Assignment 7.1

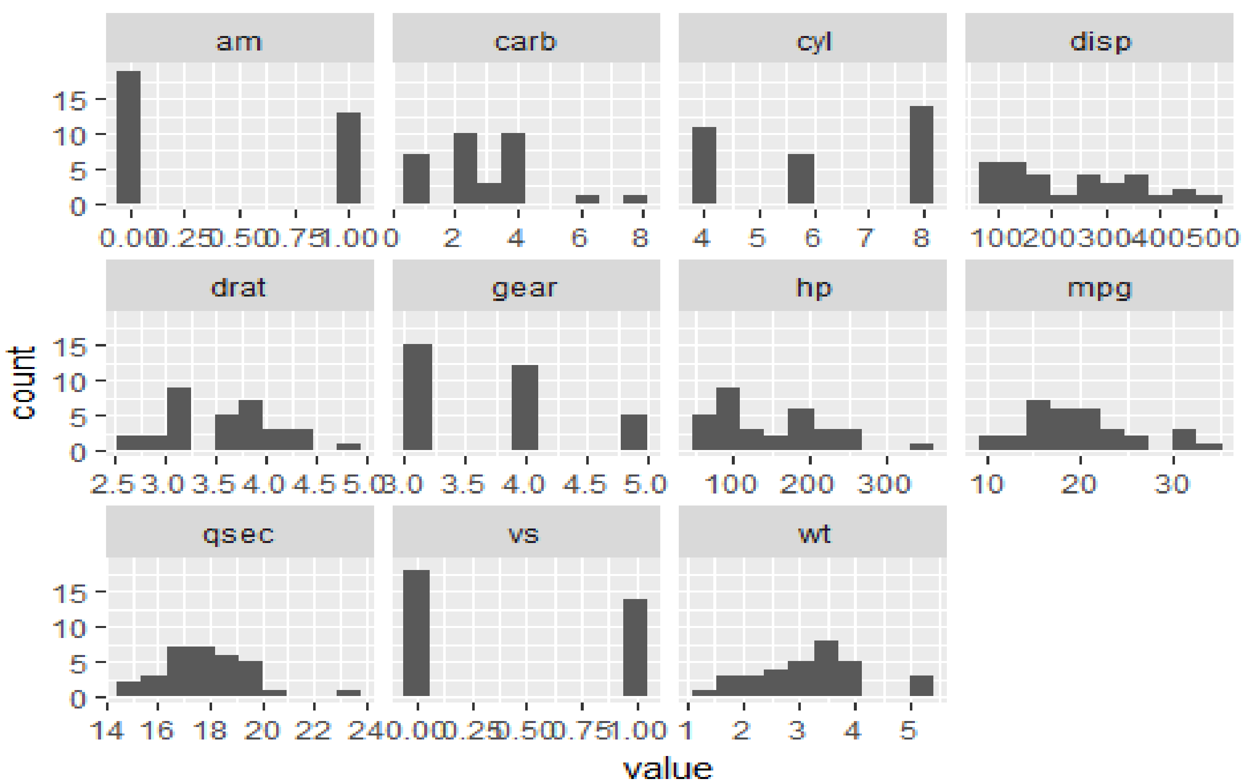
1. Histogram for all variables in a dataset mtcars. Write a program to create histograms for all columns.

2. Check the probability distribution of all variables in mtcars

3. Write a program to create boxplot for all variables.

1. Histogram for all variables in a dataset mtcars. Write a program to create histograms for all columns.

```
library(tidyr)
library(ggplot2)
ggplot(gather(mtcars), aes(value)) + geom_histogram(bins = 10) +
facet_wrap(~key, scales = 'free_x')
```



2. Check the probability distribution of all variables in mtcars.

```{r}
library(readr)
 mtcars <- read_csv("C:/Sourav/R/mtcars.csv")
 cars <- mtcars
 print(head(cars))
 column_means <- colMeans(cars)        # Get the means of each column
 print(column_means)                   # Check means
 center_matrix <- matrix( rep(column_means, nrow(cars)),   # Repeat the column means
nrow=nrow(cars),
ncol=ncol(cars),
byrow = TRUE)
 # Construct row by row
 centered <- cars - center_matrix     # Subtract column means
 print( head( centered ))            # Check the new data set
 print(colMeans(centered))           # Check the new column means to confirm they are 0
sd(centered$mpg)
column_sds <- apply(centered,      # A matrix or data frame
MARGIN = 2,     # Operate on rows(1) or columns(2)
FUN = sd)       # Function to apply
print(column_sds)                  # Check standard deviations
scale_matrix <- matrix( rep(column_sds, nrow(cars)),      # Repeat the column sds
nrow=nrow(cars),
ncol=ncol(cars),
byrow = TRUE)
centered_scaled <- centered/scale_matrix       # Divide by column sds to scale the data
summary(centered_scaled)              # Confirm that variables are on similar scales
```

```
auto_scaled <- scale(cars,          # Numeric data object
center=TRUE,      # Center the data?
scale=TRUE)       # Scale the data?
summary(auto_scaled)      # Check the auto scaled data
normally_distributed <- rnorm(10000)   # Generate normally distributed data
hist(normally_distributed, breaks=30)  # Create a histogram of the distribution
skewed_right <- rexp(10000, 0.5)      # Generate skewed data
hist(skewed_right, breaks=50)        # Create a histogram of the distribution
log_transformed <- log(skewed_right+1)
hist(log_transformed, breaks=50)
cor(cars[,1:6])        # Check the pairwise correlations of 6 variables
pairs(cars[,1:6])
```

