Intro to ML Final Exam

Abhinav Sharma

1/8/2021

Setup - Loading libraries and setting working directory. Key libraries include ggplot2, plotly, Hmisc, corrgram and DescTools for exploration; data.table, dplyr, broom for data wrangling; rpart, gbm, nnet, caret, kknn, glmnet, radiant.model and randomForest for modelling and ROCR, precrec for performance metrics and assessment.

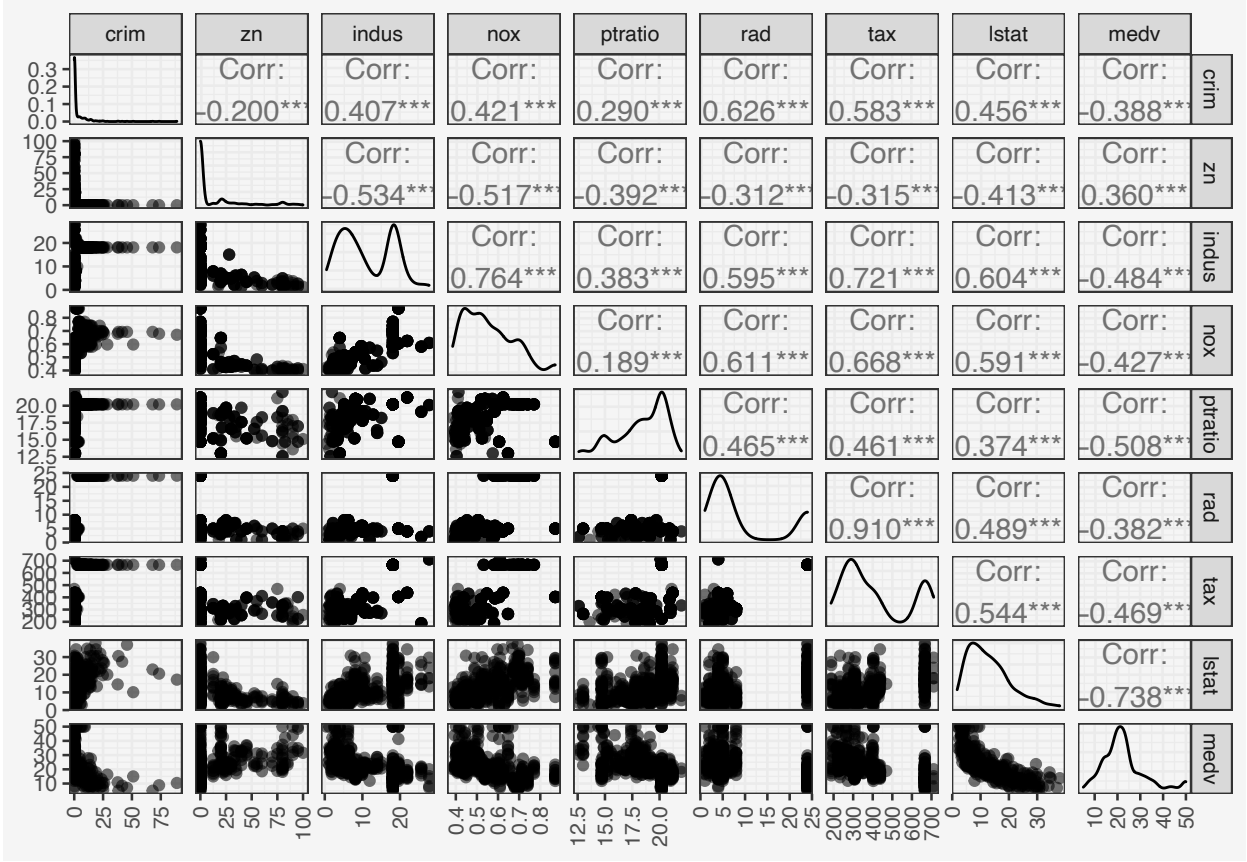
Setting up custom ggplot theme for all plotting purposes.

Chapter #2 : Ques 10

Part (a) The data represents housing value in suburbs of Boston. Each row is one suburb and each column represents a particular characterstic associated with the particular suburb such as crime rate, taxation, accessibility to highways etc. The last column medv is median value of owner-occupied homes in units of $1000s.

```
Number of rows and columns are :  506  and  14  respectively.
```

Part (b) Exploring bivariate relationships and correlations :



We observe accessibility to radial highway (rad) and taxation (tax) have almost similar distribution with a very high positive correlation. Both these variables have some correlation with crime rate (crim) as well. The crime rate (crim) and proportion of residential zone (zn) variables are highly right skewed whereas proportion of non-retail business (indus) is bi-modal.

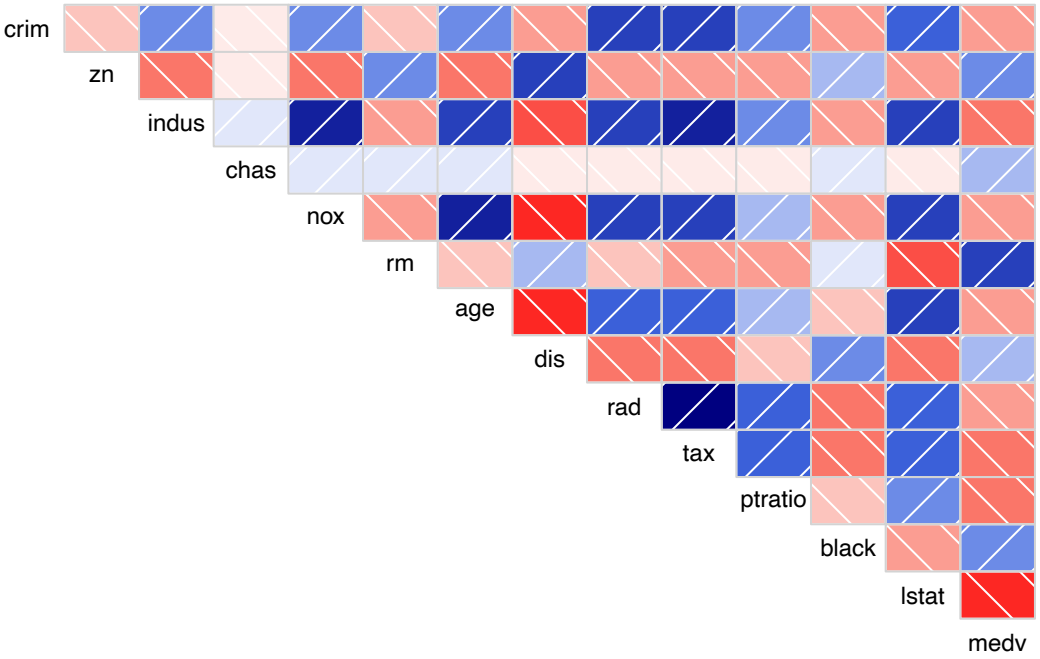
The scatterplots help us understand relationship of individual variables with median house value (medv) with it having inverse association with crime rate, NOx concentrations (nox), taxation (tax), percent of lower status population (lstat) whereas direct relatioship with proportion of residential zone (zn)

Similarly, we see tax and indus positively correlated, nox and indus positively correlated, medv and lstat negatively correlated with absolute value of correlation greater than 0.7

Part (c) We revisit correlations using corrgram function

```
   crim      zn   indus    chas     nox      rm     age     dis     rad     tax
   1.00   -0.20    0.41   -0.06    0.42   -0.22    0.35   -0.38    0.63    0.58
ptratio   black   lstat    medv
   0.29   -0.39    0.46   -0.39
```

```
Visualize correlation
```

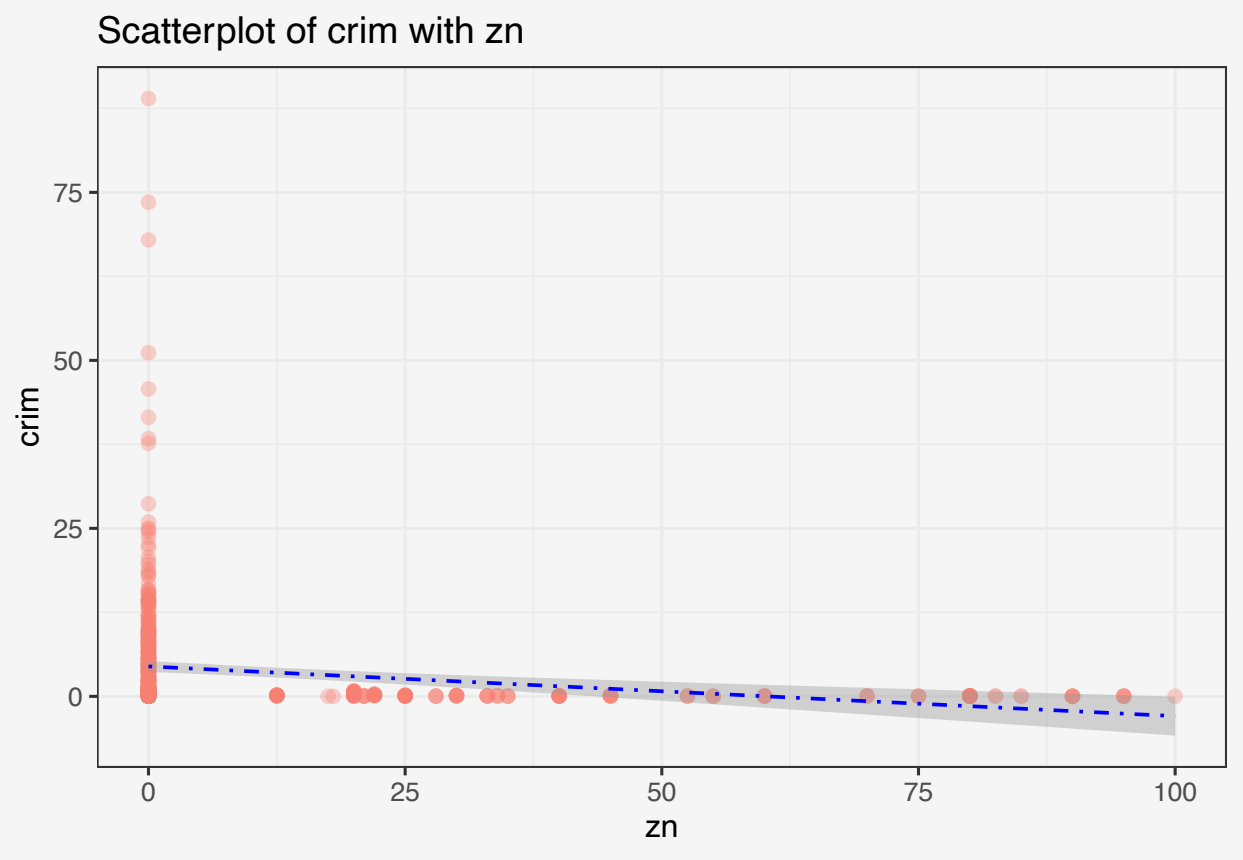


Above plot shows whether a pair of variable has direct or inverse association. The color of the tile indicated strength of association (correlation)

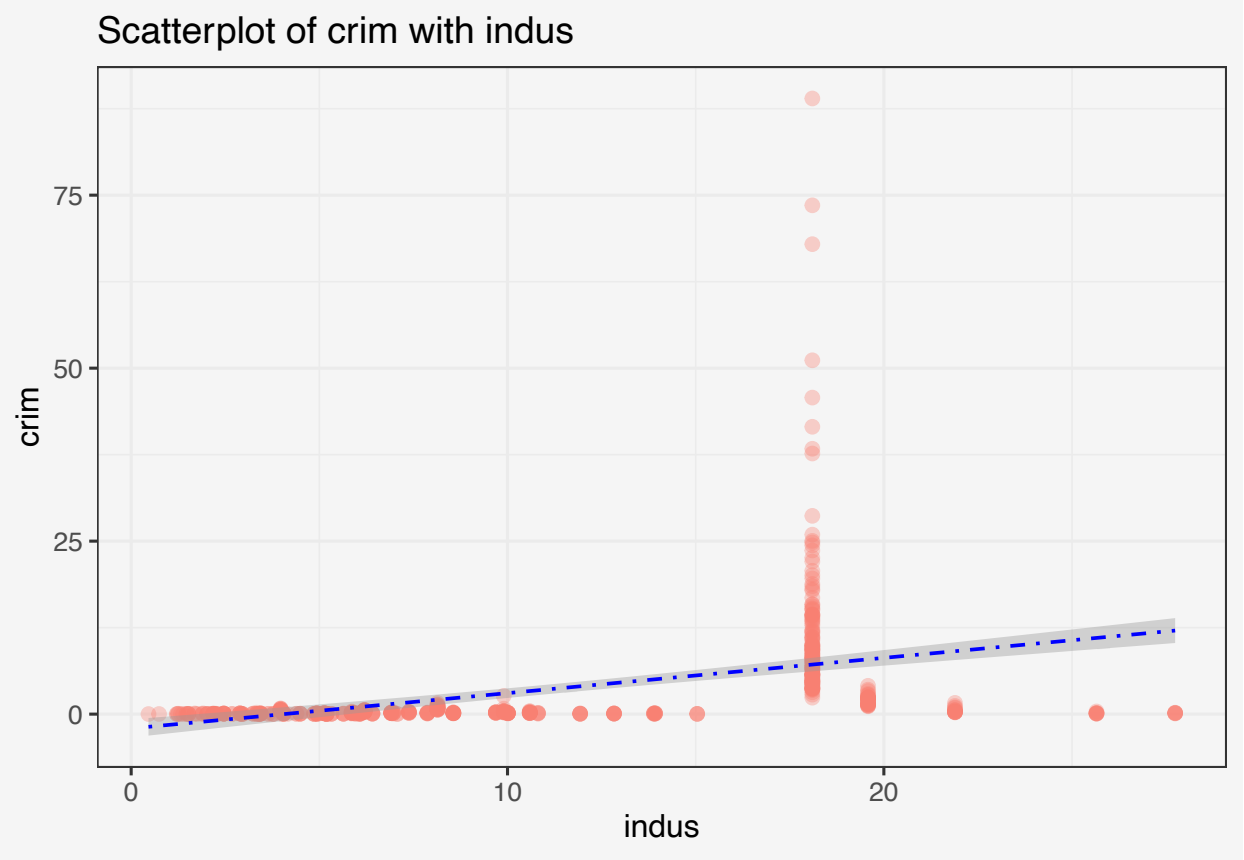
Observing the top row of the plot we see, crim has strong positive correlation with rad, tax and lstat, moderate positive correlation with indus, nox, age, ptratio. We observe moderate negative association with the variables zn, rm, dis, black and medv.

Scatterplots showing relationship with crim

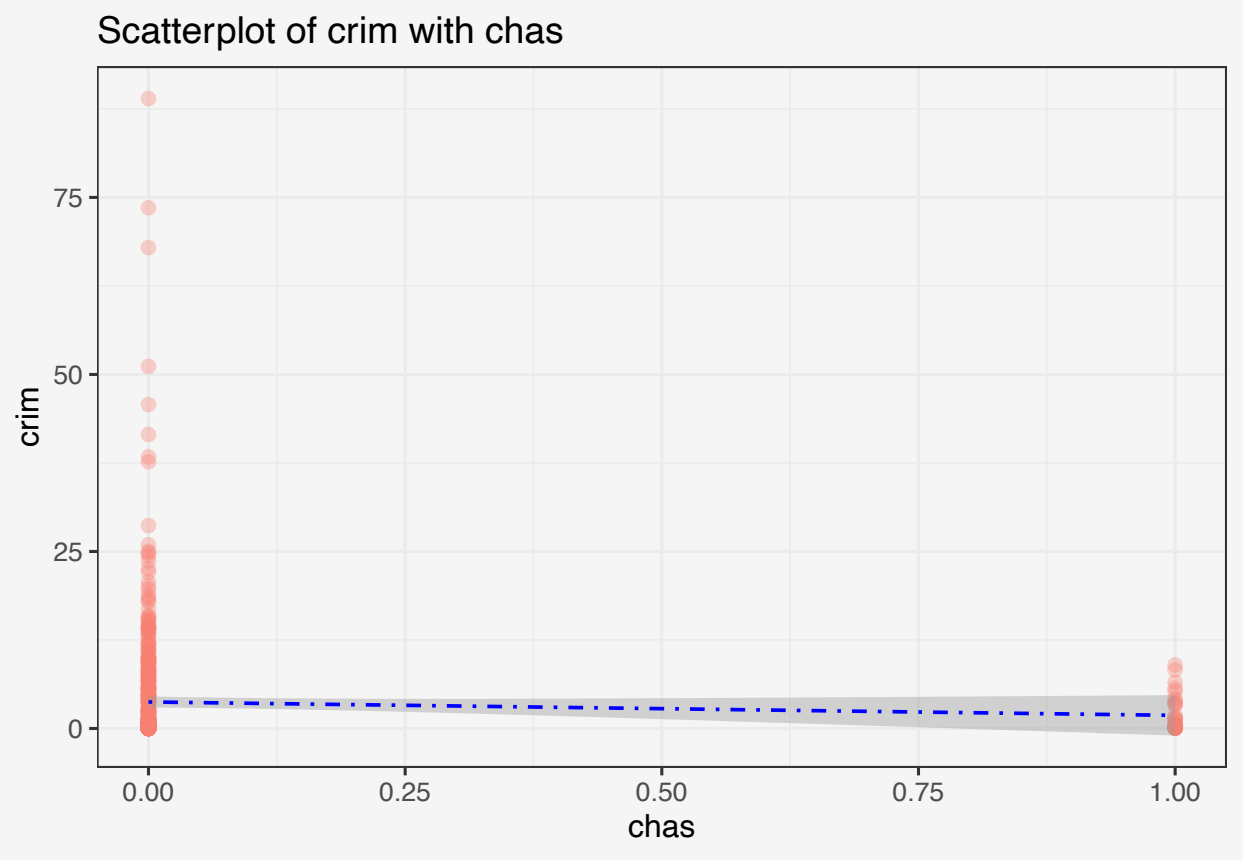
```
‘geom_smooth()‘ using formula ’y ~ x’
```



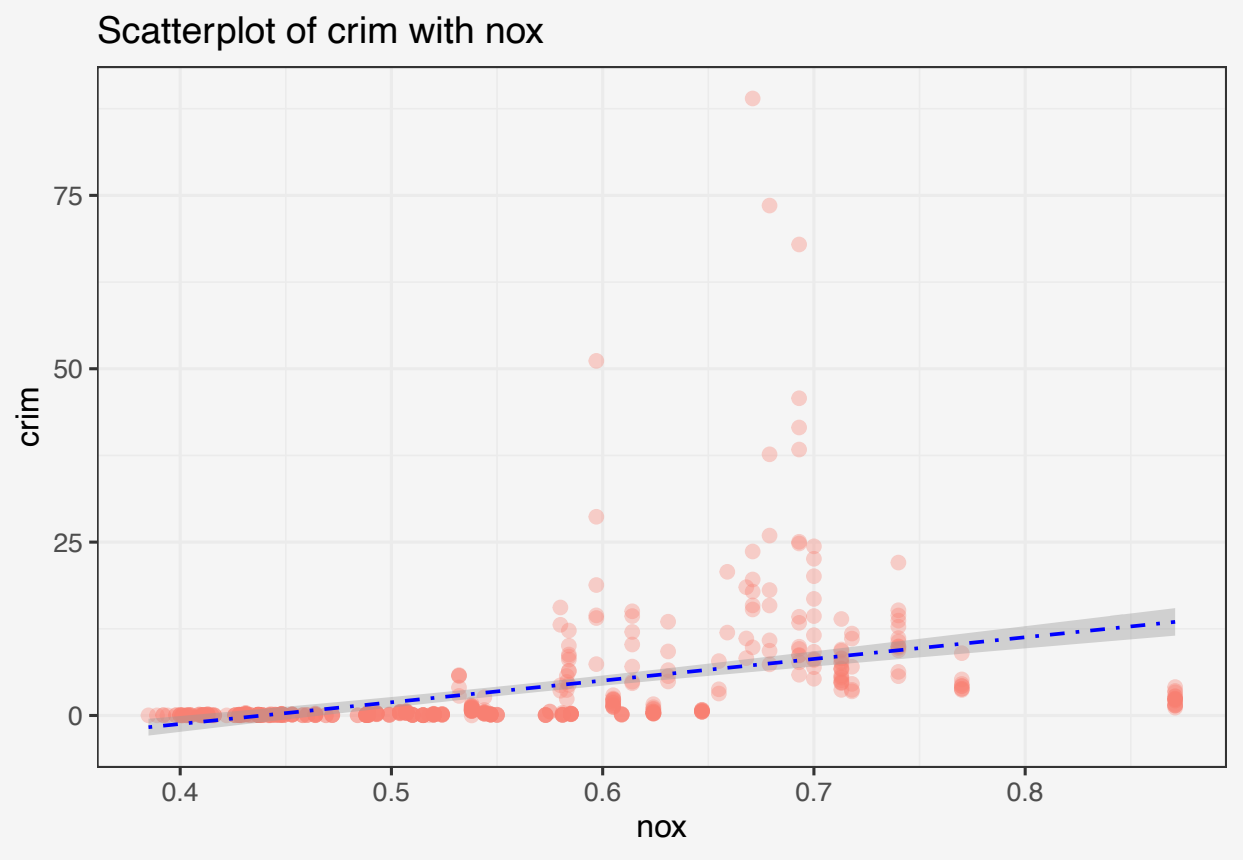

```
`geom_smooth()` using formula 'y ~ x'
```



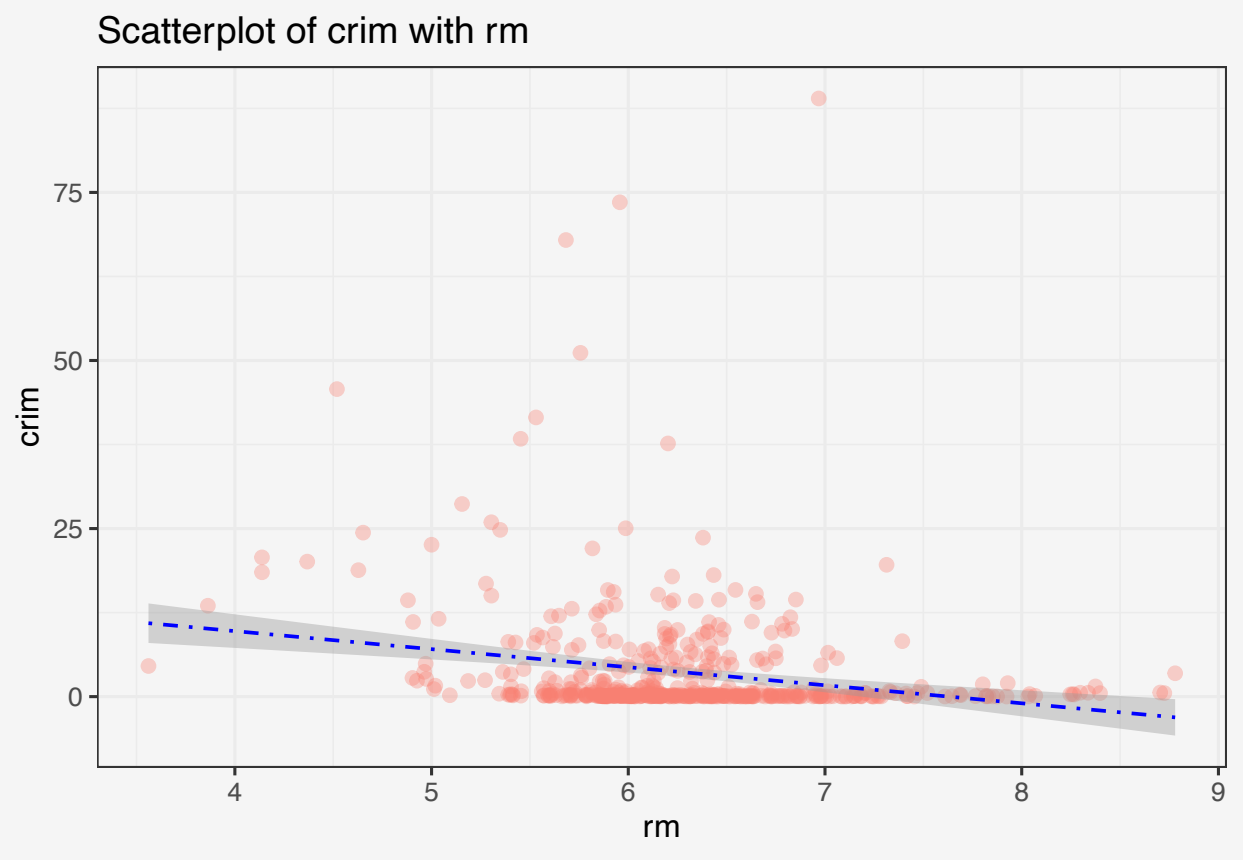

```
`geom_smooth()` using formula 'y ~ x'
```



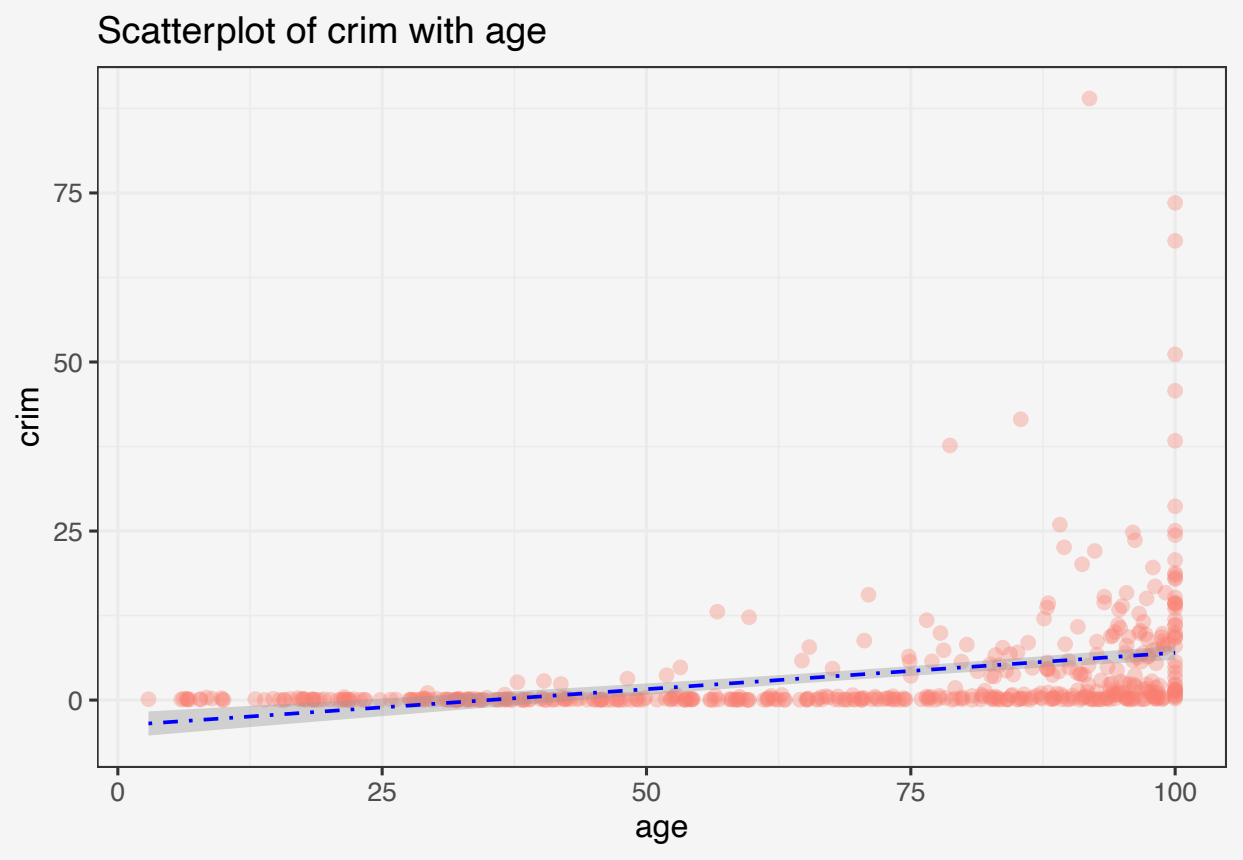

```
`geom_smooth()` using formula 'y ~ x'
```



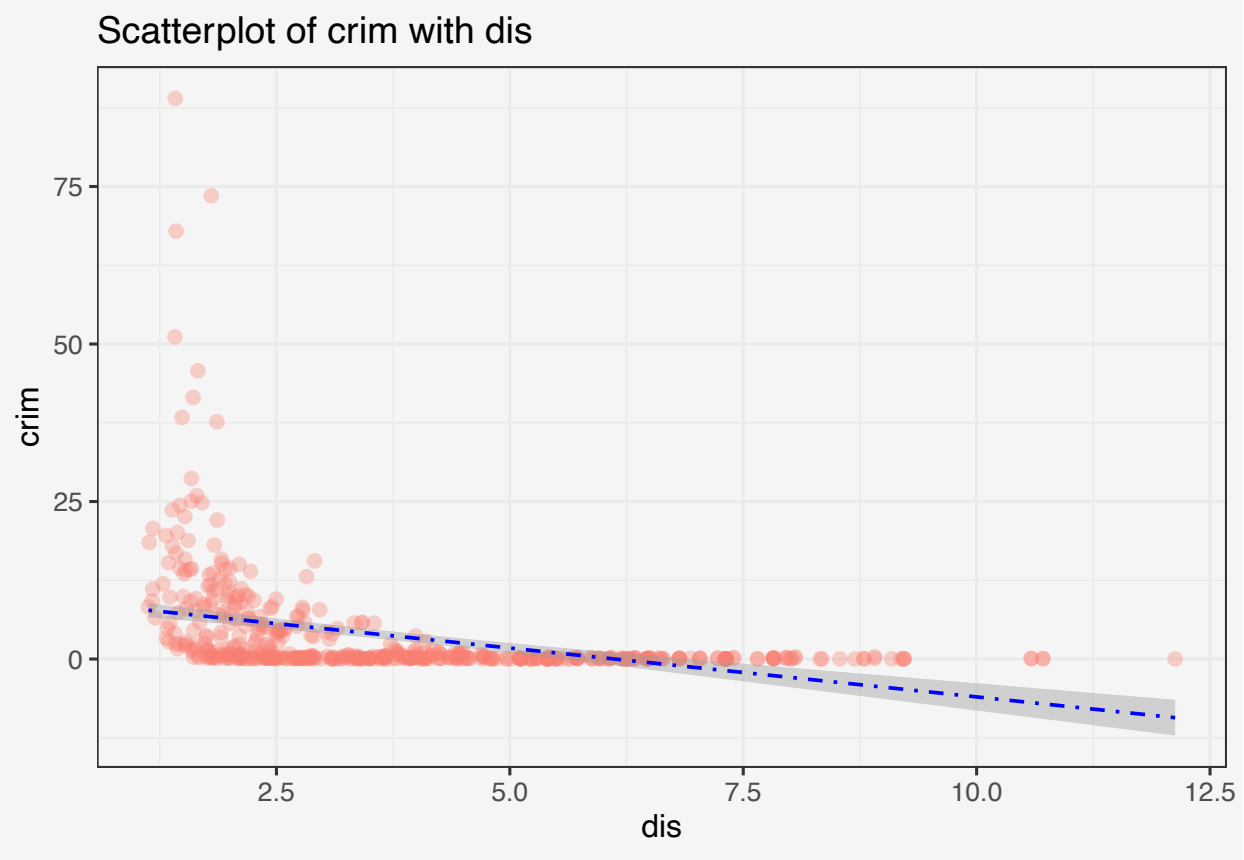

```
`geom_smooth()` using formula 'y ~ x'
```



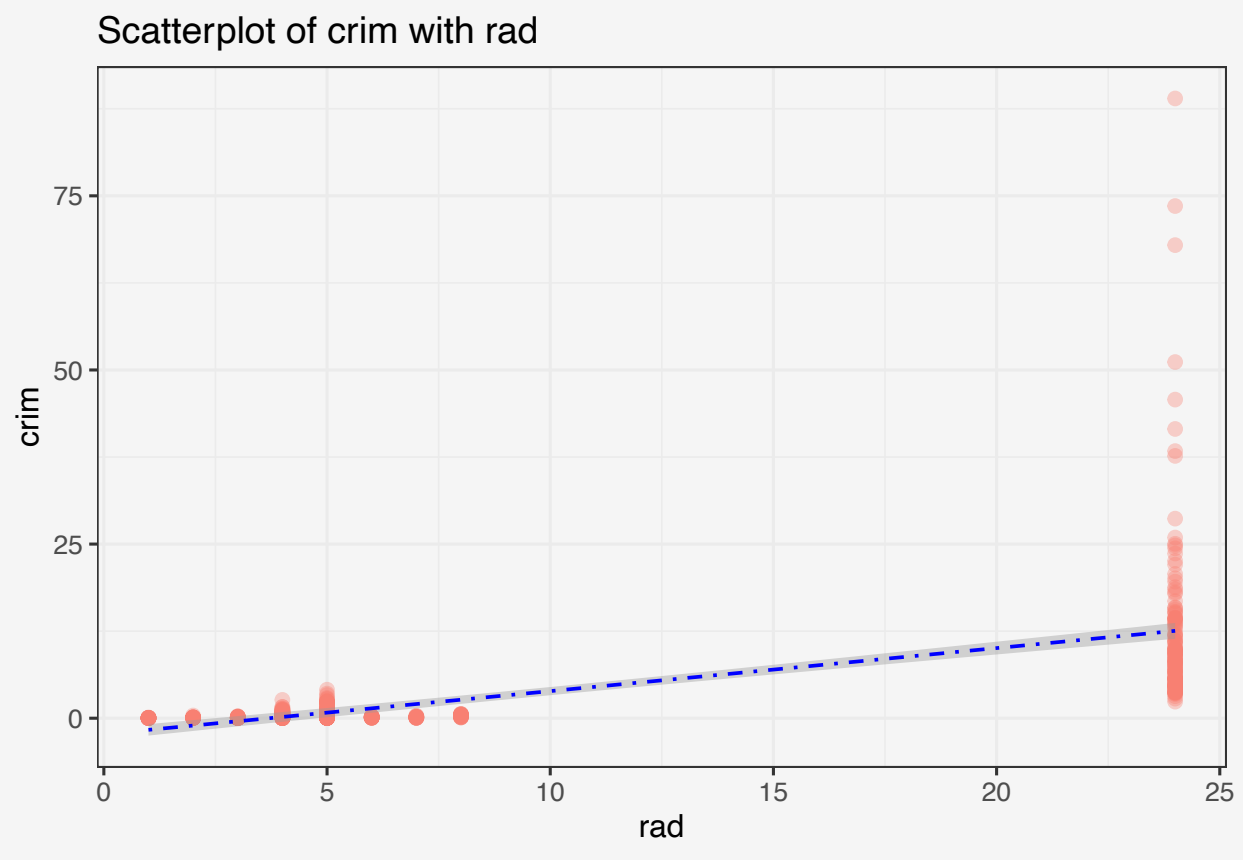

```
`geom_smooth()` using formula 'y ~ x'
```



