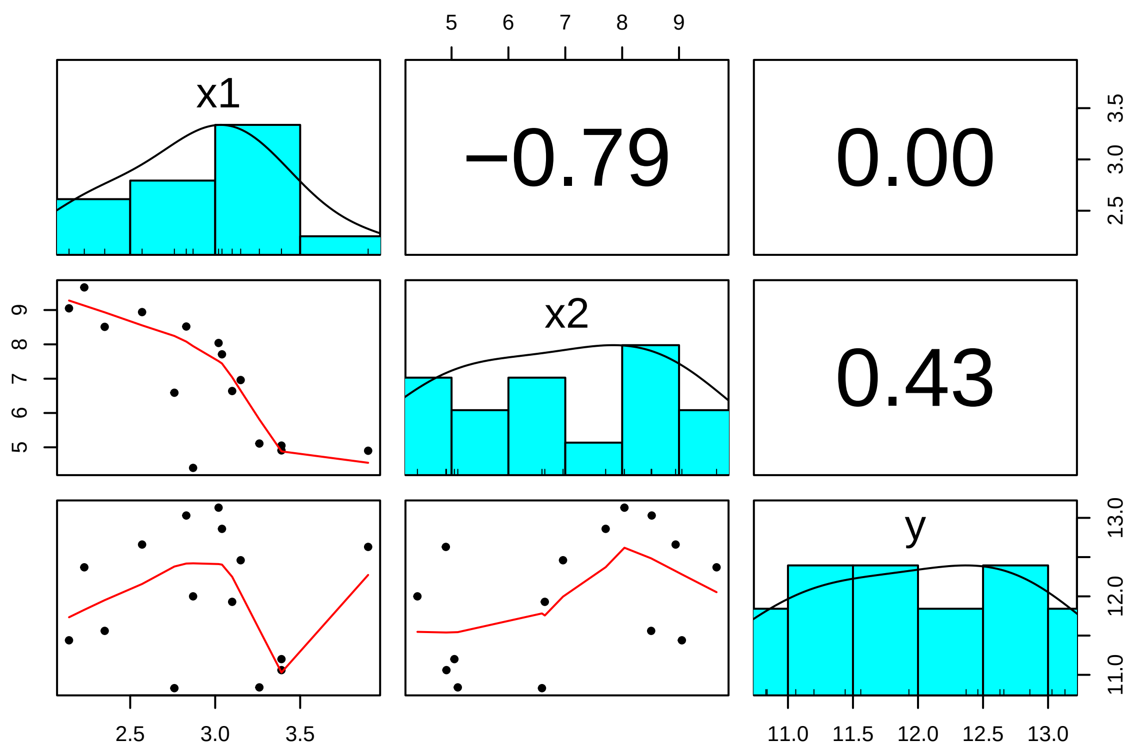
PA1 HW5

Omar Shatrat

2023-10-29

#1a

```
library(psych)

dat = data.frame(
x1=c(2.23,2.57,2.87,3.1,3.39,2.83,3.02,2.14,3.04,3.26,3.39,2.35,
2.76,3.9,3.15),
x2=c(9.66,8.94,4.4,6.64,4.91,8.52,8.04,9.05,7.71,5.11,5.05,8.51,
6.59,4.9,6.96),
y=c(12.37,12.66,12,11.93,11.06,13.03,13.13,11.44,12.86,10.84,
11.2,11.56,10.83,12.63,12.46))

pairs.panels(dat, ellipses = FALSE)
```



x1 and x2 have a strong negative correlation, x2 and y have a strong positive correlation, and x1 and y do not appear to have a correlation at all.

#1b

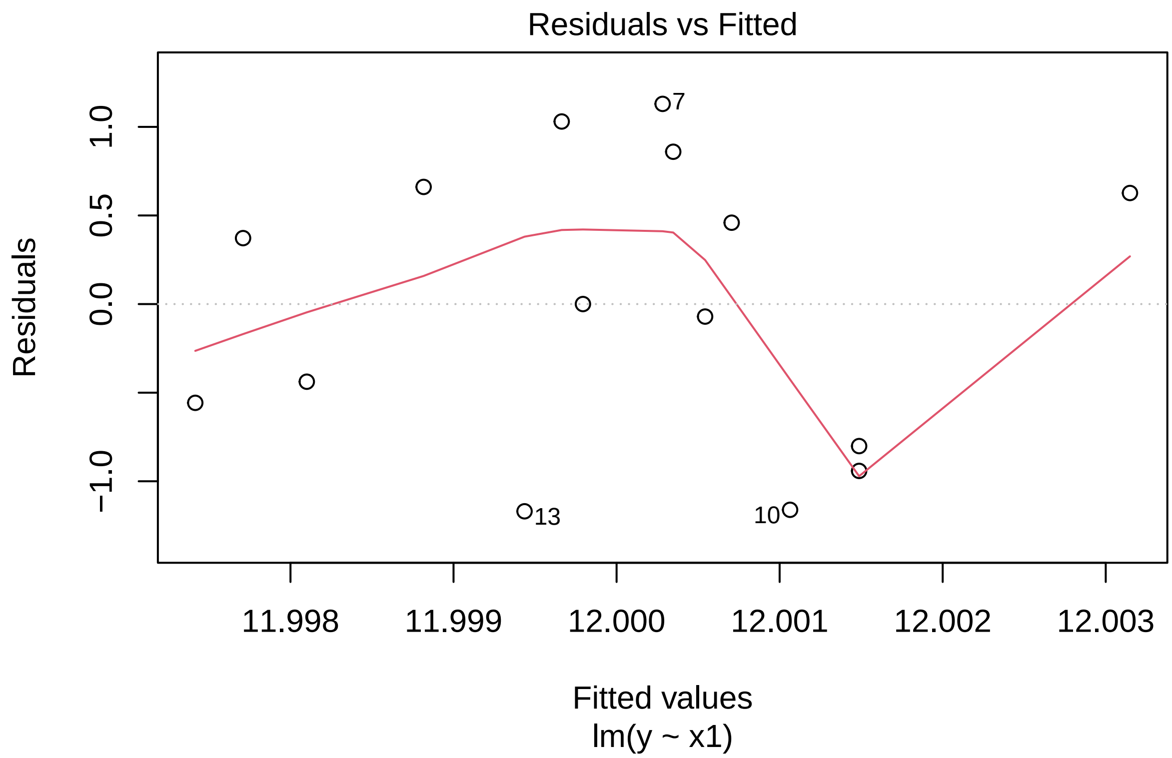
```
model1 <- lm(y~x1, data = dat)

summary(model1)
```

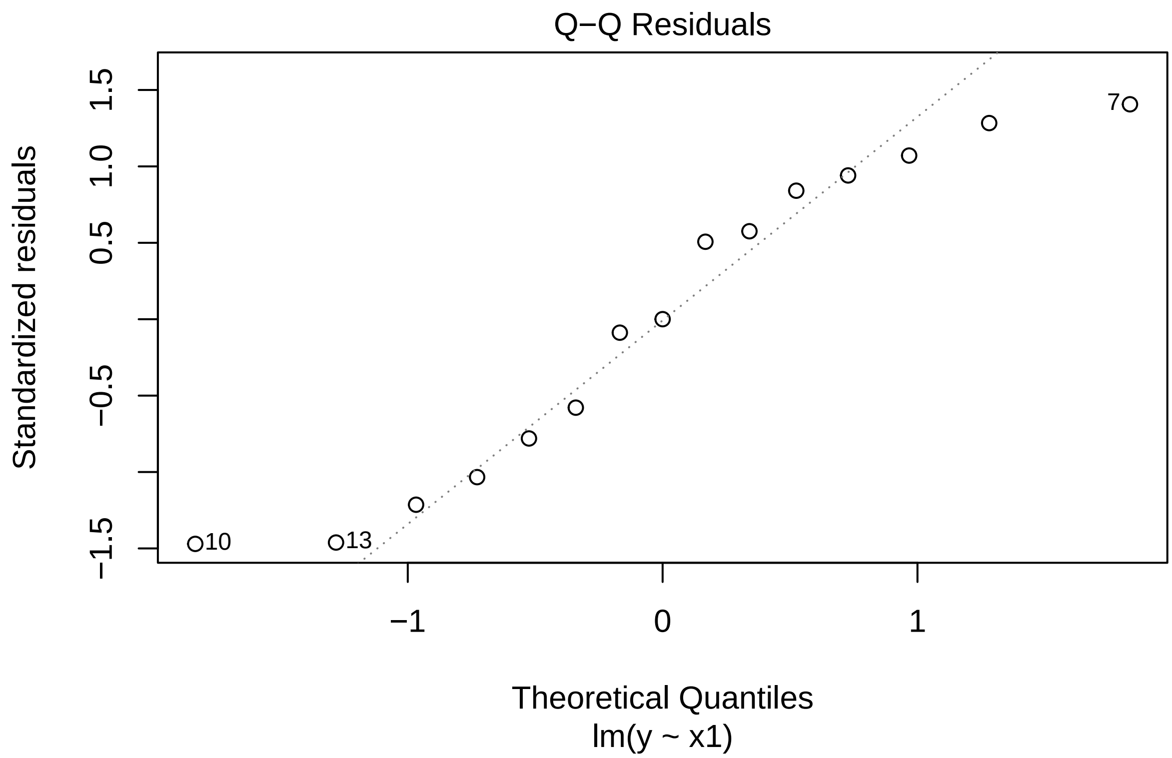
```
##
## Call:
## lm(formula = y ~ x1, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16944 -0.67945  0.00021  0.64402  1.12972
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.990446   1.383341   8.668  9.2e-07 ***
## x1           0.003257   0.465866   0.007    0.995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8324 on 13 degrees of freedom
## Multiple R-squared:  3.76e-06,   Adjusted R-squared:  -0.07692
## F-statistic: 4.888e-05 on 1 and 13 DF,  p-value: 0.9945
```

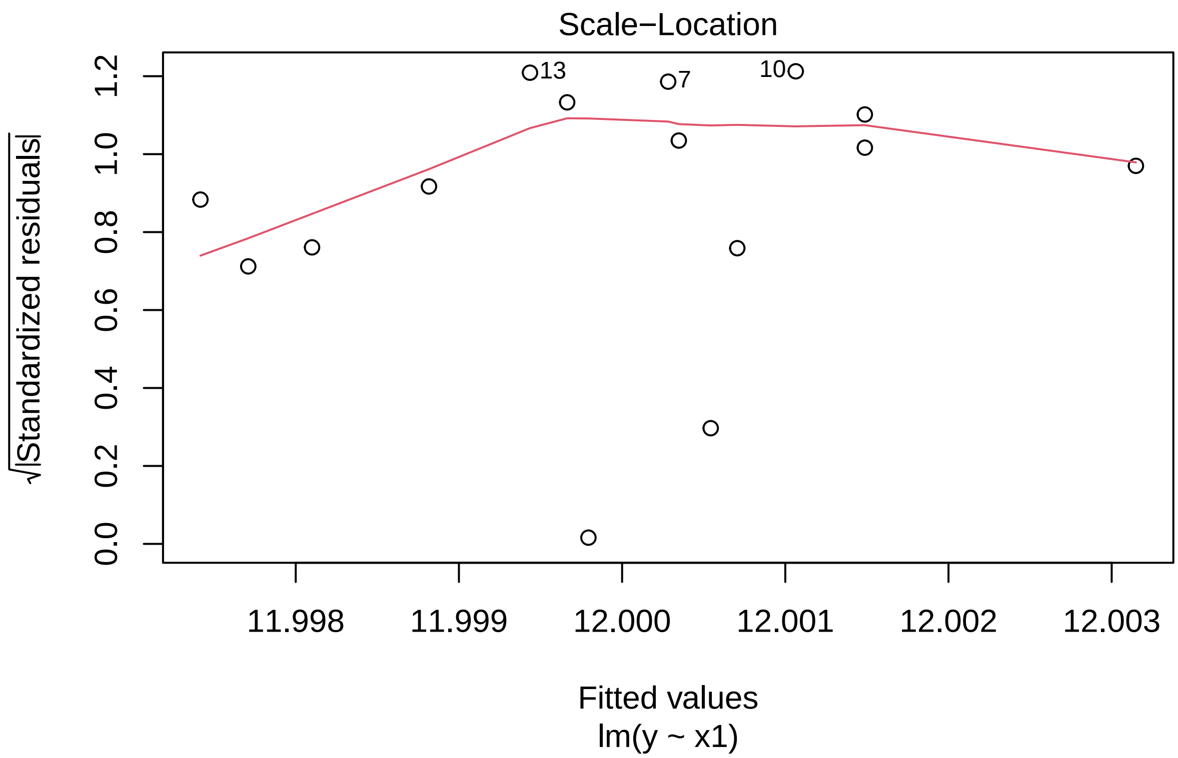
```
model1$residuals
```

```
##             1             2             3             4             5
##  0.3722908924  0.6611834468  0.0002062889 -0.0705428655 -0.9414874515
##             6             7             8             9            10
##  1.0303365766  1.1297177099 -0.5574159602  0.8596525661 -1.1610640164
##            11            12            13            14            15
## -0.8014874515 -0.4380999708 -1.1694354199  0.6268513801  0.4592942749
```

