Explanatory Data Analysis with various visualizations (ggplot2)

Contents

Choi Suhyun

Import Libraries

```
library(knitr)
library(ggplot2)
library(gridExtra)
library(lattice)
library(RColorBrewer)
library(reshape2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1 seeds.txt

```
seeds = read.delim2("/Users/hailey/Desktop/STAT3622/Assignment 1/seeds.txt",
                    sep =',')
kable(head(seeds, 5))
```

area	perimeter	compactness	length.of.kernel	width.of.kernel	asymmetry.coefficient	length.of.kernel.groove	variety
15.26	14.84	0.871	5.763	3.312	2.221	5.22	Kama
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	Kama
14.29	14.09	0.905	5.291	3.337	2.699	4.825	Kama
13.84	13.94	0.8955	5.324	3.379	2.259	4.805	Kama
16.14	14.99	0.9034	5.658	3.562	1.355	5.175	Kama

```
summary(seeds)
```

```
##      area            perimeter       compactness      length.of.kernel
##  Length:210         Length:210         Length:210         Length:210
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##  width.of.kernel    asymmetry.coefficient length.of.kernel.groove
##  Length:210         Length:210            Length:210
##  Class :character   Class :character      Class :character
##  Mode  :character   Mode  :character      Mode  :character
##    variety
##  Length:210
##  Class :character
##  Mode  :character
```

```
#Changing data type of attributes to "numeric"
i <- c(1:7)
seeds[ , i] <- apply(seeds[ , i], 2,
                    function(x) as.numeric(as.character(x)))
```

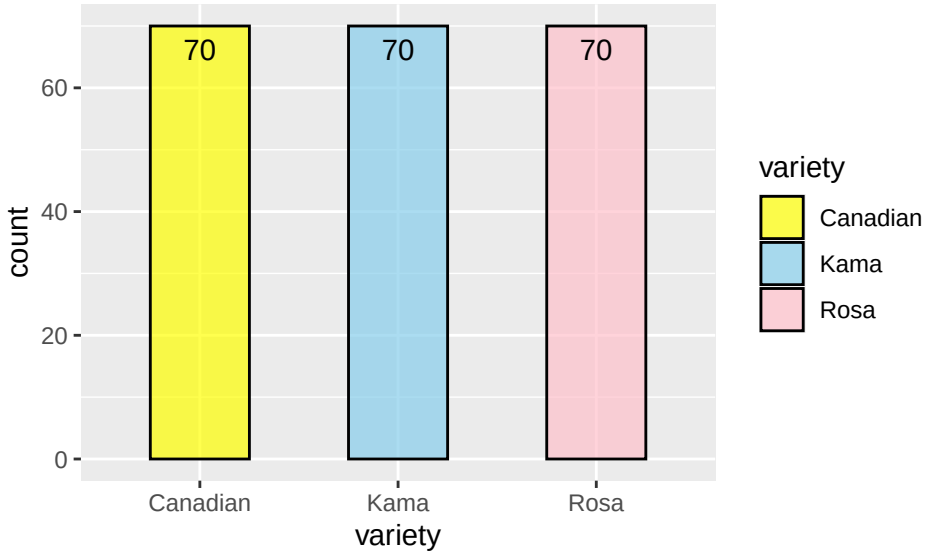
```
str(seeds)
```

```
## 'data.frame':    210 obs. of  8 variables:
##  $ area                  : num  15.3 14.9 14.3 13.8 16.1 ...
##  $ perimeter             : num  14.8 14.6 14.1 13.9 15 ...
##  $ compactness           : num  0.871 0.881 0.905 0.895 0.903 ...
##  $ length.of.kernel      : num  5.76 5.55 5.29 5.32 5.66 ...
##  $ width.of.kernel       : num  3.31 3.33 3.34 3.38 3.56 ...
##  $ asymmetry.coefficient : num  2.22 1.02 2.7 2.26 1.35 ...
##  $ length.of.kernel.groove: num  5.22 4.96 4.83 4.8 5.17 ...
##  $ variety               : chr  "Kama" "Kama" "Kama" "Kama" ...
```

1.1 Distribution of 'variety'

a) Visualize the distribution for the categorical attribute "variety".

```
ggplot(seeds, aes(variety, fill=variety)) + geom_bar(colour="black", size=0.5,
                                                     width=0.5, alpha=0.7) +
  scale_fill_manual(values = c("yellow","skyblue","pink")) +
  geom_text(stat='count', aes(label=..count..),color="black", vjust=1.6)
```

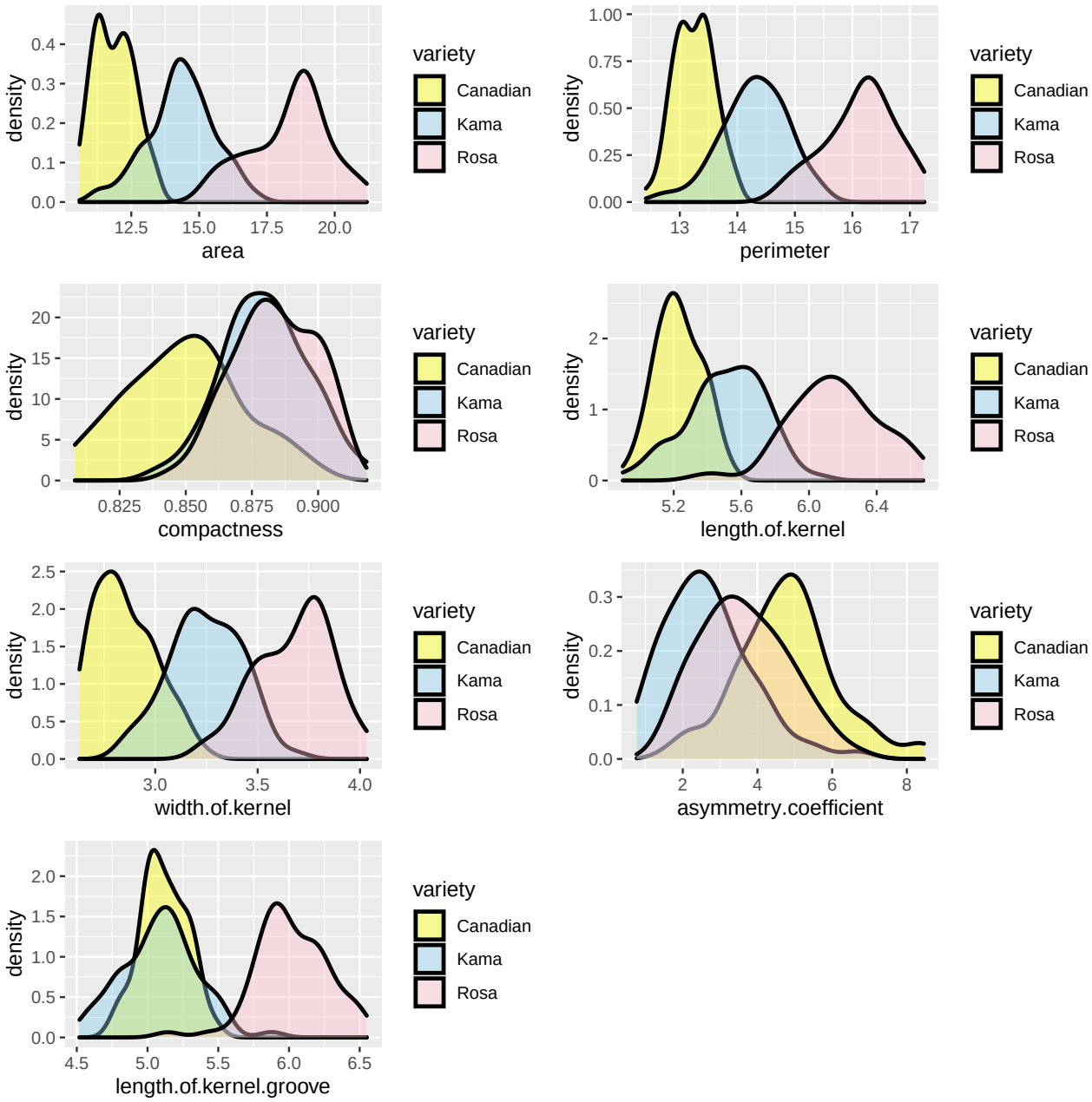


- There are three types of seeds, that are Canadian, Kama and Rosa.
- There are 70 Canadian seeds, 70 Kama seeds and 70 Rosa seeds. Therefore, there are 210 seeds in total.

1.2 Distributional difference among varieties of seeds

b) Visualize the distributional difference for each continuous attribute among three varieties of seeds.

