Hypothesis Testing

Problems

Gordon Erlebacher, 2014

What we will do

- Rather than have a drawn out discussion on hypothesis testing, let us solve some practical problems with R and in so doing, develop some skills with R
  - conversion from/to factor columns
  - read/write to files, subset extraction, plotting

To and from factors

```
df = data.frame(c(“a”,”b”,”c”), c(1,2,3))

str(df)    # 1 factor, 1 vector

dff = df

dff[1] = as.character(df[[1]])   # or
dff[[1]] = as.character(df[[1]])

str(dff)   # no more factors

dff[2] = as.factor(dff[[2]])

str(dff)  # 2nd column is now a factor
```

Naming factor colums

f = factor(c("high", "low"))

f = factor(c(hi="high", lo="low"))

HOWEVER

f$hi   does not work!!
This approach only works with lists, data.frames, etc.
but NOT with vectors.

f["hi"] does work!!!

Problem 1

- Are Cambridge Scholars taller than criminals?
- Use the dataset:

criminal_cambridge.RData

Question to answer

- Do Cambridge students, and criminals have significantly different heights?

Procedure

- Analyze (look at) the data file
- Read the data file
- Establish a factor to distinguish criminals from scholars
- Establish a Null Hypothesis H0
- Use t.test() function to answer the question

criminal_cambridge.RData

```
"source" "height.cm" "middle.finger.cm"
"1" "criminal" 170.18 12.7
"2" "criminal" 165.1 11.4
"3" "criminal" 167.64 11.9
"4" "criminal" 167.64 12.1
```

The file has column headers, elements are separated by spaces.

This suggests the use of ***read.table***

```
> tab = read.table("criminal_cambridge.RData")
> head(tab,5)
    source height.cm middle.finger.cm
1 criminal    170.18             12.7
2 criminal    165.10             11.4
3 criminal    167.64             11.9
4 criminal    167.64             12.1
5 criminal    167.64             11.7
```

The data seems to be read in properly

Check the data

Use the commands:

class, str, summary

to get a “feel” for the data

```
> class(tab)
[1] "data.frame"
> str(tab)
'data.frame':   3996 obs. of  3 variables:
 $ source          : Factor w/ 2 levels "cambridge","criminal": 2 2 2 2 2 2 2 2 2 2 ...
 $ height.cm       : num  170 165 168 168 168 ...
 $ middle.finger.cm: num  12.7 11.4 11.9 12.1 11.7 12 12 11.7 12 11.2 ...
> summary(tab)
       source        height.cm       middle.finger.cm
  cambridge: 996   Min.   :142.2   Min.   : 9.50
  criminal :3000   1st Qu.:162.6   1st Qu.:11.20
                   Median :167.6   Median :11.50
                   Mean   :168.4   Mean   :11.55
                   3rd Qu.:172.7   3rd Qu.:11.90
                   Max.   :195.6   Max.   :13.50
                                   NA's   :996
```

