Linear Regression practical

IMohorianu

15/02/2021

Prerequisites

```
library("ISLR")
library("MASS")
library("ggplot2")
library("GGally")
library("gridExtra")
library("corrgram")
```

Simple Linear Regression

Perform a full analysis on the Boston dataset.

Exploring dataset

```
?Boston
```

```
## starting httpd help server ... done
```

```
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
summary(Boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
```

```
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Training linear model on lstat feature

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

```
##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
names(lm.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
lm.fit$coefficients
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

```
confint(lm.fit)
```

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

```
predict(lm.fit, data.frame(lstat=(c(5,10,15))), interval ="confidence")
```

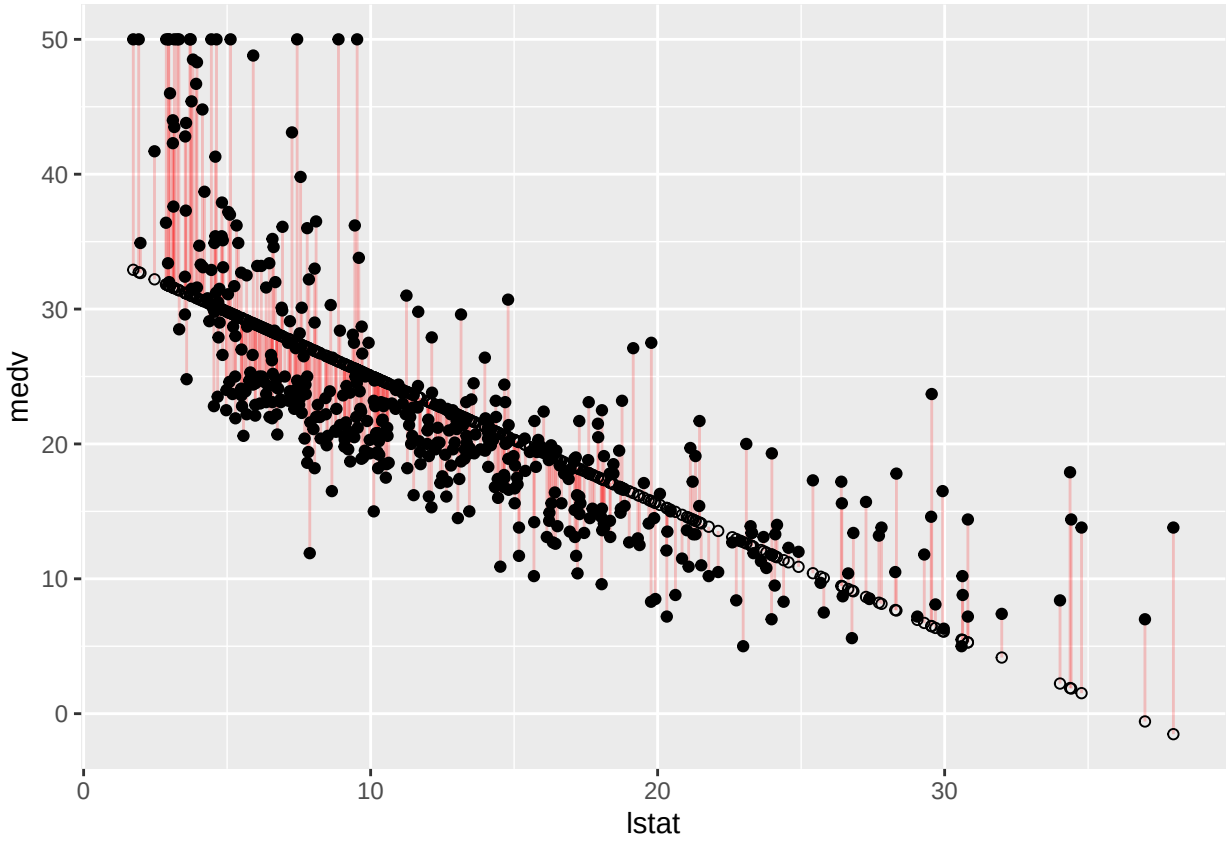
```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

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

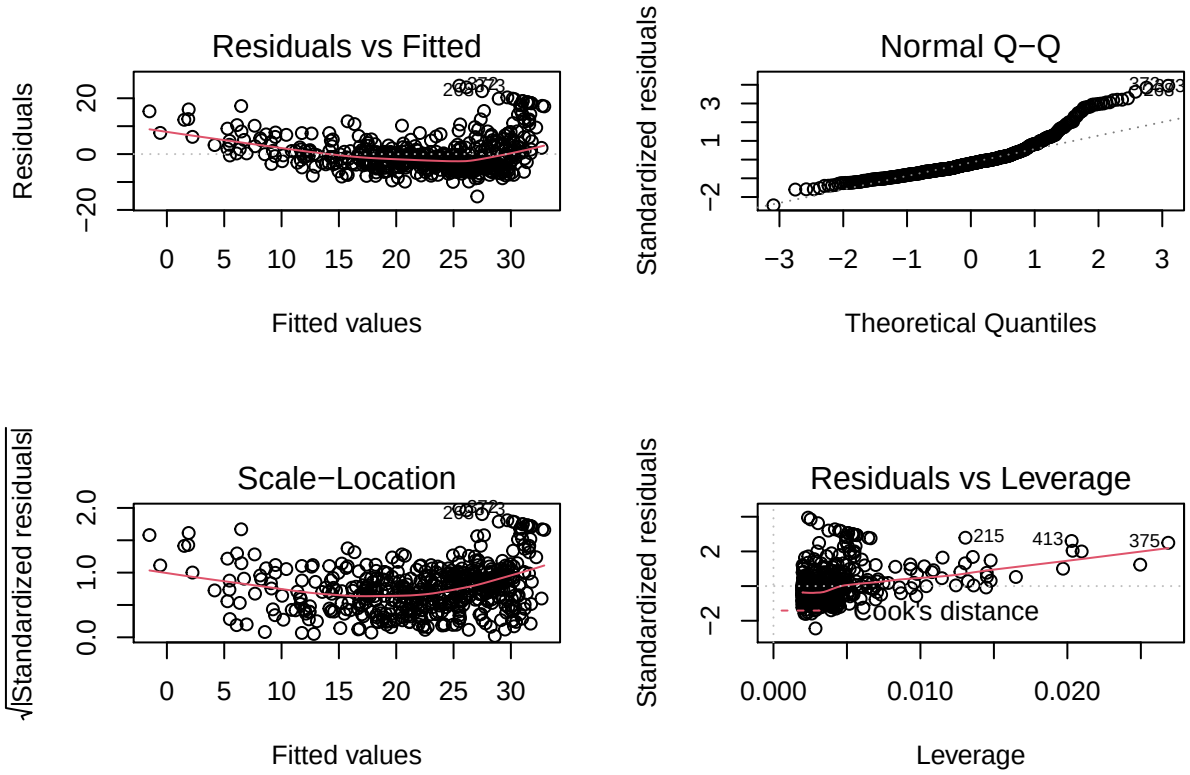
```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

```
my.boston = Boston
my.boston$predicted <- predict(lm.fit)   # Save the predicted values
my.boston$residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = lstat, y = medv)) +
  geom_segment(aes(xend = lstat, yend = predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = predicted), shape = 1)
```




```
par(mfrow=c(2,2))
plot(lm.fit)
```



Testing other predictors

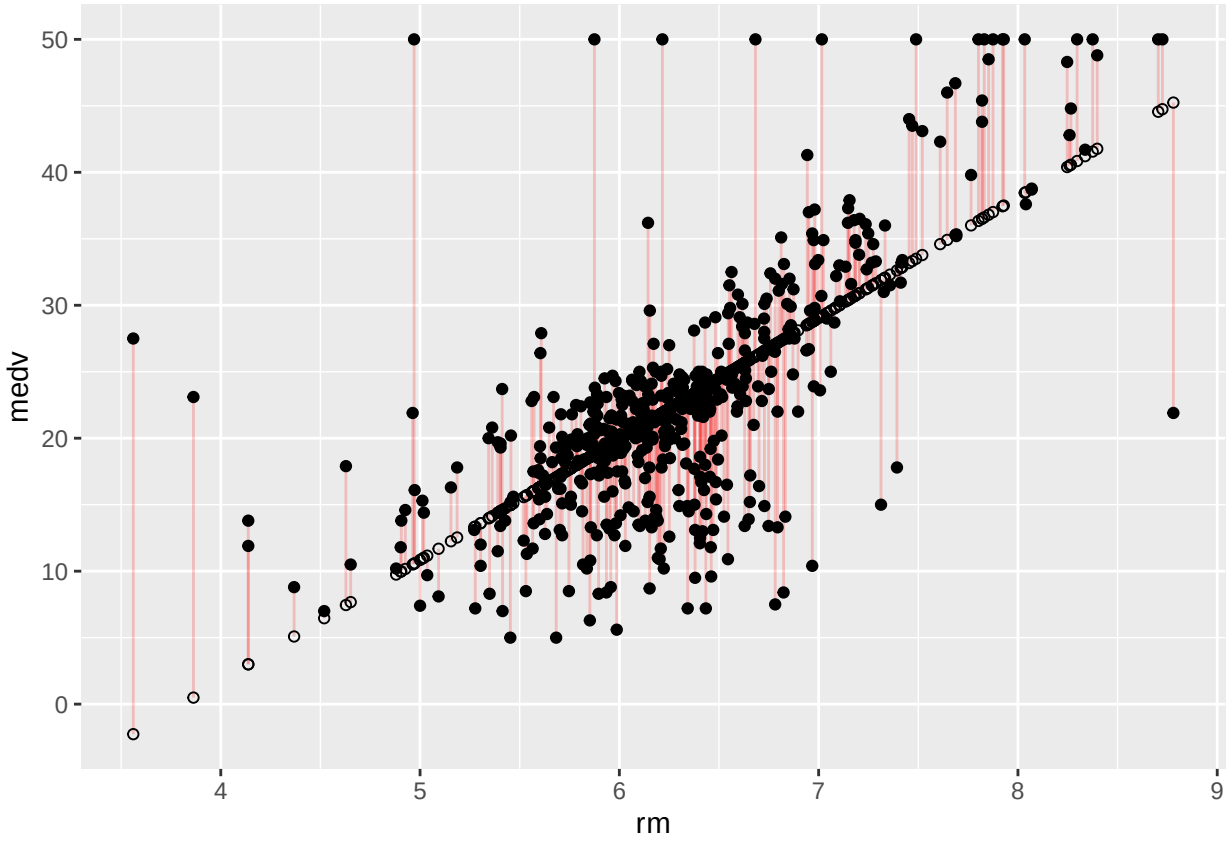
```
pdf("Boston_data.pdf", width = 20, height = 20)
ggpairs(Boston)
dev.off()
```

```
## pdf
##   2
```

Try two other predictors, rm and age.

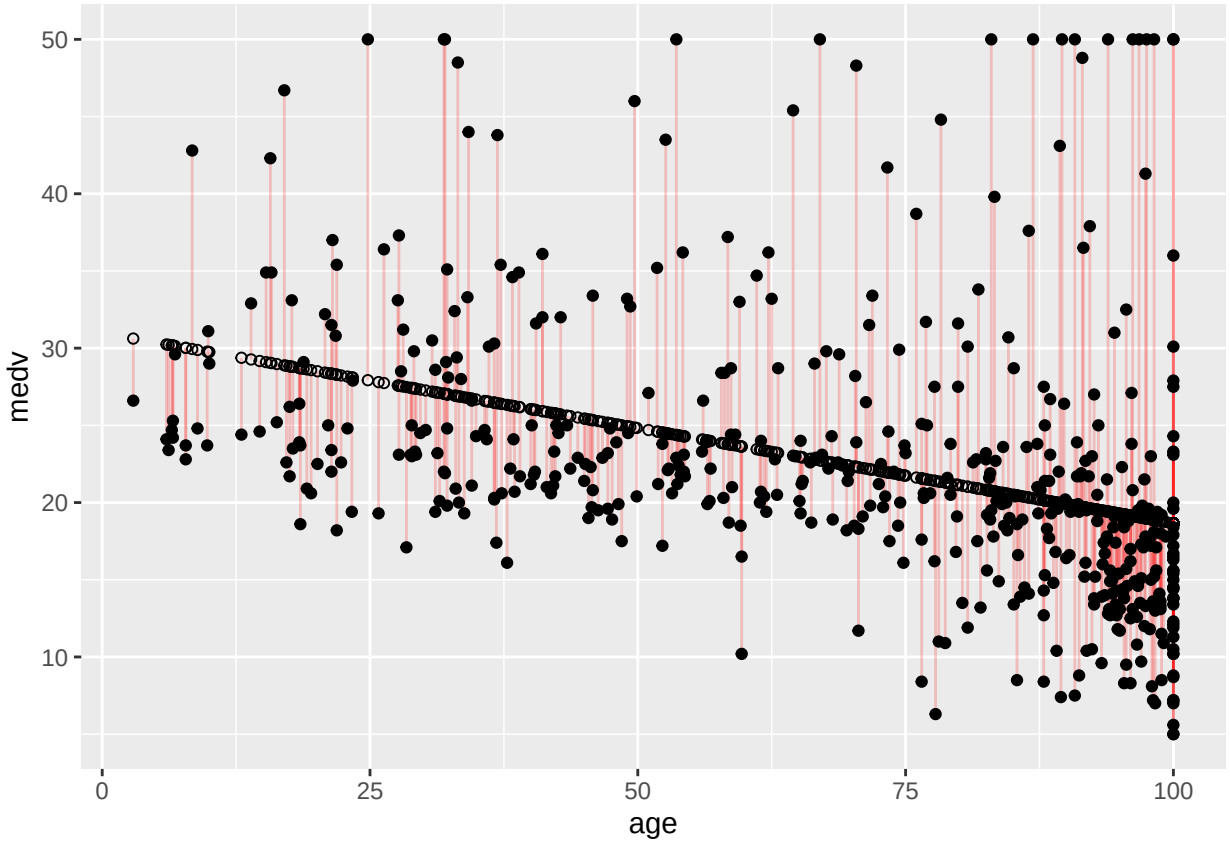
```
lm.fit = lm(medv ~ rm, data = my.boston)
my.boston$predicted <- predict(lm.fit)   # Save the predicted values
my.boston$residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = rm, y = medv)) +
  geom_segment(aes(xend = rm, yend = predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = predicted), shape = 1)
```




```
lm.fit = lm(medv ~ age, data = my.boston)
my.boston$predicted <- predict(lm.fit)   # Save the predicted values
my.boston$residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = age, y = medv)) +
  geom_segment(aes(xend = age, yend = predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = predicted), shape = 1)
```



