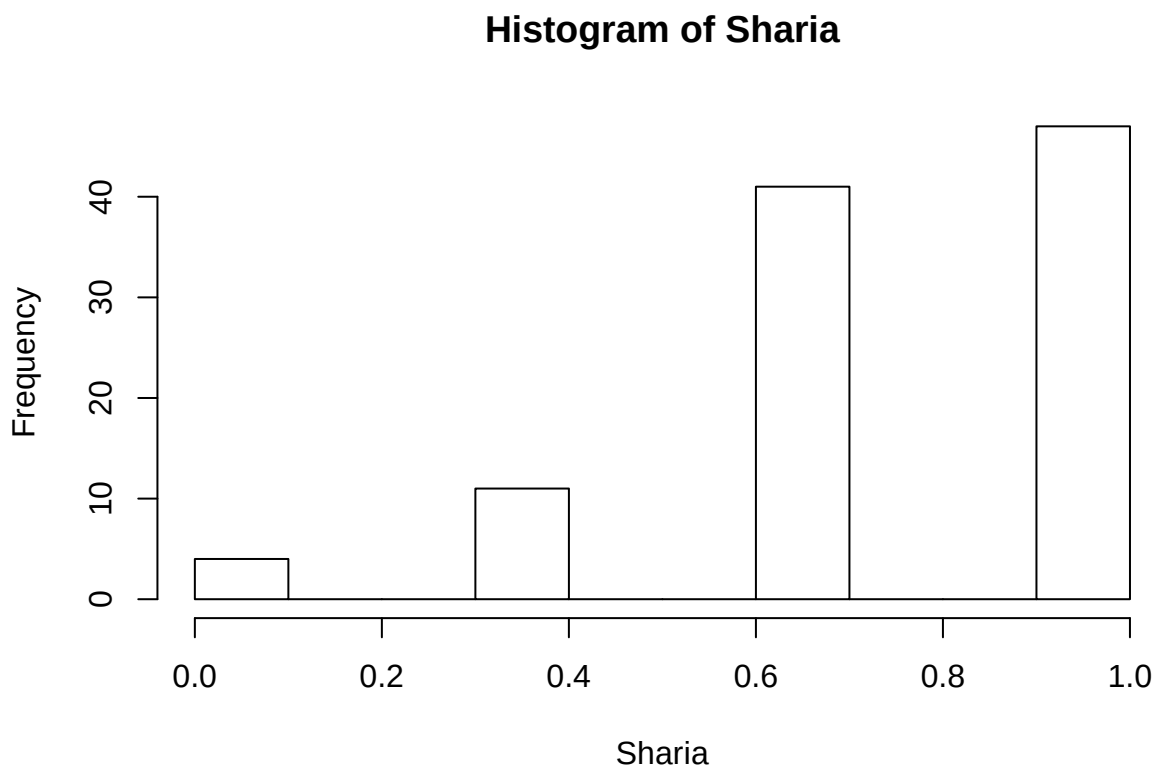
```
  # Save data
  write.csv(base_df, file="clean_data.csv", row.names=FALSE)
}
```

Load data

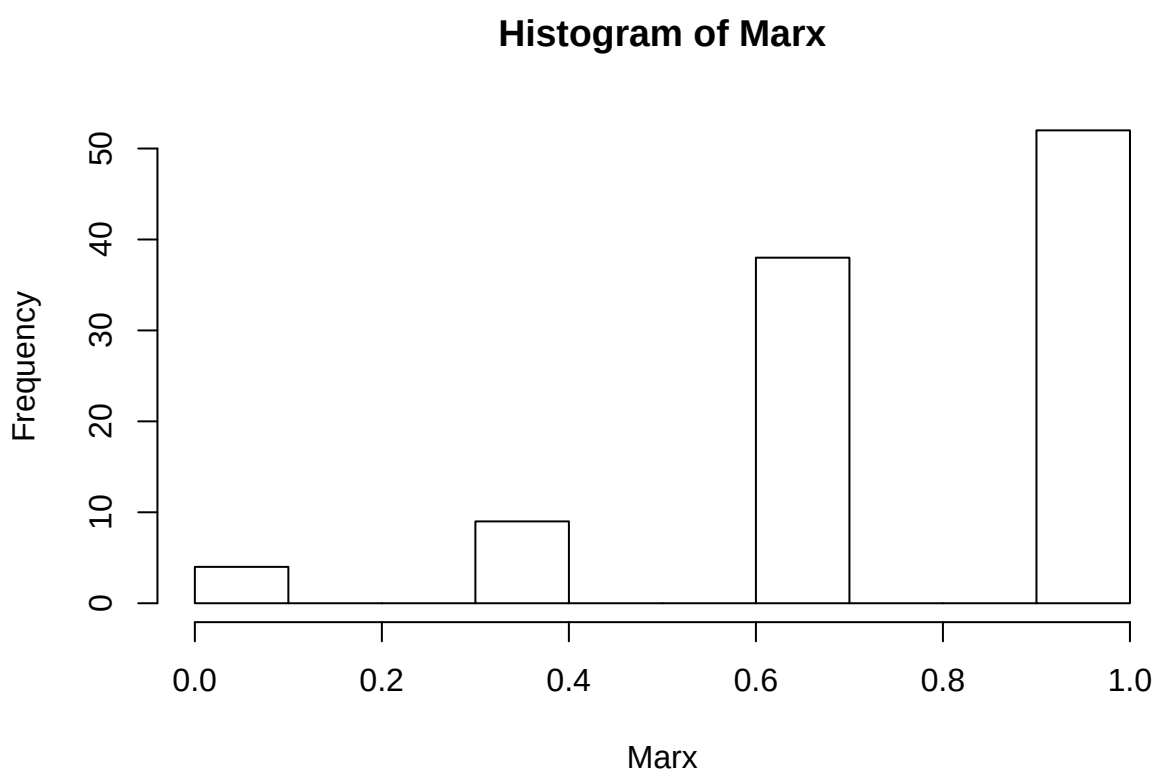
```
# Attach data

df=read.csv("C:\\dev\\bayes-present\\explore\\clean_data.csv")
attach(df)
```