```
plist <- list()
for (i in 1:7){
plist[[length(plist)+1]] <- ggplot(data=seeds, aes_string(names(seeds)[i],
                                                          fill=names(seeds)[8])) +
  geom_density(size=1, alpha=0.4) +
  scale_fill_manual( values = c("yellow","skyblue","pink"))
}

n <- length(plist)
nCol <- floor(sqrt(n))
do.call("grid.arrange", c(plist, ncol=nCol))
```



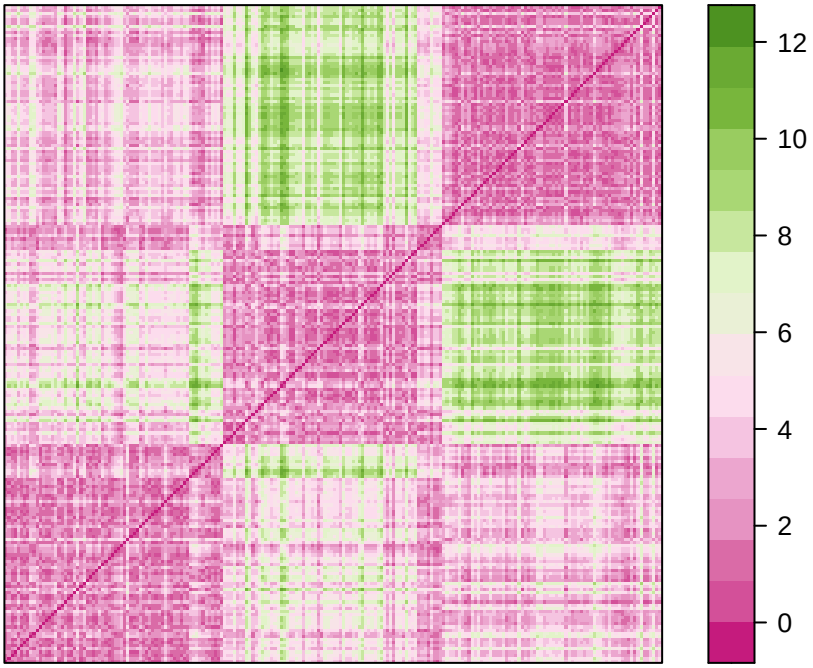
- In each plot, yellow, blue and pink graph each represents density distribution of Canadian, Kama and Rosa seeds respectively.
- For example, most of the Rosa seeds have larger area than most of the Canadian seeds.

1.3 Euclidean distance matrix

c) Visualize the Euclidean distance matrix of samples involving all continuous attributes by the heatmap.

```
dist = as.matrix(dist(seeds[,0:7], method='euclidean'))
coul <- colorRampPalette(brewer.pal(8, "PiYG"))(25)
levelplot(dist, colorkey=T, col.regions = coul,
```

```
        scales = list(at=c(0,0),tck=c(0,0)),
        xlab="", ylab="", main="Euclidean distance of all continuous samples")
```



This heatmap shows 210*210 matrix in which element in n^{th} row and m^{th} column represents the euclidean distance between n^{th} sample and m^{th} sample. In mathematics, the Euclidean distance between two points in Euclidean space is the length of a line segment between the two points, which can be calculated as:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

where p, q *= two points in Euclidean n-space,*
q_i, p_i *= Euclidean vectors, starting from the origin of the space (initial point),*
n *= n-space*

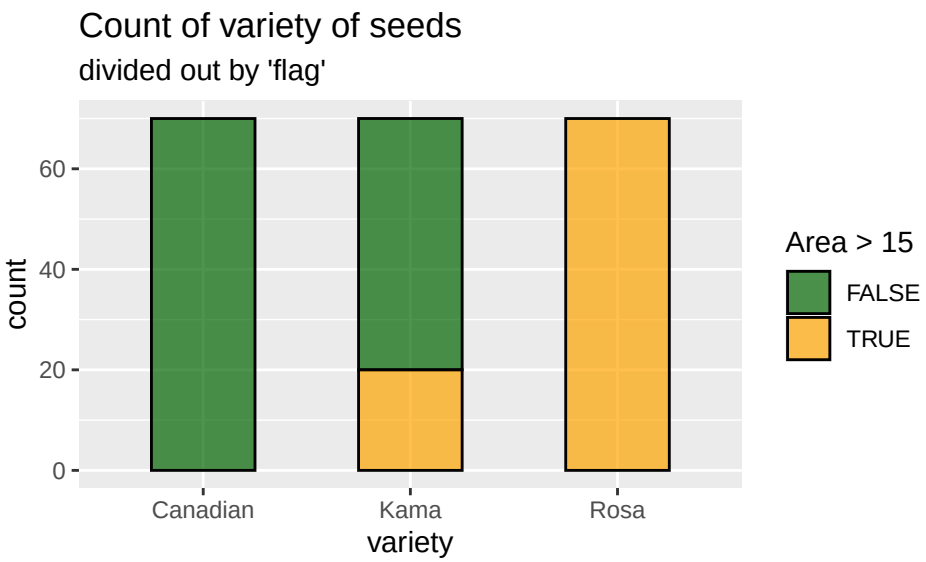
The color in the heatmap represents the value of the euclidean distance, as shown in the color label beside the heatmap. The diagonal line in the heatmap is generated as euclidean distance between a point and itself is 0.

1.4 Flag (area>15)

d) Create a new variable "flag", which takes the value "True" if the "area" is larger than 15, and "False", otherwise. Show a stacked bar graph of the sample size for each variety of seeds and how they are further divided out by "flag".

```
seeds$flag = seeds$area>15
ggplot(seeds, aes(variety, fill=flag)) +
  geom_bar(size=0.5, colour="black",position = "stack", alpha=0.7, width=0.5) +
  scale_fill_manual("Area > 15",
```

```
                    values = c("TRUE" = "orange", "FALSE" = "darkgreen")) +
  labs(title="Count of variety of seeds", subtitle = "divided out by 'flag'")
```

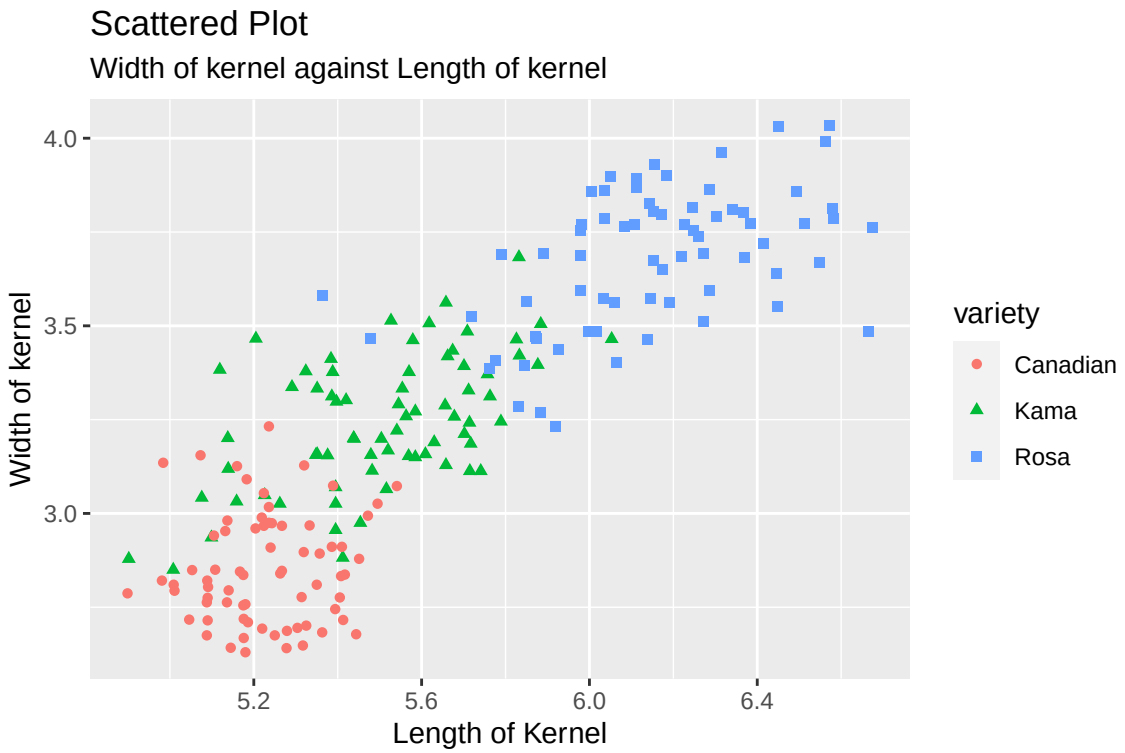


- Canadian: Area of all the 70 samples are smaller than 15.
- Kama : 50 samples have area smaller than 15. 20 samples have area larger than 15.
- Rosa : Area of all the 70 samples are larger than 15.

1.5 Scattered plot of length.of.kernel and width.of.kernel

(e) Show the scattered graph for "length.of.kernel" (x-axis) and "width.of.kernel" (y-axis) of all samples, where the colors of points indicate the varieties of seeds.

