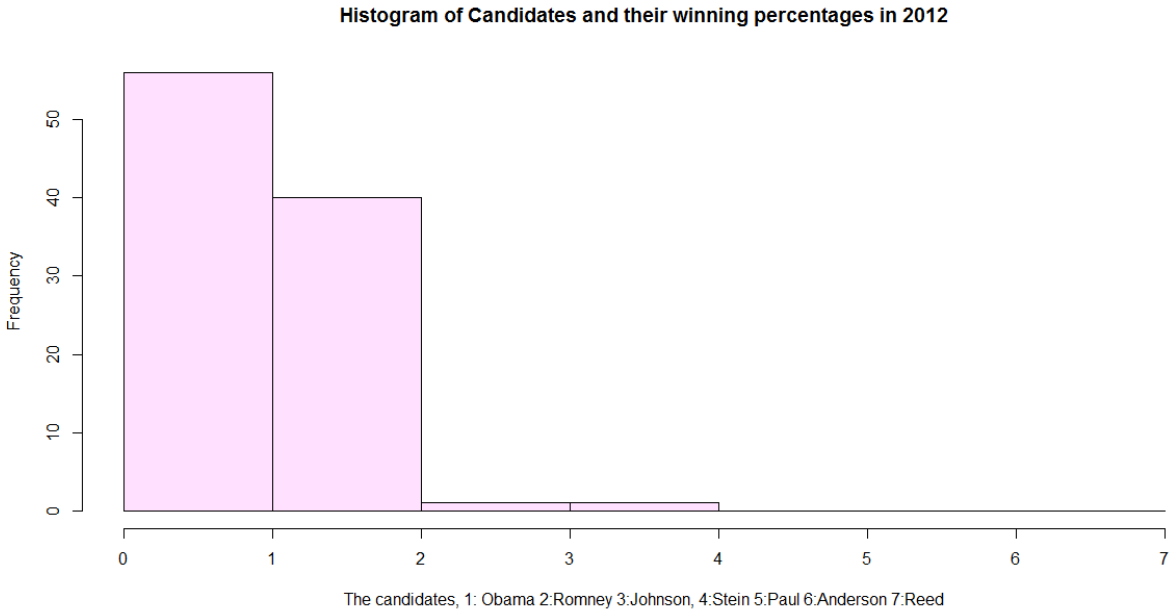
CSCI E-84 Homework 2
Submitted by: Engy Fouda

Exercise 1 Histogram

The R code:

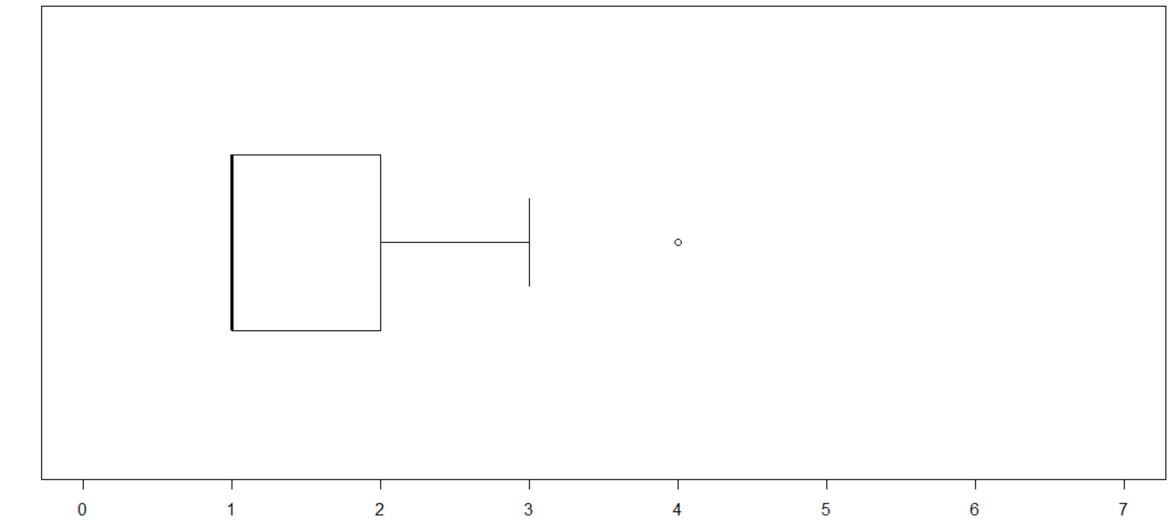
```
#Disclaimer: this code is following tutorials from www.lynda.com
#groups <- c(rep("Obama",   56.27), rep("Romney",
40.98),rep("Johnson", 1.31),            rep("Stein",  1.14),
rep("Paul",  0.29),            rep("Anderson",  0.01),
rep("Reed",  0)            )
#Q1 to make histogram for candidates & their percentages and bins on
the candidates
groups <- c(rep(1,   56.27),
            rep(2,    40.98),
            rep(3, 1.31),
            rep(4,  1.14),
            rep(5,  0.29),
            rep(6,  0.01),
            rep(7,  0)
)
hist(groups)
h <- hist(groups,  # Save histogram as object
          breaks = seq(0, 7, by = 1),
          col = "thistle1", # Or use: col = colors() [626]
          main = "Histogram of Candidates and their winning
percentages in 2012",
          xlab = "The candidates, 1: Obama 2:Romney 3:Johnson, 4:Stein
5:Paul 6:Anderson 7:Reed")
```



Exercise 2 Boxplot & Scatterplot

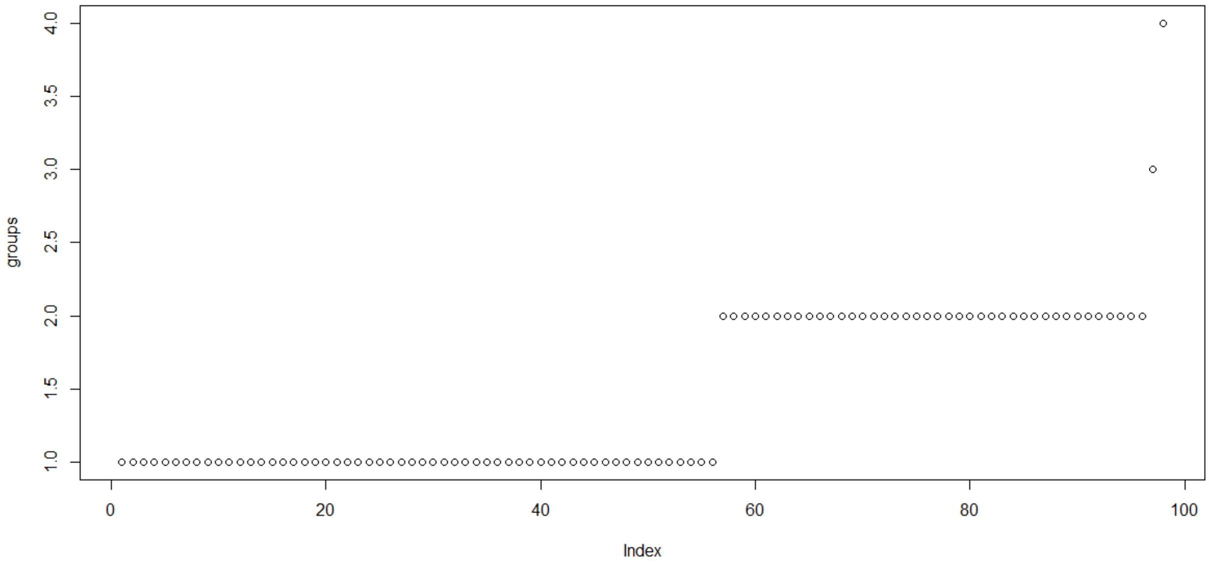
The boxplot R code:

```
boxplot(groups,horizontal = TRUE,ylim = c(0, 7),whisklty = 1)
```



The scatter plot R code:

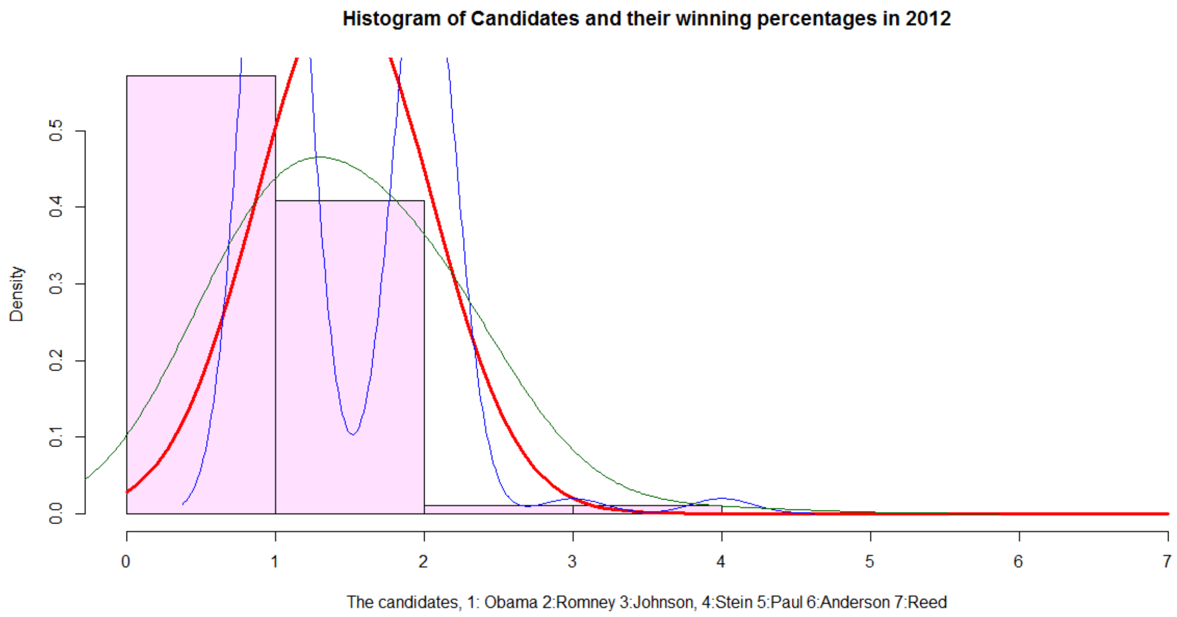
```
plot(groups)
```



Exercise 3 Overlay plot

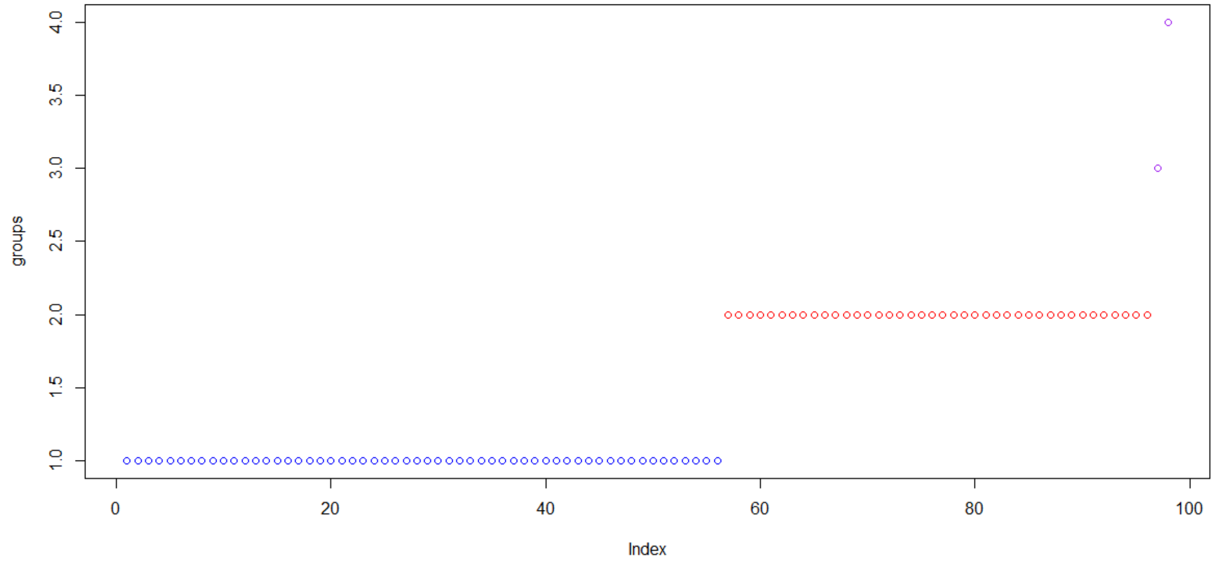
As in Lynda.com to plot the normal distribution over the density not frequency and the kernel density lines unadjusted and adjusted

```
h <- hist(groups,  # Save histogram as object
          #breaks to 7 bins according to the numberof candidates as
the question requests          #
          prob = TRUE,
          breaks = seq(0, 7, by = 1),
          col = "thistle1", # Or use: col = colors() [626]
          main = "Histogram of Candidates and their winning
percentages in 2012",
          xlab = "The candidates, 1: Obama 2:Romney 3:Johnson, 4:Stein
5:Paul 6:Anderson 7:Reed")
curve(dnorm(x, mean = mean(groups), sd = sd(groups)),
      col = "red",
      lwd = 3,
      add = TRUE)
# Plot 3 & 4: Kernel density lines (if prob = TRUE)
lines(density(groups), col = "blue")
lines(density(groups, adjust = 3), col = "darkgreen")
```



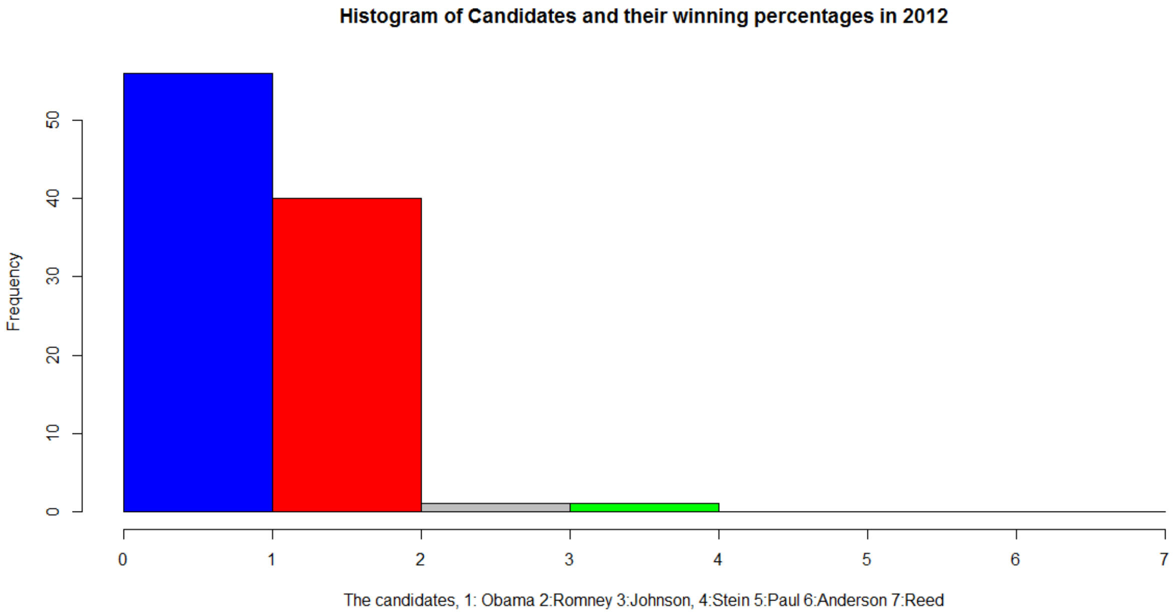
Scatter with color:

```
plot(groups, col = ifelse(groups==1,"blue",ifelse(groups==2, "red",
"purple")))
```



Histogram with color:

```
h <- hist(groups,  # Save histogram as object
          #breaks to 7 bins according to the numberof candidates as
the question requests          #
          breaks = seq(0, 7, by = 1),
          col =
c("blue","red","gray","green","yellow","yellow","yellow"),
          main = "Histogram of Candidates and their winning
percentages in 2012",
          xlab = "The candidates, 1: Obama 2:Romney 3:Johnson, 4:Stein
5:Paul 6:Anderson 7:Reed")
```


Histogram of Candidates and their winning percentages in 2012
Frequency
0
10
20
30
40
50
0
1
2
3
4
5
6
7
The candidates, 1: Obama 2:Romney 3:Johnson, 4:Stein 5:Paul 6:Anderson 7:Reed

Exercise 4 Frequency

The R code:

```
#Democratic won the following 27 states:
#CA,CO,CT,DE,DC,FL,HI,IL,IA,ME,MD,MA,MI,MN,NV,NH,NJ,NM,NY,OH,PA,RI,VT,
VA,WA,AI
#Republican won the following 24 states:
#AL,AK,AZ,AR,GA,ID,IN,KS,KY,LA,MS,MO,MT,NE,NC,ND,OK,SC,SD,TN,
TX,UT,WV,WY
groups <- c(rep("Democratic",   27),
            rep("Republic",    24))
# CREATE FREQUENCY TABLES
groups.t1 <- table(groups)  # Creates frequency table
groups.t1  # Print table

# MODIFY FREQUENCY TABLES
groups.t2 <- sort(groups.t1, decreasing = TRUE)  # Sorts by frequency,
saves table
groups.t2  # Print table
# PROPORTIONS AND PERCENTAGES
prop.table(groups.t2)  # Give proportions of total
round(prop.table(groups.t2), 2)  # Give proportions w/2 decimal places
round(prop.table(groups.t2), 2) * 100  # Give percentages w/o decimal
places
```

The output in the console:

```
> groups <- c(rep("Democratic",   27),
+              rep("Republic",    24))
> 
> # CREATE FREQUENCY TABLES
> groups.t1 <- table(groups)  # Creates frequency table
> groups.t1  # Print table
groups
Democratic   Republic 
        27         24 
> 
> # MODIFY FREQUENCY TABLES
> groups.t2 <- sort(groups.t1, decreasing = TRUE)  # Sorts by frequency, save
s table
> groups.t2  # Print table
groups
Democratic   Republic 
        27         24 
> 
> # PROPORTIONS AND PERCENTAGES
> prop.table(groups.t2)  # Give proportions of total
groups
Democratic   Republic 
 0.5294118  0.4705882 
> round(prop.table(groups.t2), 2)  # Give proportions w/2 decimal places
groups
Democratic   Republic 
```

```
      0.53       0.47
> round(prop.table(groups.t2), 2) * 100  # Give percentages w/o decimal place
s
groups
Democratic   Republic
        53         47
```