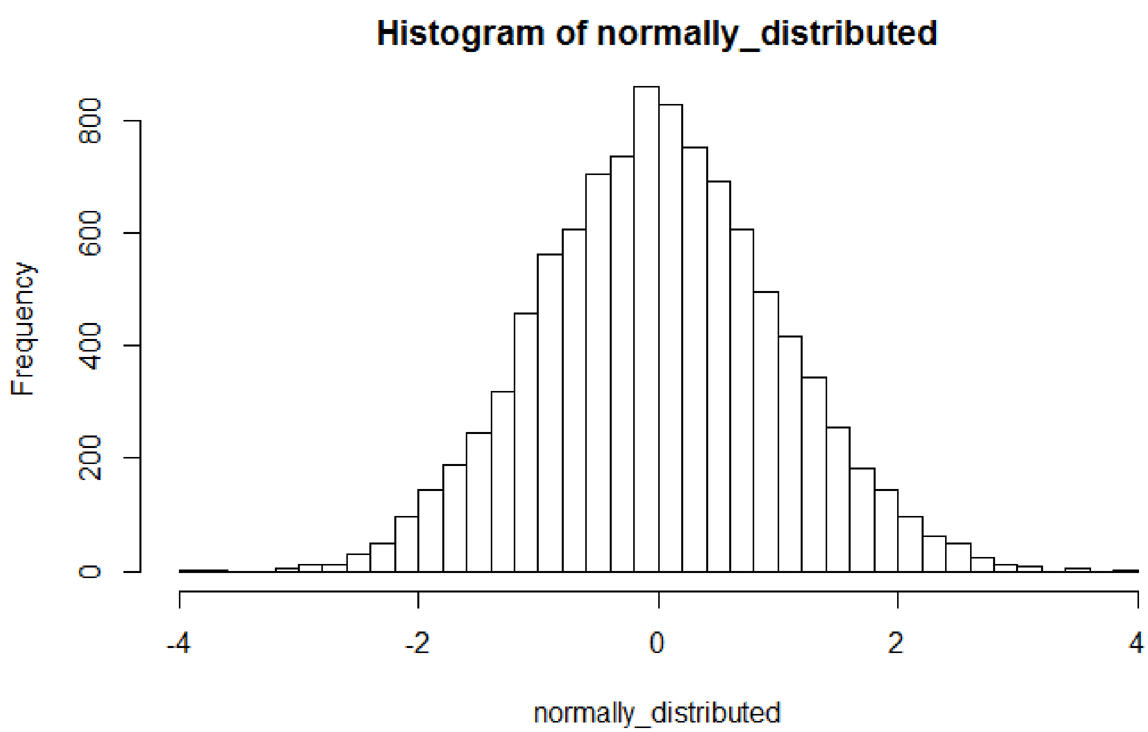
```
Max.   : 1.1899   Max.   : 1.7789   Max.   : 3.2117
      mpg               cyl               disp               hp
 Min.   :-1.6079   Min.   :-1.225   Min.   :-1.2879   Min.   :-1.3810
 1st Qu.:-0.7741   1st Qu.:-1.225   1st Qu.:-0.8867   1st Qu.:-0.7320
 Median :-0.1478   Median :-0.105   Median :-0.2777   Median :-0.3455
 Mean   : 0.0000   Mean   : 0.000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.4495   3rd Qu.: 1.015   3rd Qu.: 0.7688   3rd Qu.: 0.4859
 Max.   : 2.2913   Max.   : 1.015   Max.   : 1.9468   Max.   : 2.7466
      drat               wt               qsec                vs
 Min.   :-1.5646   Min.   :-1.7418   Min.   :-1.87401   Min.   :-0.868
 1st Qu.:-0.9661   1st Qu.:-0.6500   1st Qu.:-0.53513   1st Qu.:-0.868
 Median : 0.1841   Median : 0.1101   Median :-0.07765   Median :-0.868
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.000
 3rd Qu.: 0.6049   3rd Qu.: 0.4014   3rd Qu.: 0.58830   3rd Qu.: 1.116
 Max.   : 2.4939   Max.   : 2.2553   Max.   : 2.82675   Max.   : 1.116
       am               gear              carb
 Min.   :-0.8141   Min.   :-0.9318   Min.   :-1.1222
 1st Qu.:-0.8141   1st Qu.:-0.9318   1st Qu.:-0.5030
 Median :-0.8141   Median : 0.4236   Median :-0.5030
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 1.1899   3rd Qu.: 0.4236   3rd Qu.: 0.7352
 Max.   : 1.1899   Max.   : 1.7789   Max.   : 3.2117
            mpg        cyl       disp         hp       drat         wt
mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.6811719 -0.8676594
cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.6999381  0.7824958
disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.7102139  0.8879799
hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.4487591  0.6587479
drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.0000000 -0.7124406
wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.7124406  1.000000
```

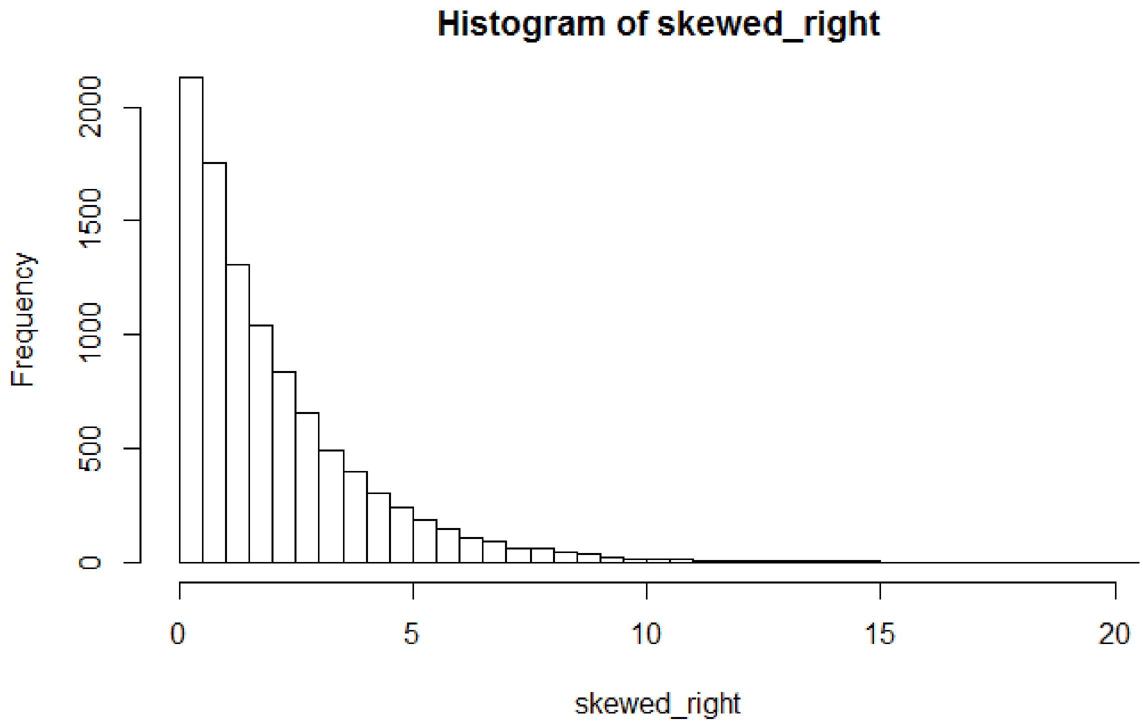
mpg <dbl>	cyl <int>	disp <dbl>	hp <int>	drat <dbl>	wt <dbl>	qsec <dbl>	vs <int>	am <int>	gear <int>
21.0	6	160	110	3.90	2.620	16.46	0	1	4
21.0	6	160	110	3.90	2.875	17.02	0	1	4
22.8	4	108	93	3.85	2.320	18.61	1	1	4
21.4	6	258	110	3.08	3.215	19.44	1	0	3
18.7	8	360	175	3.15	3.440	17.02	0	0	3
18.1	6	225	105	2.76	3.460	20.22	1	0	3