Frequency of responses

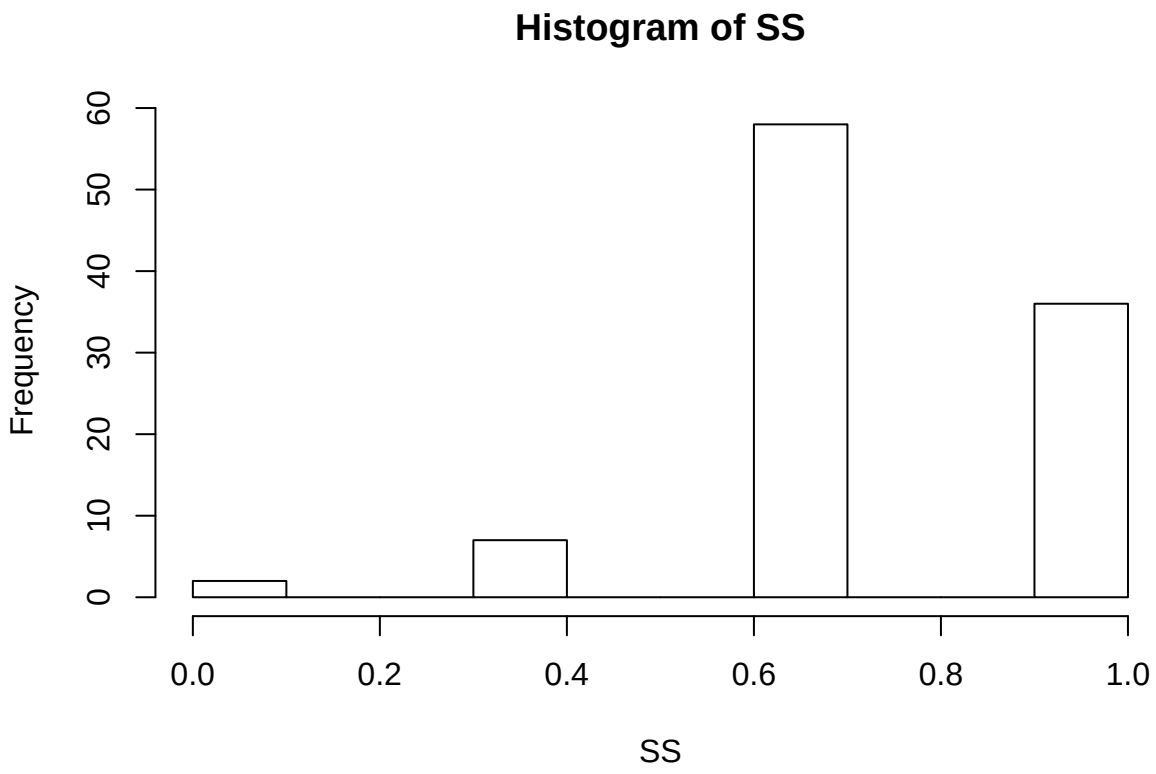
```
hist(Sharia)
```



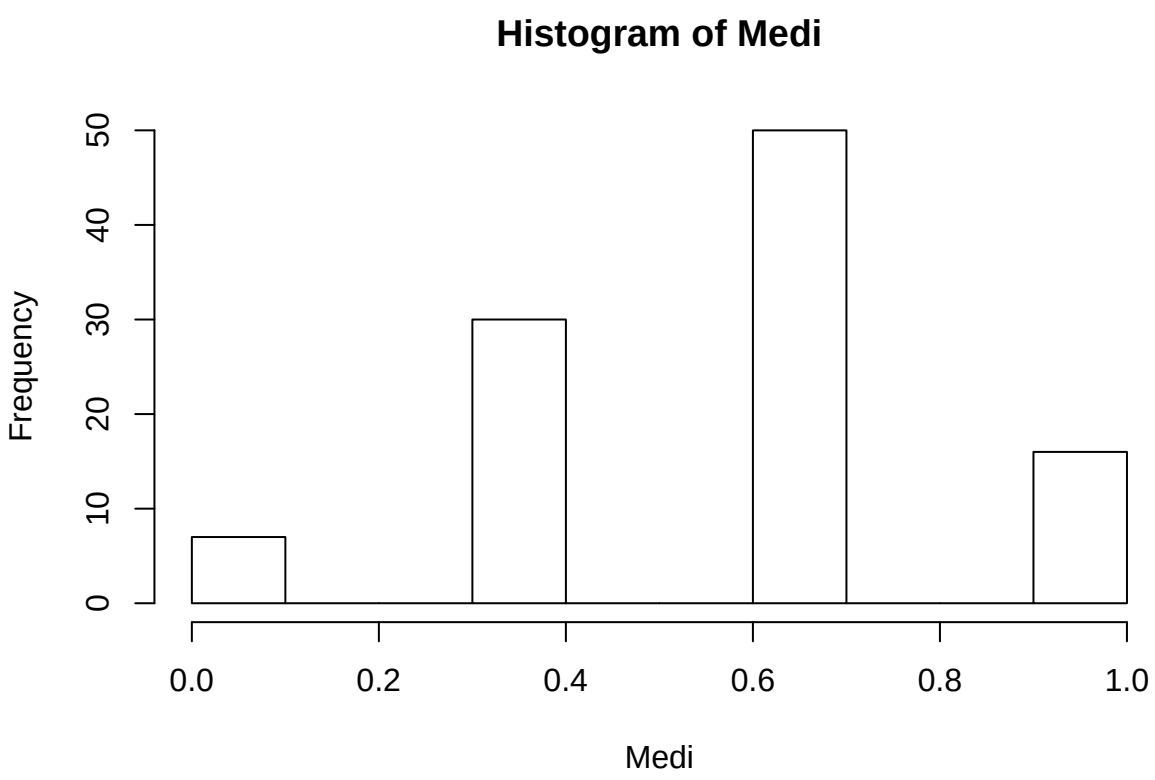

```
hist(Marx)
```


Histogram of Marx
Frequency
0
10
20
30
40
50
0.0
0.2
0.4
0.6
0.8
1.0
Marx

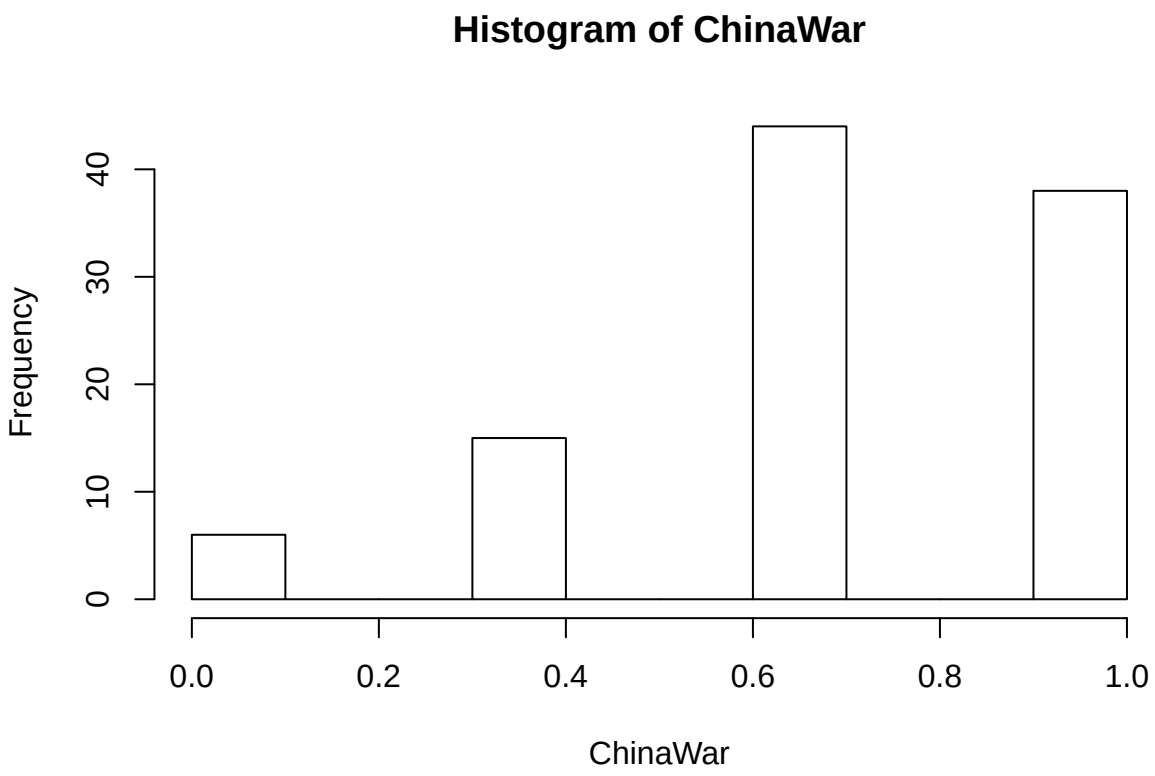
```
hist(SS)
```



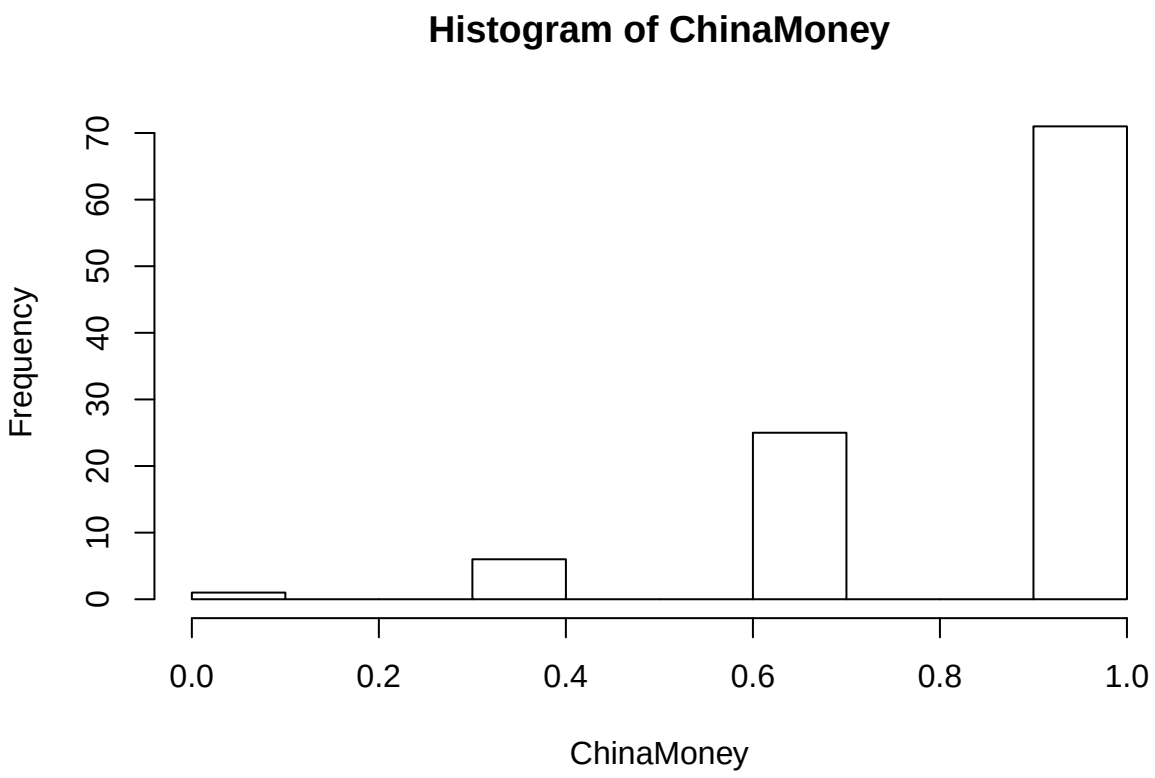

```
hist(Medi)
```



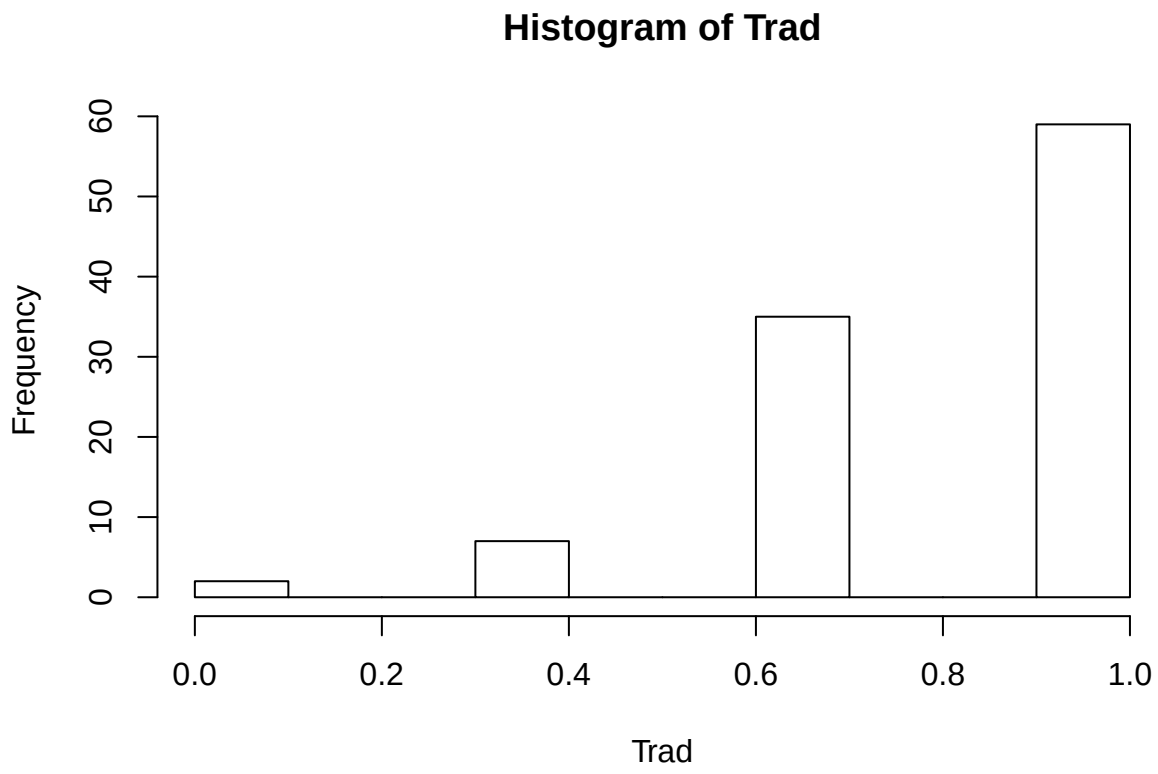

```
hist(ChinaWar)
```