```
ggplot(seeds, aes(x=length.of.kernel, y=width.of.kernel,
                  colour=variety, shape=variety)) +
  geom_point() + labs(title = "Scattered Plot",
       subtitle = "Width of kernel against Length of kernel",
       y = "Width of kernel", x = "Length of Kernel")
```

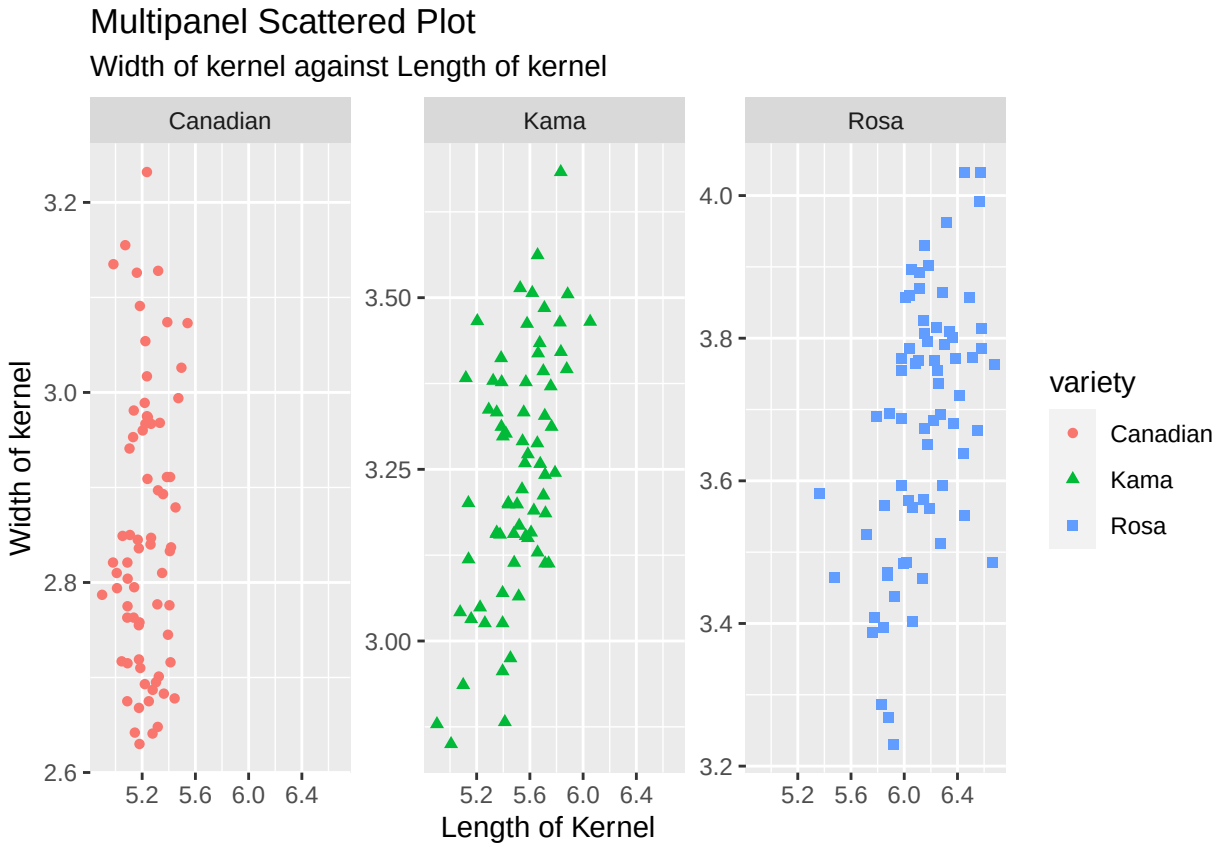


- Red, green and blue points represent Canadian, Kama and Rosa seeds respectively.
- All the points are linearly positively correlated, which implies that width of kernel increases (approximately) proportionally with length of kernel.
- Canadian seeds have relatively smaller width and length of kernel while Rosa seeds have relatively larger length and width of kernel.

1.6 Multipanel scattered plot of length.of.kernel and width.of.kernel

(f) Show the multipanel scattered plots for "length.of.kernel" (x-axis) and "width.of.kernel" (y-axis) conditional on "variety".

```
ggplot(seeds,
       aes(x=length.of.kernel, y=width.of.kernel, colour=variety, shape=variety)) +
  geom_point() + labs(title = "Multipanel Scattered Plot",
       subtitle = "Width of kernel against Length of kernel",
       y = "Width of kernel", x = "Length of Kernel") +
  facet_wrap(~ variety, scales = "free_y")
```



- Multipanel plot with one panel per "variety"
- Each panel shows width of kernel against length of kernel graph for each variety of seeds.

2 Boston.txt

List of variables 1. crim — per capita crime rate by town 2. zn — proportion of residential land zoned for lots over 25,000 sq.ft 3. indus — proportion of non-retail business acres per town 4. chas — Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) 5. nox — nitric oxides concentration (parts per 10 million) 6. rm — average number of rooms per dwelling 7. age — proportion of owner-occupied units built prior to 1940 8. dis — weighted distances to five Boston employment centres 9. rad — index of accessibility to radial highways 10. tax — full-value property-tax rate per USD 10,000 11. ptratio — pupil-teacher ratio by town 12. black — proportion of blacks by town 13. lstat — percentage of lower status of the population 14. medv — median value of owner-occupied homes in USD 1000's

All the variables are continuous except "chas" and "rad".

```
options(max.col=20)
boston = read.delim2("/Users/hailey/Desktop/STAT3622/Assignment 1/Boston.txt",
                     sep =',')
kable(head(boston, 5))
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2

```
str(boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : chr  "0.00632" "0.02731" "0.02729" "0.03237" ...
##  $ zn     : chr  "18" "0" "0" "0" ...
##  $ indus  : chr  "2.31" "7.07" "7.07" "2.18" ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : chr  "0.538" "0.469" "0.469" "0.458" ...
##  $ rm     : chr  "6.575" "6.421" "7.185" "6.998" ...
##  $ age    : chr  "65.2" "78.9" "61.1" "45.8" ...
##  $ dis    : chr  "4.09" "4.9671" "4.9671" "6.0622" ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: chr  "15.3" "17.8" "17.8" "18.7" ...
##  $ b      : chr  "396.9" "396.9" "392.83" "394.63" ...
##  $ lstat  : chr  "4.98" "9.14" "4.03" "2.94" ...
##  $ medv   : chr  "24" "21.6" "34.7" "33.4" ...
```

Data types of all variables except "chad" and "rad" need to be changed to numeric for further analysis. "chas" and "rad" can be left as integer type since they are categorical variables.

```
# Changing "character" to "numeric" type
i <- c(1,2,3,5,6,7,8,10,11,12,13,14)
boston[ , i] <- apply(boston[ , i], 2,
                      function(x) as.numeric(as.character(x)))
```

```
summary(boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
```

