STAT 393 Assignment 3 Solutions

This assignment needs you to use `R`. **You should include all your `R` code and output. Ensure that you refer to relevant code and output explicitly.** As with earlier assignments, you can either copy and paste `R` code and output into a Word document that you then save as a pdf file, or you can create a pdf by knitting a file in `R Markdown`. Ensure your submission is your own, independent work. Any sources of information that you use which are external to STAT 393 course materials must be correctly cited/referenced.

Data set: Housing Values in Suburbs of Boston

The `Boston` data frame has 506 rows (observations) on 14 variables (columns):

- crim: per capita crime rate by town.
- zn: proportion of residential land zoned for lots over 25,000 sq.ft.
- indus: proportion of non-retail business acres per town.
- chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
- nox: nitrogen oxides concentration (parts per 10 million).
- rm: average number of rooms per dwelling.
- age: proportion of owner-occupied units built prior to 1940.
- dis: weighted mean of distances to five Boston employment centres.
- rad: index of accessibility to radial highways.
- tax: full-value property-tax rate per $10,000.
- ptratio: pupil-teacher ratio by town.
- black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
- lstat: lower status of the population (percent).
- medv: median value of owner-occupied homes in $1000s.

Install the `MASS` package to access the `Boston` data set.

```
library(MASS)
attach(Boston)
head(Boston)
```

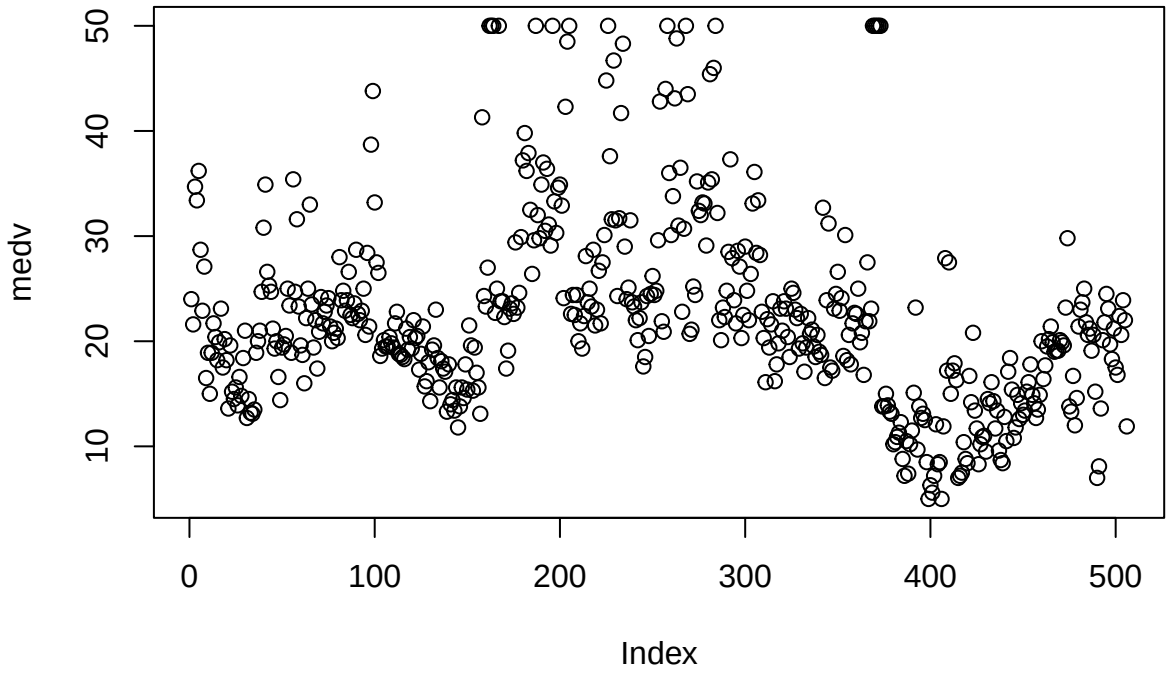
```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

1. [2 marks] Define a new variable, the reciprocal of `lstat`, as follows:

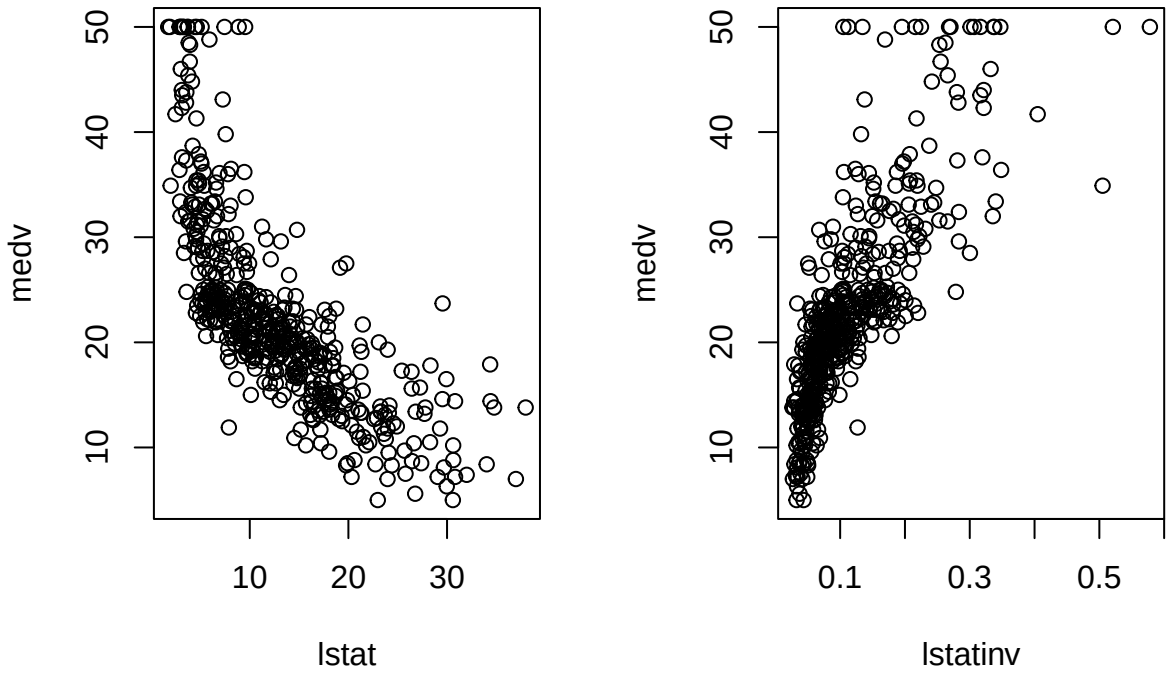
```
lstatinv <- 1/lstat
```

Although not asked to do so explicitly, it is always sensible to initially look at data graphically. So let's plot those two functions of `lstat` (i.e. untransformed and reciprocal) against the response variable from the models in Question 2, and also look at a graphical summary of the bivariate relationships among the variables in those models.

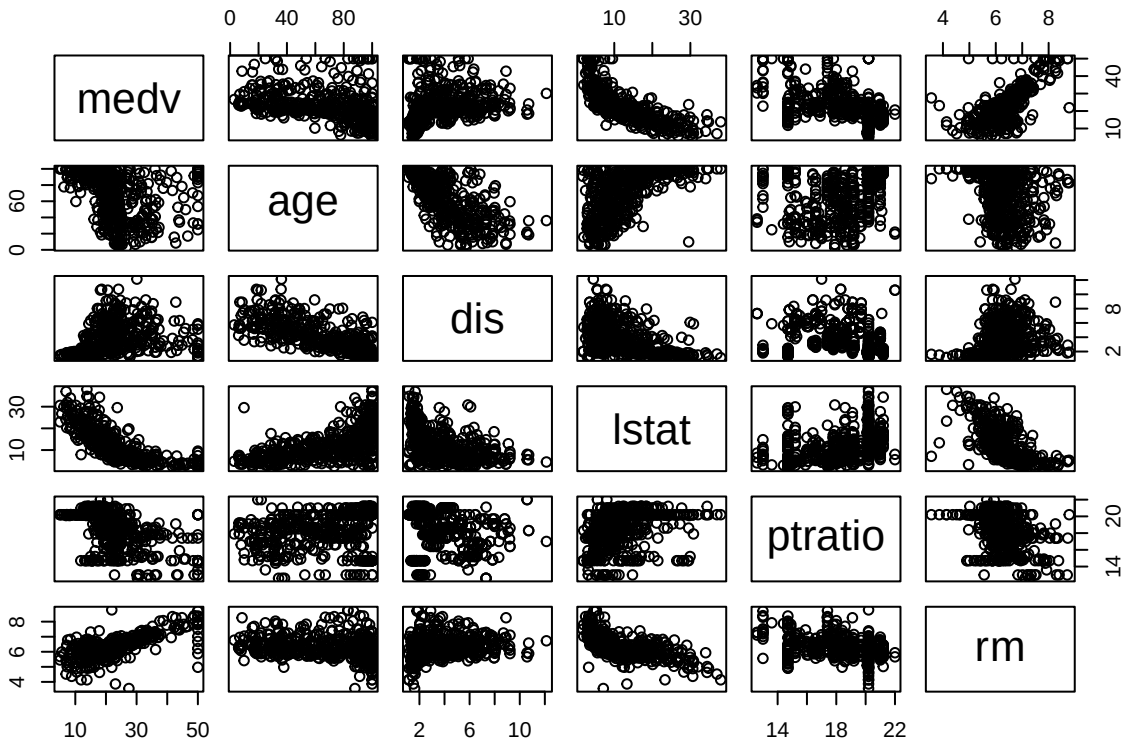
```
plot(medv)
```



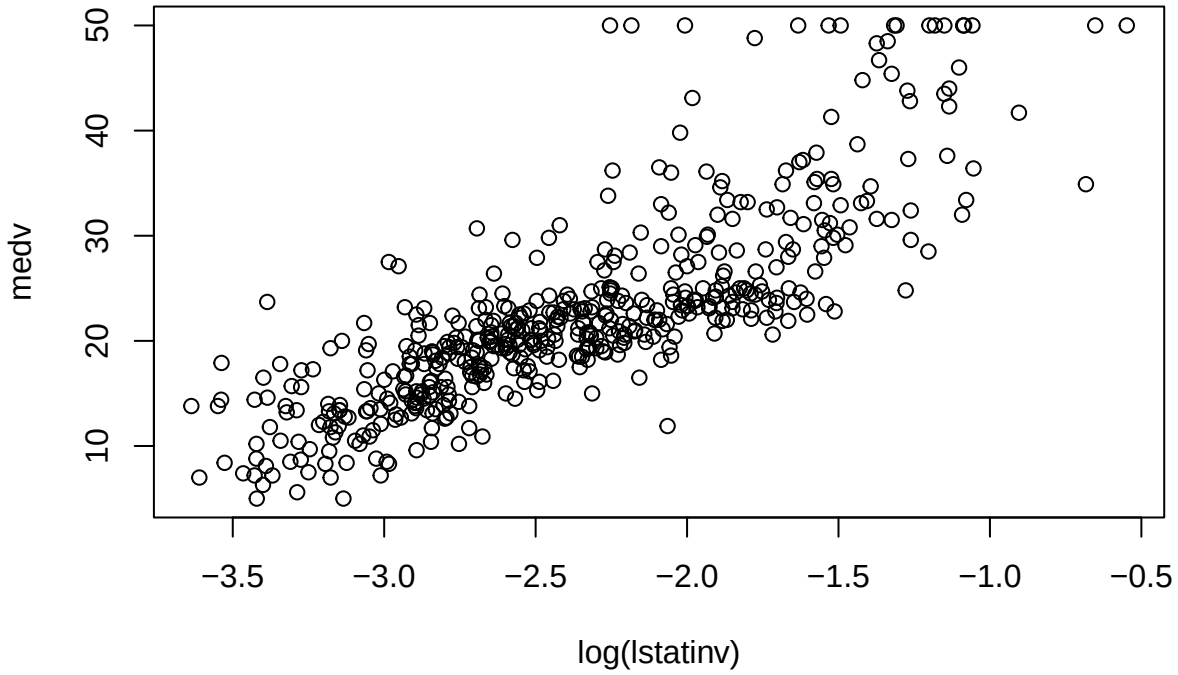