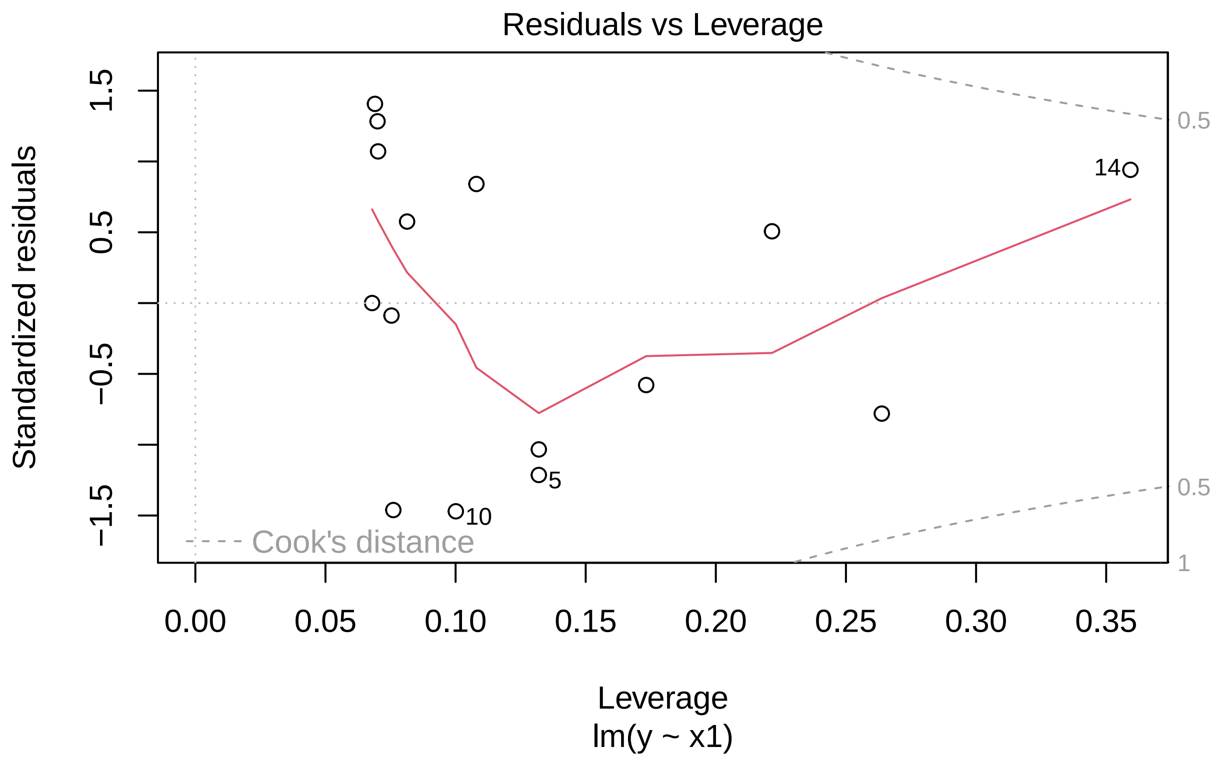
```
plot(model1)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−1.0
7
13
10
11.998
11.999
12.000
12.001
12.002
12.003
Fitted values
lm(y ~ x1)


Q−Q Residuals
Standardized residuals
1.5
1.0
0.5
−0.5
−1.5
7
10
13
−1
0
1
Theoretical Quantiles
lm(y ~ x1)



Scale−Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
13
7
10
11.998
11.999
12.000
12.001
12.002
12.003
Fitted values
lm(y ~ x1)



The model is not significant at all. The residuals are large, the p-value is huge, and the R^2 is very tiny. Also, the standard error for the x1 predictor is many magnitudes higher than its slope estimate.

#1c

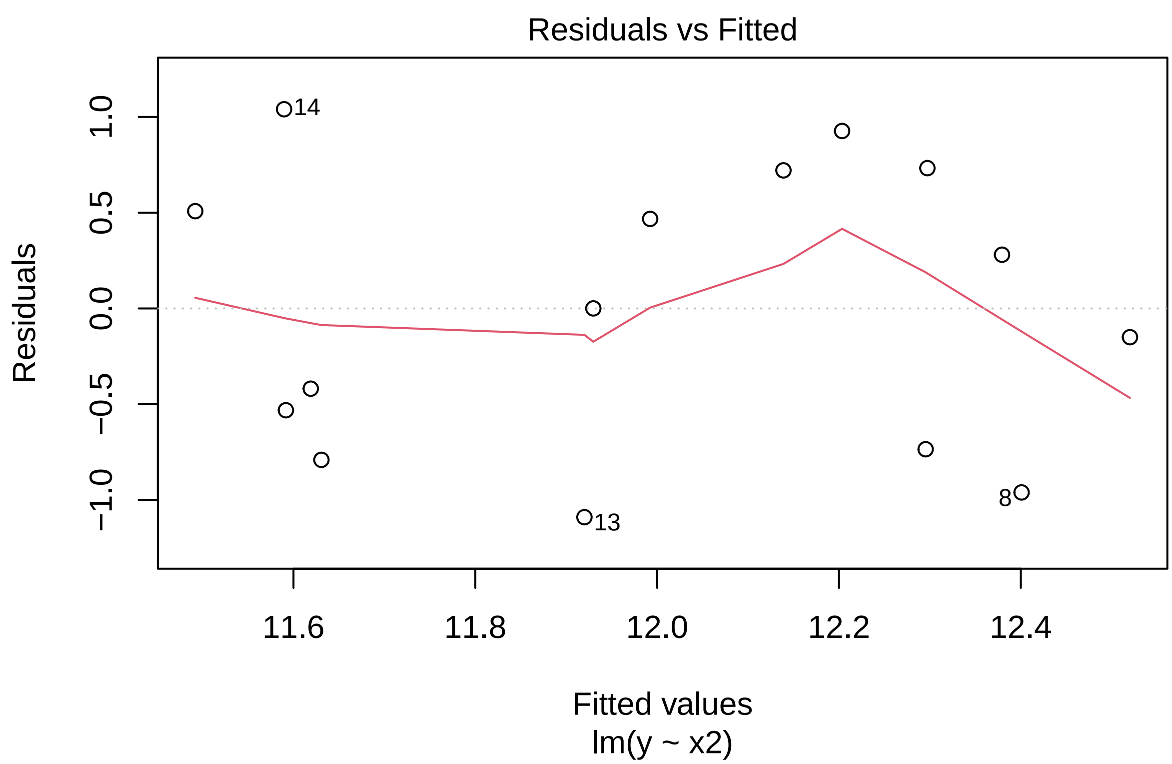
```
model2 <- lm(y~x2, data = dat)

summary(model2)
```

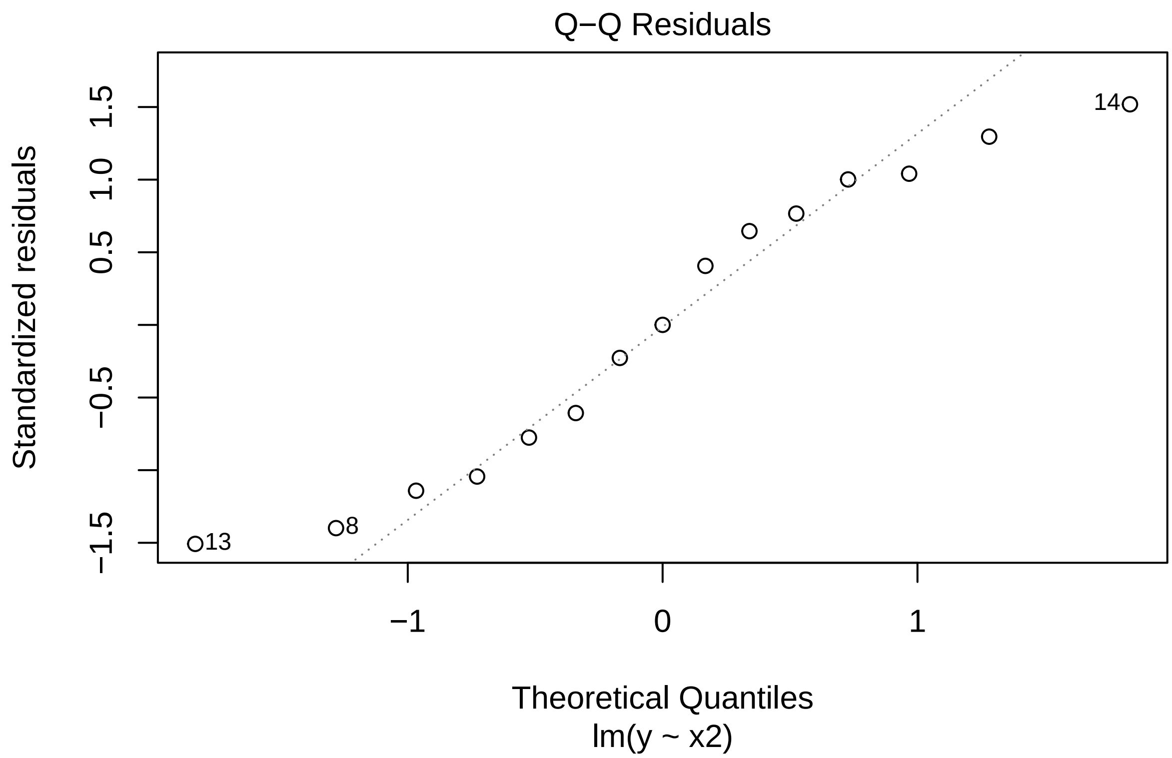
```
##
## Call:
## lm(formula = y ~ x2, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08999 -0.63345  0.00023  0.61458  1.04033
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.6319     0.8109  13.111 7.18e-09 ***
## x2            0.1955     0.1125   1.737    0.106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7499 on 13 degrees of freedom
## Multiple R-squared:  0.1884, Adjusted R-squared:  0.126
## F-statistic: 3.018 on 1 and 13 DF,  p-value: 0.106
```

```r
model2$residuals
```

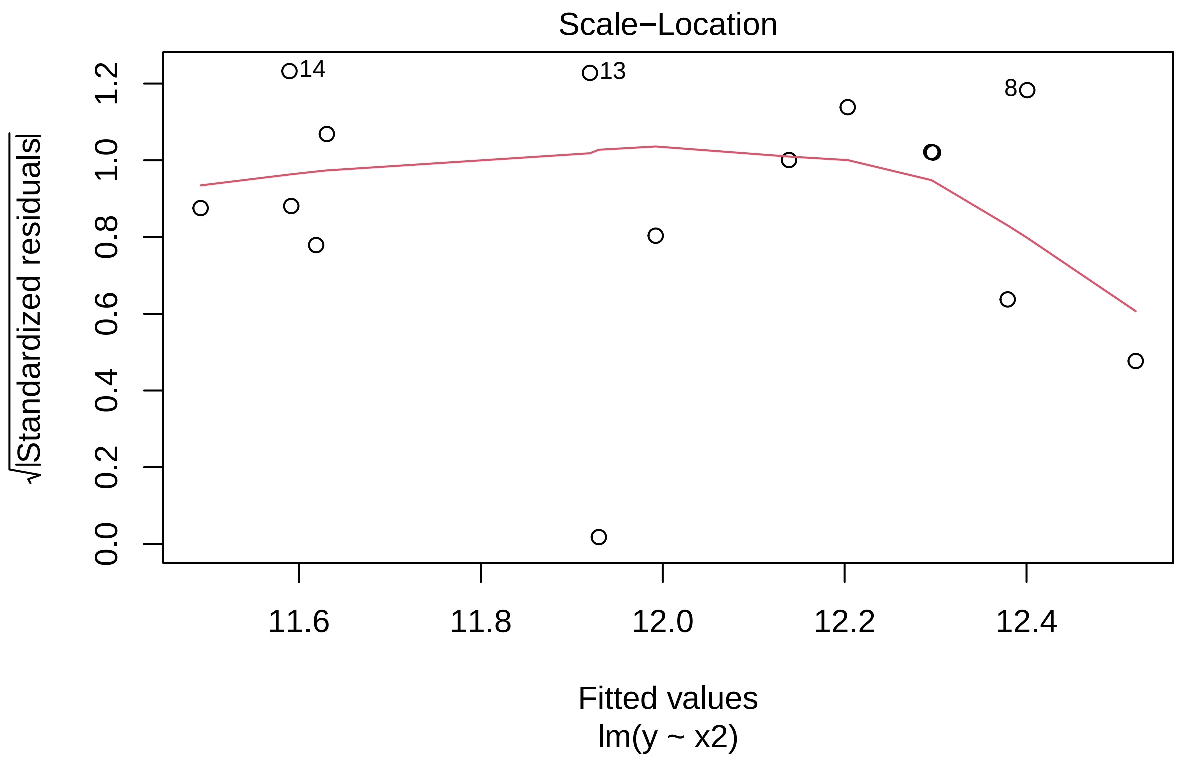
```
##             1             2             3             4             5
## -0.1500439089  0.2806845856  0.5080559260  0.0002339431 -0.5316267576
##             6             7             8             9            10
##  0.7327762074  0.9265952038 -0.9608156010  0.7210957638 -0.7907180061
##            11            12            13            14            15
## -0.4189906315 -0.7352692301 -1.0899932448  1.0403278048  0.4676879455
```

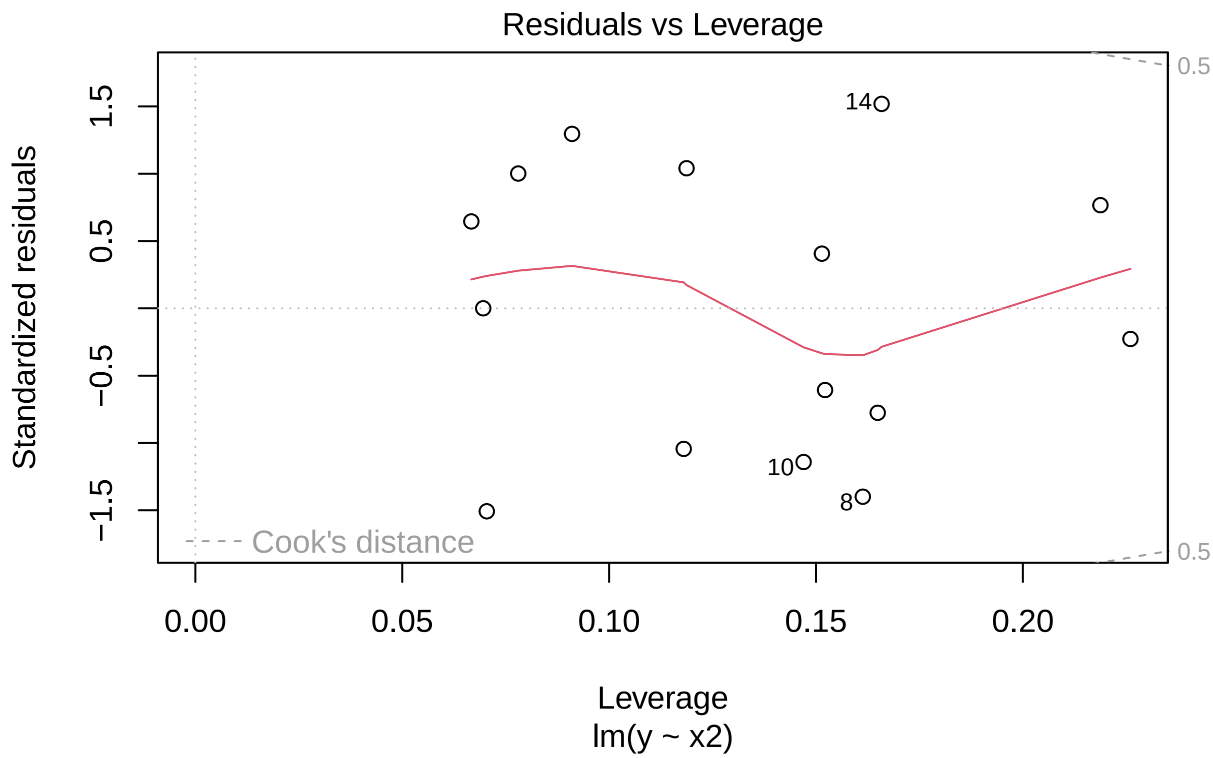
```r
plot(model2)
```





Q−Q Residuals
Standardized residuals
−1.5
−0.5
0.5
1.0
1.5
13
8
14
−1
0
1
Theoretical Quantiles
lm(y ~ x2)


Scale−Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
14
13
8
11.6
11.8
12.0
12.2
12.4
Fitted values
lm(y ~ x2)



Again, the model is once again insignificant, although the residuals and overall fit are better this time around.

#1d

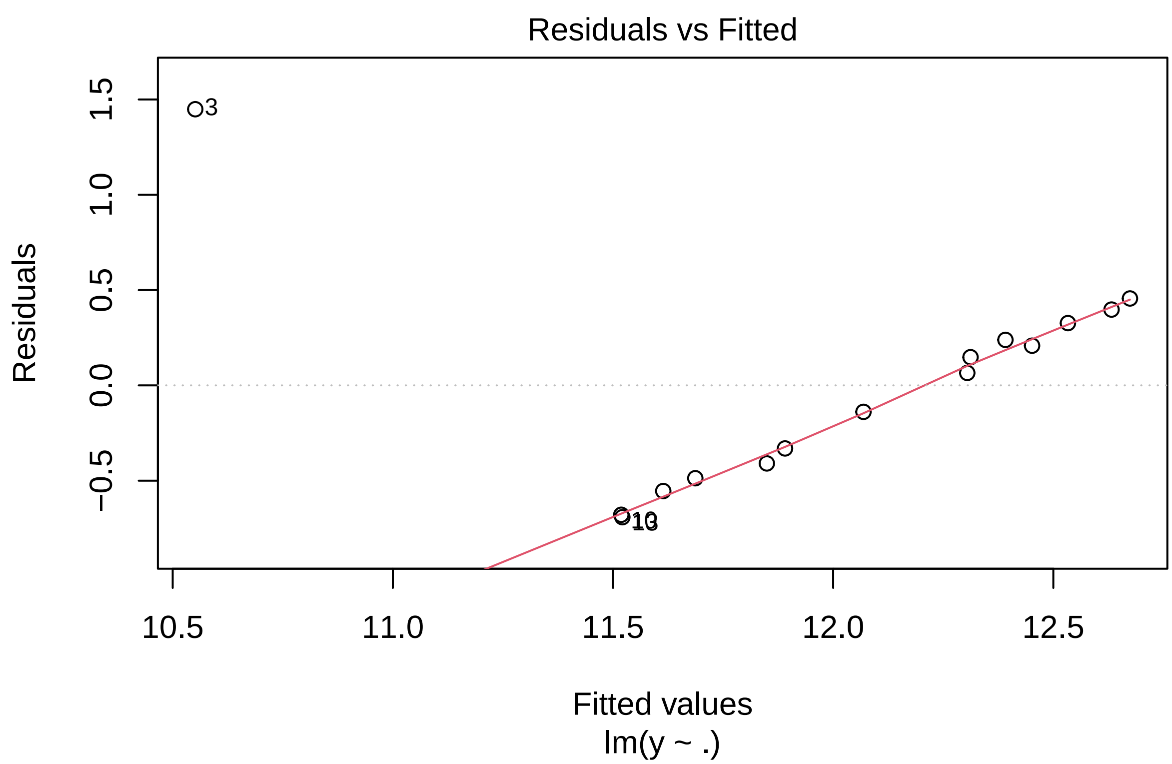
```
model3 <- lm(y~., data = dat)

