Assignment 2 - Simple and Multiple Linear Regression (I)

B. Practical

Overview of the steps

1. Load the data and get an overview of the data
2. Perform simple linear regressions
3. Use the simple linear regression models
4. Perform multiple linear regressions
5. Use the multiple linear regression model

Steps in detail

Load the data and get an overview of the data

Load the data file `Boston.rda` or `Boston.csv` .

In `R` the dataframe comes with the `MASS` library. We save the dataframe ones in `csv` and `rda` files for later use.

In [1]:

```
library(MASS)
#write.csv(Boston,"../ISLR/data/Boston.csv", row.names = TRUE)
#save(Boston,file="../ISLR/data/Boston.rda")
```

Display the number of predictors (including the response `medv`) and their names:

In [2]:

```
dim(Boston)[2]
names(Boston)
```

14

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

Print a statistic summary of the predictors and the response `medv` :

```
In [3]: summary(Boston)
```

```
      crim                zn             indus            chas        
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
 1st Qu.: 0.08204   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          black       
 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  
```

Display the number of data points:

```
In [4]: dim(Boston)[1]
```

```
506
```

Display the data in a table (subset of rows is sufficient):

In [5]:

```
Boston
```

A data.frame: 506 × 14

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black
	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90
2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90
3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83
4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63
5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90
6	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12
7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60
8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90
9	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63
10	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71
11	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52
12	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.90
13	0.09378	12.5	7.87	0	0.524	5.889	39.0	5.4509	5	311	15.2	390.50
14	0.62976	0.0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21.0	396.90
15	0.63796	0.0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	380.02
16	0.62739	0.0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21.0	395.62
17	1.05393	0.0	8.14	0	0.538	5.935	29.3	4.4986	4	307	21.0	386.85
18	0.78420	0.0	8.14	0	0.538	5.990	81.7	4.2579	4	307	21.0	386.75
19	0.80271	0.0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21.0	288.99
20	0.72580	0.0	8.14	0	0.538	5.727	69.5	3.7965	4	307	21.0	390.95
21	1.25179	0.0	8.14	0	0.538	5.570	98.1	3.7979	4	307	21.0	376.57
22	0.85204	0.0	8.14	0	0.538	5.965	89.2	4.0123	4	307	21.0	392.53
23	1.23247	0.0	8.14	0	0.538	6.142	91.7	3.9769	4	307	21.0	396.90
24	0.98843	0.0	8.14	0	0.538	5.813	100.0	4.0952	4	307	21.0	394.54
25	0.75026	0.0	8.14	0	0.538	5.924	94.1	4.3996	4	307	21.0	394.33
26	0.84054	0.0	8.14	0	0.538	5.599	85.7	4.4546	4	307	21.0	303.42
27	0.67191	0.0	8.14	0	0.538	5.813	90.3	4.6820	4	307	21.0	376.88
28	0.95577	0.0	8.14	0	0.538	6.047	88.8	4.4534	4	307	21.0	306.38
29	0.77299	0.0	8.14	0	0.538	6.495	94.4	4.4547	4	307	21.0	387.94
30	1.00245	0.0	8.14	0	0.538	6.674	87.3	4.2390	4	307	21.0	380.23
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
477	4.87141	0	18.10	0	0.614	6.484	93.6	2.3053	24	666	20.2	396.21
478	15.02340	0	18.10	0	0.614	5.304	97.3	2.1007	24	666	20.2	349.48
479	10.23300	0	18.10	0	0.614	6.185	96.7	2.1705	24	666	20.2	379.70
480	14.33370	0	18.10	0	0.614	6.229	88.0	1.9512	24	666	20.2	383.32