```
par(mfrow = c(1, 2))
plot(lstat,medv)
plot(lstatinv,medv)
```




```
par(mfrow = c(1, 1))
pairs(medv ~ age + dis + lstat + ptratio + rm)
```




```
plot(log(lstatinv),medv)
```



Note the 'flat line' of points with identical values of `medv` = 50. That indicates a likely problem with any models that assume the response variable has a continuous probability distribution (e.g. a Normal distribution). Also refer to Questions 15 and 18.

2. [12 marks] Fit the following six models, each with explanatory variables entered in alphabetical order:

$$
\begin{aligned}
\text{model1:} \quad & \text{medv} = \beta_0 + \beta_2 \text{dis} + \beta_3 \text{lstat} + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon \\
\text{model2:} \quad & \text{medv} = \beta_0 + \beta_1 \text{age} + \beta_2 \text{dis} + \beta_3 \text{lstat} + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon \\
\text{model3:} \quad & \text{medv} = \beta_0 + \beta_2 \text{dis} + \beta_3 \text{lstatinv} + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon \\
\text{model4:} \quad & \text{medv} = \beta_0 + \beta_1 \text{age} + \beta_2 \text{dis} + \beta_3 \text{lstatinv} + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon \\
\text{model5:} \quad & \text{medv} = \beta_0 + \beta_2 \text{dis} + \beta_3 \log(\text{lstatinv}) + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon \\
\text{model6:} \quad & \text{medv} = \beta_0 + \beta_1 \text{age} + \beta_2 \text{dis} + \beta_3 \log(\text{lstatinv}) + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon
\end{aligned}
$$

Include R output from the `summary()` command for each of the six models.

```
# Model 1
model1 <- lm(medv ~ dis + lstat + ptratio + rm)
summary(model1)
```

```
##
## Call:
## lm(formula = medv ~ dis + lstat + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.4172  -3.0971  -0.6397   1.8727  27.1088
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.47136    4.07802   6.001 3.77e-09 ***
## dis         -0.55193    0.12695  -4.348 1.67e-05 ***
## lstat       -0.66544    0.04675 -14.233  < 2e-16 ***
## ptratio     -0.97365    0.11603  -8.391 4.94e-16 ***
## rm           4.22379    0.42382   9.966  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.139 on 501 degrees of freedom
## Multiple R-squared:  0.6903, Adjusted R-squared:  0.6878
## F-statistic: 279.2 on 4 and 501 DF,  p-value: < 2.2e-16
```

```
# Model 2
model2 <- lm(medv ~ age + dis + lstat + ptratio + rm)
summary(model2)
```

```
##
## Call:
## lm(formula = medv ~ age + dis + lstat + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.4589  -2.9756  -0.5301   1.7054  27.7711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.33065    4.11134   6.161 1.49e-09 ***
## age         -0.02060    0.01350  -1.526    0.128
## dis         -0.71106    0.16414  -4.332 1.79e-05 ***
## lstat       -0.63468    0.05085 -12.481  < 2e-16 ***
## ptratio     -0.96601    0.11599  -8.329 7.90e-16 ***
## rm           4.32360    0.42828  10.095  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.132 on 500 degrees of freedom
## Multiple R-squared:  0.6917, Adjusted R-squared:  0.6887
## F-statistic: 224.4 on 5 and 500 DF,  p-value: < 2.2e-16
```