summary(model3)
```

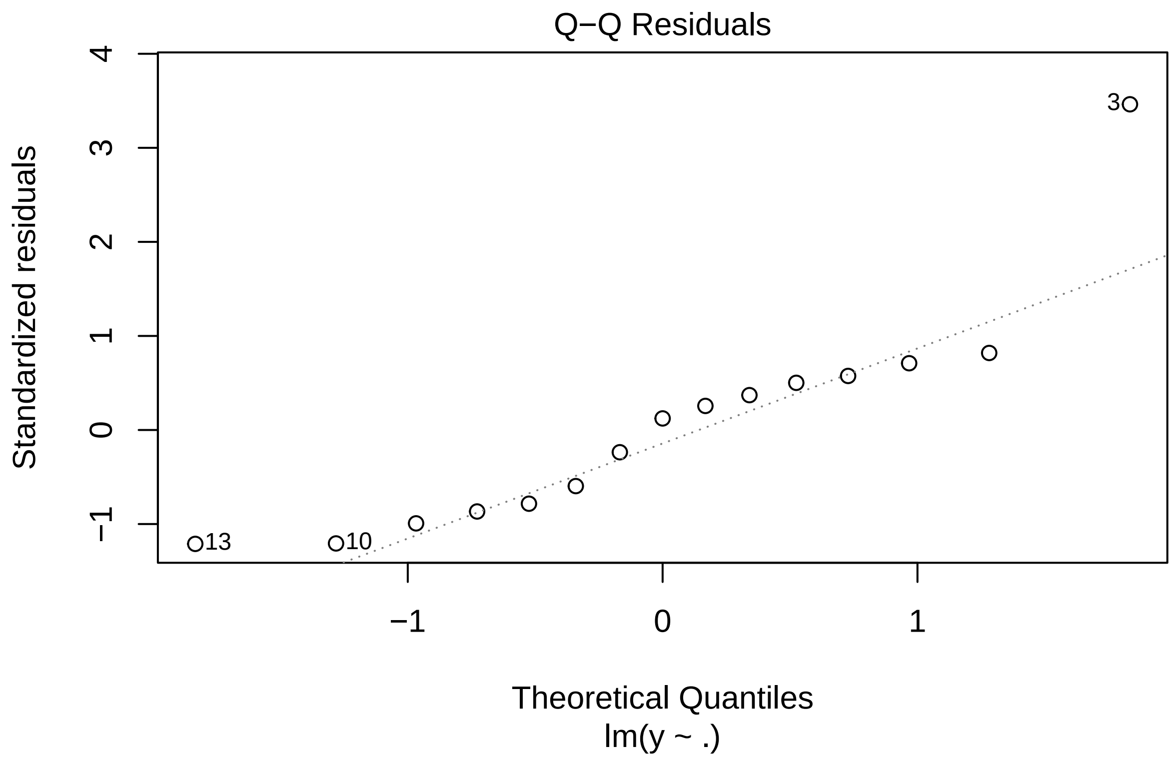
```
##
## Call:
## lm(formula = y ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69127 -0.44813  0.06541  0.28281  1.44873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8610     2.5440   1.518   0.1550
## x1            1.5339     0.5566   2.756   0.0174 *
## x2            0.5200     0.1492   3.485   0.0045 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6108 on 12 degrees of freedom
## Multiple R-squared:  0.503,  Adjusted R-squared:  0.4202
## F-statistic: 6.073 on 2 and 12 DF,  p-value: 0.01507
```

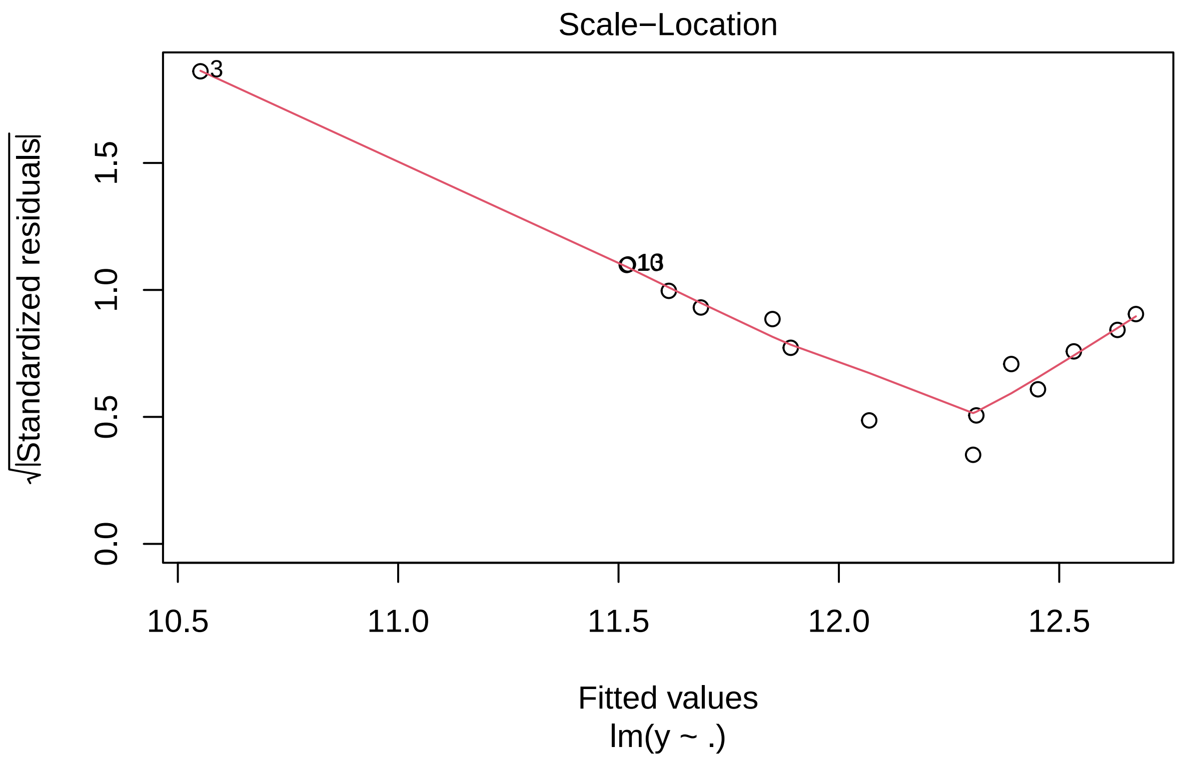
```r
model3$residuals
```

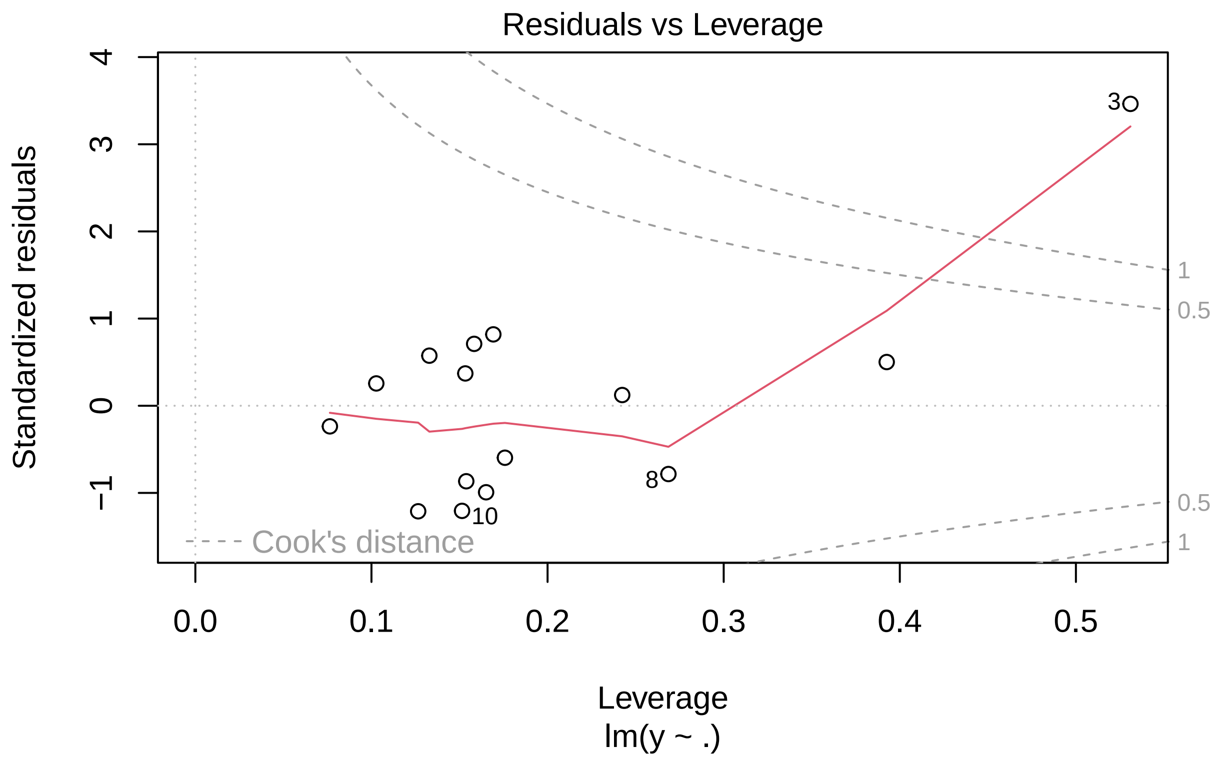
```
##           1           2           3           4           5           6 
##  0.06540834  0.20824473  1.44872750 -0.13881529 -0.55411096  0.39780579 
##           7           8           9          10          11          12 
##  0.45594150 -0.40935409  0.32685289 -0.67869178 -0.48690687 -0.33069910 
##          13          14          15 
## -0.69127417  0.23877488  0.14809662
```

```r
plot(model3)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(y ~ .)
−1
0
1
2
3
4
13
10
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
10.5
11.0
11.5
12.0
12.5
Fitted values
lm(y ~ .)
3
13
10



The model is now significant, as are each of the predictors. However, the residuals are clearly biased and form a straight line. This indicates yhat is biased and that the linearity assumption is in question.

#1e

Because each predictor is insignificant on its own, it does not make sense to use forward selection. Both predictors are significant in a full model, so backward selection makes more sense to use.