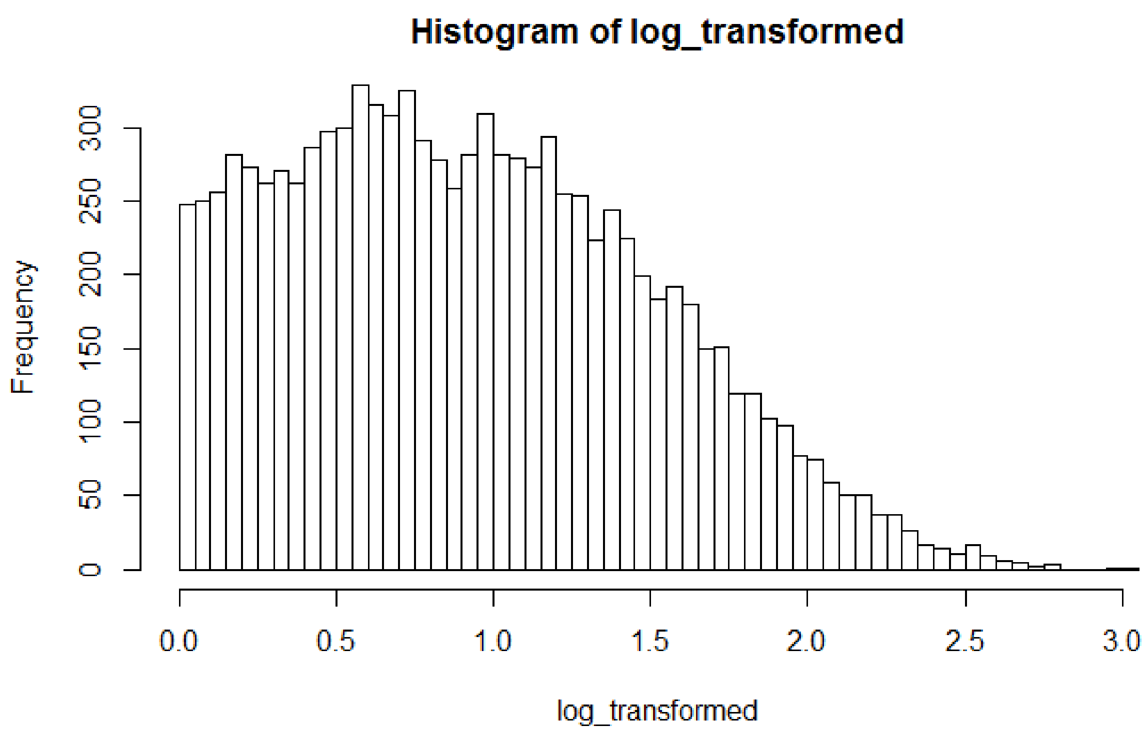
6 rows | 1-10 of 11 columns

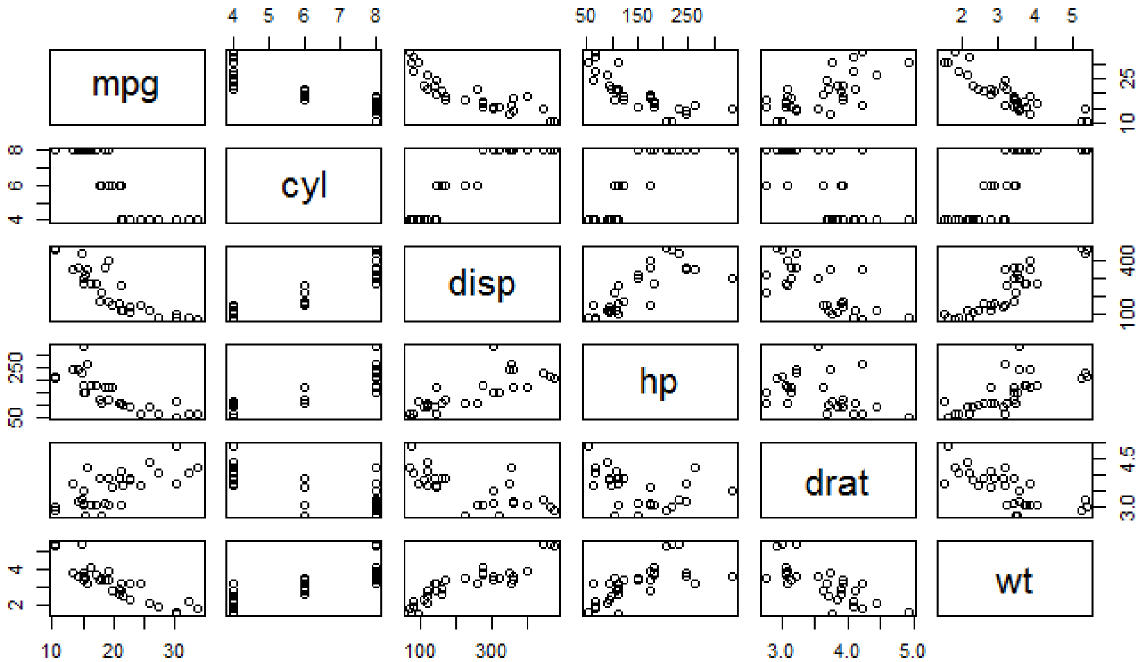
mpg <dbl>	cyl <dbl>	disp <dbl>	hp <dbl>	drat <dbl>	wt <dbl>	qsec <dbl>	
1	0.909375	-0.1875	-70.721875	-36.6875	0.3034375	-0.59725	-1.38875
2	0.909375	-0.1875	-70.721875	-36.6875	0.3034375	-0.34225	-0.82875
3	2.709375	-2.1875	-122.721875	-53.6875	0.2534375	-0.89725	0.76125
4	1.309375	-0.1875	27.278125	-36.6875	-0.5165625	-0.00225	1.59125
5	-1.390625	1.8125	129.278125	28.3125	-0.4465625	0.22275	-0.82875
6	-1.990625	-0.1875	-5.721875	-41.6875	-0.8365625	0.24275	2.37125

6 rows | 1-8 of 11 columns




Histogram of skewed_right
Frequency
0
500
1000
1500
2000
0
5
10
15
20
skewed_right