```r
# Model 3
model3 <- lm(medv ~ dis + lstatinv + ptratio + rm)
summary(model3)
```

```
##
## Call:
## lm(formula = medv ~ dis + lstatinv + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.3418  -2.6693  -0.0245   2.4907  29.6981
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.3414     3.4316   2.722  0.00671 **
## dis          -0.2761     0.1107  -2.495  0.01293 *
## lstatinv     67.2507     3.9067  17.214  < 2e-16 ***
## ptratio      -0.8553     0.1098  -7.787 3.97e-14 ***
## rm            3.5704     0.4045   8.826  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.827 on 501 degrees of freedom
## Multiple R-squared:  0.7267, Adjusted R-squared:  0.7245
## F-statistic: 333.1 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
# Model 4
model4 <- lm(medv ~ age + dis + lstatinv + ptratio + rm)
summary(model4)
```

```
##
## Call:
## lm(formula = medv ~ age + dis + lstatinv + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2835  -2.6922  -0.0058   2.3889  29.6777
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.16188    3.57450   3.123  0.00190 **
## age         -0.02194    0.01236  -1.775  0.07648 .
## dis         -0.46595    0.15374  -3.031  0.00257 **
## lstatinv    64.76166    4.14282  15.632  < 2e-16 ***
## ptratio     -0.84856    0.10967  -7.738 5.64e-14 ***
## rm           3.65969    0.40680   8.996  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.817 on 500 degrees of freedom
## Multiple R-squared:  0.7284, Adjusted R-squared:  0.7257
## F-statistic: 268.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

```r
# Model 5
model5 <- lm(medv ~ dis + log(lstatinv) + ptratio + rm)
summary(model5)
```

```
##
## Call:
## lm(formula = medv ~ dis + log(lstatinv) + ptratio + rm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.837  -2.711  -0.302   2.060  26.233
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    46.6019     4.1031  11.358  < 2e-16 ***
## dis            -0.7011     0.1126  -6.224 1.03e-09 ***
## log(lstatinv)  10.2642     0.5270  19.476  < 2e-16 ***
## ptratio        -0.8021     0.1048  -7.655 1.00e-13 ***
## rm              2.8214     0.3985   7.080 4.89e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.594 on 501 degrees of freedom
## Multiple R-squared:  0.7525, Adjusted R-squared:  0.7505
## F-statistic: 380.8 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
# Model 6
model6 <- lm(medv ~ age + dis + log(lstatinv) + ptratio + rm)
summary(model6)
```

```
##
## Call:
## lm(formula = medv ~ age + dis + log(lstatinv) + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8061  -2.7923  -0.3163   2.0477  26.1529
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   46.58415    4.10426  11.350  < 2e-16 ***
## age            0.01053    0.01238   0.851    0.395
## dis           -0.62124    0.14661  -4.237 2.69e-05 ***
## log(lstatinv) 10.48615    0.58814  17.829  < 2e-16 ***
## ptratio       -0.80281    0.10482  -7.659 9.78e-14 ***
## rm             2.74679    0.40812   6.730 4.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.595 on 500 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7504
## F-statistic: 304.6 on 5 and 500 DF,  p-value: < 2.2e-16
```

For the next six questions (Q3-Q8), recall that if `modelA` has explanatory variables that are a subset of those in `modelB`, a sequential F-test can be performed with the `R` command `anova(modelA, modelB)`. Also recall that `anova(modelA)` will break down the total sum of squares (SST) into terms explained sequentially by `modelA`, which add to give the regression sum of squares (SSR), along with the residual (or estimated error) sum of squares, SSE. Further, remember that models with the same dependent variable can be compared using `AIC()`, `BIC()`, adjusted R^2 [i.e. $(\bar{R}^2)$], or residual standard error (RSE).

3. [2 marks] Based on AIC values, rank the six models fitted in Question 2 from best to worst, for the prediction/explanation of `medv` values.

```
# AIC values
AIC(model1, model2, model3, model4, model5, model6)
```

```
##        df      AIC
## model1  6 3099.359
## model2  7 3099.007
## model3  6 3036.066
## model4  7 3034.887
## model5  6 2985.959
## model6  7 2987.227
```

With AIC as a model selection criterion, **better models have lower AIC values**. So ranked from best to worst we have: model5, model6, model4, model3, model2, model1.

4. [2 marks] Based on adjusted R^2 values, rank the six models fitted in Question 2 from best to worst, for the prediction/explanation of `medv` values.

```
# adjusted R-squared values
c(summary(model1)$adj.r.squared,summary(model2)$adj.r.squared,summary(model3)$adj.r.squared,
  summary(model4)$adj.r.squared,summary(model5)$adj.r.squared,summary(model6)$adj.r.squared)
```

```
## [1] 0.6878351 0.6886616 0.7245388 0.7257165 0.7505093 0.7503718
```

With adjusted R^2 as a model selection criterion, **better models have higher adjusted R^2 values**. So ranked from best to worst we have: model5, model6, model4, model3, model2, model1. This ranking is identical to the ranking produced using AIC in Question 3. (Also see Question 9.)

5. [2 marks] Based on BIC values, rank the six models fitted in Question 2 from best to worst, for the prediction/explanation of `medv` values.

```
# BIC values
BIC(model1, model2, model3, model4, model5, model6)
```

```
##        df      BIC
## model1  6 3124.718
## model2  7 3128.592
## model3  6 3061.425
## model4  7 3064.473
## model5  6 3011.319
## model6  7 3016.813
```

With BIC as a model selection criterion, **better models have lower BIC values**. So ranked from best to worst we have: model5, model6, model3, model4, model1, model2.

6. [6 marks] Comment on/discuss the similarities and/or differences in the model rankings that you have provided in Questions 3 to 5.

Of the six models fitted in Question 2, there are two basic forms: a reduced model with four explanatory variables plus a constant term, and a full model with five explanatory variables plus a constant term. All six models have `medv` as the response variable. The fifth explanatory variable is `age`, which is the difference between the models in three pairs: `pairA = (model1, model2)`, `pairB = (model3, model4)` and `pairC = (model5, model6)`. In all those pairs of models, the reduced model (with fewer parameters, and which omits `age`) has the lower model number. All six models include the three explanatory variables `dis, ptratio, rm` and the fourth variable is `lstat` in `pairA`, `lstatinv` in `pairB` and `log(lstatinv)` in `pairC`. Note that the predicted values in `pairC` would be identical if `log(lstat)` was used in place of `log(lstatinv)`, but clearly the sign of the estimated coefficient would switch from positive to negative.

All three model ranking criteria order `pairC` best, followed by `pairB`, with `pairA` worst. Hence the use of log-transformed `lstat` as an explanatory variable is preferred to untransformed `lstat` or the reciprocal of `lstat`. Note that transformations of these types can always be considered for any explanatory variables, as well as for response variables (with the necessary condition that for any variable q, $\log(q)$ is only

defined for $q > 0$.). Hence the number of possible combinations of variable transformations within a linear model is large, whenever the number of explanatory variables is sizeable.