Exercise 5 Descriptive

The R code:

```
PARTY=c("R","D","IND","W","IND","IND","","R","D","LIB","GRE","W","","R
","D","LIB","GRE","W","W","W","W","W","W","","R","D","LIB","GRE","SLP"
,"","D","R","LIB","GRE","PFP","AIP","W","W","W","W","W","W","W","W",""
,"D","R","LIB","GRE","AMC","PFP","UN","JUS","WTP","AMP","SLP","SUS","A
TP","OBJ","SWP","SEP","W","","D","R","LIB","IP","W","W","W","W","W","W
","","D","R","LIB","GRE","W","W","W","","D","R","DCG","LIB","W","","D"
,"R","LBF","GPF","PFF","OBF","CPF","JPF","APF","REF","SFL","SLF","W","
W","W","W","W","","R","D","LIB","W","W","W","W","W","W","W","W","","D"
,"R","LIB","GRE","","R","D","LIB","IND","IND","CON","","D","R","LIB","
GRE","W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","
W","W","W","W","W","W","W","W","W","","R","D","LIB","W","W","W","W","W
","W","W","W","W","","D","R","LIB","W","IG","CON","NP","SWP","PSL","",
"R","D","LIB","REF","W","W","W","W","W","W","W","W","W","","R","D","LI
B","IND","GRE","W","W","W","W","W","W","W","W","","R","D","LIB","GRE",
"CON","WTP","JUS","SLP","P","SWP","SEP","","D","R","LIB","GI","W","W",
"W","","D","R","LIB","GRE","W","W","W","W","W","W","W","W","W","W","W"
,"W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","","D","R
","LIB","GR","W","","D","R","GRE","UST","W","NLP","W","W","W","","DFL"
,"R","LIB","GRE","W","CON","GRT","JUS","CG","SWP","SLP","W","W","W","W
","W","W","","R","D","LIB","CON","GRE","REF","","R","D","LIB","CON",""
,"R","D","LIB","W","W","W","W","W","W","W","","R","D","LIB","W","BP","
","D","R","LIB","","IAP","","D","R","LIB","W","W","CON","W","","D","R"
,"LIB","GRE","CON","NJJ","NSA","SWP","ATP","SLP","","D","R","LIB","GRE
","NMI","CON","","Combined Parties:","D","WF","Combined
Parties:","R","CRV","LIB","GRE","W","CON","SLP","W","W","W","W","W","W
","W","","R","D","LIB","W","W","","R","DNL","LIB","W","GRE","CON","","
D","R","LIB","GRE","IND","CON","SUS","W","W","W","W","W","","R","D",""
,"D","R","LIB","PG","W","CON","PRO","","D","R","LIB","GRE","W","W","W"
,"","D","R","LIB","GRE","W","W","CON","JUS","SLP","W","W","","R","D","
LIB","GRE","CON","","R","D","LIB","CON","","R","D","IND","GRE","CON","
IND","IND","","R","D","LIB","GRE","W","W","W","W","W","W","W","","R","
D","LIB","JUS","GRE","CON","UN","W","W","W","W","W","W","W","W","","D"
,"R","LIB","JUS","W","W","SLP","W","W","W","","D","R","LIB","CON","GRE
","W","W","W","W","W","W","","D","R","LIB","GRE","CON","JUS","SLP","SW
P","","R","D","LIB","MTP","NPA","","D","R","IND","IND","W","CON","IND"
,"IND","W","W","","R","D","LIB","W","CON")
GENERAL.RESULTS=c(1255925,795696,12328,4011,3397,2981,2074338,164676,1
22640,7392,2917,2870,300495,1233654,1025232,32100,7816,289,119,17,14,7
,6,2299254,647744,394409,16276,9305,1734,1069468,7854285,4839958,14322
1,85638,53824,38372,21461,992,503,82,79,72,54,6,13038547,1323102,11852
43,35545,7508,6234,5059,2589,1260,792,679,317,308,266,235,192,189,4,25
69522,905083,634892,12580,5487,863,25,19,5,5,1,1558960,242584,165484,3
882,1940,23,7,1,413921,267070,21381,2458,2083,772,293764,4237756,41634
47,44726,8947,8154,3856,2607,1754,946,820,799,322,36,3,3,2,1,8474179,2
078688,1773827,45324,1516,432,154,55,30,21,2,1,3900050,306658,121015,3
840,3184,434697,420911,212787,9453,4402,2499,2222,652274,3019512,21352
16,56229,30222,233,185,182,121,22,16,14,10,8,5,5,4,4,4,3,3,3,2,2,2,2,2
,1,1,1,5242014,1420543,1152887,50111,625,290,35,17,8,8,7,2,1,2624534,8
22544,730617,12926,7442,3769,3038,1027,445,372,1582180,692634,440726,2
0456,5017,714,187,95,58,48,19,12,4,1,1159971,1087190,679370,17063,6872
,6337,245,60,37,30,4,2,1,1,1797212,1152262,809141,18157,6978,2508,1767
,1368,622,518,389,355,1994065,401306,292276,9352,8119,2035,62,30,71318
0,1677844,971869,30195,17110,8788,625,418,204,64,35,26,19,19,18,18,15,
13,8,7,7,7,5,3,2,2,2,1,1,1,1,2707327,1921290,1188314,30920,20691,6552,
3167767,2564569,2115256,21897,16119,7774,5147,89,68,42,4730961,1546167
,1320225,35098,13023,10533,3722,3149,1996,1092,1051,397,46,35,14,8,4,1
,2936561,710746,562949,6676,2609,1588,1016,1285584,1482440,1223796,431
51,7936,2757323,267928,201839,14165,59,39,6,5,4,2,1,484048,475064,3020
81,11109,3717,2408,794379,531373,463567,10968,5770,3240,1014918,369561
,329918,8212,1374,875,708,324,710972,2125101,1477568,21045,9888,2064,1
724,1007,710,664,521,3640292,415335,335788,27788,2691,1174,982,783758,
4485741,4337622,148119,2490431,2228060,262371,47256,39982,9076,6274,20
50,217,34,34,27,19,12,6,7081159,2270395,2178391,44515,11537,534,450537
2,188163,124827,5231,1860,1361,1185,322627,2827709,2661437,49493,18573
,12502,8152,2944,14,13,9,1,0,5580847,891325,443547,1334872,970488,7541
75,24089,19427,13275,4432,3384,1789270,2990274,2680434,49991,21341,112
19,383,28,5753670,279677,157204,4388,2421,686,617,430,416,132,64,14,44
6049,1071645,865941,16321,5446,4765,1964118,210610,145039,5795,2371,36
3815,1462330,960709,18623,6515,6022,2639,1739,2458577,4569843,3308124,
88580,24657,1287,426,374,209,162,102,87,7993851,740600,251813,12572,53
35,3817,2871,393,18,6,5,5,2,1,1,1,1017440,199239,92698,3487,1128,717,7
10,695,594,13,9,299290,1971820,1822522,31216,13058,8627,7151,76,14,3,1
,1,3854489,1755396,1290670,42202,20928,8851,4946,1318,1205,3125516,417
655,238269,6302,4406,3806,670438,1620985,1407966,20439,7665,5170,4930,
553,526,112,88,3068434,170962,69286,5326,2035,1452)

q5=data.frame(PARTY,GENERAL.RESULTS)
str(q5);

RepVoters<- subset(q5, q5$PARTY=="R");
sumRepVoters=sum(RepVoters$GENERAL.RESULTS)

DemVoters<- subset(q5, q5$PARTY=="D");
sumDemVoters=sum(DemVoters$GENERAL.RESULTS)

GreenVoters<- subset(q5, q5$PARTY=="GRE");
sumGreenVoters=sum(GreenVoters$GENERAL.RESULTS)

LibVoters<- subset(q5, q5$PARTY=="LIB");
sumLibVoters=sum(LibVoters$GENERAL.RESULTS)

sumvoter=c(sumRepVoters,sumDemVoters,sumGreenVoters,sumLibVoters);
party=c("R","D","GRE","LIB");
tableVoterParty=data.frame(sumvoter,party);
```

```
#plot(sumvoter ~ party, data = tableVoterpopYears,xlab = "party",ylab
= "voters population",main = "voters")


ranking <- tableVoterParty[order(-sumvoter),]
ranking  # Print table

# PROPORTIONS AND PERCENTAGES
prop.table(ranking$sumvoter)  # Give proportions of total
round(prop.table(ranking$sumvoter), 2)  # Give proportions w/2 decimal
places
percentages<-round(prop.table(ranking$sumvoter), 2) * 100  # Give
percentages w/o decimal places
tablePartyPercentage=data.frame(ranking$party,percentages)
tablePartyPercentage


# CALCULATE DESCRIPTIVES
summary(tableVoterParty$sumvoter)  # Summary for one variable
summary(tableVoterParty)  # Summary for entire table

# Tukey's five-number summary: minimum, lower-hinge,
# median, upper-hinge, maximum. No labels.
fivenum(tableVoterParty$sumvoter)

# Boxplot stats: hinges, n, CI, outliers
boxplot.stats(tableVoterParty$sumvoter)

# ALTERNATIVE DESCRIPTIVES
# From the package "psych"
#help(package = "psych")
#install.packages("psych")
require("psych")
describe(tableVoterParty)
```