Histogram of log_transformed
Frequency
0
50
100
150
200
250
300
0.0
0.5
1.0
1.5
2.0
2.5
3.0
log_transformed




```
require(graphics)

pairs(mtcars, main = "mtcars data", gap = 1/4)

coplot(mpg ~ disp | as.factor(cyl), data = mtcars,

      panel = panel.smooth, rows = 1)

## possibly more meaningful, e.g., for summary() or bivariate plots:

mtcars2 <- within(mtcars, {

   vs <- factor(vs, labels = c("V", "S"))

   am <- factor(am, labels = c("automatic", "manual"))

   cyl  <- ordered(cyl)

   gear <- ordered(gear)

   carb <- ordered(carb)

})

summary(mtcars2)
```

```
#### generate subset: automatic and manual cars ####

cars_auto = subset(mtcars, am == 0)

cars_manu = subset(mtcars, am == 1)

# dimensions

dim(mtcars)

dim(cars_auto); dim(cars_manu)

mean(cars_auto$mpg); mean(cars_manu$mpg)

sd(cars_auto$mpg); sd(cars_manu$mpg)

(mean(cars_manu$mpg) - mean(cars_auto$mpg))/mean(cars_auto$mpg)

#### mpg plots ####

par(mfrow = c(2, 1))

hist(cars_auto$mpg, main = "Distribution mpg - automatic transmission", xlab = "mpg")

abline(v = mean(cars_auto$mpg), col = "red")

hist(cars_manu$mpg, main = "Distribution mpg - manual transmission", xlab = "mpg")

abline(v = mean(cars_manu$mpg), col = "red")

t.test(cars_manu$mpg, cars_auto$mpg, paired = F, var.equal = F)

#### Permutation test ####

# what if I shuffle the am groups and calculate the mean?


# get target variable and group vectors

y = mtcars$mpg

group = mtcars$am

y; group

# baseline group means and difference

baselineMeans = tapply(mtcars$mpg, mtcars$am, mean)

baselineMeansDiff = baselineMeans[2] - baselineMeans[1]
```

```
tStat = function(w, g) mean(w[g == 1]) - mean(w[g == 0])

observedDiff = tStat(y, group)

# check if function works - should be 0:

baselineMeansDiff - observedDiff

# execute shuffle:

permutations = sapply(1:100000, function(i) tStat(y, sample(group)))

# shuffle experiment results plots:

par(mfrow = c(2, 1), mar = c(4, 4, 2, 2))

hist(permutations, main = "Distribution of shuffled group mean differences") # distribution
of difference of averages of permuted groups

plot(permutations, type = "b", main = "Shuffled group mean trials", xlab = "trial", ylab =
"shuffled group mean differences", ylim = c(-14, 14))

abline(h = observedDiff, col = "red", lwd = 3)

mean(permutations > observedDiff)

#### generate subset: automatic and manual cars ####

cars_auto = subset(mtcars, am == 0)

cars_manu = subset(mtcars, am == 1)


#### Visual inspection of all covariates ####

pairs(mtcars)

#### 4 bivariate analysis: hp / wt / drat / disp ####

par(mfrow = c(2, 2), mar = c(2, 3, 2, 3))


# plot1

with(mtcars, plot(hp, mpg, type = "n", main = "mpg vs. hp - by transmission type")) # no
data

with(cars_auto, points(hp, mpg, col = "red", pch = 20))
```

```
with(cars_manu, points(hp, mpg, col = "blue", pch = 20))

legend("topright", pch = 20, col = c("red", "blue"), legend = c("auto", "manu")) # add legend

model1_auto = lm(mpg ~ hp, data = cars_auto)

model1_manu = lm(mpg ~ hp, data = cars_manu)

abline(model1_auto, col = "red", lwd = 2)

abline(model1_manu, col = "blue", lwd = 2)

abline(v = 175, lty = 2)


# plot2

with(mtcars, plot(wt, mpg, type = "n", main = "mpg vs. weight - by transmission type")) #
no data

with(cars_auto, points(wt, mpg, col = "red", pch = 20))

with(cars_manu, points(wt, mpg, col = "blue", pch = 20))

legend("topright", pch = 20, col = c("red", "blue"), legend = c("auto", "manu")) # add legend

abline(v = 3.2, lty = 2)


# plot 3

with(mtcars, plot(drat, mpg, type = "n", main = "mpg vs. drat - by transmission type")) # no
data

with(cars_auto, points(drat, mpg, col = "red", pch = 20))

with(cars_manu, points(drat, mpg, col = "blue", pch = 20))

legend("topright", pch = 20, col = c("red", "blue"), legend = c("auto", "manu")) # add legend

model2_auto = lm(mpg ~ drat, data = cars_auto)

model2_manu = lm(mpg ~ drat, data = cars_manu)

abline(model2_auto, col = "red", lwd = 2)

abline(model2_manu, col = "blue", lwd = 2)

abline(v = 175, lty = 2)
```

```
# plot 4

with(mtcars, plot(disp, mpg, type = "n", main = "mpg vs. disp - by transmission type")) # no data

with(cars_auto, points(disp, mpg, col = "red", pch = 20))

with(cars_manu, points(disp, mpg, col = "blue", pch = 20))

legend("topright", pch = 20, col = c("red", "blue"), legend = c("auto", "manu")) # add legend

labels = with(mtcars, paste(as.character(disp), as.character(mpg), sep = ",")) # generate point labels

with(mtcars, text(disp, mpg, labels = labels, cex = 0.7, pos = 2))

abline(v = 167.6, lty = 2)

### analyse covariance matrix for regressor selection:

z <- cor(mtcars)

require(lattice)

# only am

data = mtcars

data$am = as.factor(data$am)

model2 = lm(mpg ~ am, data = data)


# get results

summary(model2)

#### model selection using leaps ####

library(leaps)

data = mtcars

data$log_mpg = log(data$mpg) # add log of y

#### method 1. best fit ####

regfit.full = regsubsets(log_mpg ~. , data = data, nvmax = 10)

reg.summary = summary(regfit.full)

reg.summary
```

how I select the optimal number of variables?

plot(reg.summary$cp, xlab = "Number of variables", ylab = "cp", type = "b")

regfit.fwd = regsubsets(log_mpg ~ ., data = data, nvmax = 10, method = "forward")

summary(regfit.fwd)

plot(regfit.fwd, scale = "Cp")

lm with all variables / no split

prepare data

data = mtcars

data$am = as.factor(data$am)

model1 = lm(mpg ~ ., data = data)

get results

summary(model1)

plot residual analysis

par(mfrow = c(2, 2))

plot(model1)

plot hist

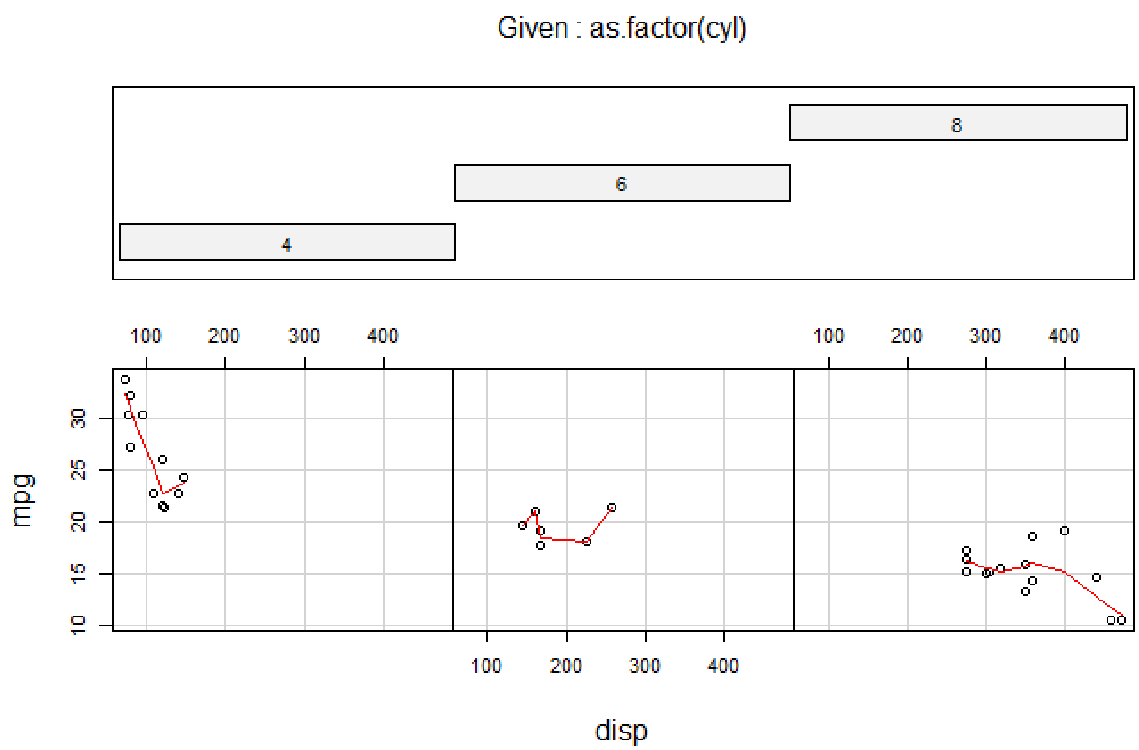
par(mfrow = c(1, 1))