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black
	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
481	5.82401	0	18.10	0	0.532	6.242	64.7	3.4242	24	666	20.2	396.90
482	5.70818	0	18.10	0	0.532	6.750	74.9	3.3317	24	666	20.2	393.07
483	5.73116	0	18.10	0	0.532	7.061	77.0	3.4106	24	666	20.2	395.28
484	2.81838	0	18.10	0	0.532	5.762	40.3	4.0983	24	666	20.2	392.92
485	2.37857	0	18.10	0	0.583	5.871	41.9	3.7240	24	666	20.2	370.73
486	3.67367	0	18.10	0	0.583	6.312	51.9	3.9917	24	666	20.2	388.62
487	5.69175	0	18.10	0	0.583	6.114	79.8	3.5459	24	666	20.2	392.68
488	4.83567	0	18.10	0	0.583	5.905	53.2	3.1523	24	666	20.2	388.22
489	0.15086	0	27.74	0	0.609	5.454	92.7	1.8209	4	711	20.1	395.09
490	0.18337	0	27.74	0	0.609	5.414	98.3	1.7554	4	711	20.1	344.05
491	0.20746	0	27.74	0	0.609	5.093	98.0	1.8226	4	711	20.1	318.43
492	0.10574	0	27.74	0	0.609	5.983	98.8	1.8681	4	711	20.1	390.11
493	0.11132	0	27.74	0	0.609	5.983	83.5	2.1099	4	711	20.1	396.90
494	0.17331	0	9.69	0	0.585	5.707	54.0	2.3817	6	391	19.2	396.90
495	0.27957	0	9.69	0	0.585	5.926	42.6	2.3817	6	391	19.2	396.90
496	0.17899	0	9.69	0	0.585	5.670	28.8	2.7986	6	391	19.2	393.29
497	0.28960	0	9.69	0	0.585	5.390	72.9	2.7986	6	391	19.2	396.90
498	0.26838	0	9.69	0	0.585	5.794	70.6	2.8927	6	391	19.2	396.90
499	0.23912	0	9.69	0	0.585	6.019	65.3	2.4091	6	391	19.2	396.90
500	0.17783	0	9.69	0	0.585	5.569	73.5	2.3999	6	391	19.2	395.77
501	0.22438	0	9.69	0	0.585	6.027	79.7	2.4982	6	391	19.2	396.90
502	0.06263	0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99
503	0.04527	0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90
504	0.06076	0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90
505	0.10959	0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45
506	0.04741	0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90

Plot some predictors (at least two) against the response values. We choose `lstat` , `rm` , and `age` .

In `R` , we need to download and install a library first.

In [6]:

```r
install.packages("ggpubr")
library("ggpubr")
```

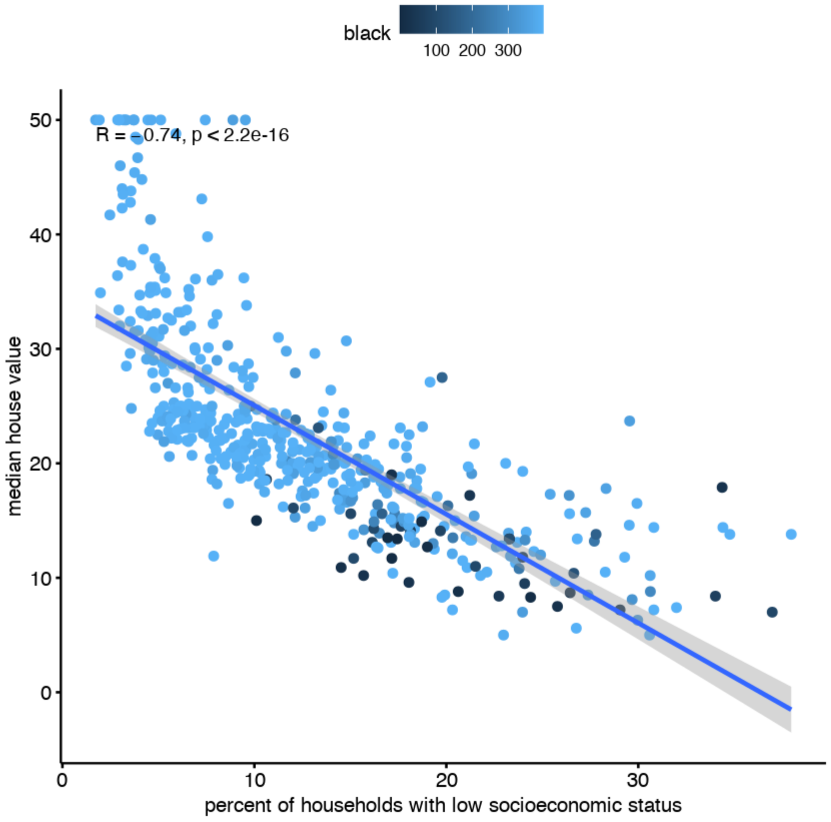
```
  There is a binary version available but the source version is late
r:
       binary source needs_compilation
ggpubr  0.4.0  0.5.0             FALSE


installing the source package ‘ggpubr’


Loading required package: ggplot2

Warning message:
“replacing previous import ‘ellipsis::check_dots_unnamed’ by ‘rlan
g::check_dots_unnamed’ when loading ‘tibble’”
Warning message:
“replacing previous import ‘ellipsis::check_dots_used’ by ‘rlang::ch
eck_dots_used’ when loading ‘tibble’”
Warning message:
“replacing previous import ‘ellipsis::check_dots_empty’ by ‘rlang::c
heck_dots_empty’ when loading ‘tibble’”
```