The output from the console:

```
> PARTY=c("R","D","IND","W","IND","IND","","R","D","LIB","GRE","W","","R","D"
,"LIB","GRE","W","W","W","W","W","W","","R","D","LIB","GRE","SLP","","D","R",
"LIB","GRE","PFP","AIP","W","W","W","W","W","W","W","W","","D","R","LIB","GRE
","AMC","PFP","UN","JUS","WTP","AMP","SLP","SUS","ATP","OBJ","SWP","SEP","W",
"","D","R","LIB","IP","W","W","W","W","W","W","","D","R","LIB","GRE","W","W",
"W","","D","R","DCG","LIB","W","","D","R","LBF","GPF","PFF","OBF","CPF","JPF"
,"APF","REF","SFL","SLF","W","W","W","W","W","","R","D","LIB","W","W","W","W"
,"W","W","W","W","","D","R","LIB","GRE","","R","D","LIB","IND","IND","CON",""
,"D","R","LIB","GRE","W","W","W","W","W","W","W","W","W","W","W","W","W","W",
"W","W","W","W","W","W","W","W","W","W","W","","R","D","LIB","W","W","W","W",
"W","W","W","W","W","","D","R","LIB","W","IG","CON","NP","SWP","PSL","","R","
D","LIB","REF","W","W","W","W","W","W","W","W","W","","R","D","LIB","IND","GR
E","W","W","W","W","W","W","W","W","","R","D","LIB","GRE","CON","WTP","JUS","
SLP","P","SWP","SEP","","D","R","LIB","GI","W","W","W","","D","R","LIB","GRE"
,"W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","W","W",
"W","W","W","W","W","W","W","","D","R","LIB","GR","W","","D","R","GRE","UST",
"W","NLP","W","W","W","","DFL","R","LIB","GRE","W","CON","GRT","JUS","CG","SW
P","SLP","W","W","W","W","W","W","","R","D","LIB","CON","GRE","REF","","R","D
","LIB","CON","","R","D","LIB","W","W","W","W","W","W","W","","R","D","LIB","
W","BP","","D","R","LIB","","IAP","","D","R","LIB","W","W","CON","W","","D","
R","LIB","GRE","CON","NJJ","NSA","SWP","ATP","SLP","","D","R","LIB","GRE","NM
I","CON","","Combined Parties:","D","WF","Combined Parties:","R","CRV","LIB",
"GRE","W","CON","SLP","W","W","W","W","W","W","W","","R","D","LIB","W","W",""
,"R","DNL","LIB","W","GRE","CON","","D","R","LIB","GRE","IND","CON","SUS","W"
,"W","W","W","W","","R","D","","D","R","LIB","PG","W","CON","PRO","","D","R",
"LIB","GRE","W","W","W","","D","R","LIB","GRE","W","W","CON","JUS","SLP","W",
"W","","R","D","LIB","GRE","CON","","R","D","LIB","CON","","R","D","IND","GRE
","CON","IND","IND","","R","D","LIB","GRE","W","W","W","W","W","W","W","","R"
,"D","LIB","JUS","GRE","CON","UN","W","W","W","W","W","W","W","W","","D","R",
"LIB","JUS","W","W","SLP","W","W","W","","D","R","LIB","CON","GRE","W","W","W
","W","W","W","","D","R","LIB","GRE","CON","JUS","SLP","SWP","","R","D","LIB"
,"MTP","NPA","","D","R","IND","IND","W","CON","IND","IND","W","W","","R","D",
"LIB","W","CON")
> GENERAL.RESULTS=c(1255925,795696,12328,4011,3397,2981,2074338,164676,122640
,7392,2917,2870,300495,1233654,1025232,32100,7816,289,119,17,14,7,6,2299254,6
47744,394409,16276,9305,1734,1069468,7854285,4839958,143221,85638,53824,38372
,21461,992,503,82,79,72,54,6,13038547,1323102,1185243,35545,7508,6234,5059,25
89,1260,792,679,317,308,266,235,192,189,4,2569522,905083,634892,12580,5487,86
3,25,19,5,5,1,1558960,242584,165484,3882,1940,23,7,1,413921,267070,21381,2458
,2083,772,293764,4237756,4163447,44726,8947,8154,3856,2607,1754,946,820,799,3
22,36,3,3,2,1,8474179,2078688,1773827,45324,1516,432,154,55,30,21,2,1,3900050
,306658,121015,3840,3184,434697,420911,212787,9453,4402,2499,2222,652274,3019
512,2135216,56229,30222,233,185,182,121,22,16,14,10,8,5,5,4,4,4,3,3,3,2,2,2,2
,2,1,1,1,5242014,1420543,1152887,50111,625,290,35,17,8,8,7,2,1,2624534,822544
,730617,12926,7442,3769,3038,1027,445,372,1582180,692634,440726,20456,5017,71
4,187,95,58,48,19,12,4,1,1159971,1087190,679370,17063,6872,6337,245,60,37,30,
4,2,1,1,1797212,1152262,809141,18157,6978,2508,1767,1368,622,518,389,355,1994
065,401306,292276,9352,8119,2035,62,30,713180,1677844,971869,30195,17110,8788
,625,418,204,64,35,26,19,19,18,18,15,13,8,7,7,7,5,3,2,2,2,1,1,1,1,2707327,192
1290,1188314,30920,20691,6552,3167767,2564569,2115256,21897,16119,7774,5147,8
9,68,42,4730961,1546167,1320225,35098,13023,10533,3722,3149,1996,1092,1051,39
7,46,35,14,8,4,1,2936561,710746,562949,6676,2609,1588,1016,1285584,1482440,12
23796,43151,7936,2757323,267928,201839,14165,59,39,6,5,4,2,1,484048,475064,30
2081,11109,3717,2408,794379,531373,463567,10968,5770,3240,1014918,369561,3299
18,8212,1374,875,708,324,710972,2125101,1477568,21045,9888,2064,1724,1007,710
,664,521,3640292,415335,335788,27788,2691,1174,982,783758,4485741,4,337,622,1
48,119,2490431,2,228,060,262,371,47256,39982,9076,6274,2050,217,34,34,27,19,1
2,6,7081159,2270395,2178391,44515,11537,534,4505372,188163,124827,5231,1860,1
361,1185,322627,2827709,2661437,49493,18573,12502,8152,2944,14,13,9,1,0,55808
47,891325,443547,1334872,970488,754175,24089,19427,13275,4432,3384,1789270,29
90274,2680434,49991,21341,11219,383,28,5753670,279677,157204,4388,2421,686,61
7,430,416,132,64,14,446049,1071645,865941,16321,5446,4765,1964118,210610,1450
39,5795,2371,363815,1462330,960709,18623,6515,6022,2639,1739,2458577,4569843,
3308124,88580,24657,1287,426,374,209,162,102,87,7993851,740600,251813,12572,5
335,3817,2871,393,18,6,5,5,2,1,1,1,1017440,199239,92698,3487,1128,717,710,695
,594,13,9,299290,1971820,1822522,31216,13058,8627,7151,76,14,3,1,1,3854489,17
55396,1290670,42202,20928,8851,4946,1318,1205,3125516,417655,238269,6302,4406
,3806,670438,1620985,1407966,20439,7665,5170,4930,553,526,112,88,3068434,1709
62,69286,5326,2035,1452)
> q5=data.frame(PARTY,GENERAL.RESULTS)
Error in data.frame(PARTY, GENERAL.RESULTS) :
  arguments imply differing number of rows: 534, 540
> GENERAL.RESULTS=c(1255925,795696,12328,4011,3397,2981,2074338,164676,122640
,7392,2917,2870,300495,1233654,1025232,32100,7816,289,119,17,14,7,6,2299254,6
47744,394409,16276,9305,1734,1069468,7854285,4839958,143221,85638,53824,38372
,21461,992,503,82,79,72,54,6,13038547,1323102,1185243,35545,7508,6234,5059,25
89,1260,792,679,317,308,266,235,192,189,4,2569522,905083,634892,12580,5487,86
3,25,19,5,5,1,1558960,242584,165484,3882,1940,23,7,1,413921,267070,21381,2458
,2083,772,293764,4237756,4163447,44726,8947,8154,3856,2607,1754,946,820,799,3
22,36,3,3,2,1,8474179,2078688,1773827,45324,1516,432,154,55,30,21,2,1,3900050
,306658,121015,3840,3184,434697,420911,212787,9453,4402,2499,2222,652274,3019
512,2135216,56229,30222,233,185,182,121,22,16,14,10,8,5,5,4,4,4,3,3,3,2,2,2,2
,2,1,1,1,5242014,1420543,1152887,50111,625,290,35,17,8,8,7,2,1,2624534,822544
,730617,12926,7442,3769,3038,1027,445,372,1582180,692634,440726,20456,5017,71
4,187,95,58,48,19,12,4,1,1159971,1087190,679370,17063,6872,6337,245,60,37,30,
4,2,1,1,1797212,1152262,809141,18157,6978,2508,1767,1368,622,518,389,355,1994
065,401306,292276,9352,8119,2035,62,30,713180,1677844,971869,30195,17110,8788
,625,418,204,64,35,26,19,19,18,18,15,13,8,7,7,7,5,3,2,2,2,1,1,1,1,2707327,192
1290,1188314,30920,20691,6552,3167767,2564569,2115256,21897,16119,7774,5147,8
9,68,42,4730961,1546167,1320225,35098,13023,10533,3722,3149,1996,1092,1051,39
7,46,35,14,8,4,1,2936561,710746,562949,6676,2609,1588,1016,1285584,1482440,12
23796,43151,7936,2757323,267928,201839,14165,59,39,6,5,4,2,1,484048,475064,30
2081,11109,3717,2408,794379,531373,463567,10968,5770,3240,1014918,369561,3299
18,8212,1374,875,708,324,710972,2125101,1477568,21045,9888,2064,1724,1007,710
,664,521,3640292,415335,335788,27788,2691,1174,982,783758,4485741,4337622,148
119,2490431,2228060,262371,47256,39982,9076,6274,2050,217,34,34,27,19,12,6,70
81159,2270395,2178391,44515,11537,534,4505372,188163,124827,5231,1860,1361,11
85,322627,2827709,2661437,49493,18573,12502,8152,2944,14,13,9,1,0,5580847,891
325,443547,1334872,970488,754175,24089,19427,13275,4432,3384,1789270,2990274,
2680434,49991,21341,11219,383,28,5753670,279677,157204,4388,2421,686,617,430,
416,132,64,14,446049,1071645,865941,16321,5446,4765,1964118,210610,145039,579
5,2371,363815,1462330,960709,18623,6515,6022,2639,1739,2458577,4569843,330812
4,88580,24657,1287,426,374,209,162,102,87,7993851,740600,251813,12572,5335,38
17,2871,393,18,6,5,5,2,1,1,1,1017440,199239,92698,3487,1128,717,710,695,594,1
3,9,299290,1971820,1822522,31216,13058,8627,7151,76,14,3,1,1,3854489,1755396,
1290670,42202,20928,8851,4946,1318,1205,3125516,417655,238269,6302,4406,3806,
670438,1620985,1407966,20439,7665,5170,4930,553,526,112,88,3068434,170962,692
86,5326,2035,1452)
> q5=data.frame(PARTY,GENERAL.RESULTS)
> str(q5);
'data.frame':   534 obs. of  2 variables:
 $ PARTY          : Factor w/ 57 levels "","AIP","AMC",..: 45 13 24 55 24 24
1 45 13 29 ...
 $ GENERAL.RESULTS: num  1255925 795696 12328 4011 3397 ...
> RepVoters<- subset(q5, q5$PARTY=="R");
> sumRepVoters=sum(RepVoters$GENERAL.RESULTS)
>
> DemVoters<- subset(q5, q5$PARTY=="D");
> sumDemVoters=sum(DemVoters$GENERAL.RESULTS)
>
> GreenVoters<- subset(q5, q5$PARTY=="GRE");
> sumGreenVoters=sum(GreenVoters$GENERAL.RESULTS)
>
> LibVoters<- subset(q5, q5$PARTY=="LIB");
> sumLibVoters=sum(LibVoters$GENERAL.RESULTS)
>
> sumvoter=c(sumRepVoters,sumDemVoters,sumGreenVoters,sumLibVoters);
> party=c("R","D","GRE","LIB");
> tableVoterParty=data.frame(sumvoter,party);
> #plot(sumvoter ~ party, data = tableVoterpopYears,xlab = "party",ylab = "vo
ters population",main = "voters")
>
>
> ranking <- tableVoterParty[order(-sumvoter),]
> ranking  # Print table
  sumvoter party
2 64096682     D
1 60671133     R
4  1172081   LIB
3   381710   GRE
>
> # PROPORTIONS AND PERCENTAGES
> prop.table(ranking$sumvoter)  # Give proportions of total
[1] 0.507408701 0.480291020 0.009278547 0.003021732
> round(prop.table(ranking$sumvoter), 2)  # Give proportions w/2 decimal plac
es
[1] 0.51 0.48 0.01 0.00
> percentages<-round(prop.table(ranking$sumvoter), 2) * 100  # Give percentag
es w/o decimal places
> tablePartyPercentage=data.frame(ranking$party,percentages)
> tablePartyPercentage
  ranking.party percentages
1             D          51
2             R          48
3           LIB           1
4           GRE           0
>
>
> # CALCULATE DESCRIPTIVES
> summary(tableVoterParty$sumvoter)  # Summary for one variable
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  381700   974500 30920000 31580000 61530000 64100000
> summary(tableVoterParty)  # Summary for entire table
    sumvoter         party
 Min.   :  381710   D  :1
 1st Qu.:  974488   GRE:1
 Median :30921607   LIB:1
 Mean   :31580402   R  :1
 3rd Qu.:61527520  
 Max.   :64096682  
> 
> # Tukey's five-number summary: minimum, lower-hinge,
> # median, upper-hinge, maximum. No labels.
> fivenum(tableVoterParty$sumvoter)
[1]   381710.0   776895.5 30921607.0 62383907.5 64096682.0
> 
> # Boxplot stats: hinges, n, CI, outliers
> boxplot.stats(tableVoterParty$sumvoter)
$stats
[1]   381710.0   776895.5 30921607.0 62383907.5 64096682.0

$n
[1] 4

$conf
[1] -17747932  79591146

$out
numeric(0)

> 
> # ALTERNATIVE DESCRIPTIVES
> # From the package "psych"
> help(package = "psych")
> install.packages("psych")
Error in install.packages : Updating loaded packages
> require("psych")
> describe(tableVoterParty)
         vars n       mean          sd     median    trimmed         mad    m
in      max
sumvoter    1 4 31580401.5 35597768.95 30921607.0 31580401.5 44692549.27 3817
10 64096682
party*      2 4        2.5        1.29        2.5        2.5        1.48     
1        4
            range skew kurtosis          se
sumvoter 63714972    0    -2.43 17798884.48
party*          3    0    -2.08        0.65
> install.packages("psych")
Installing package into ‘C:/Users/Engy/Documents/R/win-library/3.3’
(as ‘lib’ is unspecified)
Warning in install.packages :
  package ‘psych’ is in use and will not be installed
> require("psych")
> describe(tableVoterParty)
         vars n       mean          sd     median    trimmed         mad    m
in      max
sumvoter    1 4 31580401.5 35597768.95 30921607.0 31580401.5 44692549.27 3817
10 64096682
party*      2 4        2.5        1.29        2.5        2.5        1.48     
1        4
            range skew kurtosis          se
sumvoter 63714972    0    -2.43 17798884.48
party*          3    0    -2.08        0.65
```