Multiple Linear Regression

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

```
##
## Call:
## lm(formula = medv ~ ., data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
```

```
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Assess the individual predictors. Attempt a forward/backward selection. Discuss the model.

Forward selection

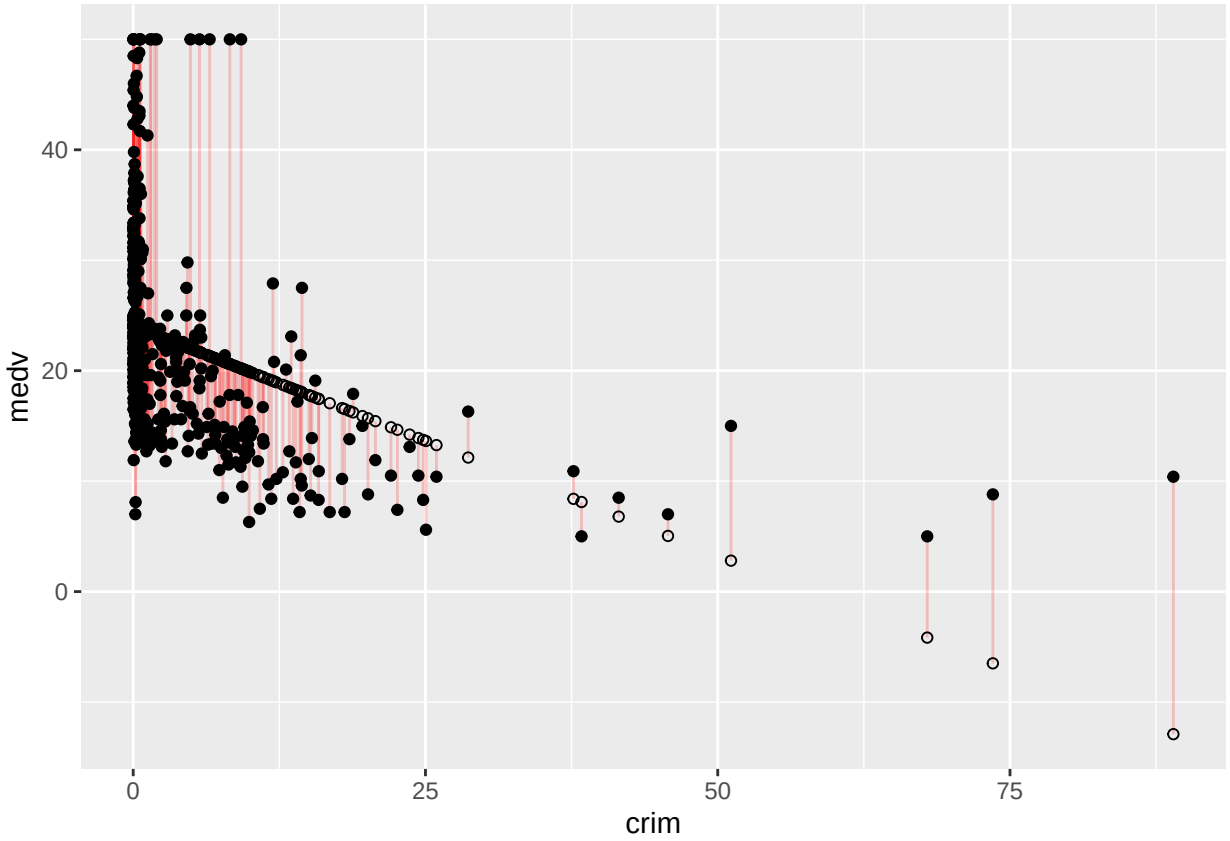
Try all individual predictors

```
lm.fit = lm(medv ~ crim, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ crim, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.957  -5.449  -2.007   2.512  29.800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.03311    0.40914   58.74   <2e-16 ***
## crim        -0.41519    0.04389   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.484 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = crim, y = medv)) +
  geom_segment(aes(xend = crim, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```

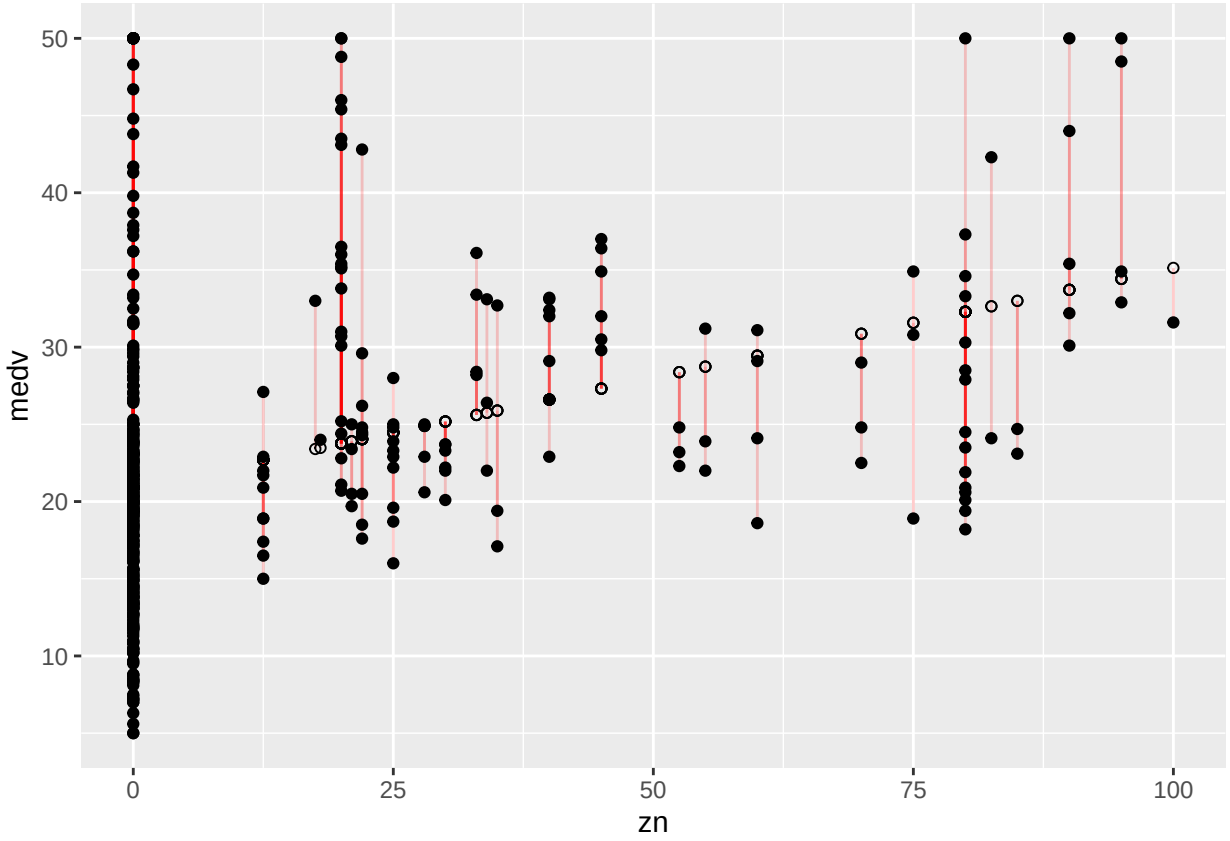



```
lm.fit = lm(medv ~ zn, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ zn, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.918  -5.518  -1.006   2.757  29.082
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.91758    0.42474  49.248   <2e-16 ***
## zn           0.14214    0.01638   8.675   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.587 on 504 degrees of freedom
## Multiple R-squared:  0.1299, Adjusted R-squared:  0.1282
## F-statistic: 75.26 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```
ggplot(my.boston, aes(x = zn, y = medv)) +
  geom_segment(aes(xend = zn, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



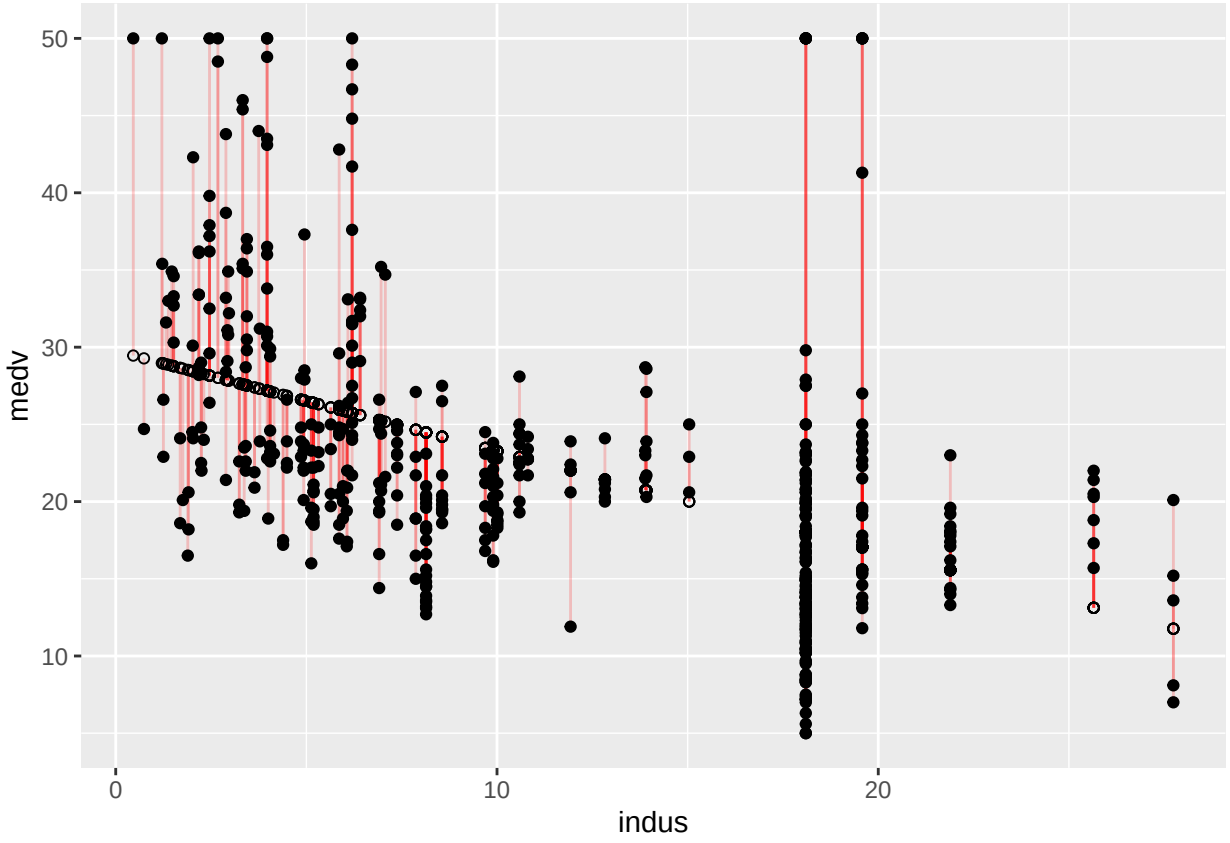

```
lm.fit = lm(medv ~ indus, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ indus, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.017  -4.917  -1.457   3.180  32.943
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.75490    0.68345   43.54   <2e-16 ***
## indus       -0.64849    0.05226  -12.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.234,  Adjusted R-squared:  0.2325
```

```
## F-statistic:   154 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = indus, y = medv)) +
  geom_segment(aes(xend = indus, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