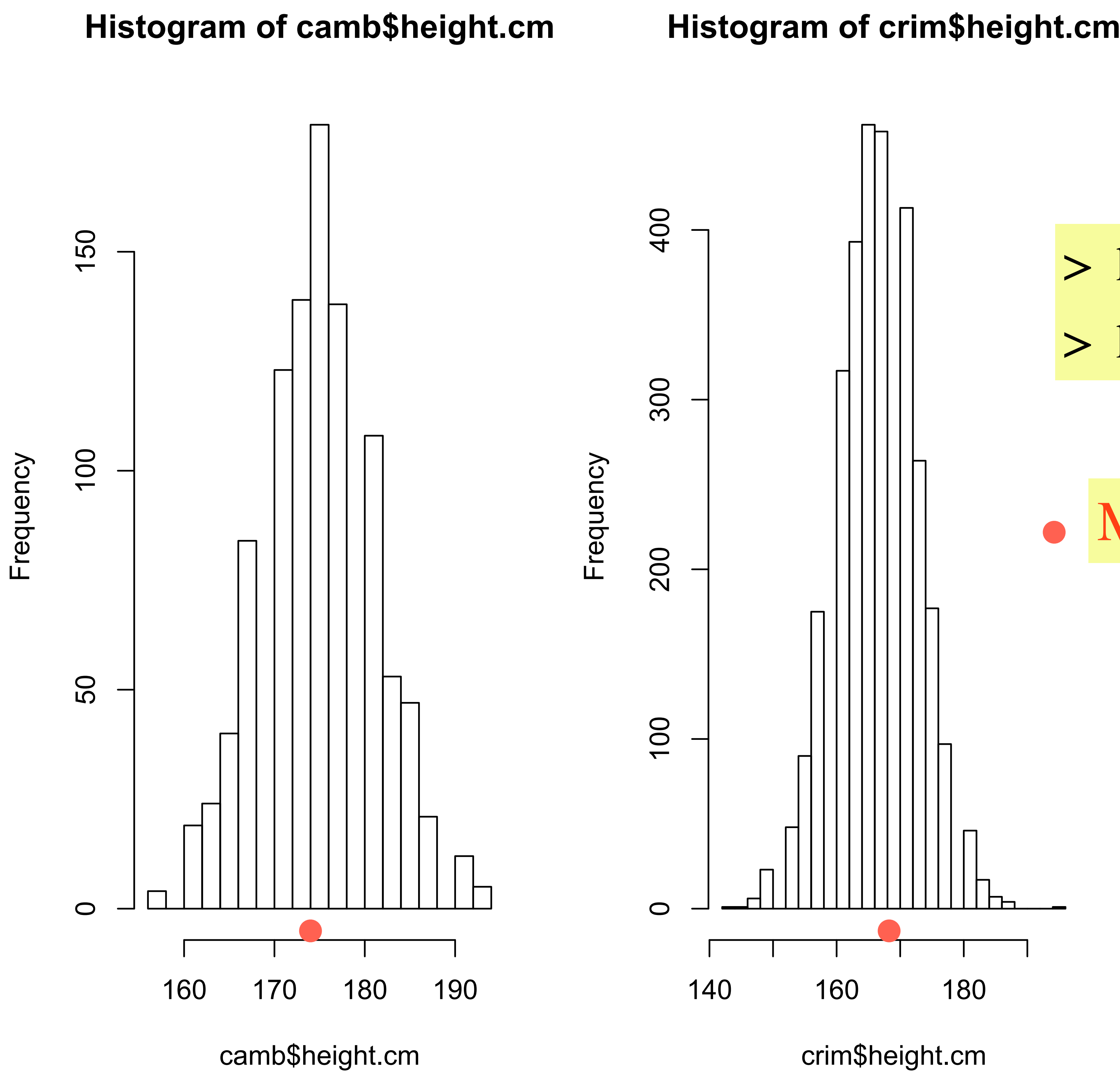
Approach 1

- Create a vector for each factor
- Use t.test(vector1, vector2) to establish whether or not the mean height of criminals and Cambridge students are significantly different

```
> height = tab[tab$source == "criminal",]
> crim = tab[tab$source == "criminal",]
> camb = tab[tab$source == "cambridge",]

> t.test(crim$height.cm, camb$height.cm)
	Welch Two Sample t-test
data:  crim$height.cm and camb$height.cm
t = -36.1876, df = 1705.635, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -9.051622 -8.120879
sample estimates:
mean of x mean of y
 166.3014  174.8877
```

p-value < 0.05, so the means <u>are significantly different</u>.

The 95% confidence of the difference between the two means *does not include zero*.




```
> hist(camb$height.cm,25)
> hist(crim$height.cm,25)
```

● Mean value

Are these distributions normal?

```
> shapiro.test(crim$height.cm)

        Shapiro-Wilk normality test

data:  crim$height.cm
W = 0.9867, p-value = 3.549e-16
```

p-value < 0.05; the distribution is significantly different from normal

How was the dataset generated

- Source of criminal data
  - data(crimtab)
  - ?crimtab
- Source of scholar height data
  - a paper that provides the number of students in different height categories
- These two sources were combined to produce one output file (criminal_cambridge.RData)

Generating script criminal_cambridge_generate.R

http://people.sc.fsu.edu/~gerlebacher/course/comp_apps_psych_s2013/code/criminal_cambridge_generate.R

Problem 2

Using the maze data set

Create a factor variable (low/high) that distinguishes between individuals who had lower than the median number of errors in their first attempt at the maze (low) and those that had a number of errors equal to or above the median (high). Is there a difference in the mean time to solve the maze on the first attempt for individuals with low/high initial errors. The last attempt? Provide graphs and statistical analysis to support your claim.