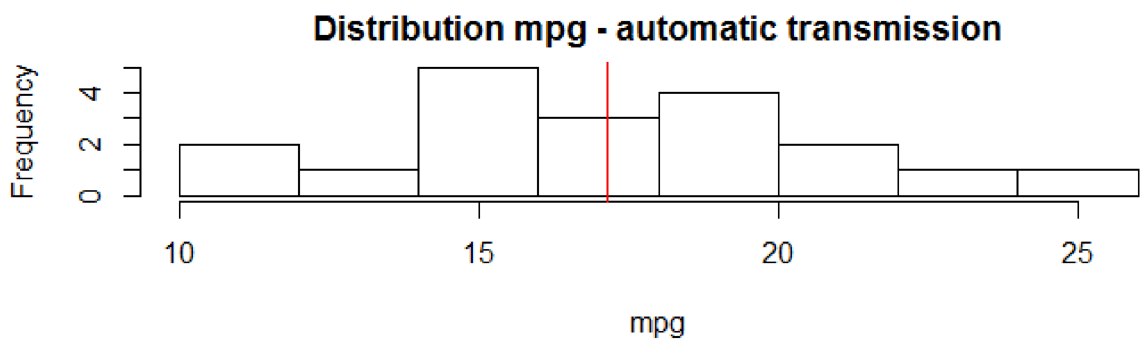
hist(model1$residuals)# normality test on residuals

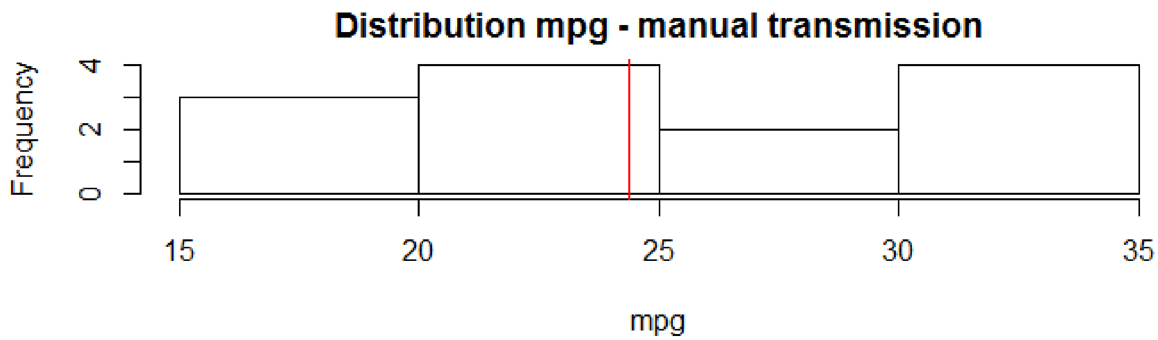
shapiro.test(model1$residuals)

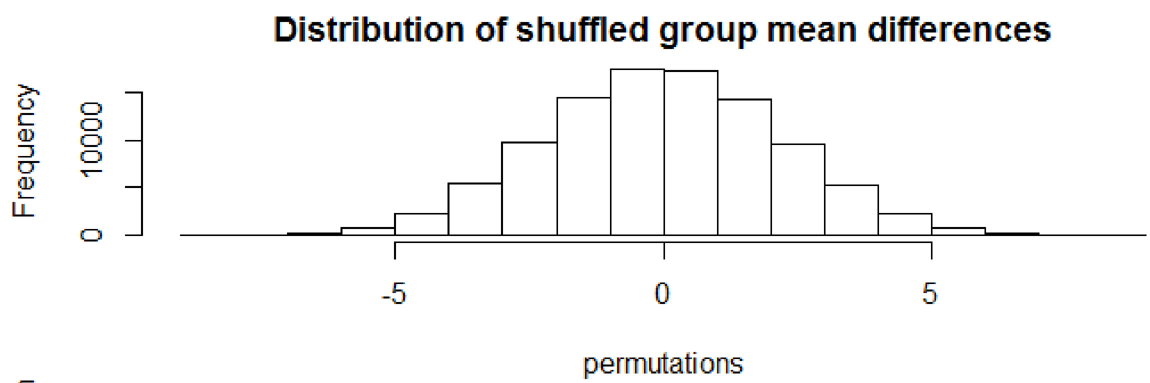
```
Min      1Q  Median      3Q     Max
-3.4506 -1.6044 -0.1196  1.2193  4.6271

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.30337   18.71788   0.657   0.5181
cyl         -0.11144    1.04502  -0.107   0.9161
disp         0.01334    0.01786   0.747   0.4635
```

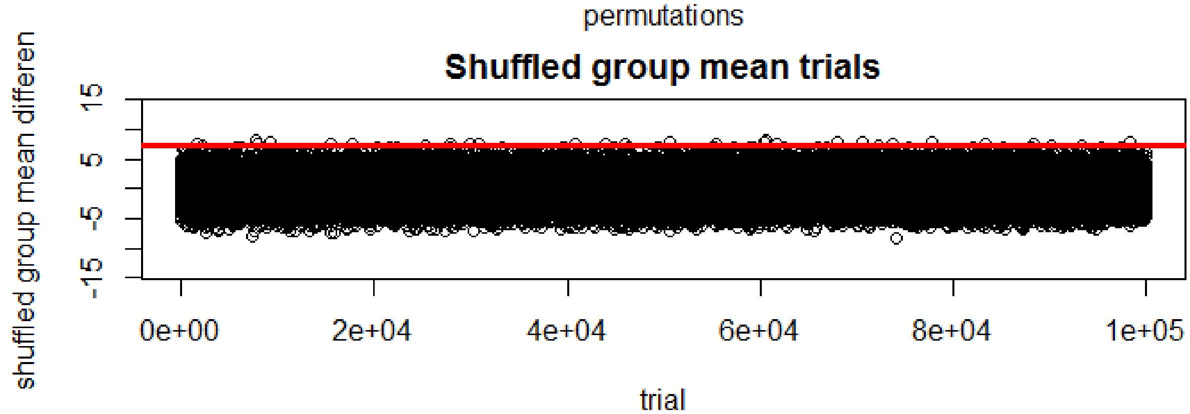
```
hp          -0.02148    0.02177  -0.987    0.3350
drat         0.78711    1.63537   0.481    0.6353
wt          -3.71530    1.89441  -1.961    0.0633 .
qsec         0.82104    0.73084   1.123    0.2739
vs           0.31776    2.10451   0.151    0.8814
am1          2.52023    2.05665   1.225    0.2340
gear         0.65541    1.49326   0.439    0.6652
carb        -0.19942    0.82875  -0.241    0.8122
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.65 on 21 degrees of freedom
Multiple R-squared:  0.869,    Adjusted R-squared:  0.8066
F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07


        Shapiro-Wilk normality test

data:  model1$residuals
W = 0.95694, p-value = 0.2261
```