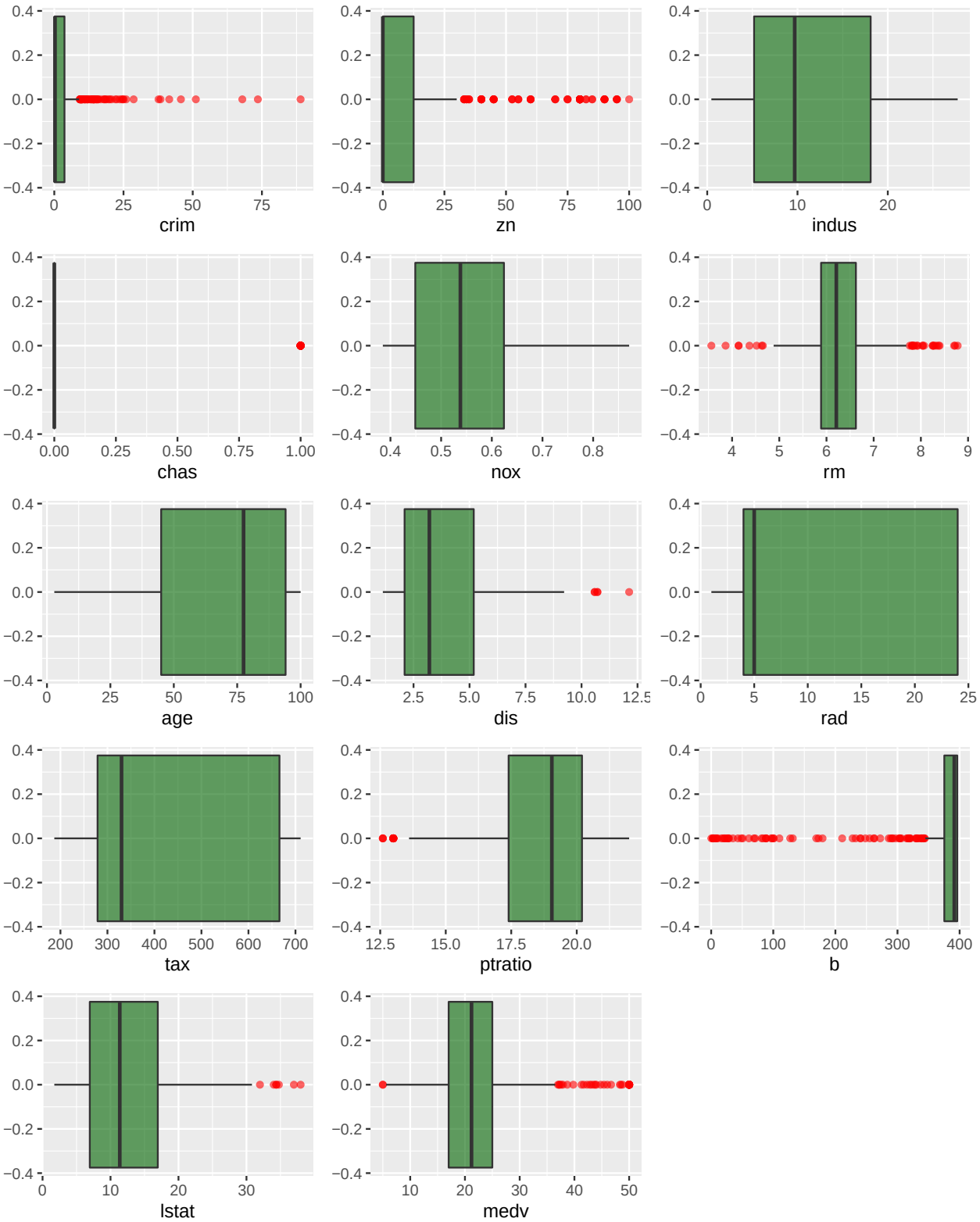
```
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio            b
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

Minimum and maximum values and interquartile ranges of 14 variables are shown above.

2.1 Distribution of attributes

(a) Visualize the distribution for all attributes.

```
plist <- list()
for (i in 1:14){
plist[[length(plist)+1]] <- ggplot(data=boston, aes_string(names(boston)[i])) +
  geom_boxplot(fill = "darkgreen", alpha=0.6 , outlier.colour = "red")
}

n <- length(plist)
nCol <- floor(sqrt(n))
do.call("grid.arrange", c(plist, ncol=nCol))
```



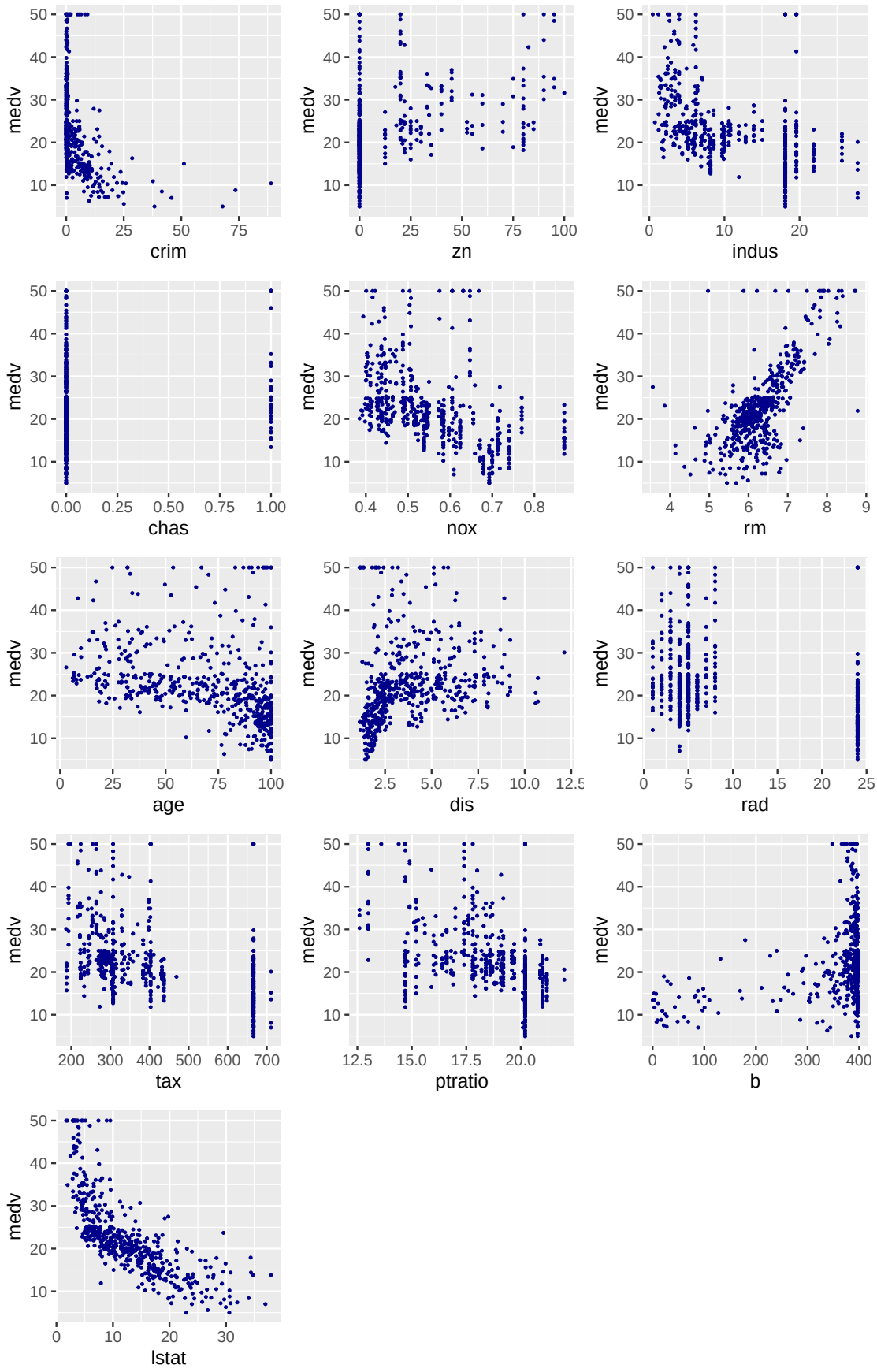
- Box plot distribution of all the attributes are generated.
- "chas" only consists of 0's and 1's.
- Red points represent outliers. For example, variable "crim", "zn", "b" and "medv" have many outliers.
- "crim", "zn", "dis", "rad", "tax"... are positively skewed.

- “age”,“b”,.. are negatively skewed.
- Target variable “medv” has approximately symmetrical distribution.

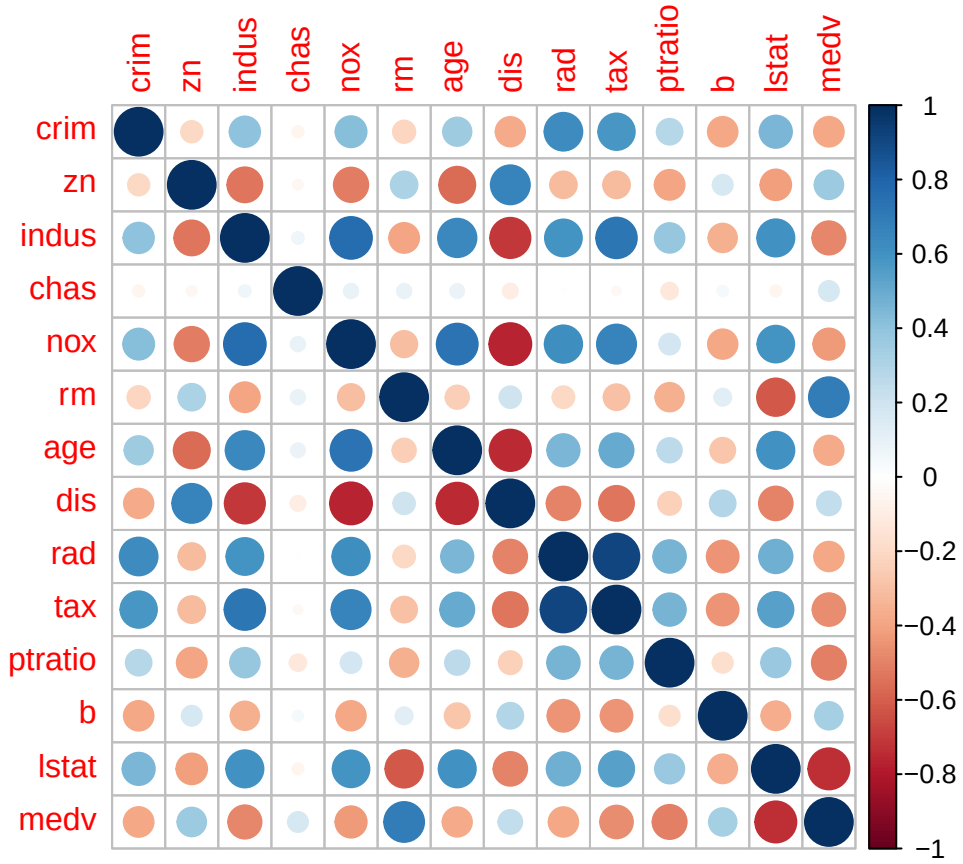
2.2 Relationship between attributes and “medv”

(b) Visualize the relationship between the variable “medv” and any other attribute. Which attributes may have impacts on the median values of owner-occupied homes (“medv”)?