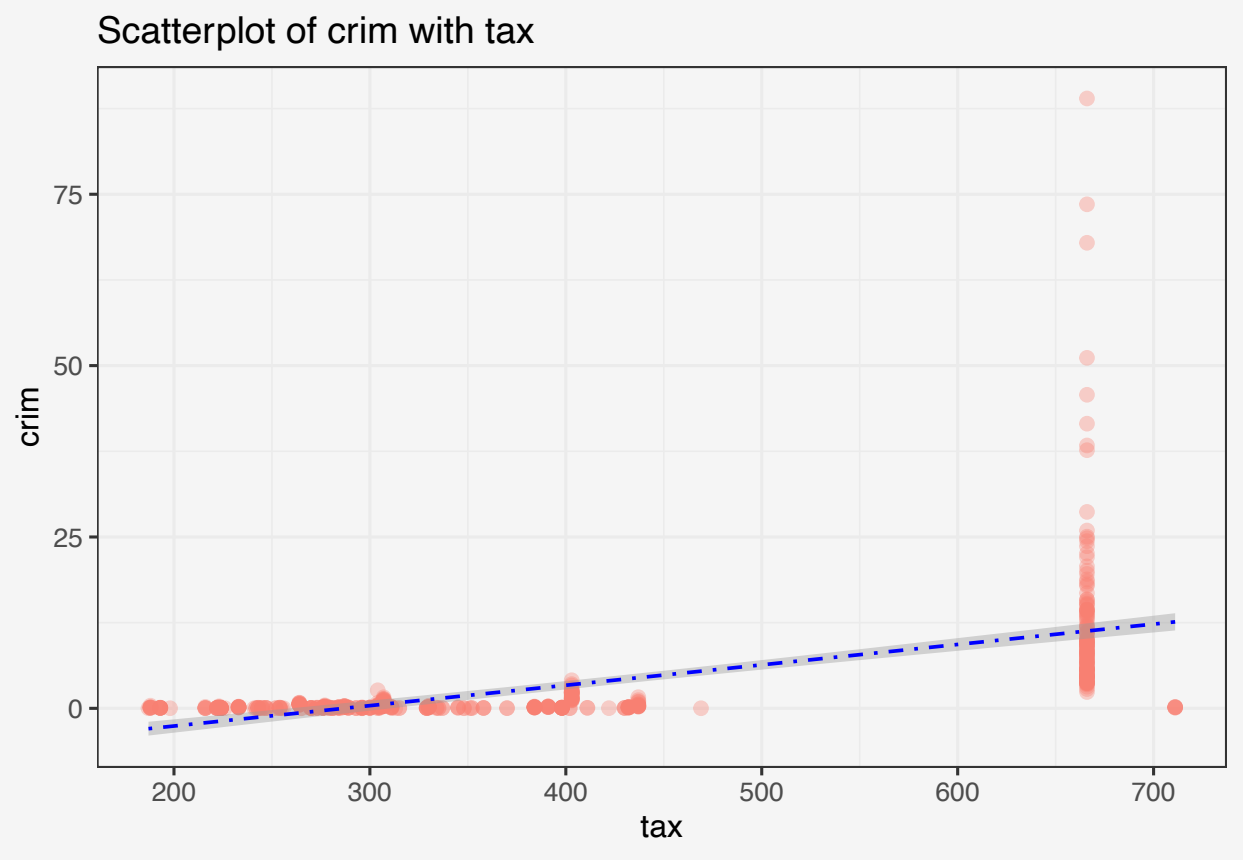

```
`geom_smooth()` using formula 'y ~ x'
```



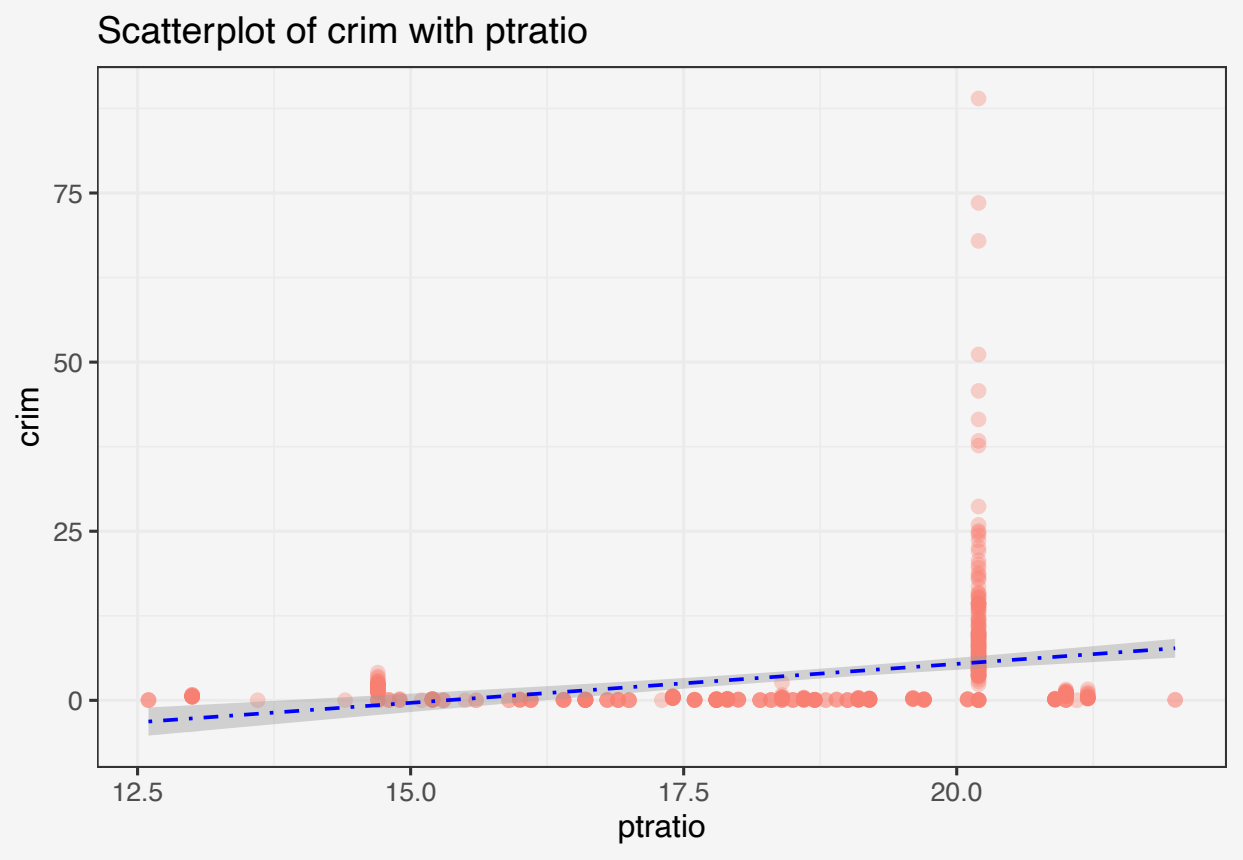

```
`geom_smooth()` using formula 'y ~ x'
```



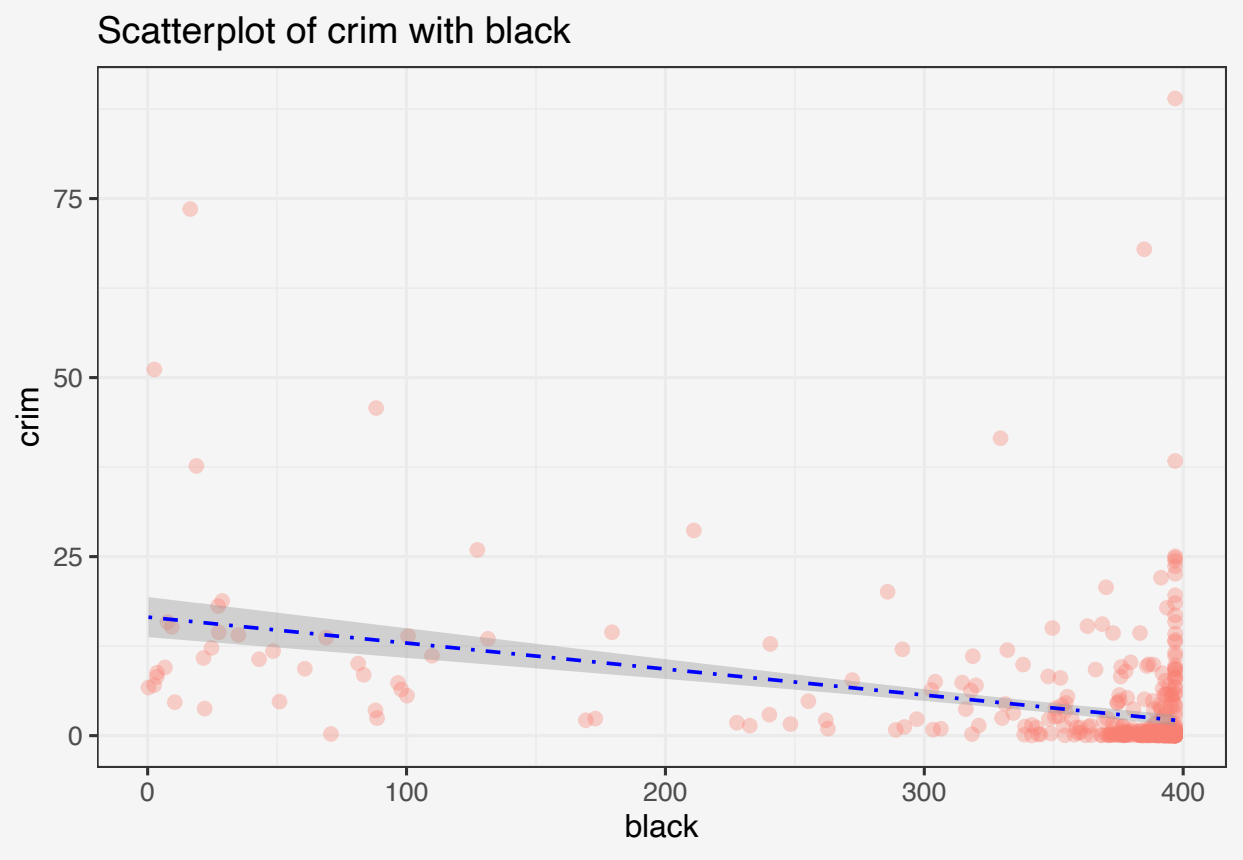

```
`geom_smooth()` using formula 'y ~ x'
```



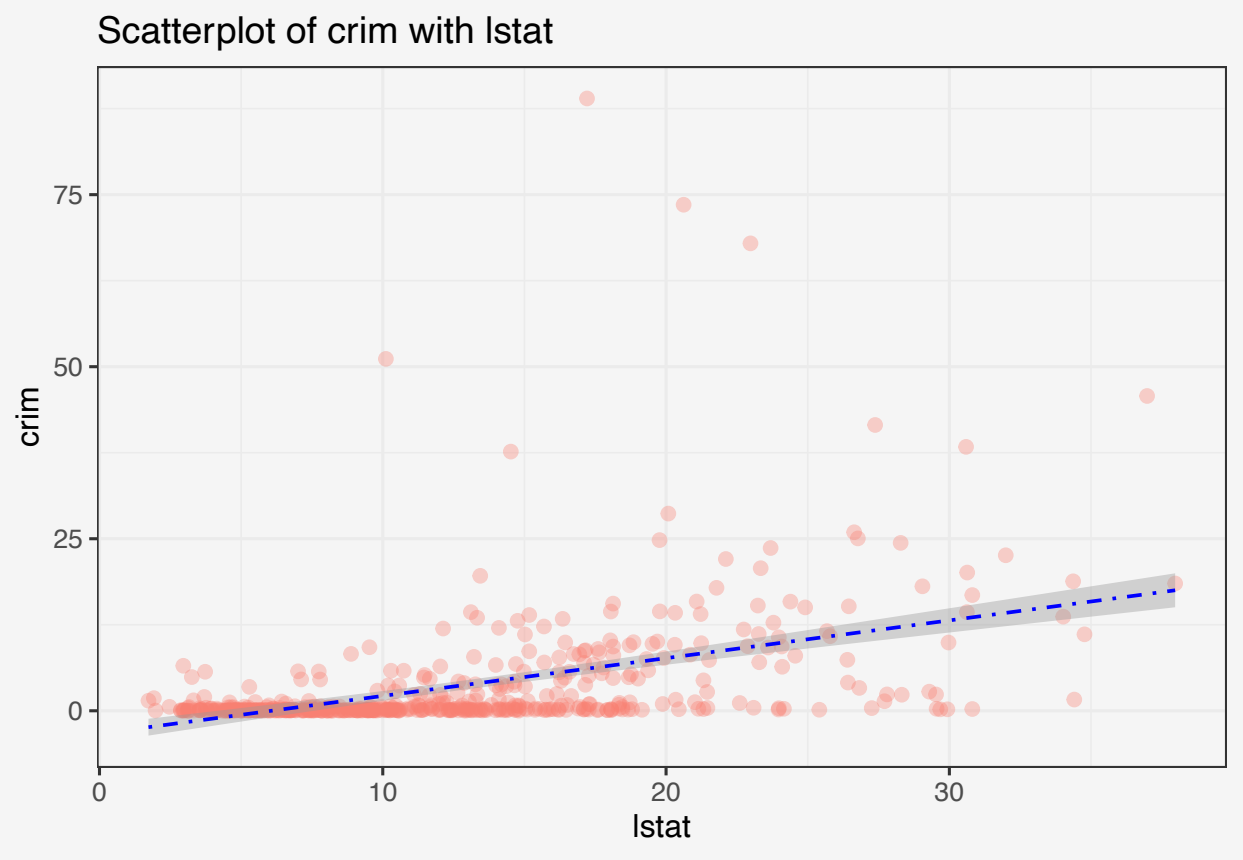

```
`geom_smooth()` using formula 'y ~ x'
```



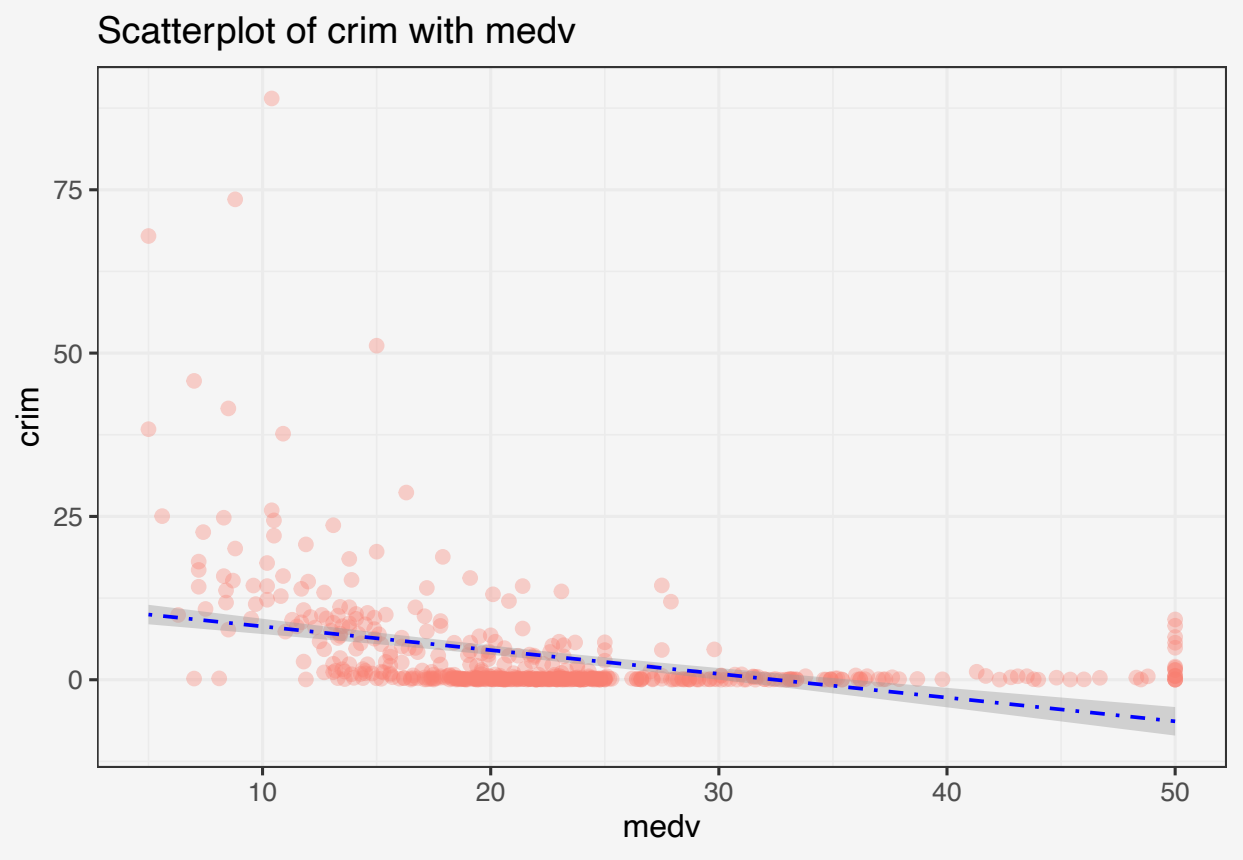

```
`geom_smooth()` using formula 'y ~ x'
```




```
`geom_smooth()` using formula 'y ~ x'
```



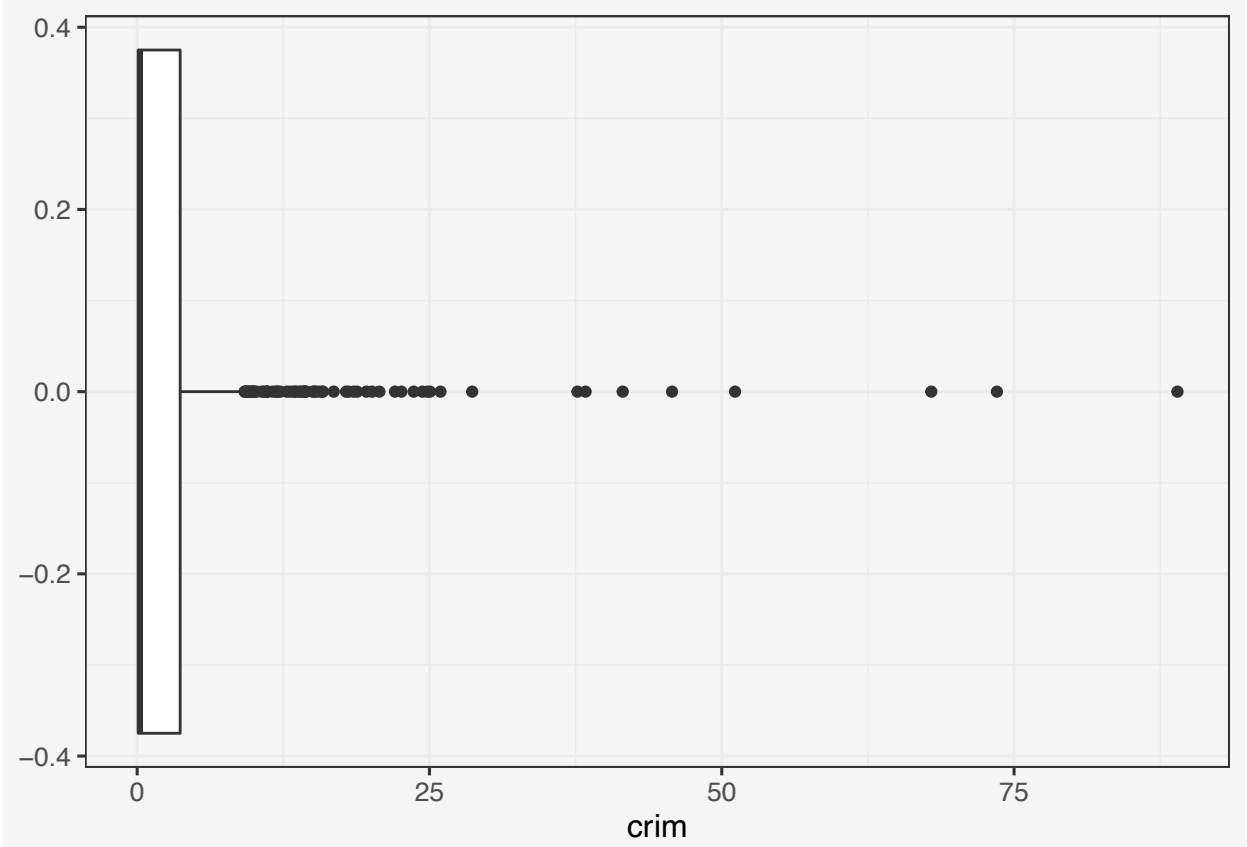

```
`geom_smooth()` using formula 'y ~ x'
```

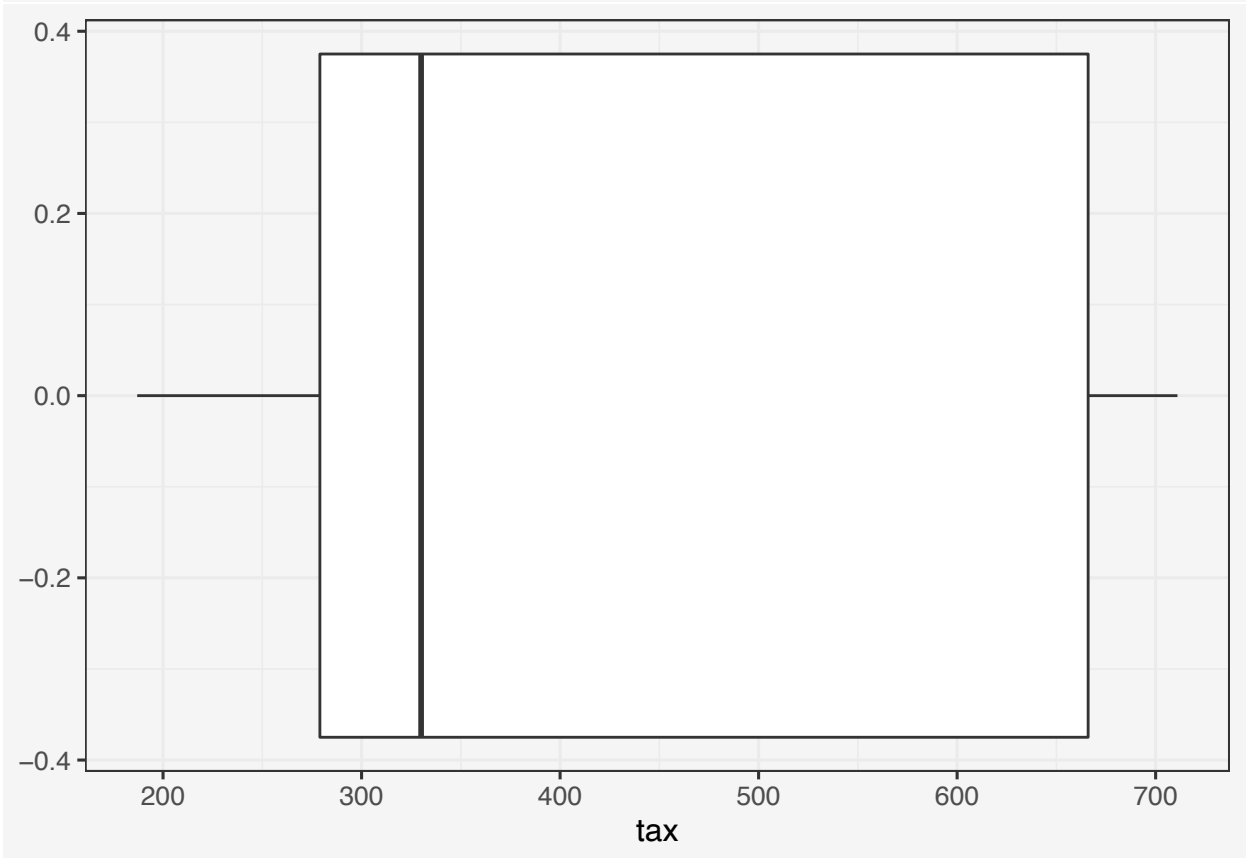


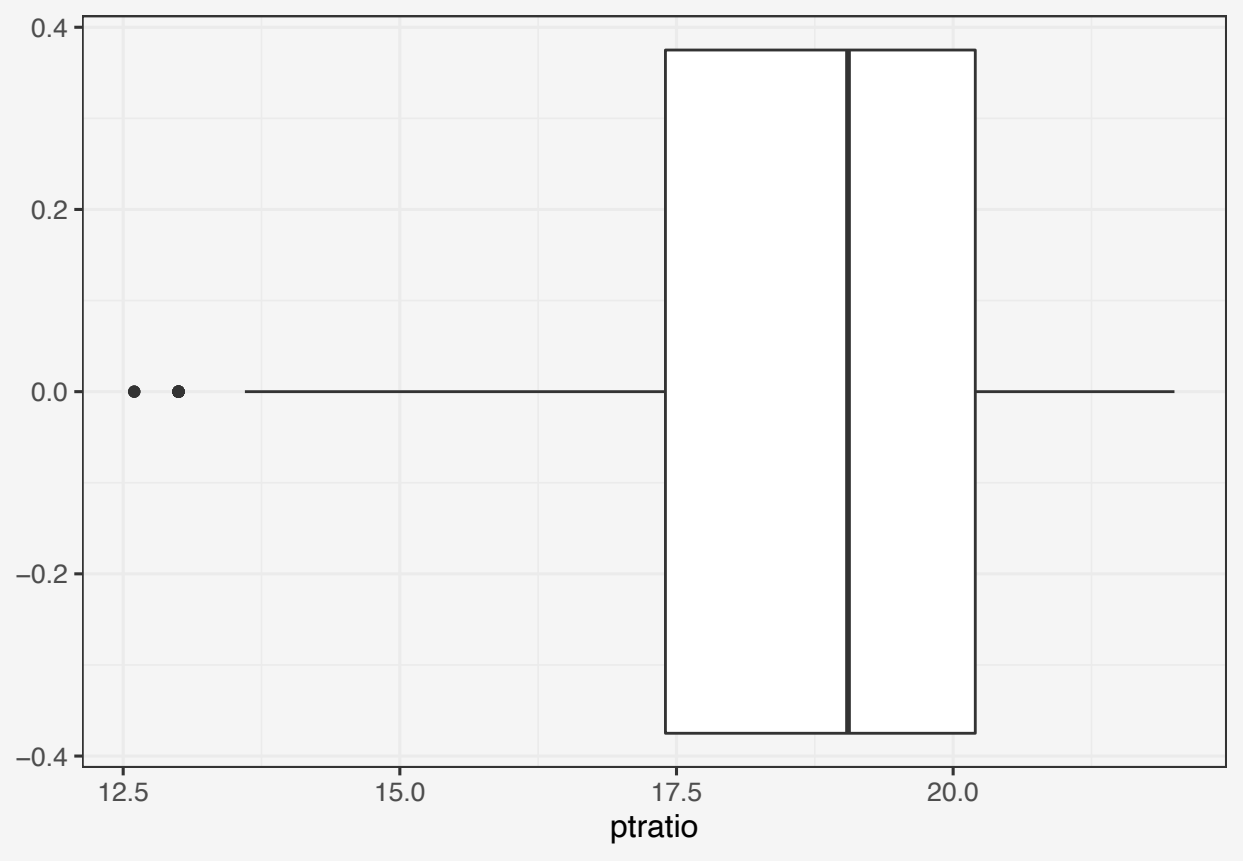
Part (d) Variable ranges

```
Range of crime rate in Boston suburbs :  0.00632 88.9762

Range of tax rate in Boston suburbs :  187 711

Range of ptratio in Boston suburbs :  12.6 22
```

Detecting outliers in variables


0.4
0.2
0.0
−0.2
−0.4
0
25
50
75
crim


0.4
0.2
0.0
−0.2
−0.4
200
300
400
500
600
700
tax



We note ranges of the relevant predictors above. Crim shows outliers with higher than (mean+2sd) while ptratio shows outliers with lower values lower than (mean-2sd) No suburbs have particularly high values. No outliers for tax variable.

While variable ptratio shows some left skew, crim shows very strong right skew indicating we might need some outlier treatment on crim as high values such as 88.97 may impact coefficients of regression and impact interpretation.

Part (e) Number of suburbs bound

```
A total of 35  suburbs (roughly  6.92 %) are bounded by the Charles river.
```

Part (f) Median ptratio

```
Median pupil-teacher ratio in the dataset is : 19.05 .
```

Part (g) Median values by variable

```
        ColMedian Range_l Range_u
crim         0.26    0.01   88.98
zn           0.00    0.00  100.00
indus        9.69    0.46   27.74
chas         0.00    0.00    1.00
nox          0.54    0.38    0.87
```

```
rm          6.21    3.56    8.78
age        77.50    2.90  100.00
dis         3.21    1.13   12.13
rad         5.00    1.00   24.00
tax       330.00  187.00  711.00
ptratio    19.05   12.60   22.00
black     391.44    0.32  396.90
lstat      11.36    1.73   37.97
medv       21.20    5.00   50.00
```

Suburbs with minimum owner-occupied units

```
      crim zn indus chas   nox    rm age    dis rad tax ptratio  black lstat
399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.90 30.59
406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 384.97 22.98
    medv
399    5
406    5
```

The suburbs in row number 399 and 406 have the lowest 'median value' of $5000. We see, compared to dataset medians these suburbs have : 1. Very high crime rate (crim) 2. Higher indus 3. Same median chas 4. Higher NOx concentration (nox) 5. Lesser #rooms (rm) 6. Higher proportion of owner-occupied units built prior to 1940 7. Lower weighted mean of distances to five Boston employment centres 8. Higher accessibility from radial highways 9. Higher taxes 10. Higher pupil-teacher ratio by town 11. Typical proportion of blacks by town 12. Higher percentage of population belonging in lower status category

Based on variable distributions and range, the crime rate, lstat and tax rates in these low median value suburbs are at the higher end of their respective distribution.

Part (h) Suburbs with high #dwellings

```
#Suburbs averaging more than 7 dwellings :  64 .

#Suburbs averaging more than 8 dwellings :  13 .
```

```
        ColMean
crim       0.72
zn        13.62
indus      7.08
chas       0.15
nox        0.54
rm         8.35
age       71.54
dis        3.43
rad        7.46
tax      325.08
ptratio   16.36
black    385.21
lstat      4.31
medv      44.20
```

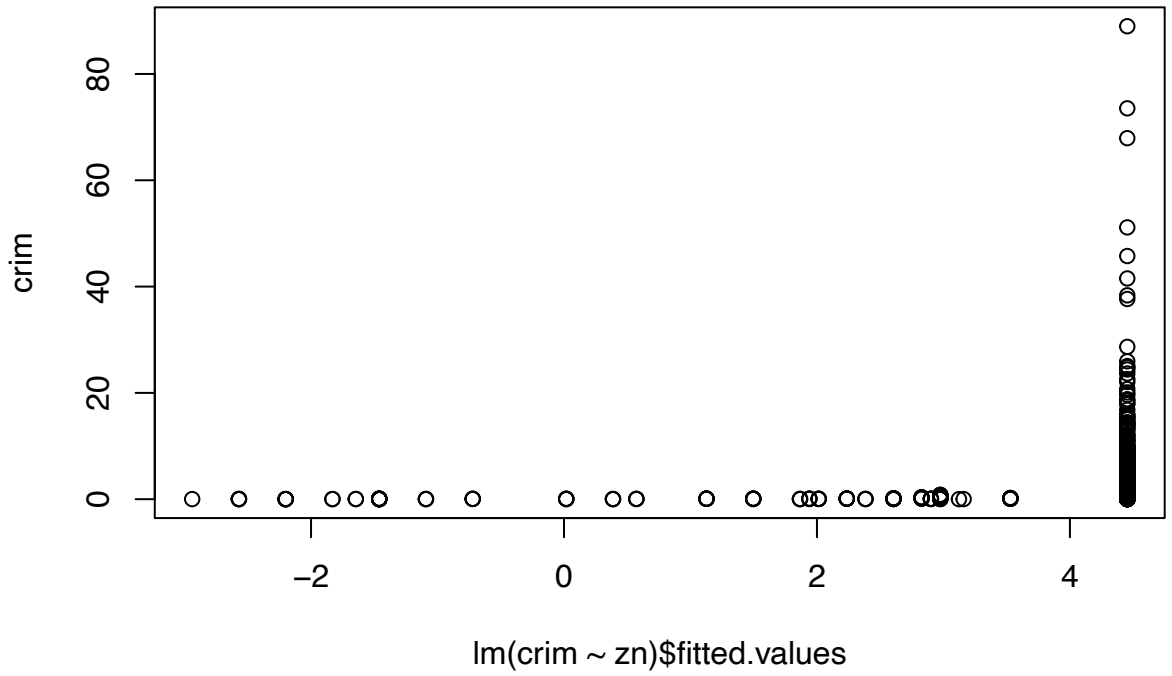
Suburbs with more than eight dwellings on average have very low crime rate (crim) and high proportion of residential land zones (zn) accompanied with better pupil-teacher ratios (ptratio) and lesser percent of lower status population (lstat) Medv is higher than median house values.

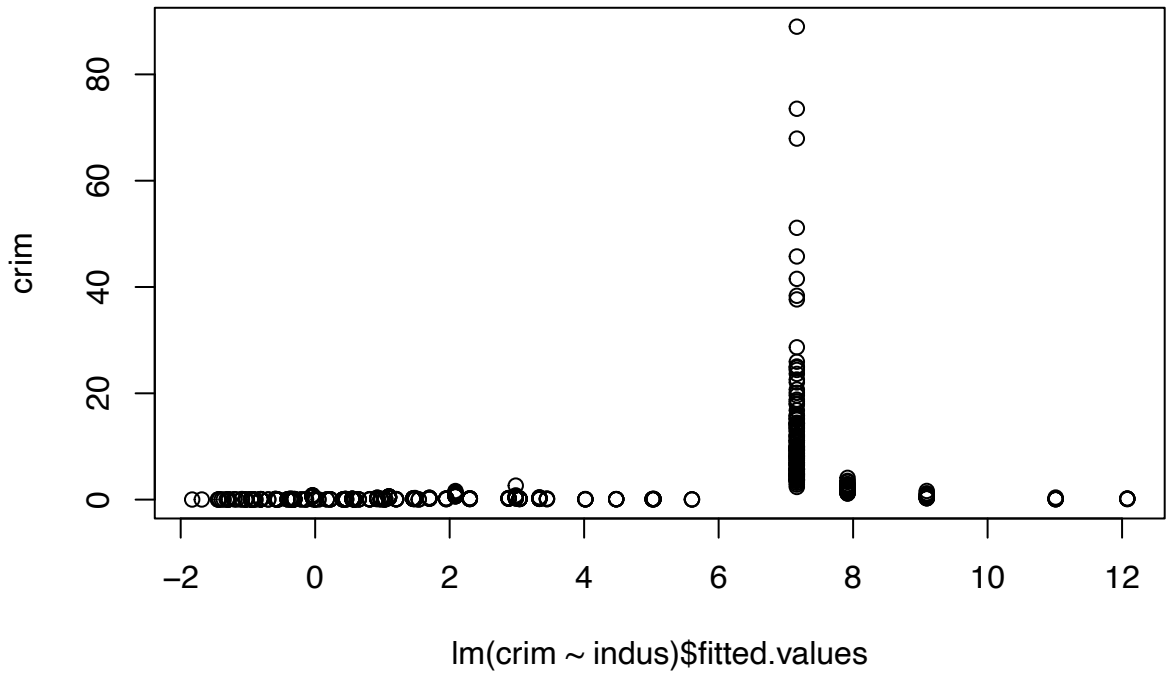
Chapter #3 : Ques 15

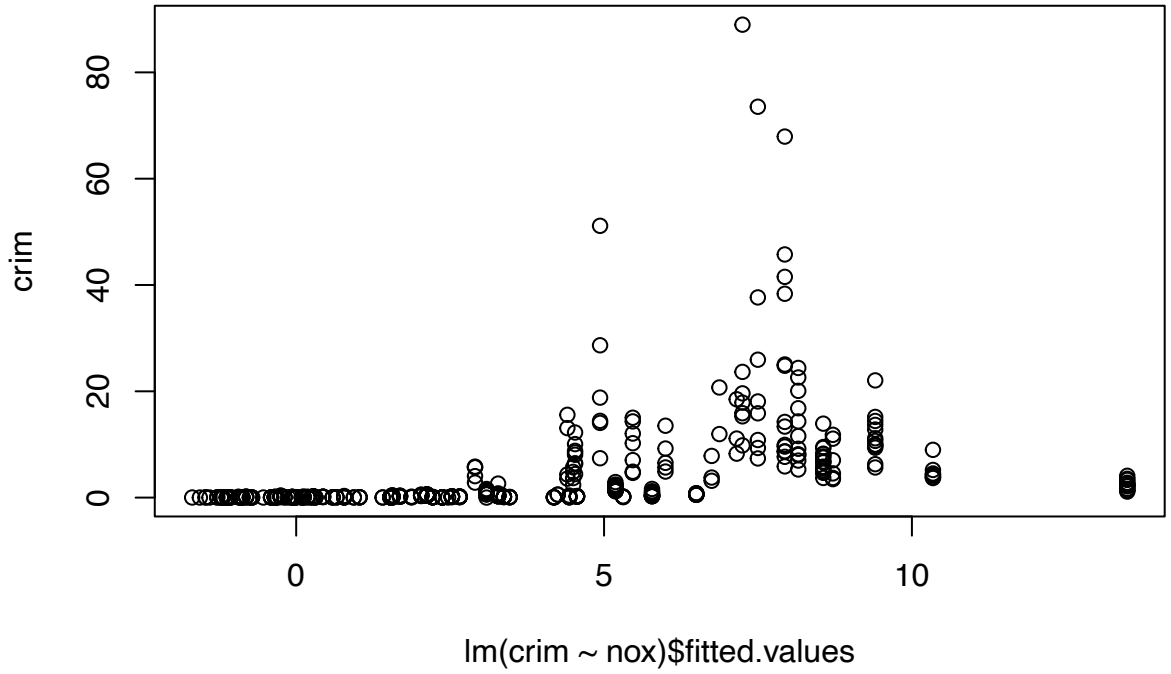
Part (a) We run linear regression by individual columns and save the summary matrix in an object. The below data frame collates the co-efficients, SEs, p-values and estimate confidence intervals for easier readability. Post that we also plot the scatterplot of crim with each of the co-variates and observe their relationship trend under linearity assumption. We also observe respective CIs. The idea is if CI includes the slope = 0 (line parallel to x-axis) we aren't sure of particular variable's coefficient not being zero, thereby not being able to confirm any association between variable and crime rate.