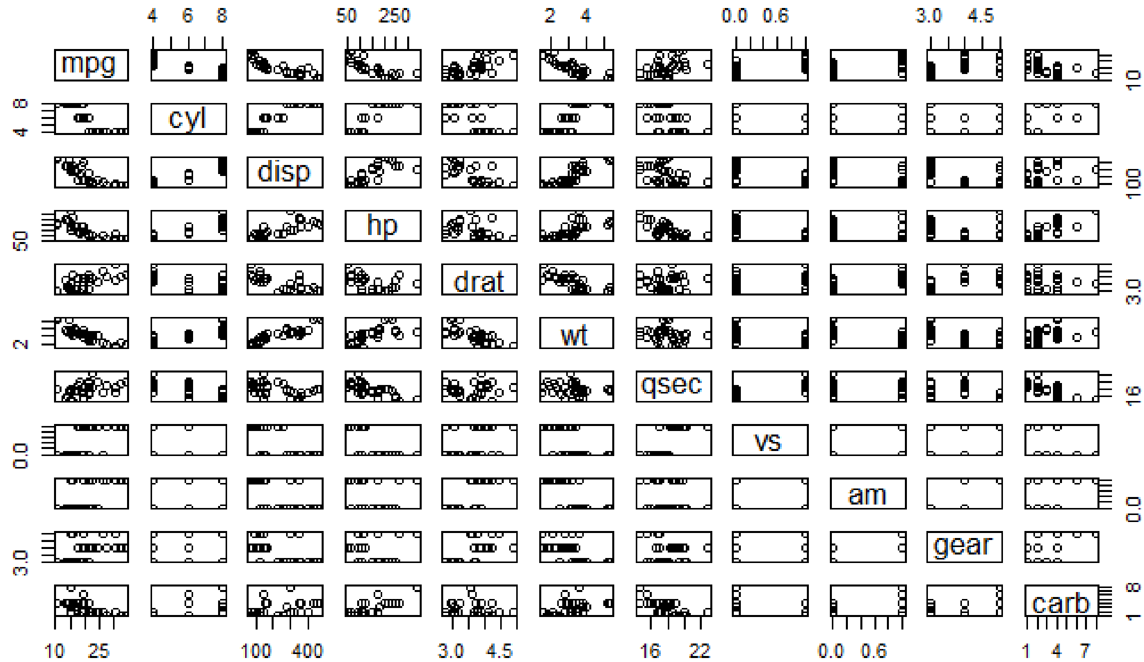



Distribution mpg - automatic transmission
Frequency
0
2
4
10
15
20
25
mpg

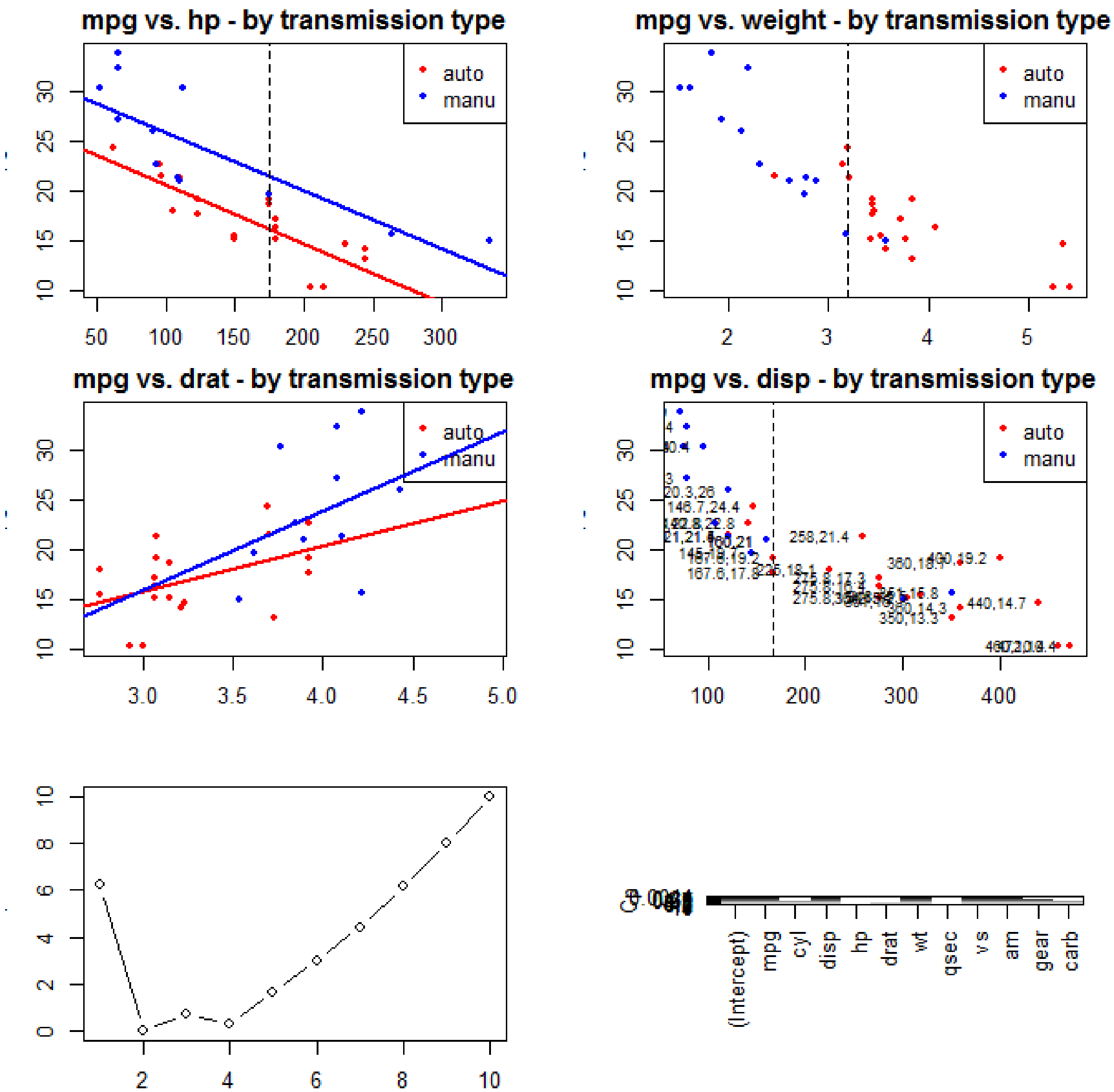

Distribution mpg - manual transmission
Frequency
0
2
4
15
20
25
30
35
mpg


Distribution of shuffled group mean differences
Frequency
0
10000
-5
0
5
permutations