The `R` function `ggscatter` even displays a regression line, confidence intervals, the Pearson coefficient of correlation, and the p value. **This is not necessary at this stage.**

In [7]:

```r
ggscatter(Boston, x = "lstat", y = "medv",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "percent of households with low socioeconomic sta
          ylab = "median house value")
```

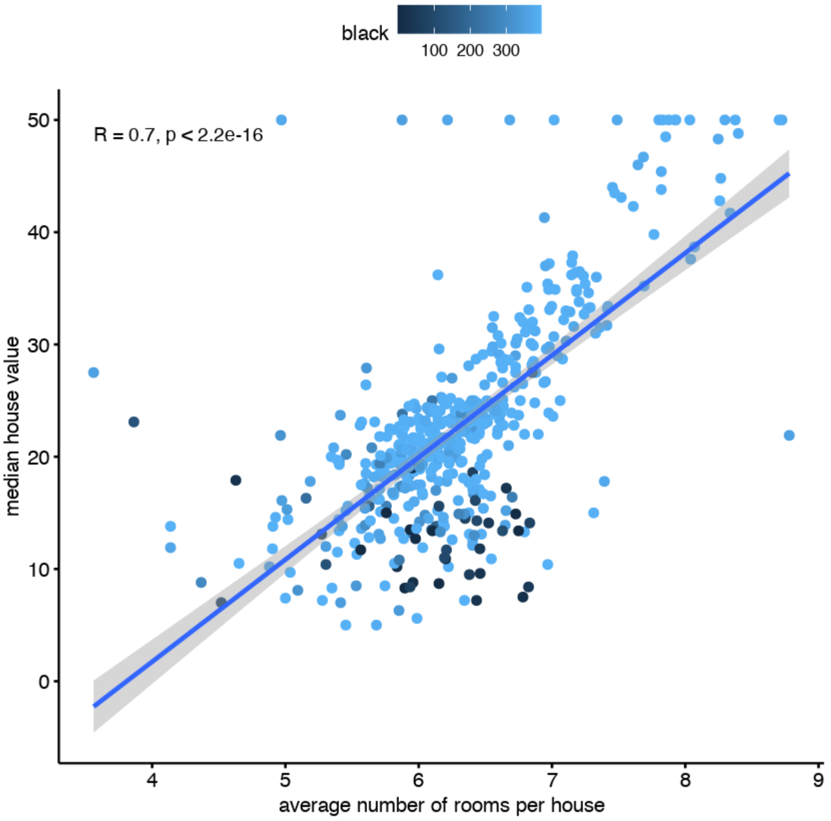
```
`geom_smooth()` using formula = 'y ~ x'
Warning message:
“The following aesthetics were dropped during statistical transforma
tion:
colour, fill
ℹ This can happen when ggplot fails to infer the correct grouping st
ructure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a nume
rical
  variable into a factor?”
```