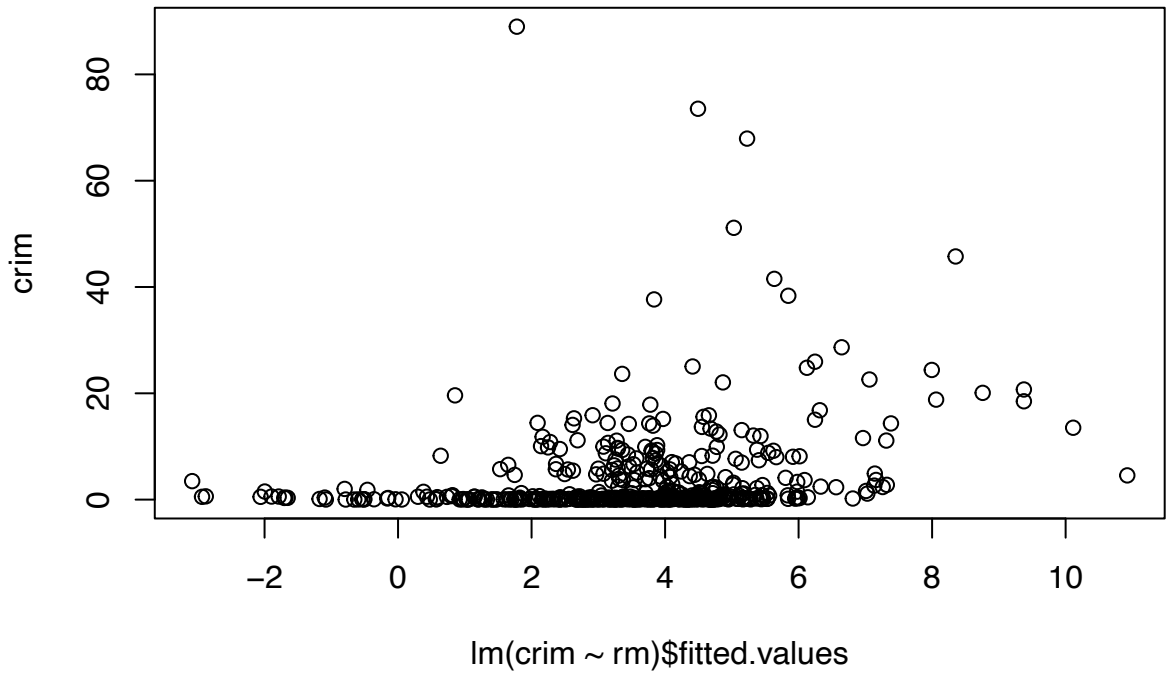
	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1	zn	-0.07	0.02	-4.59	0.00	-0.11	-0.04
2	indus	0.51	0.05	9.99	0.00	0.41	0.61
3	chas	-1.89	1.51	-1.26	0.21	-4.85	1.07
4	nox	31.25	3.00	10.42	0.00	25.36	37.14
5	rm	-2.68	0.53	-5.04	0.00	-3.73	-1.64
6	age	0.11	0.01	8.46	0.00	0.08	0.13
7	dis	-1.55	0.17	-9.21	0.00	-1.88	-1.22
8	rad	0.62	0.03	18.00	0.00	0.55	0.69
9	tax	0.03	0.00	16.10	0.00	0.03	0.03
10	ptratio	1.15	0.17	6.80	0.00	0.82	1.48
11	black	-0.04	0.00	-9.37	0.00	-0.04	-0.03
12	lstat	0.55	0.05	11.49	0.00	0.45	0.64
13	medv	-0.36	0.04	-9.46	0.00	-0.44	-0.29

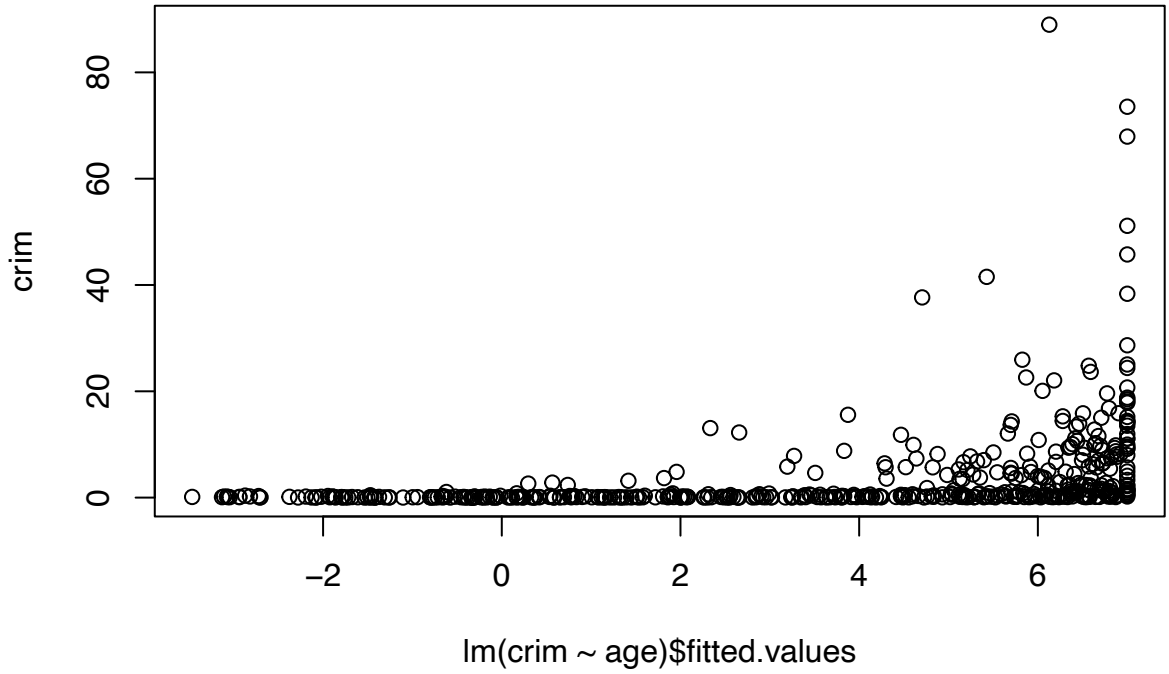
All variables except for chas (bound by Charles river) return significant coefficients for predicting crime rate. Variables zn, rm, dis returns negative slope, whereas indus, nox, age, rad, tax and ptratio return significant positive betas. Variables rad and tax return highest absolute values of t-statistic implying lower p-values and greater confidence on the betas.

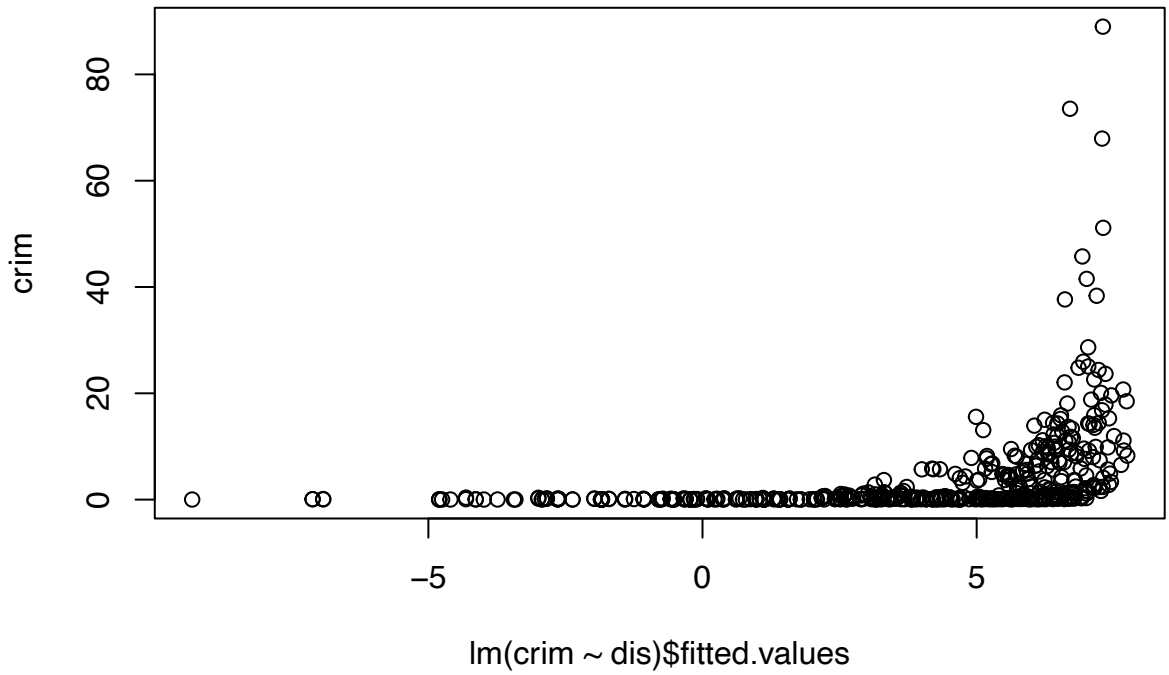


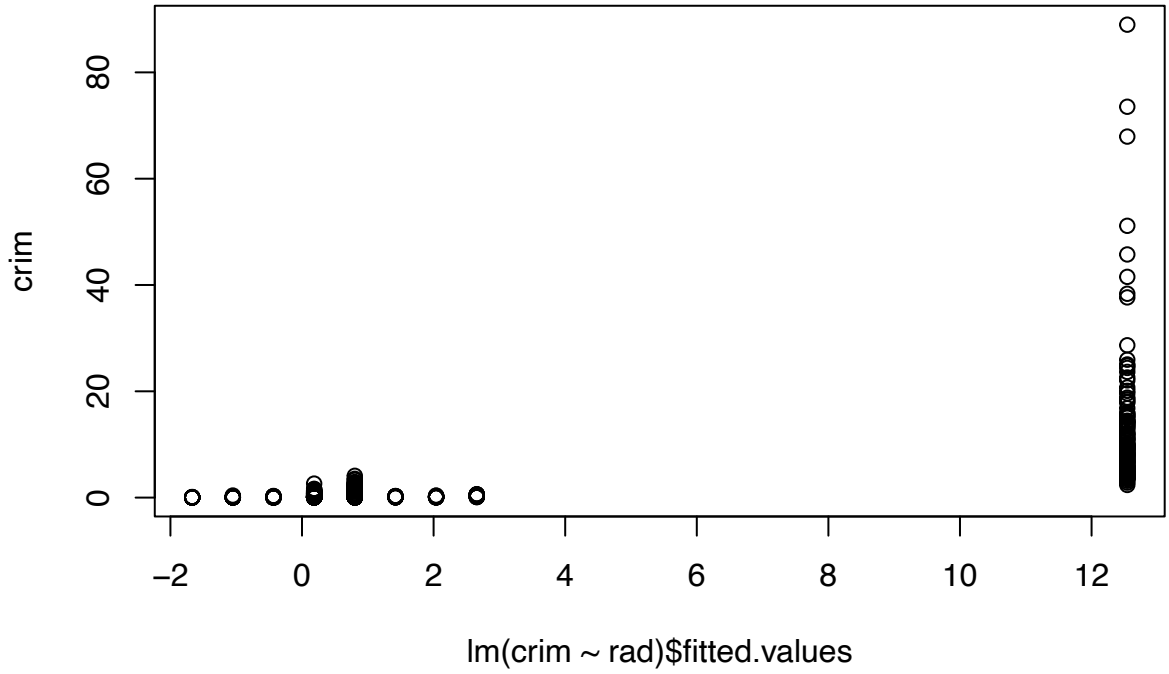

crim
0
20
40
60
80
−2
0
2
4
6
8
10
12
lm(crim ~ indus)$fitted.values

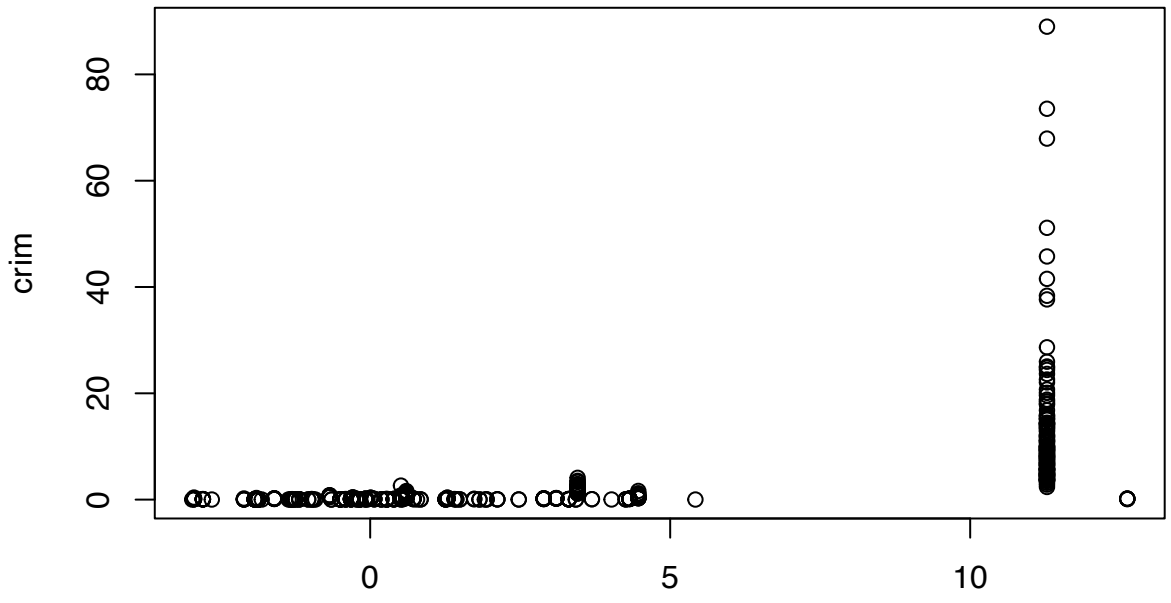

crim
0
20
40
60
80
0
5
10
lm(crim ~ nox)$fitted.values

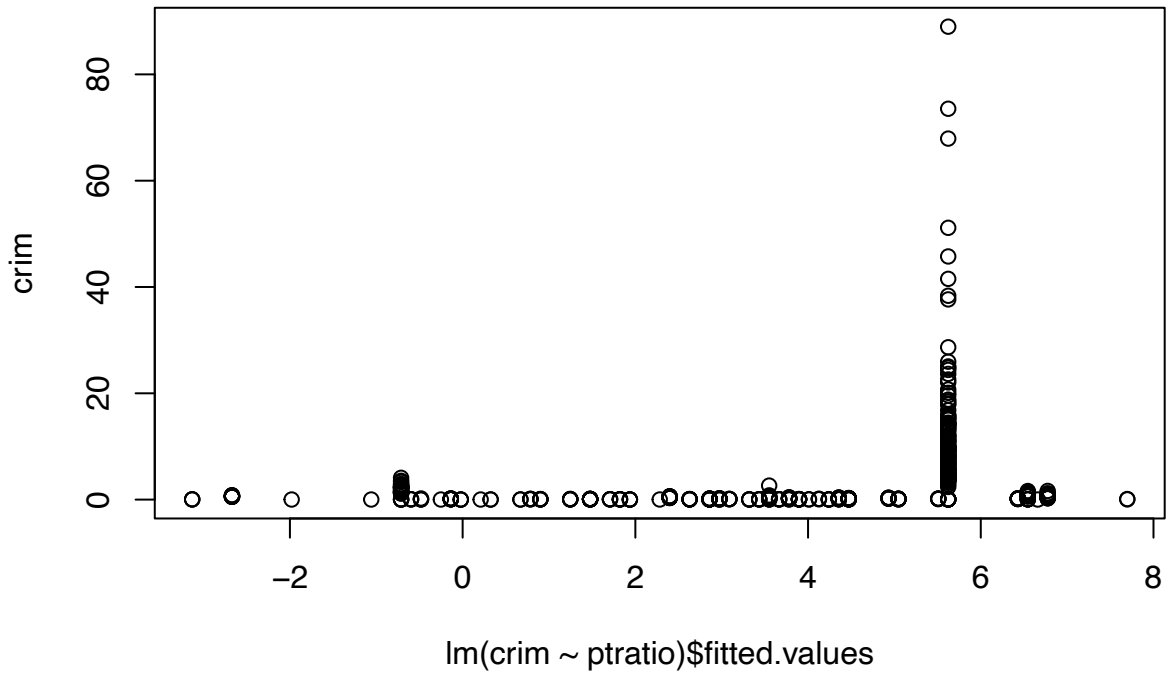

crim
0
20
40
60
80
−2
0
2
4
6
8
10
lm(crim ~ rm)$fitted.values

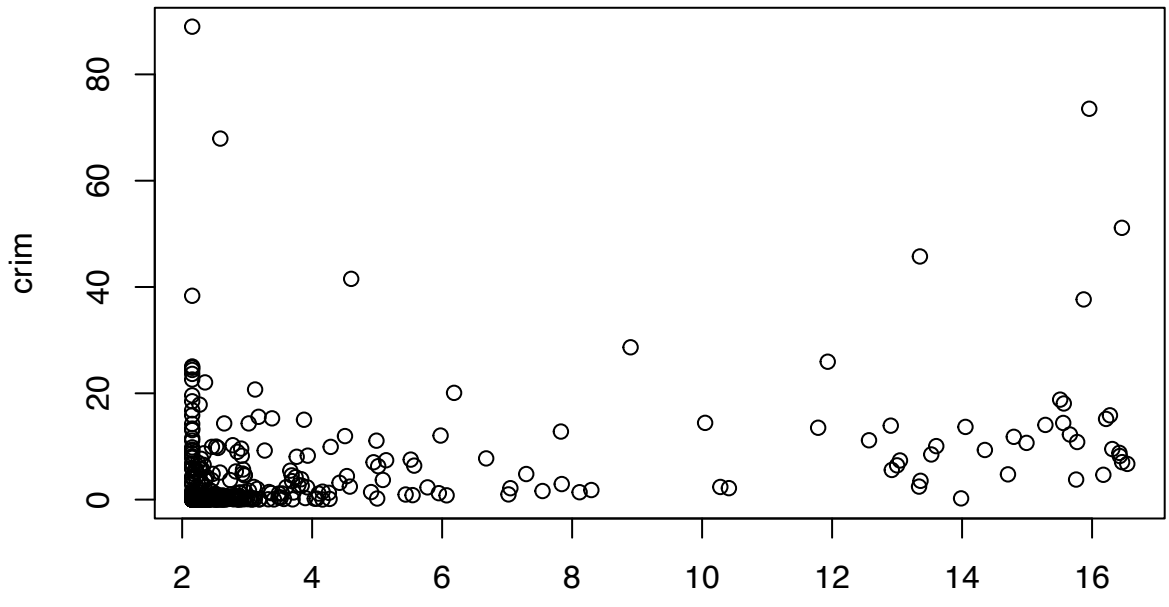

crim
0
20
40
60
80
−2
0
2
4
6
lm(crim ~ age)$fitted.values

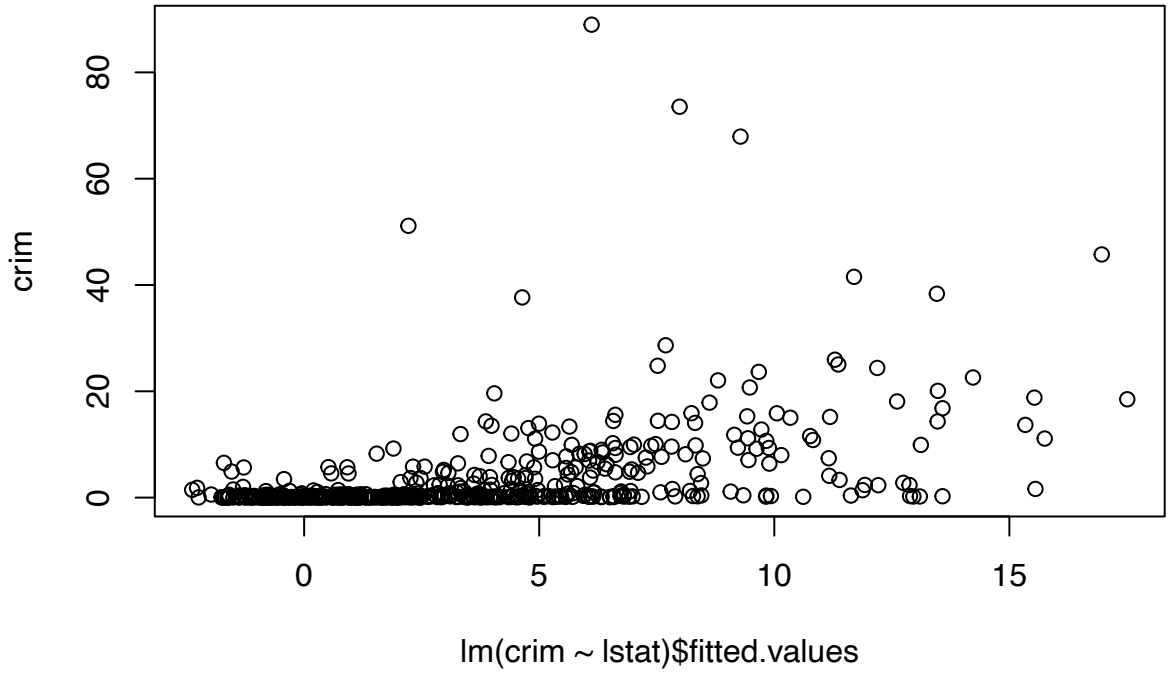

crim
0
20
40
60
80
−5
0
5
lm(crim ~ dis)$fitted.values

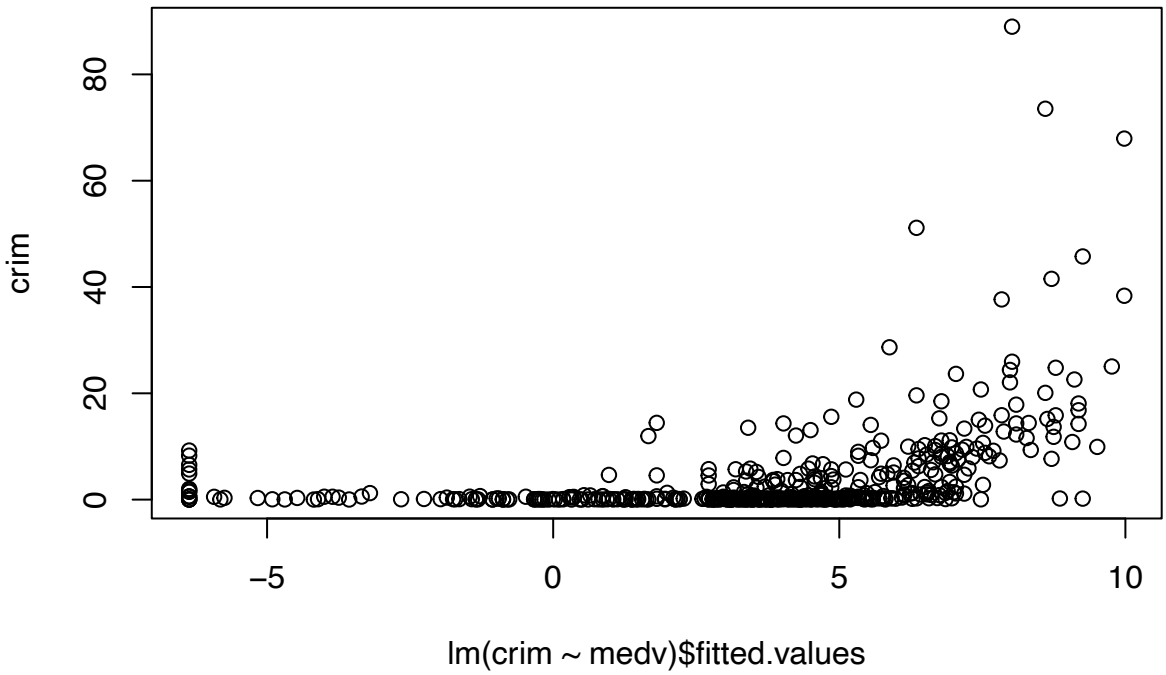

crim
0
20
40
60
80
−2
0
2
4
6
8
10
12
lm(crim ~ rad)$fitted.values


crim
0
20
40
60
80
0
5
10
lm(crim ~ tax)$fitted.values


crim
0
20
40
60
80
−2
0
2
4
6
8
lm(crim ~ ptratio)$fitted.values


crim
0
20
40
60
80
2
4
6
8
10
12
14
16
lm(crim ~ black)$fitted.values


crim
0
20
40
60
80
0
5
10
15
lm(crim ~ lstat)$fitted.values



Above plots show regression fitted values on x vs actual crim values on y! We do not see an 'x = y' relationship between these plots implying univariate linear fit might not be explaining full variance.

Plots in Q2c. show regression lines with respect to each variable and crim - blue one with OLS estimates and band of grey indicating confidence interval.

As we can see from the plot,we have atleast 95% confidence that the slope of the regression line is non-zero for all regression line except chas which has upper and lower confidence bands with positive and negative slopes respectively implying we can't satisfactorily say that slope isn't zero.

This proves aforementioned statistically significant association of all remaining variables.

Part (b)

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

Residuals:
   Min     1Q Median     3Q    Max
-9.924 -2.120 -0.353  1.019 75.051

Coefficients:
              Estimate Std. Error t value        Pr(>|t|)
(Intercept)  17.033228   7.234903   2.354        0.018949 *
zn            0.044855   0.018734   2.394        0.017025 *
indus        -0.063855   0.083407  -0.766        0.444294
chas         -0.749134   1.180147  -0.635        0.525867
nox         -10.313535   5.275536  -1.955        0.051152 .
rm            0.430131   0.612830   0.702        0.483089
age           0.001452   0.017925   0.081        0.935488
dis          -0.987176   0.281817  -3.503        0.000502 ***
rad           0.588209   0.088049   6.680 0.0000000000646 ***
tax          -0.003780   0.005156  -0.733        0.463793
ptratio      -0.271081   0.186450  -1.454        0.146611
```

```
black        -0.007538   0.003673  -2.052        0.040702 *
lstat         0.126211   0.075725   1.667        0.096208 .
medv         -0.198887   0.060516  -3.287        0.001087 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.439 on 492 degrees of freedom
Multiple R-squared:  0.454, Adjusted R-squared:  0.4396
F-statistic: 31.47 on 13 and 492 DF,  p-value: < 0.00000000000000022
```

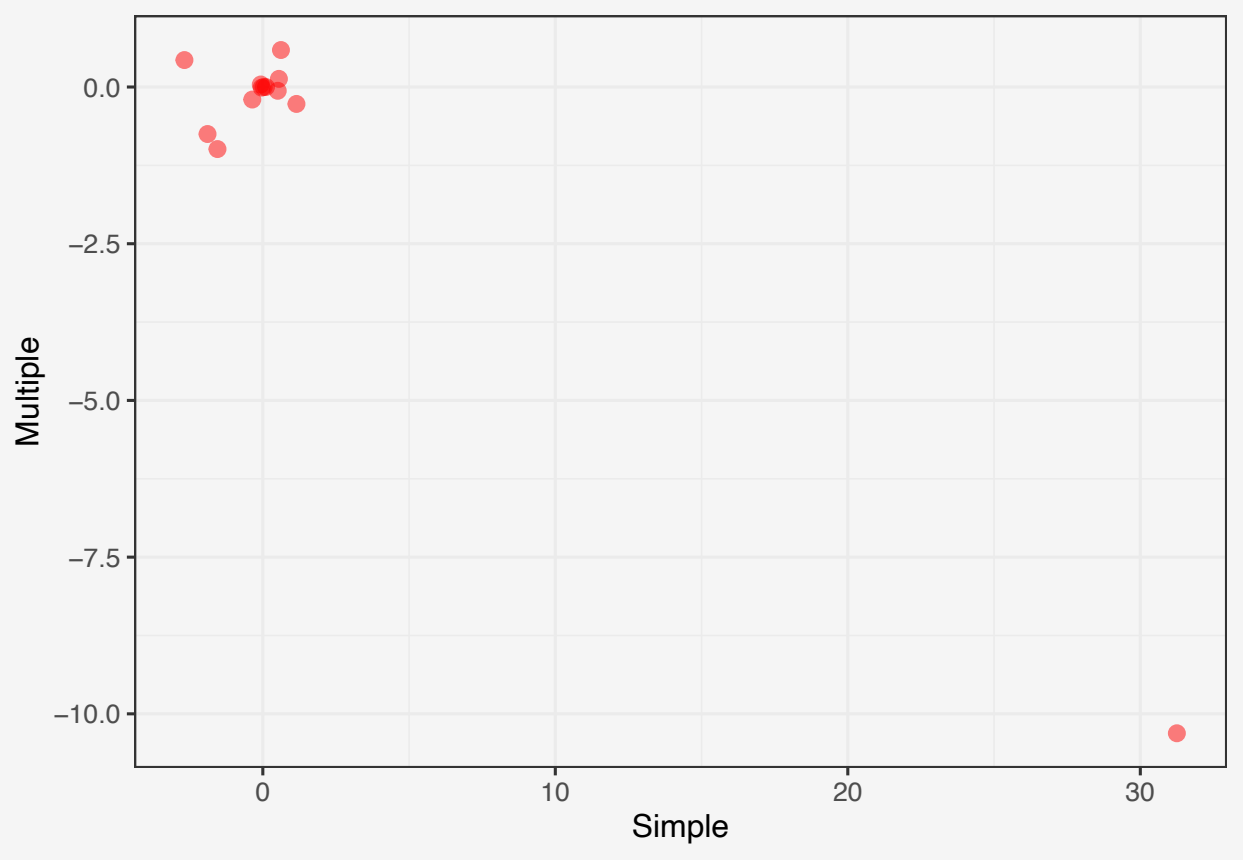
We can reject the null hypothesis (beta = 0) for the variables zn, dis, rad, black and medv.

We observe some betas may have changed directionality from their initial results in simple regression. Case in point, zn in univariate regression returns a negative coefficient whereas returns positive statistically significant in multivariate setting. This is key insight since now we know given all other covariates are kept constant, per unit change in zn, will increase crime rate which would be in contrast of observation from a scatterplot of the same that shows negative slope for fitted line.

Part (c) Compare simple linear model and MLR betas

```
        Simple Multiple
zn       -0.07     0.04
indus     0.51    -0.06
chas     -1.89    -0.75
nox      31.25   -10.31
rm       -2.68     0.43
age       0.11     0.00
dis      -1.55    -0.99
rad       0.62     0.59
tax       0.03     0.00
ptratio   1.15    -0.27
black    -0.04    -0.01
lstat     0.55     0.13
medv     -0.36    -0.20
```

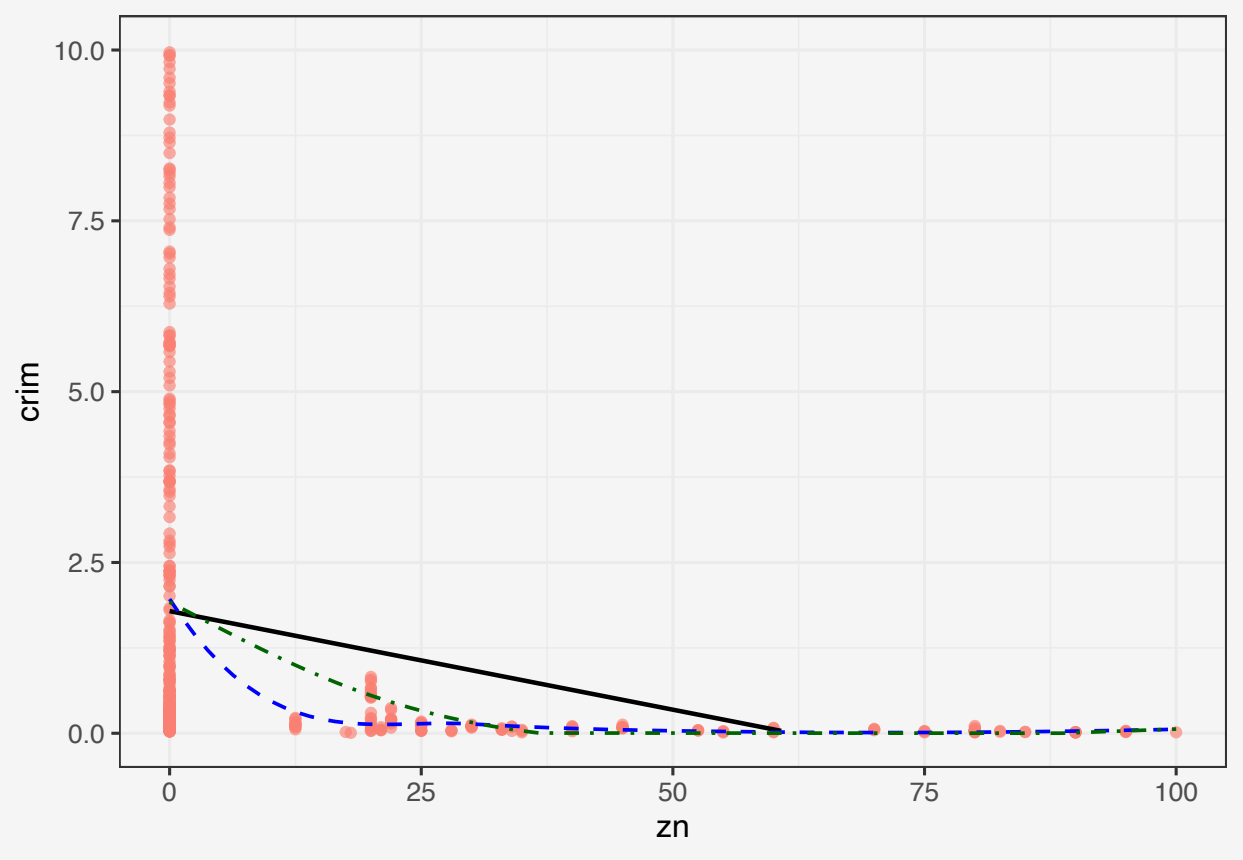
We observe directionality of coefficients changes for zn, indus, nox, rm, ptratio when all variables are considered together.



Coefficient for nox sees most deviation, changing from 31 in simple lr to -10 in multiple regression. As stated above, these changes are due to the fact that multiple regression provides coefficients under the condition that all remaining covariates are held constant whereas simple linear regression only takes in consideration the correlation between the dependent and independent variable.

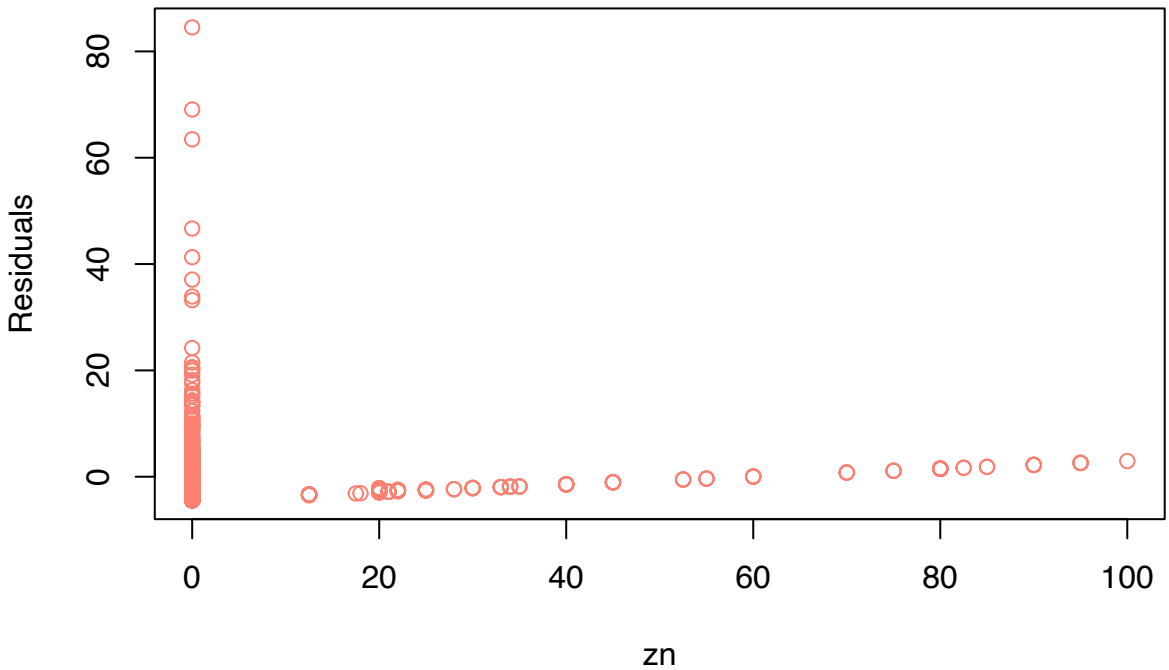
Part (d) Approach 1 : We can check whether non-linear curves are able to fit our data better and make a visual reading