```
plist <- list()
for (i in 1:13){
plist[[length(plist)+1]] <- ggplot(data=boston, aes_string(names(boston)[i],
                                                           names(boston)[14])) +
  geom_point(size=0.4, color="darkblue")
}

n <- length(plist)
nCol <- floor(sqrt(n))
do.call("grid.arrange", c(plist, ncol=nCol))
```


medv
crim
zn
indus
chas
nox
rm
age
dis
rad
tax
ptratio
b
lstat

```
corrplot(cor(boston))
```



- As you can see in the correlation plot above, lstat has the highest (negative) correlation with medv. Also rm seems to have the highest (positive) correlation with medv. These are also shown in the scatter plot of medv against lstat and rm.
- According to the correlation plot, indus and ptratio have relatively high (negative) correlation with medv as well.
- 5 attributes are positively correlated to medv while 8 attributes are negatively correlated to medv.

2.3 Linear regression model

(c) Fit a linear regression model,where the response variableis "medv" (transformation is allowed). You can choose a subset of attributes or create new attributes as covariates. Interpret the resulting model.

First, divide the data into train and test sets.

```
set.seed(12)

boston$id <- 1:nrow(boston)
train <- boston %>% dplyr::sample_frac(.75)
test  <- dplyr::anti_join(boston, train, by = 'id')

dim(train)
```

```
## [1] 380  15
```

```
dim(test)
```

```
## [1] 126  15
```

Fit the linear regression model with all the attributes excluding 15th column of train data (id).

```
lm_1 <- lm(medv~.,data=train[,1:14])
summary(lm_1)
```

```
##
## Call:
## lm(formula = medv ~ ., data = train[, 1:14])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4261  -2.6614  -0.4531   1.5223  27.8946
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.842989   5.544071   7.187 3.76e-12 ***
## crim         -0.100164   0.032544  -3.078 0.002242 **
## zn            0.047613   0.015060   3.161 0.001701 **
## indus         0.016022   0.065644   0.244 0.807314
## chas          2.200252   0.944604   2.329 0.020388 *
## nox         -22.183486   4.344536  -5.106 5.30e-07 ***
## rm            3.853848   0.478848   8.048 1.19e-14 ***
## age          -0.003804   0.014146  -0.269 0.788161
## dis          -1.521501   0.227574  -6.686 8.60e-11 ***
## rad           0.355449   0.075174   4.728 3.24e-06 ***
## tax          -0.015702   0.004228  -3.714 0.000236 ***
## ptratio      -0.973159   0.140714  -6.916 2.08e-11 ***
## b             0.007374   0.002756   2.675 0.007802 **
## lstat        -0.450847   0.057594  -7.828 5.39e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.494 on 366 degrees of freedom
## Multiple R-squared:  0.7567, Adjusted R-squared:  0.748
## F-statistic: 87.55 on 13 and 366 DF,  p-value: < 2.2e-16
```

- The F-test for linear regression tests whether any of the independent variables in a multiple linear regression model are significant. R-squared measures the proportion of variation in the dependent variable that can be attributed to the independent variable.
- Fitting all the variables gives **F-statistic: 87.55** and **R-squared: 0.7567**.
- indus, age and b are removed from the train dataset since they have high P(>|t|) value and thus are not significant.

```
lm_2 <- lm(medv~.-age-indus-b,data=train[,1:14])
summary(lm_2)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus - b, data = train[, 1:14])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -13.0197  -2.6289  -0.5559   1.6857  28.0286
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43.563772   5.365666   8.119 7.11e-15 ***
## crim         -0.107687   0.032592  -3.304  0.00105 **
## zn            0.047538   0.014891   3.192  0.00153 **
## chas          2.324859   0.946248   2.457  0.01447 *
## nox         -23.110421   3.971583  -5.819 1.29e-08 ***
## rm            3.757535   0.468134   8.027 1.35e-14 ***
## dis          -1.528008   0.215691  -7.084 7.13e-12 ***
## rad           0.339234   0.072526   4.677 4.08e-06 ***
## tax          -0.015914   0.003913  -4.067 5.82e-05 ***
## ptratio      -0.946561   0.139977  -6.762 5.33e-11 ***
## lstat        -0.473498   0.053370  -8.872  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.519 on 369 degrees of freedom
## Multiple R-squared:  0.7519, Adjusted R-squared:  0.7452
## F-statistic: 111.8 on 10 and 369 DF,  p-value: < 2.2e-16
```

- After removing indus, age and b, F-statistics **increased to 111.8** and R-squared **decreased to 0.7519**.
- Adding Interaction variables between the significant variables could give us a better model.

```
lm_3 <- lm(medv~.-age-indus-b+ rm*lstat + rm*ptratio +
             lstat*ptratio,data=train[,1:14])
summary(lm_3)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus - b + rm * lstat + rm * ptratio +
##     lstat * ptratio, data = train[, 1:14])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5310  -2.1070  -0.1881   1.5397  25.3458
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.589e+02  2.173e+01  -7.310 1.69e-12 ***
## crim          -1.301e-01  2.641e-02  -4.927 1.27e-06 ***
## zn             1.768e-02  1.274e-02   1.388 0.166049
## chas           2.013e+00  7.691e-01   2.617 0.009241 **
## nox           -1.595e+01  3.314e+00  -4.812 2.19e-06 ***
## rm             3.086e+01  2.826e+00  10.921  < 2e-16 ***
## dis           -1.122e+00  1.819e-01  -6.171 1.80e-09 ***
## rad            2.576e-01  5.872e-02   4.387 1.50e-05 ***
## tax           -1.004e-02  3.222e-03  -3.116 0.001979 **
## ptratio        8.651e+00  1.162e+00   7.445 7.03e-13 ***
## lstat          3.073e+00  5.681e-01   5.409 1.15e-07 ***
## rm:lstat      -2.991e-01  3.788e-02  -7.897 3.36e-14 ***
## rm:ptratio    -1.302e+00  1.574e-01  -8.270 2.51e-15 ***
## ptratio:lstat -9.736e-02  2.501e-02  -3.893 0.000118 ***
```

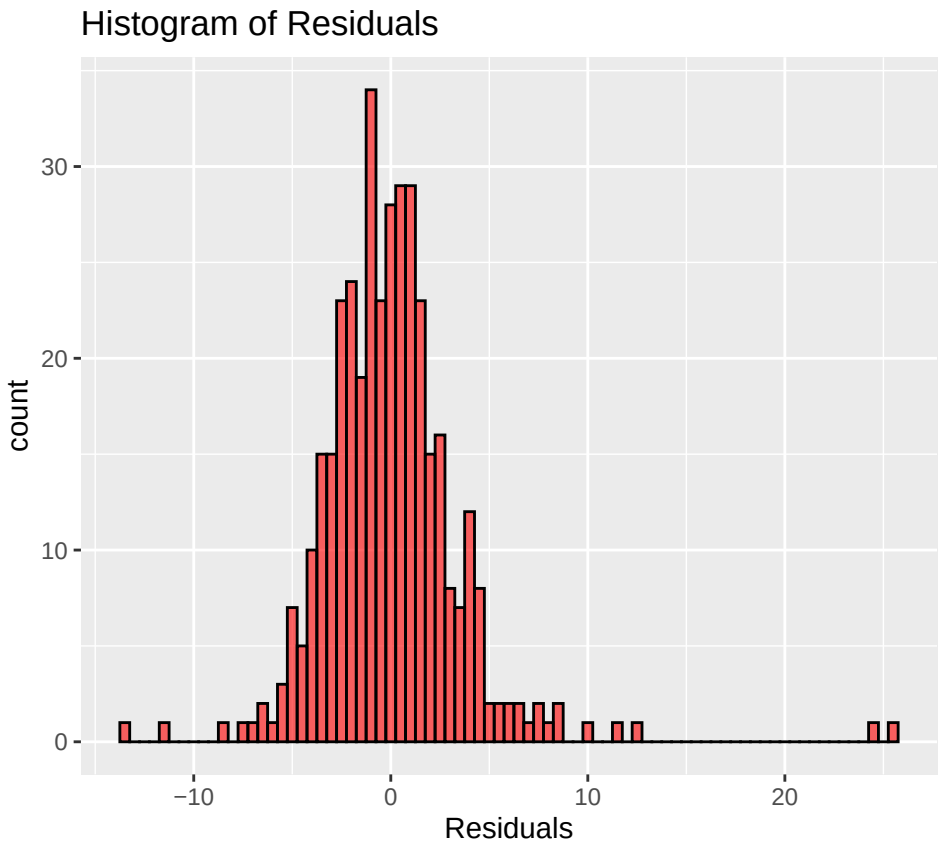
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.633 on 366 degrees of freedom
## Multiple R-squared:  0.841,  Adjusted R-squared:  0.8353
## F-statistic: 148.9 on 13 and 366 DF,  p-value: < 2.2e-16
```