#3a

```
sigma = matrix(0.9, nrow = 4, ncol = 4) + .1*diag(4)
A = chol(sigma)
A
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,]    1 0.9000000 0.9000000 0.9000000
## [2,]    0 0.4358899 0.2064742 0.2064742
## [3,]    0 0.0000000 0.3838859 0.1233919
## [4,]    0 0.0000000 0.0000000 0.3635146
```

```
t(A) %*% A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  1.0  0.9  0.9  0.9
## [2,]  0.9  1.0  0.9  0.9
## [3,]  0.9  0.9  1.0  0.9
## [4,]  0.9  0.9  0.9  1.0
```

#3b

```r
Z = matrix(rnorm(4000), nrow = 1000)
X = Z %*% A
cov(X)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.9900902 0.8941878 0.8949668 0.8874582
## [2,] 0.8941878 0.9914580 0.9038064 0.8889626
## [3,] 0.8949668 0.9038064 0.9992289 0.8937772
## [4,] 0.8874582 0.8889626 0.8937772 0.9763081
```

```r
(t(A) %*% A) - cov(X)
```

```
##             [,1]         [,2]          [,3]       [,4]
## [1,] 0.009909798  0.005812201  0.0050331898 0.01254178
## [2,] 0.005812201  0.008541988 -0.0038064156 0.01103736
## [3,] 0.005033190 -0.003806416  0.0007711413 0.00622278
## [4,] 0.012541779  0.011037364  0.0062227798 0.02369193
```

```r
mean((t(A) %*% A) - cov(X))
```

```
## [1] 0.00728729
```

#3c

```r
set.seed(12345)

# generate a new Z, A and X
Z <- matrix(rnorm(151500), nrow = 10100, ncol = 15)

# Define the covariance matrix (with cov(xj, xk) = 0.9 for j != k)
sigma <- diag(15) + 0.9 * (1 - diag(15))

# Perform the Cholesky decomposition
A <- chol(sigma)

# Multiply Z by A to get the correlated variables X
X <- Z %*% A

beta = c(1,-1,1.5,0.5,-0.5,rep(0,10))
e = rnorm(10100)*3
y = 3 + X %*% beta + e
```

#3d

```r
dat = data.frame(X)
dat$y <- y
train <- c(rep(T,100), rep(F, 10000))

training_data <- dat[train,]
```

```r
test_data <- dat[!train,]

fit <- lm(y ~ X1+X2+X3+X4+X5, data = training_data) #where is 7th estimate? do I apply model to all the
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8436 -2.0442  0.2997  1.8333  6.9526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0256     0.3295   9.183    1e-14 ***
## X1            0.9439     0.8875   1.064  0.29026
## X2           -1.6256     1.0049  -1.618  0.10906
## X3            2.7879     0.8924   3.124  0.00237 **
## X4           -0.3034     1.0439  -0.291  0.77200
## X5           -0.3711     0.8164  -0.455  0.65048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.2 on 94 degrees of freedom
## Multiple R-squared:  0.2407, Adjusted R-squared:  0.2003
## F-statistic: 5.961 on 5 and 94 DF,  p-value: 7.843e-05
```

```r
error_variance <- summary(fit)$sigma^2

confint(fit)
```

```
##                  2.5 %    97.5 %
## (Intercept)  2.3714294 3.6797538
## X1          -0.8182307 2.7059553
## X2          -3.6208313 0.3695436
## X3           1.0160970 4.5597365
## X4          -2.3759925 1.7692921
## X5          -1.9919717 1.2498126
```

The error variance is approx. 10.24. The estimates do roughly equal the true parameter values and are withing 2 standard errors. The slopes do have the correct signs, except for X4. Only the intercept and X3 are significant. The 95% CI does cover the true values for each predictor.

#3e

```r
predictions <- predict(fit, newdata = test_data)

mean((test_data$y-predict(fit, test_data))^2)
```

```
## [1] 9.449501
```

MSE = 9.45

#3f

```r
fit <- lm(y~., data = training_data)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ ., data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7768 -1.8727  0.0985  1.8531  6.4236
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.09711    0.34378   9.009 5.69e-14 ***
## X1           1.64535    1.01410   1.622  0.10845
## X2          -1.27455    1.12632  -1.132  0.26102
## X3           3.04446    0.99629   3.056  0.00301 **
## X4           0.17894    1.16865   0.153  0.87867
## X5           0.12057    0.95410   0.126  0.89974
## X6           0.42167    1.04928   0.402  0.68880
## X7          -0.05058    1.16496  -0.043  0.96547
## X8          -1.48874    1.18517  -1.256  0.21255
## X9           1.02701    1.03928   0.988  0.32589
## X10         -0.83981    1.13596  -0.739  0.46179
## X11          0.68516    1.02798   0.667  0.50691
## X12         -0.55163    1.07908  -0.511  0.61055
## X13         -1.25600    1.22391  -1.026  0.30773
## X14          0.52319    1.01348   0.516  0.60705
## X15         -0.73817    1.23259  -0.599  0.55086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.258 on 84 degrees of freedom
## Multiple R-squared:  0.2964, Adjusted R-squared:  0.1708
## F-statistic: 2.359 on 15 and 84 DF,  p-value: 0.007036
```

```r
confint(fit)
```

```
##                  2.5 %    97.5 %
## (Intercept)  2.4134711 3.7807561
## X1          -0.3712955 3.6619970
## X2          -3.5143736 0.9652658
## X3           1.0632199 5.0256971
## X4          -2.1450517 2.5029286
## X5          -1.7767581 2.0178992
## X6          -1.6649445 2.5082863
## X7          -2.3672282 2.2660665
## X8          -3.8455915 0.8681033
## X9          -1.0397031 3.0937305
```

```
## X10         -3.0987981 1.4191776
## X11         -1.3590904 2.7294016
## X12         -2.6974890 1.5942334
## X13         -3.6898753 1.1778700
## X14         -1.4922309 2.5386058
## X15         -3.1893041 1.7129587
```

All the coefficients are once again approx. equal to their true values. X5 experienced a sign flip.

Only the intercept and X3 are significant.

#3g

```
predictions <- predict(fit, newdata = test_data)

mean((test_data$y-predict(fit, test_data))^2)
```

```
## [1] 10.23477
```

The MSE = 10.23.

#3h Forward Selection

```
model_step <- lm(y~1, data=training_data)
stepwise_model <- step(model_step,scope=~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15,test='F')
```

```
## Start:  AIC=255.97
## y ~ 1
##
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + X3    1    256.55 1011.0 235.36  24.868 2.655e-06 ***
## + X1    1    195.03 1072.6 241.26  17.820 5.434e-05 ***
## + X9    1    189.14 1078.5 241.81  17.187 7.204e-05 ***
## + X14   1    182.34 1085.2 242.44  16.465 9.965e-05 ***
## + X4    1    175.79 1091.8 243.04  15.779 0.0001360 ***
## + X6    1    170.85 1096.7 243.49  15.267 0.0001718 ***
## + X11   1    168.33 1099.3 243.72  15.007 0.0001935 ***
## + X7    1    166.22 1101.4 243.91  14.790 0.0002138 ***
## + X5    1    158.74 1108.8 244.59  14.030 0.0003039 ***
## + X10   1    150.15 1117.4 245.36  13.169 0.0004545 ***
## + X13   1    149.97 1117.6 245.38  13.150 0.0004585 ***
## + X12   1    146.97 1120.6 245.65  12.853 0.0005274 ***
## + X15   1    146.37 1121.2 245.70  12.793 0.0005424 ***
## + X2    1    146.04 1121.5 245.73  12.761 0.0005509 ***
## + X8    1    140.49 1127.1 246.22  12.215 0.0007135 ***
## <none>                1267.6 255.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=235.36
## y ~ X3
##
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + X8    1    41.330  969.71 233.18  4.1343   0.04475 *
```

```
## + X13   1    37.991  973.05 233.53  3.7872   0.05454 .
## + X2    1    36.449  974.59 233.68  3.6277   0.05979 .
## + X10   1    32.480  978.56 234.09  3.2196   0.07587 .
## + X15   1    31.377  979.66 234.20  3.1068   0.08112 .
## + X12   1    24.221  986.82 234.93  2.3808   0.12609
## + X7    1    20.373  990.66 235.32  1.9948   0.16104
## <none>              1011.04 235.36
## + X5    1    14.192  996.84 235.94  1.3810   0.24281
## + X4    1    12.596  998.44 236.10  1.2237   0.27138
## + X6    1    10.300 1000.74 236.33  0.9984   0.32019
## + X11   1     8.988 1002.05 236.46  0.8700   0.35327
## + X14   1     5.646 1005.39 236.80  0.5447   0.46227
## + X9    1     2.311 1008.73 237.13  0.2222   0.63840
## + X1    1     1.146 1009.89 237.24  0.1101   0.74080
## - X3    1   256.553 1267.59 255.97 24.8678 2.655e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=233.18
## y ~ X3 + X8
##
##        Df Sum of Sq     RSS    AIC F value   Pr(>F)
## <none>                969.71 233.18
## + X1    1    12.920  956.79 233.84  1.2963 0.257719
## + X13   1     8.087  961.62 234.34  0.8074 0.371152
## + X2    1     7.928  961.78 234.36  0.7914 0.375912
## + X10   1     7.082  962.62 234.45  0.7063 0.402769
## + X15   1     5.434  964.27 234.62  0.5410 0.463832
## + X9    1     4.817  964.89 234.68  0.4792 0.490442
## + X12   1     1.821  967.89 234.99  0.1806 0.671797
## + X14   1     1.283  968.42 235.05  0.1272 0.722167
## + X11   1     0.605  969.10 235.12  0.0600 0.807084
## + X4    1     0.533  969.17 235.13  0.0528 0.818752
## + X6    1     0.518  969.19 235.13  0.0513 0.821360
## + X7    1     0.313  969.39 235.15  0.0310 0.860611
## + X5    1     0.085  969.62 235.17  0.0084 0.927101
## - X8    1    41.330 1011.04 235.36  4.1343 0.044755 *
## - X3    1   157.396 1127.10 246.22 15.7443 0.000139 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = y ~ X3 + X8, data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7523 -2.1644  0.2534  1.8345  7.6125
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1062     0.3229   9.618 8.95e-16 ***
```

```
## X3            2.9406     0.7411   3.968 0.000139 ***
## X8           -1.4696     0.7228  -2.033 0.044755 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.162 on 97 degrees of freedom
## Multiple R-squared:  0.235,  Adjusted R-squared:  0.2192
## F-statistic:  14.9 on 2 and 97 DF,  p-value: 2.278e-06
```

```r
confint(stepwise_model)
```

```
##                 2.5 %     97.5 %
## (Intercept)  2.465270  3.7471985
## X3           1.469730  4.4114622
## X8          -2.904154 -0.0351047
```