```
‘geom_smooth()‘ using formula ’y ~ x’
‘geom_smooth()‘ using formula ’y ~ x’

‘geom_smooth()‘ using formula ’y ~ s(x, bs = "cs")’
```



Plotting variables and fitting local polynomial regression fitting or generalized additive models we see there is some non-linearity present in relationship of crim with zn.

Approach 2 : Is there any pattern remaining in the residuals post linear regression?



We see non-linearity in residual vs independent variable plot meaning linear model has not been able to extract the non-linear aspect of the relationship between crim and zn.

Approach 3 : We can model polynomial terms and check whether they result significant betas

	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1	poly(zn, 3)1	-38.75	8.37	-4.63	0.00	-55.20	-22.30
2	poly(zn, 3)2	23.94	8.37	2.86	0.00	7.49	40.39
3	poly(zn, 3)3	-10.07	8.37	-1.20	0.23	-26.52	6.38
4	poly(indus, 3)1	78.59	7.42	10.59	0.00	64.01	93.18
5	poly(indus, 3)2	-24.39	7.42	-3.29	0.00	-38.98	-9.81
6	poly(indus, 3)3	-54.13	7.42	-7.29	0.00	-68.71	-39.55
7	poly(chas, 1)	-10.80	8.60	-1.26	0.21	-27.69	6.09
8	poly(nox, 3)1	81.37	7.23	11.25	0.00	67.16	95.58
9	poly(nox, 3)2	-28.83	7.23	-3.99	0.00	-43.04	-14.62
10	poly(nox, 3)3	-60.36	7.23	-8.34	0.00	-74.57	-46.15
11	poly(rm, 3)1	-42.38	8.33	-5.09	0.00	-58.74	-26.01
12	poly(rm, 3)2	26.58	8.33	3.19	0.00	10.21	42.94
13	poly(rm, 3)3	-5.51	8.33	-0.66	0.51	-21.88	10.85
14	poly(age, 3)1	68.18	7.84	8.70	0.00	52.78	83.58
15	poly(age, 3)2	37.48	7.84	4.78	0.00	22.08	52.89
16	poly(age, 3)3	21.35	7.84	2.72	0.01	5.95	36.76
17	poly(dis, 3)1	-73.39	7.33	-10.01	0.00	-87.79	-58.98
18	poly(dis, 3)2	56.37	7.33	7.69	0.00	41.97	70.78
19	poly(dis, 3)3	-42.62	7.33	-5.81	0.00	-57.03	-28.22
20	poly(rad, 3)1	120.91	6.68	18.09	0.00	107.78	134.04
21	poly(rad, 3)2	17.49	6.68	2.62	0.01	4.36	30.62
22	poly(rad, 3)3	4.70	6.68	0.70	0.48	-8.43	17.83
23	poly(tax, 3)1	112.65	6.85	16.44	0.00	99.18	126.11
24	poly(tax, 3)2	32.09	6.85	4.68	0.00	18.62	45.55
25	poly(tax, 3)3	-8.00	6.85	-1.17	0.24	-21.46	5.47
26	poly(ptratio, 3)1	56.05	8.12	6.90	0.00	40.09	72.00
27	poly(ptratio, 3)2	24.77	8.12	3.05	0.00	8.82	40.73
28	poly(ptratio, 3)3	-22.28	8.12	-2.74	0.01	-38.24	-6.32
29	poly(black, 3)1	-74.43	7.95	-9.36	0.00	-90.06	-58.80
30	poly(black, 3)2	5.93	7.95	0.75	0.46	-9.70	21.55
31	poly(black, 3)3	-4.83	7.95	-0.61	0.54	-20.46	10.79
32	poly(lstat, 3)1	88.07	7.63	11.54	0.00	73.08	103.06
33	poly(lstat, 3)2	15.89	7.63	2.08	0.04	0.90	30.88
34	poly(lstat, 3)3	-11.57	7.63	-1.52	0.13	-26.56	3.42
35	poly(medv, 3)1	-75.06	6.57	-11.43	0.00	-87.96	-62.15
36	poly(medv, 3)2	88.09	6.57	13.41	0.00	75.18	100.99
37	poly(medv, 3)3	-48.03	6.57	-7.31	0.00	-60.94	-35.13

We observe quadratic terms of the variables zn, rm, rad, tax, lstat report out significant co-efficients. Similarly cubic terms of the variables indus, nox, age, dis, ptratio, medv report significant co-efficients. This implies crime rate has valid non-linear relationships with multiple covariates.

Chapter #6 : Ques 9

Part (a) Import college data from ISLR package

```
Number of rows and columns in train are : 582  and  18  respectively.
```

```
Number of rows and columns in test are : 195  and  18  respectively.
```

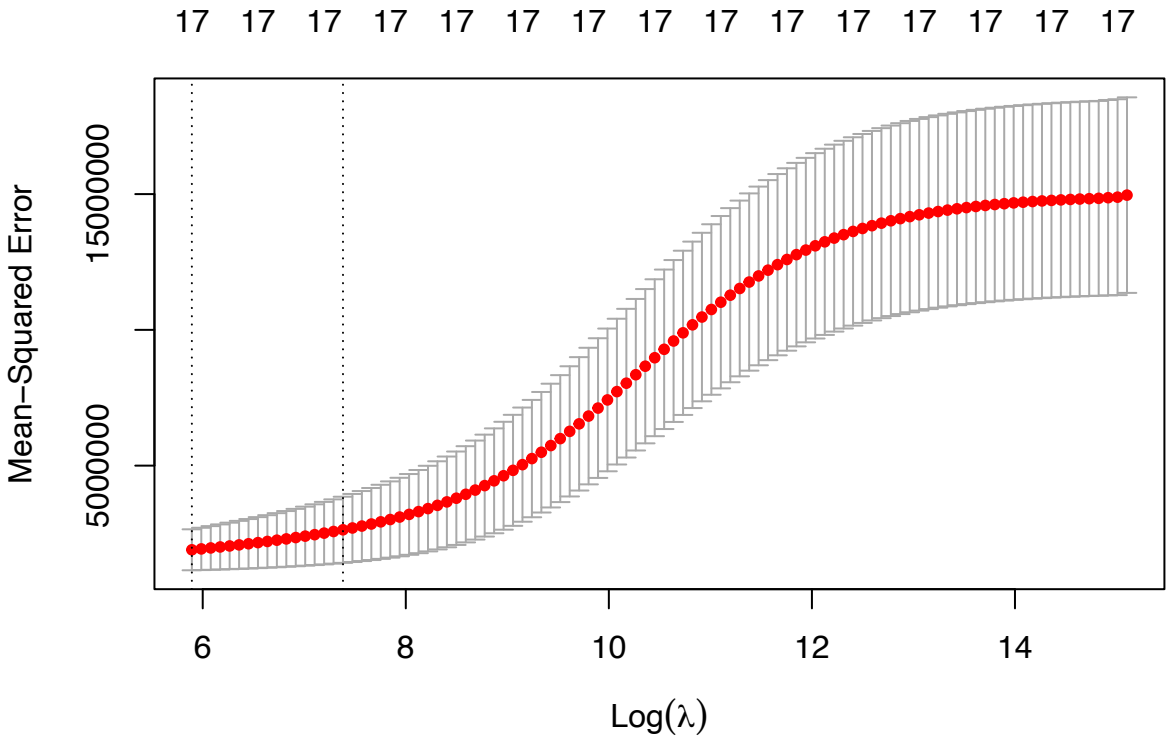
Train test split made with train set having 75% of datapoints and remaining in test set. Dependent variable is Apps. We do not see any overlap in train and test rownames.

Part (b) Running MLR

```
Call:
lm(formula = Apps ~ ., data = college)

Residuals:
    Min      1Q  Median      3Q     Max
-4908.8  -430.2   -29.5   322.3  7852.5

Coefficients:
             Estimate Std. Error t value             Pr(>|t|)
(Intercept) -445.08413  408.32855  -1.090             0.276053
Private     -494.14897  137.81191  -3.586             0.000358 ***
Accept         1.58581    0.04074  38.924 < 0.0000000000000002 ***
Enroll        -0.88069    0.18596  -4.736       0.000002602777 ***
Top10perc     49.92628    5.57824   8.950 < 0.0000000000000002 ***
Top25perc    -14.23448    4.47914  -3.178             0.001543 **
F.Undergrad    0.05739    0.03271   1.754             0.079785 .
P.Undergrad    0.04445    0.03214   1.383             0.167114
Outstate      -0.08587    0.01906  -4.506       0.000007638013 ***
Room.Board     0.15103    0.04829   3.127             0.001832 **
Books          0.02090    0.23841   0.088             0.930175
Personal       0.03110    0.06308   0.493             0.622060
PhD           -8.67850    4.63814  -1.871             0.061714 .
Terminal      -3.33066    5.09494  -0.654             0.513492
S.F.Ratio     15.38961   13.00622   1.183             0.237081
perc.alumni    0.17867    4.10230   0.044             0.965273
Expend         0.07790    0.01235   6.308       0.000000000479 ***
Grad.Rate      8.66763    2.94893   2.939             0.003390 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1041 on 759 degrees of freedom
Multiple R-squared:  0.9292,    Adjusted R-squared:  0.9276
F-statistic: 585.9 on 17 and 759 DF,  p-value: < 0.00000000000000022

Test Root mean squared error is :  904.6638

Train RMSE is : 1067.699
```

We see train RMSE is bit higher than test set RMSE.

Part (c) Ridge penalty



Above plot shows results from the devaince optimization with respect to lambda and returns lambda with minimum mean squared error and lambda with MSE within 1st standard deviation as lambda.1se

The ranges in the plot are indicative of the variance being explained, as lambda goes down we see the model being regularized and model variance being curtailed. We use lambda.1se for regularizing our model.

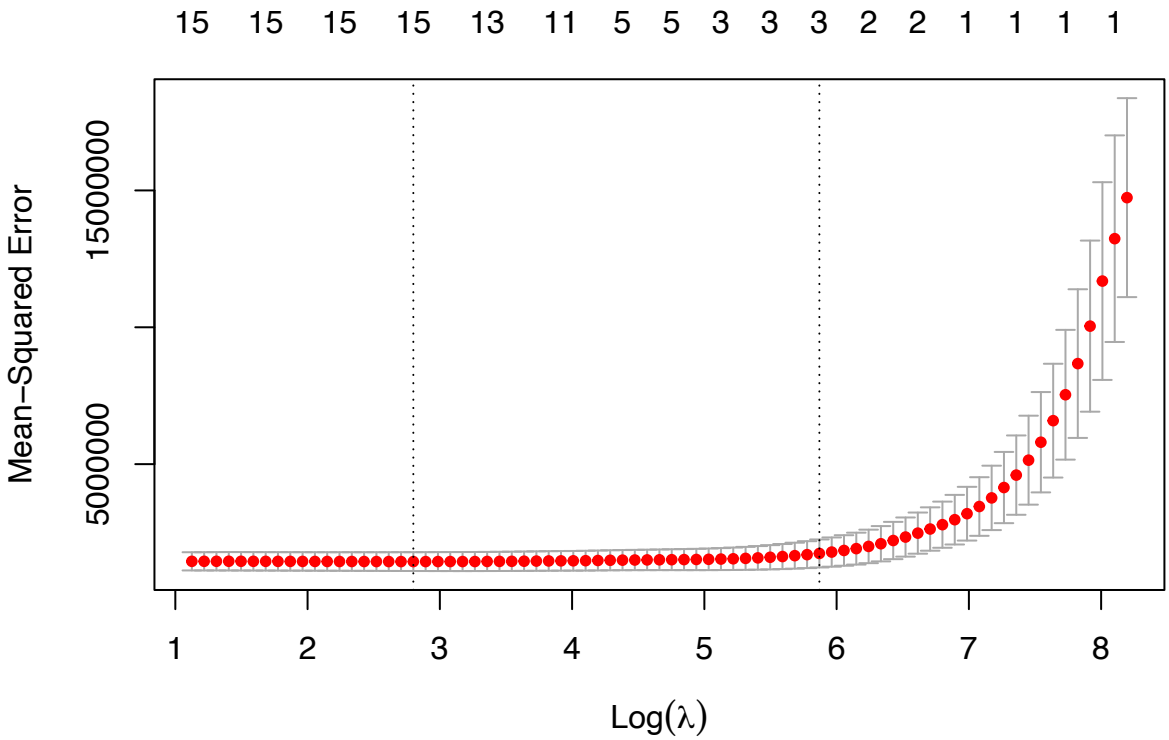
```
Test Root mean squared error for Ridge regularized model is :  1017.826
```

Ridge regression increases the test set RMSE a little bit compared to the MLR model at lambda.1se. Using lambda.min probably reduces test set RMSE marginally.

```
                VarName     Beta
(Intercept) (Intercept) -2461.71
Private         Private  -512.26
Accept           Accept     0.65
Enroll           Enroll     0.74
Top10perc     Top10perc    18.02
Top25perc     Top25perc     6.81
F.Undergrad F.Undergrad     0.12
P.Undergrad P.Undergrad     0.04
Room.Board   Room.Board     0.22
Books             Books     0.29
Personal       Personal     0.01
PhD                 PhD     1.96
Terminal       Terminal    -0.53
S.F.Ratio     S.F.Ratio    15.45
perc.alumni perc.alumni   -10.29
Expend           Expend     0.06
Grad.Rate     Grad.Rate    11.22
```

As expected we get coefficients of all variables with not-so-important betas closer to zero.

Part (d) Implementing Lasso penalty



We see model variance falls rapidly as lambda decreases with similar variance being explained at lambda.min and lambda.1se

```
Test Root mean squared error for Lasso regularized model is :  961.0086
```

```
                VarName    Beta
(Intercept) (Intercept) -348.35
Accept           Accept    1.34
Top10perc     Top10perc   19.36
Expend           Expend    0.01
```

Lasso regression reduces the test set RMSE compared to Ridge implementations. This is owing to the fact that the Lasso penalty is able to throw out unneeded variables and gives out a more generalized model.

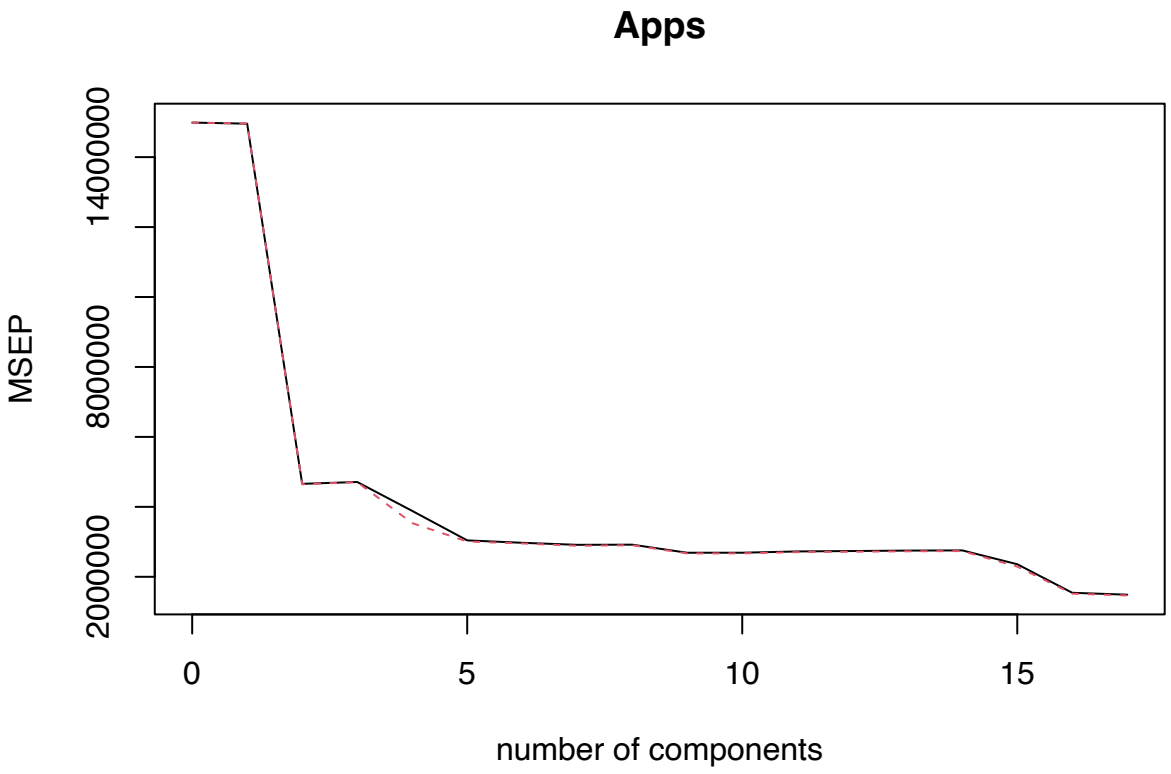
Also given variance explained is pretty much similar at lambda.min and lambda.1se, we'd expect similar test RMSE for lambda.min as well

Lasso at lambda.1se makes most betas to zero and shares betas associated with only Accept, Top10perc and Expend variables.

Part (e) Implementing Principal Component Regression

```
Data:   X dimension: 582 17
    Y dimension: 582 1
Fit method: svdpc
Number of components considered: 17

VALIDATION: RMSEP
```

```
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV            3871     3867     2158     2170     1971     1744     1724
adjCV         3871     3868     2155     2170     1880     1734     1718
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV        1707     1708     1639      1640      1651      1654      1658
adjCV     1699     1703     1633      1634      1644      1648      1652
       14 comps  15 comps  16 comps  17 comps
CV         1660      1536      1242      1220
adjCV      1654      1514      1232      1211

TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X      31.692    57.66    64.73    70.36    75.83    80.73    84.28    87.62
Apps    1.417    69.96    69.96    79.70    81.36    81.74    82.22    82.27
      9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
X       90.63     93.02     95.08     96.89     97.93     98.71     99.37
Apps    83.86     84.12     84.13     84.13     84.20     84.20     90.14
      16 comps  17 comps
X        99.84    100.00
Apps     92.32     92.49
```

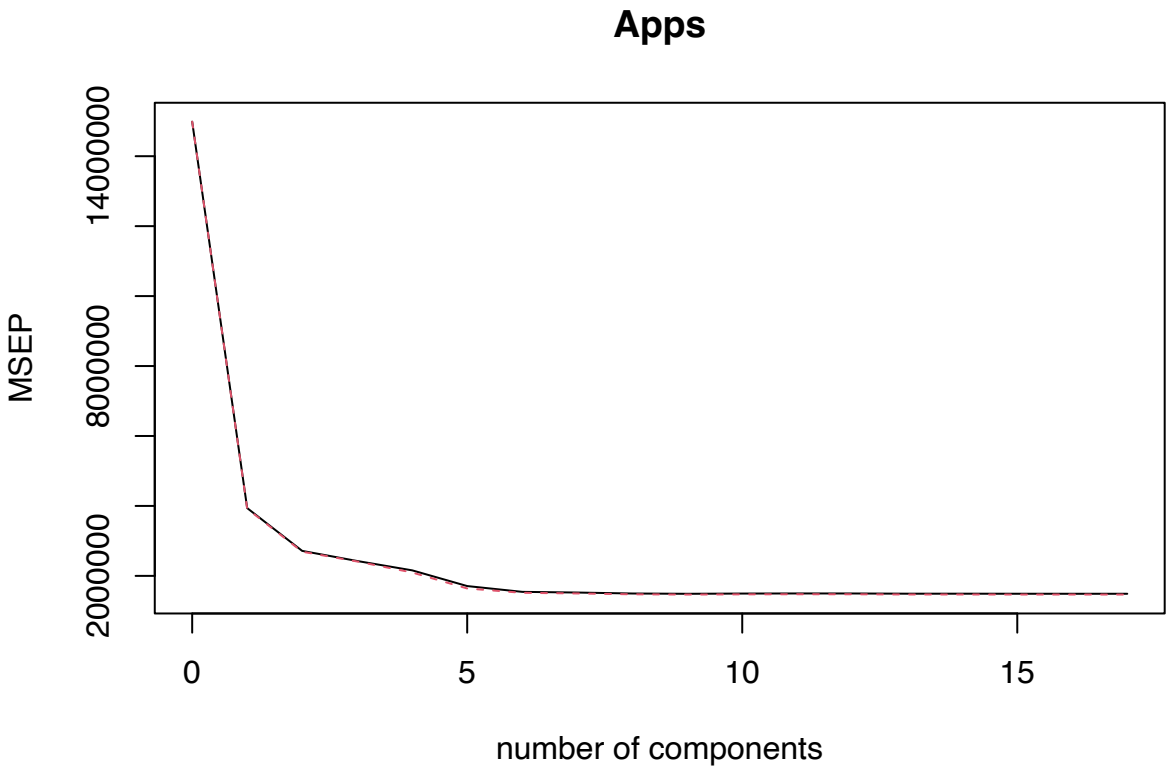


We observe the CV results and validation plots. Given the mean squared error of prediction saturates at number of components = 9, we choose M = 9 for our final model. This ensure that over model is simpler, more genralized.

```
Test Root mean squared error for PCR  model is :  1807.272
```

Even upon choosing M via cross-validation, PCR model gives worse RMSE than Ridge or Lasso.

Part (f) Implementing PLS model




```
Test Root mean squared error for PLS model is :  1008.716
```

PLS model performs better than pricipal component regression but is still not able to beat RMSE scores from Lasso at lambda.1se on test set.

Part (g) Comparing all models

```
   Model    RMSE
1:    LR  904.66
2: Ridge 1017.83
3: Lasso  961.01
4:   PCR 1807.27
5:   PLS 1008.72
```

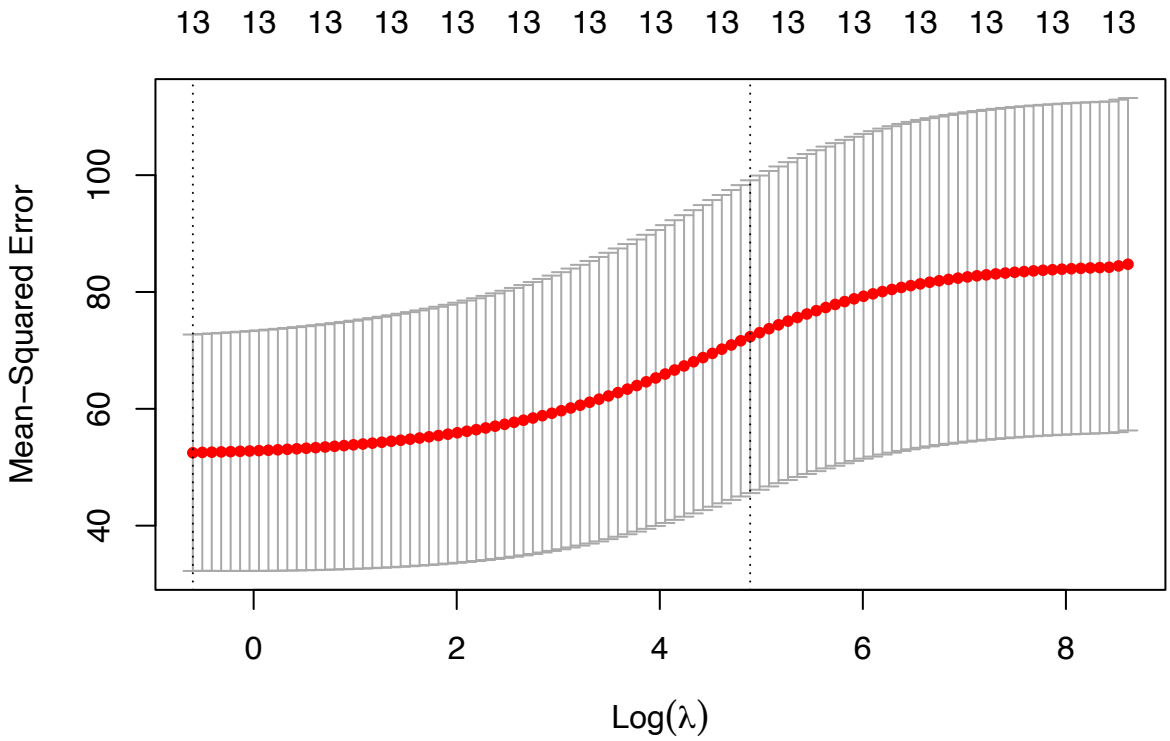
We see Lasso despite omitting out few variables gives us a better RMSE than all other regularized/PCA based models. Best RMSE is availed by the multiple linear regression. Interestingly, there's quite a difference between RMSE values of PCR and PLS despite both of them being based on dimensionality reduction while greedily maximizing variance.

This implies that the supervised transformations that PLS does post PCA offer good predictive power in estimating dependent variable.

Chapter #6 : Ques 11

Part (a) We redo the sampling train test, run Ridge, Lasso and PCR models as in the above example. In addition we do the best subset implementation and report out the RMSE from linear model of these

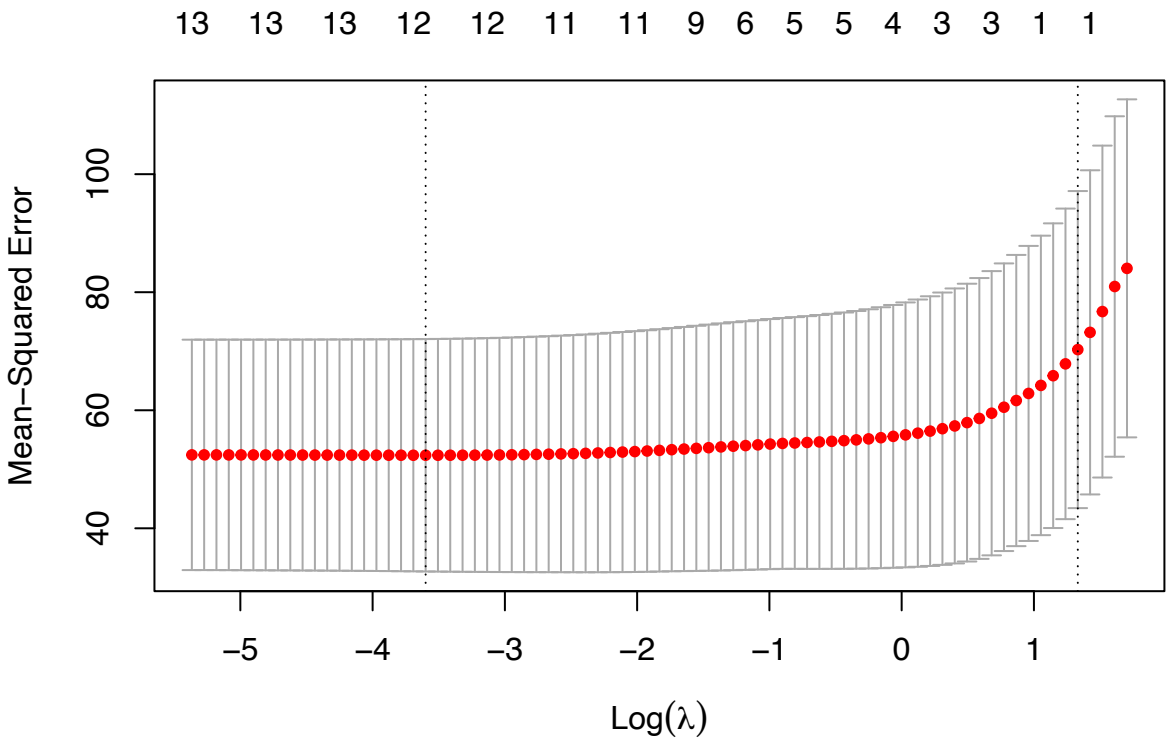
best-subsetted variables. As above, we haven't scaled the variables since glmnet is able to do so internally as stated here : https://www.statology.org/ridge-regression-in-r/



	VarName	Beta
(Intercept)	(Intercept)	2.05
zn	zn	0.00
indus	indus	0.02
chas	chas	-0.12
nox	nox	1.49
rm	rm	-0.10
age	age	0.01
dis	dis	-0.08
rad	rad	0.03
tax	tax	0.00
ptratio	ptratio	0.06
black	black	0.00
lstat	lstat	0.03
medv	medv	-0.02

We see ridge is not able to regularize our model much. Betas of quite a few variables are closer to zero.

```
Test Root mean squared error for Ridge regularized model is :  5.51462
```




```
                VarName Beta
(Intercept) (Intercept) 1.89
rad                 rad 0.19
```

Lasso regularization also has similar MSE vs lambda plot and is not reducing model variance by much. Also lasso is removing out almost all the variables. We try to use a custom lambda that is less regularized and gives out few more variables with non-zero betas.

```
                VarName  Beta
(Intercept) (Intercept)  7.18
zn                  zn  0.02
dis                dis -0.34
rad                rad  0.48
ptratio        ptratio -0.05
black            black -0.01
medv              medv -0.18
```

Lasso now is able to give out more variables with non-zero betas.

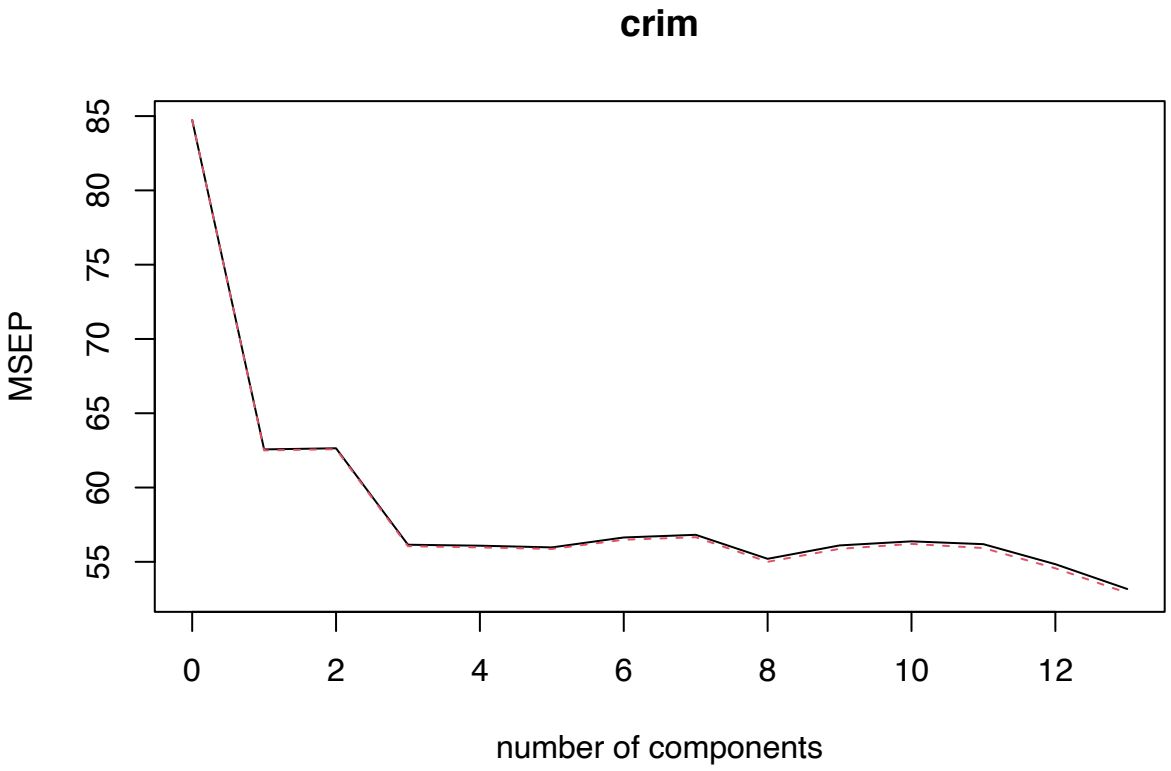
```
Test Root mean squared error for Lasso regularized model is :  4.061179
```

Test MSE is slightly lower than Ridge model at lambda.1se

Custom lambda further reduces Test MSE a bit, given we're moving our penalty parameter towards lambda.min.

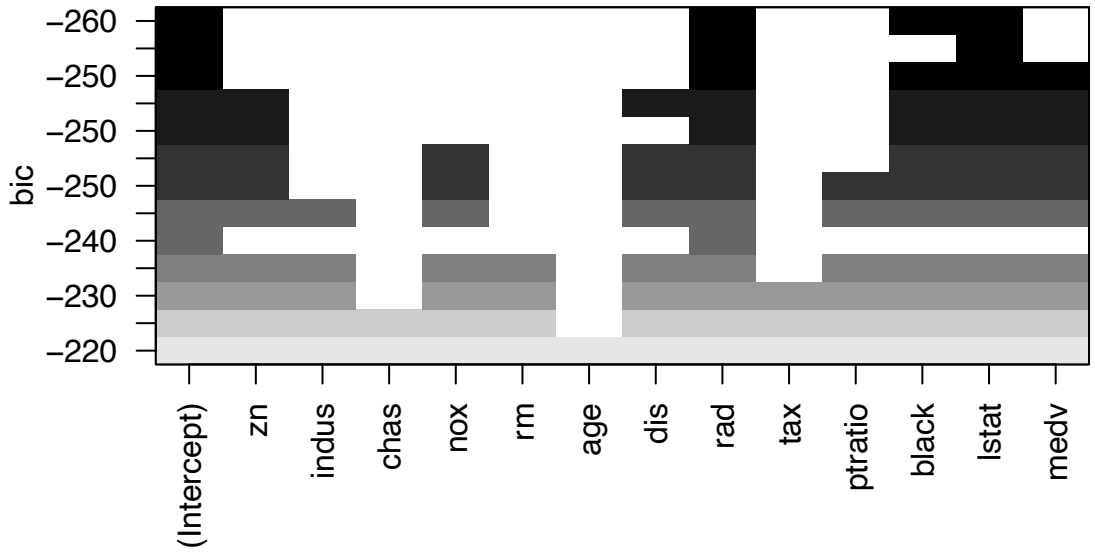
```
Data:   X dimension: 379 13
    Y dimension: 379 1
Fit method: svdpc
```

```
Number of components considered: 13

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           9.205    7.910    7.915    7.493    7.489    7.481    7.526
adjCV        9.205    7.906    7.911    7.487    7.481    7.474    7.516
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV       7.538    7.430    7.490     7.509     7.496     7.406     7.292
adjCV    7.527    7.416    7.475     7.497     7.479     7.387     7.274

TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X       48.53    60.32    70.17    76.73    83.14    88.25    91.28    93.65
crim    27.24    27.34    35.40    35.66    35.90    36.13    36.16    38.74
      9 comps  10 comps  11 comps  12 comps  13 comps
X       95.66     97.11     98.53      99.5    100.00
crim    38.77     38.88     39.21      40.9     42.61
```




```
Test Root mean squared error for PCR  model is :  4.210351
```

PCR test RMSE is also closer to Lasso Test MSE. We choose number of components such that our CV errors and Adj. R-squares are minimum or stable (thereby making our model more robust to slight hyperparameter fluctuations)