- Adding rm:lstat, rm:ptratio and ptratio:lstat improved the goodness of fit.
- **F-statistic: 148.9** and **R-squared: 0.841**

2.4 Assumptions for linear regression

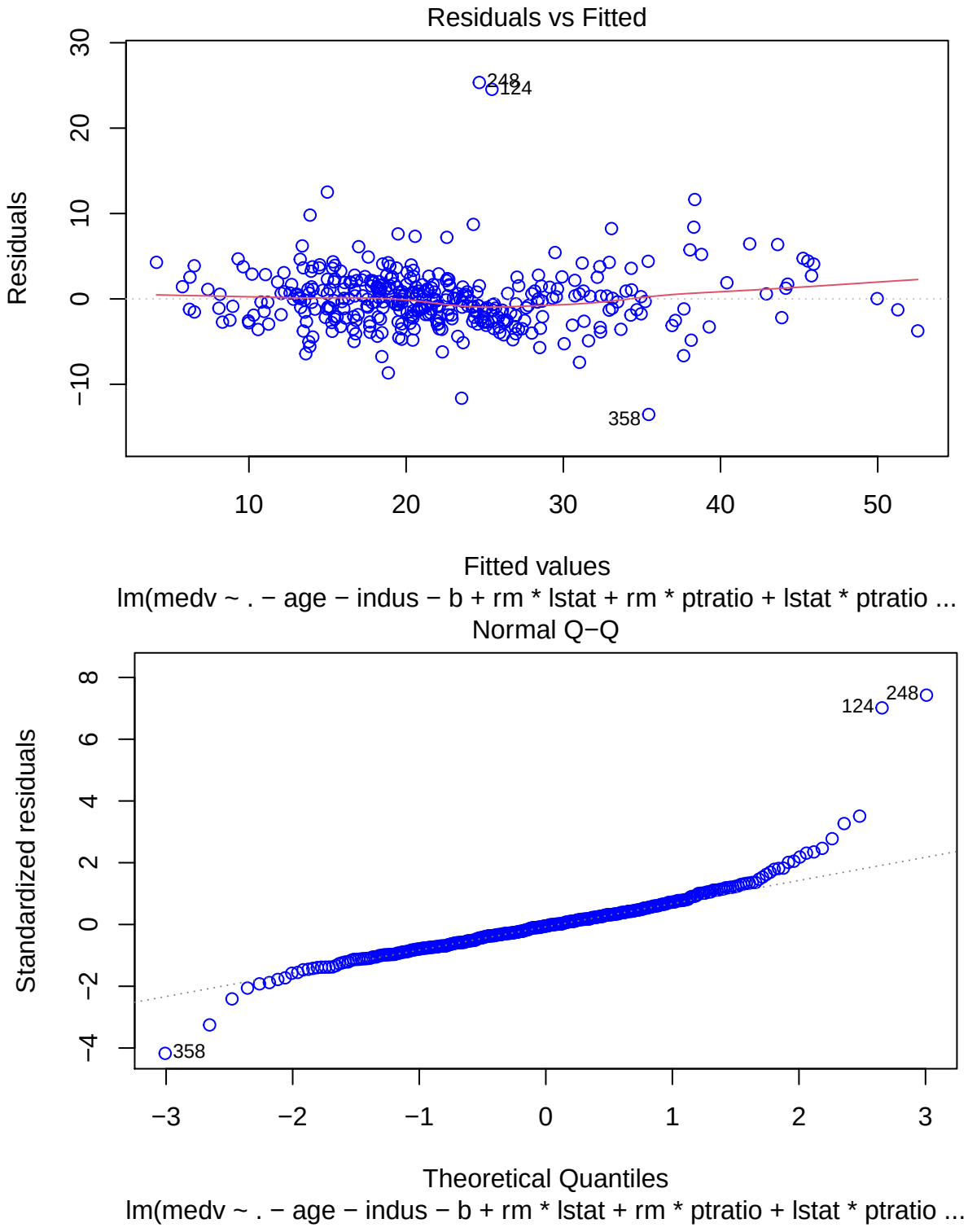
(d) Using plots to check whether the model in (c) satisfies the assumptions for linear regression. Interpret the results.

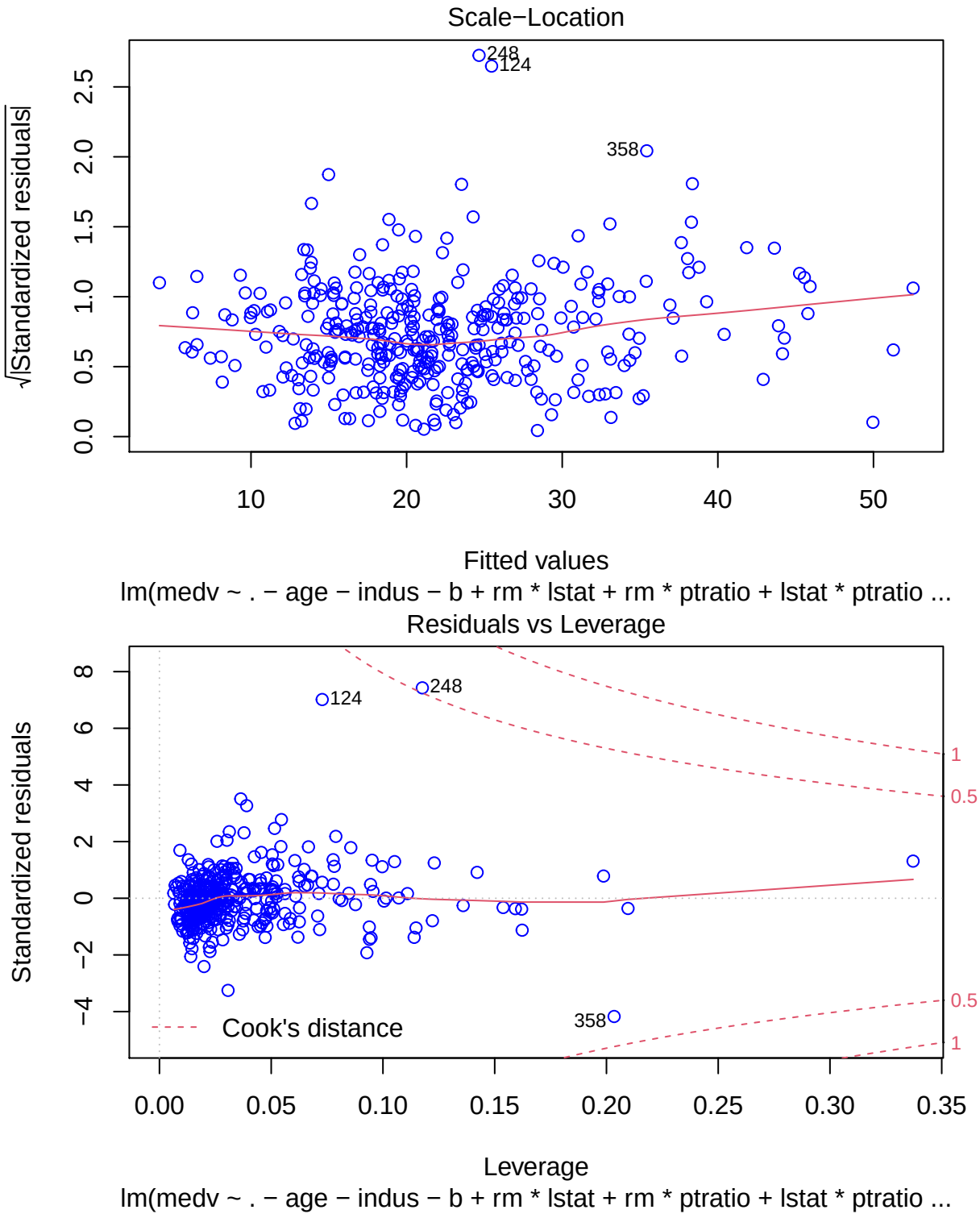
```
#Multivariate Normality

residuals <- data.frame('Residuals' = lm_3$residuals)
ggplot(residuals, aes(x=Residuals)) +
  geom_histogram(binwidth=0.5, color='black', fill='red', alpha=0.6) +
  ggtitle('Histogram of Residuals')
```



Multivariate normality: The residuals can be considered as normally distributed according to the histogram above.

```
#No multicollinearity

plot(lm_3, col='Blue')
```


Residuals vs Fitted
Residuals
Fitted values
lm(medv ~ . − age − indus − b + rm * lstat + rm * ptratio + lstat * ptratio ...
248
124
358
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(medv ~ . − age − indus − b + rm * lstat + rm * ptratio + lstat * ptratio ...
248
124
358



Homoscedasticity: Variance of error terms are similar across the values of the independent variables. A plot of standardized residuals versus predicted values can show whether points are equally distributed across all values of the independent variables.