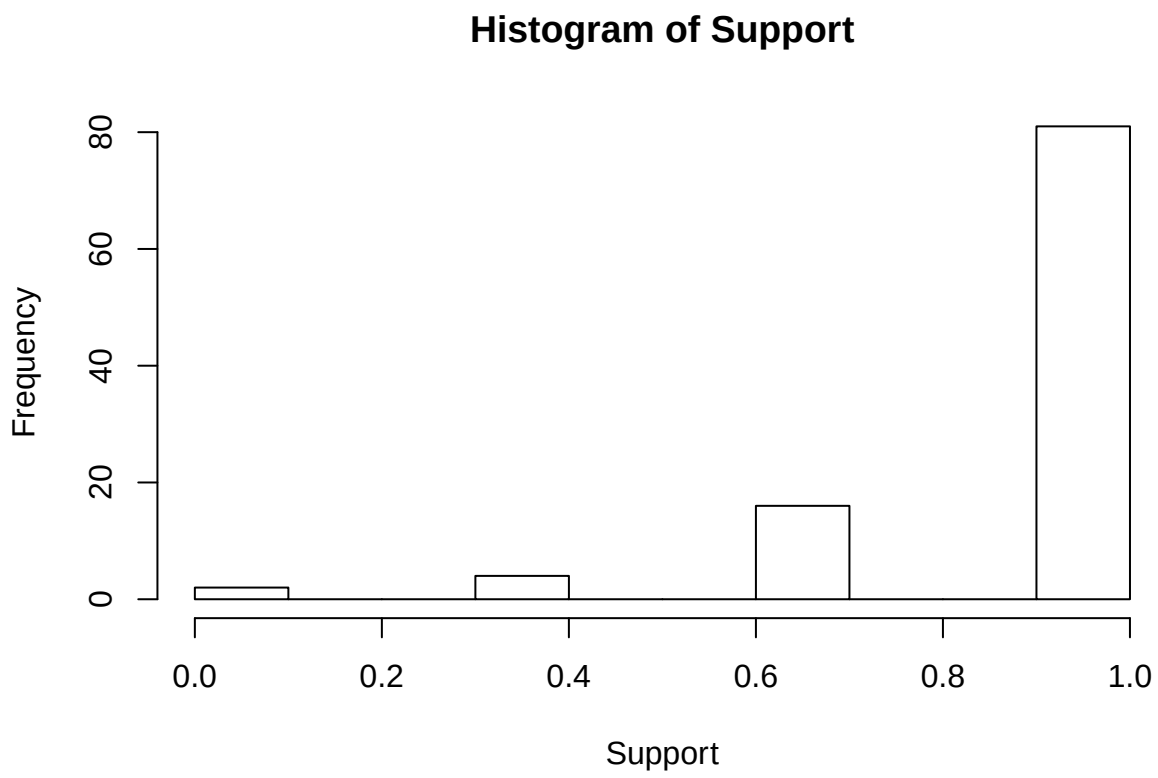

```
hist(ChinaMoney)
```


Histogram of ChinaMoney
Frequency
0
10
20
30
40
50
60
70
0.0
0.2
0.4
0.6
0.8
1.0
ChinaMoney

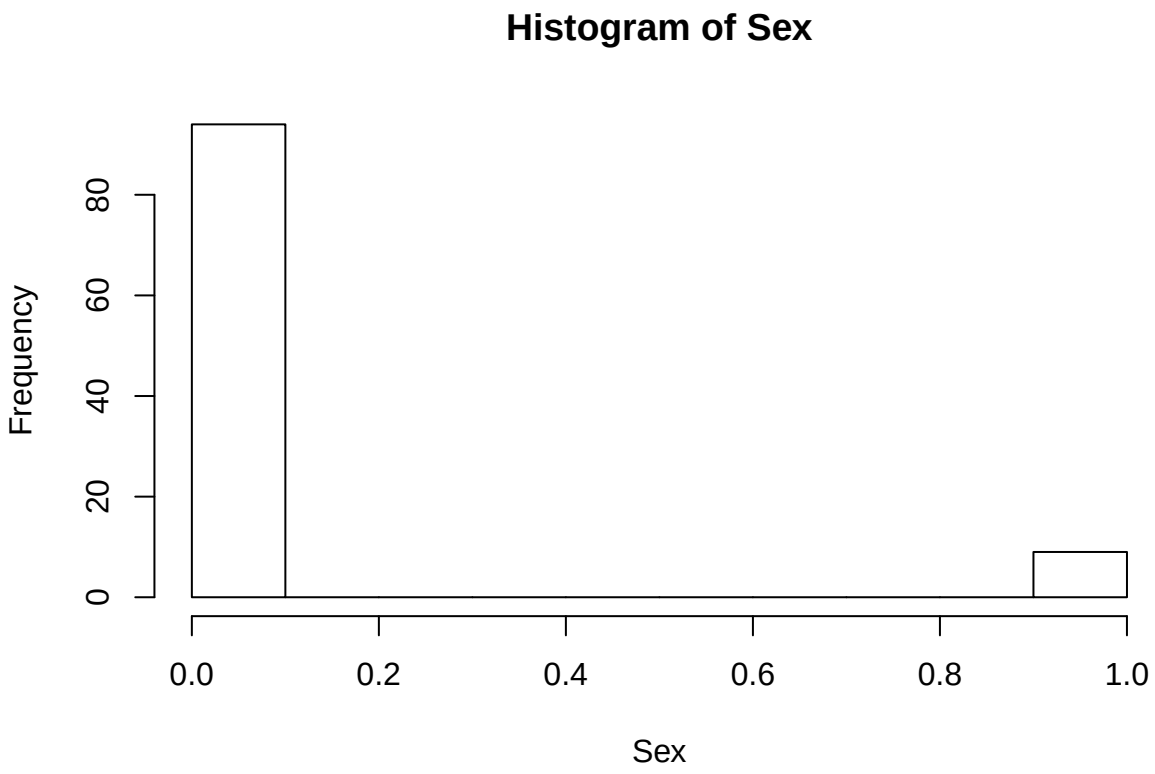
```
hist(Trad)
```



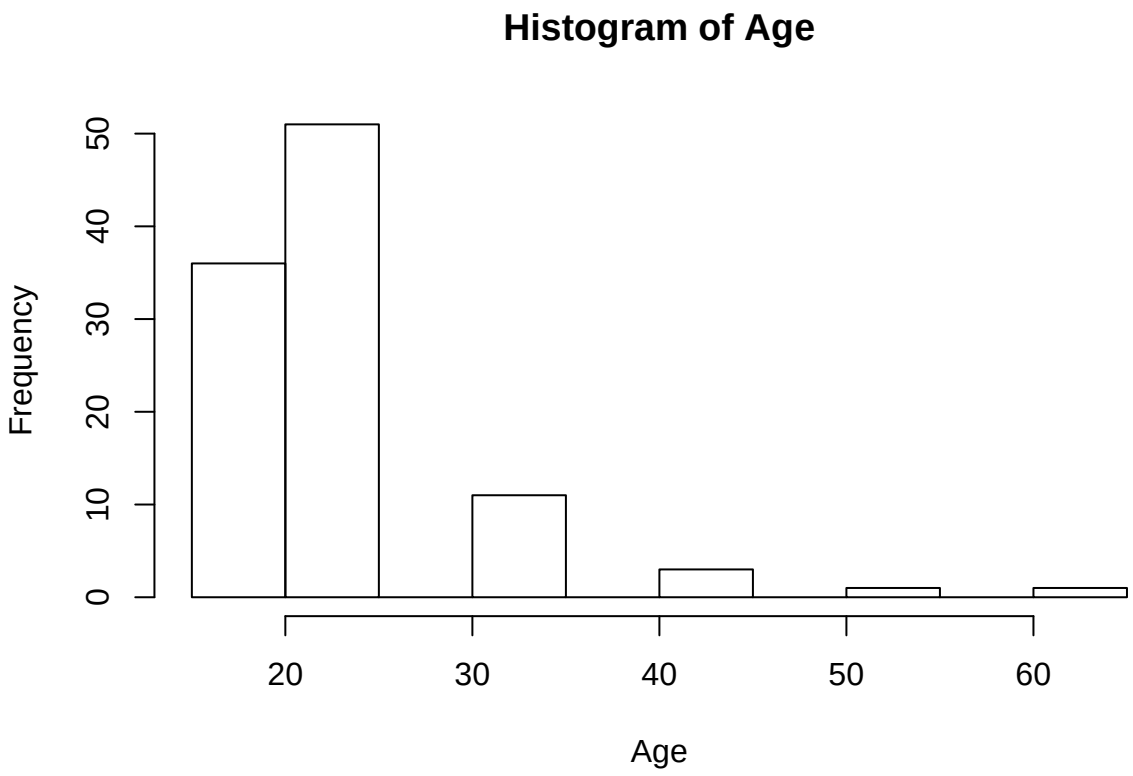

```
hist(Support)
```


Histogram of Support
Frequency
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
Support

```
hist(Sex)
```




```
hist(Age)
```