```
NULL


Subset selection object
Call: regsubsets.formula(crim ~ ., data = Boston, nvmax = 15, method = "forward")
13 Variables  (and intercept)
        Forced in Forced out
zn          FALSE      FALSE
indus       FALSE      FALSE
chas        FALSE      FALSE
nox         FALSE      FALSE
rm          FALSE      FALSE
age         FALSE      FALSE
dis         FALSE      FALSE
rad         FALSE      FALSE
tax         FALSE      FALSE
ptratio     FALSE      FALSE
black       FALSE      FALSE
lstat       FALSE      FALSE
medv        FALSE      FALSE
1 subsets of each size up to 13
Selection Algorithm: forward
          zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " "
2  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   "*"   " "
3  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " "
4  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   "*"
5  ( 1 )  "*" " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   "*"
6  ( 1 )  "*" " "   " "  " " " " " " " "*" "*" " " " "     "*"   "*"   "*"
7  ( 1 )  "*" " "   " "  "*" " " " " " "*" "*" " " " "     "*"   "*"   "*"
8  ( 1 )  "*" " "   " "  "*" " " " " " "*" "*" " " "*"     "*"   "*"   "*"
9  ( 1 )  "*" "*"   " "  "*" " " " " " "*" "*" " " "*"     "*"   "*"   "*"
10  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" " " "*"     "*"   "*"   "*"
11  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
12  ( 1 ) "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
```

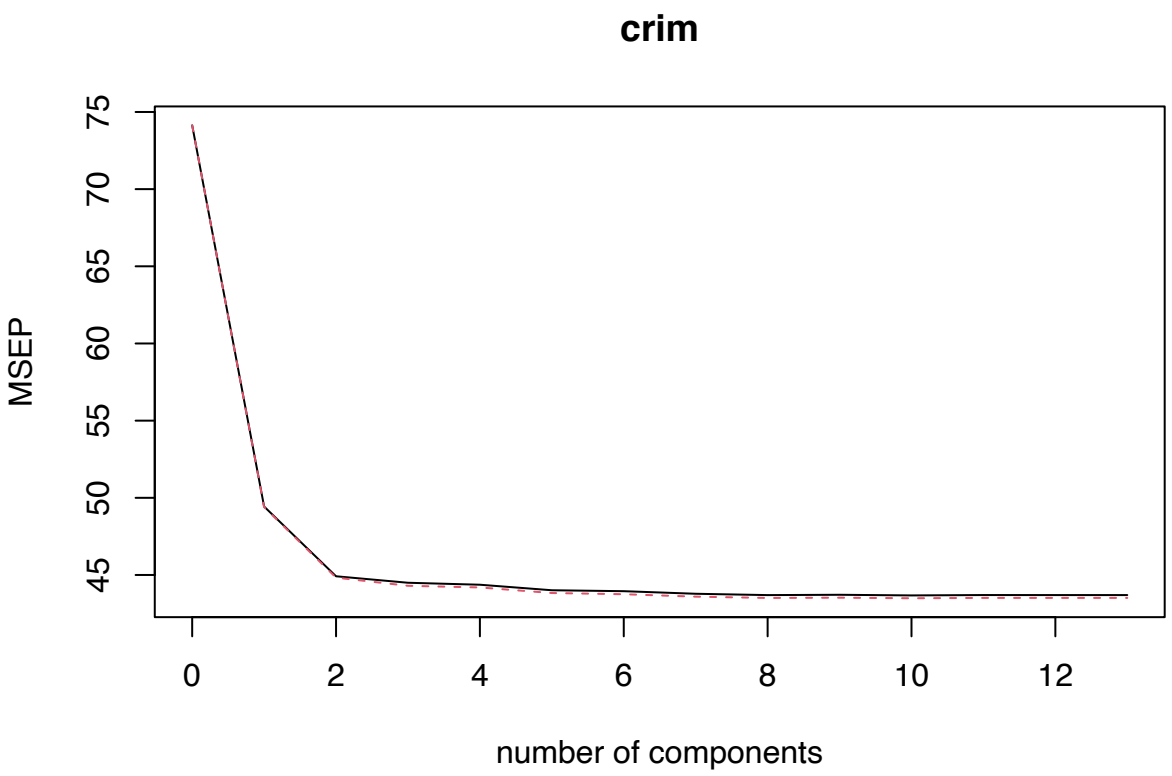
```
[1] 0.4425053


Call:
lm(formula = crim ~ zn + indus + nox + dis + rad + ptratio +
    black + lstat + medv, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max
-9.946 -2.086 -0.358  0.974 75.644

Coefficients:
              Estimate Std. Error t value             Pr(>|t|)
(Intercept)  19.124636   6.095437   3.138             0.001805 **
zn            0.042788   0.017967   2.381             0.017620 *
indus        -0.099386   0.074207  -1.339             0.181085
nox         -10.466490   5.053655  -2.071             0.038869 *
dis          -1.002598   0.269182  -3.725             0.000218 ***
rad           0.539504   0.049953  10.800 < 0.0000000000000002 ***
ptratio      -0.270836   0.185183  -1.463             0.144230
black        -0.008004   0.003613  -2.215             0.027199 *
lstat         0.117806   0.069398   1.698             0.090223 .
medv         -0.180594   0.054155  -3.335             0.000918 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.422 on 496 degrees of freedom
Multiple R-squared:  0.4524,    Adjusted R-squared:  0.4425
F-statistic: 45.54 on 9 and 496 DF,  p-value: < 0.00000000000000022


Test Root mean squared error is :  3.832065


        Model     RMSE
1: Subsetting 3.832065
2:      Ridge 5.514620
3:      Lasso 4.061179
4:        PCR 4.210351
```

We see best subsetting offers us lowest test set RMSE. From the percent deviation optimization plots of ridge and Lasso, we see that the variance does not drastically change for different values of lambda meaning both L1-L2 regularizations are not able to easily generalize the regression model. Now given Principal component regression is also not able to get better test RMSE, it suggests the actual subset of columns is better able to model unseen data than orthogonal principal components attempting to explain most variance in data.

This could probably be due to relatively high correlation of target variables with x. The orthogonal PCs might be missing out on some correlation and essentially have slightly more information loss.

Hence, despite lasso and PCR being good tools, in theory to weed out unnecessary information and generate stable robust models, in this case, I'd prefer the 'best subset' model as validated by test RMSEs above.

Part (b) It might be interesting to see how PCLS - using supervised transformation of pricipal components explains variance in the data and if it is able to beat the Best subset RMSE benchmark.




```
Test Root mean squared error for PLS model is :  3.798221
```

Very interesting to observe, the supervised transformations PLS model is conducting upon the principal components is able to perform better than the previous best, 'best subset' model. The improvement is although highly sensitive to the number of principal components we avail. If we use lesser number of principal components we essentially risk losing out on the information contained in the variables. On the contrary, if we assume more principal components we have the risk of our model being not generalized enough for test data.

Part (c) The two models that perform best on the hold-out sample are the best subset model and the Partial least squares based on Principal Components. Both variables do not leverage all the variables and the reduction in number of variables helps reduce excess variance in the model, thereby providing robust predictions.

The Best subset method drops out chas, rm, age and tax. Chas is not significant in linear model as well implying the variable might not have the signals to aid in increasing predictive power. Rm almost follow a gaussian distribution. All variables rm, age and tax may have been removed since their effect might have been captured through other correlated variables and adding them ends up over-complicating the model. Similarly, PLS model gives similar accuracy out-of-sample with 5 principal components, maybe reducing confounding effect of independent variables with each other. Compared to the 9-variable best-subset it is a simpler model but loses on model interpretability.

To summarize, if model interpretation is a necessity, the preffered model would be Best subset model vs if accuracy on out-of-sample data is key priority for our model, the PLS model holds best promise of a simpler, robust model.

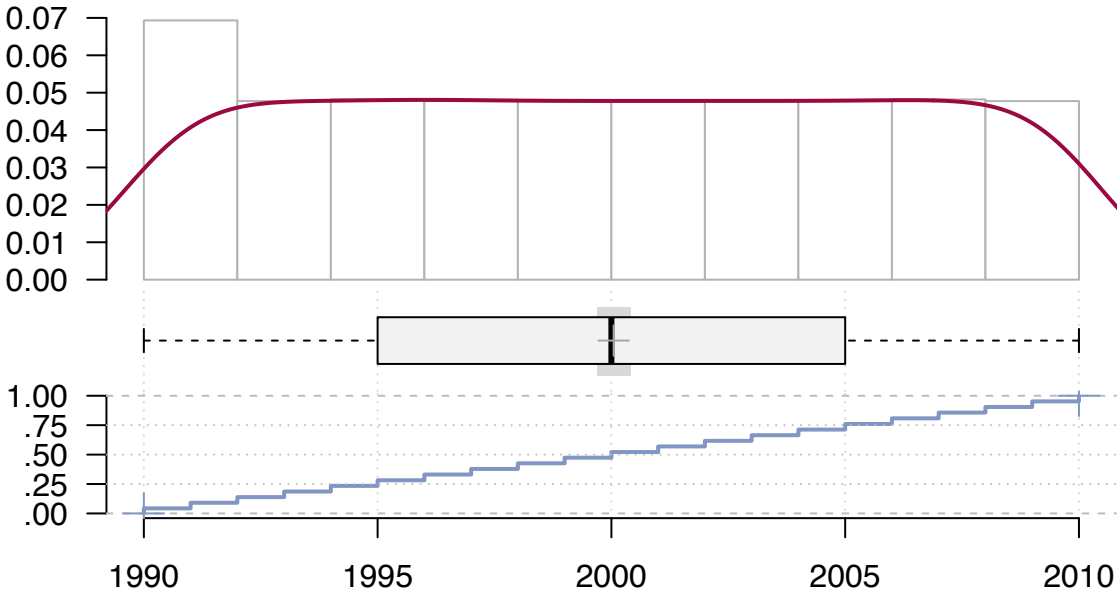
One key point to note is predictions of Crim are coming out to be negative as well given our model wouldn't understand that negative crim rate is not plausible. It treats crime rate as a numeric value that could be scaled and go below 0 if signals in the data suggest to reduce crime rate too much.

Chapter #4 : Ques 10

Part (a) Quick EDA of all variables in new data

```
----------------------------------------------------------------------------
Describe df (data.frame):

data frame: 1089 obs. of  9 variables
        1089 complete cases (100.0%)

  Nr  ColName    Class    NAs  Levels
  1   Year       numeric  .
  2   Lag1       numeric  .
  3   Lag2       numeric  .
  4   Lag3       numeric  .
  5   Lag4       numeric  .
  6   Lag5       numeric  .
  7   Volume     numeric  .
  8   Today      numeric  .
  9   Direction  factor   .    (2): 1-Down, 2-Up


----------------------------------------------------------------------------
1 - Year (numeric)

    length         n       NAs    unique        0s      mean    meanCI'
     1'089     1'089         0        21         0  2'000.05  1'999.69
              100.0%      0.0%               0.0%            2'000.41

       .05       .10       .25    median       .75       .90       .95
  1'991.00  1'992.00  1'995.00  2'000.00  2'005.00  2'008.00  2'009.00

     range        sd     vcoef       mad       IQR      skew      kurt
     20.00      6.03      0.00      7.41     10.00     -0.00     -1.21

lowest : 1'990.0 (47), 1'991.0 (52), 1'992.0 (52), 1'993.0 (52), 1'994.0 (52)
highest: 2'006.0 (52), 2'007.0 (53), 2'008.0 (52), 2'009.0 (52), 2'010.0 (52)

' 95%-CI (classic)
```

1 – Year (numeric)



---

2 - Lag1 (numeric)

length	n	NAs	unique	0s	mean	meanCI'
1'089	1'089	0	1'004	0	0.1506	0.0104
	100.0%	0.0%		0.0%		0.2907

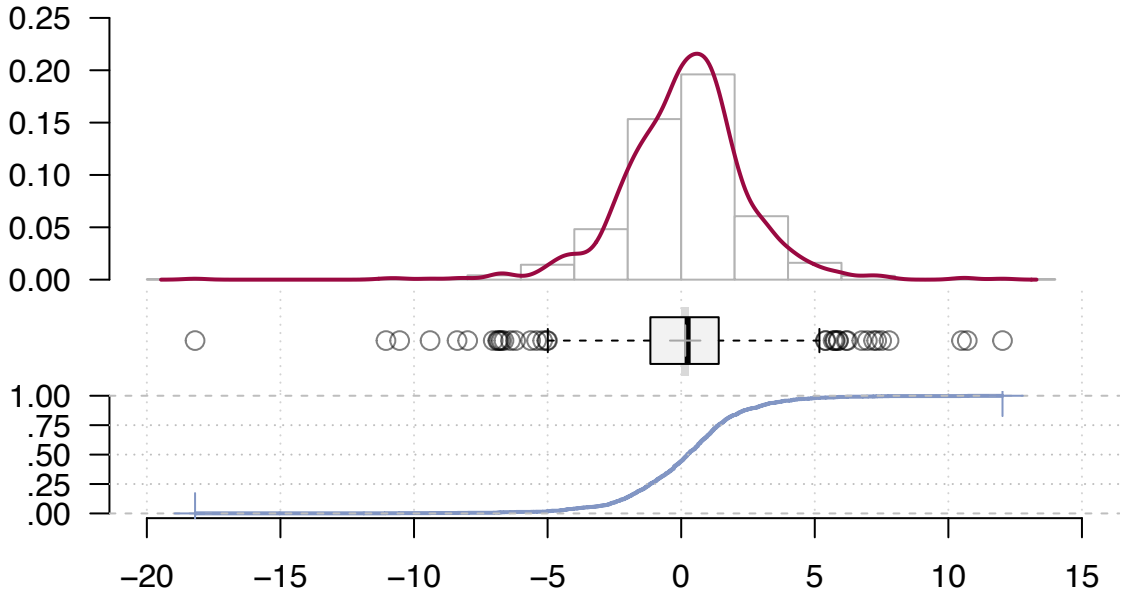
.05	.10	.25	median	.75	.90	.95
-3.6296	-2.4304	-1.1540	0.2410	1.4050	2.8070	3.7376

range	sd	vcoef	mad	IQR	skew	kurt
30.2210	2.3570	15.6524	1.8651	2.5590	-0.4806	5.6689

lowest : -18.1950, -11.0500, -10.5380, -9.3990, -8.3890
highest: 7.503, 7.78, 10.491, 10.707, 12.026

' 95%-CI (classic)

2 – Lag1 (numeric)



---

3 - Lag2 (numeric)

length	n	NAs	unique	0s	mean	meanCI'
1'089	1'089	0	1'005	0	0.1511	0.0109
	100.0%	0.0%		0.0%		0.2912

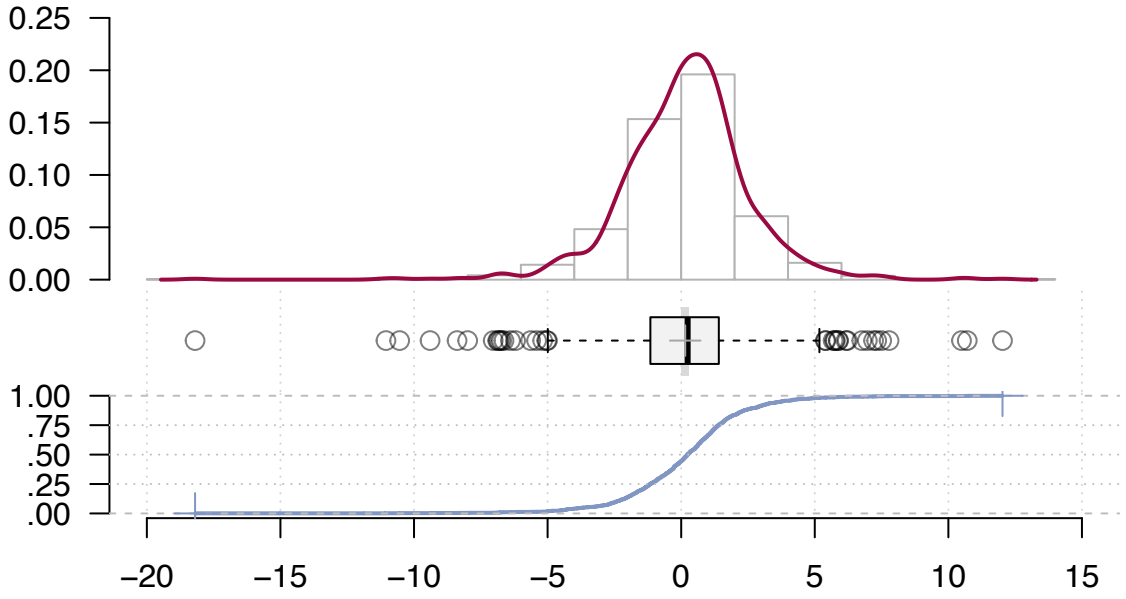
.05	.10	.25	median	.75	.90	.95
-3.6296	-2.4304	-1.1540	0.2410	1.4090	2.8070	3.7376

range	sd	vcoef	mad	IQR	skew	kurt
30.2210	2.3573	15.6028	1.8651	2.5630	-0.4810	5.6658

lowest : -18.1950, -11.0500, -10.5380, -9.3990, -8.3890
highest: 7.503, 7.78, 10.491, 10.707, 12.026

' 95%-CI (classic)

3 – Lag2 (numeric)



---

4 - Lag3 (numeric)

length	n	NAs	unique	0s	mean	meanCI'
1'089	1'089	0	1'005	0	0.1472	0.0069
	100.0%	0.0%		0.0%		0.2876

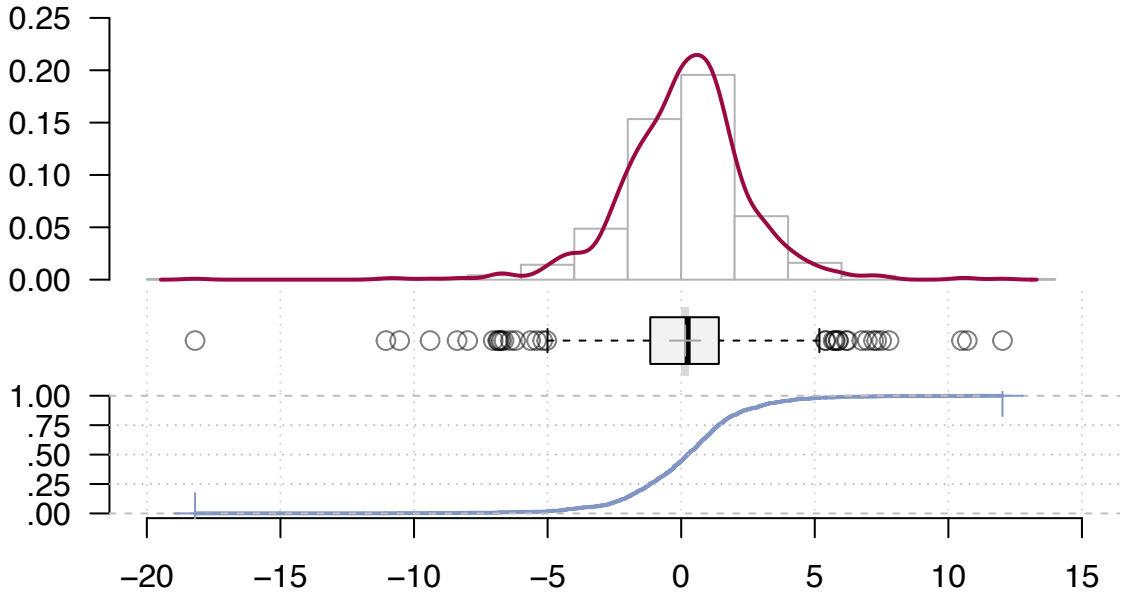
.05	.10	.25	median	.75	.90	.95
-3.6676	-2.4410	-1.1580	0.2410	1.4090	2.8070	3.7376

range	sd	vcoef	mad	IQR	skew	kurt
30.2210	2.3605	16.0355	1.8740	2.5670	-0.4788	5.6233

lowest : -18.1950, -11.0500, -10.5380, -9.3990, -8.3890
highest: 7.503, 7.78, 10.491, 10.707, 12.026

' 95%-CI (classic)

4 – Lag3 (numeric)



---

5 - Lag4 (numeric)

length	n	NAs	unique	0s	mean	meanCI'
1'089	1'089	0	1'005	0	0.1458	0.0055
	100.0%	0.0%		0.0%		0.2862

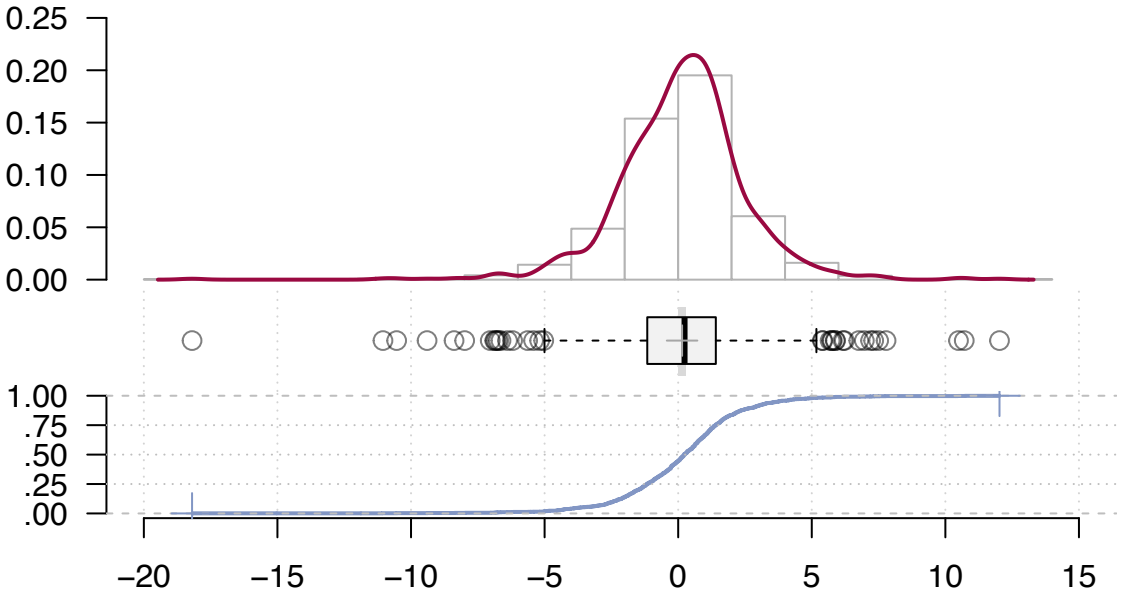
.05	.10	.25	median	.75	.90	.95
-3.6676	-2.4410	-1.1580	0.2380	1.4090	2.8070	3.7376

range	sd	vcoef	mad	IQR	skew	kurt
30.2210	2.3603	16.1865	1.8740	2.5670	-0.4773	5.6254

lowest : -18.1950, -11.0500, -10.5380, -9.3990, -8.3890
highest: 7.503, 7.78, 10.491, 10.707, 12.026

' 95%-CI (classic)

5 – Lag4 (numeric)



---

6 - Lag5 (numeric)

length	n	NAs	unique	0s	mean	meanCI'
1'089	1'089	0	1'005	0	0.1399	-0.0005
	100.0%	0.0%		0.0%		0.2803

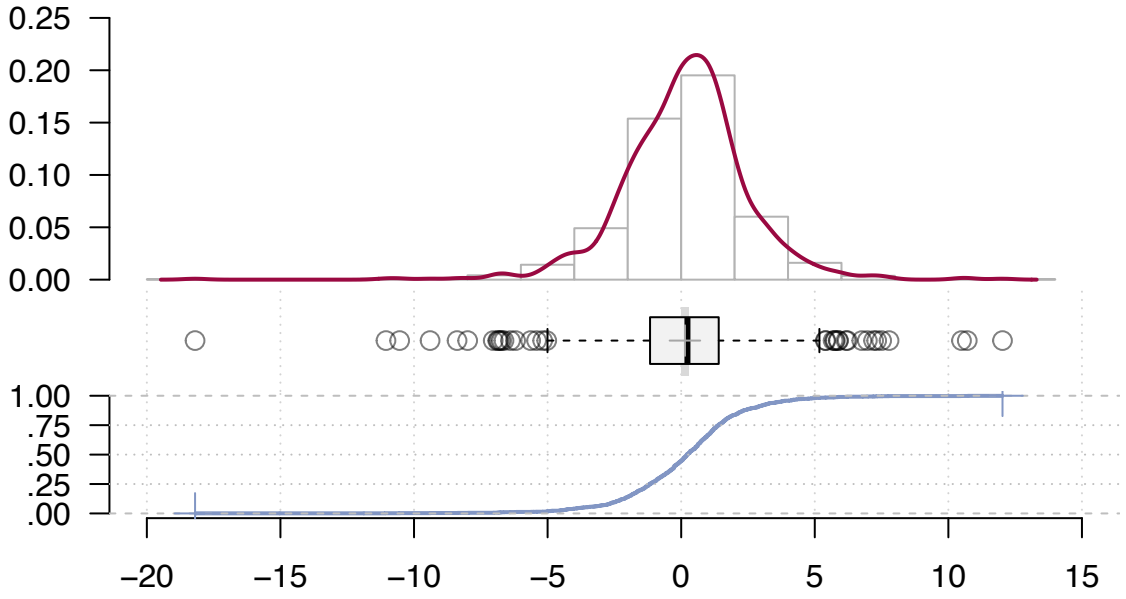
.05	.10	.25	median	.75	.90	.95
-3.6676	-2.4452	-1.1660	0.2340	1.4050	2.7982	3.7376

range	sd	vcoef	mad	IQR	skew	kurt
30.2210	2.3613	16.8793	1.8755	2.5710	-0.4741	5.6092

lowest : -18.1950, -11.0500, -10.5380, -9.3990, -8.3890
highest: 7.503, 7.78, 10.491, 10.707, 12.026

' 95%-CI (classic)

6 – Lag5 (numeric)



---

7 - Volume (numeric)

length	n	NAs	unique	0s	mean	meanCI'
1'089	1'089	0	= n	0	1.574618	1.474332
	100.0%	0.0%		0.0%		1.674903

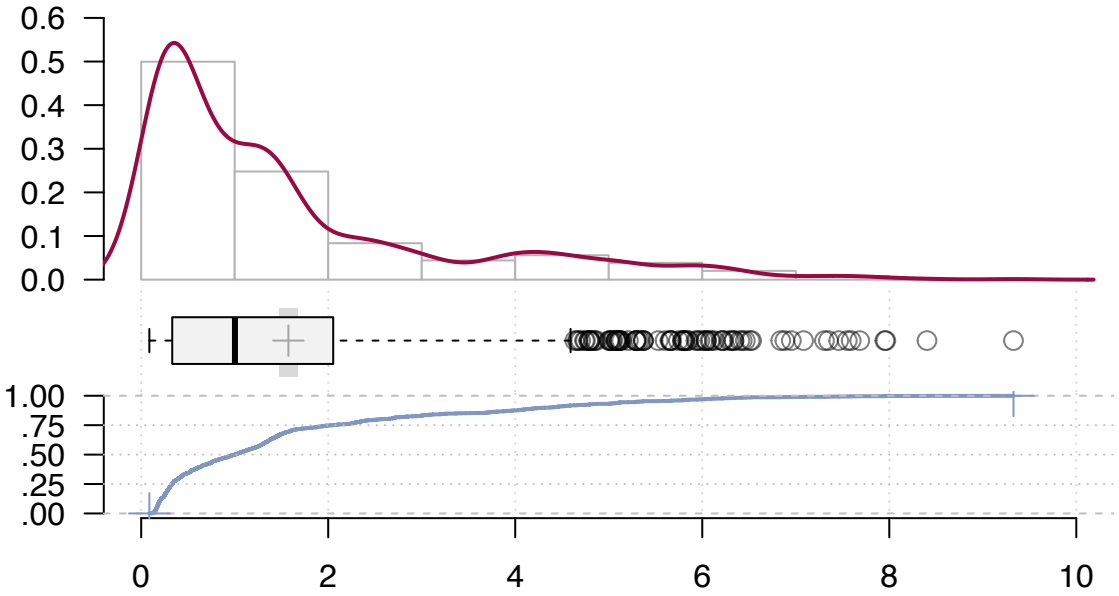
.05	.10	.25	median	.75	.90	.95
0.167552	0.195550	0.332022	1.002680	2.053727	4.365844	5.310663

range	sd	vcoef	mad	IQR	skew	kurt
9.240749	1.686636	1.071140	1.038535	1.721705	1.615958	2.062587

lowest : 0.087465, 0.115742, 0.123924, 0.125075, 0.126532
highest: 7.683886, 7.952024, 7.963276, 8.403358, 9.328214

' 95%-CI (classic)

7 – Volume (numeric)



---

8 - Today (numeric)

length	n	NAs	unique	0s	mean	meanCI’
1’089	1’089	0	1’003	0	0.1499	0.0098
	100.0%	0.0%		0.0%		0.2900

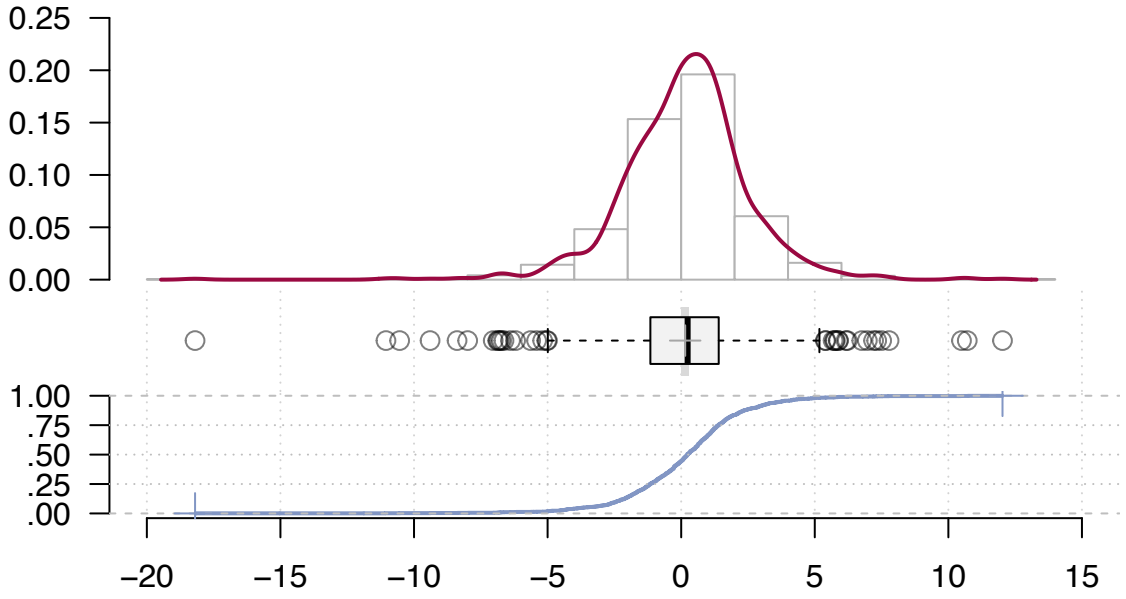
.05	.10	.25	median	.75	.90	.95
-3.6296	-2.4304	-1.1540	0.2410	1.4050	2.8070	3.7376

range	sd	vcoef	mad	IQR	skew	kurt
30.2210	2.3569	15.7234	1.8651	2.5590	-0.4798	5.6696

lowest : -18.1950, -11.0500, -10.5380, -9.3990, -8.3890
highest: 7.503, 7.78, 10.491, 10.707, 12.026

’ 95%-CI (classic)

8 – Today (numeric)

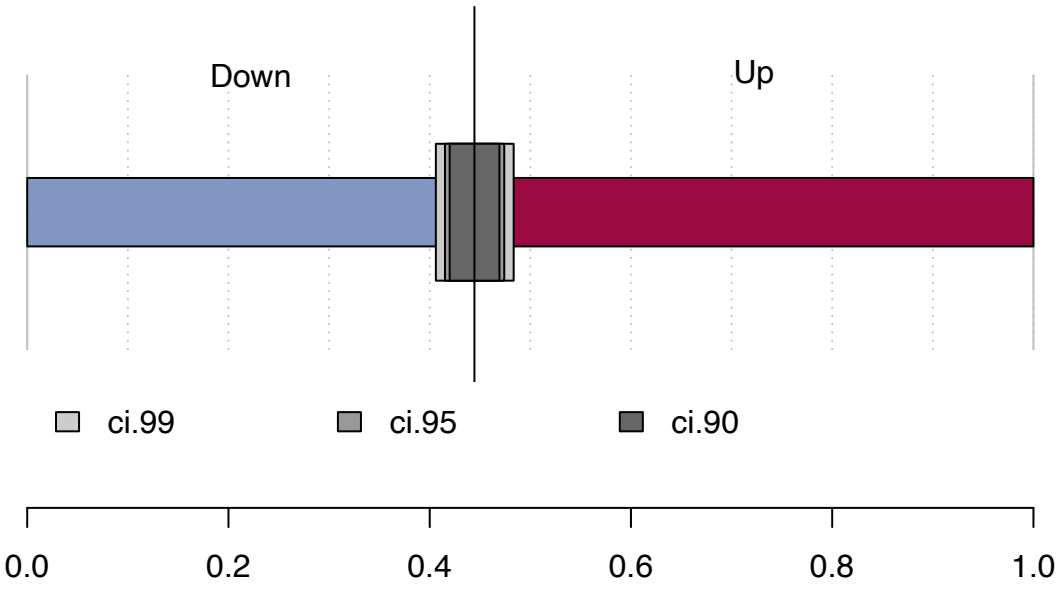


---

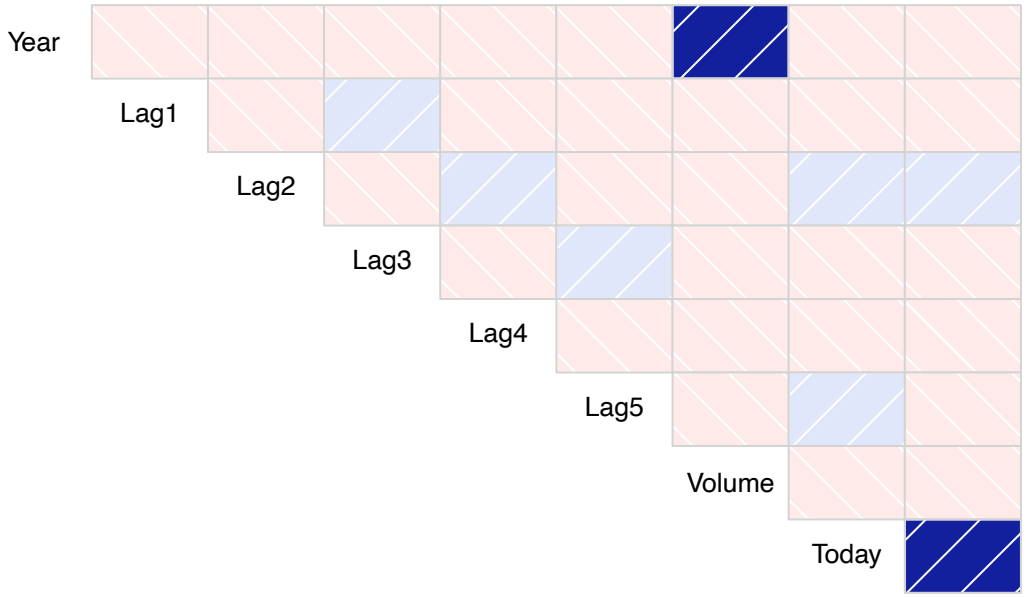
9 - Direction (factor - dichotomous)

```
  length      n    NAs unique
   1'089  1'089      0      2
         100.0%   0.0%

      freq   perc  lci.95  uci.95'
Down   484  44.4%   41.5%   47.4%
Up     605  55.6%   52.6%   58.5%