Exercise 6 Single proportion hypothesis

The R code:

```
# Based on the historical data of Maine:
http://www.270towin.com/states/Maine
#Since 1960, Republicans won 6 times out of 14 times, and Democratic
won 8 times
#As I expect that the Democratic will win in Maine because Maine is a
solid Democratic state
#My Null Hypothesis will be Republicans win in 2016
#It is a 2-tailed Hypothesis
#H0: R wins
#Ha: R loses

# PROP TEST
prop.test(6, 14)
```

The output from the console:

```
> prop.test(6, 14)

	1-sample proportions test with continuity correction

data:  6 out of 14, null probability 0.5
X-squared = 0.071429, df = 1, p-value = 0.7893
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.1881363 0.7035185
sample estimates:
        p 
0.4285714
```

For extra credit:
The Analysis of the output:
Can't reject the Null Hypothesis as p-value > 0.5.
Can't reject that Republican wins the election in Maine!
Probably because Republicans win almost half of the elections -Republicans won 6 times out of 14 times- historically back to 1960.
However, the confidence interval does not contain zero which is conflicting with the p-value?!

Exercise 7 Single mean hypothesis

The R code:

```
#Based upon the values of the votes from the following sources
#https://ballotpedia.org/Presidential_election_in_Maine,_2016
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_2004
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_2000
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1996
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1992
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1988
#https://en.wikipedia.org/wiki/United_States_presidential_election,_19
84
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1984
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1980
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1976
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1972
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_
Maine,_1968
#https://en.wikipedia.org/wiki/United_States_presidential_election,_19
64
#https://en.wikipedia.org/wiki/United_States_presidential_election,_19
60

#Again running the Null Hypothesis as Republican will win by
#collecting the Republican votes in the elections from 2012 till 1988
RepVote=c(292276,295273,330201,286616,186378,206504,307131,336500,2385
22,236320,256458,169254,118701,240608)
t.test(RepVote)

# Two-sided t-test
t.test(RepVote, alternative = "two.sided")
```

The output from the console:

```
> RepVote=c(292276,295273,330201,286616,186378,206504,307131,336500,238522,23
6320,256458,169254,118701,240608)
> t.test(RepVote)

	One Sample t-test

data:  RepVote
t = 14.722, df = 13, p-value = 1.738e-09
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 213359.6 286746.4
sample estimates:
mean of x 
   250053 

>
> # Two-sided t-test
> t.test(RepVote, alternative = "two.sided")

	One Sample t-test

data:  RepVote
t = 14.722, df = 13, p-value = 1.738e-09
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 213359.6 286746.4
sample estimates:
mean of x 
   250053 
```

For extra credit:

The Analysis of the output:

Can reject the Null Hypothesis as p-value < 0.5.

Can reject that the Republican wins the election in Maine!

The R code for less historical data:

```
#Based upon the values of the votes from the following sources
#https://ballotpedia.org/Presidential_election_in_Maine,_2016
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_Maine,_2004
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_Maine,_2000
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_Maine,_1996
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_Maine,_1992
#https://en.wikipedia.org/wiki/United_States_presidential_election_in_Maine,_1988
#https://en.wikipedia.org/wiki/United_States_presidential_election,_1984

#Again running the Null Hypothesis as Republican will win by
#collecting the Republican votes in the elections from 2012 till 1988
RepVote=c(292276,295273,330201,286616,186378,206504,307131)
t.test(RepVote)

# Two-sided t-test
t.test(RepVote, alternative = "two.sided")
```

The output from the console:

```
> RepVote=c(292276,295273,330201,286616,186378,206504,307131)
> t.test(RepVote)

        One Sample t-test

data:  RepVote
t = 13.363, df = 6, p-value = 1.087e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 222237.2 321871.1
sample estimates:
mean of x 
 272054.1 

> t.test(RepVote, alternative = "two.sided")

        One Sample t-test

data:  RepVote
t = 13.363, df = 6, p-value = 1.087e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 222237.2 321871.1
sample estimates:
mean of x 
 272054.1 
```

For extra credit:
The Analysis of the output:
Can reject the Null Hypothesis as p-value < 0.5.
Can reject that the Republican wins the election in Maine!

I just wanted to verify that my calculations are correct; hence, took less historical data where the Democrats won more times. The calculations are valid as the more historic data:
-back to 1960- ; the p-value is 1.738e-09
while with less historic data -back to 1988- the p-value is higher 1.087e-05
because the democrats rate was higher as well.
Nevertheless, in both cases we can reject the null hypothesis.