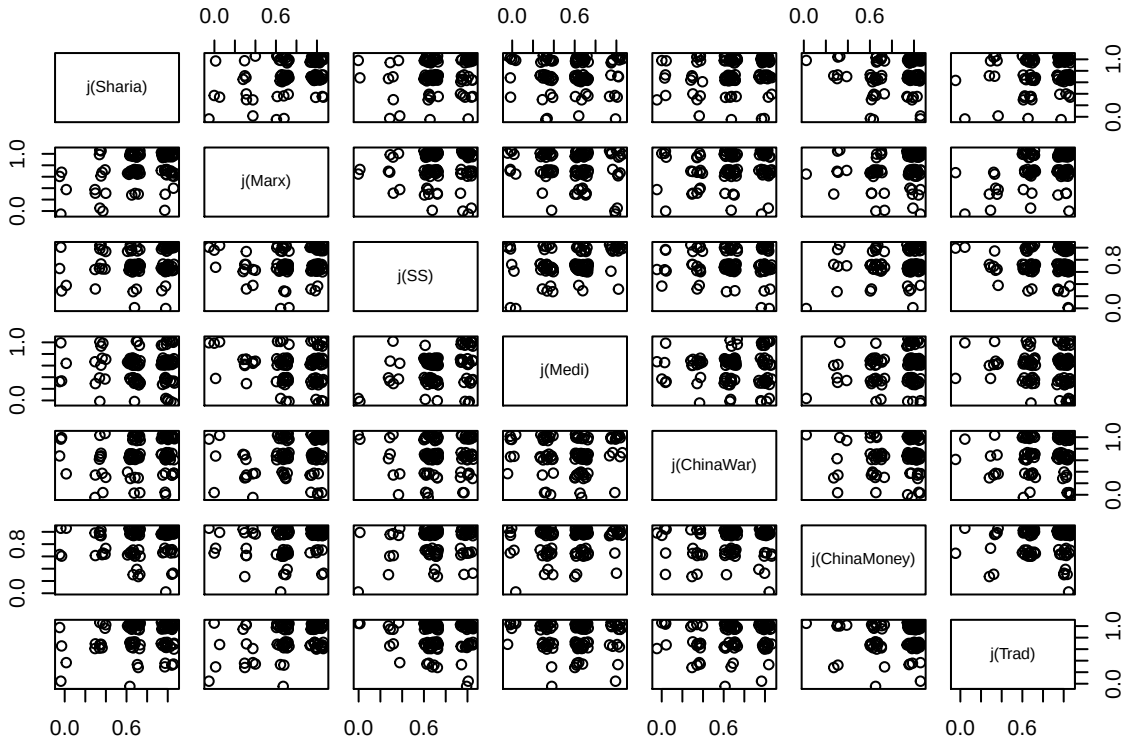


Scatterplot comparison of reponses

```
library(car)
```

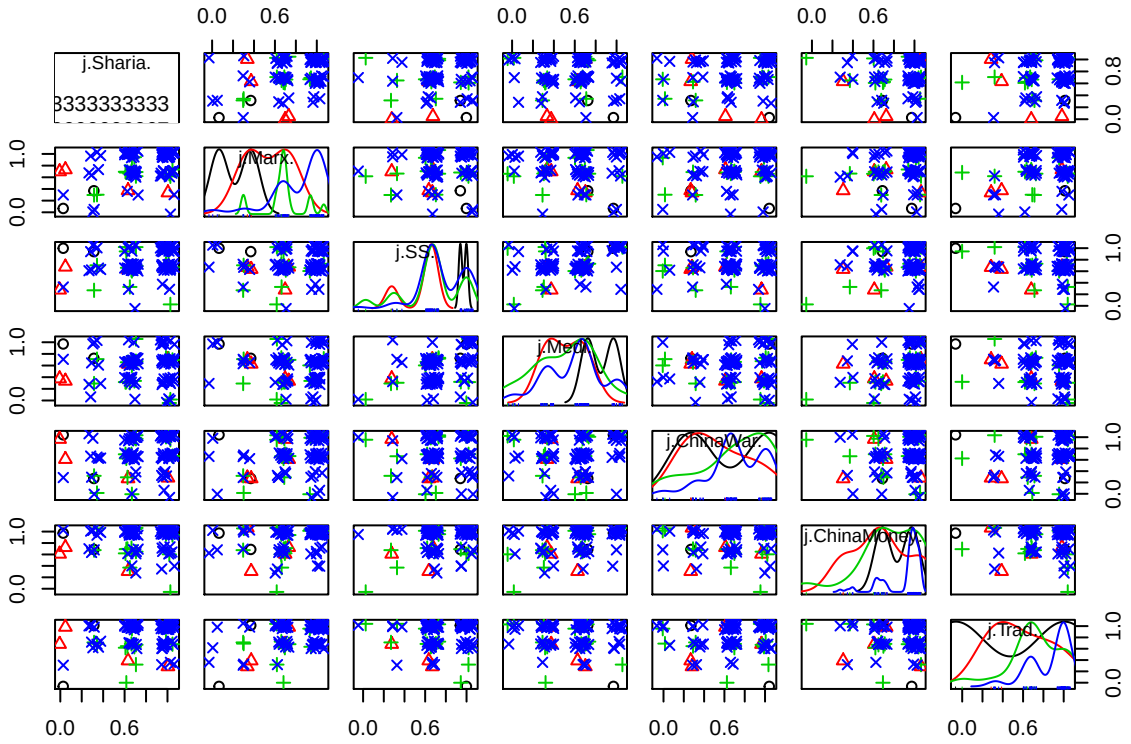
```
## Warning: package 'car' was built under R version 3.2.5
```

```
j=function(s) {
  return(jitter(s, factor = 1))
}

pairs(~j(Sharia)+j(Marx)+j(SS)+j(Medi)+j(ChinaWar)+j(ChinaMoney)+j(Trad), data=df)
```



In terms of Support

```
scatterplotMatrix(~j(Sharia)+j(Marx)+j(SS)+j(Medi)+j(ChinaWar)+j(ChinaMoney)+j(Trad)|Support, data=df,
                  reg.line=F, spread=FALSE, lwd=0, smoother=NULL)
```

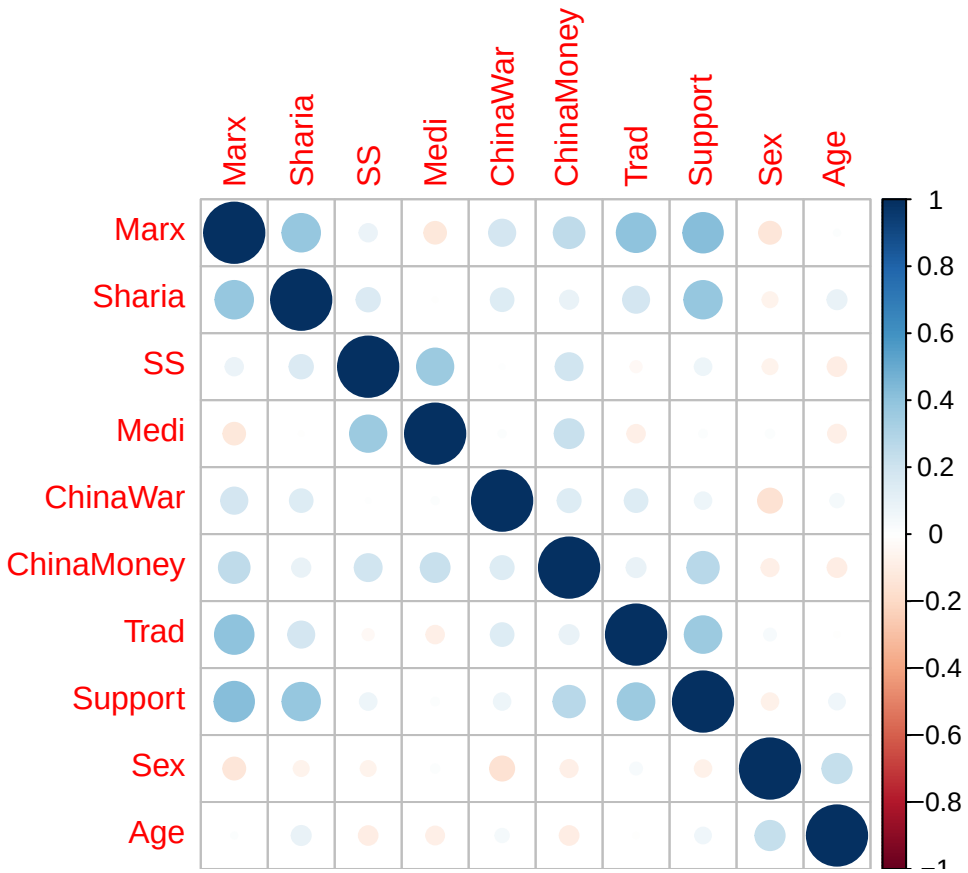


Correlation Visualization

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.2.5
```

```
M <- cor(df)
corrplot(M, method="circle")
```

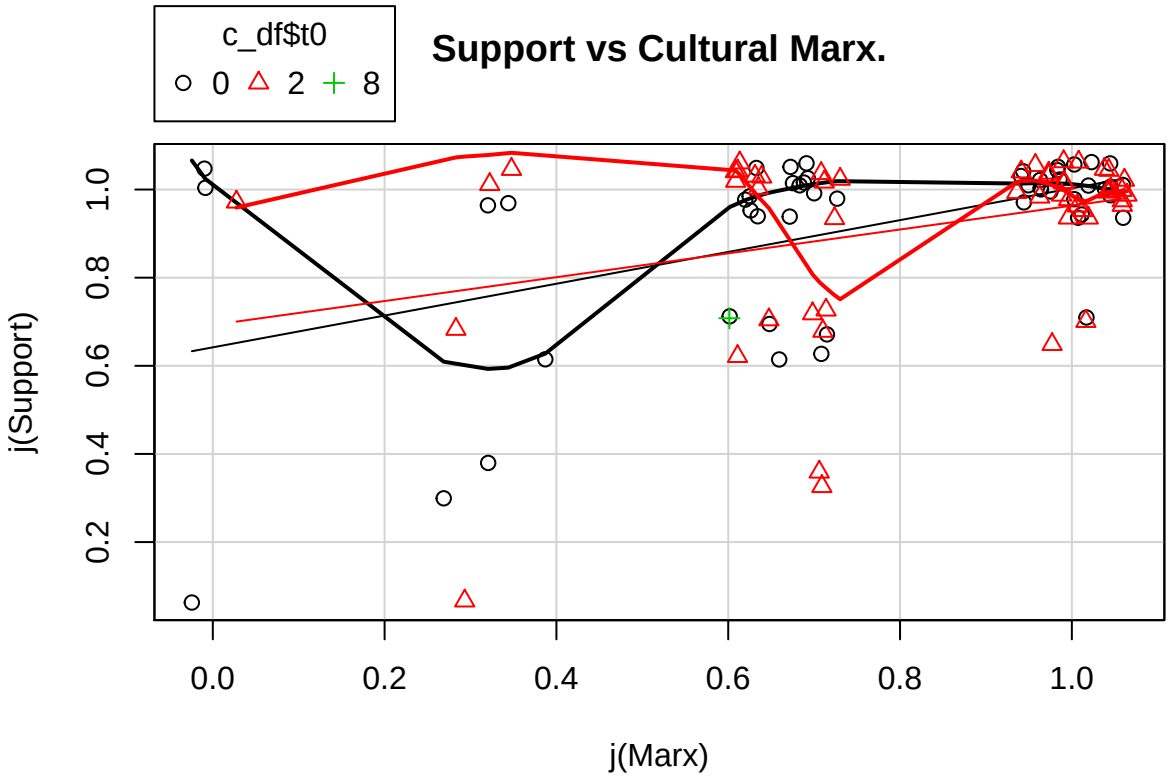


Load clustering results from classify_people.py