3 mpg.txt

A data frame with 234 rows and 11 variables:

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

3.1 Average displacement for each year

(a) Display a table of the average displacement ("displ") for each year.

```
unique(mpg$year)
```

```
## [1] 1999 2008
```

```
mpg_year <- split(mpg, mpg$year)
mean1 <- mean(mpg_year$`1999`$displ)
mean2 <- mean(mpg_year$`2008`$displ)

year <- c("1999","2008")
mean_displ <- c(mean1, mean2)

df <- data.frame(year,mean_displ)
kable(df)
```

year	mean_displ
1999	3.281197
2008	3.662393

3.2 Median highway mileage per year

(b) Display a table of the median highway mileage ("hwy") per year for each type of car.

```
unique(mpg$class)
```

```
## [1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
## [6] "pickup"     "subcompact"
```

```
mpg_99 <- split(mpg_year$`1999`, mpg_year$`1999`$class)
med_99 <- c(median(mpg_99[[1]][["hwy"]]),median(mpg_99[[2]][["hwy"]]),
            median(mpg_99[[3]][["hwy"]]),median(mpg_99[[4]][["hwy"]]),
            median(mpg_99[[5]][["hwy"]]),median(mpg_99[[6]][["hwy"]]),
            median(mpg_99[[7]][["hwy"]]))
mpg_08 <- split(mpg_year$`2008`, mpg_year$`2008`$class)
med_08 <- c(median(mpg_08[[1]][["hwy"]]),median(mpg_08[[2]][["hwy"]]),
            median(mpg_08[[3]][["hwy"]]),median(mpg_08[[4]][["hwy"]]),
            median(mpg_08[[5]][["hwy"]]),median(mpg_08[[6]][["hwy"]]),
            median(mpg_08[[7]][["hwy"]]))

med <- data.frame(med_99, med_08)
rownames(med) <- c(unique(mpg$class))
colnames(med) <- c("Median hwy in 1999","Median hwy in 2008")
med
```

```
##            Median hwy in 1999 Median hwy in 2008
## compact                  24.5               25.0
## midsize                  26.0               29.0
## suv                      26.0               28.0
## 2seater                  22.0               23.0
## minivan                  17.0               17.0
## pickup                   26.0               26.5
## subcompact               17.0               18.0
```

3.3 displ and cyl in descending order

(c) Display the first five observations in descending order by two attributes, displacement ("displ") and number of cylinders ("cyl").

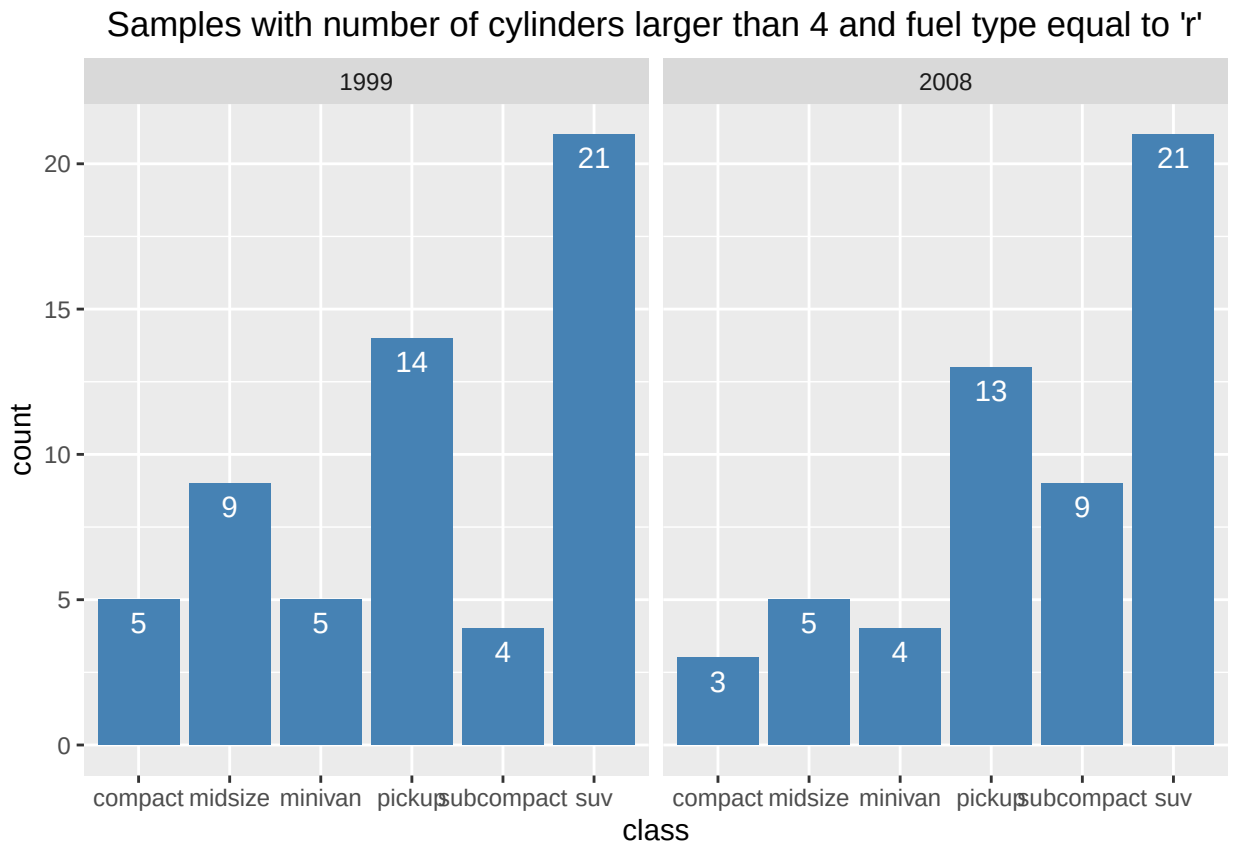
```
head(arrange(mpg, desc(displ), desc(cyl)),5)
```

```
## # A tibble: 5 x 11
##   manufacturer model     displ  year   cyl trans   drv     cty   hwy fl    class
##   <chr>        <chr>     <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
## 1 chevrolet    corvette    7    2008     8 manual~ r        15    24 p     2sea~
## 2 chevrolet    k1500 ta~   6.5  1999     8 auto(l~ 4        14    17 d     suv
## 3 chevrolet    corvette    6.2  2008     8 manual~ r        16    26 p     2sea~
## 4 chevrolet    corvette    6.2  2008     8 auto(s~ r        15    25 p     2sea~
## 5 jeep         grand ch~   6.1  2008     8 auto(l~ 4        11    14 p     suv
```

3.4 cyl $>$ 4 and fl $==$ "r"

(d) Visualize the number of samples for each type of car per year with number of cylinders larger than 4 and fuel type equal to "r".

```
tmp = filter(mpg, cyl>4 & fl=="r")
ggplot(tmp, aes(x=class)) + geom_bar(fill="steelblue")+
  geom_text(stat='count', aes(label=..count..),color="white", vjust=1.6) +
  ggtitle("Samples with number of cylinders larger than 4 and fuel type equal to 'r'") +
  theme(plot.title = element_text(hjust = 0.5)) +
  facet_wrap(~year)
```