' 95%-CI (Wilson)
```

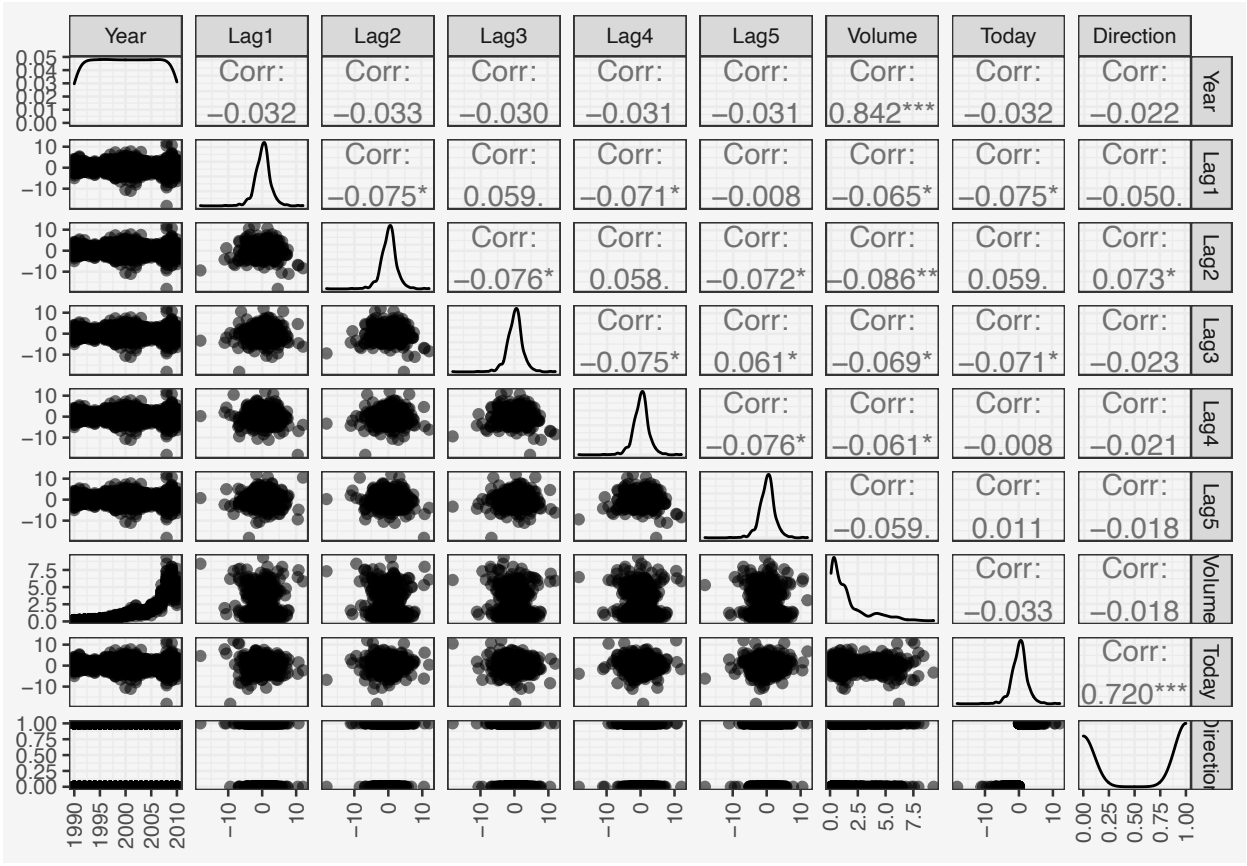
9 – Direction (factor – dichotomous)



Correlations



Pair plots



While univariate plots for all lag variables seem very much the same in terms of the shape of their distributions, we see there is no overall correlation amidst them. Volume variable is highly correlated with Year which makes sense since there wouldn't be too much variation in terms of total volume traded in a given year. Good years should have higher volumes traded, whereas recession periods should have lesser volumes traded.

Direction is highly correlated with the variable Today given Direction (current week up or down) is derived from the variable Today (current week percentage return)

Volume seems to be consistently increasing with the years!

Part (b)

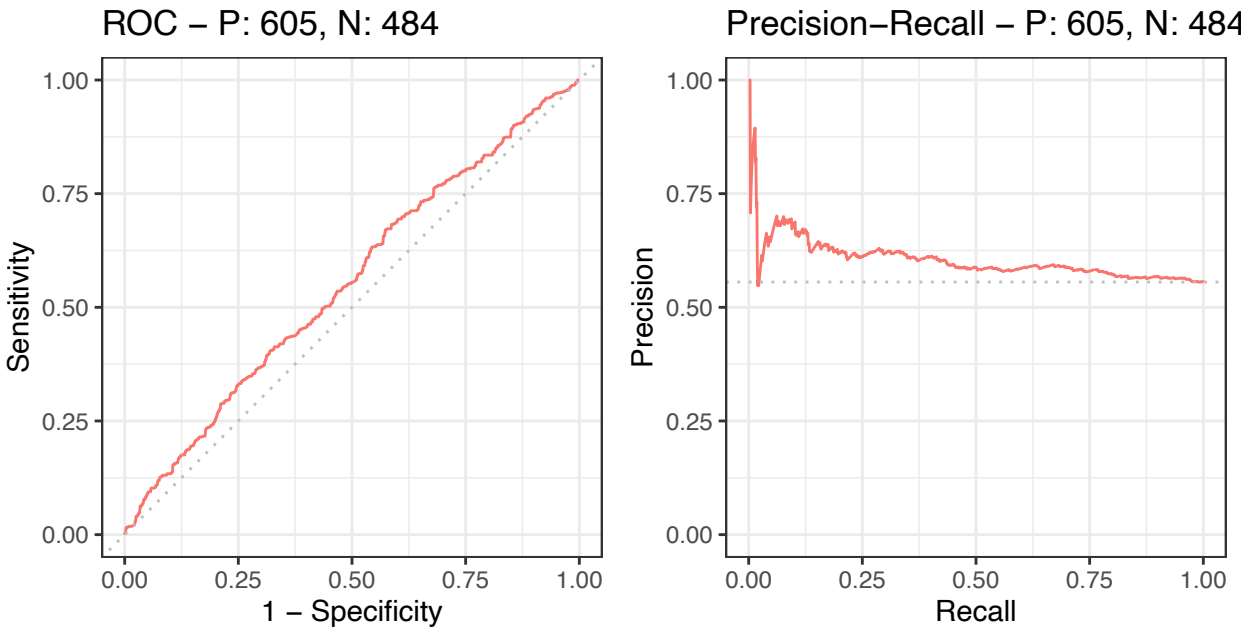
```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
```

```
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4

Number of Fisher Scoring iterations: 4
```

In addition to the Intercept only the Lag2 variable (% Return of last 2 weeks) gives out significant co-efficient.

Part (c) ROC and PR curves along with performance metrics of logit model with multiple variables

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  54  48
         1 430 557

               Accuracy : 0.5611
                 95% CI : (0.531, 0.5908)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : 0.369

                  Kappa : 0.035

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.11157
            Specificity : 0.92066
         Pos Pred Value : 0.52941
         Neg Pred Value : 0.56434
             Prevalence : 0.44444
         Detection Rate : 0.04959
   Detection Prevalence : 0.09366
      Balanced Accuracy : 0.51612

       'Positive' Class : 0


Logit Model regressed using all variables gives AUC of  0.5536985
```



We see not a great model with an AUC 0.55, implying the model is not able to segregate the two classes apart from each other as in evitable from the ROC curve. The model with 0.5 cutoff predicts majority datapoints as ones with Upper Direction giving out high amount of False Positives, thereby a very low precision despite good enough recall.

Since $Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$ we have overall Accuracy as 0.5611. We also know that Precision is fraction of total predictions that landed correctly. With $Precision = \frac{TP}{(TP+FP)}$ we avail precision as $557/(557+430) = 0.5643$. As stated earlier, this low precision is since our models predicts majority of observations as Up, thereby false positives or Type I error is high.

We also see recall availed as $Recall = \frac{TP}{TP+FN}$ thereby dependent upon False Negatives or Type II error. Recall = Specificity = 0.92 which is good but at a very high price of bad precision.

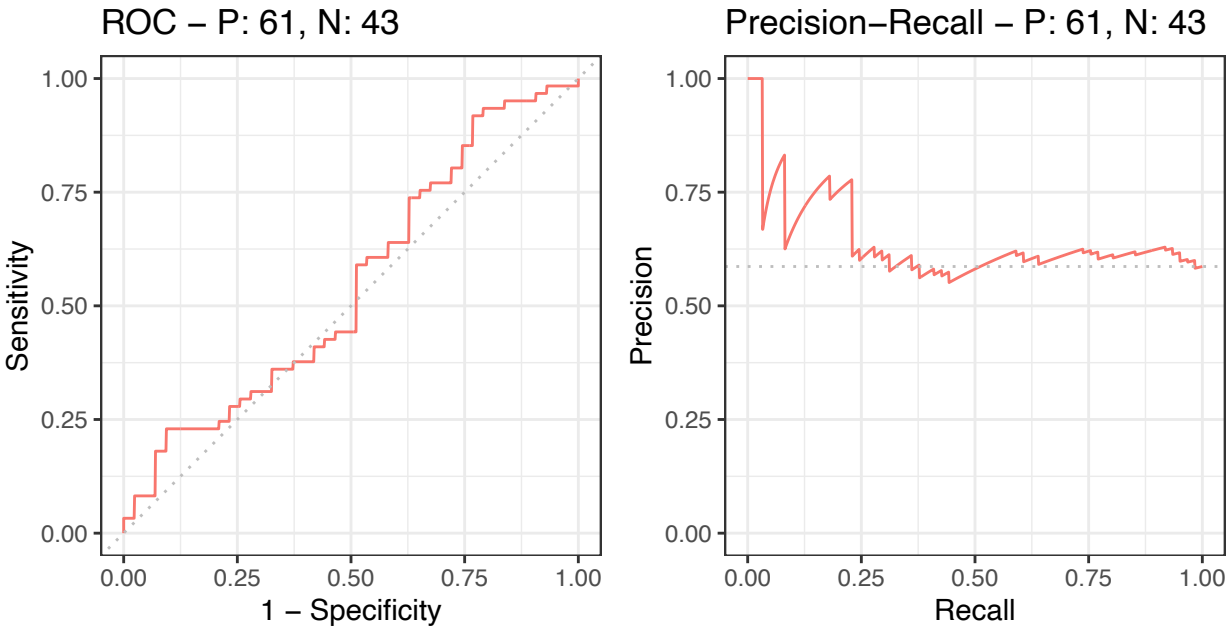
Part (d) Train test split and using Lag2 as independent variable

We observe following confusion matrix and performance metrics for the model at cutoff = 0.5

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  9  5
         1 34 56

               Accuracy : 0.625
                 95% CI : (0.5247, 0.718)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.2439

                  Kappa : 0.1414

 Mcnemar's Test P-Value : 0.00000734

            Sensitivity : 0.20930
```

```
            Specificity : 0.91803
         Pos Pred Value : 0.64286
         Neg Pred Value : 0.62222
             Prevalence : 0.41346
         Detection Rate : 0.08654
   Detection Prevalence : 0.13462
      Balanced Accuracy : 0.56367

       'Positive' Class : 0
```

```
Logit Model regressed using Lag2 gives AUC of  0.546321
```



The problem of overpredicting the class Up persists with overall accuracy = 0.625, recall = 0.918 but a low precision of 0.622

The ROC curve is intersecting the x = y line implying that the model often performs worse than a random classifier implying there is too much uncertainty associated with the predictions of the model at different probability cutoffs with certain instances when model performs worse than a random classifier. This is interesting given the 'statistically significant' variable Lags2 is not sufficient in itself or does not have adequate patterns to be able to segregate the target classes on it's own when tested on the out-of-sample data.

Part (g) Use Knn

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 25 38
         1 18 23

               Accuracy : 0.4615
                 95% CI : (0.3633, 0.562)
```

```
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.99616

                  Kappa : -0.0389

 Mcnemar's Test P-Value : 0.01112

            Sensitivity : 0.5814
            Specificity : 0.3770
         Pos Pred Value : 0.3968
         Neg Pred Value : 0.5610
             Prevalence : 0.4135
         Detection Rate : 0.2404
   Detection Prevalence : 0.6058
      Balanced Accuracy : 0.4792

       'Positive' Class : 0
```

Model performance decreases a bit.

Part (h) KNN with k = 1 (1st nearest neighbour) performs worse than the logistic regression on the out-of-sample data.

We observe a decrease in the accuracy and the model is not being able to pick some signals up relevant to segregating Upwards vs Downwards. Accuracy = 0.4615, Recall = 0.3770 and Precision drops as well due to reduction in False positives at 0.5609

Ideally as we increase k, the model should be able to learn more patterns of Direction at closer (X) values and be able to refine it's solutions

Thus logistic gives best Test error from all models compared so far (based on test accuracy metric)

Part (i) Testing variable combinations with step-wise

```
Start:  AIC=1358.33
Direction ~ Year + Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume

         Df Deviance    AIC
- Year    1   1342.3 1356.3
- Lag3    1   1342.6 1356.6
- Volume  1   1343.1 1357.1
- Lag4    1   1343.5 1357.5
- Lag5    1   1344.0 1358.0
<none>        1342.3 1358.3
- Lag2    1   1344.6 1358.6
- Lag1    1   1346.8 1360.8

Step:  AIC=1356.33
Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume

         Df Deviance    AIC
- Lag3    1   1342.6 1354.6
- Lag4    1   1343.5 1355.5
```

```
- Lag5    1   1344.0 1356.0
<none>        1342.3 1356.3
- Lag2    1   1344.6 1356.6
- Volume  1   1345.1 1357.1
- Lag1    1   1346.9 1358.9

Step:  AIC=1354.61
Direction ~ Lag1 + Lag2 + Lag4 + Lag5 + Volume

         Df Deviance    AIC
- Lag4    1   1343.6 1353.6
- Lag5    1   1344.3 1354.3
<none>        1342.6 1354.6
- Lag2    1   1345.1 1355.1
- Volume  1   1345.2 1355.2
- Lag1    1   1347.3 1357.3

Step:  AIC=1353.64
Direction ~ Lag1 + Lag2 + Lag5 + Volume

         Df Deviance    AIC
- Lag5    1   1345.1 1353.1
<none>        1343.6 1353.6
- Volume  1   1345.9 1353.9
- Lag2    1   1346.0 1354.0
- Lag1    1   1348.0 1356.0

Step:  AIC=1353.14
Direction ~ Lag1 + Lag2 + Volume

         Df Deviance    AIC
- Volume  1   1347.0 1353.0
<none>        1345.1 1353.1
- Lag2    1   1347.8 1353.8
- Lag1    1   1349.4 1355.4

Step:  AIC=1352.96
Direction ~ Lag1 + Lag2

       Df Deviance    AIC
<none>      1347.0 1353.0
- Lag2  1   1350.5 1354.5
- Lag1  1   1350.5 1354.5


Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = train_df[!names(train_df) %in%
    c("Today")])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6149  -1.2565   0.9989   1.0875   1.5330

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  0.21109    0.06456   3.269  0.00108 **
Lag1        -0.05421    0.02886  -1.878  0.06034 .
Lag2         0.05384    0.02905   1.854  0.06379 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1347.0  on 982  degrees of freedom
AIC: 1353

Number of Fisher Scoring iterations: 4


Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  7  8
         1 36 53

               Accuracy : 0.5769
                 95% CI : (0.4761, 0.6732)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.6193

                  Kappa : 0.035

 Mcnemar's Test P-Value : 0.00004693

            Sensitivity : 0.16279
            Specificity : 0.86885
         Pos Pred Value : 0.46667
         Neg Pred Value : 0.59551
             Prevalence : 0.41346
         Detection Rate : 0.06731
   Detection Prevalence : 0.14423
      Balanced Accuracy : 0.51582

       'Positive' Class : 0
```

We avail an accuracy of 0.5769 with a precision of 0.5955 and recall of 0.8688.

Only Lag1 and Lag2 variables are kept in stepwise regression. But Lag1 and Lag2 despite being uncorrelated might have a dependency upon each other. It could be worthwhile to explore the interaction effects associated with them.

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag1:Lag2, family = binomial,
    data = train_df[!names(train_df) %in% c("Today")])

Deviance Residuals:
   Min      1Q  Median      3Q     Max
```

```
-1.573  -1.259   1.003   1.086   1.596

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.211419   0.064589   3.273  0.00106 **
Lag1        -0.051505   0.030727  -1.676  0.09370 .
Lag2         0.053471   0.029193   1.832  0.06700 .
Lag1:Lag2    0.001921   0.007460   0.257  0.79680
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1346.9  on 981  degrees of freedom
AIC: 1354.9

Number of Fisher Scoring iterations: 4


Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  7  8
         1 36 53

               Accuracy : 0.5769
                 95% CI : (0.4761, 0.6732)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.6193

                  Kappa : 0.035

 Mcnemar's Test P-Value : 0.00004693

            Sensitivity : 0.16279
            Specificity : 0.86885
         Pos Pred Value : 0.46667
         Neg Pred Value : 0.59551
             Prevalence : 0.41346
         Detection Rate : 0.06731
   Detection Prevalence : 0.14423
      Balanced Accuracy : 0.51582

       'Positive' Class : 0
```

We see no change in either the predictive power of any significant p-value for any interaction effects in Lag variables. Essentially this means, that of all the information given, Lag1 and Lag2 have the most useful information while their effects are independent of the value of the other variable.

To further improve predictive power, we might need to try other transformations or algorithms.

```
Call:
```

```
glm(formula = Direction ~ Lag1 + Lag2 + I(Lag1^2) + I(Lag2^2),
    family = binomial, data = train_df[!names(train_df) %in%
        c("Today")])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8756  -1.2476   0.9862   1.0961   1.2992

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.180384   0.071513   2.522   0.0117 *
Lag1        -0.053598   0.029552  -1.814   0.0697 .
Lag2         0.062467   0.030664   2.037   0.0416 *
I(Lag1^2)    0.001427   0.005131   0.278   0.7810
I(Lag2^2)    0.004576   0.004746   0.964   0.3350
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1345.7  on 980  degrees of freedom
AIC: 1355.7

Number of Fisher Scoring iterations: 4


Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  7  9
         1 36 52

               Accuracy : 0.5673
                 95% CI : (0.4665, 0.6641)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.6921094

                  Kappa : 0.0168

 Mcnemar's Test P-Value : 0.0001063

            Sensitivity : 0.16279
            Specificity : 0.85246
         Pos Pred Value : 0.43750
         Neg Pred Value : 0.59091
             Prevalence : 0.41346
         Detection Rate : 0.06731
   Detection Prevalence : 0.15385
      Balanced Accuracy : 0.50762

       'Positive' Class : 0
```

Again, we do not see any improvement by including quadratic transformations of pertinent Lag values. Our hope id with KNN and better Ks, the model is also able to map the changes from Up to Down and vice versa better than the k=1 model. Testing out different k-values :

```
Model performance with k = 5


Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 25 35
         1 18 26

               Accuracy : 0.4904
                 95% CI : (0.391, 0.5903)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.98112

                  Kappa : 0.0072

 Mcnemar's Test P-Value : 0.02797

            Sensitivity : 0.5814
            Specificity : 0.4262
         Pos Pred Value : 0.4167
         Neg Pred Value : 0.5909
             Prevalence : 0.4135
         Detection Rate : 0.2404
   Detection Prevalence : 0.5769
      Balanced Accuracy : 0.5038

       'Positive' Class : 0


Model performance with k = 10


Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 23 31
         1 20 30

               Accuracy : 0.5096
                 95% CI : (0.4097, 0.609)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.9540

                  Kappa : 0.0257

 Mcnemar's Test P-Value : 0.1614

            Sensitivity : 0.5349
```

```
            Specificity : 0.4918
         Pos Pred Value : 0.4259
         Neg Pred Value : 0.6000
             Prevalence : 0.4135
         Detection Rate : 0.2212
   Detection Prevalence : 0.5192
      Balanced Accuracy : 0.5133

       'Positive' Class : 0


Model performance with k = 20

Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 24 28
         1 19 33

               Accuracy : 0.5481
                 95% CI : (0.4474, 0.6459)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.8152

                  Kappa : 0.0962

 Mcnemar's Test P-Value : 0.2432

            Sensitivity : 0.5581
            Specificity : 0.5410
         Pos Pred Value : 0.4615
         Neg Pred Value : 0.6346
             Prevalence : 0.4135
         Detection Rate : 0.2308
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.5496

       'Positive' Class : 0


Model performance with k = 50

Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 17 21
         1 26 40

               Accuracy : 0.5481
                 95% CI : (0.4474, 0.6459)
    No Information Rate : 0.5865
```

```
    P-Value [Acc > NIR] : 0.8152

                  Kappa : 0.052

 Mcnemar's Test P-Value : 0.5596

            Sensitivity : 0.3953
            Specificity : 0.6557
         Pos Pred Value : 0.4474
         Neg Pred Value : 0.6061
             Prevalence : 0.4135
         Detection Rate : 0.1635
   Detection Prevalence : 0.3654
      Balanced Accuracy : 0.5255

       'Positive' Class : 0


Model performance with k = 100

Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 10 12
         1 33 49

               Accuracy : 0.5673
                 95% CI : (0.4665, 0.6641)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.692109

                  Kappa : 0.0386

 Mcnemar's Test P-Value : 0.002869

            Sensitivity : 0.23256
            Specificity : 0.80328
         Pos Pred Value : 0.45455
         Neg Pred Value : 0.59756
             Prevalence : 0.41346
         Detection Rate : 0.09615
   Detection Prevalence : 0.21154
      Balanced Accuracy : 0.51792

       'Positive' Class : 0
```

With k = 100 we are able to achieve relatively higher accuracy = 0.5673 with low precision of 0.5975 and recall of 0.8033 But given higher values of k will end up increasing the complexity of the model, we could also avail almost similar test performance at slightly lower values of k, say k = 20.

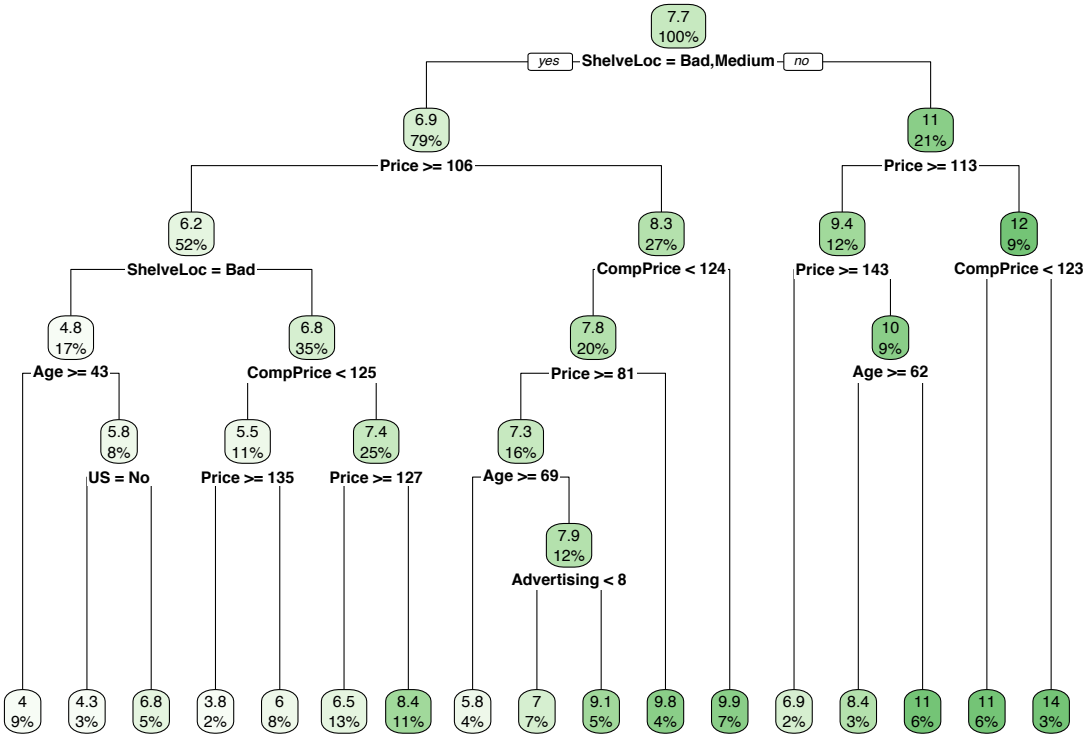
We're able to push for marginally higher Precision compared to logistic regression but overall accuracy and recall are still lacking!

Chapter #8 : Ques 8

Part (a) Descriptive stats of Carseats

```
Number of rows of train and test are : 300  and  100  respectively.
```

Part (b) Decision trees using rpart



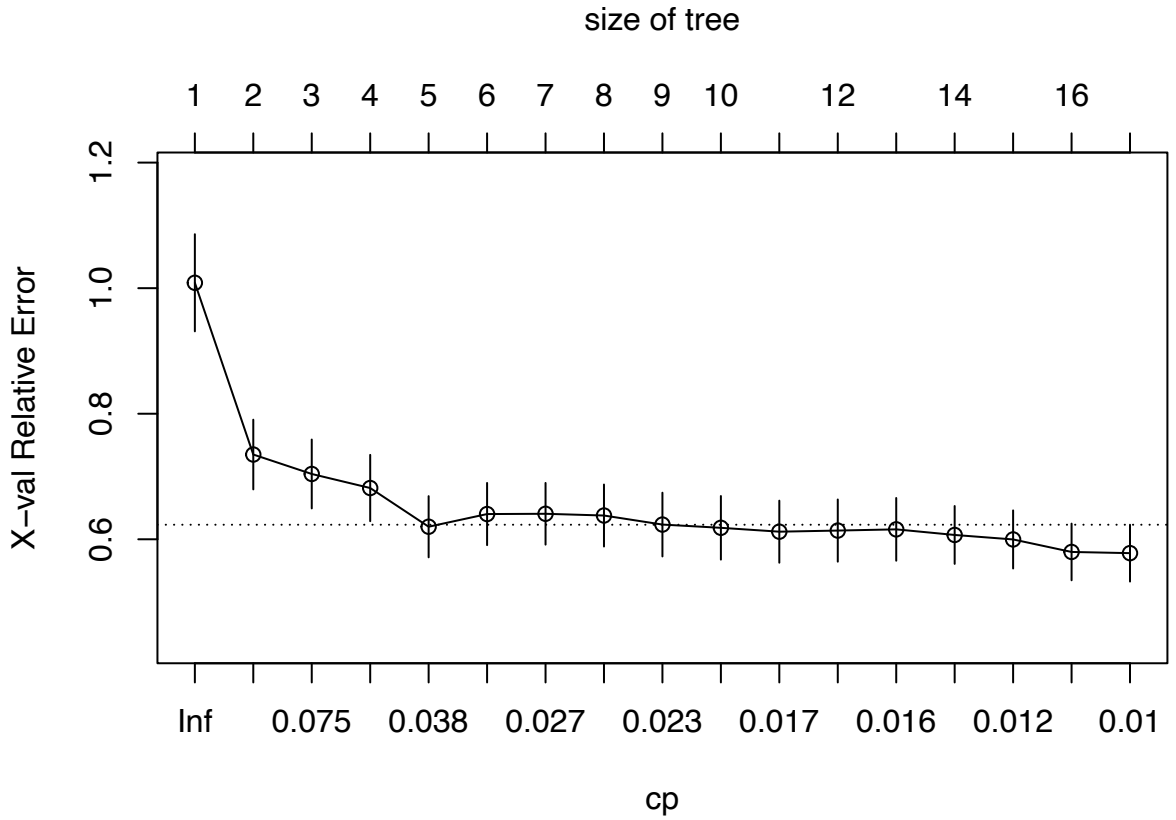
From the tree we can interpret, characteristics conducive to sales are having Good Shelveloc, being more priced higher as well and greater Age and marketing spends. A combination of where these characterstics intersect offers much better value for Sales.

```
Mean Squared Error on the test set is : 4.565464
```

Part (c)

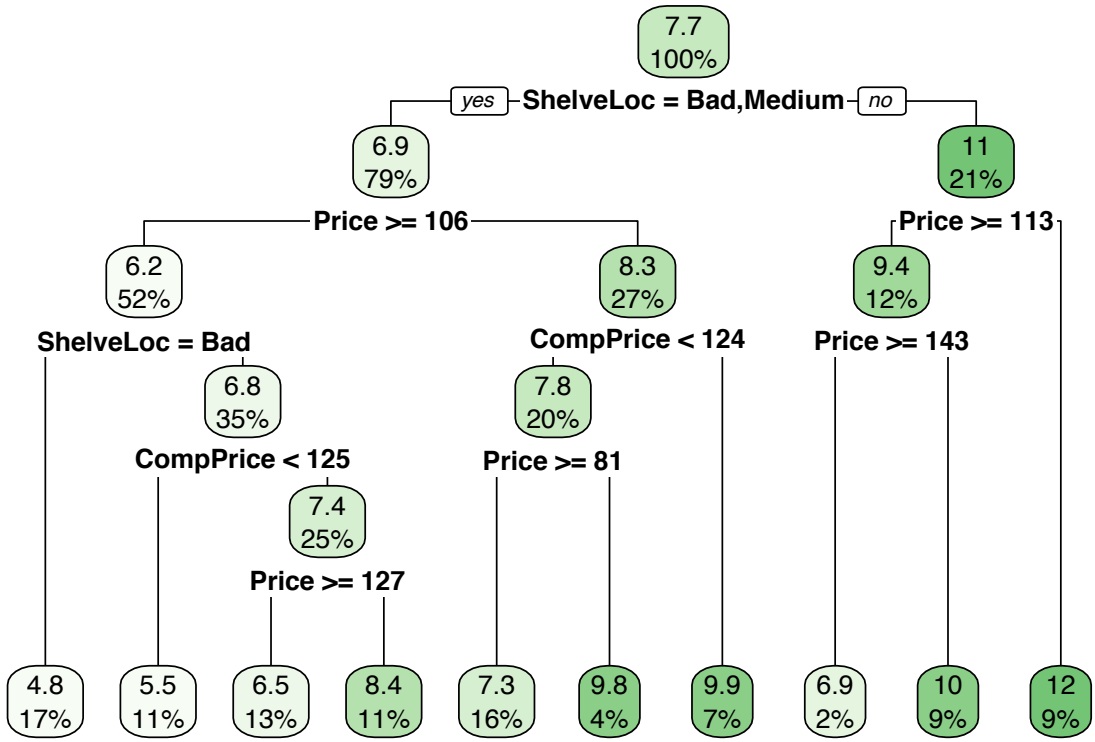
```
Regression tree:
rpart(formula = Sales ~ ., data = train, method = "anova")

Variables actually used in tree construction:
[1] Advertising Age         CompPrice   Price       ShelveLoc   US

Root node error: 2471.2/300 = 8.2374

n= 300

         CP nsplit rel error  xerror     xstd
1  0.273036      0   1.00000 1.00857 0.077370
2  0.099151      1   0.72696 0.73499 0.055536
```

```
3  0.056403      2   0.62781 0.70403 0.054875
4  0.047136      3   0.57141 0.68166 0.052836
5  0.031059      4   0.52427 0.62001 0.048743
6  0.027734      5   0.49321 0.64020 0.049554
7  0.027124      6   0.46548 0.64061 0.049154
8  0.024376      7   0.43836 0.63787 0.049413
9  0.021397      8   0.41398 0.62345 0.050773
10 0.017239      9   0.39258 0.61829 0.050687
11 0.017142     10   0.37534 0.61213 0.049441
12 0.016729     11   0.35820 0.61389 0.049558
13 0.014837     12   0.34147 0.61583 0.049960
14 0.013116     13   0.32664 0.60690 0.046115
15 0.011021     14   0.31352 0.59977 0.046301
16 0.010115     15   0.30250 0.57984 0.045168
17 0.010000     16   0.29239 0.57793 0.045274
```



The printcp and plotcp methods in rpart cross-validate and return the cross-validated relative error with respect to the number of splits and complexity parameter, along with the recommendation (dashed horizontal line) of which minimum error or which split and complexity parameter would work out best for a given tree.

Here ideal cp is 0.02-0.038 with 5-10 splits!



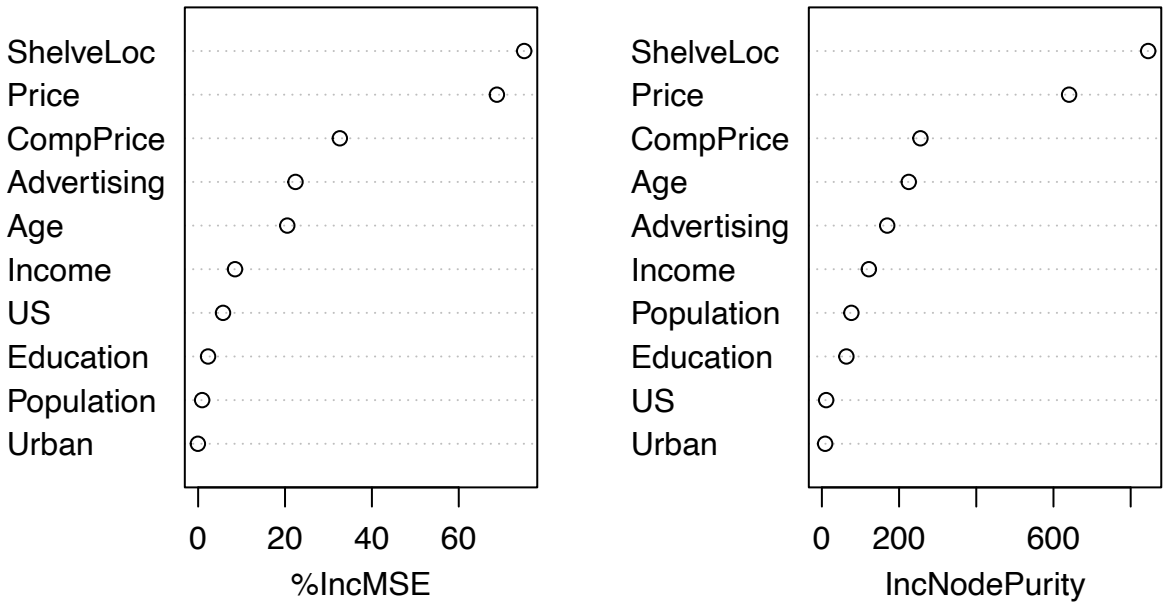
Tree at cp = 0.02

```
Mean Squared Error of pruned tree on the test set is : 4.334698
```

Using a pruned tree with 10 leaf nodes at a complexity parameter of 0.02, we're able to slightly reduce the test set MSE from 4.565 to 4.335. However if we keep on pruning, after certain time, we might underfit the model and MSE might rise again.

Part (d) RF Implementation

```
             %IncMSE IncNodePurity
CompPrice      32.62        255.35
Income          8.49        121.75
Advertising    22.41        169.30
Population      0.92         76.43
Price          68.78        640.50
ShelveLoc      75.07        845.44
Age            20.53        225.10
Education       2.29         63.53
Urban          -0.06          8.52
US              5.73         11.14
```




```
Mean Squared Error of RF with m = 10 on the test set is : 2.169766
```

Similar to the rpart plots, top features are ShelveLoc, Price and CompPrice followed by Age and advertising that were key to certain child nodes in the rpart tree!

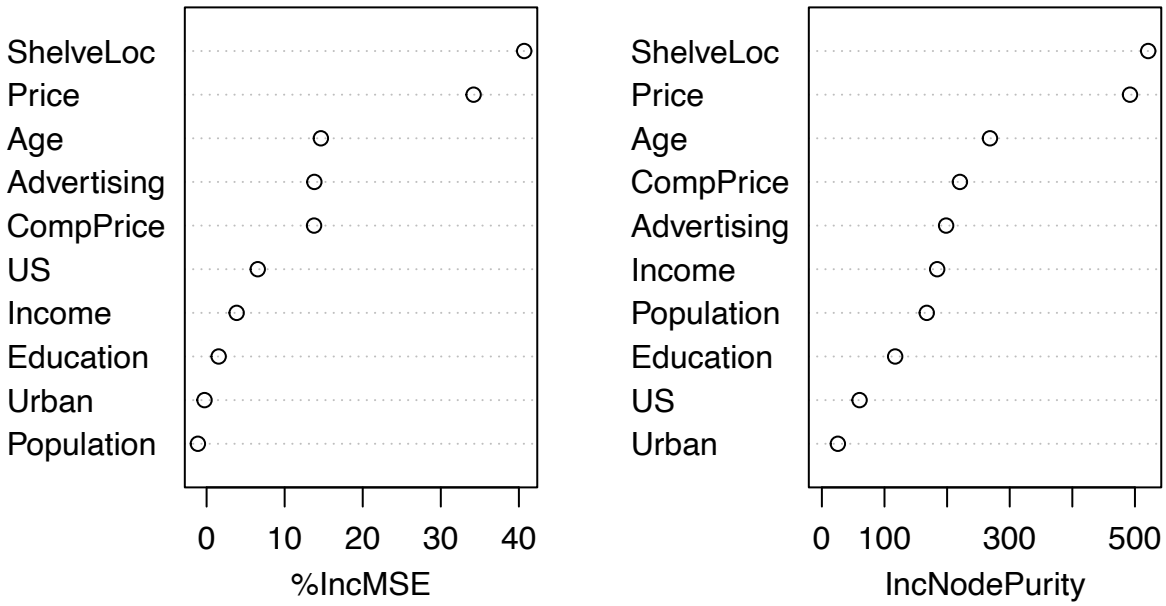
We see RF is able to considerably reduce the out-of-sample MSE (2.17) on account of bagging multiple samples with replacement.

RF with multiple m-values