```
c_df=read.csv("C:\\dev\\bayes-present\\explore\\cluster_data_25.csv")
```

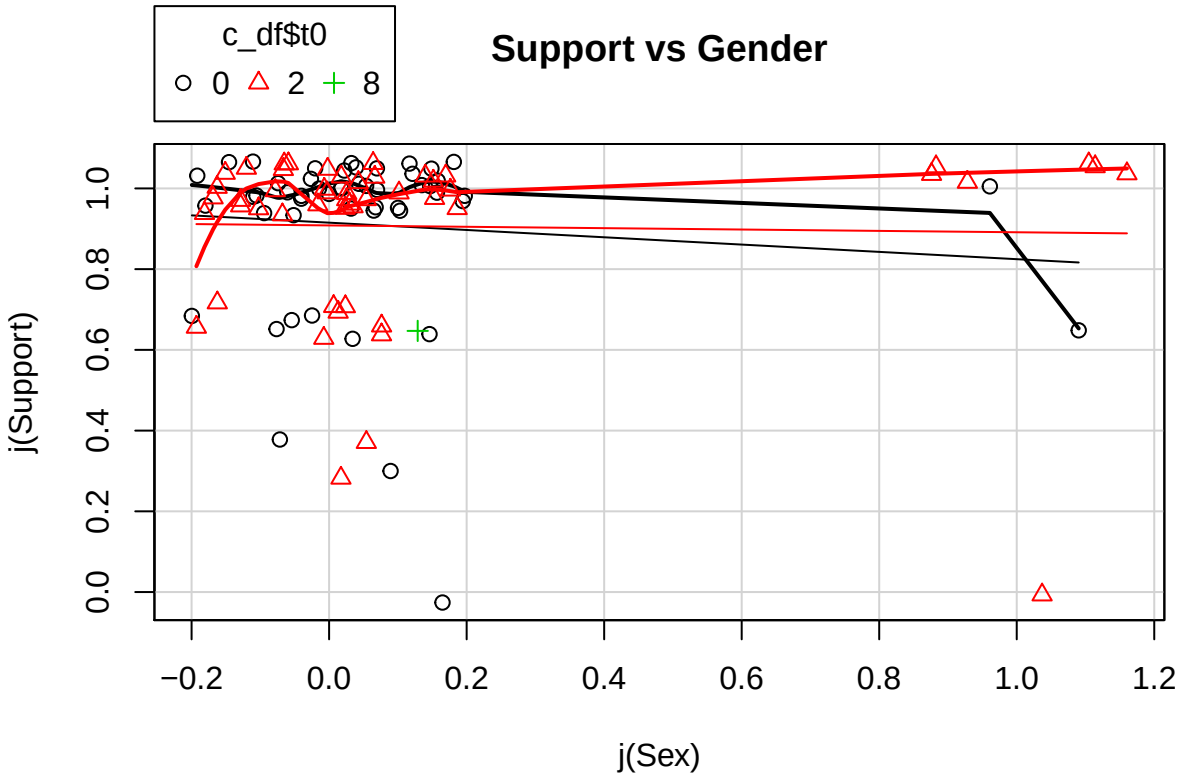
```
scatterplot(j(Support) ~  j(Marx)| c_df$t0, data=df,
main="Support vs Cultural Marx.")
```

```
## Warning in smoother(.x[subs], .y[subs], col = col[i], log.x =
## logged("x"), : could not fit smooth
```



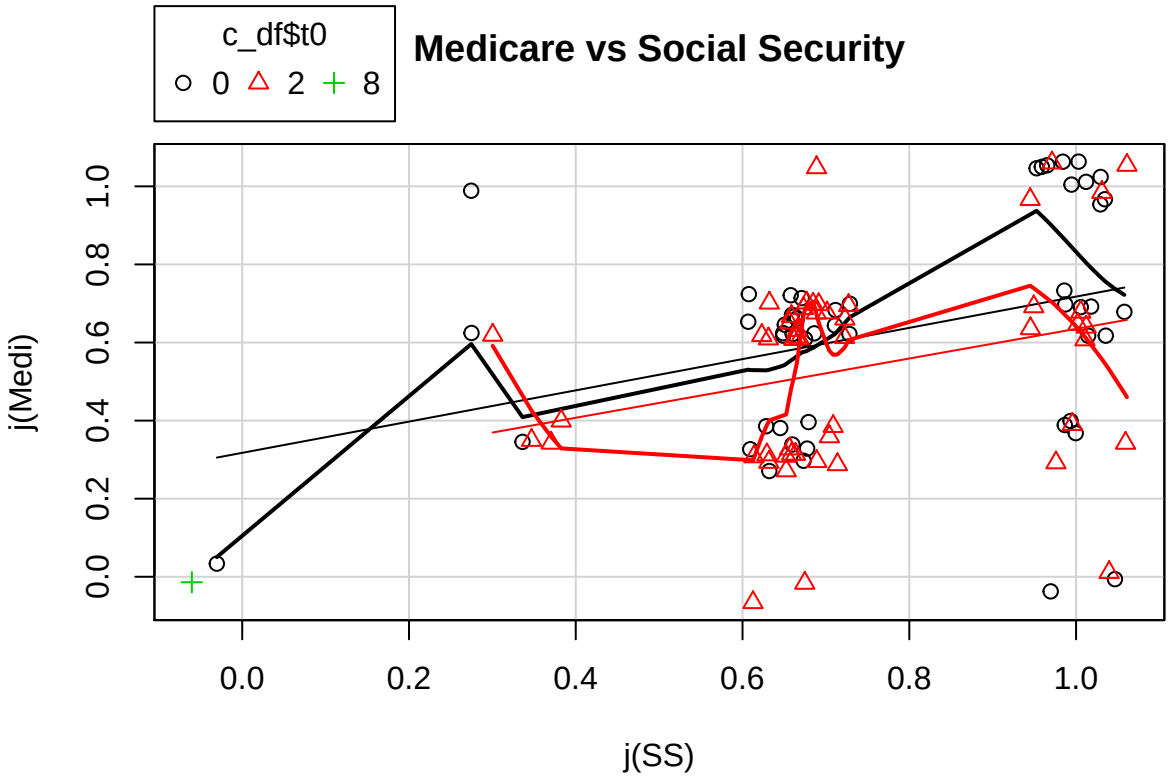

```
scatterplot(j(Support) ~  j(Sex)| c_df$t0, data=df,
            main="Support vs Gender")
```

```
## Warning in smoother(.x[subs], .y[subs], col = col[i], log.x =
## logged("x"), : could not fit smooth
```



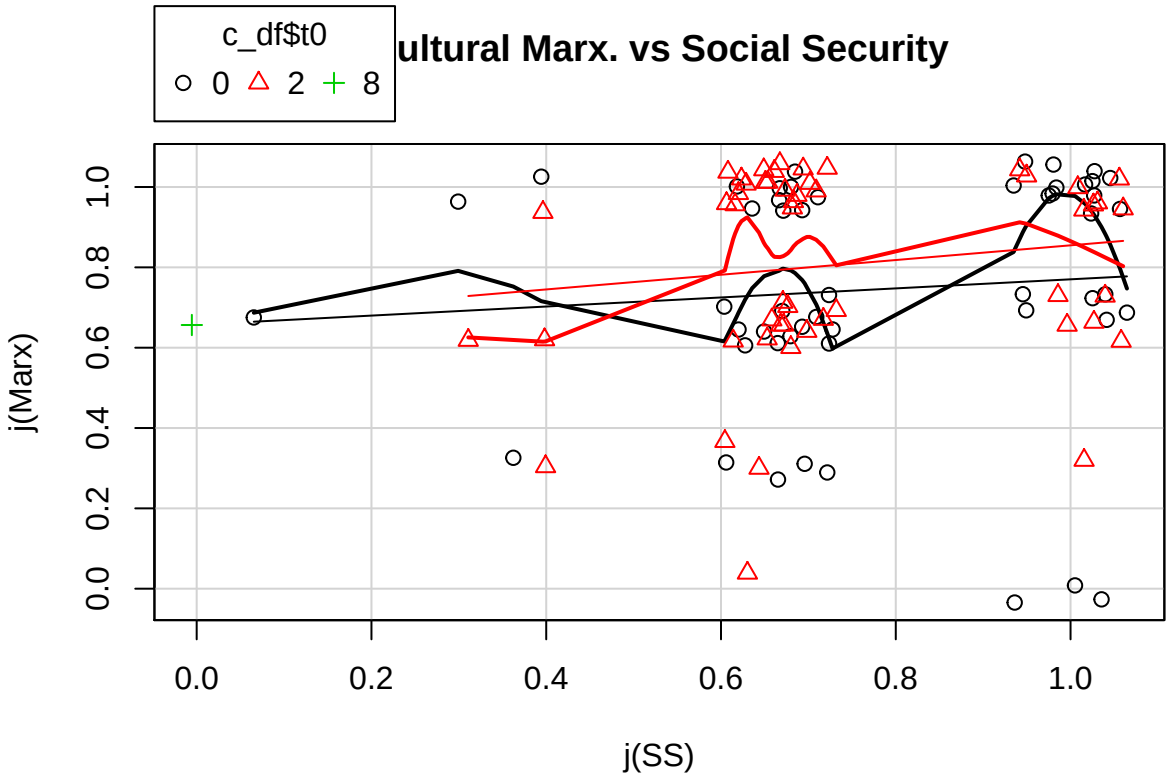