In [8]:

```
ggscatter(Boston, x = "rm", y = "medv",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "average number of rooms per house",
          ylab = "median house value")
```

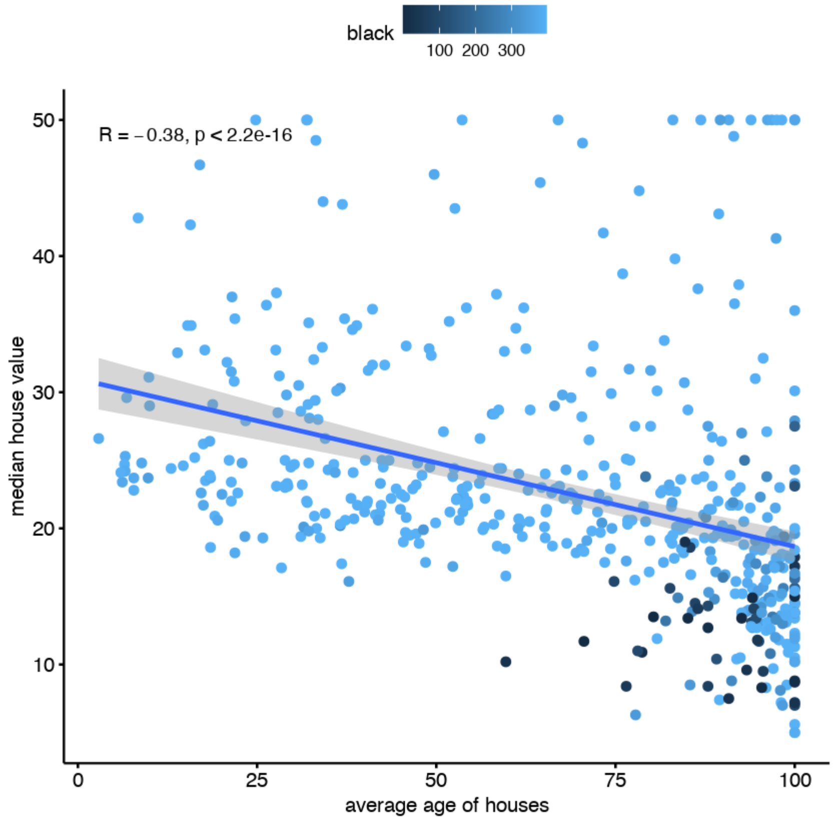
```
`geom_smooth()` using formula = 'y ~ x'
Warning message:
“The following aesthetics were dropped during statistical transforma
tion:
colour, fill
ℹ This can happen when ggplot fails to infer the correct grouping st
ructure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a nume
rical
  variable into a factor?”
```



In [9]:

```r
ggscatter(Boston, x = "age", y = "medv",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "average age of houses",
          ylab = "median house value")
```

```
`geom_smooth()` using formula = 'y ~ x'
Warning message:
“The following aesthetics were dropped during statistical transforma
tion:
colour, fill
ℹ This can happen when ggplot fails to infer the correct grouping st
ructure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a nume
rical
  variable into a factor?”
```



Perform simple linear regressions

Fit a simple linear regression model, with `medv` as the response and some (at least two) predictors individually. We choose `lstat` , `rm` , and `age` .