10pts. Create list data structure that contains the appropriately labeled mean, median, range, and standard deviation of final maze solution times for the high and low initial error groups. Neatly print the contents of these structures to the screen.

Provide a script that does all of the above.

Information about the dataset

http://opl.apa.org/Experiments/About/AboutMazes.aspx

Break problem into smaller parts

First: look at the *maze* file

Maze_UniversityOfIllinois.csv

TODO List

- ?Maze_UniversityOfIllinois

- look up file on Google

Look at file structure

vi Maze_UniversityOfIllinois.csv **# use an editor to look at the file**

UserID,Gender,ClassID,Age,DateTaken,TimeInExperiment,T1T,T1E,T2T,T2E,T3T,T3E,T4T
,T4E,T5T,T5E,T6T,T6E,T7T,T7E,T8T,T8E,T9T,T9E,T10T,T10E,T11T,T11E,T12T,T12E,T1
3T,T13E,T14T,T14E,T15T,T15E
.....

What can one say about the file?

- elements are separated by commas

Use read.csv("Maze_UniversityOfIllinois.csv")

```
> df = read.csv("Maze_UniversityOfIllinois.csv")
> class(df)       # data.frame
```

Check the data.frame

```
> head(df,2)
  UserID Gender ClassID Age DateTaken TimeInExperiment    T1T T1E    T2T T2E
1 105995      F    4812  17 9/23/2011           171.31 14.886   1  9.198   0
2 106850      M    4812  17 9/28/2011           439.03 35.786   1 19.477   2
     T3T T3E    T4T T4E    T5T T5E    T6T T6E    T7T T7E    T8T T8E    T9T T9E
1 11.798   1  7.426   1  8.693   1 13.126   3  6.862   0  6.938   0  7.277   0
2 23.680   3 15.792   2 13.034   1 17.026   2 16.320   2 21.837   2 17.405   2
   T10T T10E  T11T T11E  T12T T12E  T13T T13E   T14T T14E  T15T T15E
1 8.081    0 7.342    0 6.886    0 5.936    0 10.584    2 6.437    0
2 9.399    0 8.991    0 8.321    0 7.642    0  7.579    0 6.403    0
```

Structure of data.frame

```
> str(df)
'data.frame':	92 obs. of  36 variables:
 $ UserID          : int  105995 106850 107822 108023 109503 110315 111124 112382 112395 113090 ...
 $ Gender          : Factor w/ 2 levels "F","M": 1 2 1 1 2 1 1 1 2 2 ...
 $ ClassID         : int  4812 4812 4812 4812 4812 4812 4812 4812 4812 4812 ...
 $ Age             : int  17 17 17 18 19 17 16 17 17 17 ...
 $ DateTaken       : Factor w/ 77 levels "1/28/2012","10/1/2009",..: 74 77 13 15 3 6 10 14 14 31 ...
 $ TimeInExperiment: num  171 439 241 264 168 ...
 $ T1T             : num  14.9 35.8 16.5 24.7 16.4 ...
 $ T1E             : int  1 1 0 1 2 2 2 2 0 2 ...
 $ T2T             : num  9.2 19.5 10.2 9.1 21 ...
```

2 factor columns

remainder are integer or numerical columns

How many rows/columns?

```
> ncol(df)
[1] 36
> nrow(df)
[1] 92
> length(df)
[1] 36    # nb of columns
```

column names

```
> colnames(df)
 [1] "UserID"           "Gender"           "ClassID"          "Age"
 [5] "DateTaken"        "TimeInExperiment" "T1T"              "T1E"
 [9] "T2T"              "T2E"              "T3T"              "T3E"
[13] "T4T"              "T4E"              "T5T"              "T5E"
[17] "T6T"              "T6E"              "T7T"              "T7E"
[21] "T8T"              "T8E"              "T9T"              "T9E"
[25] "T10T"             "T10E"             "T11T"             "T11E"
[29] "T12T"             "T12E"             "T13T"             "T13E"
[33] "T14T"             "T14E"             "T15T"             "T15E"
>
```

Vector of strings, colnames(df)[2] returns “Gender”

Still do not know meaning of column labels

Next step

Meaning of Columns

T1T	Time in 1st trial of maze
T2T	Time in 2nd trial of maze
T3T	
...	