```
Randomforest with m = 2
```

```
             %IncMSE IncNodePurity
CompPrice      13.77        220.46
Income          3.85        184.18
Advertising    13.80        198.43
Population     -1.12        167.52
Price          34.20        492.13
ShelveLoc      40.68        521.21
Age            14.64        268.48
Education       1.55        117.06
Urban          -0.27         25.51
US              6.55         60.17
```

bagged2

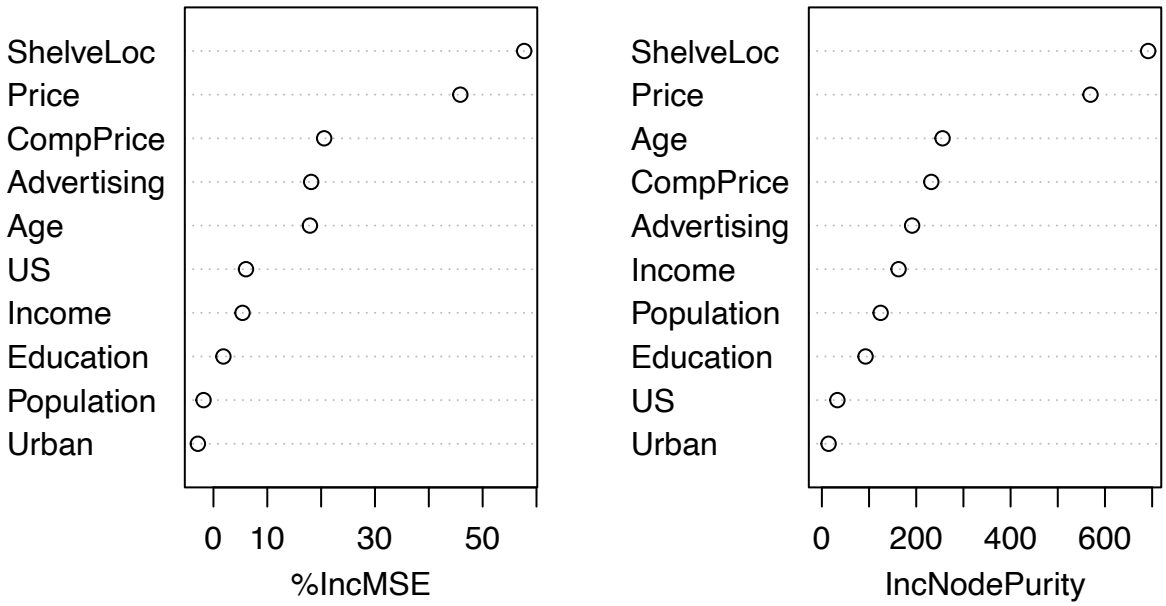



```
Mean Squared Error of RF with m = 2 on the test set is : 3.069031
```

```
Randomforest with m = 4
```

	%IncMSE	IncNodePurity
CompPrice	21	232
Income	5	163
Advertising	18	191
Population	-2	125
Price	46	569
ShelveLoc	58	692
Age	18	256
Education	2	93
Urban	-3	14
US	6	33

bagged4

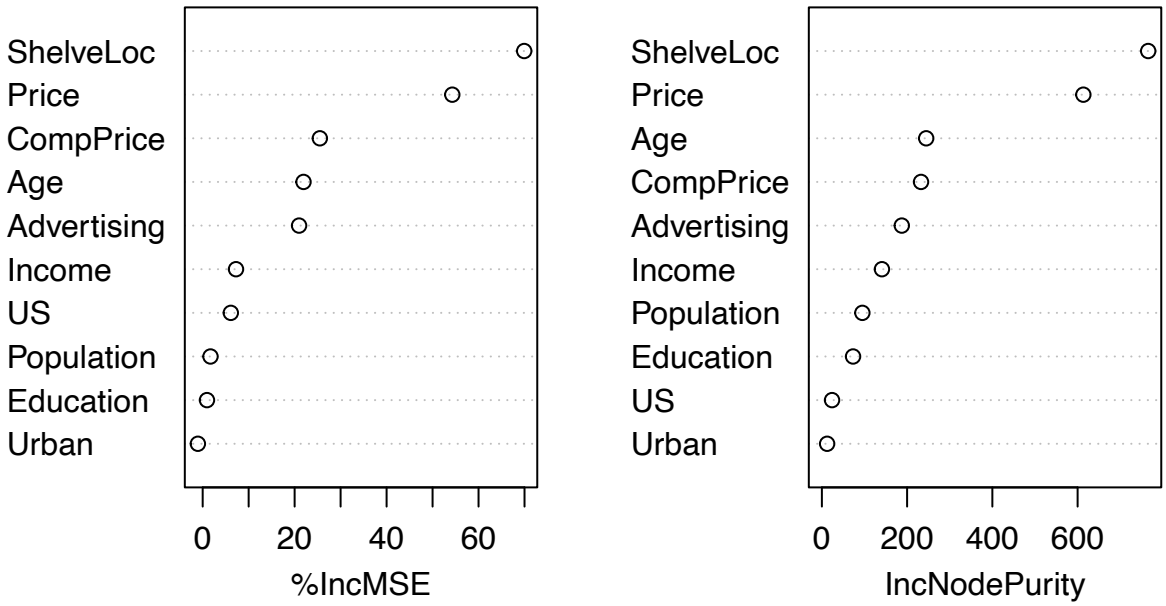



```
Mean Squared Error of RF with m = 4 on the test set is : 2.449714
```

```
Randomforest with m = 6
```

	%IncMSE	IncNodePurity
CompPrice	25.49	232.64
Income	7.25	140.89
Advertising	20.97	187.50
Population	1.70	95.00
Price	54.29	613.45
ShelveLoc	69.93	765.67
Age	21.93	244.80
Education	0.94	73.15
Urban	-1.04	12.37
US	6.12	23.87

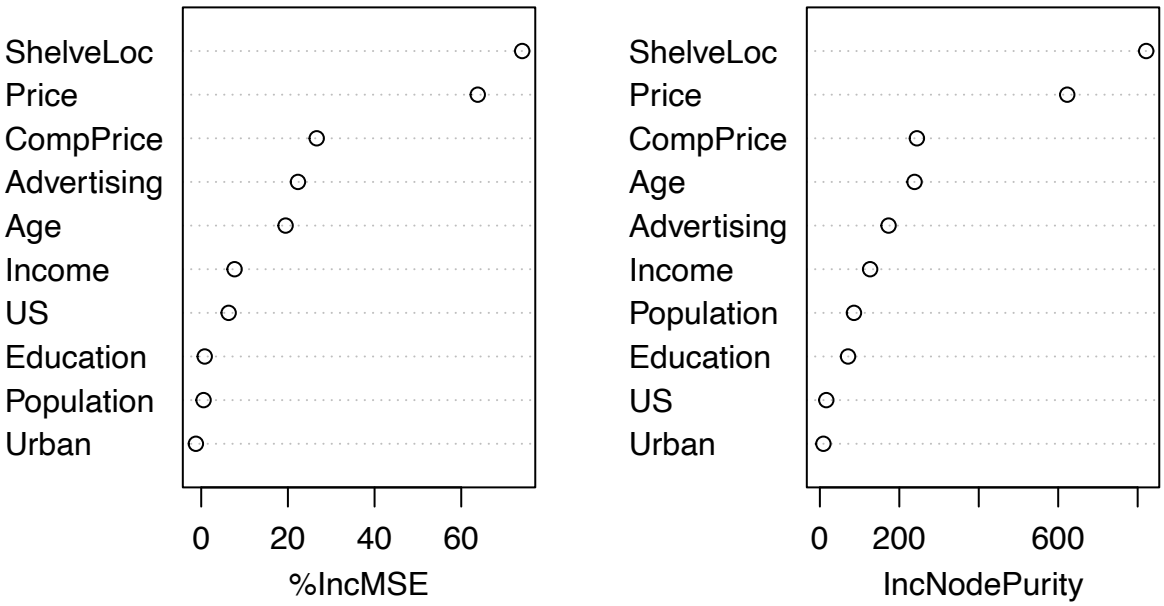
bagged6




```
Mean Squared Error of RF with m = 6 on the test set is : 2.244454
```

```
Randomforest with m = 8
```

	%IncMSE	IncNodePurity
CompPrice	26.65	244.26
Income	7.69	126.47
Advertising	22.32	172.93
Population	0.54	85.84
Price	63.81	622.71
ShelveLoc	74.07	821.53
Age	19.47	238.32
Education	0.81	70.93
Urban	-1.24	8.91
US	6.33	16.19




```
Mean Squared Error of RF with m = 8 on the test set is : 2.257829
```

The top 2 variables are constant for all values of m, viz. Shelveloc and Price in order. The next three important variables are Price, Age and Advertising as seen above, but the order of importance changes with different values of m.

MSEs for different m are comparable but none of the m values are able to beat the bagging benchmark test MSE.

As m keeps on increasing, the test MSE in this case keeps on decreasing with m = 6 giving best MSE as 2.24 amongst the ones tried above.

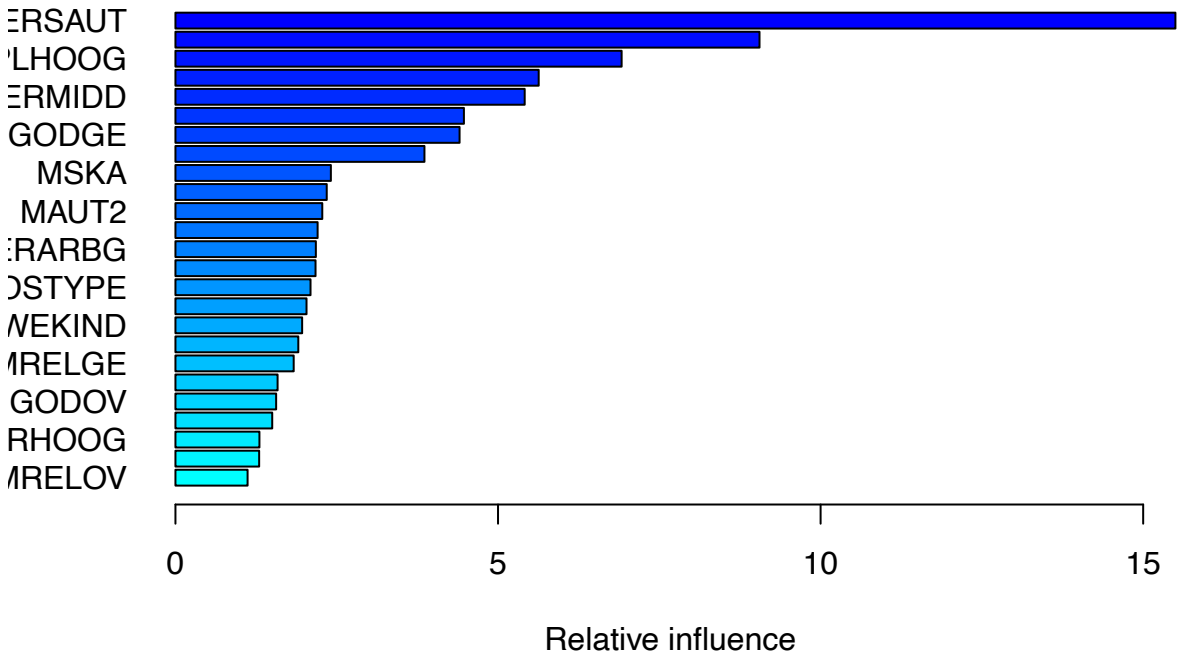
MSE further decreases going from m=8 to m=10

Chapter #8 : Ques 11

Part (a) Load data

```
Number of rows in train and test are :  1000  and  4822  respectively.
```

Part (b) Boosting implementation



	var	rel.inf
PPERSAUT	PPERSAUT	15.49989017
MKOOPKLA	MKOOPKLA	9.05289495
MOPLHOOG	MOPLHOOG	6.91606553
PBRAND	PBRAND	5.63022222
MBERMIDD	MBERMIDD	5.41325272
MINK3045	MINK3045	4.47115058
MGODGE	MGODGE	4.40406789
ABRAND	ABRAND	3.85938452
MSKA	MSKA	2.40898085
MSKC	MSKC	2.34487132
MAUT2	MAUT2	2.27582500
PWAPART	PWAPART	2.20354003
MBERARBG	MBERARBG	2.17655343
MAUT1	MAUT1	2.17159225
MOSTYPE	MOSTYPE	2.09353380
MGODPR	MGODPR	2.03188459
MFWEKIND	MFWEKIND	1.96373482
MINKGEM	MINKGEM	1.90443015
MRELGE	MRELGE	1.83283054
MAUT0	MAUT0	1.58329901
MGODOV	MGODOV	1.56072741
PBYSTAND	PBYSTAND	1.50017494
MBERHOOG	MBERHOOG	1.30158671
MSKB1	MSKB1	1.29749550
MRELOV	MRELOV	1.11833559
MFGEKIND	MFGEKIND	1.10990180
MINK7512	MINK7512	1.02647138
MHKOOP	MHKOOP	1.02151649
MGODRK	MGODRK	0.97254661
MINKM30	MINKM30	0.79601630
MHHUUR	MHHUUR	0.66284922

MOPLMIDD	MOPLMIDD	0.64335880
MBERBOER	MBERBOER	0.56820183
PLEVEN	PLEVEN	0.54022775
MINK4575	MINK4575	0.48768588
MGEMOMV	MGEMOMV	0.46433447
MSKD	MSKD	0.46370647
MGEMLEEF	MGEMLEEF	0.46278608
MFALLEEN	MFALLEEN	0.45131665
PMOTSCO	PMOTSCO	0.40617001
MZPART	MZPART	0.39133647
MZFONDS	MZFONDS	0.35913399
MBERARBO	MBERARBO	0.34151150
APERSAUT	APERSAUT	0.34037181
MOSHOOFD	MOSHOOFD	0.33134723
MINK123M	MINK123M	0.31807358
MSKB2	MSKB2	0.28020970
MRELSA	MRELSA	0.24811103
MOPLLAAG	MOPLLAAG	0.20903129
MBERZELF	MBERZELF	0.08745917
MAANTHUI	MAANTHUI	0.00000000
PWABEDR	PWABEDR	0.00000000
PWALAND	PWALAND	0.00000000
PBESAUT	PBESAUT	0.00000000
PVRAAUT	PVRAAUT	0.00000000
PAANHANG	PAANHANG	0.00000000
PTRACTOR	PTRACTOR	0.00000000
PWERKT	PWERKT	0.00000000
PBROM	PBROM	0.00000000
PPERSONG	PPERSONG	0.00000000
PGEZONG	PGEZONG	0.00000000
PWAOREG	PWAOREG	0.00000000
PZEILPL	PZEILPL	0.00000000
PPLEZIER	PPLEZIER	0.00000000
PFIETS	PFIETS	0.00000000
PINBOED	PINBOED	0.00000000
AWAPART	AWAPART	0.00000000
AWABEDR	AWABEDR	0.00000000
AWALAND	AWALAND	0.00000000
ABESAUT	ABESAUT	0.00000000
AMOTSCO	AMOTSCO	0.00000000
AVRAAUT	AVRAAUT	0.00000000
AAANHANG	AAANHANG	0.00000000
ATRACTOR	ATRACTOR	0.00000000
AWERKT	AWERKT	0.00000000
ABROM	ABROM	0.00000000
ALEVEN	ALEVEN	0.00000000
APERSONG	APERSONG	0.00000000
AGEZONG	AGEZONG	0.00000000
AWAOREG	AWAOREG	0.00000000
AZEILPL	AZEILPL	0.00000000
APLEZIER	APLEZIER	0.00000000
AFIETS	AFIETS	0.00000000
AINBOED	AINBOED	0.00000000
ABYSTAND	ABYSTAND	0.00000000

Few top variables explaining dependent variable best are : PPERSAUT > MKOOPKLA > MOPLHOOG > PBRAND > MBERMIDD at the given hyperparameters

Part (c) Test performance metrics at cutoff 0.2

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4413  255
         1  120   34

               Accuracy : 0.9222
                 95% CI : (0.9143, 0.9296)
    No Information Rate : 0.9401
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1167

 Mcnemar's Test P-Value : 0.000000000004525

            Sensitivity : 0.9735
            Specificity : 0.1176
         Pos Pred Value : 0.9454
         Neg Pred Value : 0.2208
             Prevalence : 0.9401
         Detection Rate : 0.9152
   Detection Prevalence : 0.9681
      Balanced Accuracy : 0.5456

       'Positive' Class : 0


Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4183  231
         1  350   58

               Accuracy : 0.8795
                 95% CI : (0.87, 0.8886)
    No Information Rate : 0.9401
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1035

 Mcnemar's Test P-Value : 0.0000009807

            Sensitivity : 0.9228
            Specificity : 0.2007
         Pos Pred Value : 0.9477
         Neg Pred Value : 0.1422
```

```
            Prevalence : 0.9401
        Detection Rate : 0.8675
  Detection Prevalence : 0.9154
     Balanced Accuracy : 0.5617

      'Positive' Class : 0
```

Proportion of people predicted to purchase who actually made one, equal model precision where $Precision = \frac{TP}{(TP+FP)} = \frac{34}{(34+120)} = 0.2207$

Even though overall accuracy is high, we see very low precision and recall metrics for this model. This is predominantly due to the high class imbalance present in the data. In addition to this, the amount of data used in train is very less and might not have captured enough signal to be able to identify class 1 from class 0. Even if the model recommends majority of observations as predicted purchase, the model hits decent accuracy but low values of precision and recall suggest model hasn't learnt segregation of categories well enough.

Despite this, a random guess would be correct 348/(348+5474) = 5.9% times whereas using boosting it's 22%. Hence it's 4x improvement over a random classifier.

Comparing with logit, we see precision as 58/(58+350) = 14.2% suggesting the amount of improvement that Boosting brings out is significant.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4183  231
         1  350   58

               Accuracy : 0.8795
                 95% CI : (0.87, 0.8886)
    No Information Rate : 0.9401
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1035

 Mcnemar's Test P-Value : 0.0000009807

            Sensitivity : 0.9228
            Specificity : 0.2007
         Pos Pred Value : 0.9477
         Neg Pred Value : 0.1422
             Prevalence : 0.9401
         Detection Rate : 0.8675
   Detection Prevalence : 0.9154
      Balanced Accuracy : 0.5617

       'Positive' Class : 0
```

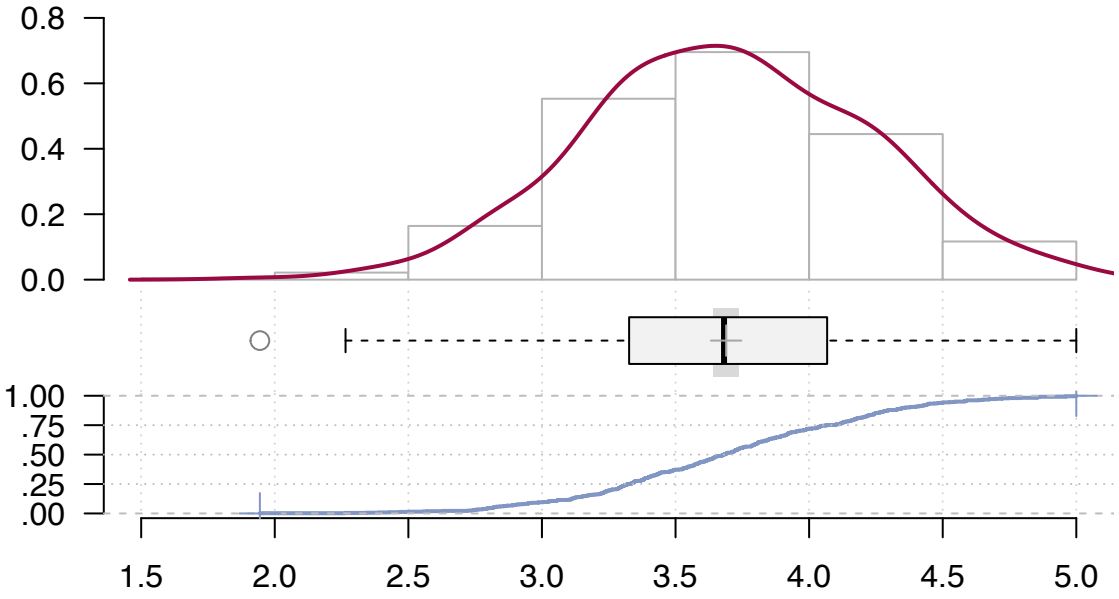
We see logistic model offers a better Recall of 0.20 which is a lift from 0.11 of the boosting model. However, the precision further drops since we're recommending more number of observations herein and making a lot more of Flase positives or type I errors. Precision drops from 22% to 14%, meaning still a 2.2x increase in predictability than a random choice.

Problem #1 : Beauty Pays!

Part (a) Descriptive Stats

```
------------------------------------------------------------------------------
Describe beauty (data.table, data.frame):

data frame: 463 obs. of  6 variables
        463 complete cases (100.0%)

  Nr  ColName      Class    NAs  Levels
  1   CourseEvals  numeric  .
  2   BeautyScore  numeric  .
  3   female       integer  .
  4   lower        integer  .
  5   nonenglish   integer  .
  6   tenuretrack  integer  .


------------------------------------------------------------------------------
1 - CourseEvals (numeric)

    length         n       NAs    unique        0s       mean     meanCI'
       463       463         0       462         0   3.689415   3.641383
             100.0%      0.0%               0.0%              3.737446

       .05       .10       .25    median       .75        .90        .95
  2.819824  3.024451  3.326226  3.681537  4.067423   4.360165   4.545751

     range        sd     vcoef       mad       IQR       skew       kurt
  3.055757  0.525928  0.142551  0.541740  0.741197  -0.038030  -0.165956

lowest : 1.944243, 2.264917, 2.340655, 2.389286, 2.428782
highest: 4.856975, 4.860266, 4.959779, 4.96073, 5.0 (2)

' 95%-CI (classic)
```

1 – CourseEvals (numeric)



---

2 - BeautyScore (numeric)

length	n	NAs	unique	0s	mean’
463	463	0	94	0	-0.0883490
	100.0%	0.0%		0.0%	

.05	.10	.25	median	.75	.90
-1.1679070	-1.0707340	-0.7446180	-0.1563634	0.4572534	1.0659060

range	sd	vcoef	mad	IQR	skew
3.4205170	0.7886476	-8.9265005	0.8723329	1.2018714	0.5124670

meanCI
-0.1603735
-0.0163246

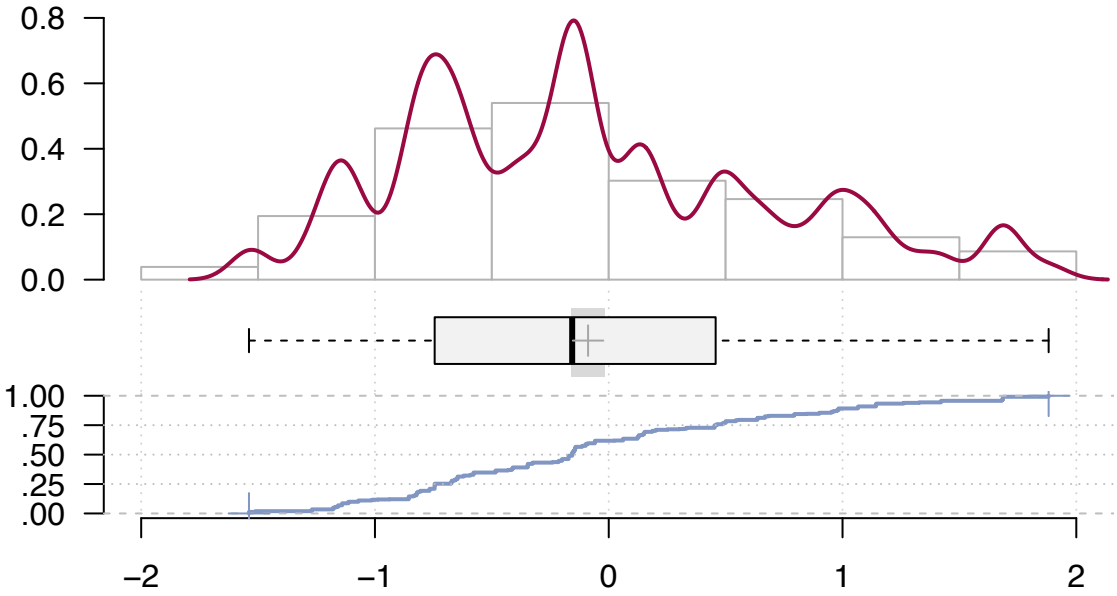
.95
1.4214450

kurt
-0.4048811

lowest : -1.5388430 (6), -1.5112680 (3), -1.2699750 (7), -1.1787380 (7), -1.1679070 (3)
highest: 1.681103 (3), 1.684802 (3), 1.685985 (8), 1.687167 (2), 1.881674 (4)

’ 95%-CI (classic)

2 – BeautyScore (numeric)

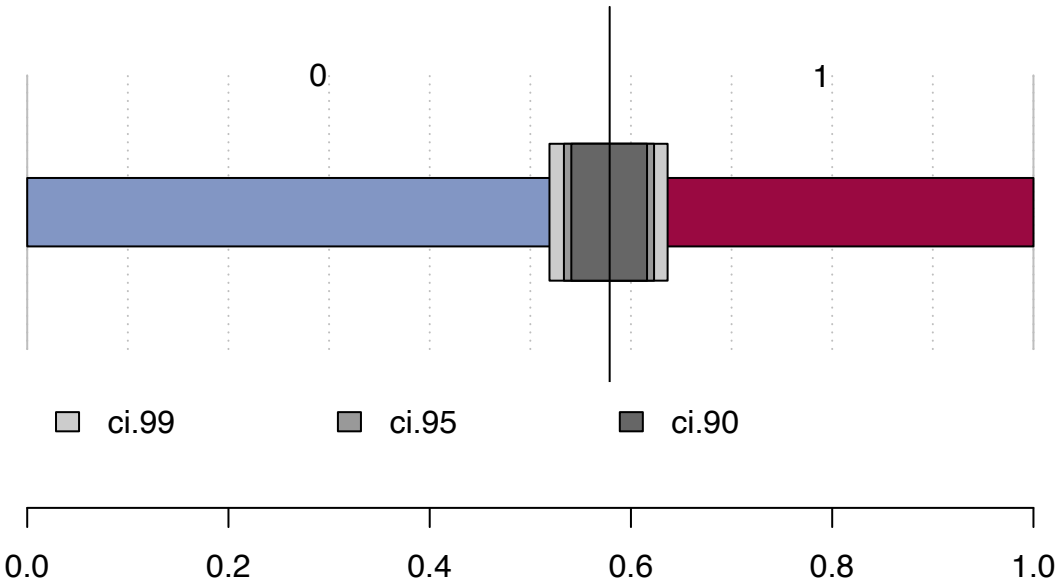


---

3 - female (integer - dichotomous)

```
  length      n    NAs unique
     463    463      0      2
          100.0%   0.0%

   freq   perc  lci.95  uci.95'
0   268  57.9%   53.3%   62.3%
1   195  42.1%   37.7%   46.7%

' 95%-CI (Wilson)
```

3 – female (integer – dichotomous)



---

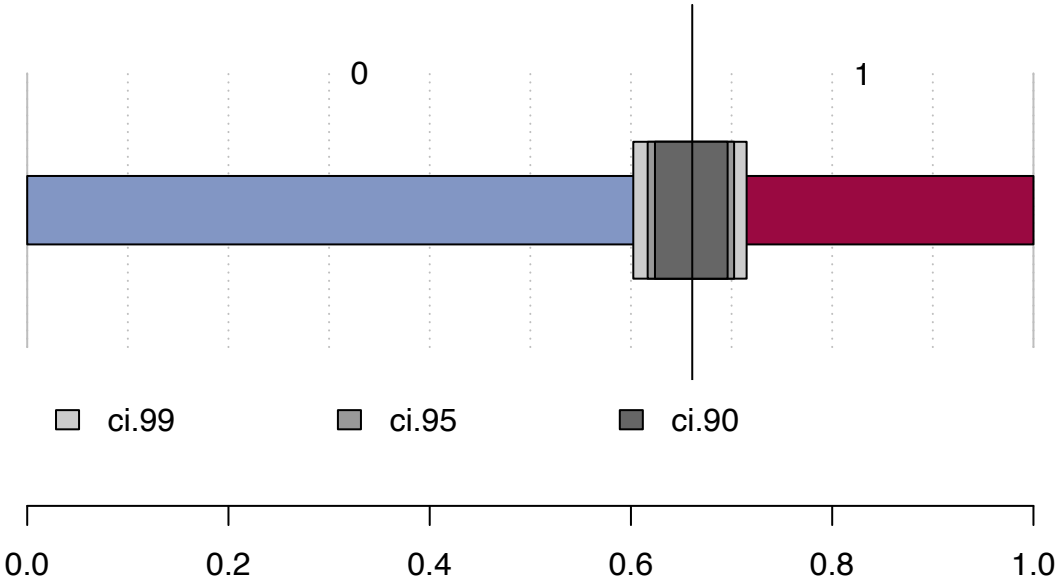
4 - lower (integer - dichotomous)

length	n	NAs	unique
463	463	0	2
	100.0%	0.0%	

	freq	perc	lci.95	uci.95’
0	306	66.1%	61.7%	70.3%
1	157	33.9%	29.7%	38.3%

’ 95%-CI (Wilson)

4 – lower (integer – dichotomous)

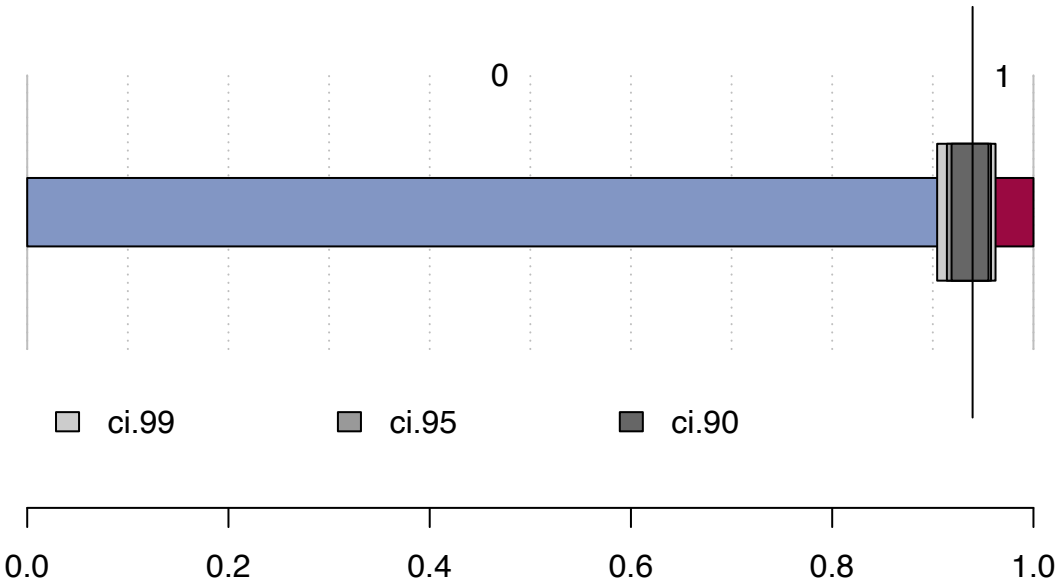


---

5 - nonenglish (integer - dichotomous)

```
  length      n    NAs unique
     463    463      0      2
          100.0%   0.0%

   freq   perc  lci.95  uci.95'
0   435  94.0%   91.4%   95.8%
1    28   6.0%    4.2%    8.6%

' 95%-CI (Wilson)
```

5 – nonenglish (integer – dichotomous)

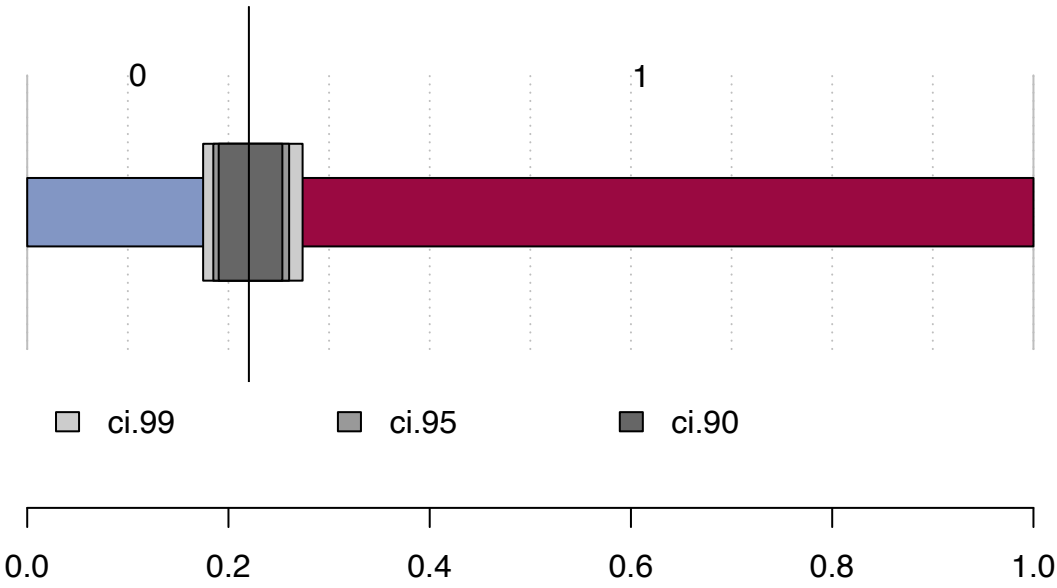


---

6 - tenuretrack (integer - dichotomous)

```
  length      n    NAs unique
     463    463      0      2
          100.0%   0.0%

   freq   perc  lci.95  uci.95'
0   102  22.0%   18.5%   26.0%
1   361  78.0%   74.0%   81.5%