```
scatterplot(j(Medi) ~  j(SS)| c_df$t0, data=df,
            main="Medicare vs Social Security")
```

```
## Warning in smoother(.x[subs], .y[subs], col = col[i], log.x =
## logged("x"), : could not fit smooth
```




```
scatterplot(j(Marx) ~  j(SS)| c_df$t0, data=df,
            main="Cultural Marx. vs Social Security")
```

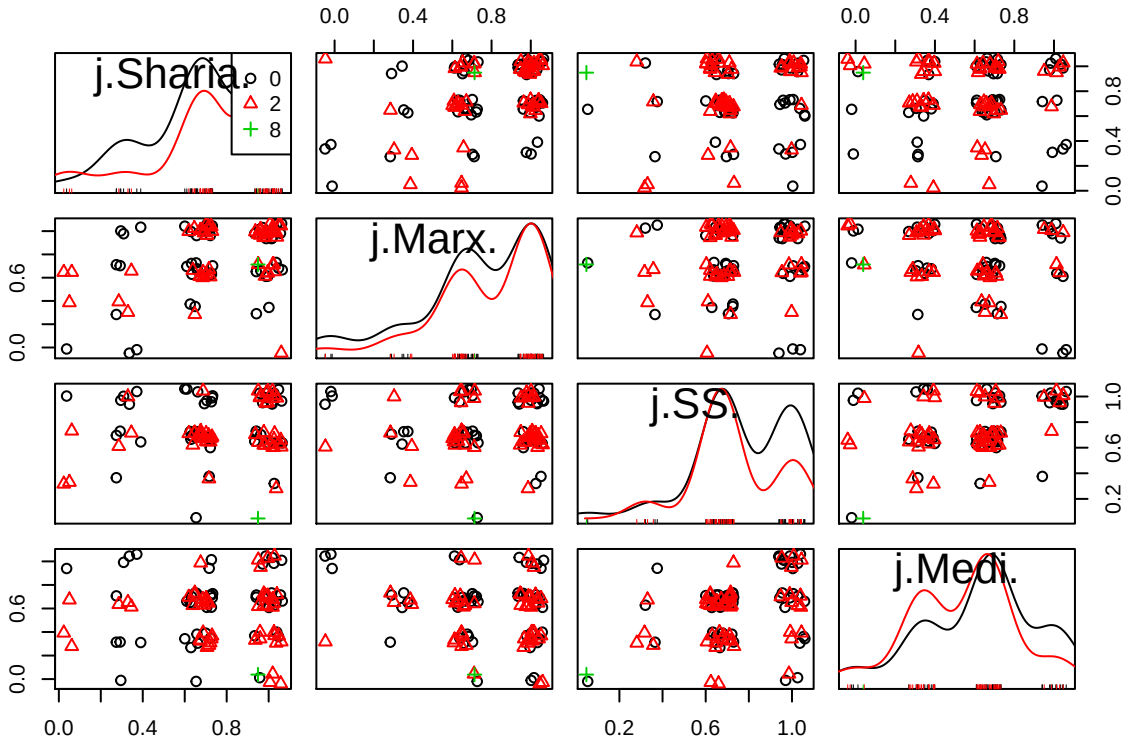
```
## Warning in smoother(.x[subs], .y[subs], col = col[i], log.x =
## logged("x"), : could not fit smooth
```



In terms of group

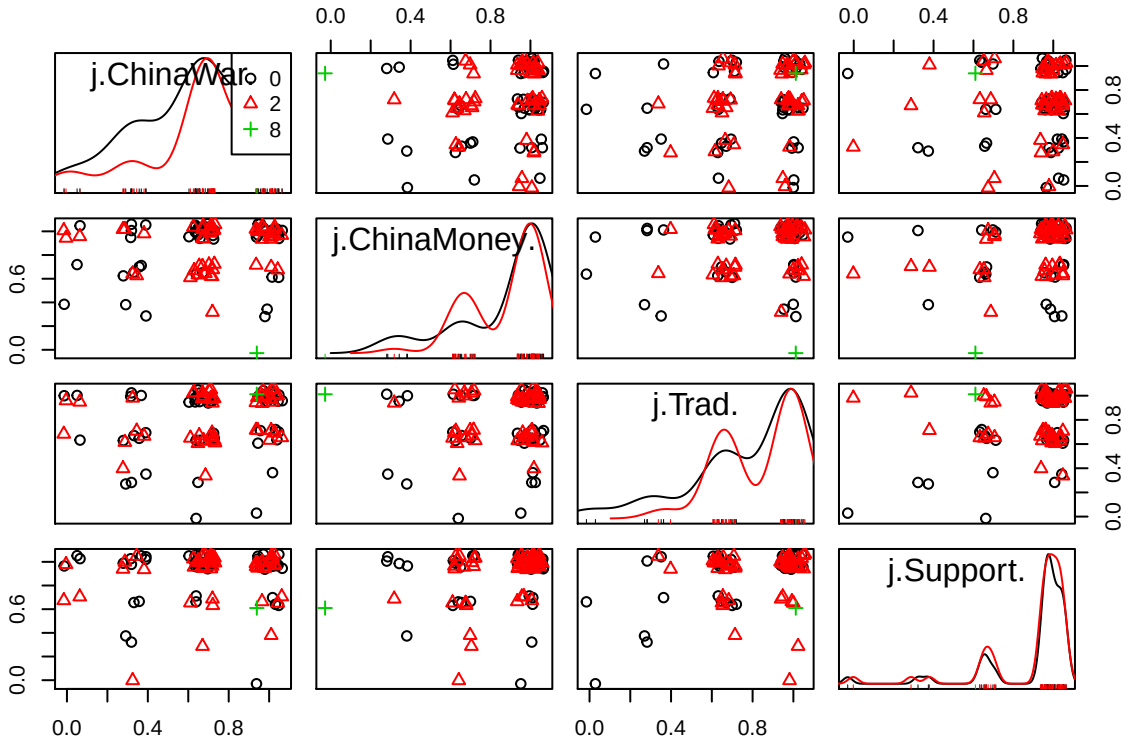
```
scatterplotMatrix(~j(Sharia)+j(Marx)+j(SS)+j(Medi)|c_df$t0, data=df,
reg.line=F, spread=FALSE, lwd=0, smoother=NULL)
```

```
## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically

## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically

## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically

## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically
```




```
scatterplotMatrix(~j(ChinaWar)+j(ChinaMoney)+j(Trad)+j(Support)|c_df$t0, data=df,
                  reg.line=F, spread=FALSE, lwd=0, smoother=NULL)
```

```
## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically

## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically

## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically

## Warning in diag.panel(...): cannot estimate density for group 8
## Error in density.default(xx, adjust = adjust, na.rm = TRUE) :
##   need at least 2 points to select a bandwidth automatically
```



principle component

```
pcs = prcomp(df, scale=TRUE)
```

bayesmix

```
library(bayesmix)
```

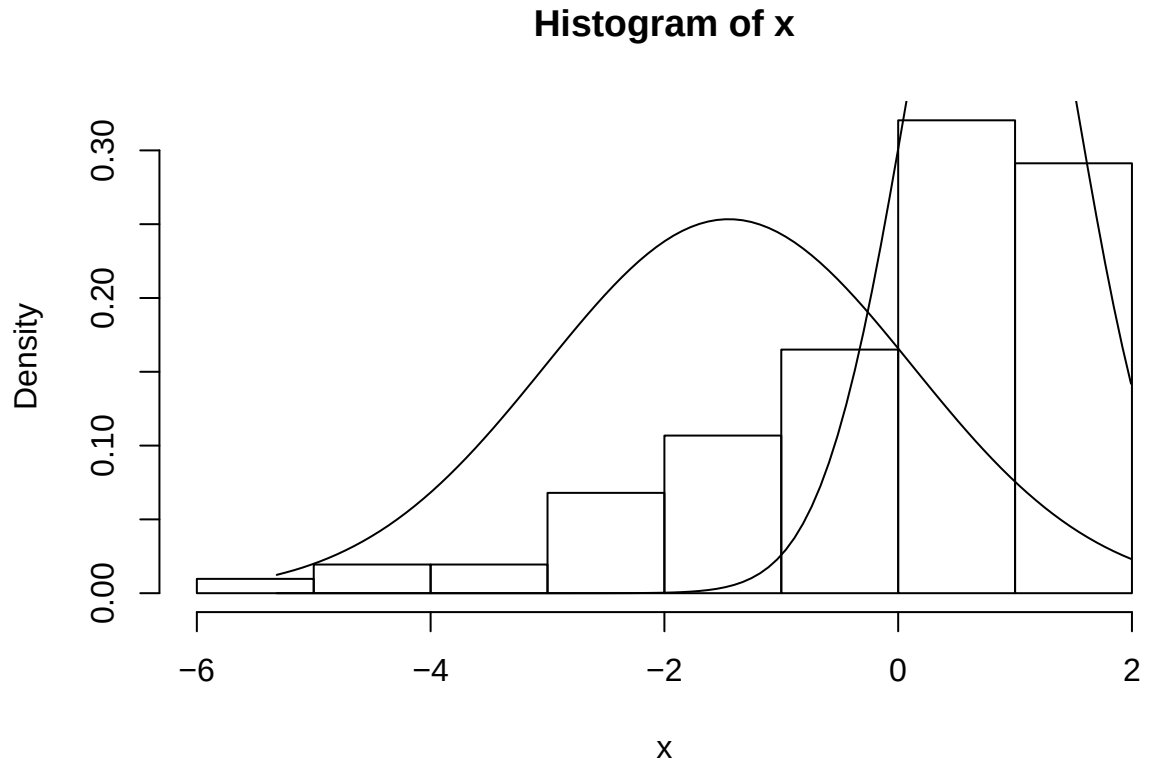
```
## Warning: package 'bayesmix' was built under R version 3.2.5
```