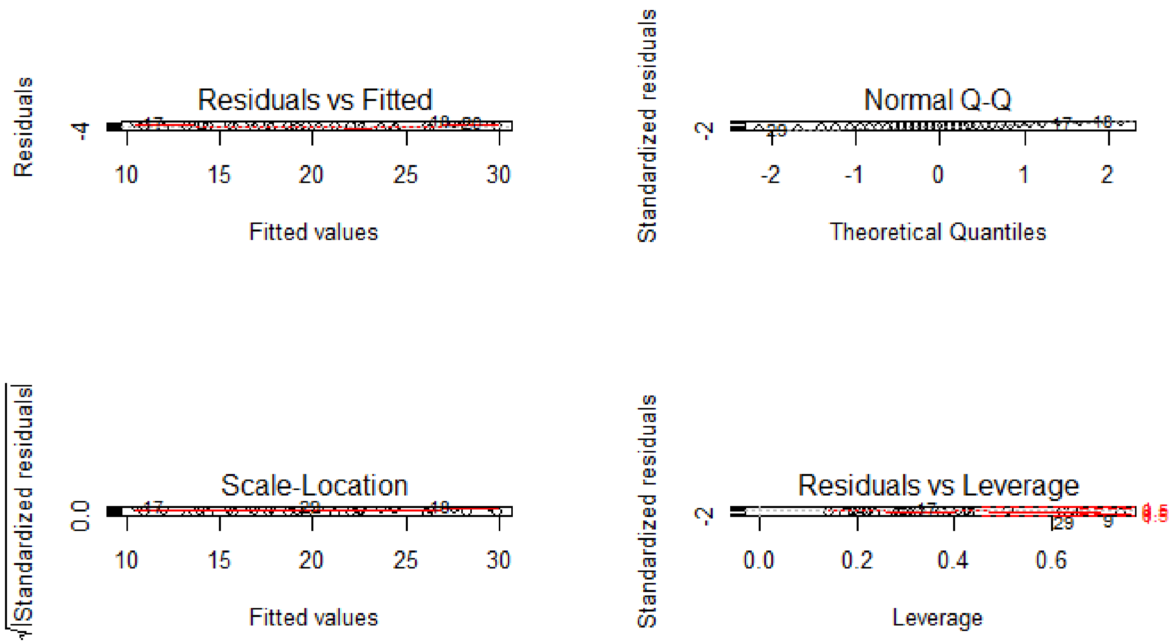

Shuffled group mean trials
shuffled group mean differen
-15
-5
5
15
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
trial

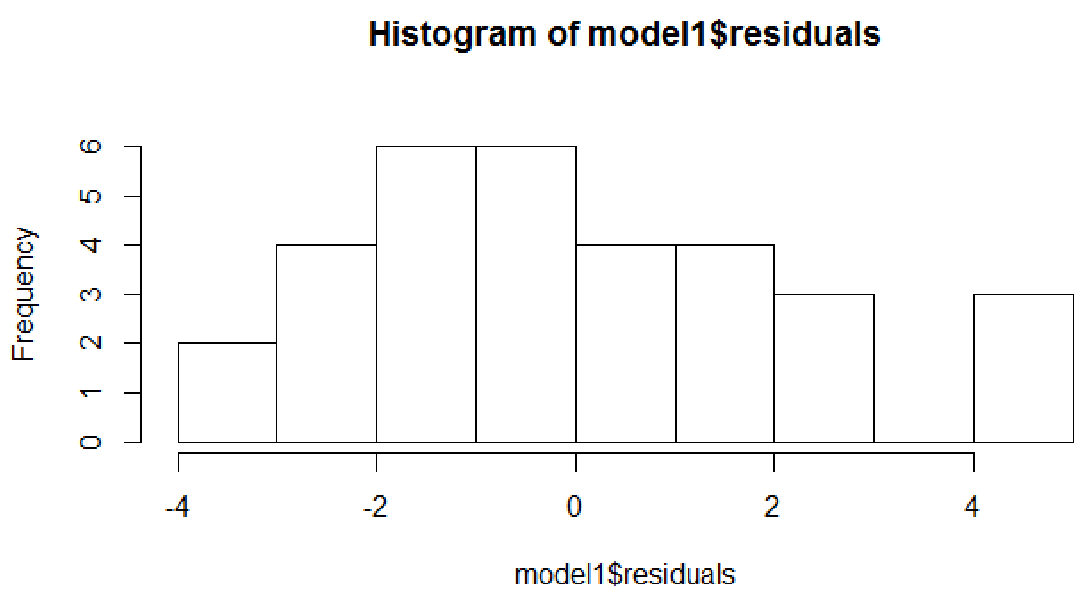


mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb



mpg vs. hp - by transmission type
auto
manu
mpg vs. weight - by transmission type
auto
manu
mpg vs. drat - by transmission type
auto
manu
mpg vs. disp - by transmission type
auto
manu
(Intercept)
mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb

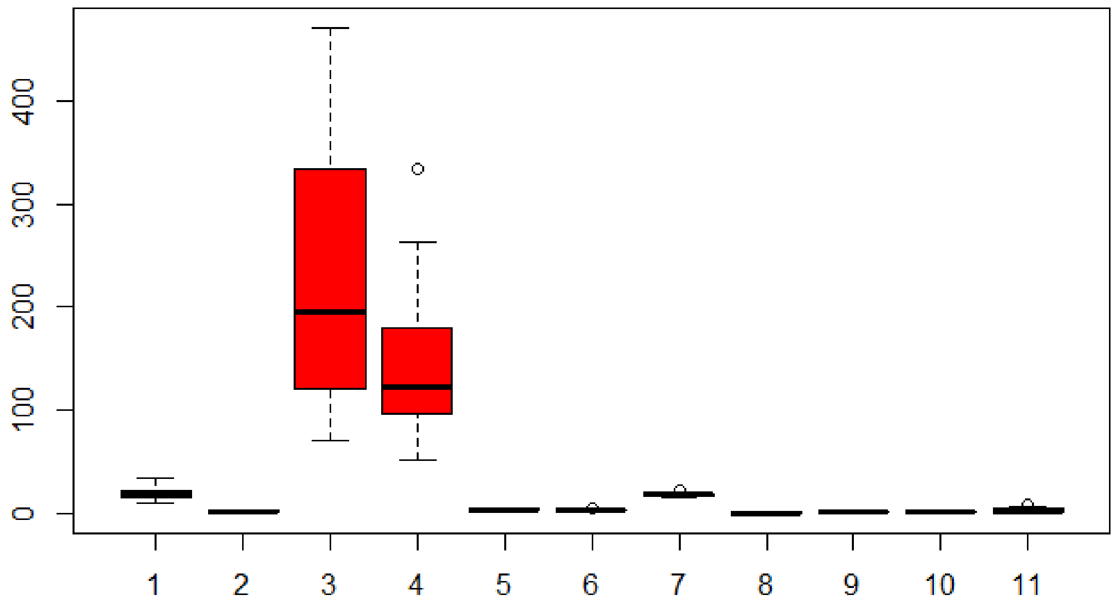

Residuals vs Fitted
Residuals
Fitted values
10
15
20
25
30
Normal Q-Q
Standardized residuals
Theoretical Quantiles
-2
-1
0
1
2
Scale-Location
√|Standardized residuals|
Fitted values
10
15
20
25
30
Residuals vs Leverage
Standardized residuals
Leverage
0.0
0.2
0.4
0.6