#3h Backward Selection

```r
fit <- step(fit, direction = 'both')
```

```
## Start:  AIC=250.81
## y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 +
##     X12 + X13 + X14 + X15
##
##        Df Sum of Sq    RSS    AIC
## - X7    1     0.020 891.86 248.81
## - X5    1     0.170 892.01 248.83
## - X4    1     0.249 892.09 248.84
## - X6    1     1.715 893.55 249.00
## - X12   1     2.775 894.61 249.12
## - X14   1     2.829 894.67 249.13
## - X15   1     3.808 895.65 249.24
## - X11   1     4.716 896.56 249.34
## - X10   1     5.803 897.64 249.46
## - X9    1    10.368 902.21 249.97
## - X13   1    11.181 903.02 250.06
## - X2    1    13.596 905.43 250.32
## - X8    1    16.753 908.59 250.67
## <none>              891.84 250.81
## - X1    1    27.949 919.79 251.90
## - X3    1    99.141 990.98 259.35
##
## Step:  AIC=248.81
## y ~ X1 + X2 + X3 + X4 + X5 + X6 + X8 + X9 + X10 + X11 + X12 +
##     X13 + X14 + X15
##
##        Df Sum of Sq    RSS    AIC
## - X5    1     0.152 892.01 246.83
## - X4    1     0.231 892.09 246.84
## - X6    1     1.734 893.59 247.01
## - X12   1     2.756 894.61 247.12
## - X14   1     2.827 894.69 247.13
```

```
## - X15   1     3.861 895.72 247.25
## - X11   1     4.724 896.58 247.34
## - X10   1     5.882 897.74 247.47
## - X9    1    10.397 902.26 247.97
## - X13   1    11.272 903.13 248.07
## - X2    1    13.629 905.49 248.33
## <none>              891.86 248.81
## - X8    1    18.293 910.15 248.84
## - X1    1    28.016 919.87 249.91
## + X7    1     0.020 891.84 250.81
## - X3    1    99.609 991.47 257.40
##
## Step:  AIC=246.83
## y ~ X1 + X2 + X3 + X4 + X6 + X8 + X9 + X10 + X11 + X12 + X13 +
##     X14 + X15
##
##        Df Sum of Sq    RSS    AIC
## - X4    1     0.278 892.29 244.86
## - X6    1     1.744 893.76 245.03
## - X12   1     2.645 894.66 245.13
## - X14   1     2.856 894.87 245.15
## - X15   1     3.761 895.77 245.25
## - X11   1     4.683 896.69 245.35
## - X10   1     5.741 897.75 245.47
## - X9    1    11.093 903.10 246.07
## - X13   1    11.266 903.28 246.09
## - X2    1    13.485 905.50 246.33
## <none>              892.01 246.83
## - X8    1    18.160 910.17 246.85
## - X1    1    28.105 920.12 247.93
## + X5    1     0.152 891.86 248.81
## + X7    1     0.002 892.01 248.83
## - X3    1    99.908 991.92 255.45
##
## Step:  AIC=244.86
## y ~ X1 + X2 + X3 + X6 + X8 + X9 + X10 + X11 + X12 + X13 + X14 +
##     X15
##
##        Df Sum of Sq     RSS    AIC
## - X6    1     1.657  893.95 243.05
## - X12   1     2.416  894.70 243.13
## - X14   1     3.064  895.35 243.21
## - X15   1     3.975  896.26 243.31
## - X11   1     4.841  897.13 243.40
## - X10   1     5.466  897.75 243.47
## - X9    1    10.815  903.10 244.07
## - X13   1    10.991  903.28 244.09
## - X2    1    13.261  905.55 244.34
## <none>               892.29 244.86
## - X8    1    18.594  910.88 244.92
## - X1    1    30.687  922.98 246.24
## + X4    1     0.278  892.01 246.83
## + X5    1     0.199  892.09 246.84
## + X7    1     0.004  892.28 246.86
```

```
## - X3    1   108.286 1000.58 254.32
##
## Step:  AIC=243.05
## y ~ X1 + X2 + X3 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15
##
##        Df Sum of Sq     RSS    AIC
## - X12   1     1.931  895.88 241.26
## - X14   1     3.376  897.32 241.42
## - X15   1     3.575  897.52 241.45
## - X10   1     5.166  899.11 241.62
## - X11   1     5.310  899.26 241.64
## - X13   1    10.163  904.11 242.18
## - X9    1    13.007  906.95 242.49
## - X2    1    13.727  907.67 242.57
## - X8    1    17.124  911.07 242.94
## <none>               893.95 243.05
## - X1    1    30.104  924.05 244.36
## + X6    1     1.657  892.29 244.86
## + X5    1     0.201  893.74 245.03
## + X4    1     0.190  893.76 245.03
## + X7    1     0.000  893.95 245.05
## - X3    1   112.298 1006.24 252.88
##
## Step:  AIC=241.26
## y ~ X1 + X2 + X3 + X8 + X9 + X10 + X11 + X13 + X14 + X15
##
##        Df Sum of Sq     RSS    AIC
## - X14   1     2.655  898.53 239.56
## - X11   1     4.335  900.21 239.75
## - X15   1     5.561  901.44 239.88
## - X10   1     5.975  901.85 239.93
## - X13   1    10.383  906.26 240.42
## - X9    1    12.042  907.92 240.60
## - X2    1    14.287  910.16 240.84
## <none>               895.88 241.26
## - X8    1    18.184  914.06 241.27
## - X1    1    28.913  924.79 242.44
## + X12   1     1.931  893.95 243.05
## + X6    1     1.172  894.70 243.13
## + X5    1     0.064  895.81 243.26
## + X4    1     0.030  895.85 243.26
## + X7    1     0.000  895.88 243.26
## - X3    1   114.969 1010.85 251.34
##
## Step:  AIC=239.56
## y ~ X1 + X2 + X3 + X8 + X9 + X10 + X11 + X13 + X15
##
##        Df Sum of Sq     RSS    AIC
## - X15   1     3.903  902.43 237.99
## - X11   1     4.080  902.61 238.01
## - X10   1     6.139  904.67 238.24
## - X13   1    11.402  909.93 238.82
## - X2    1    12.260  910.79 238.91
## - X9    1    13.689  912.22 239.07
```

```
## - X8    1    16.903  915.43 239.42
## <none>               898.53 239.56
## - X1    1    32.553  931.08 241.12
## + X14   1     2.655  895.88 241.26
## + X6    1     1.492  897.04 241.39
## + X12   1     1.210  897.32 241.42
## + X4    1     0.141  898.39 241.54
## + X5    1     0.118  898.41 241.55
## + X7    1     0.084  898.45 241.55
## - X3    1   126.019 1024.55 250.68
##
## Step:  AIC=237.99
## y ~ X1 + X2 + X3 + X8 + X9 + X10 + X11 + X13
##
##        Df Sum of Sq     RSS    AIC
## - X11   1     4.159  906.59 236.45
## - X10   1     9.844  912.28 237.08
## - X9    1    12.423  914.86 237.36
## - X2    1    13.646  916.08 237.49
## - X13   1    15.335  917.77 237.68
## <none>               902.43 237.99
## - X8    1    18.610  921.04 238.03
## - X1    1    29.397  931.83 239.20
## + X15   1     3.903  898.53 239.56
## + X12   1     2.817  899.62 239.68
## + X14   1     0.997  901.44 239.88
## + X6    1     0.805  901.63 239.90
## + X4    1     0.155  902.28 239.97
## + X5    1     0.006  902.43 239.99
## + X7    1     0.004  902.43 239.99
## - X3    1   124.400 1026.84 248.91
##
## Step:  AIC=236.45
## y ~ X1 + X2 + X3 + X8 + X9 + X10 + X13
##
##        Df Sum of Sq     RSS    AIC
## - X10   1     8.485  915.08 235.38
## - X2    1    10.845  917.44 235.64
## - X13   1    13.060  919.65 235.88
## - X9    1    15.569  922.16 236.16
## - X8    1    16.693  923.29 236.28
## <none>               906.59 236.45
## - X1    1    30.470  937.06 237.76
## + X11   1     4.159  902.43 237.99
## + X15   1     3.981  902.61 238.01
## + X12   1     1.737  904.86 238.26
## + X6    1     1.185  905.41 238.32
## + X14   1     0.839  905.76 238.36
## + X4    1     0.320  906.27 238.42
## + X5    1     0.007  906.59 238.45
## + X7    1     0.001  906.59 238.45
## - X3    1   125.426 1032.02 247.41
##
## Step:  AIC=235.38
```

```
## y ~ X1 + X2 + X3 + X8 + X9 + X13
##
##        Df Sum of Sq     RSS    AIC
## - X9    1    10.207  925.29 234.49
## - X2    1    14.067  929.15 234.91
## <none>               915.08 235.38
## - X13   1    18.690  933.77 235.41
## - X8    1    20.014  935.09 235.55
## + X10   1     8.485  906.59 236.45
## - X1    1    29.017  944.10 236.51
## + X15   1     7.386  907.69 236.57
## + X12   1     3.630  911.45 236.99
## + X11   1     2.800  912.28 237.08
## + X14   1     0.562  914.52 237.32
## + X6    1     0.518  914.56 237.33
## + X5    1     0.269  914.81 237.35
## + X7    1     0.237  914.84 237.36
## + X4    1     0.056  915.02 237.38
## - X3    1   117.830 1032.91 245.50
##
## Step:  AIC=234.49
## y ~ X1 + X2 + X3 + X8 + X13
##
##        Df Sum of Sq     RSS    AIC
## - X2    1    11.643  936.93 233.74
## - X13   1    13.178  938.46 233.91
## - X8    1    15.369  940.65 234.14
## <none>               925.29 234.49
## + X9    1    10.207  915.08 235.38
## + X11   1     5.470  919.82 235.90
## - X1    1    32.185  957.47 235.91
## + X15   1     4.314  920.97 236.03
## + X10   1     3.123  922.16 236.16
## + X6    1     2.671  922.61 236.20
## + X14   1     1.615  923.67 236.32
## + X12   1     1.045  924.24 236.38
## + X4    1     0.127  925.16 236.48
## + X5    1     0.095  925.19 236.48
## + X7    1     0.021  925.26 236.49
## - X3    1   137.543 1062.83 246.35
##
## Step:  AIC=233.74
## y ~ X1 + X3 + X8 + X13
##
##        Df Sum of Sq     RSS    AIC
## <none>               936.93 233.74
## - X13   1    19.857  956.79 233.84
## - X1    1    24.690  961.62 234.34
## - X8    1    25.377  962.31 234.42
## + X2    1    11.643  925.29 234.49
## + X9    1     7.783  929.15 234.91
## + X15   1     6.518  930.41 235.05
## + X10   1     5.414  931.52 235.16
## + X12   1     2.907  934.02 235.43
```

```
## + X6     1      1.945  934.98 235.54
## + X4     1      1.821  935.11 235.55
## + X11    1      1.655  935.27 235.57
## + X7     1      0.236  936.69 235.72
## + X5     1      0.127  936.80 235.73
## + X14    1      0.103  936.83 235.73
## - X3     1    126.212 1063.14 244.38
```

```r
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ X1 + X3 + X8 + X13, data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2310 -1.8975  0.2254  1.6861  7.4489
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0673     0.3217   9.533 1.64e-15 ***
## X1            1.3978     0.8835   1.582 0.116921
## X3            3.0285     0.8466   3.577 0.000548 ***
## X8           -1.5716     0.9797  -1.604 0.112011
## X13          -1.4510     1.0226  -1.419 0.159185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.14 on 95 degrees of freedom
## Multiple R-squared:  0.2609, Adjusted R-squared:  0.2297
## F-statistic: 8.382 on 4 and 95 DF,  p-value: 7.787e-06
```

No, not all of the right variables did not make it in, only X1 and X3 did.

#3i

```r
predictions <- predict(fit, newdata = test_data)

mean((test_data$y-predict(fit, test_data))^2)
```

```
## [1] 10.04426
```

MSE = 10.04

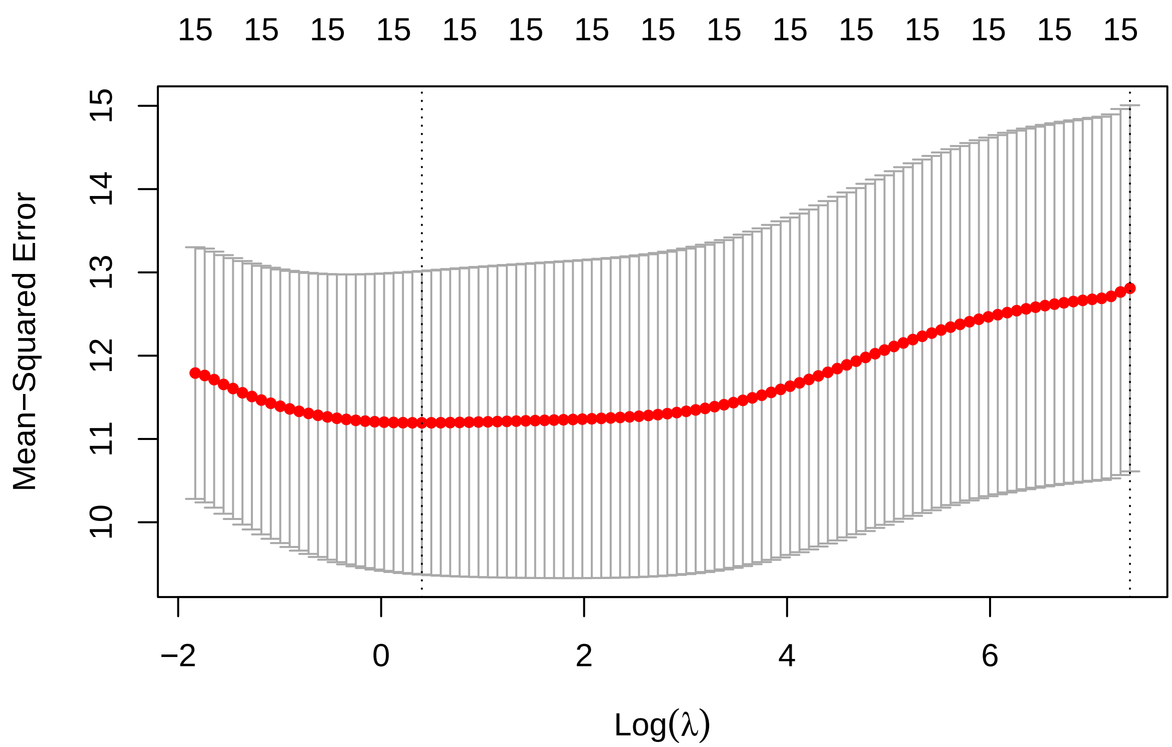
#3j

```r
set.seed(12345)

library('glmnet')
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

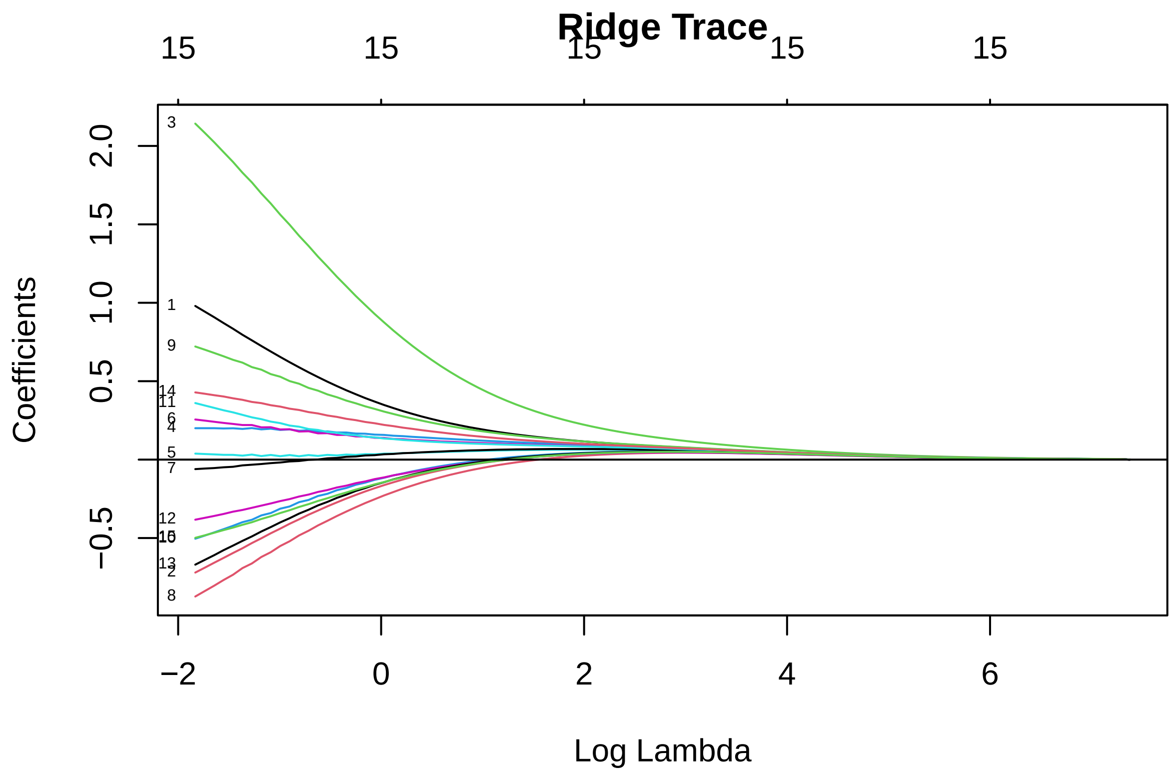
```r
X <- as.matrix(training_data[,-16])
#X <- scale(X)
y <- as.numeric(training_data$y)

cv_params <- cv.glmnet(X,y, alpha = 0)
plot(cv_params)
```




```r
best_lambda <- cv_params$lambda.min

fit <- glmnet(X, y, alpha = 0, lambda = best_lambda)
summary(fit)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      15     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       5     -none-    call
## nobs       1     -none-    numeric
```