' 95%-CI (Wilson)
```

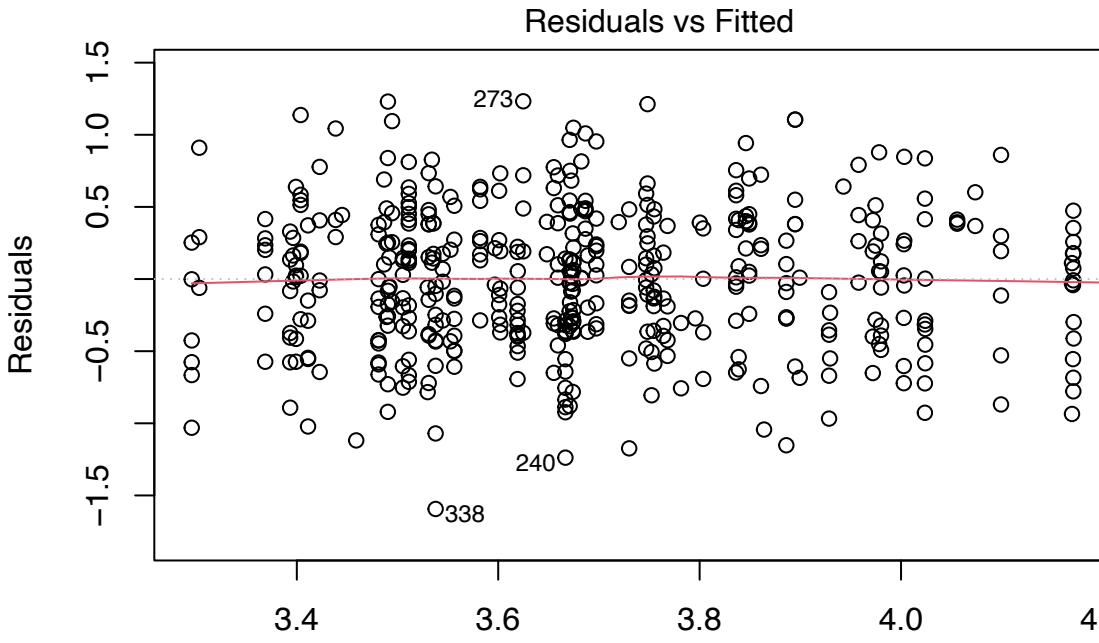
6 – tenuretrack (integer – dichotomous)



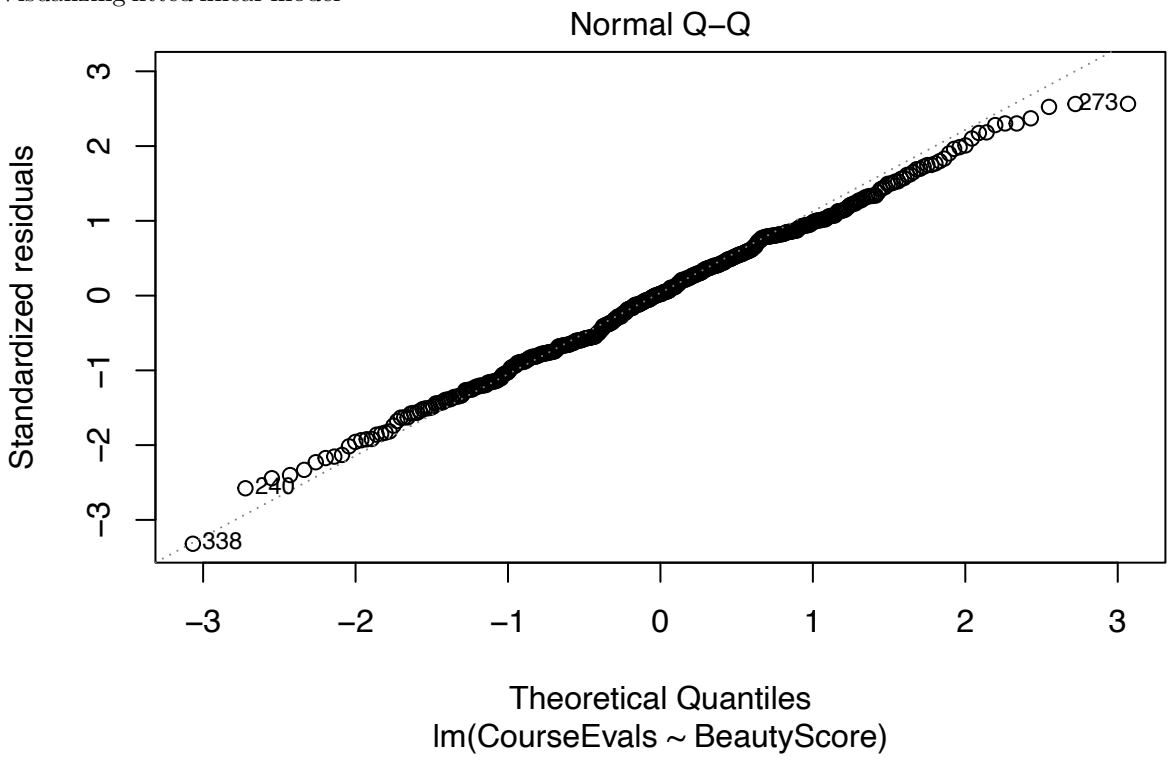
Model course evals with beauty score

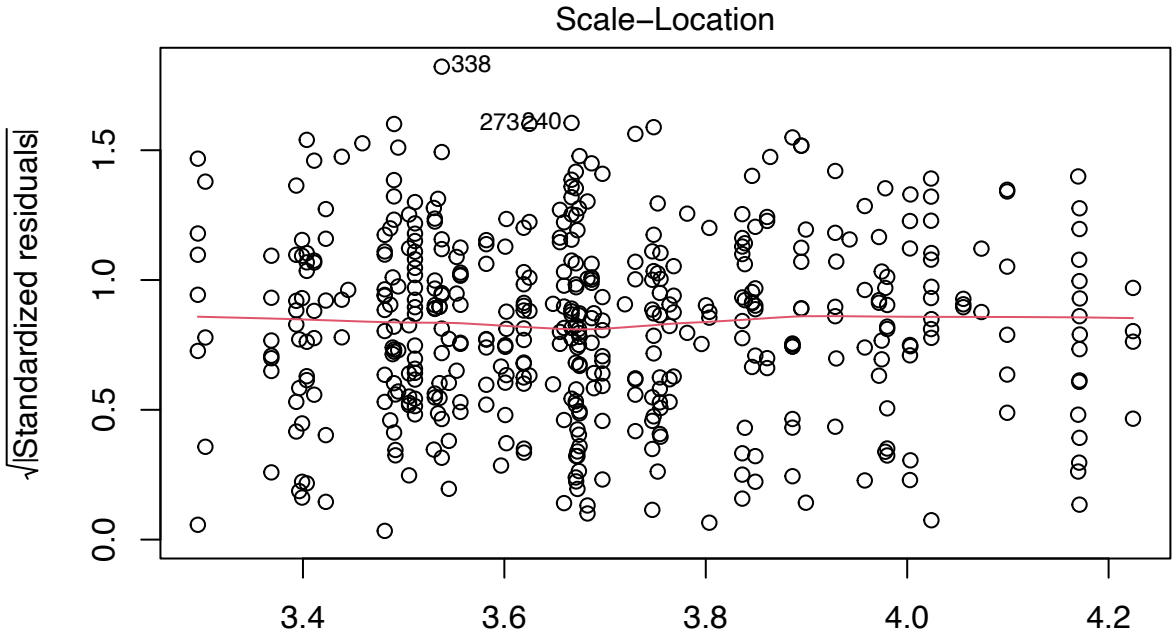
```
Call:
lm(formula = CourseEvals ~ BeautyScore, data = beauty)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5936 -0.3346  0.0097  0.3702  1.2321

Coefficients:
            Estimate Std. Error t value            Pr(>|t|)
(Intercept)  3.71340    0.02249 165.119 <0.0000000000000002 ***
BeautyScore  0.27148    0.02837   9.569 <0.0000000000000002 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.4809 on 461 degrees of freedom
Multiple R-squared:  0.1657,    Adjusted R-squared:  0.1639
F-statistic: 91.57 on 1 and 461 DF,  p-value: < 0.00000000000000022
```

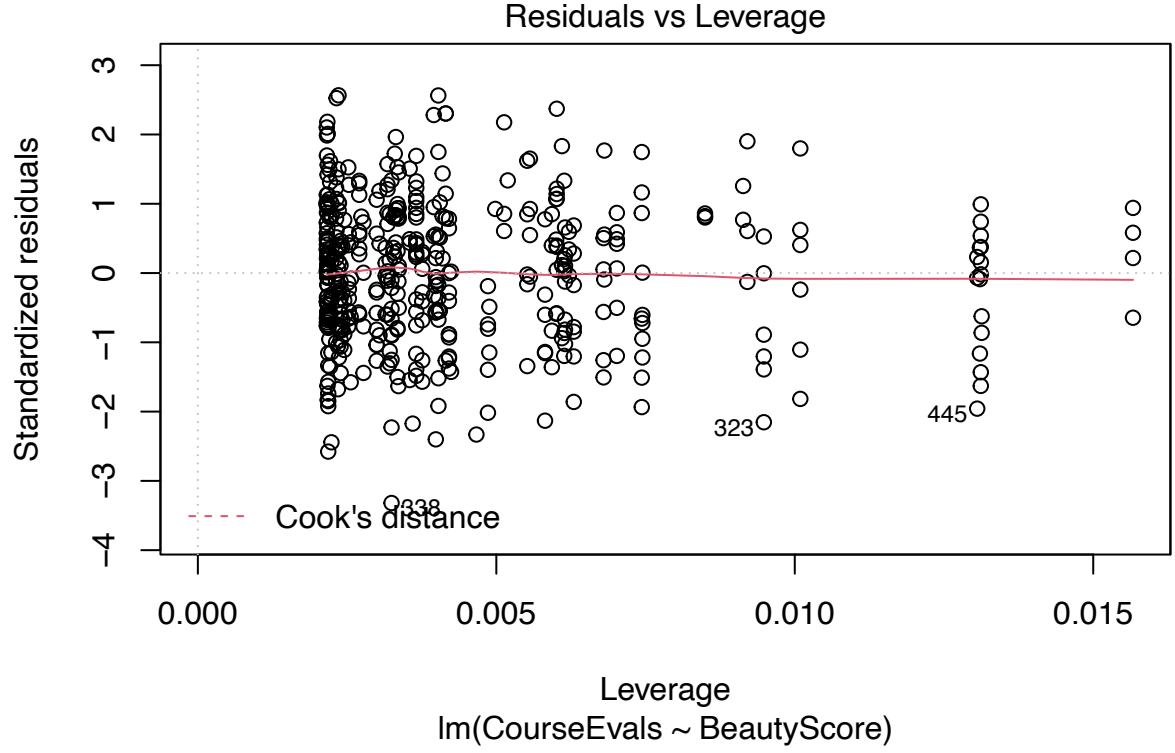
Univariate linear regression points towards very strong positive association between beauty metric and the course evaluations. However beauty alone does not explain much of the variance of CourseEvals since we see a poor R-sq of 0.1657. This means we need to have a multivariate analysis of other determinant variables that might also be predictive of CourseEvals and gauge the coefficient of BeautyScore in such a setting, keeping all other determinants constant.



Visualizing fitted linear model





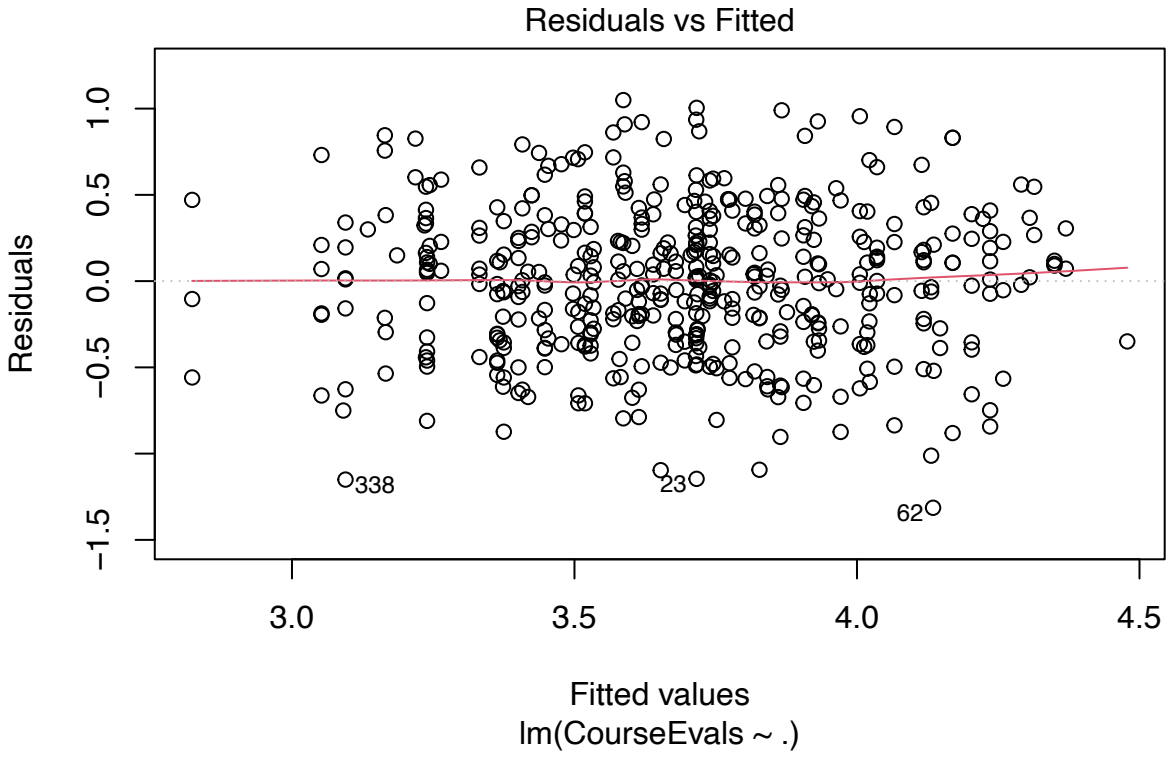


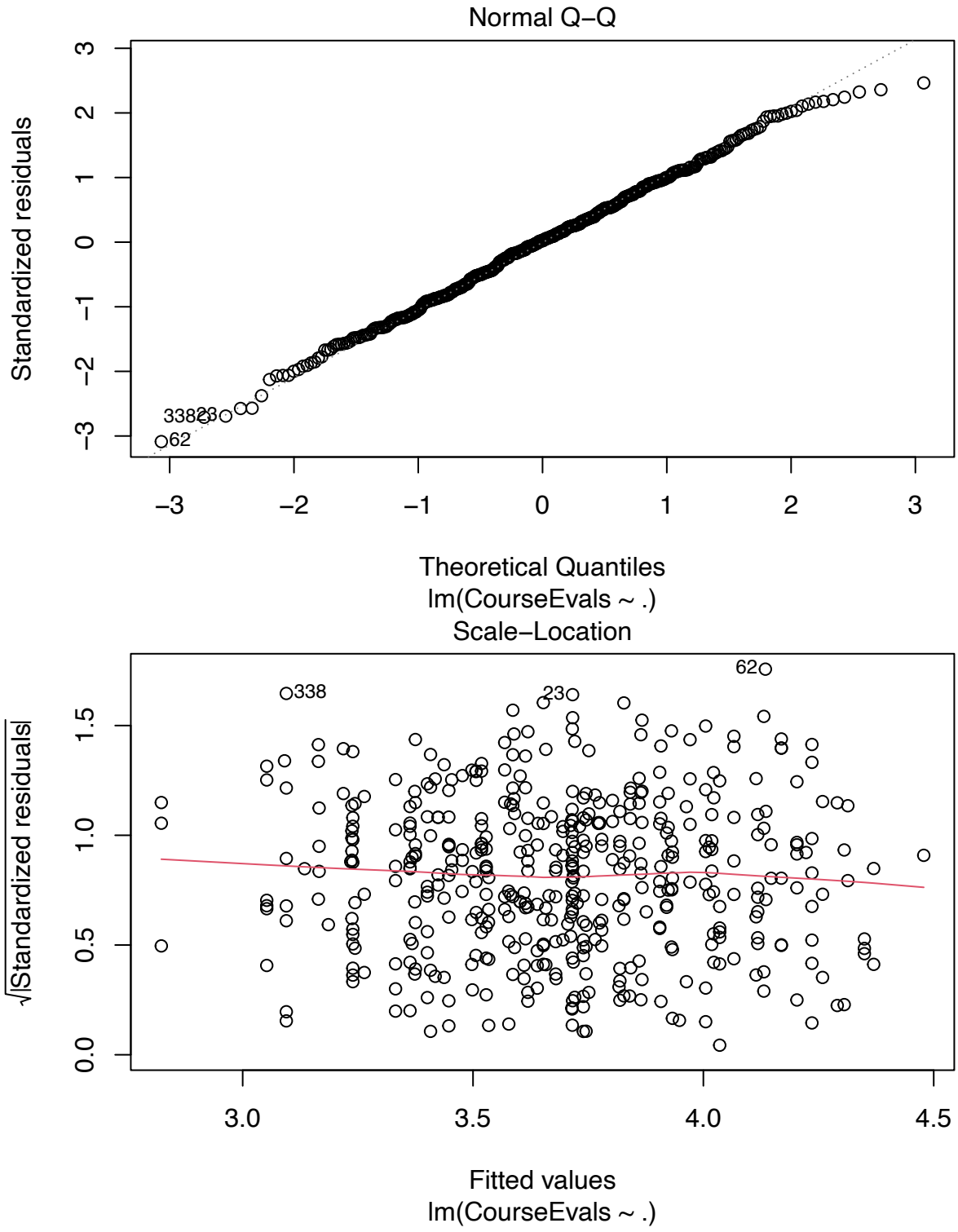
Model course evals with all covariates

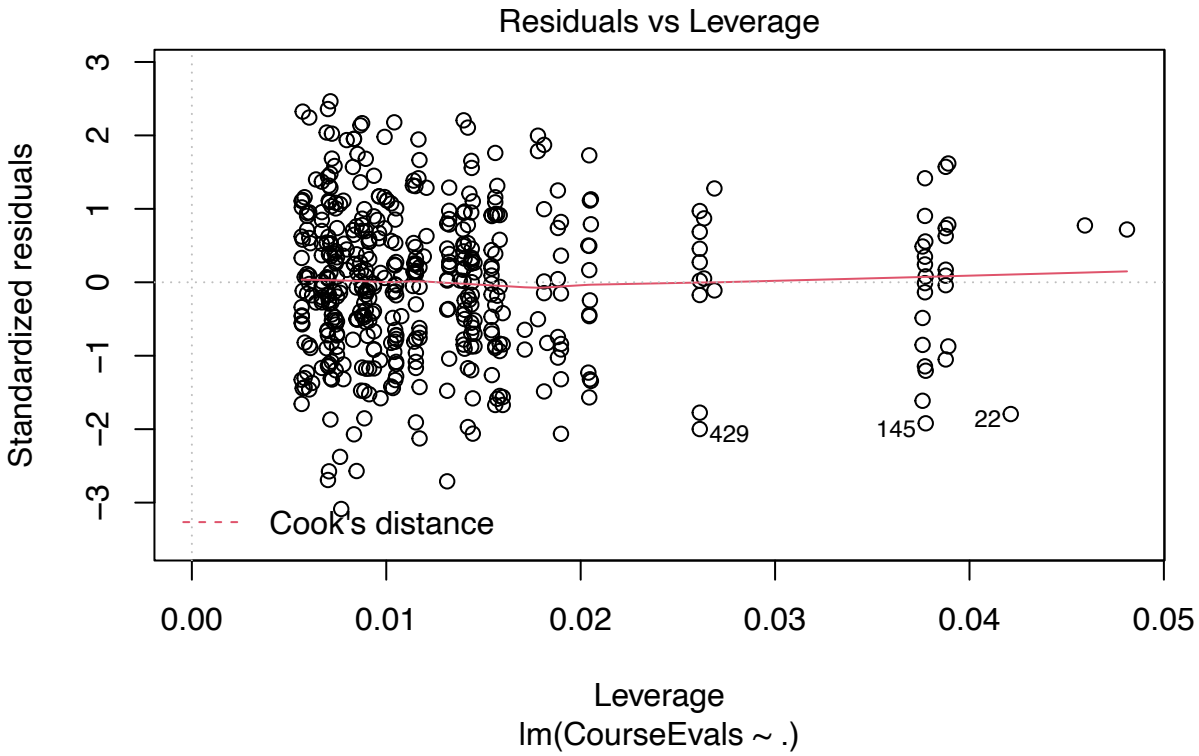
```
Call:
lm(formula = CourseEvals ~ ., data = beauty)

Residuals:
```

```
     Min       1Q   Median       3Q      Max
-1.31385 -0.30202  0.01011  0.29815  1.04929

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept)  4.06542    0.05145  79.020 < 0.0000000000000002 ***
BeautyScore  0.30415    0.02543  11.959 < 0.0000000000000002 ***
female      -0.33199    0.04075  -8.146  0.00000000000000362 ***
lower       -0.34255    0.04282  -7.999  0.00000000000001038 ***
nonenglish  -0.25808    0.08478  -3.044              0.00247 **
tenuretrack -0.09945    0.04888  -2.035              0.04245 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4273 on 457 degrees of freedom
Multiple R-squared:  0.3471,    Adjusted R-squared:  0.3399
F-statistic: 48.58 on 5 and 457 DF,  p-value: < 0.00000000000000022
```

Visualizing fitted linear model




Normal Q–Q
Standardized residuals
Theoretical Quantiles
lm(CourseEvals ~ .)
Scale–Location
√|Standardized residuals|
Fitted values
lm(CourseEvals ~ .)
338
23
62



With the multivariate analysis we see all variables have significant effect on CourseEvals with beauty being one of them as well. The directionality of beauty variable co-efficient does not change sign means given other determinants such as gender, non-english, lower and tenuretrack kept constant, beauty score still has significant association with course evaluations.

Part (b) While even the multivariate analysis gives out significant betas for Beauty Score, since we can't attribute association to causation we can really not say anything about whether higher beauty score is causing higher course evaluations leading to discrimination. It may very well happen that some confounding variable unused in the analysis is the reason for the strong direct relationship with BeautyScore.

Case in point, it could be that the calculation of beauty score leverages how consistent or structured people are in their lifestyle and maintaining their appearance is only a part of their behaviour in general. Regression model using such a metric while shows direct relationship with beauty score, in reality could mean direct relationship with having structured / disciplined lifestyle.

This is where Dr. Hamermesh is suggesting that disentangling this relationship as being driven by positive confounding factors such as productivity or taking variable at face value and assuming it to be driven by discrimination based on beauty perception is probably impossible!

Problem #2 : Housing Price Structure

Summarize raw data

```
      Home            Nbhd          Offers          SqFt
 Min.   :  1.00   Min.   :1.000   Min.   :1.000   Min.   :1450
 1st Qu.: 32.75   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1880
 Median : 64.50   Median :2.000   Median :3.000   Median :2000
 Mean   : 64.50   Mean   :1.961   Mean   :2.578   Mean   :2001
 3rd Qu.: 96.25   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2140
```

```
 Max.   :128.00   Max.   :3.000   Max.   :6.000   Max.   :2590
   Brick              Bedrooms       Bathrooms         Price
 Length:128         Min.   :2.000   Min.   :2.000   Min.   : 69100
 Class :character   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:111325
 Mode  :character   Median :3.000   Median :2.000   Median :125950
                    Mean   :3.023   Mean   :2.445   Mean   :130427
                    3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:148250
                    Max.   :5.000   Max.   :4.000   Max.   :211200
```

Part (1) Model with all covariates

```
Call:
lm(formula = Price ~ ., data = housing)

Residuals:
     Min       1Q   Median       3Q      Max
-27337.3  -6549.5    -41.7   5803.4  27359.3

Coefficients:
             Estimate Std. Error t value           Pr(>|t|)
(Intercept)  2159.498   8877.810   0.243            0.80823
Nbhd2       -1560.579   2396.765  -0.651            0.51621
Nbhd3       20681.037   3148.954   6.568 0.0000000013779549 ***
Offers      -8267.488   1084.777  -7.621 0.0000000000064691 ***
SqFt           52.994      5.734   9.242 0.0000000000000011 ***
BrickYes    17297.350   1981.616   8.729 0.0000000000000178 ***
Bedrooms     4246.794   1597.911   2.658            0.00894 **
Bathrooms    7883.278   2117.035   3.724            0.00030 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10020 on 120 degrees of freedom
Multiple R-squared:  0.8686,    Adjusted R-squared:  0.861
F-statistic: 113.3 on 7 and 120 DF,  p-value: < 0.00000000000000022
```

We see keeping all other co-variates constant, we see significant direct relationship between price of a house and it being a brick-house, owing to p-value way below alpha (0.05)

Part (2) Since Neighbourhood goes into the model matrix as a factor with all it's levels one hot encoded, we see the model does not revert out a beta corresponding to the first level of the factor. This means, given all things constant, say if the average price of an apartment in neighborhood 3 is \$100k, we can use the betas of other neighbourhoods to compare prices with neighborhood 3. Since beta for neighborhood 1 and 2 are both ~ -\$19 to -\$20k, we can say the average prices in those neighborhoods, given other covariates kept constant would be approximately \$80k and \$79k respectively.

Thus there is a premium associated with houses in neighborhood 3.

To summarize brick houses are approximately \$17k more expensive whereas neighbourhood 3 houses are roughly \$19k to\$20k more expensive.

Please note - we're assuming linear relationship between price and number of rooms and bathrooms based on intuitive sense check despite the number of unique values in these variables is not too high.

Part (3)

```
Call:
lm(formula = Price ~ Offers + SqFt + Bedrooms + Bathrooms + Brick *
    Nbhd, data = housing)

Residuals:
     Min       1Q   Median       3Q      Max
-27225.1  -5219.0   -273.7   4297.4  27507.2

Coefficients:
                Estimate Std. Error t value             Pr(>|t|)
(Intercept)     3695.511   8829.382   0.419              0.67631
Offers         -8381.770   1068.248  -7.846 0.000000000002152450 ***
SqFt              53.745      5.686   9.453 0.000000000000000396 ***
Bedrooms        4777.216   1586.397   3.011              0.00318 **
Bathrooms       6457.287   2160.867   2.988              0.00341 **
BrickYes       12093.056   4082.168   2.962              0.00369 **
Nbhd2          -1317.656   2679.849  -0.492              0.62385
Nbhd3          16980.797   3437.529   4.940 0.000002604352876062 ***
BrickYes:Nbhd2  2668.449   5068.893   0.526              0.59957
BrickYes:Nbhd3 11933.197   5341.027   2.234              0.02735 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9847 on 118 degrees of freedom
Multiple R-squared:  0.8752,    Adjusted R-squared:  0.8657
F-statistic: 91.94 on 9 and 118 DF,  p-value: < 0.00000000000000022
```

From the above summary table, we see co-efficients for combination of the neighborhood 3 and brick house are positive and significant at significance level alpha = 0.05. Here we're taking considering both main and interaction effects (Brick*Nbhd) ; using only interaction terms (Brick:Nbhd) might give us a better significance level since main effects might get distributed into interaction terms

On average all brick houses are priced approximately -$9k to $12k less than brick houses in neighborhood 3. The main effects apprise us that other neighbourhood houses, not controlling for them being brick houses are prices $17k to \18k lesser than neighborhood 3 houses.

Note these differences are not equal to the sum of differences we availed from LR without interactions (~$36k to $37k) and this is because the effect size of one variable say neighborhood depends upon the value of other variable - house being brick house.

Hence there's a premium on neighborhood 3 brick house, but there's no extra premium as in it's not greater than the summation of premiums from linear model without interactio#### Part (4)

By definition in given problems, neighborhoods 1 and 2 are both traditional house settings. Seeing coefficients of these neighborhoods in previous regression result, we see betas for neighborhood 1 and 2 are pretty much similar and both categories do not have significant p-values. (Intercept and Nbhd2 both with p-val > 0.5)

So, there is no definite mathematical difference between betas for the two neighborhoods suggesting combining the categories might be valid proposition.

Hence we experiment with a model with neighborhoods 1 and 2 combined and accept it if it reduces the adjusted R-sq. This will help us avail a simpler model more robust to correctly predict the unseen test data.

```
Call:
lm(formula = Price ~ Offers + SqFt + Bedrooms + Bathrooms + Brick *
```

```
    NewNbhd, data = housing)

Residuals:
     Min       1Q   Median       3Q      Max
-26710.2  -5797.0   -277.9   4337.6  26483.2

Coefficients:
                 Estimate Std. Error t value             Pr(>|t|)
(Intercept)       3407.487   8559.390   0.398              0.69126
Offers           -8298.791    997.359  -8.321     0.00000000000016 ***
SqFt                53.728      5.488   9.790 < 0.0000000000000002 ***
Bedrooms          4652.044   1554.250   2.993              0.00335 **
Bathrooms         6407.659   2137.027   2.998              0.00330 **
BrickYes         13664.535   2327.643   5.871     0.00000003952808 ***
NewNbhd          17709.779   2949.399   6.005     0.00000002102630 ***
BrickYes:NewNbhd 10361.809   4100.564   2.527              0.01281 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9780 on 120 degrees of freedom
Multiple R-squared:  0.8748,    Adjusted R-squared:  0.8675
F-statistic: 119.8 on 7 and 120 DF,  p-value: < 0.00000000000000022
```

Given we do see slight improvement in adjusted R sq. and coefficients of neighborhood 1 and 2 weren't significant, it does make sense to combine them into one category!

Problem #3 : What causes what??

Part (1) Different cities will have different co-efficients for their relationship of crime with respect to High alert status and could potentially lead to averaging out the individual effects in case multiple cities are taken at once. For example, consider city 1 where regression line slope suggests that crime increases by certain rate with respect High alert. On the contrary city2 could have a regression line slope suggesting crime decreasing by the same rate with respect setting of the high alert. In case we plot these two cities data together, it results in cancellation of individual effects and the overall regression line for the two cities might result out with slop zero suggesting no relationship, which is definitely fallacious given our premise.

Part (2) Given the issue is quantifying crime with respect to number of police personnel or higher security is that wherever there is crime, only those places would be followed up by higher number of police personnel. It wasn't feasible to avail a comparison set of measuring crime rate with different status of security settings while keeping all other variables constant since you wouldn't expect a large security need to be in an area without reasons other than crime.

Researchers at UPenn first isolated a controlled demographic (geography of Washington DC) first where such a case would be possible. Given DC is considered high risk area to terror attacks, they have excess police personnel whenever there is an orange alert in the city. This means researchers are able to compare crime rates on days of normal vs High alert in the city while keeping any possible confounding variables constant. This data would then be helpful in isolating the effect of High alert status on crime, given all other covariates such as geography, people on the street were checked to be almost constant.

In Table 2, we quantify this when we see daily crime in DC decreased by roughly 7.3 on high alert status when modeled while not controlling for metro activity. Whereas it decreased by 6 on high alert status when controlled for Metro ridership as well. The second model provides more information for predicting crime rate and increases R-square (goodness of fit) from 0.14 to 0.17. The relatively low values of R-sq also suggest that while high alert status or increased security personnel and absence of people from metro ridership both

have a significant impact in reducing the crime rate, these do not fully explain the crime rate owing to such low amounts of dependent variable's variance being explained.

Part (3) The hypothesis behind metro ridership was that a higher security on high alert status would discourage public outings, which in turn would discourage any criminals to target public in their criminal activities. The difference in availability of public targets might become the cause of difference in crime on normal vs excess police days. Owing to this it becomes essential to control for the people on the streets or 'access to criminal activity'

The researchers took the metro ridership as a proxy for people travelling or people in public and being prone to criminal activity and validated that this activity was the same for normal vs excess police days. This approach ensures, if all other possible reasons that could have caused a change in crime rate are kept the same, whatever increase or decrease we see in crime rate on excess police days can be directly attributed to excess police presence!

In the table, the variable undergoes Log transformation to curtail the amount of deviations.

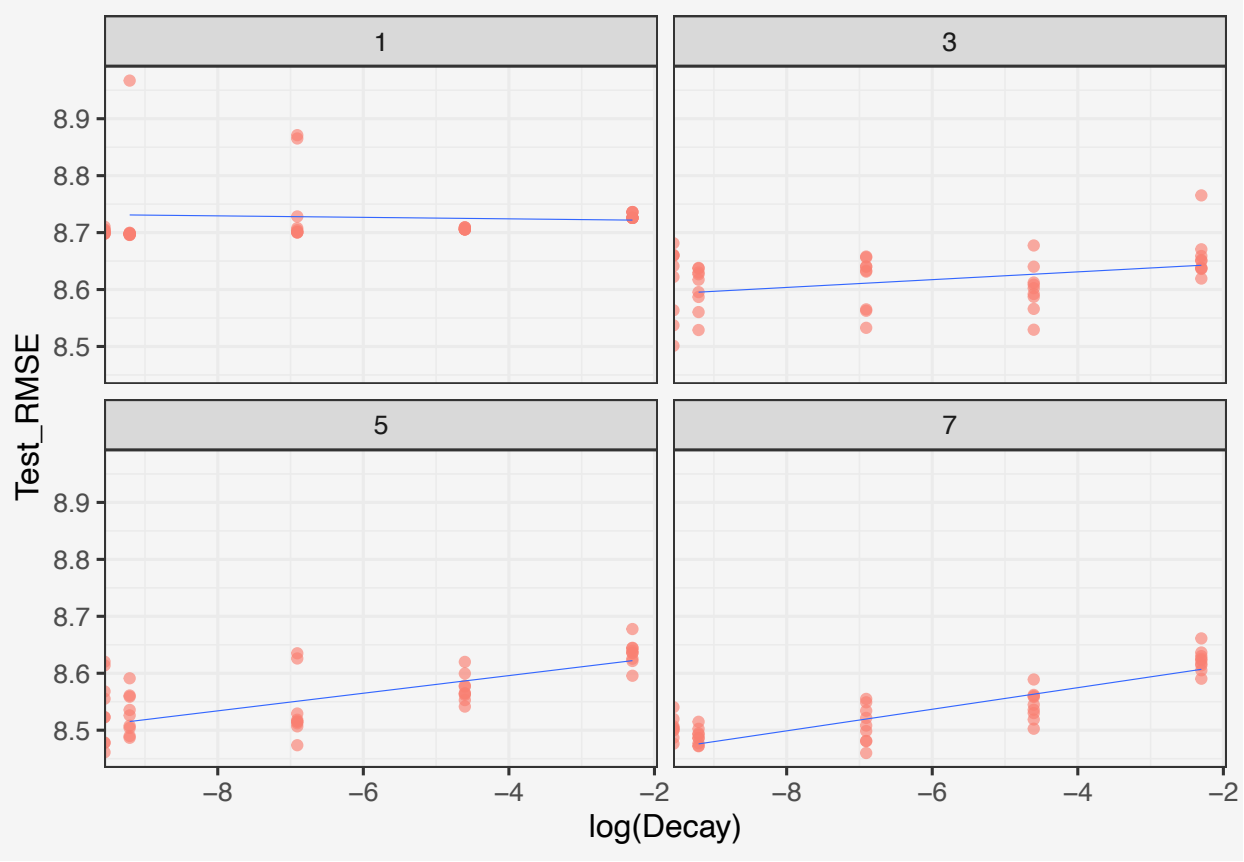
Part (4) The table 4 gives out coefficients with High alert (High security personnel) interacting with the concerned demographic. We see two interaction terms in the coefficient table that say while effect of High alert is significant and is reducing crime rate in District 1, we can't say anything concrete about the High alert impact on crime rate in Other districts, given the associated p-val is not sufficiently low enough.

Adding one more covariate despite not giving out significant p-value reduces the absolute value of betas for the variables returning significant coefficients. The directionality of the coefficients in maintained as in the previous table.

Problem #4 : Neural networks

Plotting Test Errors graphically

```
Best decay and size parameters with minimum TEST RMSE :

   Iteration Size Decay Test_RMSE
1:         6    7 0.001  8.459949

`geom_smooth()` using formula 'y ~ x'
```



Problem #5 : Contribution to the project

The topic we chose for our group project was a classic classification problem relevant in banking industry - Churn prediction of customers. We decided as a group on the data for the problems and aviled it from Kaggle.

The data was clean and we didn't need to do any null value imputations or data cleaning as such. We decided to initially divide the work into EDA, modelling with logistic, Random forest and XGBoost. I took upon the EDA work along with one more collegue and would later on contribute in the modelling part of the exercise as well.

For EDA, I started out with the univariate analysis of our data. Libraries like Hmisc and DescTools helped me gauge a rough sense of data and give directions in which I wanted to pursue EDA. The idea was EDA should start formulating hypothesis of which variables were useful in modelling or if they needed any transformations etc. I did the univariate analysis of both categorical and continuous variables, completed the bivariate analysis with the ggpairs function to see scatterplots, correlations and relationship with dependent variable in one go. This helped us understand which columns needed to be specifically explored further to check if they're able to differentiate in churn or not.

Collaborating with one of my collegues, we did more EDA that showed variables distributions being different for the two classes in question. I cleaned up all EDAs with the custom theme setting of ggplot and coerced all plots to be plotly using ggplotly to standardize the format of all plots in the deck. Post this I explored modelling logistic regression; one of my peers was searching for best logistic model manually by searching for the best AIC and was using the lrm function for logistic. I tried to leverage a stepwise glm implementation to check if despite the tunnel-visioning that stepwise suffers from, are we able to create a better AIC model than random search. It turned out that the implementation of glm vs lrm for same set of variables caused difference in test set results, which was an intersting find.

My other collegues ran the RF and GBM models, but everyone so far was using train error metrics such as AIC, accuracy on their respective train-test split etc. Given we wanted to compare results from different experiments, I went through the literature to understand we need to use the same train-test split for all models, evaluate them on test set only and instead of using error metric as accuracy, AUC would be a better comparator. I edited the final code to create train-test split upfront, used mine and my colleagues' models and ran them on the train set, used out-of-sample test set to generate predictions on each of the models and generate performance metrics (using ROCR package) for us to be able to compare the models.

Amongst the performance metrics, I explored precision, recall, ROC curve, Area under ROC curve, F1 score for each of the models and formulated my understanding of situations of when we'd use different models such as Bagging vs Boosting which were essentially same in terms of AUC (probability threshold cut-off agnostic) but different in terms of Precision and Recall and offered best F1s at very different probability cut-offs.