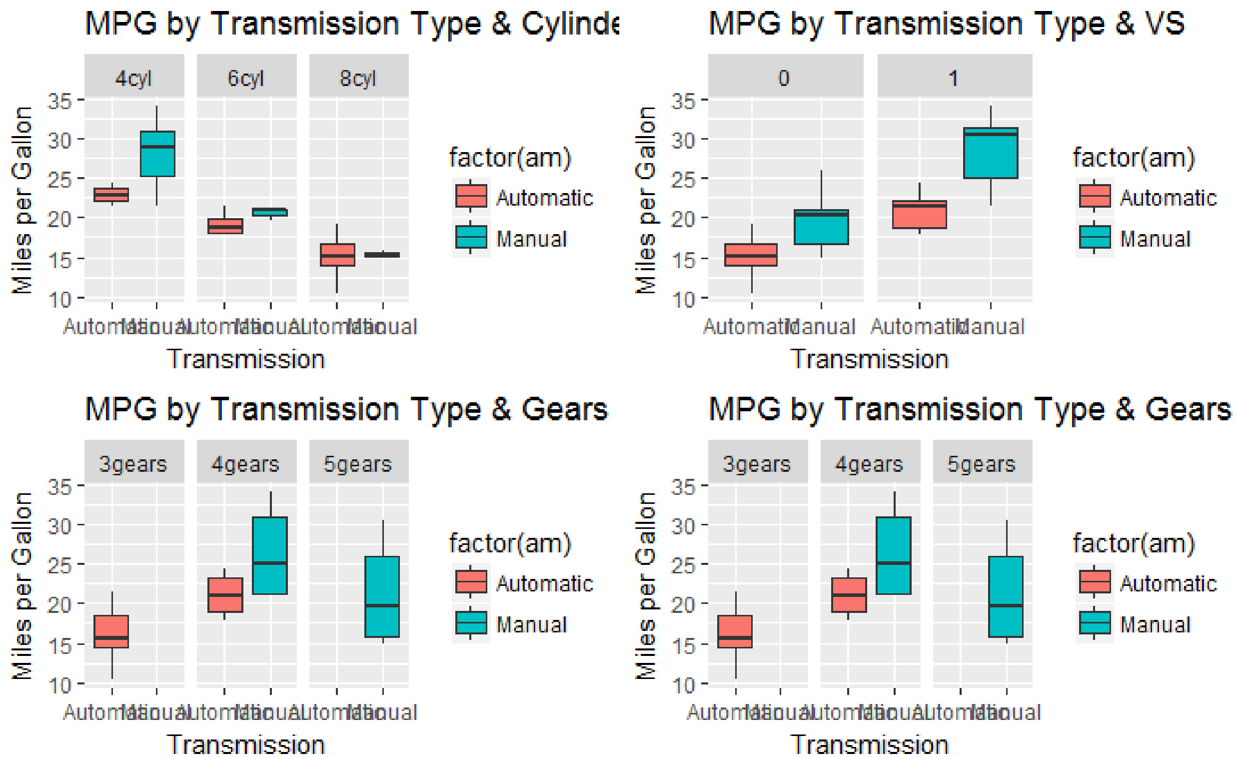

Histogram of model1$residuals
Frequency
0
1
2
3
4
5
6
-4
-2
0
2
4
model1$residuals

3. Write a program to create boxplot for all variables

```
library(psych)
describe(mtcars)
boxplot(mtcars$mpg,mtcars$cyl,mtcars$disp,mtcars$hp,mtcars$drat,mtcars$wt,mtc
ars$qsec,mtcars$vs,mtcars$am,mtcars$gear,mtcars$carb,col = "red")
library(ggplot2)
library(car)
library(corrgram)
data=mtcars
name=mtcars
mtcars$am <- as.factor(mtcars$am)
levels(mtcars$am) <- c("Automatic", "Manual")
head(mtcars)
summary(mtcars)

describe(mtcars)

boxplot(mtcars$mpg,mtcars$cyl,mtcars$disp,mtcars$hp,mtcars$drat,mtcars$wt,mtc
ars$qsec,mtcars$vs,mtcars$am,mtcars$gear,mtcars$carb,col = "red")


plot1 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+
geom_boxplot(notch=F)+facet_grid(.~cyl)+scale_x_discrete("Transmission")+
scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Cylinder")
plot1 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+
geom_boxplot(notch=F)+facet_grid(.~cyl)+scale_x_discrete("Transmission")+
scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Cylinder")
plot2 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+
geom_boxplot(notch=F)+facet_grid(.~vs)+scale_x_discrete("Transmission")+
scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
VS")
plot3 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+
geom_boxplot(notch=F)+facet_grid(.~gear)+scale_x_discrete("Transmission")+
scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Gears")
plot4 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+
geom_boxplot(notch=F)+facet_grid(.~carb)+scale_x_discrete("Transmission")+
scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
```

```
Carburetors")
grid.arrange(plot1, plot2, plot3, plot3, nrow=2, ncol=2)
```





R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for autho ring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(psych)

describe(mtcars)

boxplot(mtcars$mpg,mtcars$cyl,mtcars$disp,mtcars$hp,mtcars$drat,mtcars$wt,mtc
ars$qsec,mtcars$vs,mtcars$am,mtcars$gear,mtcars$carb,col = "red")

library(ggplot2)

library(car)

library(corrgram)

library(reshape)

library(dplyr)

library(gridExtra)

data=mtcars

name=mtcars

mtcars$am <- as.factor(mtcars$am)

levels(mtcars$am) <- c("Automatic", "Manual")

head(mtcars)

summary(mtcars)

plot1 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+

geom_boxplot(notch=F)+facet_grid(.~cyl)+scale_x_discrete("Transmission")+

scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Cylinder")

plot1 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+

geom_boxplot(notch=F)+facet_grid(.~cyl)+scale_x_discrete("Transmission")+

scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Cylinder")

plot2 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+

geom_boxplot(notch=F)+facet_grid(.~vs)+scale_x_discrete("Transmission")+

scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
VS")

plot3 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+

geom_boxplot(notch=F)+facet_grid(.~gear)+scale_x_discrete("Transmission")+
```

```r
scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Gears")

plot4 <- ggplot(mtcars, aes(x=factor(am),y=mpg,fill=factor(am)))+

geom_boxplot(notch=F)+facet_grid(.~carb)+scale_x_discrete("Transmission")+

scale_y_continuous("Miles per Gallon")+ggtitle("MPG by Transmission Type &
Carburetors")

grid.arrange(plot1, plot2, plot3, plot3, nrow=2, ncol=2)

summary(cars)
```

dbl>	n <dbl>	mean <dbl>	sd <dbl>	median <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>	
mpg	1	32	20.09	6.03	19.20	19.70	5.41	10.40	33.90
cyl*	2	32	2.09	0.89	2.00	2.12	1.48	1.00	3.00
disp	3	32	230.72	123.94	196.30	222.52	140.48	71.10	472.00
hp	4	32	146.69	68.56	123.00	141.19	77.10	52.00	335.00
drat	5	32	3.60	0.53	3.70	3.58	0.70	2.76	4.93
wt	6	32	3.22	0.98	3.33	3.15	0.77	1.51	5.42
qsec	7	32	17.85	1.79	17.71	17.83	1.42	14.50	22.90
vs	8	32	0.44	0.50	0.00	0.42	0.00	0.00	1.00
am*	9	32	1.41	0.50	1.00	1.38	0.00	1.00	2.00
gear*	10	32	1.69	0.74	2.00	1.62	1.48	1.00	3.00

Next
12
Previous
1-10 of 11 rows | 1-10 of 13 columns

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.