```r
plot(cv_params$glmnet.fit, xvar = 'lambda', label = TRUE, main="Ridge Trace"); abline(h=0)
```



#3k

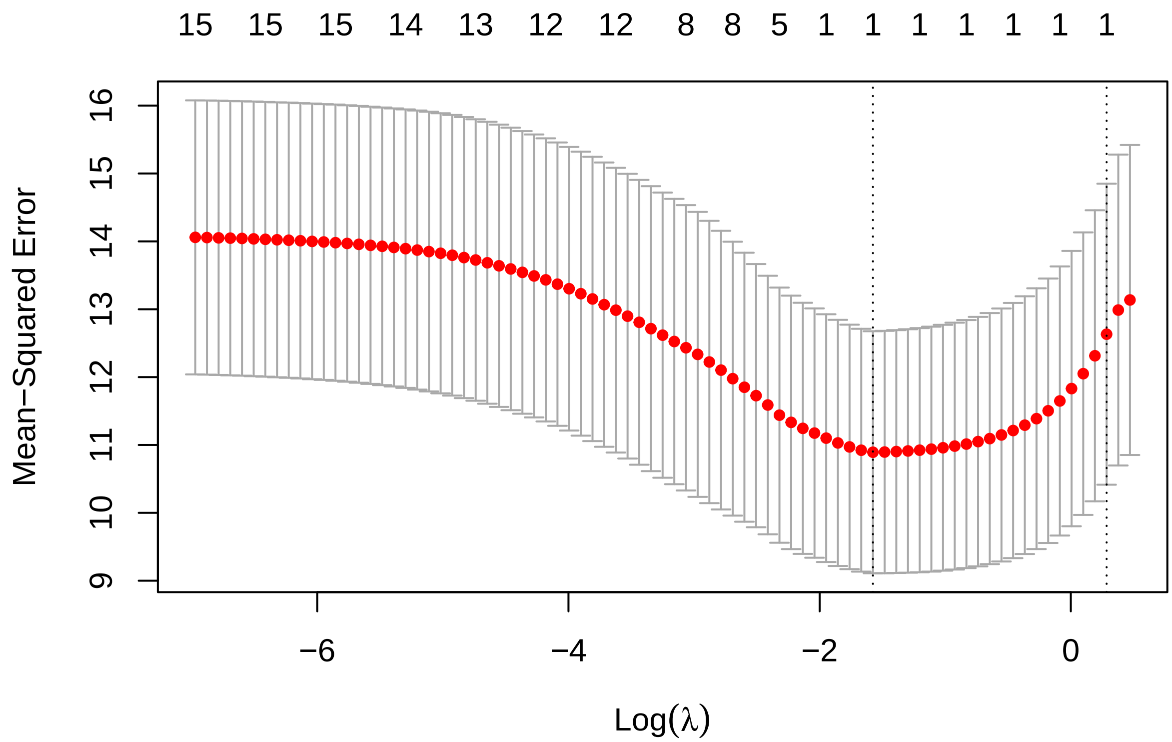
```r
ridge_predictions <- predict(fit, s = best_lambda, newx = scale(as.matrix(test_data[,-16])) )
mse <- mean((test_data$y - ridge_predictions)^2)
mse
```

```
## [1] 9.444578
```

MSE = 9.44

#3l

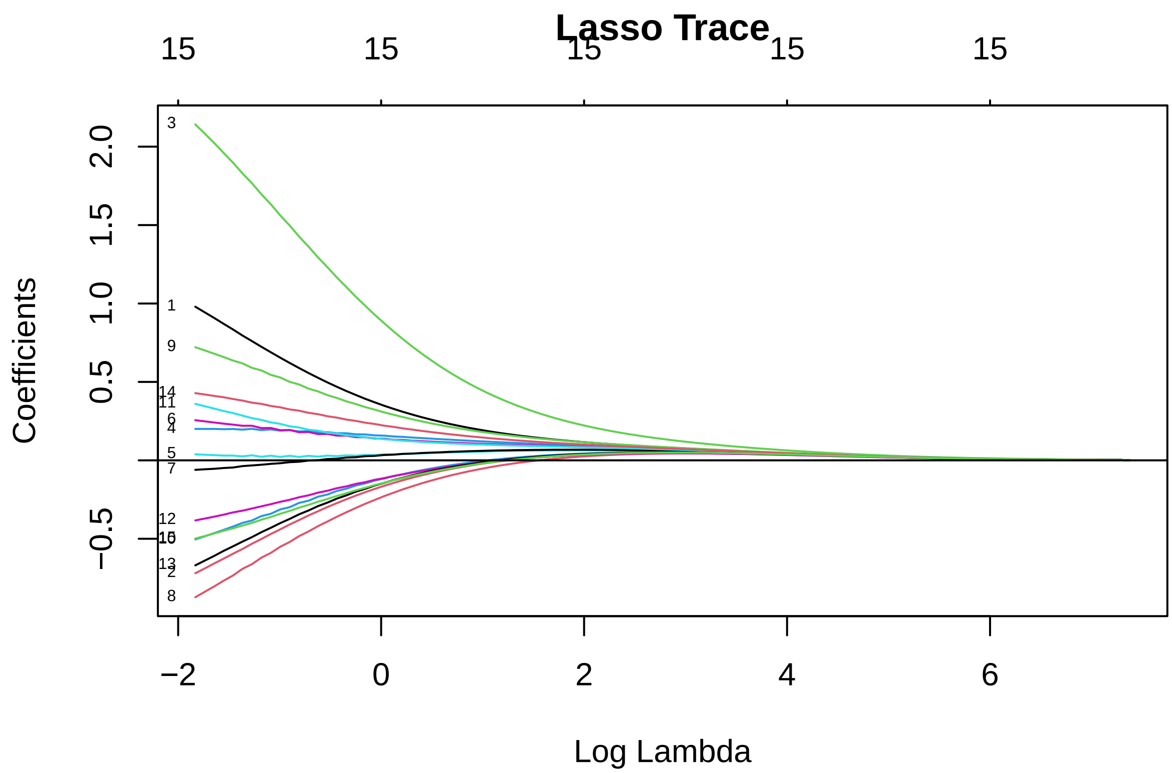
```r
X <- as.matrix(training_data[,-16])
y <- as.numeric(training_data$y)
cvfit <- cv.glmnet(X, y, alpha = 1)  # Alpha = 1 for lasso
plot(cvfit)
```




```r
best_lambda <- cvfit$lambda.min

fit <- glmnet(X, y, alpha = 1, lambda = best_lambda)
summary(fit)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      15     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       5     -none-    call
## nobs       1     -none-    numeric
```

```r
plot(cv_params$glmnet.fit, xvar = 'lambda', label = TRUE, main="Lasso Trace"); abline(h=0)
```



#3m

```r
lasso_predictions <- predict(fit, newx = as.matrix(test_data[,-16]) )
mse <- mean((test_data$y - lasso_predictions)^2)
mse
```

```
## [1] 9.423881
```

MSE = 9.42

#3n

```r
source("hw5.R")

hw5(rho = 0.9, sigmae = 5)
```

```
## -------------------------------------------
## correlation between x:  0.9
## Error variance:  25
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0727  -3.4071   0.4994   3.0555  11.5877
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0427     0.5491   5.541 2.73e-07 ***
## X1            0.9064     1.4791   0.613   0.5415
## X2           -2.0427     1.6748  -1.220   0.2256
## X3            3.6465     1.4873   2.452   0.0161 *
## X4           -0.8389     1.7398  -0.482   0.6308
## X5           -0.2851     1.3606  -0.210   0.8345
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.333 on 94 degrees of freedom
## Multiple R-squared:  0.1186, Adjusted R-squared:  0.07173
## F-statistic:  2.53 on 5 and 94 DF,  p-value: 0.03405
##
## OLS x1-x5: 26.24862
## OLS x1-x15: 28.4299
## backward ( 2 ): 26.96192
## forward ( 2 ): 26.96192
```

```r
hw5(rho = 0.9, sigmae = 3)
```

```
## -------------------------------------------
## correlation between x:  0.9
## Error variance:  9
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8436 -2.0442  0.2997  1.8333  6.9526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0256     0.3295   9.183    1e-14 ***
## X1            0.9439     0.8875   1.064  0.29026
## X2           -1.6256     1.0049  -1.618  0.10906
## X3            2.7879     0.8924   3.124  0.00237 **
## X4           -0.3034     1.0439  -0.291  0.77200
## X5           -0.3711     0.8164  -0.455  0.65048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.2 on 94 degrees of freedom
## Multiple R-squared:  0.2407, Adjusted R-squared:  0.2003
## F-statistic: 5.961 on 5 and 94 DF,  p-value: 7.843e-05
##
## OLS x1-x5: 9.449501
## OLS x1-x15: 10.23477
## backward ( 4 ): 10.04426
## forward ( 2 ): 9.879715
```

```r
hw5(rho = 0.9, sigmae = 1)
```

```
## --------------------------------------------
## correlation between x:  0.9
## Error variance:  1
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.61454 -0.68141  0.09989  0.61110  2.31754
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0085     0.1098  27.395  < 2e-16 ***
## X1            0.9813     0.2958   3.317 0.001294 **
## X2           -1.2085     0.3350  -3.608 0.000497 ***
## X3            1.9293     0.2975   6.486 4.07e-09 ***
## X4            0.2322     0.3480   0.667 0.506172
## X5           -0.4570     0.2721  -1.680 0.096373 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.067 on 94 degrees of freedom
## Multiple R-squared:  0.7193, Adjusted R-squared:  0.7044
## F-statistic: 48.18 on 5 and 94 DF,  p-value: < 2.2e-16
##
## OLS x1-x5: 1.049945
## OLS x1-x15: 1.137196
## backward ( 4 ): 1.14975
## forward ( 4 ): 1.14975
```

```r
hw5(rho = 0.5, sigmae = 5)
```

```
## --------------------------------------------
## correlation between x:  0.5
## Error variance:  25
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0727  -3.4071   0.4994   3.0555  11.5877
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0427     0.5491   5.541 2.73e-07 ***
## X1            0.8773     0.6134   1.430 0.155995
## X2           -1.4325     0.7637  -1.876 0.063802 .
## X3            2.4677     0.6751   3.655 0.000423 ***
```

```
## X4           -0.1112     0.7949  -0.140 0.889065
## X5           -0.4022     0.6193  -0.649 0.517672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.333 on 94 degrees of freedom
## Multiple R-squared:  0.1641, Adjusted R-squared:  0.1197
## F-statistic: 3.691 on 5 and 94 DF,  p-value: 0.004288
##
## OLS x1-x5: 26.24862
## OLS x1-x15: 28.4299
## backward ( 4 ): 27.28365
## forward ( 2 ): 26.96062
```

```
hw5(rho = 0.5, sigmae = 3)
```

```
## -------------------------------------------
## correlation between x:  0.5
## Error variance:  9
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8436 -2.0442  0.2997  1.8333  6.9526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0256     0.3295   9.183 1.00e-14 ***
## X1            0.9264     0.3681   2.517  0.01353 *
## X2           -1.2595     0.4582  -2.749  0.00718 **
## X3            2.0806     0.4051   5.136 1.51e-06 ***
## X4            0.1333     0.4769   0.279  0.78050
## X5           -0.4413     0.3716  -1.188  0.23799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.2 on 94 degrees of freedom
## Multiple R-squared:  0.3092, Adjusted R-squared:  0.2724
## F-statistic: 8.414 on 5 and 94 DF,  p-value: 1.31e-06
##
## OLS x1-x5: 9.449501
## OLS x1-x15: 10.23477
## backward ( 4 ): 10.06145
## forward ( 4 ): 10.06145
```

```
hw5(rho = 0.5, sigmae = 1)
```

```
## -------------------------------------------
## correlation between x:  0.5
## Error variance:  1
##
```

```
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.61454 -0.68141  0.09989  0.61110  2.31754
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0085     0.1098  27.395  < 2e-16 ***
## X1            0.9755     0.1227   7.951 4.05e-12 ***
## X2           -1.0865     0.1527  -7.113 2.21e-10 ***
## X3            1.6935     0.1350  12.542  < 2e-16 ***
## X4            0.3778     0.1590   2.376 0.019524 *
## X5           -0.4804     0.1239  -3.879 0.000195 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.067 on 94 degrees of freedom
## Multiple R-squared:  0.771,  Adjusted R-squared:  0.7588
## F-statistic: 63.31 on 5 and 94 DF,  p-value: < 2.2e-16
##
## OLS x1-x5: 1.049945
## OLS x1-x15: 1.137196
## backward ( 6 ): 1.093831
## forward ( 6 ): 1.093831
```

```r
hw5(rho = 0.1, sigmae = 5)
```

```
## ------------------------------------------
## correlation between x:  0.1
## Error variance:  25
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0727  -3.4071   0.4994   3.0555  11.5877
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.04265    0.54911   5.541 2.73e-07 ***
## X1           0.88334    0.48641   1.816   0.0726 .
## X2          -1.27395    0.59604  -2.137   0.0352 *
## X3           2.22519    0.53174   4.185 6.42e-05 ***
## X4           0.02041    0.62941   0.032   0.9742
## X5          -0.42305    0.48730  -0.868   0.3875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.333 on 94 degrees of freedom
## Multiple R-squared:  0.1973, Adjusted R-squared:  0.1546
## F-statistic: 4.621 on 5 and 94 DF,  p-value: 0.0008171
```

```
##
## OLS x1-x5: 26.24862
## OLS x1-x15: 28.4299
## backward ( 4 ): 27.49461
## forward ( 4 ): 27.49461
```

```r
hw5(rho = 0.1, sigmae = 3)
```

```
## -------------------------------------------
## correlation between x:  0.1
## Error variance:  9
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8436 -2.0442  0.2997  1.8333  6.9526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0256     0.3295   9.183 1.00e-14 ***
## X1            0.9300     0.2918   3.187  0.00195 **
## X2           -1.1644     0.3576  -3.256  0.00157 **
## X3            1.9351     0.3190   6.065 2.74e-08 ***
## X4            0.2122     0.3776   0.562  0.57544
## X5           -0.4538     0.2924  -1.552  0.12398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.2 on 94 degrees of freedom
## Multiple R-squared:  0.3608, Adjusted R-squared:  0.3268
## F-statistic: 10.61 on 5 and 94 DF,  p-value: 4.24e-08
##
## OLS x1-x5: 9.449501
## OLS x1-x15: 10.23477
## backward ( 4 ): 10.27753
## forward ( 4 ): 10.27753
```