In [10]:

```
lm.fit=lm(medv~lstat ,data=Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ lstat, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.168  -3.990  -1.318   2.034  24.500 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.216 on 504 degrees of freedom
Multiple R-squared:  0.5441,    Adjusted R-squared:  0.5432 
F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

In [11]:

```
lm.fit=lm(medv~rm ,data=Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ rm, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.346  -2.547   0.090   2.986  39.433 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,    Adjusted R-squared:  0.4825 
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

In [12]:

```
lm.fit=lm(medv~age ,data=Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.097  -5.138  -1.958   2.397  31.338 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 30.97868    0.99911  31.006   <2e-16 ***
age         -0.12316    0.01348  -9.137   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.527 on 504 degrees of freedom
Multiple R-squared:  0.1421,    Adjusted R-squared:  0.1404 
F-statistic: 83.48 on 1 and 504 DF,  p-value: < 2.2e-16
```

Interprete the results. *Your interpretation of the results goes here!*

Obtain a confidence interval for the coefficient estimates for the individual models.

In [13]:

```
lm.fit=lm(medv~lstat ,data=Boston)
confint(lm.fit)
```

A matrix: 2 × 2 of type dbl

	2.5 %	**97.5 %**
(Intercept)	33.448457	35.6592247
lstat	-1.026148	-0.8739505

In [14]:

```
lm.fit=lm(medv~rm ,data=Boston)
confint(lm.fit)
```

A matrix: 2 × 2 of type dbl

	2.5 %	**97.5 %**
(Intercept)	-39.876641	-29.464601
rm	8.278855	9.925363

In [15]:

```
lm.fit=lm(medv~age ,data=Boston)
confint(lm.fit)
```

A matrix: 2 × 2 of type dbl

	2.5 %	**97.5 %**
(Intercept)	29.0157516	32.94160395
age	-0.1496469	-0.09667852

Interprete the results. *Your interpretation of the results goes here!*

Use the simple linear regression models

Predict the `medv` response values for some selected predictor values. Calculate the prediction intervals for these values.

In [16]:

```r
lm.fit=lm(medv~lstat,data=Boston)
predict(lm.fit,data.frame(lstat=c(5,10,15)), interval ="prediction
```

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	29.80359	17.565675	42.04151
2	25.05335	12.827626	37.27907
3	20.30310	8.077742	32.52846

In [17]:

```r
lm.fit=lm(medv~rm,data=Boston)
predict(lm.fit,data.frame(rm=c(5,6.5,8)), interval ="prediction")
```

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	10.83992	-2.214474	23.89432
2	24.49309	11.480391	37.50578
3	38.14625	25.058353	51.23415

In [18]:

```r
lm.fit=lm(medv~age,data=Boston)
predict(lm.fit,data.frame(age=c(25,50,75)), interval ="prediction'
```

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	27.89961	11.090368	44.70885
2	24.82054	8.043748	41.59734
3	21.74147	4.971031	38.51192

Interprete the results. *Your interpretation of the results goes here!*

Perform multiple linear regressions

Fit `medv` as response with the predictors selected before altogether.

In [19]:

```
lm.fit=lm(medv~lstat+rm+age ,data=Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ lstat + rm + age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-18.210  -3.467  -1.053   1.957  27.500 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.175311   3.181924  -0.369    0.712    
lstat       -0.668513   0.054357 -12.298   <2e-16 ***
rm           5.019133   0.454306  11.048   <2e-16 ***
age          0.009091   0.011215   0.811    0.418    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.542 on 502 degrees of freedom
Multiple R-squared:  0.639,	Adjusted R-squared:  0.6369 
F-statistic: 296.2 on 3 and 502 DF,  p-value: < 2.2e-16
```

Interprete the results. *Your interpretation of the results goes here!*

Fit `medv` as response with all available predictors altogether.