T1E	Number of errors in 1st trial of maze
T2E	Number of errors in 2nd trial of maze
T3E	
...	

Which columns are needed?

- Simplify the table to only what is needed

Create a factor variable (low/high) that distinguishes between individuals who had lower than the **median number of errors in their first attempt** at the maze (low) and those that had a number of errors equal to or above the median (high).

We will work with first and last attempt, so we would need the columns T1T, T1E, T15T, T15E

Simplify data.frame?

We will work with first and last attempt, so we would need the columns T1T, T1E, T15T, T15E

```
> df1 = cbind(df$T1T, df$T1E, df$T15T, df$T15E)          # class is matrix
> df1 = data.frame(T1T=df$T1T, T1E=df$T1E, T15T=df$T15T, T15E=df$T15E)  # class is data.frame
```

Alternative (less typing)

```
> df1 = with(df, data.frame(T1T, T1E, T15T, T15E))        # class is data.frame
```

Take a look at df1 :

```
> head(df1)
       T1T T1E   T15T T15E
[1,] 14.886   1  6.437    0
[2,] 35.786   1  6.403    0
[3,] 16.524   0  8.832    0
[4,] 24.653   1 11.366    0
[5,] 16.392   2  5.767    0
[6,] 26.374   2 13.344    2
```

We notice:

1) time in the maze seems to have decreased between first and 15th trial

2) number of errors also seems to have decreased

We are ready to solve the first part of the problem

Create a factor variable (low/high) that distinguishes between individuals who had lower than the median number of errors in their first attempt at the maze (low) and those that had a number of errors equal to or above the median (high).

1) compute the median
2) factor “low”:   lower than median number oferrors in df1$T1E
3) factor “high”: equal or above number of errors in df1$T1E

```
> str(df1)
'data.frame':   92 obs. of  4 variables:
 $ T1T : num  14.9 35.8 16.5 24.7 16.4 ...
 $ T1E : int  1 1 0 1 2 2 2 2 0 2 ...
 $ T15T: num  6.44 6.4 8.83 11.37 5.77 ...
 $ T15E: int  0 0 0 0 0 2 1 0 0 0 ...
```

Create the factor

```
lo.hi = factor(c(lo="low", hi="high"))
```

```
> df$err1 = hi.lo["lo" ]

> head(df1, 3)

  T1T      T1E  T15T   T15E     err1

1 14.886     1   6.437    0        low
2 35.786     1   6.403    0        low
3 16.524     0   8.832    0        low
```

Change last column to high if err1 is greater or equal to the median

```
> df2 = cbind(df1,err1="low")   # last col is a factor
> df2 = cbind(df1, err1=lo.hi[1]) # last col is a factor

# created a data.frame with an additional column

# Using str, one sees that the new column is a factor with
# one level.
```

Always check with **str**

```
> df2 = data.frame(df1, err1="low")
> str(df2)
'data.frame': 92 obs. of  5 variables:
 $ T1T : num  14.9 35.8 16.5 24.7 16.4 ...
 $ T1E : int  1 1 0 1 2 2 2 2 0 2 ...
 $ T15T: num  6.44 6.4 8.83 11.37 5.77 ...
 $ T15E: int  0 0 0 0 0 2 1 0 0 0 ...
 $ err1: Factor w/ 1 level "low": 1 1 1 1 1 1 1 1 1 1 ...
```

Next step: change appropriate values of err1 to “high”

```
> df2$err1[df2$T1E >= m] = hi.lo[2];

> str(df2$err1)

 Factor w/ 2 levels "high","low": 1 1 2 1 1 1 1 1 2 1 ...

> head(df2,4)

     T1T T1E   T15T T15E err1
1 14.886   1  6.437    0 high
2 35.786   1  6.403    0 high
3 16.524   0  8.832    0  low
4 24.653   1 11.366    0 high
```

Factor

Next step

Is there a difference in the mean time to solve the maze on the first attempt for individuals with low/high initial errors. **The last attempt?** Provide graphs and statistical analysis to support your claim.