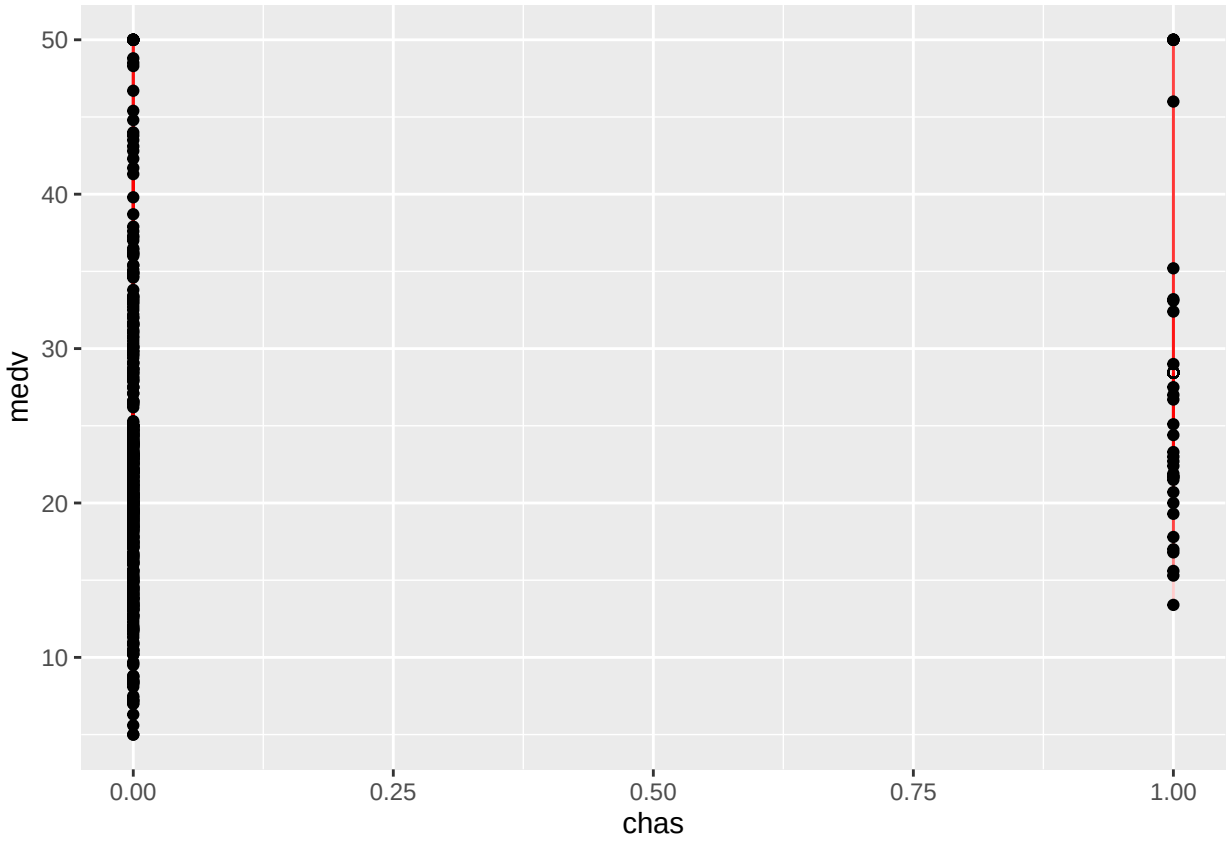

```
lm.fit = lm(medv ~ chas, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ chas, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.094  -5.894  -1.417   2.856  27.906
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.0938     0.4176  52.902  < 2e-16 ***
## chas          6.3462     1.5880   3.996 7.39e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.064 on 504 degrees of freedom
## Multiple R-squared:  0.03072,    Adjusted R-squared:  0.02879
## F-statistic: 15.97 on 1 and 504 DF,  p-value: 7.391e-05
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = chas, y = medv)) +
  geom_segment(aes(xend = chas, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



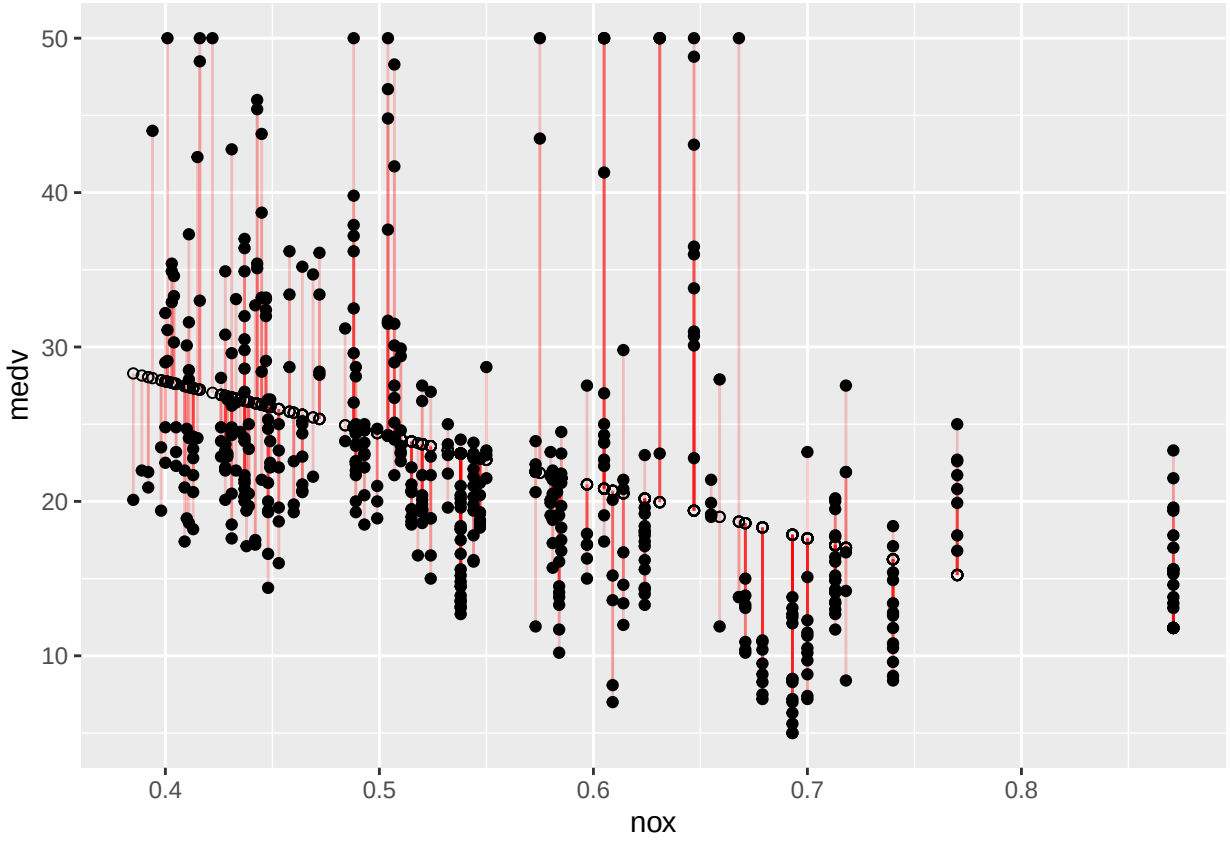

```
lm.fit = lm(medv ~ nox, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ nox, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.691  -5.121  -2.161   2.959  31.310
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.346      1.811   22.83   <2e-16 ***
## nox          -33.916      3.196  -10.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.323 on 504 degrees of freedom
## Multiple R-squared:  0.1826, Adjusted R-squared:  0.181
## F-statistic: 112.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = nox, y = medv)) +
  geom_segment(aes(xend = nox, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



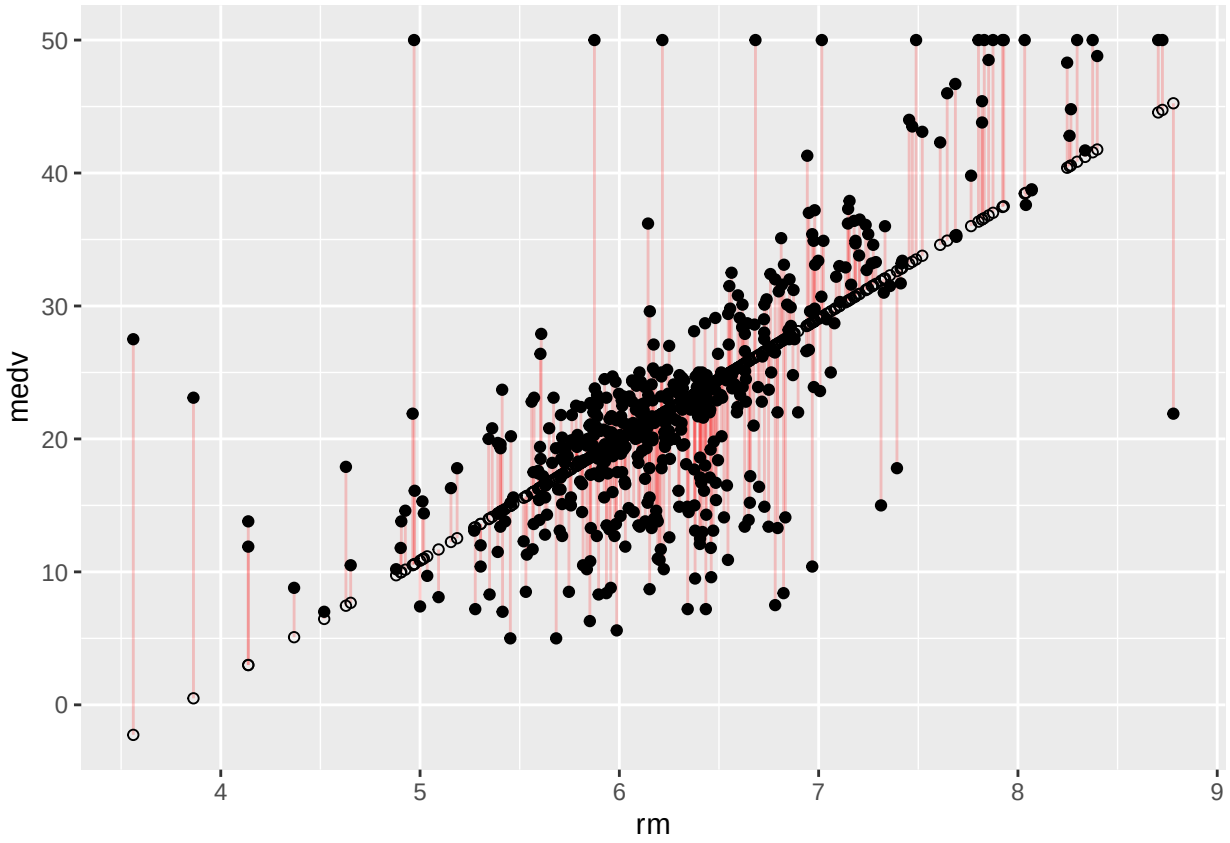

```
lm.fit = lm(medv ~ rm, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
```

```
## lm(formula = medv ~ rm, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.346  -2.547   0.090   2.986  39.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -34.671      2.650  -13.08   <2e-16 ***
## rm             9.102      0.419   21.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.616 on 504 degrees of freedom
## Multiple R-squared:  0.4835, Adjusted R-squared:  0.4825
## F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = rm, y = medv)) +
  geom_segment(aes(xend = rm, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```

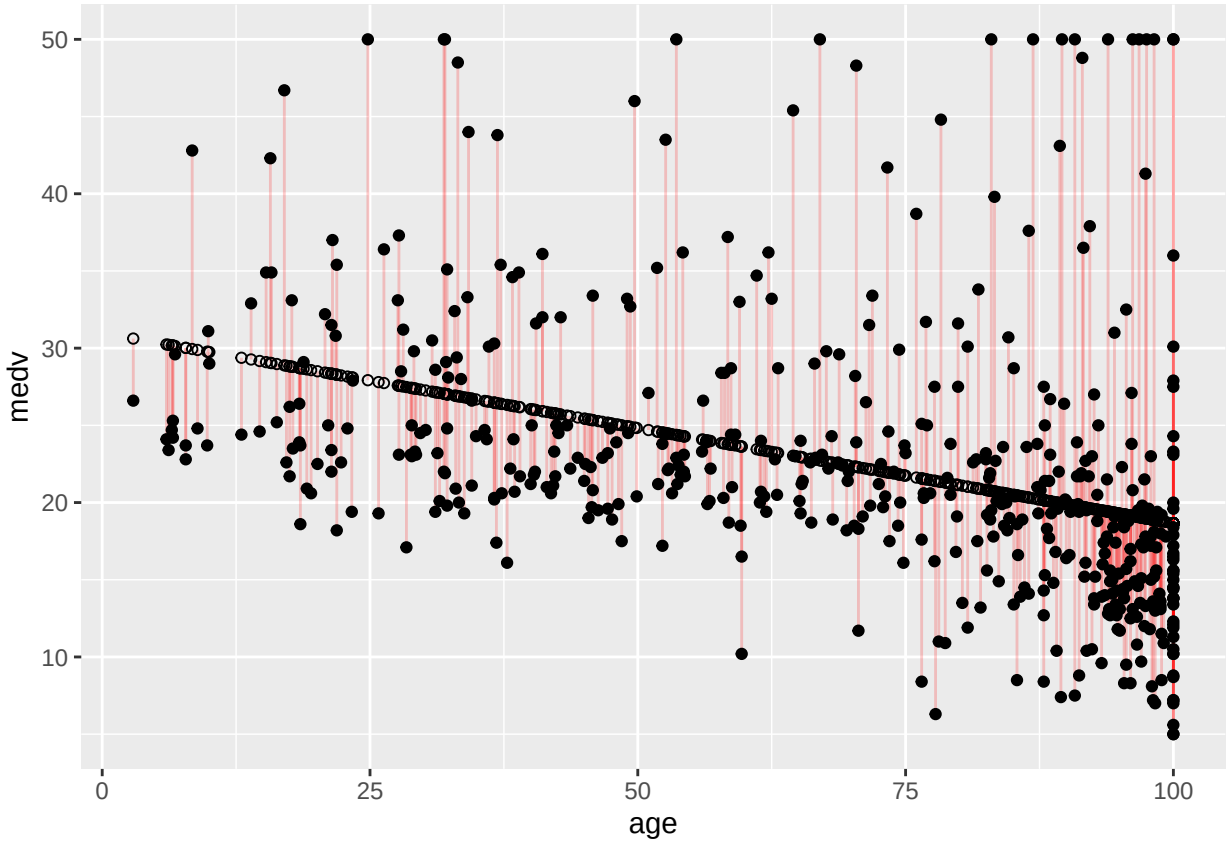


```r
lm.fit = lm(medv ~ age, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ age, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.097  -5.138  -1.958   2.397  31.338
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.97868    0.99911  31.006   <2e-16 ***
## age         -0.12316    0.01348  -9.137   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.527 on 504 degrees of freedom
## Multiple R-squared:  0.1421, Adjusted R-squared:  0.1404
## F-statistic: 83.48 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = age, y = medv)) +
  geom_segment(aes(xend = age, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```