In [20]:

```
lm.fit=lm(medv~. ,data=Boston)
summary(lm.fit)
```

```

Call:
lm(formula = medv ~ ., data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 ** 
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288    
chas         2.687e+00  8.616e-01   3.118 0.001925 ** 
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229    
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 ** 
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,	Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Interprete the results. *Your interpretation of the results goes here!*

In [21]:

```
install.packages("corrplot")
source("http://www.sthda.com/upload/rquery_cormat.r")
```

```
  There is a binary version available but the source version is late
r:
         binary source needs_compilation
corrplot   0.89   0.92             FALSE


installing the source package ‘corrplot’
```

Check the correlation between the predictors.

In R , we need to download and install a library and an external function first.

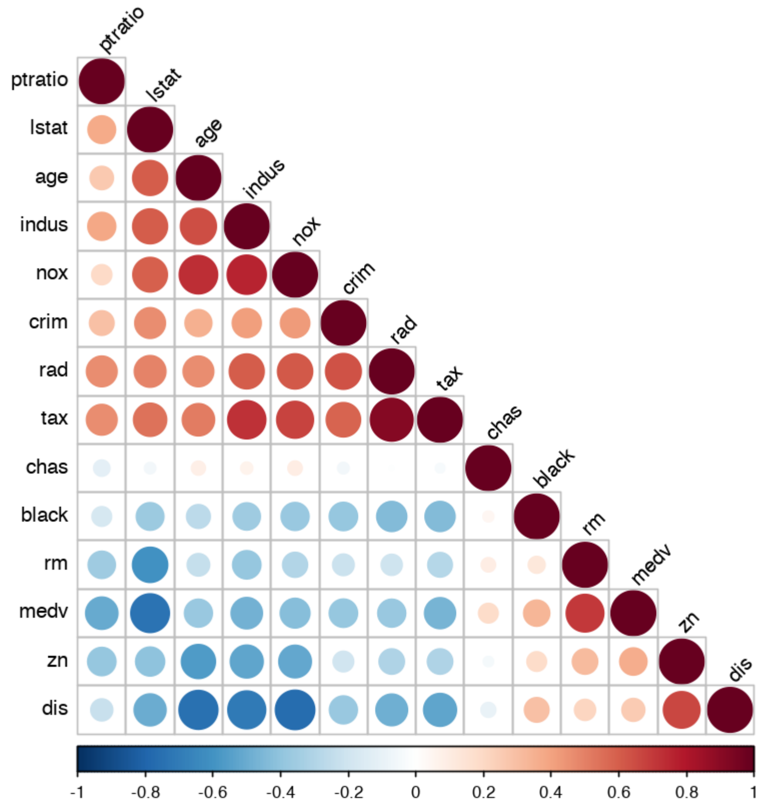
In [22]:

```
rquery.cormat(Boston)
```

```
corrplot 0.92 loaded
```

```
$r
        ptratio  lstat   age indus   nox   crim     rad    tax   cha
s black
ptratio       1
lstat      0.37      1
age        0.26    0.6     1
indus      0.38    0.6  0.64     1
nox        0.19   0.59  0.73  0.76     1
crim       0.29   0.46  0.35  0.41  0.42      1
rad        0.46   0.49  0.46   0.6  0.61   0.63       1
tax        0.46   0.54  0.51  0.72  0.67   0.58    0.91      1
chas      -0.12 -0.054 0.087 0.063 0.091 -0.056 -0.0074 -0.036
1
black     -0.18  -0.37 -0.27 -0.36 -0.38  -0.39   -0.44  -0.44  0.04
9     1
rm        -0.36  -0.61 -0.24 -0.39  -0.3  -0.22   -0.21  -0.29  0.09
1  0.13
medv      -0.51  -0.74 -0.38 -0.48 -0.43  -0.39   -0.38  -0.47   0.1
8  0.33
zn        -0.39  -0.41 -0.57 -0.53 -0.52   -0.2   -0.31  -0.31 -0.04
3  0.18
dis       -0.23   -0.5 -0.75 -0.71 -0.77  -0.38   -0.49  -0.53 -0.09
9  0.29
          rm medv   zn dis
ptratio
lstat
age
indus
nox
crim
rad
tax
chas
black
rm         1
medv     0.7    1
zn      0.31 0.36    1
dis     0.21 0.25 0.66   1