First attempt

- Use a Student t-test using the t.test() function

```
low =  df2[df2$err1=="low",”T1E”]
high = df2[df2$err1 == “high”, “T1E”]
```

Null Hypothesis: the means of *low* and *high* are equal

t.test(low, high)

Welch Two Sample t-test

data: low and high

t = -16.8809, df = 72, **p-value < 2.2e-16**

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-1.991119 -1.570525

sample estimates:

mean of x mean of y

0.000000 1.780822

The two means are significantly different. The difference cannot be attributed to chance.

There is a 5% chance that the two means are actually the same and the difference is just due to the specific samples chosen.

Second Attempt

- Instead of extracting two different sets, take advantage of the fact that we have a **factor column** : df2$err1
- Use a different form of t.test

```
t.test(vector ~ factor)
```

Caveat: the factor can only have two levels

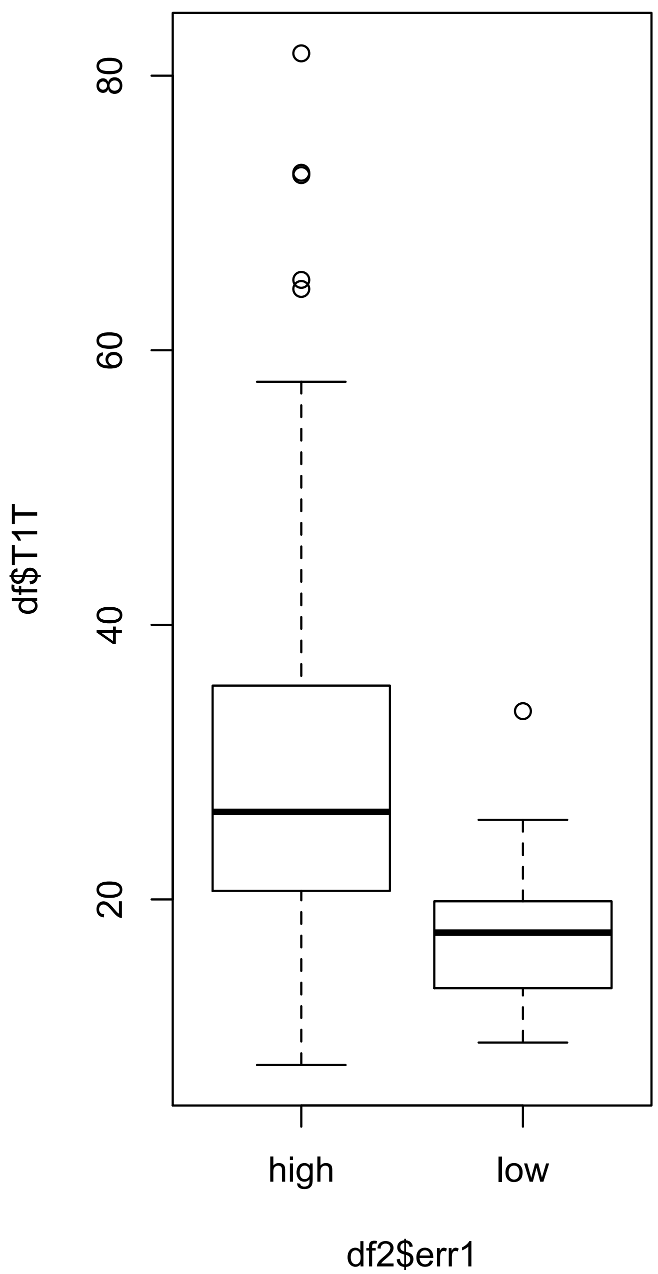
```
> t.test(df2$T1E ~ df2$err1)

        Welch Two Sample t-test

data:  df2$T1E by df2$err1
t = 16.8809, df = 72, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.570525 1.991119
sample estimates:
mean in group high  mean in group low
          1.780822           0.000000
```