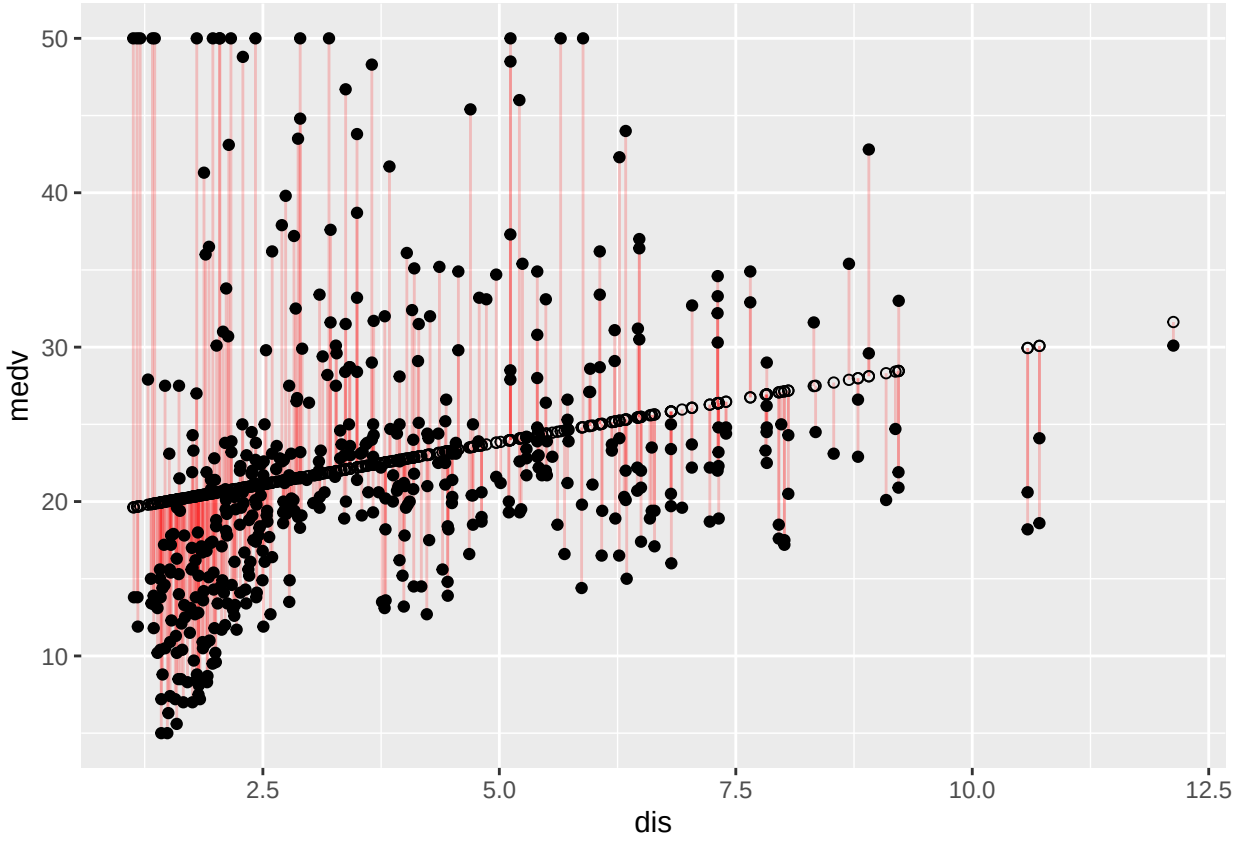



```
lm.fit = lm(medv ~ dis, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ dis, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.016  -5.556  -1.865   2.288  30.377
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.3901     0.8174  22.499  < 2e-16 ***
## dis           1.0916     0.1884   5.795 1.21e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 504 degrees of freedom
## Multiple R-squared:  0.06246,    Adjusted R-squared:  0.0606
## F-statistic: 33.58 on 1 and 504 DF,  p-value: 1.207e-08
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```
ggplot(my.boston, aes(x = dis, y = medv)) +
  geom_segment(aes(xend = dis, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



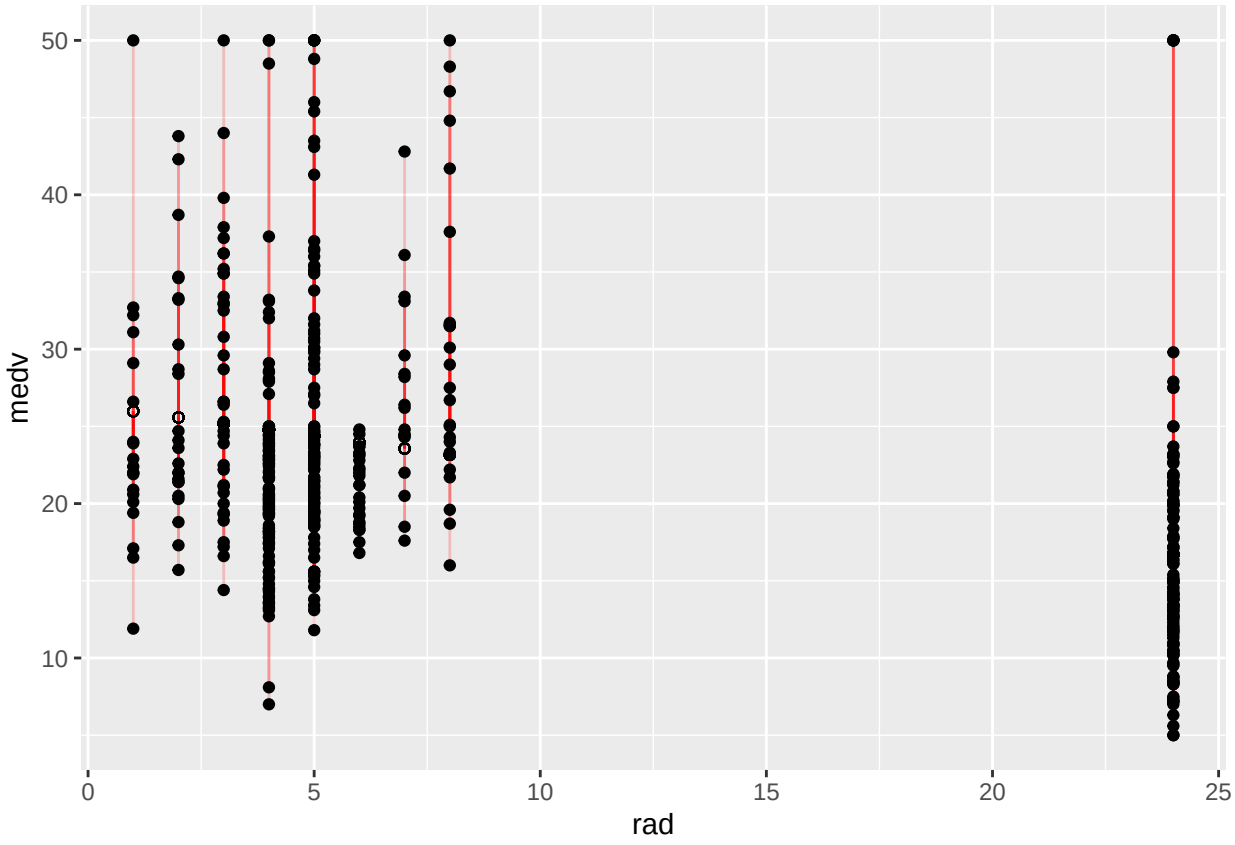

```
lm.fit = lm(medv ~ rad, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ rad, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.770  -5.199  -1.967   3.321  33.292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.38213    0.56176  46.964   <2e-16 ***
## rad         -0.40310    0.04349  -9.269   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.509 on 504 degrees of freedom
## Multiple R-squared:  0.1456, Adjusted R-squared:  0.1439
```

```
## F-statistic: 85.91 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = rad, y = medv)) +
  geom_segment(aes(xend = rad, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



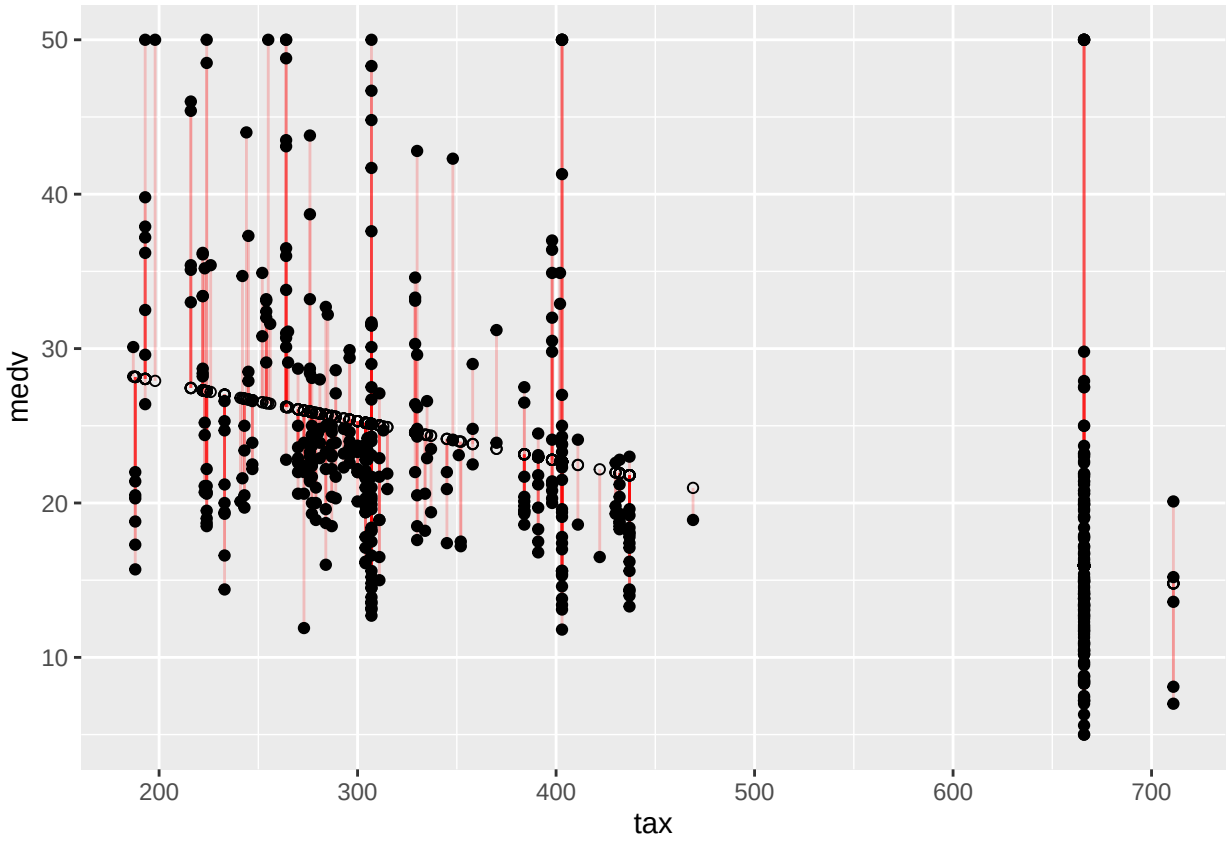

```
lm.fit = lm(medv ~ tax, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ tax, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.091  -5.173  -2.085   3.158  34.058
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 32.970654   0.948296   34.77   <2e-16 ***
## tax         -0.025568   0.002147  -11.91   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.133 on 504 degrees of freedom
## Multiple R-squared:  0.2195, Adjusted R-squared:  0.218
## F-statistic: 141.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = tax, y = medv)) +
  geom_segment(aes(xend = tax, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



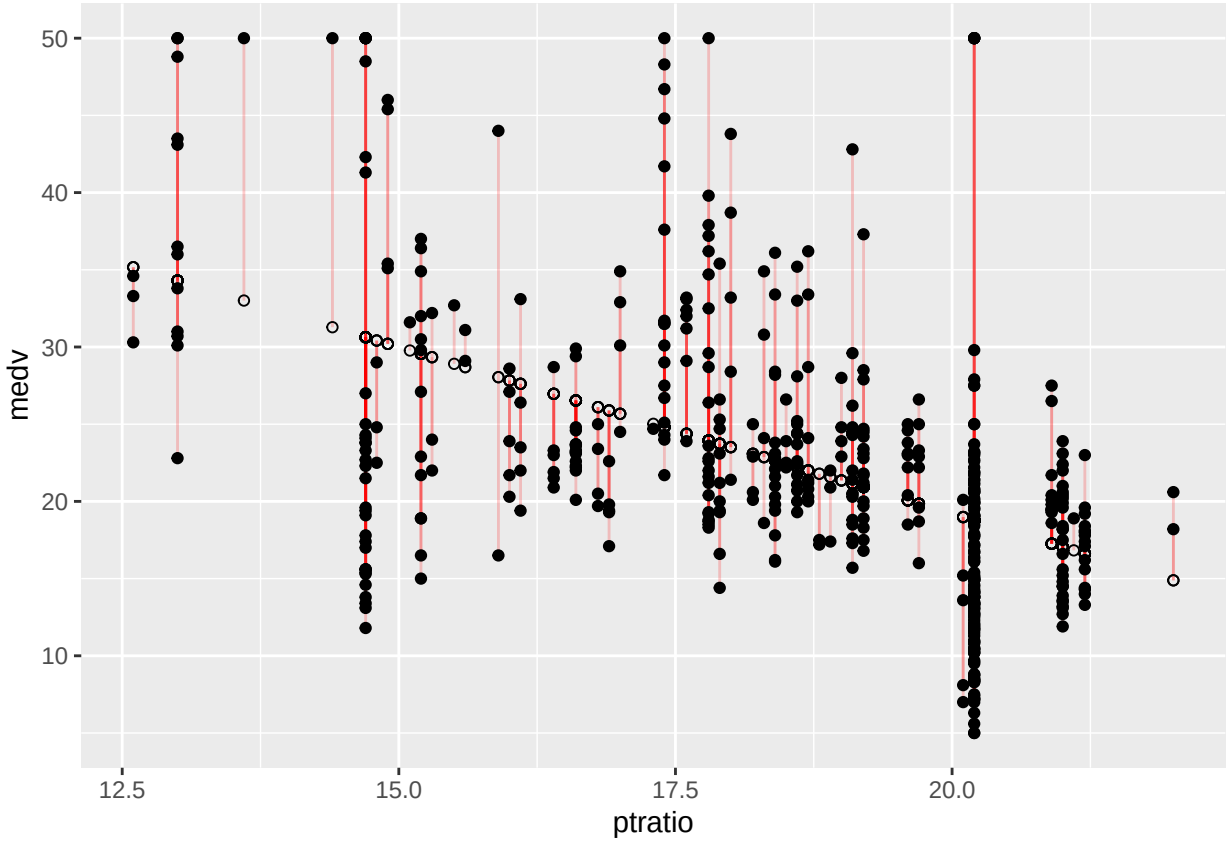

```
lm.fit = lm(medv ~ ptratio, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ptratio, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8342  -4.8262  -0.6426   3.1571  31.2303
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   62.345      3.029   20.58   <2e-16 ***
## ptratio       -2.157      0.163  -13.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.931 on 504 degrees of freedom
## Multiple R-squared:  0.2578, Adjusted R-squared:  0.2564
## F-statistic: 175.1 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = ptratio, y = medv)) +
  geom_segment(aes(xend = ptratio, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