Overall, all three criteria favour the simpler model within `pairC` (i.e. model5). Hence model5 is ranked best by all three model selection methods. For BIC the simpler model is also preferred in `pairA` and `pairB`, while both AIC and adjusted R^2 prefer the more complex model, which includes `age`, in `pairA` and `pairB`. Note the t-tests for the coefficient on `age` in all three pairs of models do not reject the null hypothesis that the population coefficient is zero, using a classical 5% significance level. Hence sequential tests for the significance of `age` would never include that explanatory variable when all other variables are in the model, which agrees with the models selected by BIC. (Also see Questions 7 and 8.) Recall that both AIC and BIC are information criteria that penalize the maximised log-likelihood, with a penalty term that multiplies the total number of parameters in the model by 2 in the case of AIC and by $\log(n)$ in the case of BIC. Here, with $n = 506$, the penalty multiplier for each additional parameter is 2 (as always) for AIC and $\log(506) = 6.2265$ for BIC. With such a sizeable penalty term, it is not all that surprising that BIC always selects the reduced model within each pair. Note that is also because the **additional** contribution of `age` is small, to the variation in `medv` that the models can explain. Within model pairs, changes in R^2 (and adjusted R^2) are small.

7. [3 marks] Explain why the explanatory variable `age` is statistically significant in `anova(model6)` but not significant in `summary(model6)`.

```
anova(model6)
```

```
## Analysis of Variance Table
##
## Response: medv
##                Df  Sum Sq Mean Sq   F value    Pr(>F)
## age             1  6069.8  6069.8  287.4589 < 2.2e-16 ***
## dis             1    99.2    99.2    4.6974   0.03068 *
## log(lstatinv)   1 23488.6 23488.6 1112.4013 < 2.2e-16 ***
## ptratio         1  1544.6  1544.6   73.1533 < 2.2e-16 ***
## rm              1   956.5   956.5   45.2977 4.651e-11 ***
## Residuals     500 10557.6    21.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model6)
```

```
##
## Call:
## lm(formula = medv ~ age + dis + log(lstatinv) + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8061  -2.7923  -0.3163   2.0477  26.1529
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   46.58415    4.10426  11.350  < 2e-16 ***
## age            0.01053    0.01238   0.851    0.395
## dis           -0.62124    0.14661  -4.237 2.69e-05 ***
## log(lstatinv) 10.48615    0.58814  17.829  < 2e-16 ***
## ptratio       -0.80281    0.10482  -7.659 9.78e-14 ***
## rm             2.74679    0.40812   6.730 4.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.595 on 500 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7504
## F-statistic: 304.6 on 5 and 500 DF,  p-value: < 2.2e-16
```

The R command `anova(model6)` breaks down the total sum of squares (SST) into terms explained sequentially by the explanatory variables in `model6`, which add to give the regression sum of squares (SSR), along with the residual (or estimated error) sum of squares, SSE. Since the explanatory variables were entered alphabetically, `age` is entered first in `model6`, followed by `dis, log(lstatinv), ptratio` and `rm`. Hence the sum of squares explained by `age` is equivalent to fitting a simple regression of `medv` on `age`, which is highly significant, with a p-value of approximately zero. (In R try `anova(lm(medv~age))` and see!) Conversely, the t-test for the coefficient on `age` in `summary(model6)` tests the **additional** contribution that `age` makes to the explanation of variation in `medv`, when all the other explanatory variables (i.e. `dis, log(lstatinv), ptratio, rm`) are already included in the model. Using notation from Question 2, the relevant hypotheses for the t-test of `age` in `model6` are

$$H_0 : \beta_1|\beta_0, \beta_2, \beta_3, \beta_4, \beta_5 = 0 \text{ vs } H_1 : \beta_1|\beta_0, \beta_2, \beta_3, \beta_4, \beta_5 \neq 0.$$

The null hypothesis is not rejected at any conventional significance level, since the relevant p-value is 0.395. So `age` is not a useful linear predictor of `medv`, **given** the contributions of `dis, log(lstatinv), ptratio` and `rm`.

8. [3 marks] Discuss the result from `anova(model5, model6)` and the statistical significance of the explanatory variable `age` in `summary(model6)`. Include in your discussion explicit commentary about the F-statistic from `anova(model5, model6)` and the t-statistic on `age` in `summary(model6)`.

```
anova(model5,model6)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ dis + log(lstatinv) + ptratio + rm
## Model 2: medv ~ age + dis + log(lstatinv) + ptratio + rm
##   Res.Df   RSS Df Sum of Sq     F Pr(>F)
## 1    501 10573
## 2    500 10558  1    15.287 0.724 0.3953
```

The R command `anova(model5, model6)` performs a sequential F-test of the contribution of the **additional** explanatory variables in `model6`, compared to the contribution of the explanatory variables in `model5`. As already discussed in Questions 6 and 7, the only additional explanatory variable in `model6` is `age`. So the F-statistic from `anova(model5, model6)` and the t-statistic on `age` in `summary(model6)` are testing exactly the same null hypothesis, viz.

$$H_0 : \beta_1|\beta_0, \beta_2, \beta_3, \beta_4, \beta_5 = 0 \text{ vs } H_1 : \beta_1|\beta_0, \beta_2, \beta_3, \beta_4, \beta_5 \neq 0.$$

The value of the F-statistic is the value of the t-statistic squared: $0.724 = 0.851^2$. The p-values are necessarily identical ($= 0.395$) and the null hypothesis is not rejected at any conventional significance level. As noted in Question 7, `age` is not a useful linear predictor of `medv`, **given** the contributions of `dis, log(lstatinv), ptratio` and `rm`.

Consider only `model3` in Questions 9-17, that is:

$$\text{model3:} \qquad \text{medv} = \beta_0 + \beta_2 \text{dis} + \beta_3 \text{lstatinv} + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon$$

9. [2 marks] Give an interpretation of the R^2 value for `model3`.

```
# R-squared value for model3
summary(model3)$r.squared
```

```
## [1] 0.7267207
```

In a linear model, R^2 gives the proportion of the variation in the response variable, y, that is explained by the explanatory variables, $\mathbf{X}$. So R^2 divides the regression sum of squares by the total sum of squares, that is:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}.$$

Here, explanatory variables `dis, lstatinv, ptratio, rm` explain 72.67% of the variation in response variable `medv`.

Note, in the adjusted R^2, as used to rank models in Question 4, SSE and SST are each divided by their degrees of freedom, $n-p$ and $n-1$ respectively, where p is the number of estimated β coefficients. Adjusted R^2 can no longer be interpreted as the proportion of variation in y explained by $\mathbf{X}$, but when n is large and p is small (as here), the difference between R^2 and adjusted R^2 is necessarily small, since the ratio of $n-p$ and $n-1$ is close to unity. Asymptotically, R^2 and adjusted R^2 are equivalent.

10. [5 marks] Use the `predict()` function to compute a 95% confidence interval and a 95% prediction interval for the response variable (`medv`) in `model3`, evaluated at the mean values of all the explanatory variables in the model.

```
new <- data.frame(dis=mean(dis), lstatinv=mean(lstatinv), ptratio=mean(ptratio), rm=mean(rm))
new
```

```
##        dis  lstatinv  ptratio       rm
## 1 3.795043 0.1127939 18.45553 6.284634
```

```
# 95% CI for response variable medv:
predict(model3, newdata=new, interval="confidence")
```

```
##        fit     lwr      upr
## 1 22.53281 22.1112 22.95441
```

```
# 95% prediction interval for response variable medv:
predict(model3, newdata=new, interval="prediction")
```

```
##        fit      lwr      upr
## 1 22.53281 13.03969 32.02592
```

The 95% confidence and prediction intervals for `medv` are both centered on the same estimate of 22.53, which, as the intervals were calculated at the mean of the explanatory variables, is also the sample mean of `medv`:

```
mean(medv)
```

```
## [1] 22.53281
```

The confidence interval (for the mean of `medv`) is (22.11, 22.95) and the prediction interval (for individual values of `medv`) is (13.04, 32.03). Note that there is far more uncertainty associated with prediction of the individual response values than there is with prediction of the mean of the response variable. Note too that these intervals, calculated at the mean of the explanatory variables, are narrower than they would be at any other values of the explanatory variables.