Exercise 8 chi-square test

The R code:

```
#http://www.maine.gov/sos/cec/elec/data/index.html
#http://www.maine.gov/sos/cec/elec/data/r-e-active.pdf
#I followed the instruction here to create the table:
#http://www.cyclismo.org/tutorial/R/tables.html

MaineVoters2016=matrix(c(264673,319679,398180),ncol=3)
colnames(MaineVoters2016) <-c("R","D","others")
MaineVoters2016 <- as.table(MaineVoters2016)
round(prop.table(MaineVoters2016), 2)  # Show as proportions w/2
digits

chi1 <- chisq.test(MaineVoters2016)  # Save tests as object "chi1"
chi1  # Check results

#compare it with 2012 percentages
#The source of 2012 percentages is:
http://www.270towin.com/states/Maine
#and from here
https://en.wikipedia.org/wiki/United_States_presidential_election_in_M
aine,_2012
chi2 <- chisq.test(MaineVoters2016, p = c(.41, 0.563, .027))
chi2
```

The output from the console:

```
> MaineVoters2016=matrix(c(264673,319679,398180),ncol=3)
> colnames(MaineVoters2016) <-c("R","D","others")
> MaineVoters2016 <- as.table(MaineVoters2016)
> round(prop.table(MaineVoters2016), 2)  # Show as proportions w/2 digits
     R    D others
A 0.27 0.33   0.41
> 
> 
> chi1 <- chisq.test(MaineVoters2016)  # Save tests as object "chi1"
> chi1  # Check results

	Chi-squared test for given probabilities

data:  MaineVoters2016
X-squared = 27492, df = 2, p-value < 2.2e-16

> 
> #compare it with 2012 percetages
> #The source of 2012 percetanges is: http://www.270towin.com/states/Maine
> #and from here https://en.wikipedia.org/wiki/United_States_presidential_ele
ction_in_Maine,_2012
> chi2 <- chisq.test(MaineVoters2016, p = c(.41, 0.563, .027))
> chi2

	Chi-squared test for given probabilities

data:  MaineVoters2016
X-squared = 5352600, df = 2, p-value < 2.2e-16
```

chi-square test with green party details

The R code:

```
#http://www.maine.gov/sos/cec/elec/data/index.html
#http://www.maine.gov/sos/cec/elec/data/r-e-active.pdf
#I followed the instruction here to create the table:
#http://www.cyclismo.org/tutorial/R/tables.html

MaineVoters2016=matrix(c(264673,319679,39133,359047),ncol=4)
colnames(MaineVoters2016) <-c("R","D","GRE","Unenrolled")
MaineVoters2016 <- as.table(MaineVoters2016)
round(prop.table(MaineVoters2016), 2)  # Show as proportions w/2
digits

chi1 <- chisq.test(MaineVoters2016)  # Save tests as object "chi1"
chi1  # Check results

#compare it with 2012 percentages
#The source of 2012 percentages is:
http://www.270towin.com/states/Maine
#and from here
https://en.wikipedia.org/wiki/United_States_presidential_election_in_M
aine,_2012
chi2 <- chisq.test(MaineVoters2016, p = c(.41, 0.563, .011, 0.016))
chi2
```

The output from the console:

```
> colnames(MaineVoters2016) <-c("R","D","GRE","Unenrolled")
> MaineVoters2016 <- as.table(MaineVoters2016)
> MaineVoters2016
       R      D    GRE Unenrolled
A 264673 319679  39133     359047
> MaineVoters2016=matrix(c(264673,319679,39133,359047),ncol=4)
> colnames(MaineVoters2016) <-c("R","D","GRE","Unenrolled")
> MaineVoters2016 <- as.table(MaineVoters2016)
> round(prop.table(MaineVoters2016), 2)  # Show as proportions w/2 digits
     R    D  GRE Unenrolled
A 0.27 0.33 0.04       0.37
>
>
> chi1 <- chisq.test(MaineVoters2016)  # Save tests as object "chi1"
> chi1  # Check results

        Chi-squared test for given probabilities

data:  MaineVoters2016
X-squared = 249760, df = 3, p-value < 2.2e-16

> chi2 <- chisq.test(MaineVoters2016, p = c(.41, 0.563, .011, 0))
Error in chisq.test(MaineVoters2016, p = c(0.41, 0.563, 0.011, 0)) :
  probabilities must sum to 1.
```

```
> chi2
Error: object 'chi2' not found
> chi2 <- chisq.test(MaineVoters2016, p = c(.41, 0.563, .011, 0.016))
> chi2

	Chi-squared test for given probabilities

data:  MaineVoters2016
X-squared = 7718200, df = 3, p-value < 2.2e-16
```

Exercise 9 Decision Tree

The R code:

```
D<-c(401306)
R<-c(292276)
other<-c(19598)
winparty<-c("D")
partyTree <- data.frame(D,R,other,winparty)
str(partyTree)
fit <- rpart( winparty ~ D + R + other, method="class",
data.frame(partyTree))
printcp(fit)
summary(fit)

tr <- tree( winparty ~ D + R + other, method="class",
data.frame(partyTree))
# see summary
summary(tr)
#plot tree
plot(tr)

require("party")
# grow tree
ct <- ctree( winparty ~ D + R + other, method="class",
data.frame(partyTree))
# see summary
summary(ct)
#plot tree
plot(ct)
```

The output from the console:

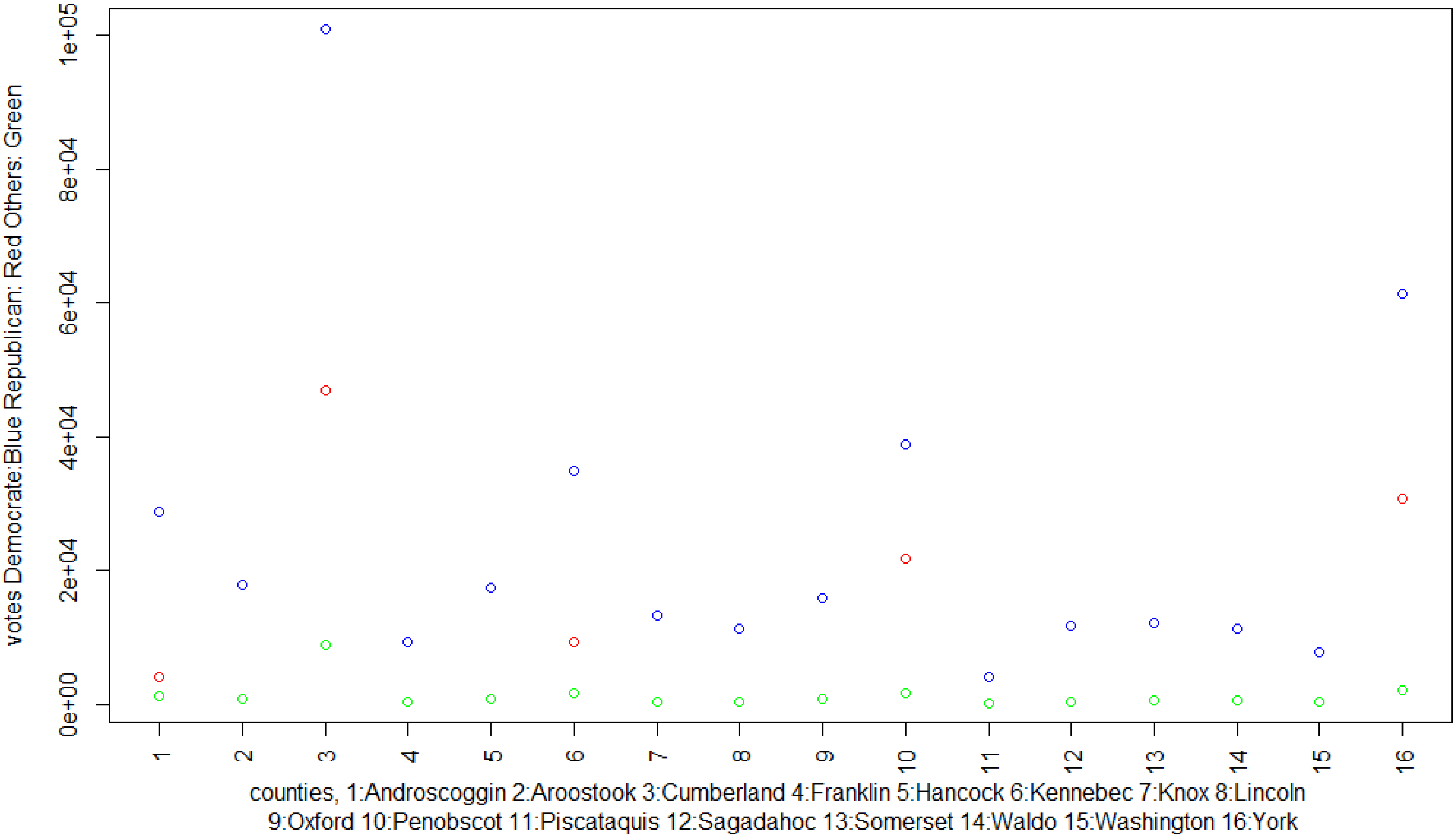
```
> D<-c(401306)
> R<-c(292276)
> other<-c(19598)
> partyTree <- data.frame(D,R,other)
> str(partyTree)
'data.frame':   1 obs. of  3 variables:
 $ D    : num 401306
 $ R    : num 292276
 $ other: num 19598
> library("rpart", lib.loc="~/R/win-library/3.3")
> winparty<-c("D")
> partyTree <- data.frame(D,R,other,winparty)
> str(partyTree)
'data.frame':   1 obs. of  4 variables:
 $ D       : num 401306
 $ R       : num 292276
 $ other   : num 19598
 $ winparty: Factor w/ 1 level "D": 1
> fit <- rpart( winparty ~ D + R + other, method="class", data.frame(partyTre
e))
```

```
> printcp(fit)

Classification tree:
rpart(formula = winparty ~ D + R + other, data = data.frame(partyTree), 
    method = "class")

Variables actually used in tree construction:
character(0)

Root node error: 0/1 = 0

n= 1 

   CP nsplit rel error xerror xstd
1 NaN      0       NaN    NaN  NaN
> summary(fit)
Call:
rpart(formula = winparty ~ D + R + other, data = data.frame(partyTree), 
    method = "class")
  n= 1 

   CP nsplit rel error xerror xstd
1 NaN      0       NaN    NaN  NaN

Node number NA: NA observations
Error in if (ff$complexity[i] < cp || is.leaf[i]) cat("\n") else cat(",    co
mplexity param=",  : 
  missing value where TRUE/FALSE needed
> tr <- tree( winparty ~ D + R + other, method="class", data.frame(partyTree)
)
> summary(tr)

Classification tree:
tree(formula = winparty ~ D + R + other, data = data.frame(partyTree), 
    method = "class")
Variables actually used in tree construction:
character(0)
Number of terminal nodes:  1 
Residual mean deviance:  NaN = 0 / 0 
Misclassification error rate: 0 = 0 / 1 
> plot(tr)
Error in plot.tree(tr) : cannot plot singlenode tree
> ct <- ctree( winparty ~ D + R + other, method="class", data.frame(partyTree
))
Error in ctree(winparty ~ D + R + other, method = "class", data.frame(partyTr
ee)) : 
  unused argument (method = "class")
```