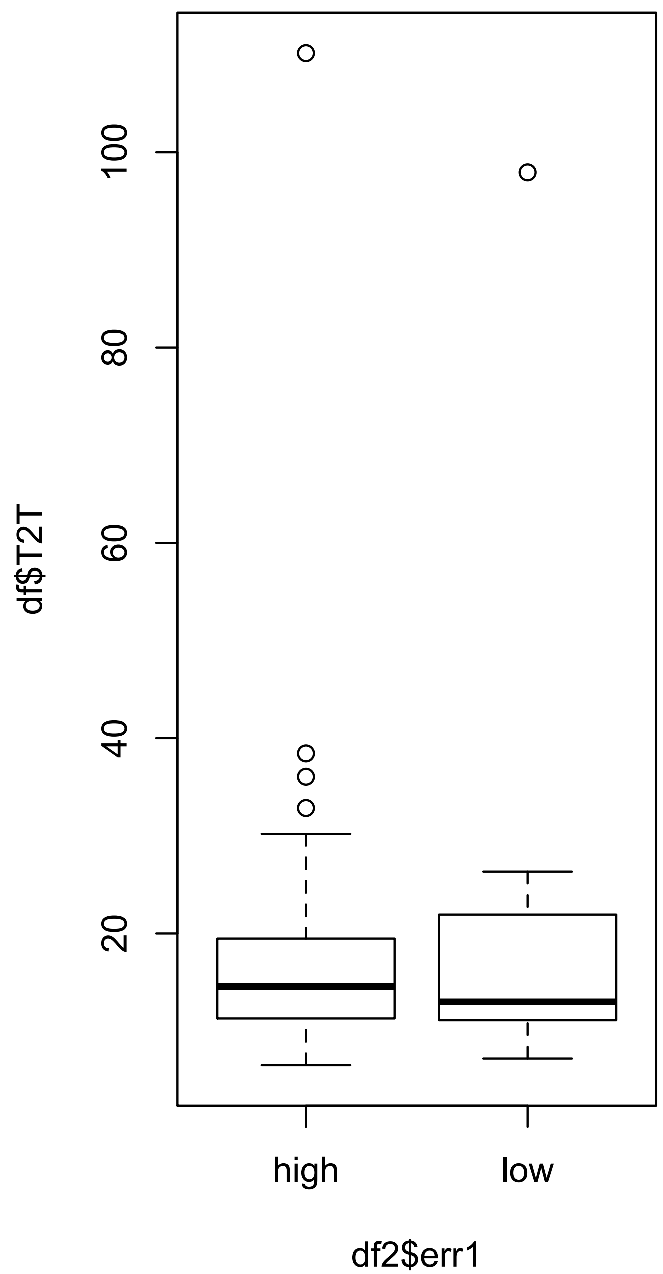
Same analysis at 15th trial

```
> t.test(df2$T15E ~ df2$err1)

        Welch Two Sample t-test

data:  df2$T15E by df2$err1

t = 1.8195, df = 54.342, p-value = 0.07434

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

 -0.02456637  0.50762333

sample estimates:

mean in group high  mean in group low

         0.4520548          0.2105263
```

$p > 0.05$, so that chance is sufficient to explain the discrepency in the means. The null hypothesis cannot be rejected. there is not enough evidence.