For the next four questions (Q11-14), use matrix methods to reproduce for `model3` some of the standard results from the `lm()` command.

11. [2 marks] Use the function `model.matrix()` to create the design matrix $\mathbf{X}$ of `model3`, then print the first 10 rows of the matrix $\mathbf{X}$.

```
X<-model.matrix(model3)
X[1:10,]
```

```
##    (Intercept)    dis   lstatinv ptratio    rm
## 1            1 4.0900 0.20080321    15.3 6.575
## 2            1 4.9671 0.10940919    17.8 6.421
## 3            1 4.9671 0.24813896    17.8 7.185
## 4            1 6.0622 0.34013605    18.7 6.998
## 5            1 6.0622 0.18761726    18.7 7.147
## 6            1 6.0622 0.19193858    18.7 6.430
## 7            1 5.5605 0.08045052    15.2 6.012
## 8            1 5.9505 0.05221932    15.2 6.172
## 9            1 6.0821 0.03341129    15.2 5.631
## 10           1 6.5921 0.05847953    15.2 6.004
```

12. [2 marks] Calculate and print the LSE

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

```
y <- medv
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta_hat
```

```
##                    [,1]
## (Intercept)  9.3413774
## dis         -0.2760822
## lstatinv    67.2507079
## ptratio     -0.8552911
## rm           3.5703841
```

13. [2 marks] Calculate the predicted values

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

Print out the first 10 predicted values.

```
y_hat <- X %*% beta_hat
y_hat[1:10]
```

```
##  [1] 32.10568 23.02915 35.08660 39.53371 29.80870 27.53934 21.68130 20.24632
##  [9] 17.01356 19.89037
```

14. [8 marks] Calculate and print the residual standard error (RSE), then calculate the covariance matrix of $\hat{\boldsymbol{\beta}}$,

$$Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}.$$

Given $Var(\hat{\boldsymbol{\beta}})$, calculate and print the standard errors of the estimated regression coefficients, $\text{SE}(\hat{\boldsymbol{\beta}})$.

```
# Calculate and print RSE
SSE <- t(y-y_hat) %*% (y-y_hat)
n <- length(y)
p <- ncol(X)
RSE <- sqrt(SSE/(n-p))
RSE
```

```
##          [,1]
## [1,] 4.827045
```

```
# Calculate (and print - although not requested) the covariance matrix of beta_hat
sigma_sq_hat <- as.numeric(RSE^2)
Var_beta_hat <- sigma_sq_hat*solve(t(X) %*% X)
Var_beta_hat
```

```
##             (Intercept)          dis   lstatinv      ptratio           rm
## (Intercept) 11.77565657 -0.074310878  2.9831110 -0.278284986 -1.057845818
## dis         -0.07431088  0.012249179 -0.1262980  0.001309687  0.002848119
## lstatinv     2.98311096 -0.126298006 15.2622272  0.082170096 -0.913622490
## ptratio     -0.27828499  0.001309687  0.0821701  0.012063950  0.006587455
## rm          -1.05784582  0.002848119 -0.9136225  0.006587455  0.163655200
```

```
# Calculate and print standard errors of beta_hat
SE_beta <- sqrt(diag(Var_beta_hat))
SE_beta
```

```
## (Intercept)         dis    lstatinv     ptratio          rm
##   3.4315677   0.1106760   3.9066901   0.1098360   0.4045432
```

The above matrix methods have successfully reproduced some of the standard results from the `lm()` command, as can be seen in a comparison with `summary(model3)`:

```
summary(model3)
```

```
##
```

```
## Call:
## lm(formula = medv ~ dis + lstatinv + ptratio + rm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.3418  -2.6693  -0.0245   2.4907  29.6981
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.3414     3.4316   2.722  0.00671 **
## dis          -0.2761     0.1107  -2.495  0.01293 *
## lstatinv     67.2507     3.9067  17.214  < 2e-16 ***
## ptratio      -0.8553     0.1098  -7.787 3.97e-14 ***
## rm            3.5704     0.4045   8.826  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.827 on 501 degrees of freedom
## Multiple R-squared:  0.7267, Adjusted R-squared:  0.7245
## F-statistic: 333.1 on 4 and 501 DF,  p-value: < 2.2e-16
```

For the next three questions (Q15-17), make use of the diagnostic metrics available in the `augment()` function from package `broom`. You will probably find it helpful to use syntax from the `tidyverse` package as well; for example, to add index numbers to the rows of data, as in the lecture notes (see pp.6-7 in "Diagnostics for linear regression"). You should also include specific diagnostic plots, as appropriate. As above, only consider `model3` for these questions:

$$\text{model3:} \qquad \text{medv} = \beta_0 + \beta_2 \text{dis} + \beta_3 \text{lstatinv} + \beta_4 \text{ptratio} + \beta_5 \text{rm} + \varepsilon$$

15. a. [2 marks] State the maximum value of the response variable `medv` in the Boston data.
    b. [2 marks] Print out diagnostic information about the five observations with the largest fitted values from `model3`.

```
# Q15a
max(medv)
```

```
[1] 50
```

```
length(which(Boston$medv==50))
```

```
[1] 16
```

```
which(Boston$medv==50)
```

```
 [1] 162 163 164 167 187 196 205 226 258 268 284 369 370 371 372 373
```

15. a. The maximum value of the response variable `medv` is 50. Recall the comment made when looking at the plots displayed in Question 1: there are several ($n = 16$) values of `medv` = 50, which seems a 'hard limit' on the maximum value. Given `medv` is a median value of owner-occupied homes (in $1000s) such a 'hard limit' is somewhat intriguing: the data seems to be censored, and the observations that are censored are listed above. See Question 18 too.

15. b. The diagnostic information about the five observations with the largest fitted values from `model3` is given below. Note that four of those five fitted values are greater than the 'hard limit' of `medv` = 50, which guarantees that the residuals for those observations will be negative.

```
library(tidyverse)
library(broom)

# Q15b
model.diag.metrics3 <- augment(model3)
model.diag.metrics3 <- model.diag.metrics3 %>%
  mutate(index = 1:nrow(model.diag.metrics3)) %>%
```

```
  select(index, medv, .fitted, .resid, .hat, .sigma, .cooksd, .std.resid)
model.diag.metrics3 %>%
  top_n(5, wt = .fitted)
```