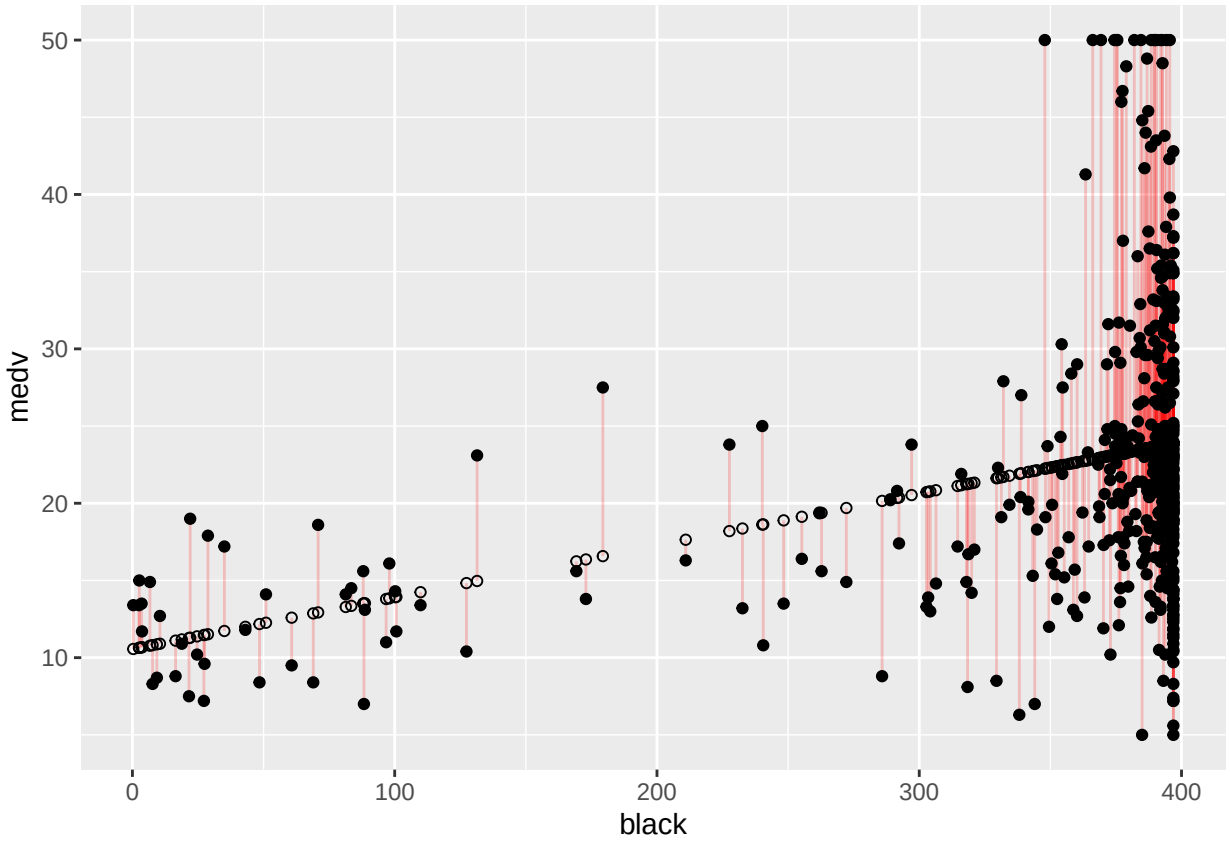

```
lm.fit = lm(medv ~ black, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
```

```
## lm(formula = medv ~ black, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.884  -4.862  -1.684   2.932  27.763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.551034   1.557463   6.775 3.49e-11 ***
## black        0.033593   0.004231   7.941 1.32e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.679 on 504 degrees of freedom
## Multiple R-squared:  0.1112, Adjusted R-squared:  0.1094
## F-statistic: 63.05 on 1 and 504 DF,  p-value: 1.318e-14
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = black, y = medv)) +
  geom_segment(aes(xend = black, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```

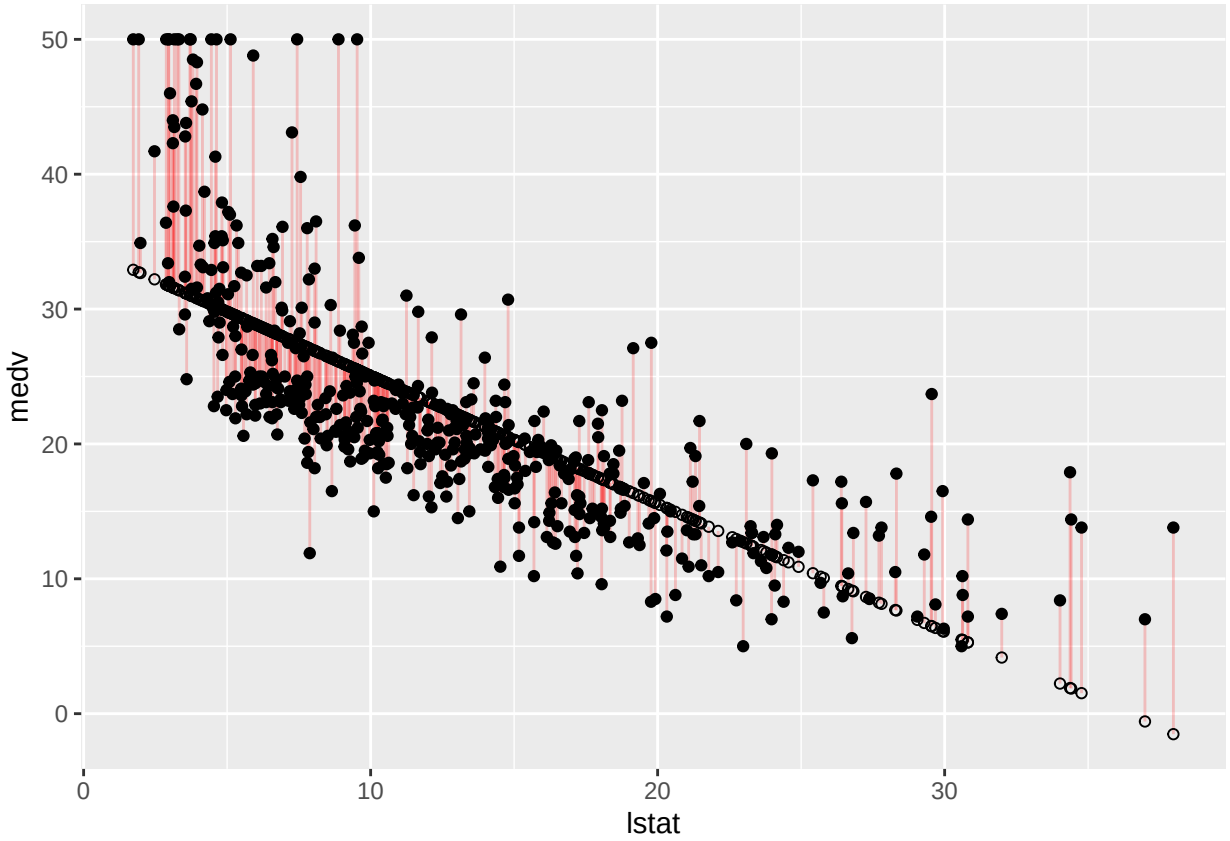


```r
lm.fit = lm(medv ~ lstat, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = lstat, y = medv)) +
  geom_segment(aes(xend = lstat, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```




```
lm.fit1 = lm(medv ~ lstat, data = my.boston)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

Selecting next feature

```r
lm.fit = lm(medv ~ lstat + rm, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.076  -3.516  -1.010   1.909  28.131
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.35827    3.17283  -0.428    0.669
## lstat       -0.64236    0.04373 -14.689   <2e-16 ***
## rm           5.09479    0.44447  11.463   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.54 on 503 degrees of freedom
## Multiple R-squared:  0.6386, Adjusted R-squared:  0.6371
## F-statistic: 444.3 on 2 and 503 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```r
lm.fit = lm(medv ~ lstat + age, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```r
lm.fit = lm(medv ~ lstat + nox, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + nox, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -3.936  -1.379   1.948  24.389
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.10207    1.40146  24.333   <2e-16 ***
## lstat       -0.96004    0.04805 -19.979   <2e-16 ***
## nox          1.04245    2.96130   0.352    0.725
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.221 on 503 degrees of freedom
## Multiple R-squared:  0.5443, Adjusted R-squared:  0.5424
## F-statistic: 300.3 on 2 and 503 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```r
lm.fit2 = lm(medv ~ lstat + rm, data = my.boston)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.076  -3.516  -1.010   1.909  28.131
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.35827    3.17283  -0.428    0.669
## lstat       -0.64236    0.04373 -14.689   <2e-16 ***
## rm           5.09479    0.44447  11.463   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.54 on 503 degrees of freedom
## Multiple R-squared:  0.6386, Adjusted R-squared:  0.6371
## F-statistic: 444.3 on 2 and 503 DF,  p-value: < 2.2e-16
```

```r
lm.fit = lm(medv ~ lstat + rm + tax, data = my.boston)
summary(lm.fit)
```

```
lm.fit = lm(medv ~ lstat + rm + ptratio, data = my.boston)
summary(lm.fit)

lm.fit = lm(medv ~ lstat + rm + indus, data = my.boston)
summary(lm.fit)

lm.fit = lm(medv ~ lstat + rm + dis, data = my.boston)
summary(lm.fit)

lm.fit = lm(medv ~ lstat + rm + black, data = my.boston)
summary(lm.fit)

lm.fit = lm(medv ~ lstat + rm + rad, data = my.boston)
summary(lm.fit)

lm.fit = lm(medv ~ lstat + rm + zn, data = my.boston)
summary(lm.fit)

lm.fit = lm(medv ~ lstat + rm + crim, data = my.boston)
summary(lm.fit)
```

```
lm.fit2 = lm(medv ~ lstat + rm + ptratio, data = my.boston)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm + ptratio, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4871  -3.1047  -0.7976   1.8129  29.6559
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.56711    3.91320   4.745 2.73e-06 ***
## lstat       -0.57181    0.04223 -13.540  < 2e-16 ***
## rm           4.51542    0.42587  10.603  < 2e-16 ***
## ptratio     -0.93072    0.11765  -7.911 1.64e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.229 on 502 degrees of freedom
## Multiple R-squared:  0.6786, Adjusted R-squared:  0.6767
## F-statistic: 353.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Backward selection

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

```
##
## Call:
## lm(formula = medv ~ ., data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ . - age, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
```

```
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ . - indus, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - indus, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.587  -2.737  -0.506   1.742  26.212
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.636e+01  5.091e+00   7.143 3.30e-12 ***
## crim        -1.084e-01  3.281e-02  -3.304 0.001022 **
## zn           4.593e-02  1.364e-02   3.368 0.000816 ***
## chas         2.716e+00  8.562e-01   3.173 0.001605 **
## nox         -1.743e+01  3.681e+00  -4.735 2.87e-06 ***
## rm           3.797e+00  4.158e-01   9.132  < 2e-16 ***
## age          6.971e-04  1.320e-02   0.053 0.957898
## dis         -1.490e+00  1.948e-01  -7.648 1.08e-13 ***
## rad          2.999e-01  6.367e-02   4.710 3.22e-06 ***
## tax         -1.178e-02  3.378e-03  -3.489 0.000529 ***
## ptratio     -9.471e-01  1.296e-01  -7.308 1.10e-12 ***
## black        9.282e-03  2.682e-03   3.461 0.000586 ***
## lstat       -5.235e-01  5.052e-02 -10.361  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.741 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ . - nox, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - nox, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.040  -2.831  -0.823   1.573  27.220
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.499999   4.364331   5.385 1.13e-07 ***
## crim        -0.098898   0.033486  -2.953 0.003293 **
## zn           0.048706   0.014003   3.478 0.000549 ***
## indus       -0.054945   0.060543  -0.908 0.364563
## chas         2.552290   0.878930   2.904 0.003851 **
## rm           3.991551   0.424714   9.398  < 2e-16 ***
## age         -0.015599   0.013001  -1.200 0.230790
## dis         -1.214013   0.195327  -6.215 1.09e-09 ***
## rad          0.262181   0.067033   3.911 0.000105 ***
## tax         -0.013639   0.003828  -3.563 0.000402 ***
## ptratio     -0.752176   0.126074  -5.966 4.64e-09 ***
## black        0.010247   0.002734   3.748 0.000199 ***
## lstat       -0.540980   0.051643 -10.475  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.844 on 493 degrees of freedom
## Multiple R-squared:  0.7292, Adjusted R-squared:  0.7226
## F-statistic: 110.6 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ . - zn, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - dis, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - tax, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - ptratio, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - black, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - chas, data = my.boston)
summary(lm.fit)
```

```
lm.fit.back1 = lm(medv ~ . - age, data = my.boston)
summary(lm.fit.back1)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
```

```
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

Selecting next feature to remove.