```r
hw5(rho = 0.1, sigmae = 1)
```

```
## -------------------------------------------
## correlation between x:  0.1
## Error variance:  1
##
## Call:
## lm(formula = y ~ X1 + X2 + X3 + X4 + X5, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.61454 -0.68141  0.09989  0.61110  2.31754
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.00853    0.10982  27.395  < 2e-16 ***
## X1           0.97667    0.09728  10.040  < 2e-16 ***
## X2          -1.05479    0.11921  -8.848 5.16e-14 ***
## X3           1.64504    0.10635  15.468  < 2e-16 ***
## X4           0.40408    0.12588   3.210  0.00182 **
## X5          -0.48461    0.09746  -4.972 2.97e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.067 on 94 degrees of freedom
## Multiple R-squared:  0.8089, Adjusted R-squared:  0.7988
## F-statistic:  79.6 on 5 and 94 DF,  p-value: < 2.2e-16
##
## OLS x1-x5: 1.049945
## OLS x1-x15: 1.137196
## backward ( 6 ): 1.095551
## forward ( 6 ): 1.095551
```

Low noise, low multicollinearity and Low noise, moderate multicollinearity tended to perform the best out of all the models. In instances where there is high multicollinearity and it is desired to preserve all the features, it makes sense to use ridge regression because it will shrink coefficients and help prevent overfitting.

Stepwise is useful in stances where multicollinearity is less present as it will help with selecting relevant features in a more simple way.

Finally, when multicollinearity is low, it may make sense to not apply any selection or shrinkage because the models already tend to perform well.

#4a

```
customer <- read.csv('customer2.csv')

customer$logtarg <- log(customer$target + 1)

head(customer)
```

```
##     id train target  logtarg
## 1  957     0  44.94 3.827336
## 2 2062     0   0.00 0.000000
## 3 2232     1   0.00 0.000000
## 4 2623     0   0.00 0.000000
## 5 3000     1   0.00 0.000000
## 6 3689     0   0.00 0.000000
```

```
summary(customer)
```

```
##        id                train            target           logtarg
##  Min.   :     957   Min.   :0.0000   Min.   :  0.000   Min.   :0.0000
##  1st Qu.: 4448960   1st Qu.:0.0000   1st Qu.:  0.000   1st Qu.:0.0000
##  Median : 8090750   Median :0.0000   Median :  0.000   Median :0.0000
##  Mean   : 8563488   Mean   :0.3308   Mean   :  3.241   Mean   :0.2529
##  3rd Qu.:13378724   3rd Qu.:1.0000   3rd Qu.:  0.000   3rd Qu.:0.0000
##  Max.   :16456238   Max.   :1.0000   Max.   :739.480   Max.   :6.6073
```

#4b

```r
library(dplyr)
```

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

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

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

```r
orders <- read.csv('orders.csv')

orders <- orders %>%
  mutate(t = as.numeric(as.Date("2014/11/25") - as.Date(orddate, format = "%d%b%Y")) / 365.25)

head(orders)
```

```
##    id   orddate ordnum category qty     price        t
## 1 957 10FEB2008  38650       35   1  5.010658 6.789870
## 2 957 10FEB2008  38650       35   1 20.426102 6.789870
## 3 957 10FEB2008  38650       19   1 20.400543 6.789870
## 4 957 15MAR2008  48972       40   1 25.539017 6.696783
## 5 957 22NOV2008 150011       40   1 14.316170 6.006845
## 6 957 22NOV2008 150011       40   1  8.589699 6.006845
```

```r
summary(orders)
```

```
##        id              orddate              ordnum           category
##  Min.   :     957   Length:353687      Min.   :   1018   Min.   : 1.00
##  1st Qu.: 3929256   Class :character   1st Qu.: 397351   1st Qu.:14.00
##  Median : 6353495   Mode  :character   Median : 728198   Median :20.00
##  Mean   : 6791632                      Mean   : 692588   Mean   :32.55
##  3rd Qu.: 8720240                      3rd Qu.:1004519   3rd Qu.:36.00
##  Max.   :16456238                      Max.   :1256189   Max.   :99.00
##       qty               price                t
##  Min.   :    0.00   Min.   :   0.000   Min.   :0.002738
##  1st Qu.:    1.00   1st Qu.:   5.113   1st Qu.:1.229295
##  Median :    1.00   Median :   8.666   Median :2.729637
##  Mean   :    1.12   Mean   :  11.495   Mean   :2.958282
##  3rd Qu.:    1.00   3rd Qu.:  12.782   3rd Qu.:4.528405
##  Max.   :35026.00   Max.   :5010.660   Max.   :7.058179
```

#4c

```r
# Calculate "tof" (time on file) as the maximum value of "t" for each customer
tof <- orders %>%
  group_by(id) %>%
  summarize(tof = max(t))


r <- orders %>%
  arrange(id, t) %>%
  group_by(id) %>%
  filter(!duplicated(orddate)) %>%
  mutate(r = ifelse(is.na(t - lag(t)), 0, t - lag(t))) %>%
  ungroup()

# Calculate "f" (frequency) as the count of distinct order numbers for each customer
f <- orders %>%
  group_by(id) %>%
  summarize(f = n_distinct(ordnum))

# Calculate "m" (monetary) as the sum of the product of "price" and "qty" for each customer
m <- orders %>%
  group_by(id) %>%
  summarize(m = sum(price * qty))

# Merge the calculated variables into a single "RFM" table
RFM <- tof %>%
  inner_join(r, by = "id") %>%
  inner_join(f, by = "id") %>%
  inner_join(m, by = "id")


head(RFM)
```

```
## # A tibble: 6 x 11
##      id   tof orddate    ordnum category   qty price     t      r     f     m
##   <int> <dbl> <chr>       <int>    <int> <int> <dbl> <dbl>  <dbl> <int> <dbl>
## 1   957  6.79 29JUL2014 1191182       37     1  7.95 0.326 0         14  396.
## 2   957  6.79 19JUL2014 1185048        5     1  5    0.353 0.0274    14  396.
## 3   957  6.79 27JUL2013  979370       44     1 10    1.33  0.977     14  396.
## 4   957  6.79 20FEB2013  905635       26     1  5.90 1.76  0.430     14  396.
## 5   957  6.79 28JUL2012  786021       20     2 12.9  2.33  0.567     14  396.
## 6   957  6.79 19JUN2012  771540       19     1  7.95 2.43  0.107     14  396.
```

```r
summary(RFM)
```

```
##        id                 tof               orddate              ordnum
##  Min.   :     957   Min.   :0.002738   Length:101890      Min.   :   1018
##  1st Qu.: 3887200   1st Qu.:3.646817   Class :character   1st Qu.: 364750
##  Median : 6109373   Median :5.831622   Mode  :character   Median : 689970
##  Mean   : 6677319   Mean   :5.005597                      Mean   : 669095
##  3rd Qu.: 8689822   3rd Qu.:6.789870                      3rd Qu.: 982021
##  Max.   :16456238   Max.   :7.058179                      Max.   :1256189
##     category          qty               price               t
##  Min.   : 1.00   Min.   :  0.000   Min.   :   0.00   Min.   :0.002738
```

```
##  1st Qu.:14.00   1st Qu.:  1.000   1st Qu.:   6.95   1st Qu.:1.322382
##  Median :20.00   Median :  1.000   Median :   9.95   Median :2.959617
##  Mean   :32.65   Mean   :  1.036   Mean   :  13.92   Mean   :3.087835
##  3rd Qu.:37.00   3rd Qu.:  1.000   3rd Qu.:  15.24   3rd Qu.:4.714579
##  Max.   :99.00   Max.   :100.000   Max.   :5010.66   Max.   :7.058179
##        r                 f                m
##  Min.   :0.00000   Min.   :  1.00   Min.   :    0.0
##  1st Qu.:0.03833   1st Qu.:  6.00   1st Qu.:  168.1
##  Median :0.16701   Median : 11.00   Median :  361.5
##  Mean   :0.36928   Mean   : 15.66   Mean   :  710.0
##  3rd Qu.:0.45722   3rd Qu.: 20.00   3rd Qu.:  743.7
##  Max.   :6.89665   Max.   :160.00   Max.   :41029.9
```

#4d

```r
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# Join the customer and RFM tables
merged_data <- inner_join(customer, RFM, by = "id")

# Subset the data to include only the training data (where train = 1)
training_data <- merged_data %>% filter(train == 1)
test_data <- merged_data %>% filter(train == 0)

# Perform the regression
model <- lm(logtarg ~ log(tof + .00001) + log(r + .00001) + log(f + .00001) + log(m + 1 + .00001), data

# Show a summary of the fitted model
summary(model)
```

```
##
## Call:
## lm(formula = logtarg ~ log(tof + 1e-05) + log(r + 1e-05) + log(f +
##     1e-05) + log(m + 1 + 1e-05), data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1044 -0.6469 -0.3913 -0.0575  5.7826
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.511178   0.044922 -11.379   < 2e-16 ***
## log(tof + 1e-05)    -0.334292   0.011960 -27.950   < 2e-16 ***
```

```
## log(r + 1e-05)     -0.006921   0.001835  -3.771 0.000163 ***
## log(f + 1e-05)      0.319108   0.014383  22.186  < 2e-16 ***
## log(m + 1 + 1e-05)  0.124147   0.011473  10.821  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.207 on 34167 degrees of freedom
## Multiple R-squared:  0.082,  Adjusted R-squared:  0.08189
## F-statistic:   763 on 4 and 34167 DF,  p-value: < 2.2e-16
```

#4e

```r
test_predictions <- predict(model, newdata = test_data)

# Calculate the squared errors
squared_errors <- (test_data$logtarg - test_predictions)^2

# Calculate the mean squared error (MSE)
mse <- mean(squared_errors)
mse
```

```
## [1] 1.420175
```

MSE = 1.42

#5a

```r
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library("psych")
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```r
crime_data <- read.csv("bike.csv")

crime_data2 <- crime_data[,c(4, 5, 6, 7, 8, 11, 13, 22, 24, 34, 43, 45)]

#pairs.panels(crime_data2,
#             ellipses = FALSE)


colnames(crime_data2)
```

```
##  [1] "CTA_TRAIN_STATIONS"         "BIKE_ROUTES"
##  [3] "Limited_Business_License"   "Retail_Food_Establishment"
##  [5] "CAPACITY"                   "CBD"
##  [7] "EDU"                        "DECEPTIVE_PRACTICE"
##  [9] "HOMICIDE"                   "OFFENSE_INVOLVING_CHILDREN"
## [11] "THEFT"                      "trips"
```

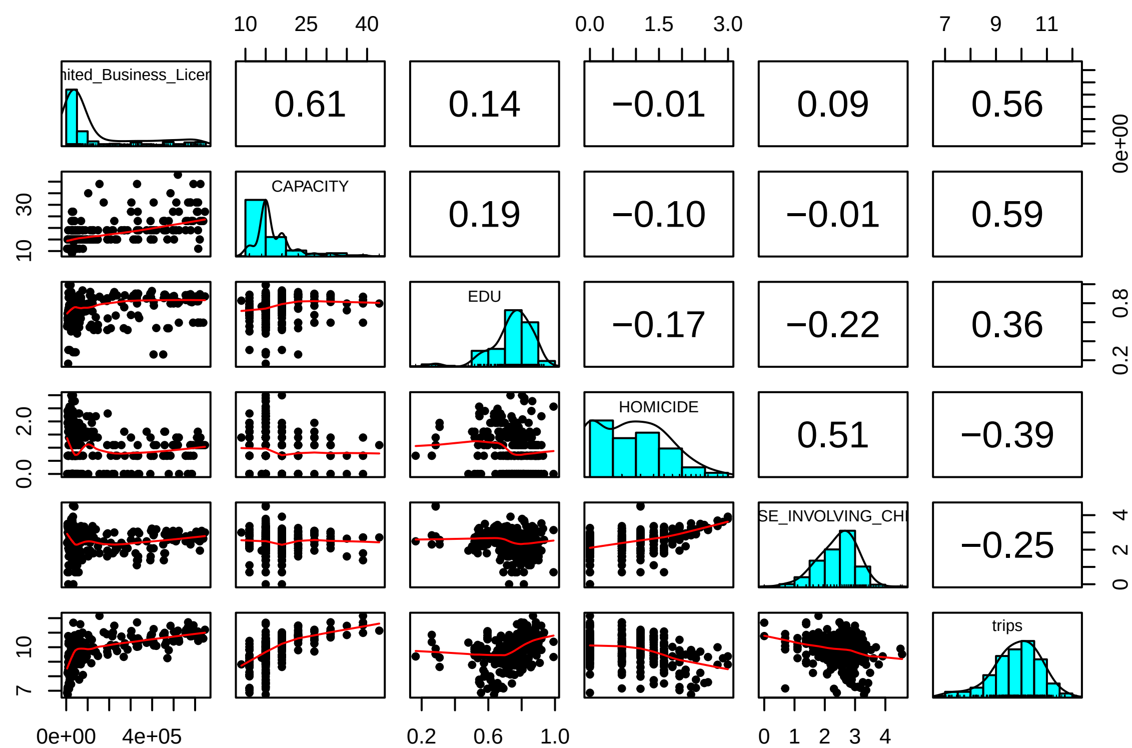
```r
crime_data3 <- crime_data2[, c(3, 5, 7, 9, 10, 12)]

crime_model <- lm(trips ~ ., data = crime_data3)
summary(crime_model)
```

```
##
## Call:
## lm(formula = trips ~ ., data = crime_data3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37918 -0.33800  0.05101  0.41554  1.56899
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 8.162e+00  3.041e-01  26.842  < 2e-16 ***
## Limited_Business_License    1.755e-06  2.453e-07   7.155 6.70e-12 ***
## CAPACITY                    5.767e-02  8.600e-03   6.707 1.02e-10 ***
## EDU                         1.370e+00  2.997e-01   4.572 7.13e-06 ***
## HOMICIDE                   -3.419e-01  5.578e-02  -6.129 2.83e-09 ***
## OFFENSE_INVOLVING_CHILDREN -1.365e-01  6.470e-02  -2.110   0.0357 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6277 on 294 degrees of freedom
## Multiple R-squared:  0.5792, Adjusted R-squared:  0.572
## F-statistic: 80.92 on 5 and 294 DF,  p-value: < 2.2e-16
```

```r
drop1(crime_model)
```

```
## Single term deletions
##
## Model:
## trips ~ Limited_Business_License + CAPACITY + EDU + HOMICIDE +
##     OFFENSE_INVOLVING_CHILDREN
##                            Df Sum of Sq    RSS     AIC
## <none>                                  115.85 -273.44
## Limited_Business_License    1   20.1722 136.02 -227.29
## CAPACITY                    1   17.7234 133.57 -232.74
## EDU                         1    8.2363 124.09 -254.84
## HOMICIDE                    1   14.8036 130.66 -239.37
## OFFENSE_INVOLVING_CHILDREN  1    1.7536 117.61 -270.94
```