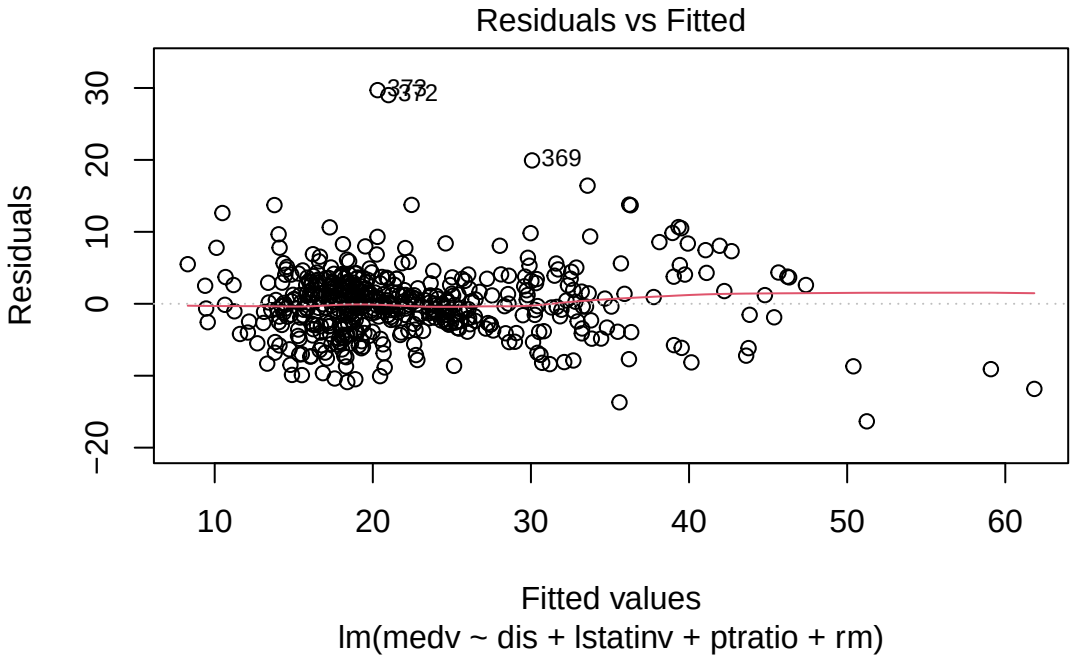
```
# A tibble: 5 x 8
  index  medv .fitted .resid   .hat .sigma .cooksd .std.resid
  <int> <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>      <dbl>
1    41  34.9    51.2 -16.3  0.0782   4.77 0.211       -3.53
2   162  50      61.8 -11.8  0.114    4.80 0.174       -2.60
3   163  50      59.1  -9.09 0.0814   4.81 0.0684      -1.96
4   205  50      47.4   2.61 0.0223   4.83 0.00136      0.547
5   233  41.7    50.4  -8.69 0.0375   4.82 0.0262      -1.84
```

16. a. [2 marks] Print out diagnostic information about the five observations with the largest (absolute value) residuals from `model3`.
    b. [2 marks] Print diagnostic plot 1, which displays Residuals vs Fitted values.
    c. [3 marks] How many observations have **standardized** residuals greater than 3 in absolute value, indicating they may be possible outliers?
    d. [2 marks] What percentage of the data (to 2dp) are possible outliers, using the definition from Q16c?

```
# Q16a
model.diag.metrics3 %>%
  top_n(5, wt = abs(.resid))
```

```
## # A tibble: 5 x 8
##   index  medv .fitted .resid    .hat .sigma .cooksd .std.resid
##   <int> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>      <dbl>
## 1    41  34.9    51.2  -16.3 0.0782    4.77  0.211       -3.53
## 2   369  50      30.1   19.9 0.0701    4.74  0.276        4.28
## 3   370  50      33.6   16.4 0.0244    4.77  0.0594       3.44
## 4   372  50      21.0   29.0 0.00635   4.65  0.0464       6.03
## 5   373  50      20.3   29.7 0.00779   4.64  0.0599       6.18
```

```
# Q16b
plot(model3,1)
```



```
# Q16c
length(which(abs(model.diag.metrics3$.std.resid) > 3))
```

```
## [1] 5
```

```
model.diag.metrics3[abs(model.diag.metrics3$.std.resid) > 3,]
```

```
## # A tibble: 5 x 8
##   index  medv .fitted .resid    .hat .sigma .cooksd .std.resid
##   <int> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>      <dbl>
## 1    41  34.9    51.2  -16.3 0.0782    4.77  0.211       -3.53
## 2   369  50      30.1   19.9 0.0701    4.74  0.276        4.28
## 3   370  50      33.6   16.4 0.0244    4.77  0.0594       3.44
## 4   372  50      21.0   29.0 0.00635   4.65  0.0464       6.03
## 5   373  50      20.3   29.7 0.00779   4.64  0.0599       6.18
```

16. c. There are 5 observations that have standardized residuals greater than 3 in absolute value: observations (41, 369, 370, 372, 373). Note that four of those five observations have values of `medv` equal to the maximum (censored) value of 50.

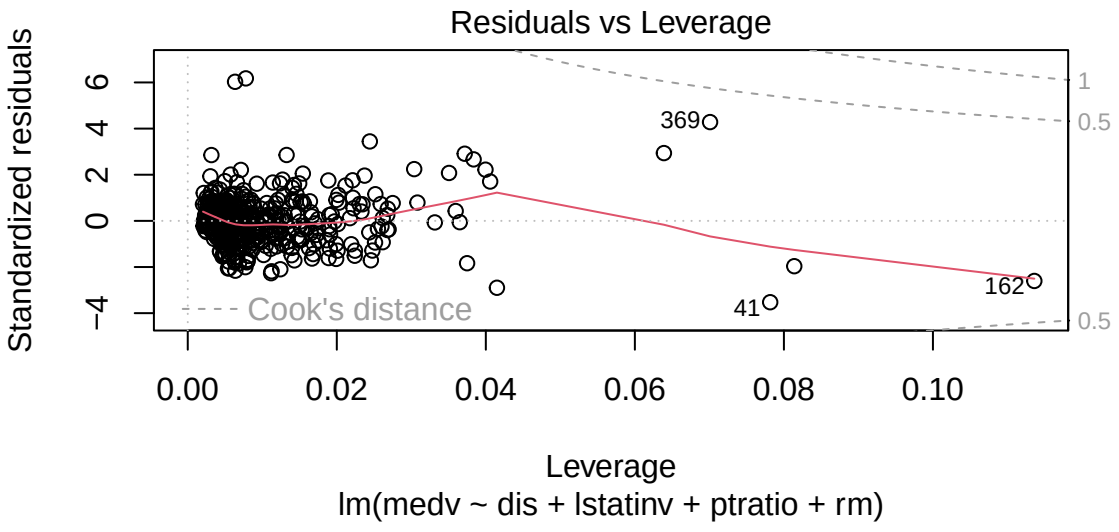
16. d. The proportion of the data that are possible outliers, using the definition from Q16c, is $5/n = 5/506 = 0.0099$. Hence the percentage of the data that are possible outliers is 0.99%.

17. a. [2 marks] Print out diagnostic information about the five observations with the largest leverages from `model3`.
    b. [2 marks] Print diagnostic plot 5, which displays the (standardized) Residuals vs Leverages.
    c. [4 marks] Calculate the value $2p/n$ that identifies observations with "high leverage". How many times bigger (to 2dp) than $2p/n$ is the highest leverage value?

```
# Q17a
model.diag.metrics3 %>%
  top_n(5, wt = .hat)
```

```
## # A tibble: 5 x 8
##   index  medv .fitted .resid   .hat .sigma .cooksd .std.resid
##   <int> <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>      <dbl>
## 1    41  34.9    51.2 -16.3  0.0782   4.77  0.211       -3.53
## 2   162  50      61.8 -11.8  0.114    4.80  0.174       -2.60
## 3   163  50      59.1  -9.09 0.0814   4.81  0.0684      -1.96
## 4   366  27.5    13.8  13.7  0.0639   4.79  0.118        2.94
## 5   369  50      30.1  19.9  0.0701   4.74  0.276        4.28
```

```
# Q17b
plot(model3,5)
```



```
# Q17c
n <- length(y)
p <- ncol(X)
high.lev <- 2*p/n
high.lev
```

```
## [1] 0.01976285
```

```
max(model.diag.metrics3$.hat)/high.lev
```

```
## [1] 5.748462
```

17. c. The value of $2p/n$ that identifies observations with "high leverage" is $2 \times 5/n = 10/506 = 0.0198 \approx 0.02$. The highest leverage value is $\max(h_{ii}) = 0.1136$. Hence (to 2dp) the highest leverage value is 5.75 times bigger than $2p/n$. There are 52 observations with "high leverages", which is just over 10% of the data:

```
num.high.lev <- length(which(model.diag.metrics3$.hat > high.lev))
num.high.lev
```

```
## [1] 52
```

```
round(100*num.high.lev/n,2)
```

```
## [1] 10.28
```

Note that three of the five observations with the highest leverages have values of `medv` equal to the maximum (censored) value of 50.