```
lm.fit = lm(medv ~ . - age - indus, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5984  -2.7386  -0.5046   1.7273  26.2373
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## zn            0.045845   0.013523   3.390 0.000754 ***
## chas          2.718716   0.854240   3.183 0.001551 **
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## black         0.009291   0.002674   3.475 0.000557 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.736 on 494 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7348
## F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ . - age - nox, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age - nox, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.8808  -2.8171  -0.7587   1.7176  26.6875
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.108557   4.354055   5.307 1.68e-07 ***
## crim        -0.098378   0.033498  -2.937 0.003471 **
## zn           0.050975   0.013881   3.672 0.000267 ***
## indus       -0.060498   0.060393  -1.002 0.316957
## chas         2.486994   0.877634   2.834 0.004789 **
## rm           3.894948   0.417198   9.336  < 2e-16 ***
## dis         -1.121279   0.179465  -6.248 8.99e-10 ***
## rad          0.265392   0.067009   3.961 8.58e-05 ***
## tax         -0.013877   0.003824  -3.629 0.000315 ***
## ptratio     -0.750648   0.126124  -5.952 5.04e-09 ***
## black        0.010097   0.002732   3.696 0.000244 ***
## lstat       -0.564127   0.047926 -11.771  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.846 on 494 degrees of freedom
## Multiple R-squared:  0.7284, Adjusted R-squared:  0.7224
## F-statistic: 120.5 on 11 and 494 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ . - age - zn, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - dis, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - tax, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - ptratio, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - black, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - chas, data = my.boston)
summary(lm.fit)
```

```
lm.fit.back2 = lm(medv ~ . - age - indus, data = my.boston)
summary(lm.fit.back2)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5984  -2.7386  -0.5046   1.7273  26.2373
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## zn            0.045845   0.013523   3.390 0.000754 ***
## chas          2.718716   0.854240   3.183 0.001551 **
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## black         0.009291   0.002674   3.475 0.000557 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.736 on 494 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7348
## F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

Selecting third feature to remove.

```
lm.fit = lm(medv ~ . - age - indus - chas, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus - chas, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3716  -2.7943  -0.5508   1.8942  26.3982
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.620311   5.113241   7.162 2.90e-12 ***
## crim         -0.114056   0.033032  -3.453 0.000602 ***
## zn            0.045742   0.013647   3.352 0.000864 ***
## nox         -16.469153   3.556086  -4.631 4.65e-06 ***
## rm            3.844639   0.409818   9.381  < 2e-16 ***
## dis          -1.526099   0.187136  -8.155 2.89e-15 ***
## rad           0.315531   0.063785   4.947 1.04e-06 ***
## tax          -0.012674   0.003391  -3.737 0.000208 ***
## ptratio      -0.978442   0.129857  -7.535 2.34e-13 ***
## black         0.009730   0.002695   3.611 0.000337 ***
## lstat        -0.528103   0.047827 -11.042  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.78 on 495 degrees of freedom
## Multiple R-squared:  0.7353, Adjusted R-squared:  0.7299
## F-statistic: 137.5 on 10 and 495 DF,  p-value: < 2.2e-16
```

```r
lm.fit = lm(medv ~ . - age - indus - crim, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus - crim, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2687  -2.6207  -0.5015   1.8076  26.7261
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.225508   5.106760   6.898 1.62e-11 ***
## zn            0.041732   0.013600   3.069 0.002269 **
## chas          2.871517   0.861510   3.333 0.000923 ***
## nox         -16.511255   3.560778  -4.637 4.53e-06 ***
## rm            3.832274   0.410268   9.341  < 2e-16 ***
## dis          -1.420274   0.186277  -7.625 1.26e-13 ***
## rad           0.238893   0.061292   3.898 0.000111 ***
## tax          -0.011430   0.003404  -3.357 0.000847 ***
## ptratio      -0.935501   0.130311  -7.179 2.59e-12 ***
## black         0.010320   0.002682   3.847 0.000135 ***
## lstat        -0.547851   0.047271 -11.590  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.784 on 495 degrees of freedom
## Multiple R-squared:  0.7348, Adjusted R-squared:  0.7295
## F-statistic: 137.2 on 10 and 495 DF,  p-value: < 2.2e-16
```

```r
lm.fit = lm(medv ~ . - age - indus - zn, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - indus - rm, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - indus - nox, data = my.boston)
summary(lm.fit)
lm.fit = lm(medv ~ . - age - indus - tax, data = my.boston)
summary(lm.fit)
```

```r
lm.fit.back3 = lm(medv ~ . - age - indus - chas, data = my.boston)
summary(lm.fit.back3)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus - chas, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3716  -2.7943  -0.5508   1.8942  26.3982
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.620311   5.113241   7.162 2.90e-12 ***
## crim         -0.114056   0.033032  -3.453 0.000602 ***
## zn            0.045742   0.013647   3.352 0.000864 ***
## nox         -16.469153   3.556086  -4.631 4.65e-06 ***
## rm            3.844639   0.409818   9.381  < 2e-16 ***
## dis          -1.526099   0.187136  -8.155 2.89e-15 ***
## rad           0.315531   0.063785   4.947 1.04e-06 ***
## tax          -0.012674   0.003391  -3.737 0.000208 ***
## ptratio      -0.978442   0.129857  -7.535 2.34e-13 ***
## black         0.009730   0.002695   3.611 0.000337 ***
## lstat        -0.528103   0.047827 -11.042  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.78 on 495 degrees of freedom
## Multiple R-squared:  0.7353, Adjusted R-squared:  0.7299
## F-statistic: 137.5 on 10 and 495 DF,  p-value: < 2.2e-16
```

And so on.

```
lm.fit = lm(medv ~ . - age - indus - chas - crim, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age - indus - chas - crim, data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8917  -2.7329  -0.4988   1.8547  26.6433
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.459724   5.158054   6.875 1.87e-11 ***
## zn            0.041396   0.013737   3.013 0.002715 **
## nox         -15.502932   3.583879  -4.326 1.84e-05 ***
## rm            3.879580   0.414180   9.367  < 2e-16 ***
## dis          -1.451648   0.187926  -7.725 6.26e-14 ***
## rad           0.252412   0.061778   4.086 5.12e-05 ***
## tax          -0.012360   0.003427  -3.606 0.000342 ***
## ptratio      -0.968703   0.131248  -7.381 6.69e-13 ***
## black         0.010842   0.002705   4.008 7.06e-05 ***
## lstat        -0.555124   0.047699 -11.638  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.832 on 496 degrees of freedom
## Multiple R-squared:  0.7289, Adjusted R-squared:  0.724
## F-statistic: 148.2 on 9 and 496 DF,  p-value: < 2.2e-16
```

Interaction terms

```
lm.fit = lm(medv ~ lstat * age, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat * age, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Non-linear transformations

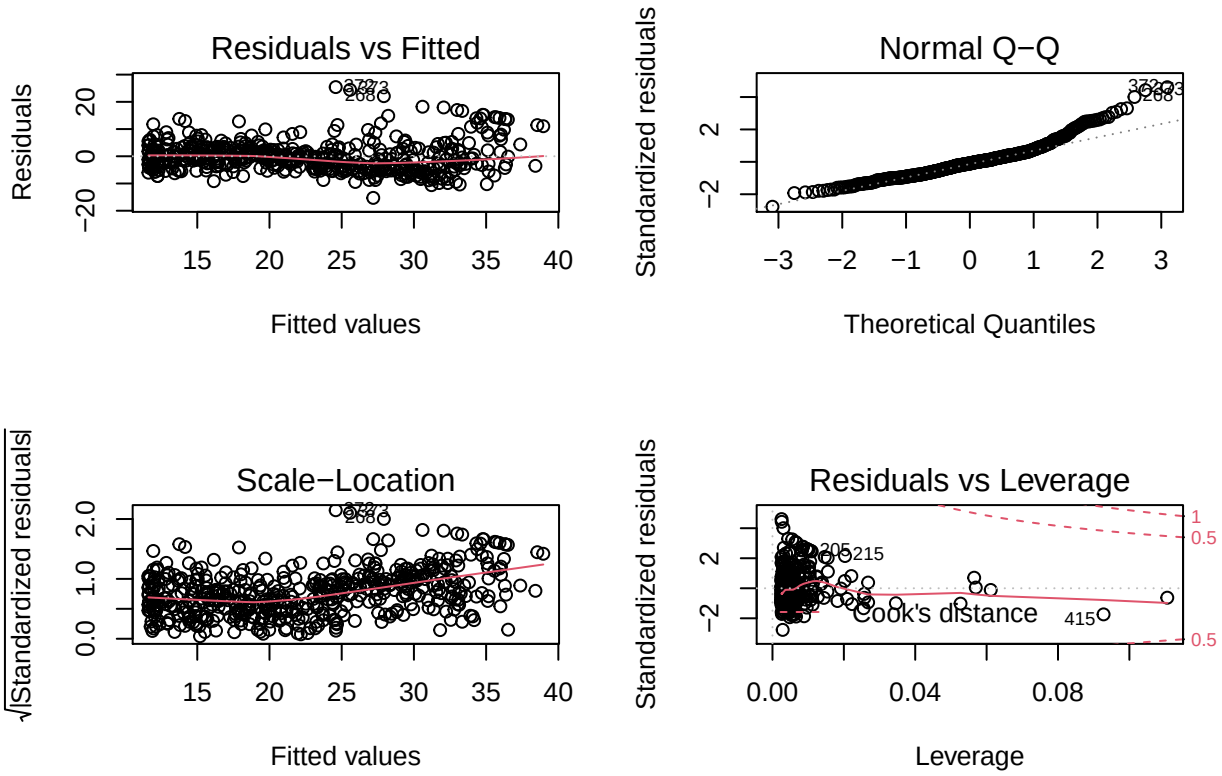
```
lm.fit2 = lm(medv ~ lstat + I(lstat^2), data = my.boston)
summary (lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = my.boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ lstat, data = my.boston)
lm.fit2 = lm(medv ~ lstat + I(lstat^2), data = my.boston)
anova(lm.fit ,lm.fit2)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(lm.fit2)
```



Qualitative Predictors

Qualitative Predictors on two levels

```
summary(my.boston)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm              age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

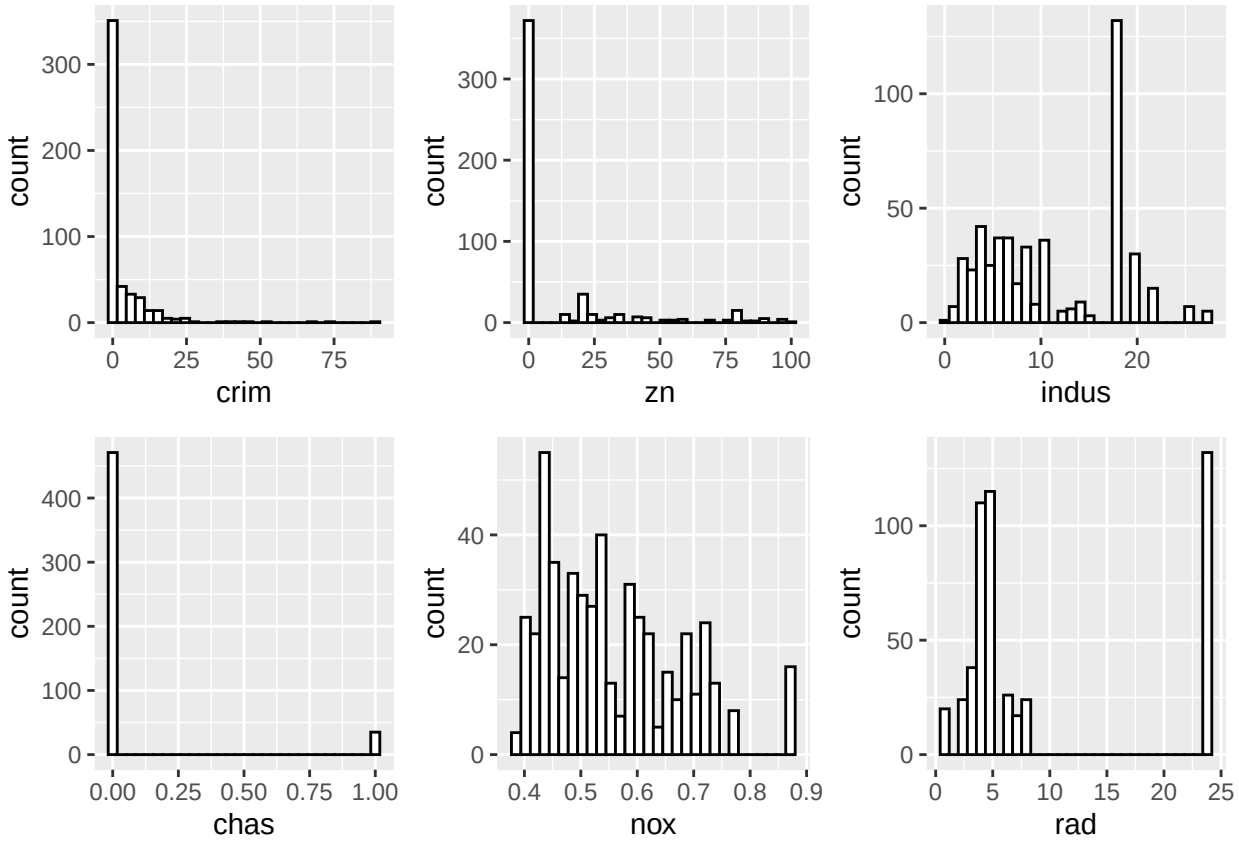
```
str(my.boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
p1 <- ggplot(my.boston, aes(x=crim)) + geom_histogram(color="black", fill="white")
p2 <- ggplot(my.boston, aes(x=zn)) + geom_histogram(color="black", fill="white")
```

```
p3 <- ggplot(my.boston, aes(x=indus)) + geom_histogram(color="black", fill="white")
p4 <- ggplot(my.boston, aes(x=chas)) + geom_histogram(color="black", fill="white")
p5 <- ggplot(my.boston, aes(x=nox)) + geom_histogram(color="black", fill="white")
p6 <- ggplot(my.boston, aes(x=rm)) + geom_histogram(color="black", fill="white")
p7 <- ggplot(my.boston, aes(x=age)) + geom_histogram(color="black", fill="white")
p8 <- ggplot(my.boston, aes(x=dis)) + geom_histogram(color="black", fill="white")
p9 <- ggplot(my.boston, aes(x=rad)) + geom_histogram(color="black", fill="white")
p10 <- ggplot(my.boston, aes(x=tax)) + geom_histogram(color="black", fill="white")
p11 <- ggplot(my.boston, aes(x=ptratio)) + geom_histogram(color="black", fill="white")
p12 <- ggplot(my.boston, aes(x=black)) + geom_histogram(color="black", fill="white")
p13 <- ggplot(my.boston, aes(x=lstat)) + geom_histogram(color="black", fill="white")
p14 <- ggplot(my.boston, aes(x=medv)) + geom_histogram(color="black", fill="white")

grid.arrange(p1, p2, p3, p4, p5, p9, nrow = 2)
```




```
my.boston$chas = as.factor(my.boston$chas)
str(my.boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
```