Look at times it took to solve the maze





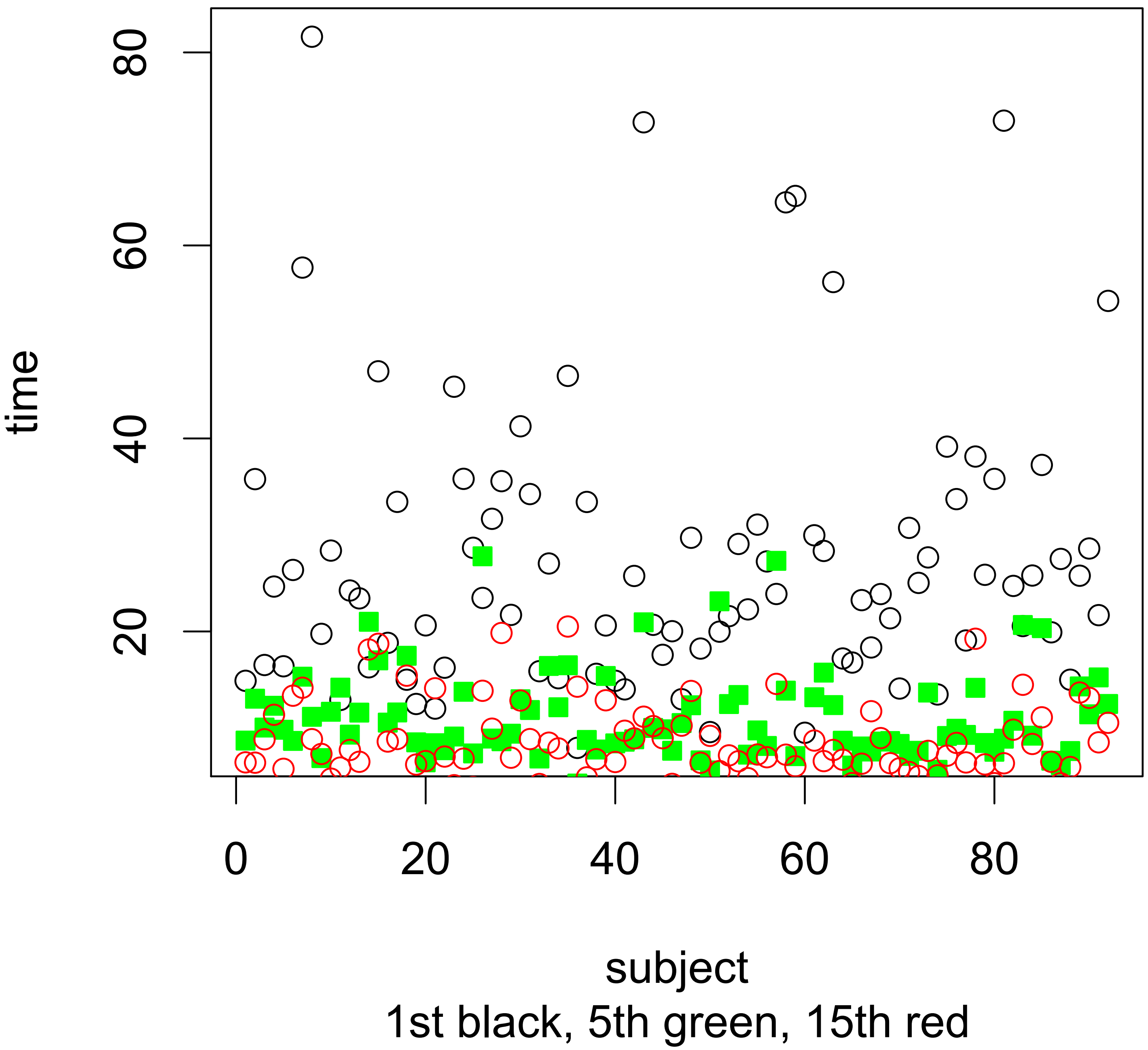
plot(vector ~ factor)

generates box plots

1) the time in the maze decreased overall by the 15th trial

2) The groups above and below the median error in trial 1 are now performing similarly