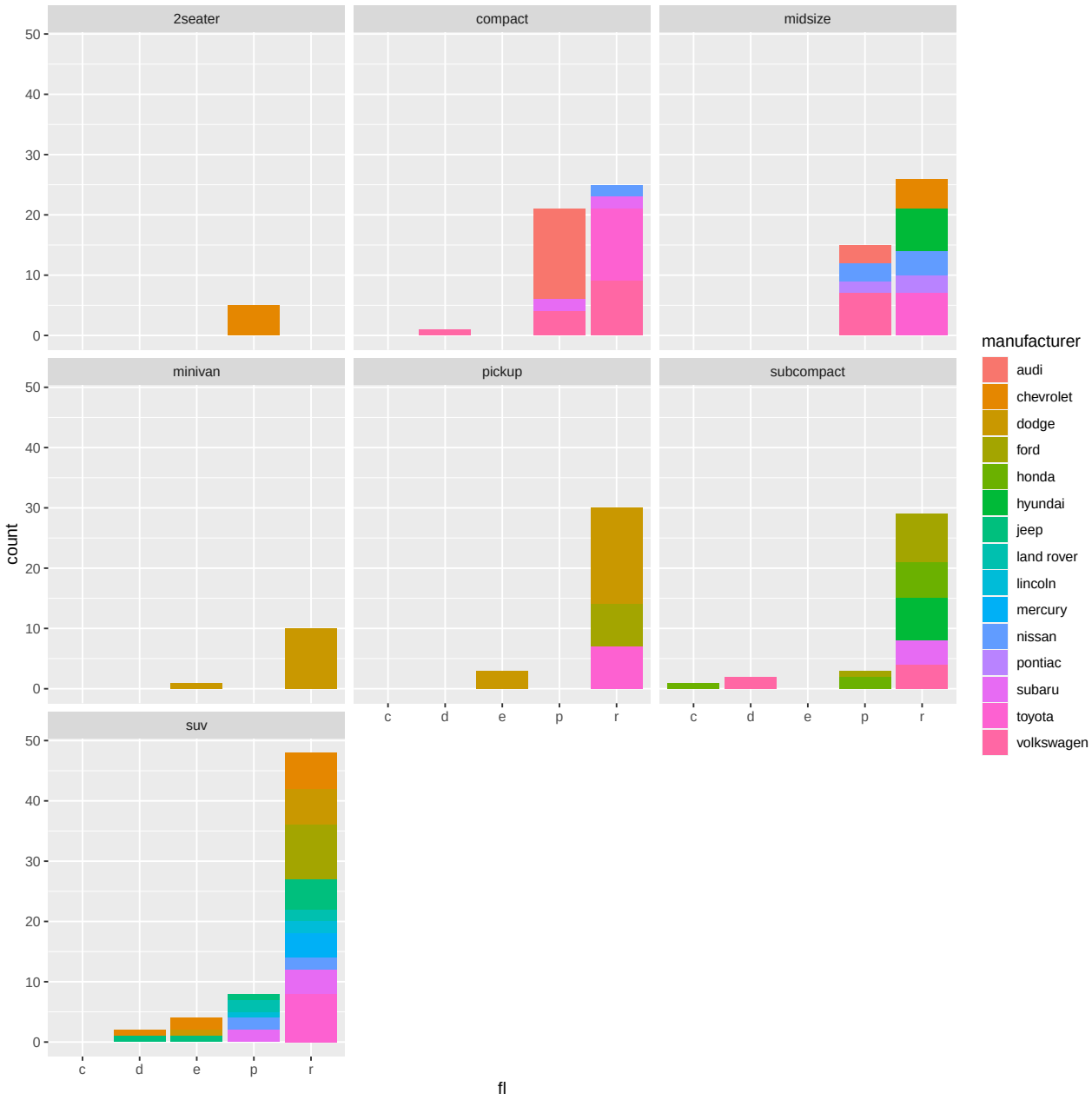
***

3.5 Attributes affected by manufacturer?

(e) Given the type of car, the attributes could also be affected by the manufacturer. Explore graphically if this is the case.

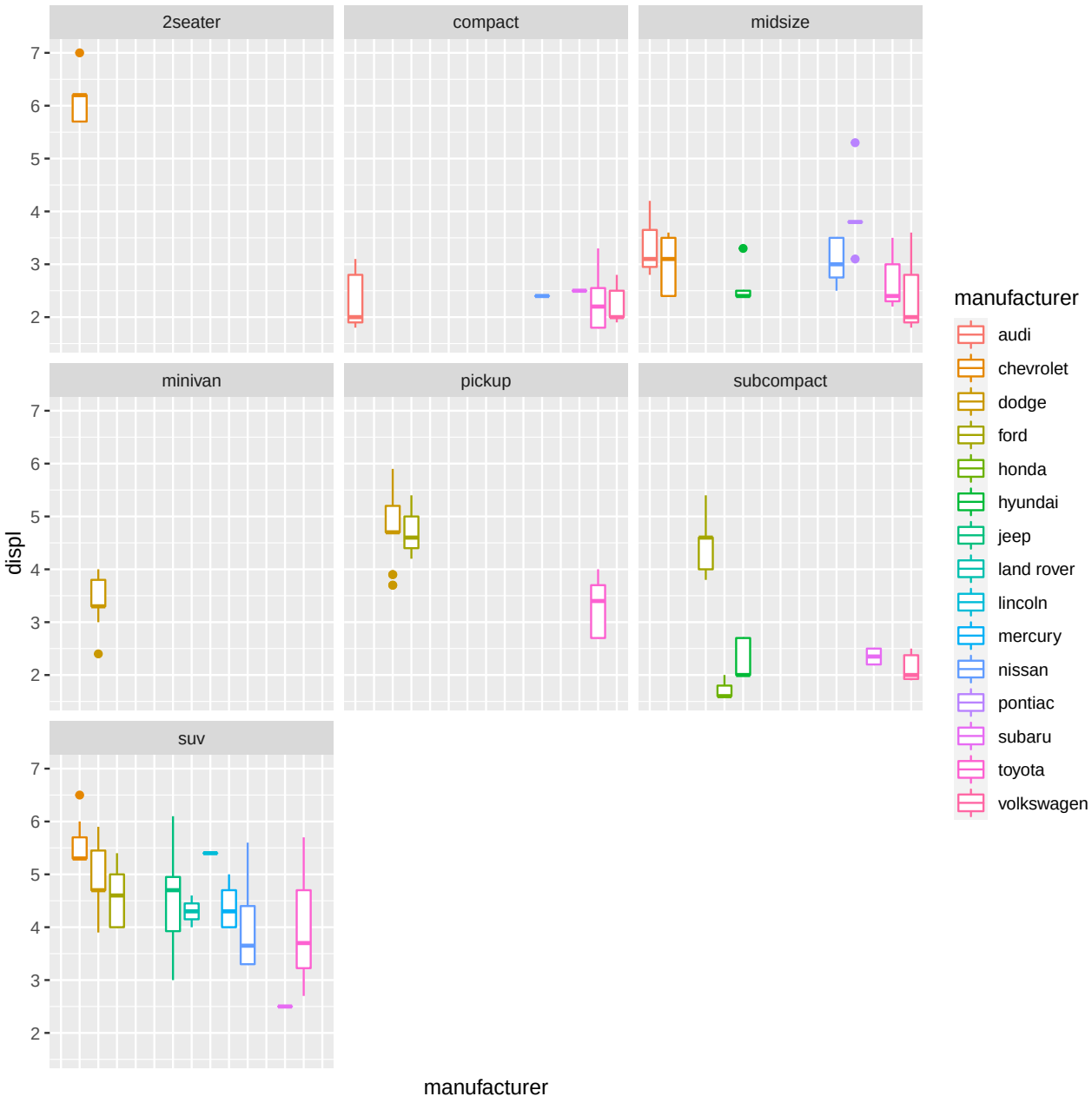
The relationship between two categorical variables can be visualized with stacked barplot. For example, relationship between "fl" and "manufacturer" is shown below. For each type of car, each barplot represents the count of certain fuel type. We can see that cars with fuel type "r" have the most diversified manufacturer.

```
ggplot(mpg,
       aes(x = fl,
           fill = manufacturer)) +
  geom_bar(position = "stack") +
  facet_wrap(~class)
```



On the other hand, the set of box plots below shows the relationship between continuous variable (eg.displ) and categorical variable (eg.manufacturer). For example, midsize cars manufactured by Pontiac have some outliers in engine displacement in liters (displ). Also for subcompact cars, cars manufactured by Ford have the highest engine displacement among them.

```
ggplot(mpg,
       aes(y = displ,
           x = manufacturer,
           color = manufacturer)) +
  geom_boxplot() +
  facet_wrap(~class) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```


2seater
compact
midsize
minivan
pickup
subcompact
suv
displ
manufacturer
manufacturer
audi
chevrolet
dodge
ford
honda
hyundai
jeep
land rover
lincoln
mercury
nissan
pontiac
subaru
toyota
volkswagen