```
get_mm=function(vec, p_string) {
  model = BMMmodel(
    vec, k = 2,
    priors = list(kind = "independence", parameter=p_string),
    no.empty.classes=TRUE
  )
  control = JAGScontrol(variables = c("mu", "tau", "eta", "S"), burn.in = 1000, n.iter = 5000, seed = 1

  z = JAGSrun(vec, model = model, control = control, initialValues=list(S0=4))
  return(z)
}

mm1 = get_mm(pcs$x[,1],"priorsUncertain")
```

```
## Compiling model graph
```

```
##    Declaring variables
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 105
##    Unobserved stochastic nodes: 108
##    Total graph size: 549
##
## Initializing model
```

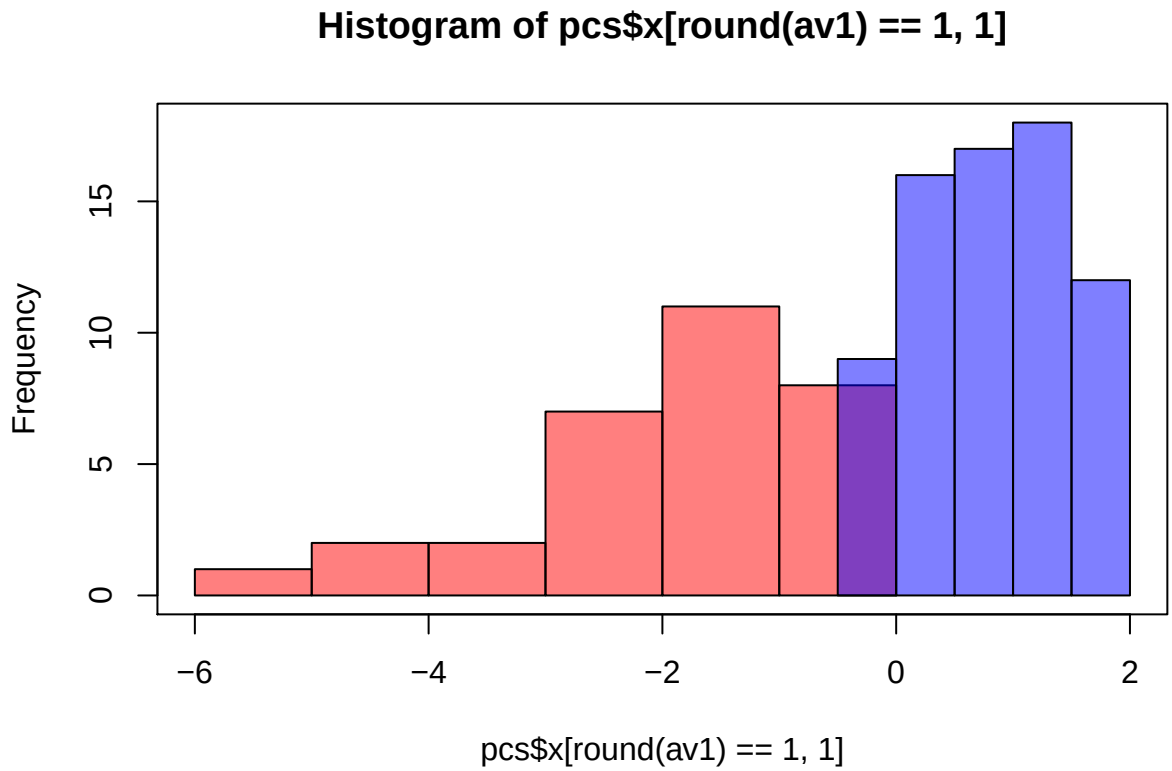
```
plot_hist_with_norms=function(x, mm) {
  # Assume mm is a mixture model with two components
  hist(x, freq=FALSE)
  xfit=seq(min(x), max(x), length=100)
  y1=dnorm(xfit, mean=mean(mm$results[,"mu[1]"]), sd=sqrt(mean(mm$results[,"sigma2[1]"])))
  y2=dnorm(xfit, mean=mean(mm$results[,"mu[2]"]), sd=sqrt(mean(mm$results[,"sigma2[2]"])))
  lines(xfit, y1)
  lines(xfit, y2)

}

plot_hist_with_norms(pcs$x[,1], mm1)
```



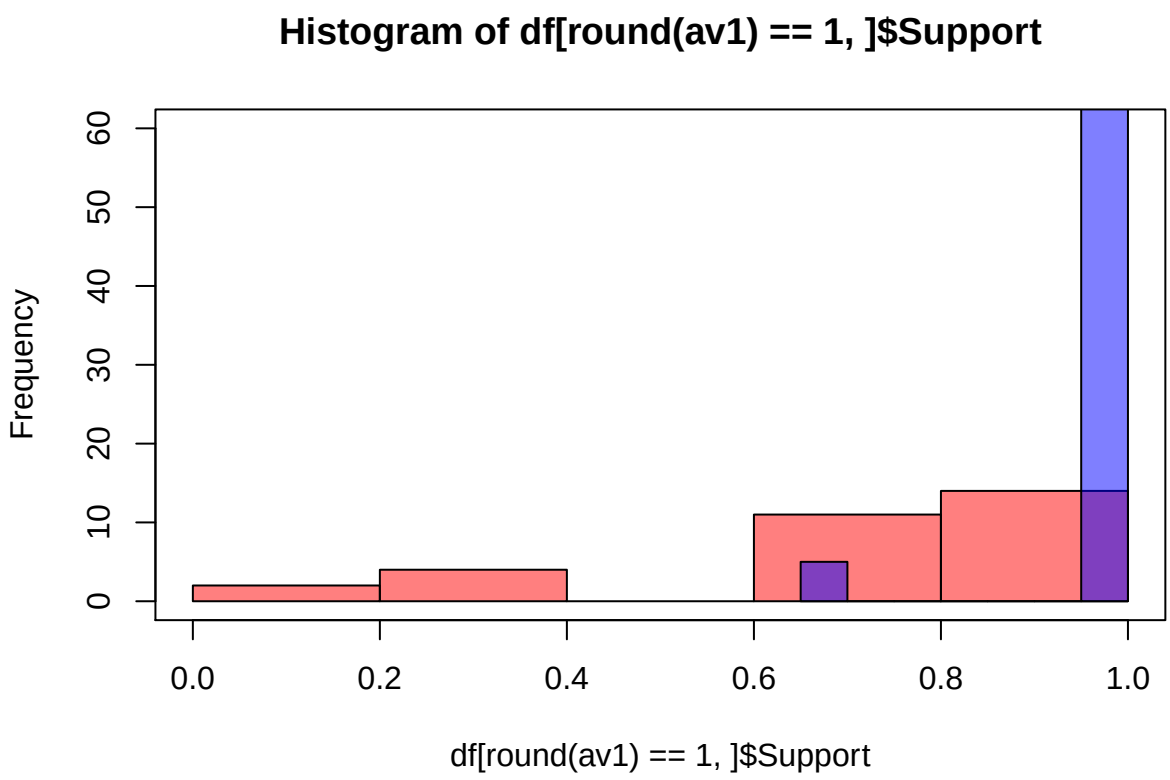

```
get_average_groups=function(mm, n) {
  av=c()
```

```
  for(i in 1:n) {
    av[i] = mean(mm$results[,paste("S[", i, "]", sep="")])
  }
  return(av)
}

# Read in EM with two fixed clusters over 10000 trials
c_2_df=read.csv("C:\\dev\\bayes-present\\explore\\cluster_data_2_comps_10000_trials.csv")

c_2_avg=c()
for(i in 1:103) {
  c_2_avg[i] = mean(t(c_2_df[i,])) + 1
}

av1=get_average_groups(mm1, 103)
```

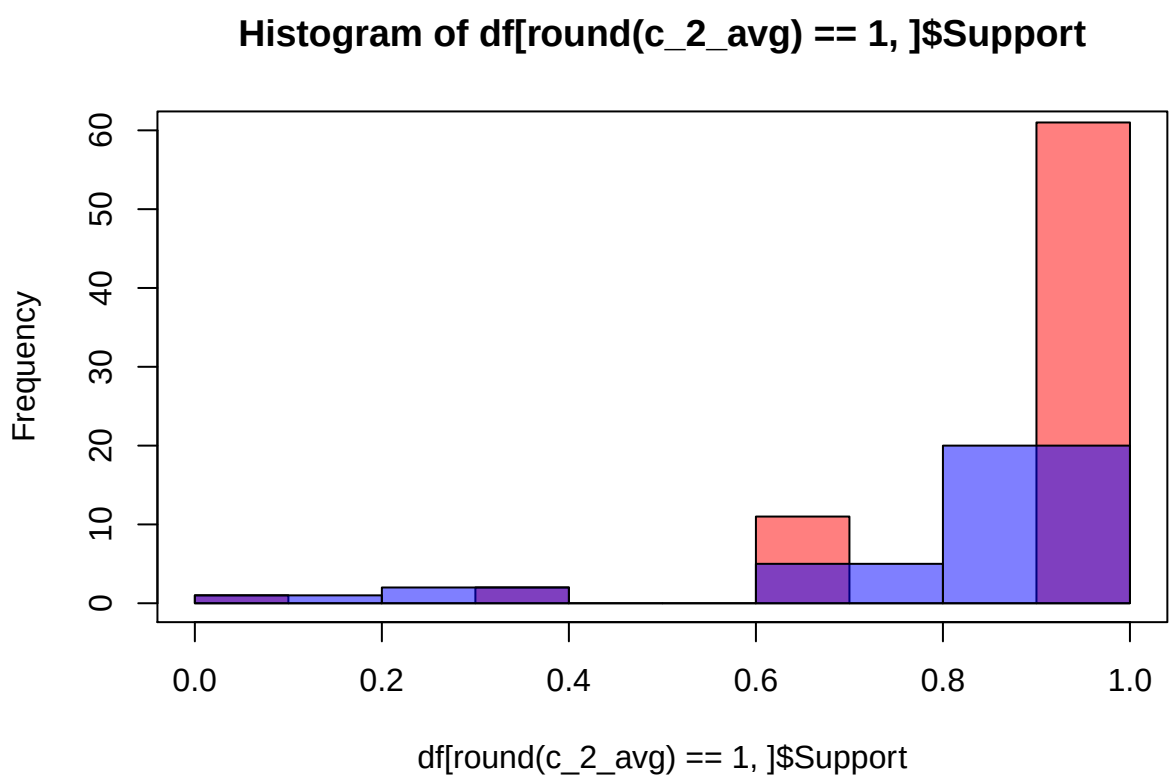
```
# Histogram Colored (blue and red)
hist(pcs$x[round(av1)==1,1], col=rgb(1,0,0,0.5), xlim=c(-6, 2), ylim=c(0, 18))
hist(pcs$x[round(av1)==2,1], col=rgb(0,0,1,0.5), add=T)
box()
```



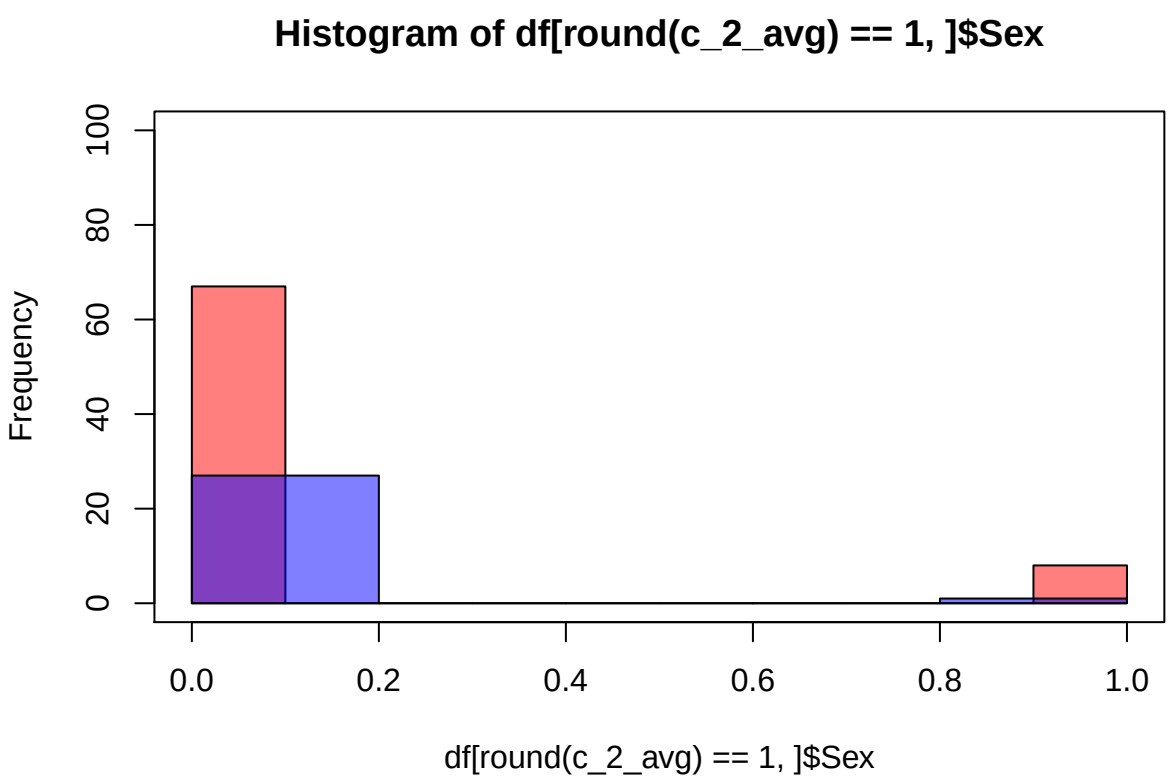