18. [6 marks] Given your answers to Questions 15-17, discuss any features of the Boston data which are problematic for prediction using any of the linear regression models fitted in this assignment. Despite those issues, which of the six models fitted in Question 2 do you recommend as 'best', and why?

It is clear from the earlier answers that many of the 'most problematic' observations in the Boston data are observations which have values of the response variable `medv` equal to the maximum (censored) value of 50. It is those 16 censored values which produce the points that lie on a straight line within some of the residual diagnostic plots. It is also a subset of those censored points that have the highest leverage values and the largest (absolute) standardised residuals, or the appearance of outliers. For model3, only four predicted values were larger than the censored value of 50, but such predictions are problematic too – since such values are apparently 'impossible'. One option could be to remove those 16 observations and refit models to the uncensored data – but a search for a 'good' subset of explanatory variables could also consider several of the other possible predictors that are available within the full dataset.

Of the models considered in Question 2, model5 seems the best. It was chosen by all three model selection criteria, and was also selected in preference to model6 when using sequential F-testing (or equivalently, standard t-tests of the model coefficients). The standard diagnostic plots for model5 are given below, and the noted problems due to the censored observations are clearly still apparent,

Note there is a cluster of eight observations around index 370, which have 8 of the 10 highest Cook's distances. Understanding why those observations are so influential may be useful.

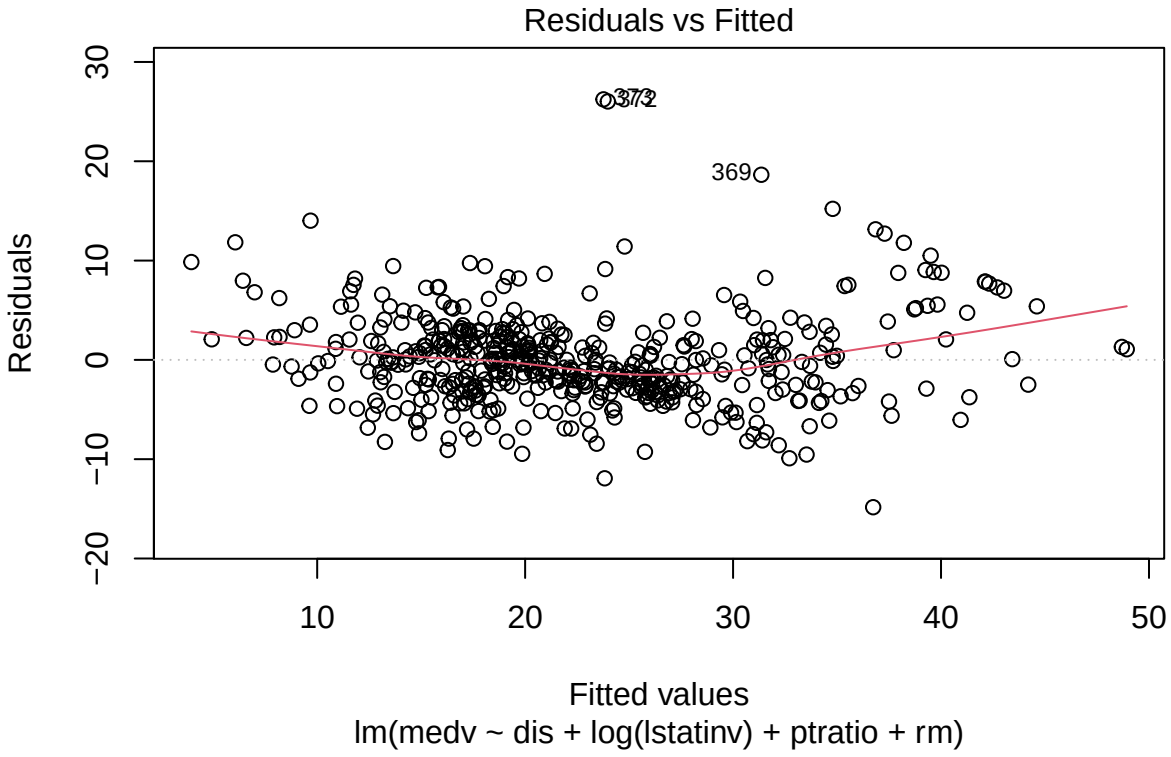
```
model.diag.metrics5 <- augment(model5)

model.diag.metrics5 <- model.diag.metrics5 %>%
  mutate(index = 1:nrow(model.diag.metrics5)) %>%
  select(index, medv, .fitted, .resid, .hat, .sigma, .cooksd, .std.resid)


model.diag.metrics5 %>%
  top_n(10, wt = .cooksd)
```

```
## # A tibble: 10 x 8
##    index  medv .fitted .resid    .hat .sigma .cooksd .std.resid
##    <int> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>      <dbl>
##  1   226  50      39.5  10.5  0.0301    4.57  0.0335       2.32
##  2   268  50      37.3  12.7  0.0340    4.56  0.0558       2.82
##  3   365  21.9    36.7 -14.8  0.0405    4.55  0.0918      -3.30
##  4   366  27.5    19.2   8.33 0.0830    4.58  0.0649       1.89
##  5   368  23.1    13.7   9.45 0.0452    4.58  0.0419       2.10
##  6   369  50      31.4  18.6  0.0686    4.52  0.260        4.20
##  7   370  50      34.8  15.2  0.0256    4.55  0.0590       3.35
##  8   371  50      38.2  11.8  0.0315    4.57  0.0442       2.61
##  9   372  50      24.0  26.0  0.00877   4.45  0.0573       5.69
## 10   373  50      23.8  26.2  0.0123    4.44  0.0820       5.75
```

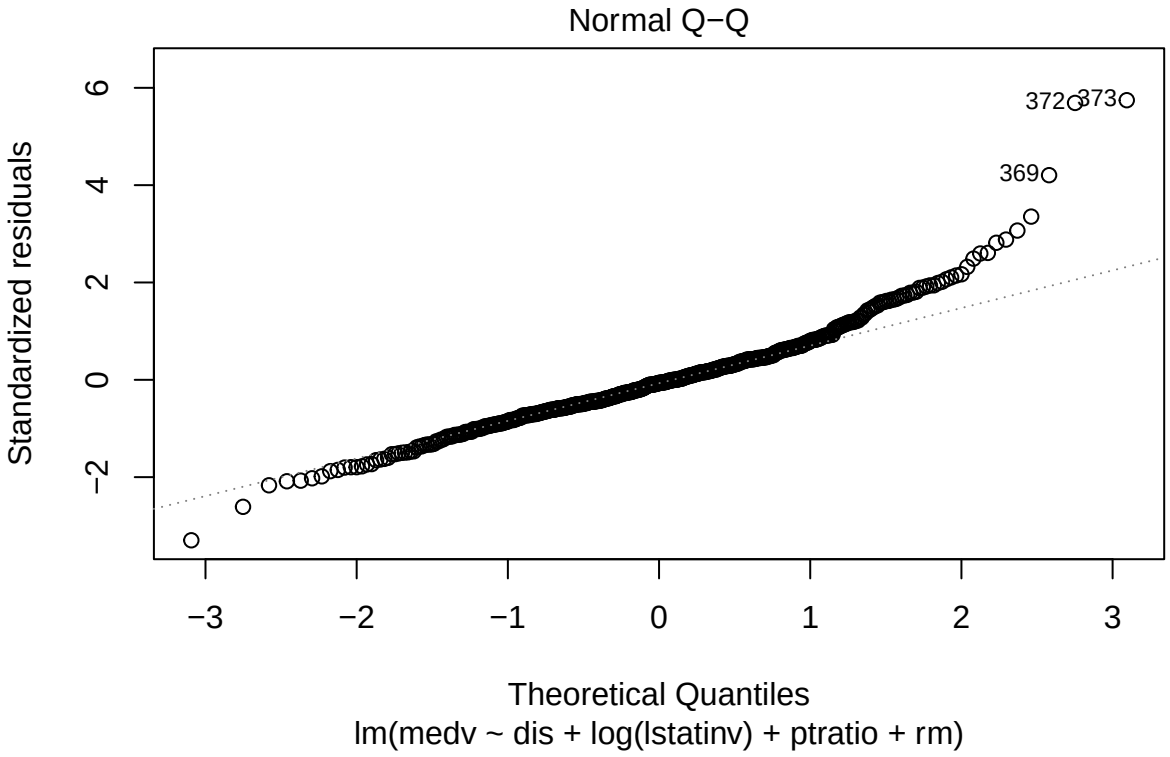
```
Boston[365:373,c(6,8,11,13,14)]
```

```
##        rm    dis ptratio lstat medv
## 365 8.780 1.9047    20.2  5.29 21.9
## 366 3.561 1.6132    20.2  7.12 27.5
## 367 4.963 1.7523    20.2 14.00 21.9
## 368 3.863 1.5106    20.2 13.33 23.1
## 369 4.970 1.3325    20.2  3.26 50.0
## 370 6.683 1.3567    20.2  3.73 50.0
## 371 7.016 1.2024    20.2  2.96 50.0
## 372 6.216 1.1691    20.2  9.53 50.0
## 373 5.875 1.1296    20.2  8.88 50.0
```