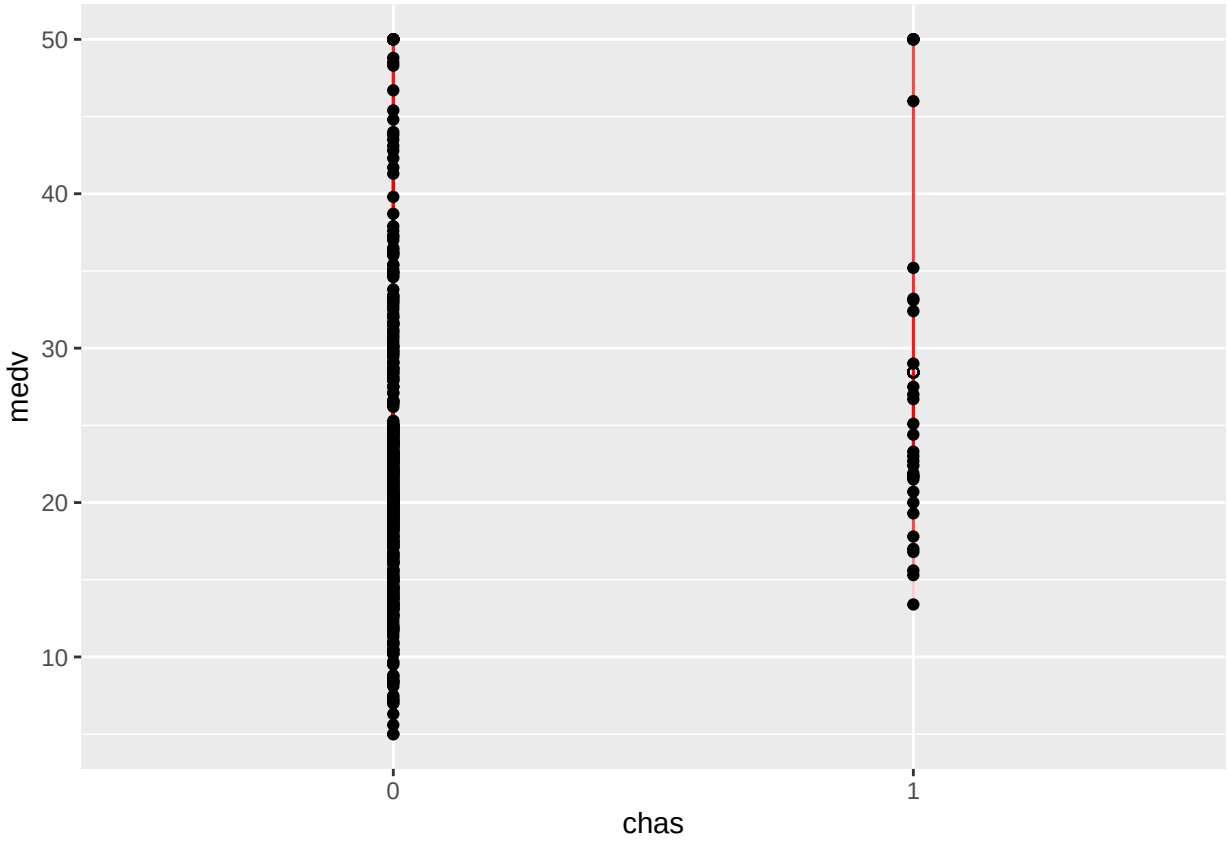
```
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
lm.fit = lm(medv ~ chas, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ chas, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.094  -5.894  -1.417   2.856  27.906
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.0938     0.4176  52.902  < 2e-16 ***
## chas1         6.3462     1.5880   3.996 7.39e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.064 on 504 degrees of freedom
## Multiple R-squared:  0.03072,    Adjusted R-squared:  0.02879
## F-statistic: 15.97 on 1 and 504 DF,  p-value: 7.391e-05
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = chas, y = medv)) +
  geom_segment(aes(xend = chas, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



Qualitative Predictors on multiple levels

```
my.boston$rad = as.factor(my.boston$rad)
str(my.boston)
```

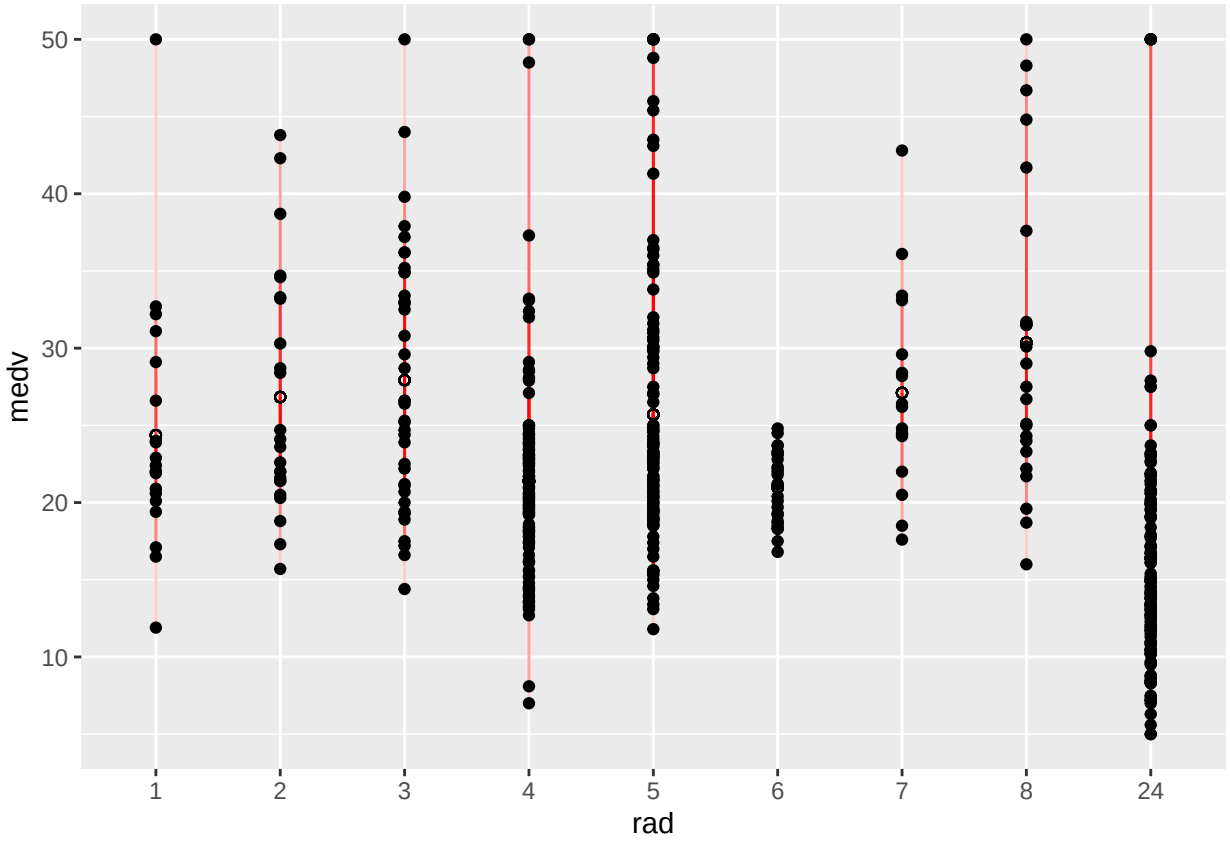
```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : Factor w/ 9 levels "1","2","3","4",..: 1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
lm.fit = lm(medv ~ rad, data = my.boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ rad, data = my.boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.387  -5.280  -1.732   3.175  33.596
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.365      1.821  13.383  < 2e-16 ***
## rad2           2.468      2.465   1.001   0.3172
## rad3           3.564      2.249   1.584   0.1137
## rad4          -2.978      1.979  -1.504   0.1331
## rad5           1.342      1.973   0.680   0.4966
## rad6          -3.388      2.422  -1.399   0.1624
## rad7           2.741      2.686   1.020   0.3080
## rad8           5.993      2.465   2.431   0.0154 *
## rad24         -7.961      1.954  -4.075 5.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.142 on 497 degrees of freedom
## Multiple R-squared:  0.2287, Adjusted R-squared:  0.2162
## F-statistic: 18.42 on 8 and 497 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.boston, aes(x = rad, y = medv)) +
  geom_segment(aes(xend = rad, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```


50
40
30
20
10
medv
1
2
3
4
5
6
7
8
24
rad

Additional exercise

- Perorm simple linear regression on the Auto dataset, with mpg as the response and horsepower as the predictor. Discuss the model in detail.
- Perform multiple linear regression on the Auto dataset.
  - use the summary, str and ggpairs function to understand the data
  - evaluate the correlations between variables using a correlogram
  - perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. discuss the results. Attempt a forward/ backward selection.
  - Assess potential interactions between predictors
  - Assess non-linear relationships.

```
my.auto = Auto
```

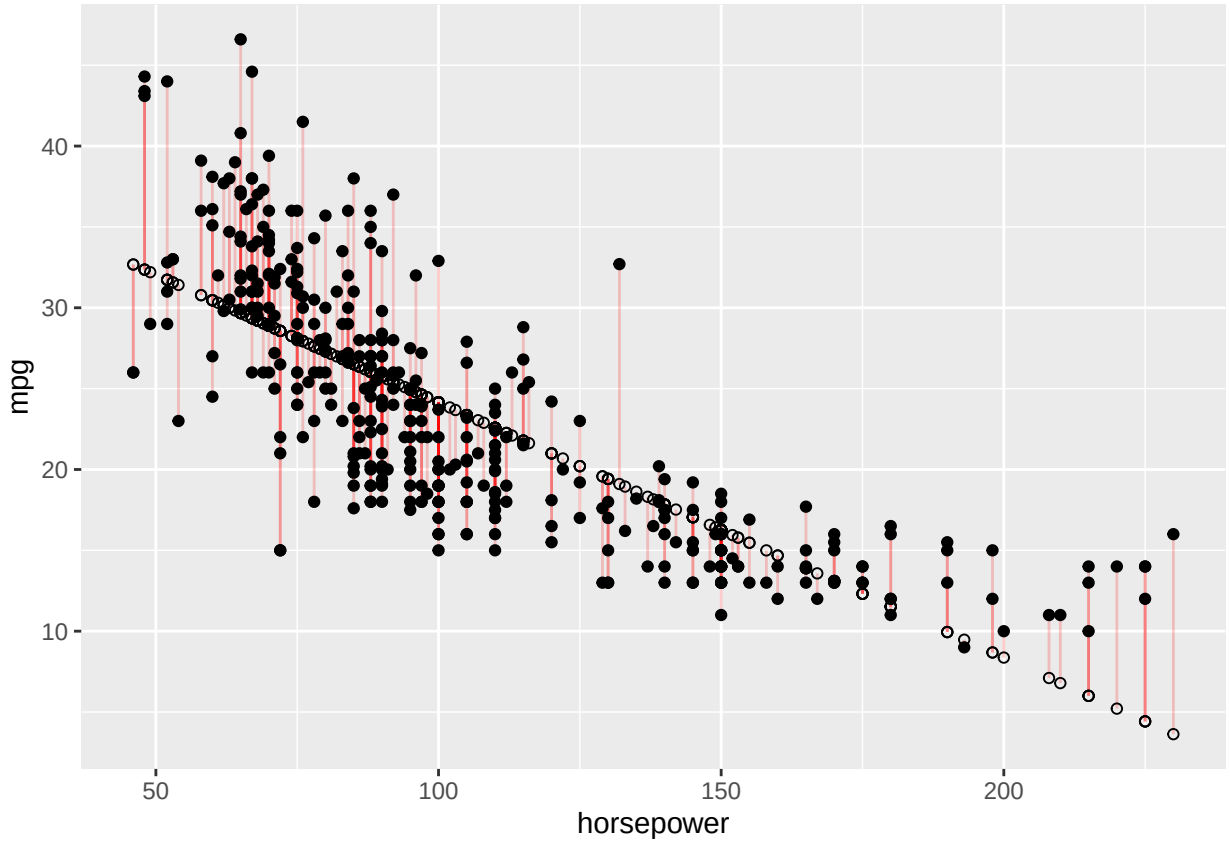
Simple Linear Regression

```
lm.mpg.hp = lm(mpg ~ horsepower, data = my.auto)
summary(lm.mpg.hp)
```

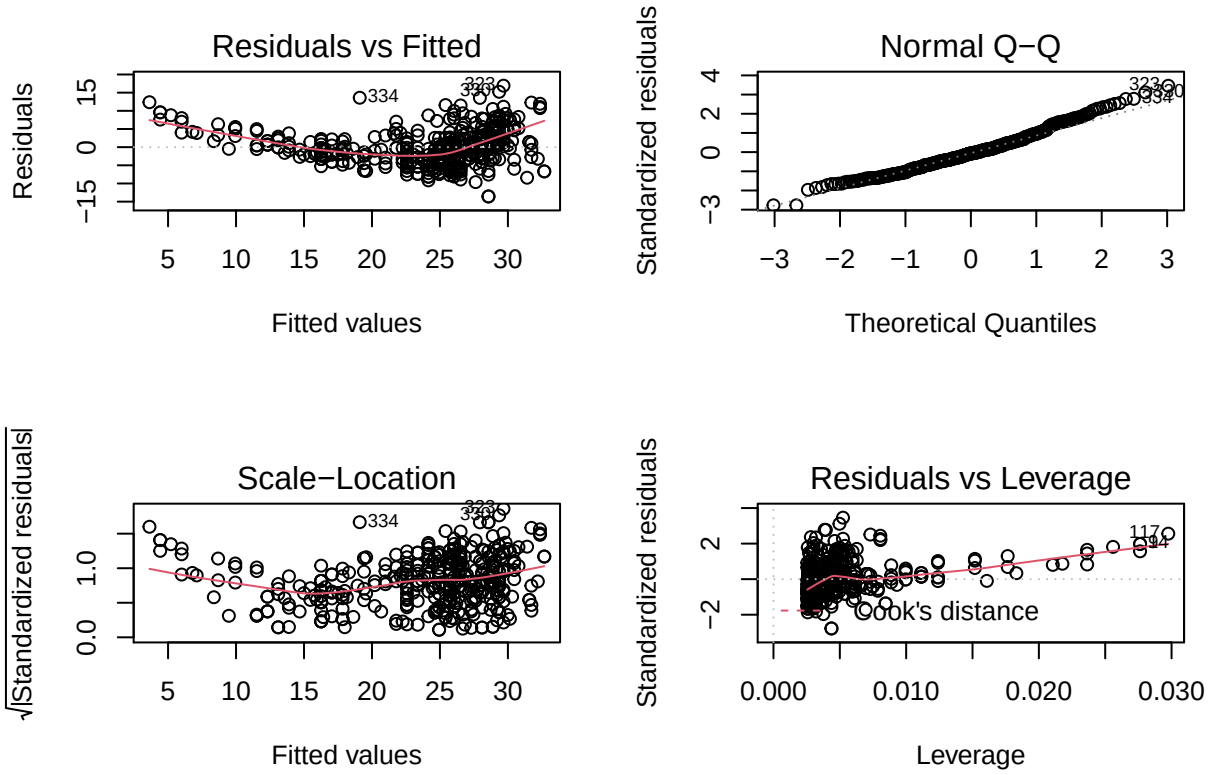
```
##
## Call:
## lm(formula = mpg ~ horsepower, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.mpg.hp)   # Save the predicted values
my.residuals <- residuals(lm.mpg.hp) # Save the residual values

ggplot(my.auto, aes(x = horsepower, y = mpg)) +
  geom_segment(aes(xend = horsepower, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```




```
par(mfrow = c(2,2))
plot(lm.mpg.hp)
```



Multiple linear regression

Assess the individual predictors. Attempt a forward/backward selection. Discuss the model.