```
# Histogram Colored (blue and red)
hist(df[round(av1)==1,]$Support, col=rgb(1,0,0,0.5), ylim=c(0, 60))
hist(df[round(av1)==2,]$Support, col=rgb(0,0,1,0.5), add=T)
box()
```




```
hist(df[round(c_2_avg)==1,]$Support, col=rgb(1,0,0,0.5), ylim=c(0, 60))
hist(df[round(c_2_avg)==2,]$Support, col=rgb(0,0,1,0.5), add=T)
box()
```




```
hist(df[round(c_2_avg)==1,]$Sex, col=rgb(1,0,0,0.5), ylim=c(0, 100))
hist(df[round(c_2_avg)==2,]$Sex, col=rgb(0,0,1,0.5), add=T)
box()
```




```
percentage_female_c2_1 = mean(df[round(c_2_avg)==1,]$Sex)
percentage_female_c2_2 = mean(df[round(c_2_avg)==2,]$Sex)
```

```
# Gets differential entropy of observations which are
# assumed to follow multivariate normal
dif_entropy=function(obs) {
  k=ncol(obs)
  return((k/2*(1+log(2*pi))) + (1/2*log(det(cov(obs)))))
}

random_entropy_draws=function(obs) {
  ITERATIONS = 1000
  trial=c()
  sizes=c()
  for(it in 1:ITERATIONS) {
    ent=NA
    # Make sure we ignore degenerate cases
    while(!is.numeric(ent) | !is.finite(ent)) {
      partition=sample(0:1,nrow(obs),replace=T)
      ent=dif_entropy(df[partition==1,])
      sizes[it] = sum(partition)
      trial[it] = ent
    }
  }
  print(paste(ITERATIONS, "trials"))
  print(paste("Average trial size:", mean(sizes)))
```

```
  return(trial)
}

mc_draws=random_entropy_draws(df)
```

```
## [1] "1000 trials"
## [1] "Average trial size: 51.491"
```

```
em_1_draws=random_entropy_draws(df[round(c_2_avg)==1,])
```

```
## [1] "1000 trials"
## [1] "Average trial size: 37.582"
```

```
em_2_draws=random_entropy_draws(df[round(c_2_avg)==2,])
```

```
## [1] "1000 trials"
## [1] "Average trial size: 13.904"
```

```
bay_1_draws=random_entropy_draws(df[round(av1)==1,])
```

```
## [1] "1000 trials"
## [1] "Average trial size: 15.432"
```

```
bay_2_draws=random_entropy_draws(df[round(av1)==2,])
```

```
## [1] "1000 trials"
## [1] "Average trial size: 36.055"
```

```
report=function(draws, df) {
  print(summary(draws))
  print("99% Conf:")
  print(quantile(draws, c(.005, .995)))
  print(paste("Sample DE:", dif_entropy(df)))
}

report(mc_draws, df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8783  2.3440  2.7020  2.6780  3.0450  4.0160
## [1] "99% Conf:"
##     0.5%    99.5%
## 1.262369 3.830572
## [1] "Sample DE: 3.13084121839615"
```

```
report(em_1_draws, df[round(c_2_avg)==1,])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7846  2.3380  2.6510  2.6440  2.9810  3.9200
## [1] "99% Conf:"
##     0.5%    99.5%
## 1.358337 3.649411
## [1] "Sample DE: 3.20063505243795"
```

```
report(em_2_draws, df[round(c_2_avg)==2,])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8075  2.2990  2.6610  2.6360  3.0140  3.9240
## [1] "99% Conf:"
##     0.5%    99.5%
## 1.179824 3.617831
## [1] "Sample DE: -0.664629594117098"
```

```
report(bay_1_draws, df[round(av1)==1,])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5285  2.3110  2.7060  2.6210  2.9740  3.6920
## [1] "99% Conf:"
##      0.5%     99.5%
## 0.9300716 3.6080783
## [1] "Sample DE: 4.10330660772681"
```

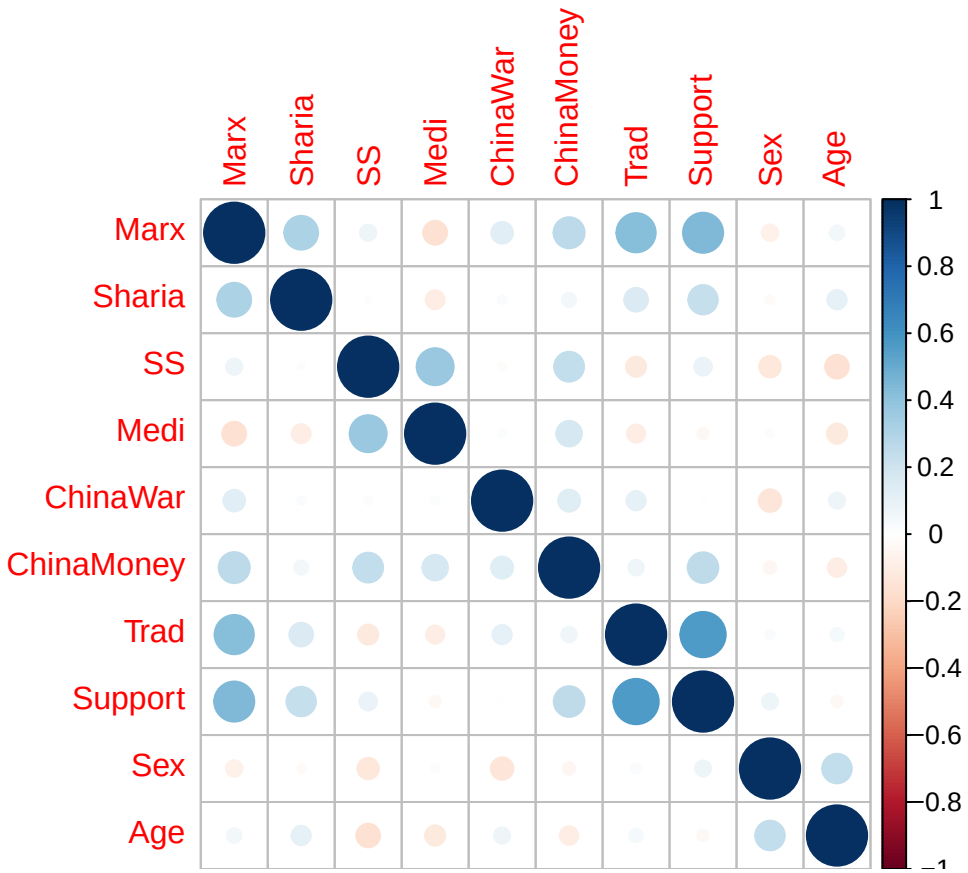
```
report(bay_1_draws, df[round(av1)==2,])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5285  2.3110  2.7060  2.6210  2.9740  3.6920
## [1] "99% Conf:"
##      0.5%     99.5%
## 0.9300716 3.6080783
## [1] "Sample DE: 0.442575424990931"
```

Clustering has

Correlation Visualization

```
library(corrplot)
M <- cor(df[round(c_2_avg)==1,])
corrplot(M, method="circle")
```


Marx
Sharia
SS
Medi
ChinaWar
ChinaMoney
Trad
Support
Sex
Age
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8