```r
pairs.panels(crime_data3,
             ellipses = FALSE)
```




```r
vif(crime_model)
```

```
##   Limited_Business_License                   CAPACITY
##                   1.613528                   1.623234
##                        EDU                   HOMICIDE
##                   1.098626                   1.376734
```

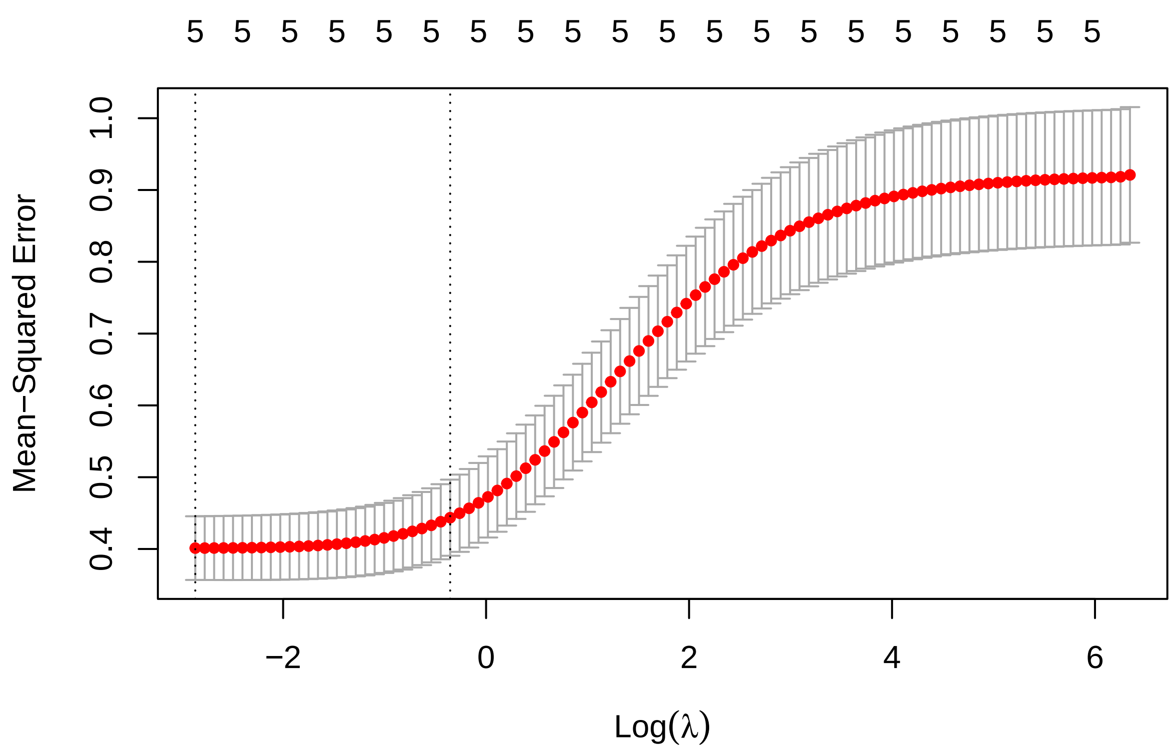
```
## OFFENSE_INVOLVING_CHILDREN
##                    1.422630
```

We arrived at our selection of features by using the following procedure: select the predictors which have the highest correlation with y, create a scatter plot matrix of them, and prune the matrix of features that have high multi-collinearity with other features. Doing this yielded a model with higher significance and R^2 than anything else we tried, including aggregating categories and applying interaction terms.

Number of businesses, capacity, and education all had positive coefficients. A neighborhood with a high number of businesses might have more attractions that are worth biking to. A neighborhood with a large capacity might mean a bike is needed to get around more easily. With respect to education, more educated people tend to live in more affluent neighborhoods, such as Evanston, which tend to have more bike-friendly infrastructure.

#5b

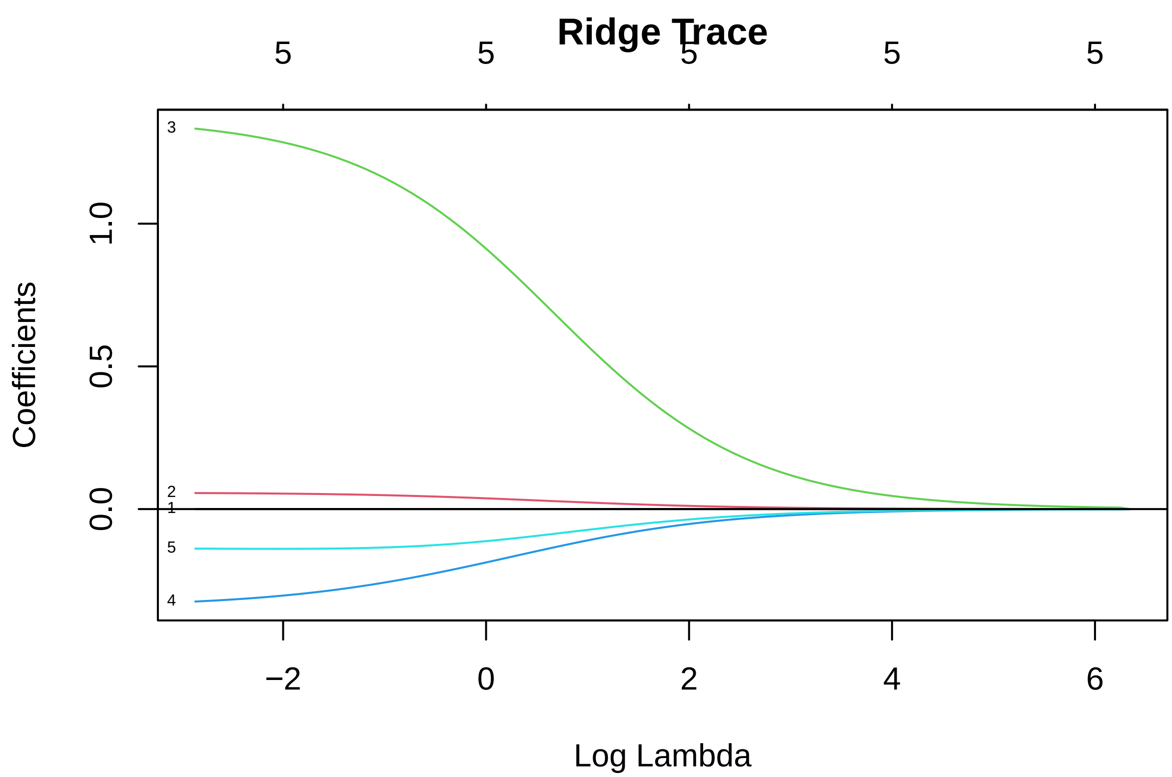
```
X <- as.matrix(crime_data3[,-6])
y <- as.numeric(crime_data3$trips)

cv_params <- cv.glmnet(X,y, alpha = 0)
plot(cv_params)
```



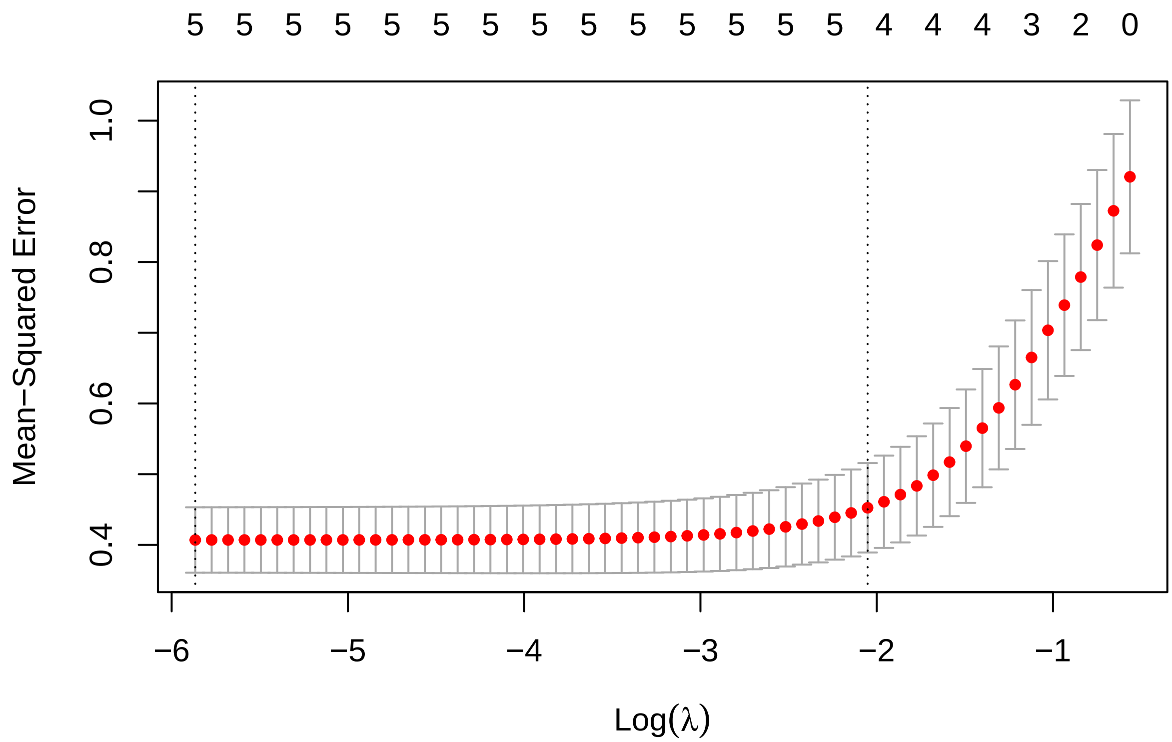

```
best_lambda <- cv_params$lambda.min

fit <- glmnet(X, y, alpha = 0, lambda = best_lambda)
summary(fit)
```

```
##            Length Class     Mode
## a0         1      -none-    numeric
## beta       5      dgCMatrix S4
## df         1      -none-    numeric
## dim        2      -none-    numeric
## lambda     1      -none-    numeric
## dev.ratio  1      -none-    numeric
## nulldev    1      -none-    numeric
## npasses    1      -none-    numeric
## jerr       1      -none-    numeric
## offset     1      -none-    logical
## call       5      -none-    call
## nobs       1      -none-    numeric
```

```
plot(cv_params$glmnet.fit, xvar = 'lambda', label = TRUE, main="Ridge Trace"); abline(h=0)
```



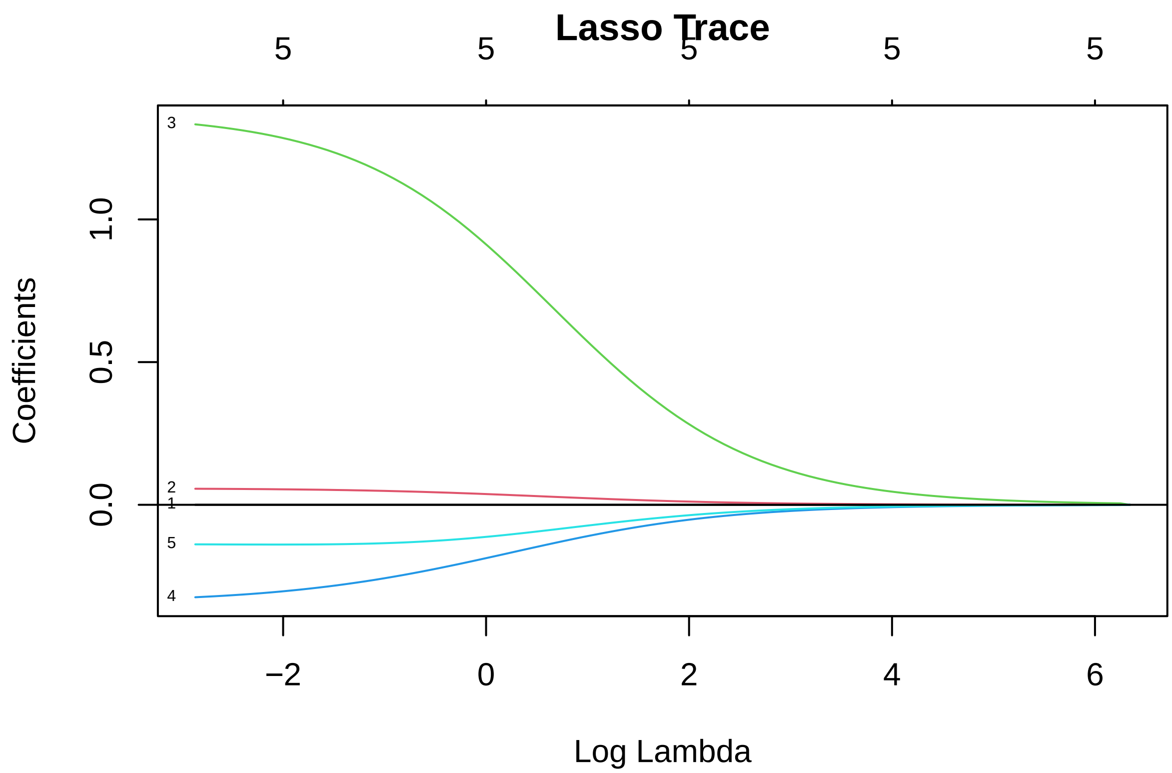

```
X <- as.matrix(crime_data3[,-6])
y <- as.numeric(crime_data3$trips)

cvfit <- cv.glmnet(X, y, alpha = 1)  # Alpha = 1 for lasso
plot(cvfit)
```




```r
best_lambda <- cvfit$lambda.min

fit <- glmnet(X, y, alpha = 1, lambda = best_lambda)
summary(fit)
```

```
##           Length Class     Mode
## a0        1      -none-    numeric
## beta      5      dgCMatrix S4
## df        1      -none-    numeric
## dim       2      -none-    numeric
## lambda    1      -none-    numeric
## dev.ratio 1      -none-    numeric
## nulldev   1      -none-    numeric
## npasses   1      -none-    numeric
## jerr      1      -none-    numeric
## offset    1      -none-    logical
## call      5      -none-    call
## nobs      1      -none-    numeric
```

```r
plot(cv_params$glmnet.fit, xvar = 'lambda', label = TRUE, main="Lasso Trace"); abline(h=0)
```


Lasso Trace
5
5
5
5
5
Coefficients
1.0
0.5
0.0
3
2
1
5
4
−2
0
2
4
6
Log Lambda