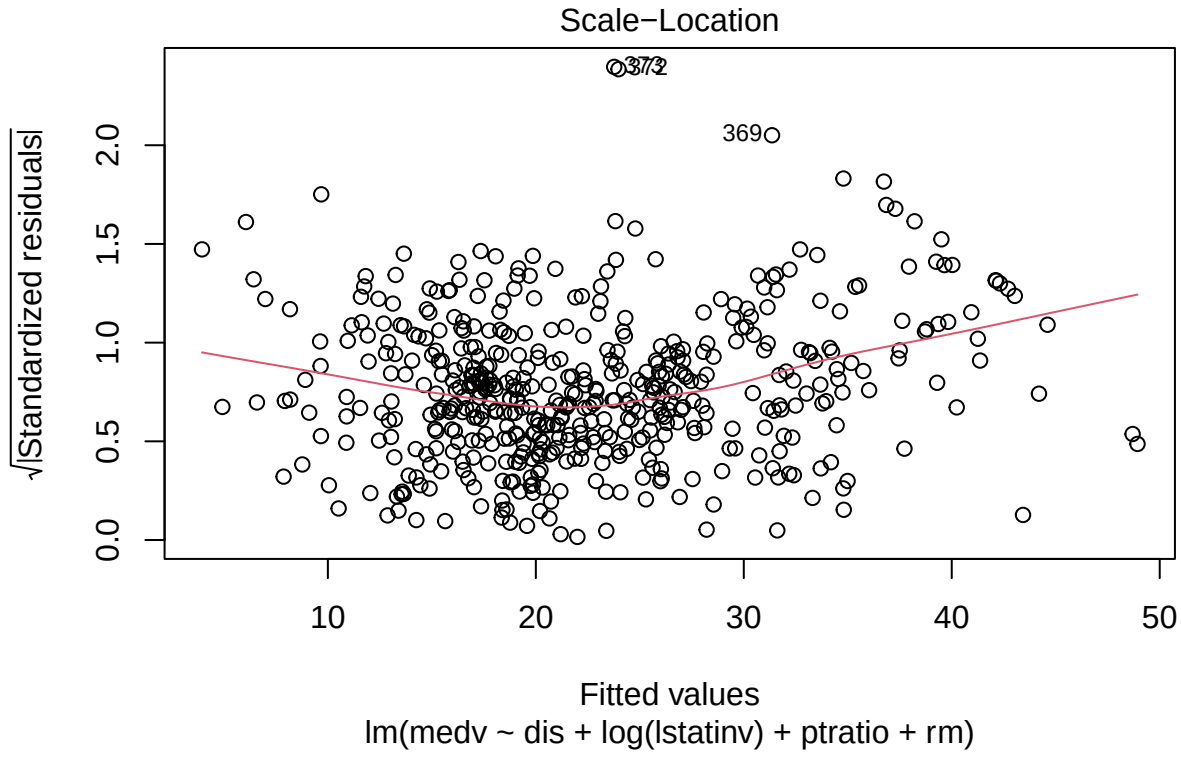
```
plot(model5, 1)
```



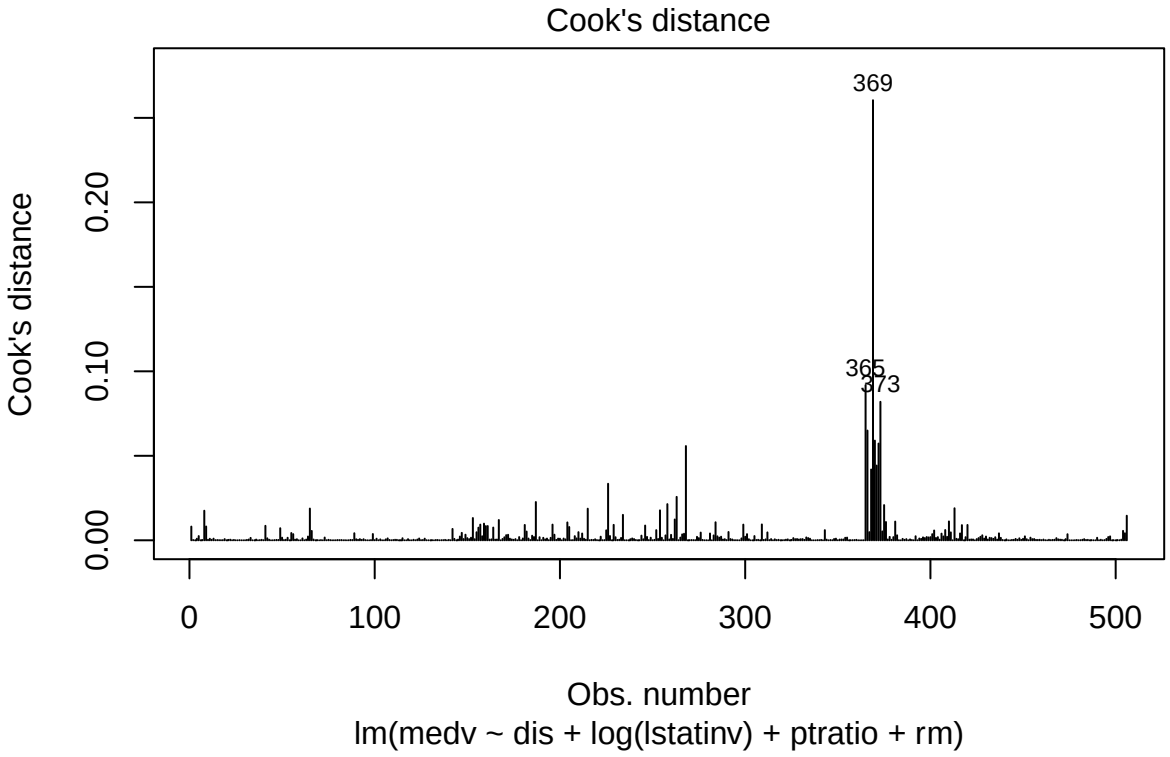
```r
plot(model5, 2)
```




```r
plot(model5, 3)
```



```
plot(model5, 4)
```




```
plot(model5, 5)
```