$p
        ptratio   lstat     age   indus      nox    crim      rad
tax
ptratio       0
lstat     3e-18       0
age     2.3e-09 2.8e-51       0
indus   3.8e-19 1.4e-51 8.4e-61       0
nox     1.9e-05   6e-49 7.5e-86 7.9e-98        0
crim    2.9e-11 2.7e-27 2.9e-16 1.5e-21  3.8e-23       0
rad     1.8e-28 9.9e-32 2.4e-27 8.4e-50  3.3e-53 2.7e-56        0
tax     5.7e-28 2.6e-40 2.6e-34   3e-82  1.1e-66 2.4e-47 4.1e-195
0
chas     0.0062    0.23   0.052    0.16     0.04    0.21     0.87
0.42
black     6e-05 1.7e-17 3.9e-10 1.2e-16  7.8e-19 2.5e-19  6.6e-26 1.
4e-25
rm      1.6e-16   1e-53 4.5e-08 5.3e-20  3.8e-12 6.3e-07  1.9e-06 2.
1e-11
medv    1.6e-34 5.1e-88 1.6e-18 4.9e-31  7.1e-24 1.2e-19  5.5e-19 5.
6e-29
zn      5.3e-20 2.9e-22 7.6e-45 1.3e-38  7.2e-36 5.5e-06    7e-13 4.
4e-13
```

```
dis     1.2e-07 6.4e-33 9.9e-92 3.6e-78 4.2e-100 8.5e-19  1.4e-32
1e-38
           chas   black      rm    medv      zn dis
ptratio
lstat
age
indus
nox
crim
rad
tax
chas          0
black      0.27       0
rm         0.04  0.0039       0
medv    7.4e-05 1.3e-14 2.5e-74       0
zn         0.34 7.2e-05 6.9e-13 5.7e-17       0
dis       0.026 2.3e-11 3.2e-06 1.2e-08 9.7e-66   0

$sym
        ptratio lstat age indus nox crim rad tax chas black rm medv
zn dis
ptratio 1
lstat   .       1
age             .     1
indus   .       .     ,   1
nox             .     ,   ,     1
crim            .     .   .     .   1
rad     .       .     .   .     ,   ,    1
tax     .       .     .   ,     ,   .    *   1
chas                                             1
black           .         .     .   .    .   .        1
rm      .       ,         .                                 1
medv    .       ,     .   .     .   .    .   .        .     ,  1
zn      .       .     .   .     .        .   .              .  .
1
dis             .     ,   ,     ,   .    .   .
,  1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```



Interprete the results. *Your interpretation of the results goes here!*

Use the multiple linear regression model

Select some predictor values.

In [23]:

```
lstatC=c(5,10,15)
rmC=c(5,6.5,8)
selected_predictor_values = expand.grid(lstat = lstatC, rm = rmC)
selected_predictor_values
```

A data.frame: 9 × 2

lstat	rm
<dbl>	<dbl>
5	5.0
10	5.0
15	5.0
5	6.5
10	6.5
15	6.5
5	8.0
10	8.0
15	8.0

Predict the `medv` response values for some selected predictor values. Calculate the prediction intervals for these values.

In [24]:

```
lm.fit=lm(medv~lstat+rm ,data=Boston)
predict(lm.fit, selected_predictor_values, interval ="prediction")
```

A matrix: 9 × 3 of type dbl

	fit	lwr	upr
1	20.90388	9.889729	31.91802
2	17.69208	6.722152	28.66202
3	14.48029	3.537875	25.42271
4	28.54606	17.635923	39.45619
5	25.33427	14.437027	36.23150
6	22.12247	11.221204	33.02374
7	36.18824	25.225479	47.15100
8	32.97645	21.995024	43.95787
9	29.76466	18.747835	40.78148

Interprete the results. *Your interpretation of the results goes here!*