Plot the sample

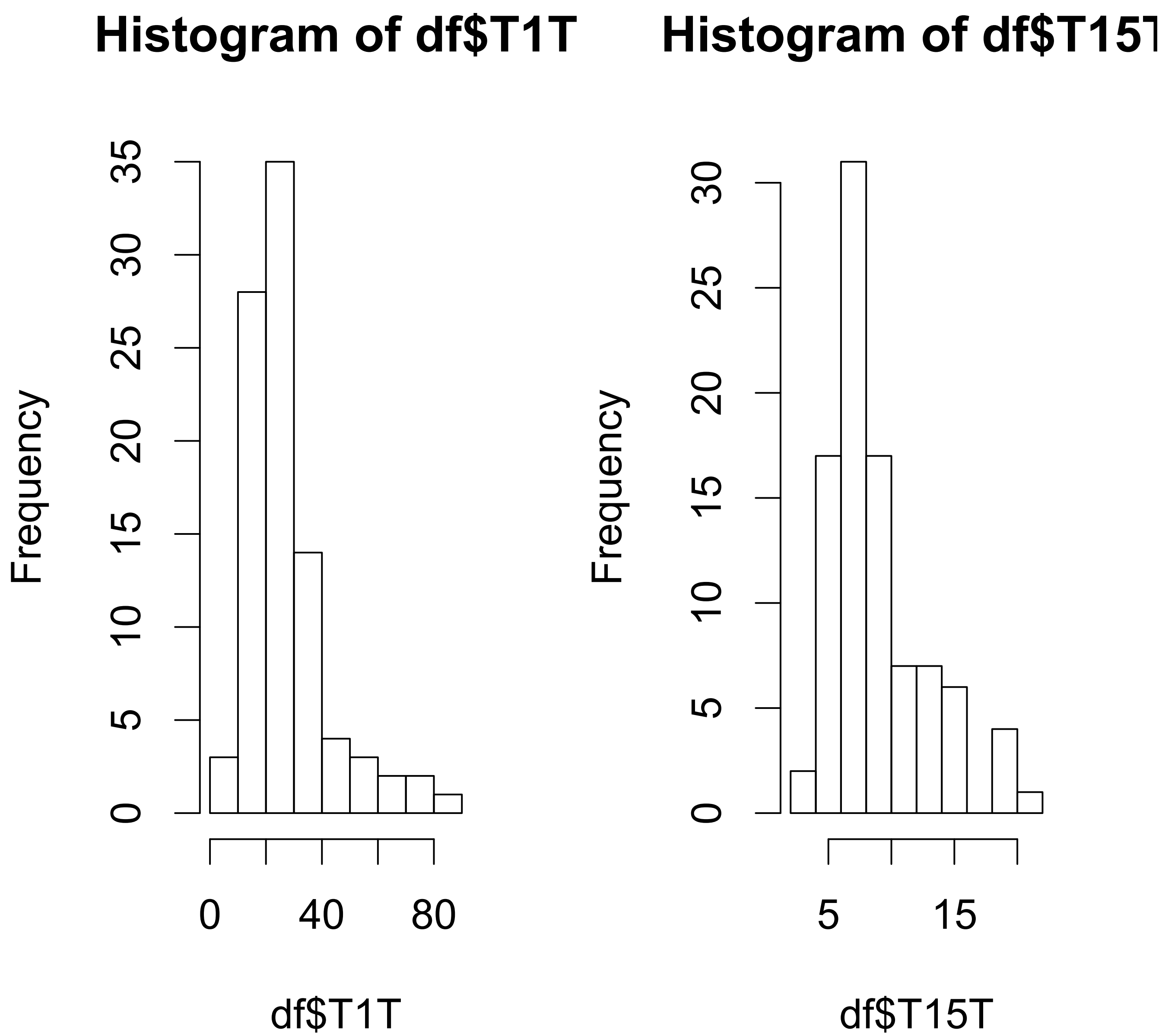



```
par(cex=1.5)

plot(X$T1T,col='black', sub="1st black, 5th green, 15th red",
   xlab="subject", ylab="time")

points(X$T5T,col='green', pch=22, bg='green')

points(X$T15T,col='red')
```

Clearly the time to complete the maze goes down as the number of trials goes up.

Histograms

```
> plot(mfrow=c(1,2))
> hist(df$T1T)
> hist(df$T15T)
```



Histograms do not appear to be normal

Check for normality

```
> shapiro.test(df$T1T)

	Shapiro-Wilk normality test

data:  df$T1T
W = 0.8438, p-value = 1.975e-08
```

Very small p indicates that df$T1T is “significantly” different from normal