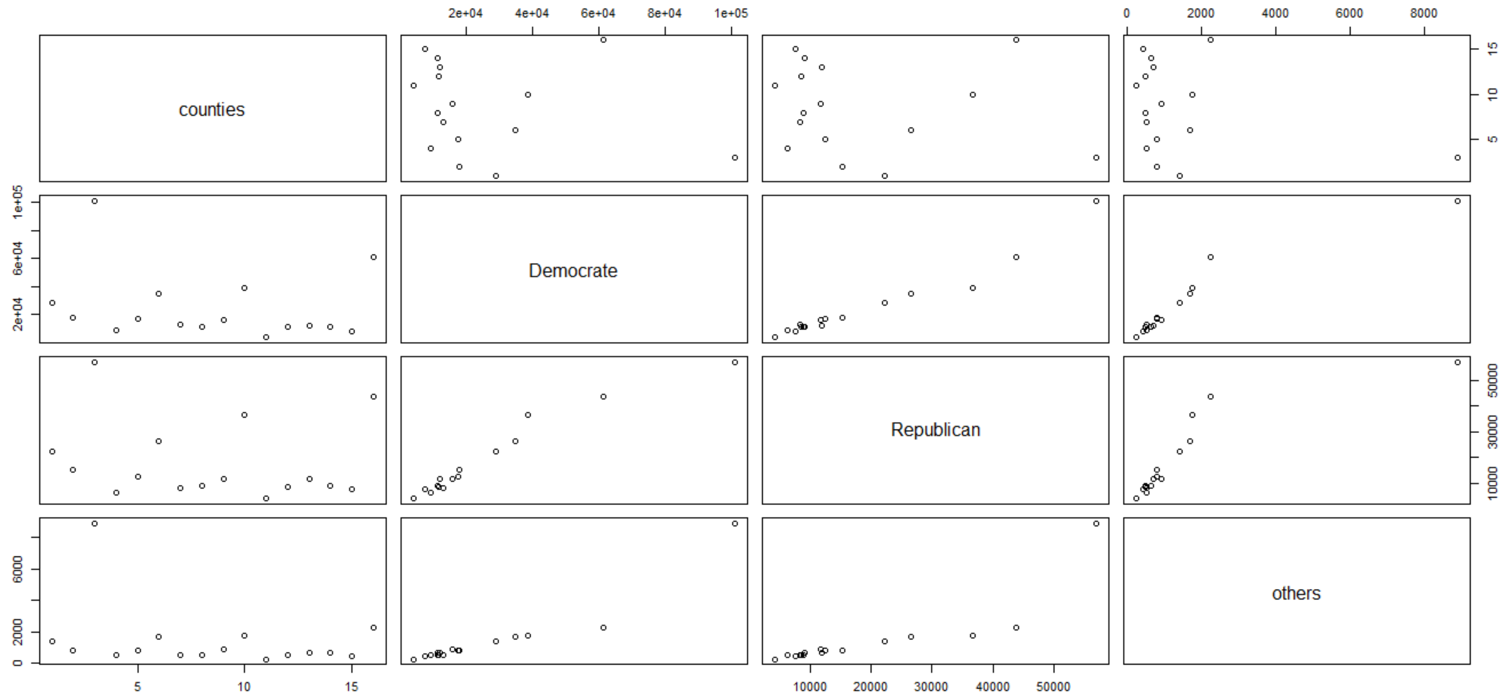
Exercise 10 Scatter plot of the counties and parties

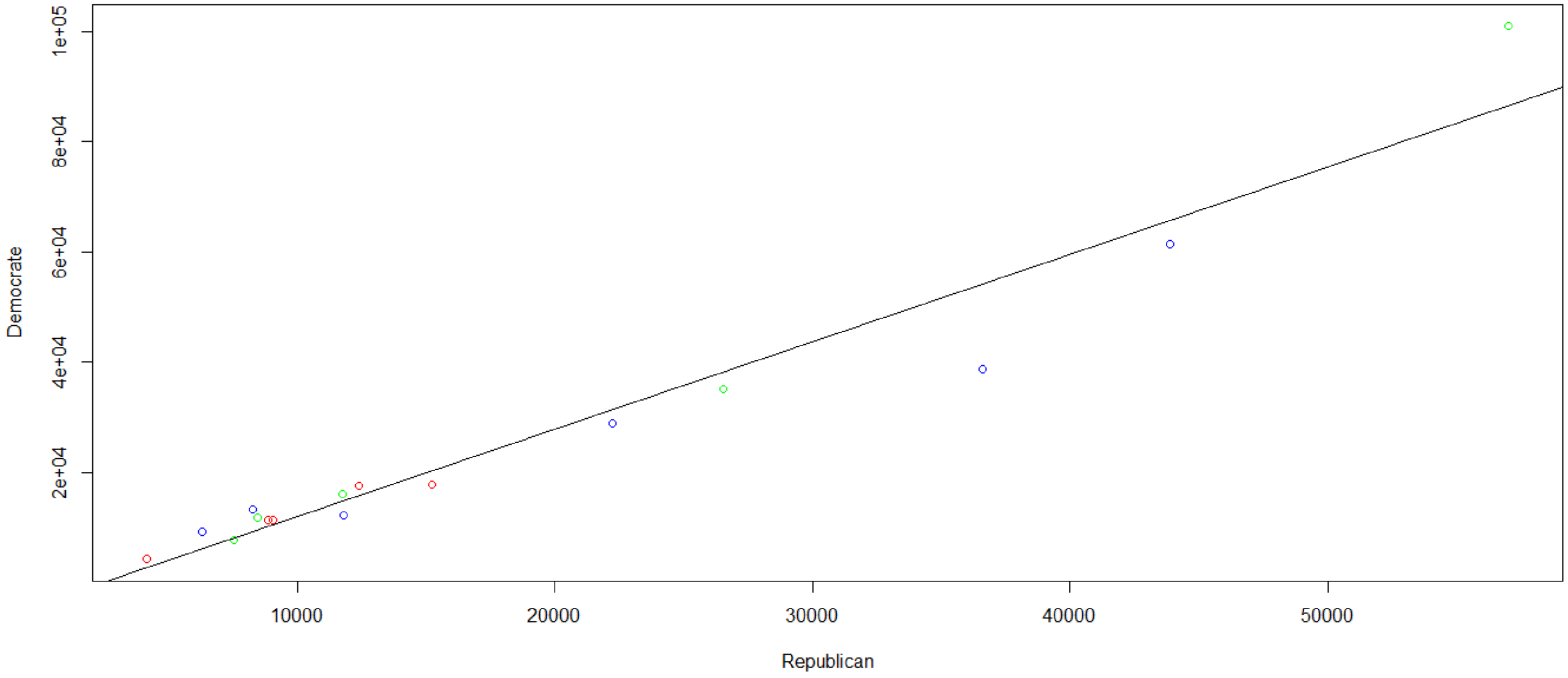
```
counties=c("Androscoggin","Aroostook","Cumberland","Franklin","Hancock
","Kennebec","Knox","Lincoln","Oxford","Penobscot","Piscataquis","Saga
dahoc","Somerset","Waldo","Washington","York");
Democrate=c(28912,17844,100977,9292,17533,35022,13227,11298,16052,3881
6,4168,11821,12211,11292,7797,61492)
Republican=c(22210,15229,57008,6305,12398,26506,8262,8884,11766,36592,
4168,8429,11798,9055,7539,43879)
others=c(1417,793,8910,515,807,1697,513,486,916,1763,245,484,696,629,4
20,2256)
MaineCounties=data.frame(Democrate,Republican,others,count = c(1:16));
counties1=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
plot(counties1,Democrate,col="blue",xlab = "counties, 1:Androscoggin
2:Aroostook 3:Cumberland 4:Franklin 5:Hancock 6:Kennebec 7:Knox
8:Lincoln \n 9:Oxford 10:Penobscot 11:Piscataquis 12:Sagadahoc
13:Somerset 14:Waldo 15:Washington 16:York", ylab="votes
Democrate:Blue Republican: Red Others:
Green",xaxt="n",ylim=c(1417,1e+05))
axis(1, at = seq(1, 16, by = 1), las=2)
par(new=TRUE)
plot(counties1,Republican,col="red",ylim=c(2e+04,1e+05),axes=FALSE,ann
=FALSE)
par(new=TRUE)
plot(counties1,others,col="green",ylim=c(1417,1e+05),axes=FALSE,ann=FA
LSE)


library(ggplot2)
library(reshape2)


countychoice=c("D","D","D","D","D","D","D","D","D","D","R","D","D","D"
,"D","D");
#countychoice=c(1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1)
MaineCounties=data.frame(counties,Democrate,Republican,others);
plot(Democrate~Republican,col=c("blue","red","green"))

vote.mod1 = lm(Democrate~Republican, data = MaineCounties)
abline(lm(Democrate~Republican))

#-------------------------------------------
#Scatter plot as in Lynda

MaineCounties=data.frame(Republican,Democrate,others,counties);
MaineCounties[1:3]

# Modified scatterplot matrices

# Create palette with RColorBrewer
require("RColorBrewer")
```

```
display.brewer.pal(3, "Pastel2")

# Put histograms on the diagonal (from "pairs" help)
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y,  ...)
  # Removed "col = "cyan" from code block; original below
  # rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

pairs(MaineCounties[1:3],
      panel = panel.smooth,  # Optional smoother
      main = "Scatterplot Maine Counties",
      diag.panel = panel.hist,
      pch = 16,
      col = brewer.pal(3, "Pastel2"))

# Similar with "car" package
# Gives kernal density and rugplot for each variable
library(car)
scatterplotMatrix(~Democrate+Republican+others | counties,
                  data = MaineCounties,
                  col = brewer.pal(3, "Dark2"),
                  main="Scatterplot Matrix for MAine Counties Data
Using \"car\" Package")
```

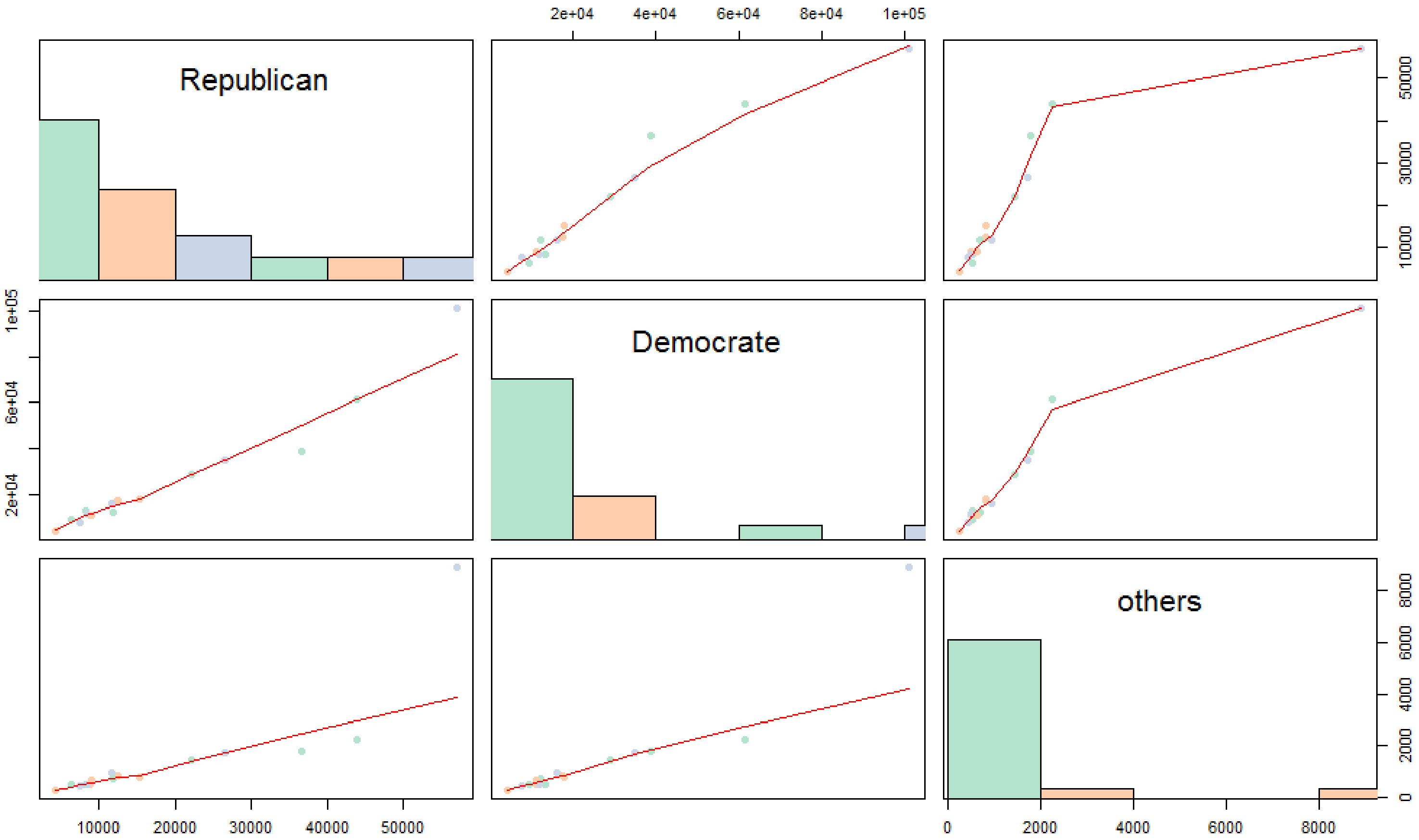

votes Democrate:Blue Republican: Red Others: Green
1e+05
8e+04
6e+04
4e+04
2e+04
0e+00
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
counties, 1:Androscoggin 2:Aroostook 3:Cumberland 4:Franklin 5:Hancock 6:Kennebec 7:Knox 8:Lincoln
9:Oxford 10:Penobscot 11:Piscataquis 12:Sagadahoc 13:Somerset 14:Waldo 15:Washington 16:York

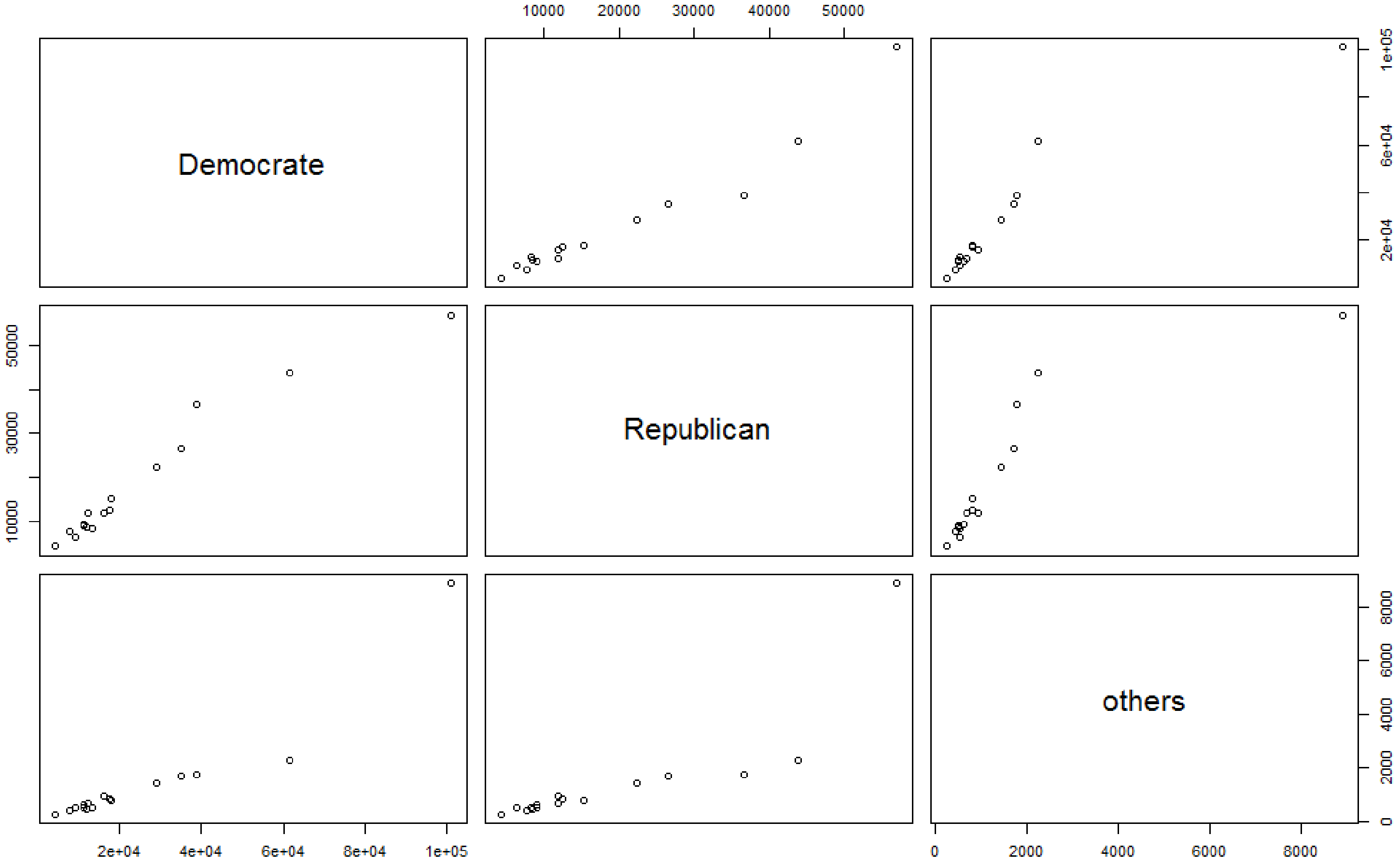
```
More plots:
```


counties
Democrate
Republican
others
2e+04
4e+04
6e+04
8e+04
1e+05
0
2000
4000
6000
8000
15
10
5
1e+05
6e+04
2e+04
50000
30000
10000
6000
2000
0
5
10
15
10000
20000
30000
40000
50000


1e+05
8e+04
6e+04
4e+04
2e+04
Democrate
10000
20000
30000
40000
50000
Republican

Scatterplot Maine Counties




Democrate
Republican
others
10000
20000
30000
40000
50000
1e+05
6e+04
2e+04
50000
30000
10000
8000
6000
4000
2000
0
2e+04
4e+04
6e+04
8e+04
1e+05
0
2000
4000
6000
8000