```
summary(my.auto)
```

```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##
##   acceleration        year           origin                      name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5
##  Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5
##  Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
##  Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
##                                                  (Other)           :365
```

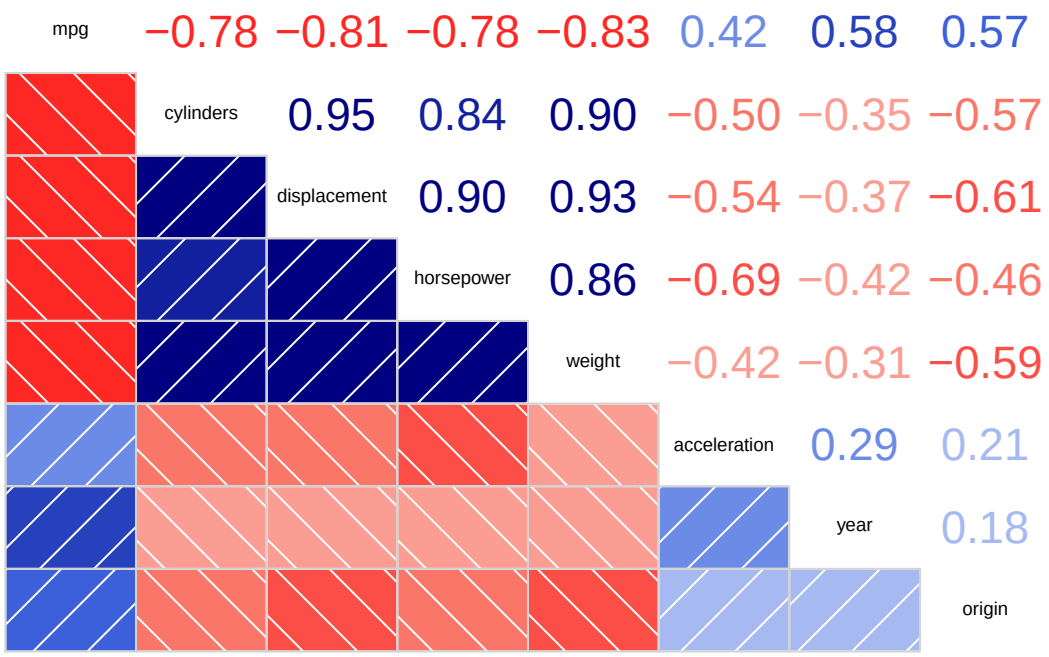
```
str(my.auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 1
```

```
pdf("auto_data.pdf", width = 20, height = 20)
ggpairs(my.auto[,1:8])
dev.off()
```

```
## pdf
##   2
```

```
cor.matrix <- cor(my.auto[,1:8])
corrgram::corrgram(cor.matrix, order=FALSE,upper.panel=panel.cor)
```



```r
lm.all = lm(mpg ~ . - name, data = my.auto)
summary(lm.all)
```

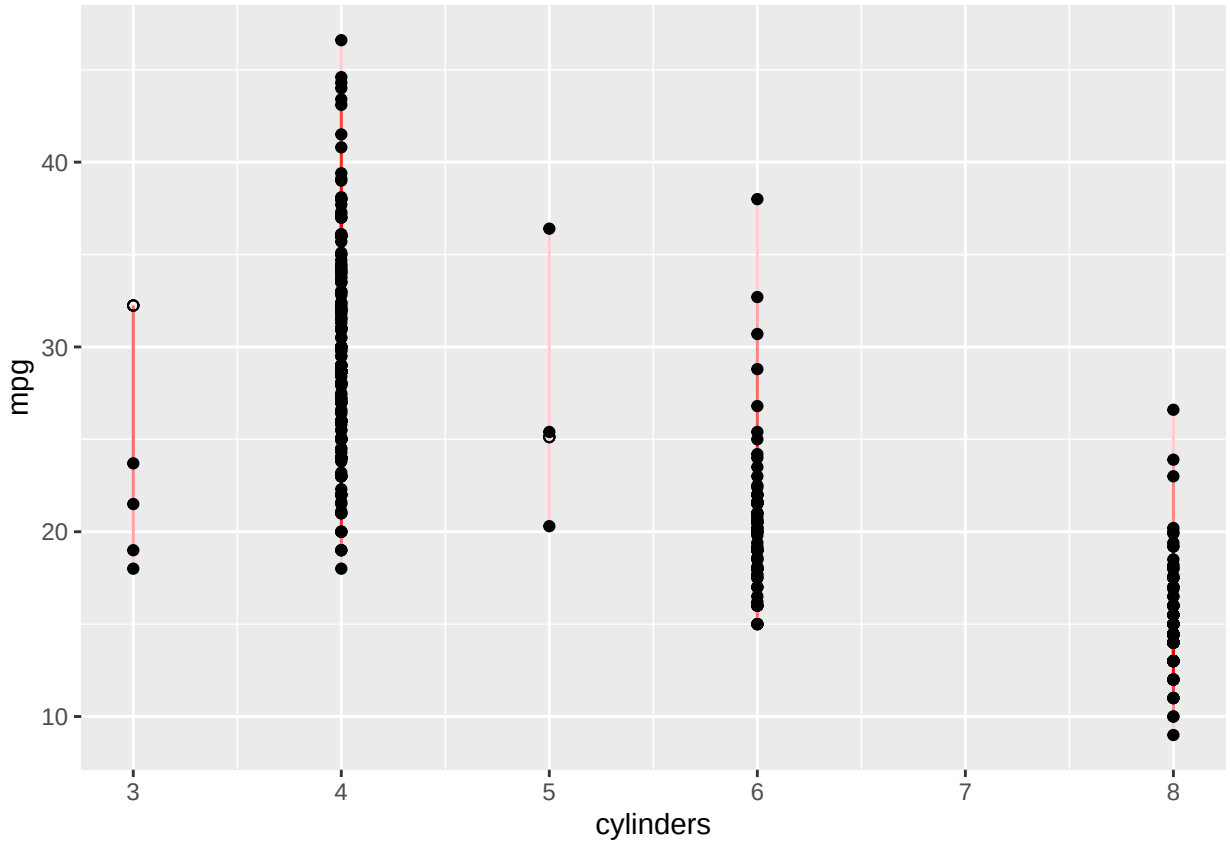
```
##
## Call:
## lm(formula = mpg ~ . - name, data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

```r
lm.fit = lm(mpg ~ cylinders, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2413  -3.1832  -0.6332   2.5491  17.9168
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.9155     0.8349   51.40   <2e-16 ***
## cylinders    -3.5581     0.1457  -24.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.914 on 390 degrees of freedom
## Multiple R-squared:  0.6047, Adjusted R-squared:  0.6037
## F-statistic: 596.6 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```
ggplot(my.auto, aes(x = cylinders, y = mpg)) +
  geom_segment(aes(xend = cylinders, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



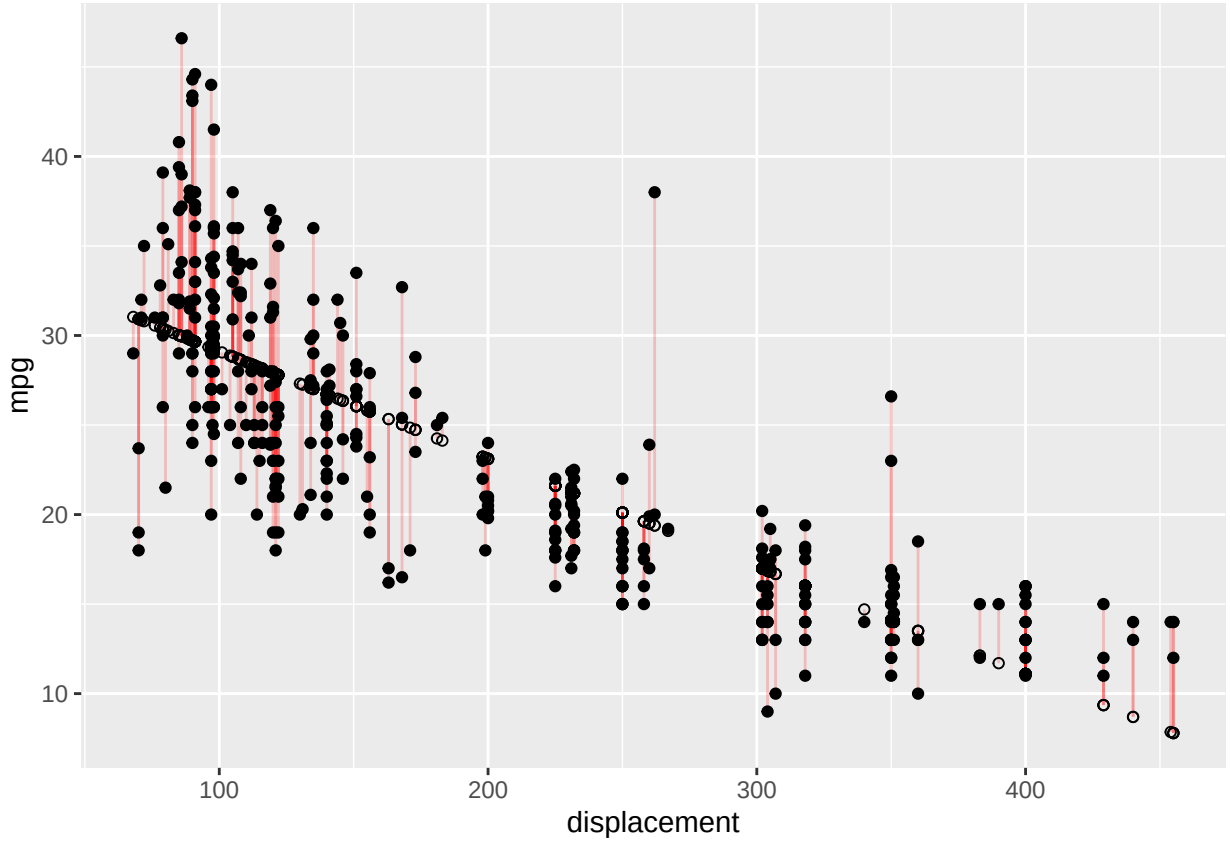

```
lm.fit = lm(mpg ~ displacement, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ displacement, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9170  -3.0243  -0.5021   2.3512  18.6128
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.12064    0.49443   71.03   <2e-16 ***
## displacement -0.06005    0.00224  -26.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.635 on 390 degrees of freedom
```

```
## Multiple R-squared:  0.6482, Adjusted R-squared:  0.6473
## F-statistic: 718.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.auto, aes(x = displacement, y = mpg)) +
  geom_segment(aes(xend = displacement, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



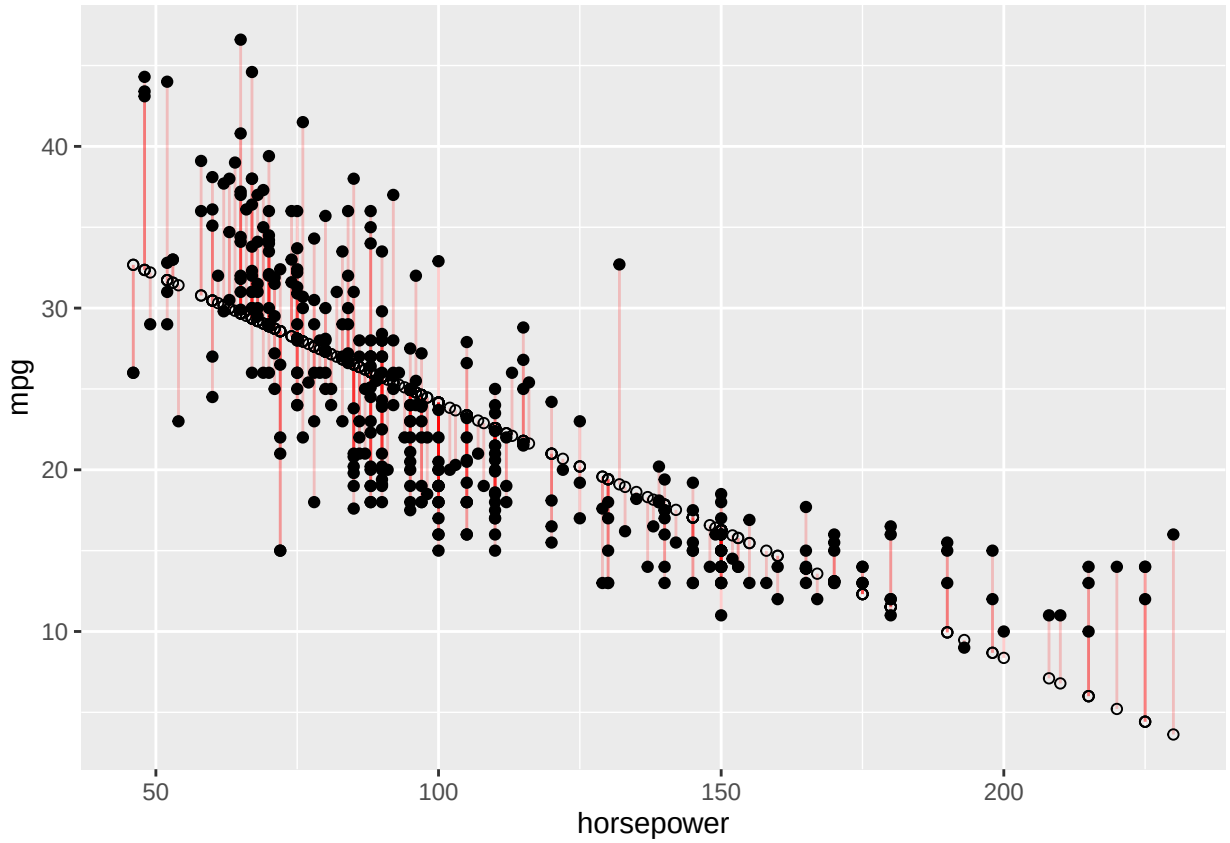

```
lm.fit = lm(mpg ~ horsepower, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
```

```
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.auto, aes(x = horsepower, y = mpg)) +
  geom_segment(aes(xend = horsepower, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



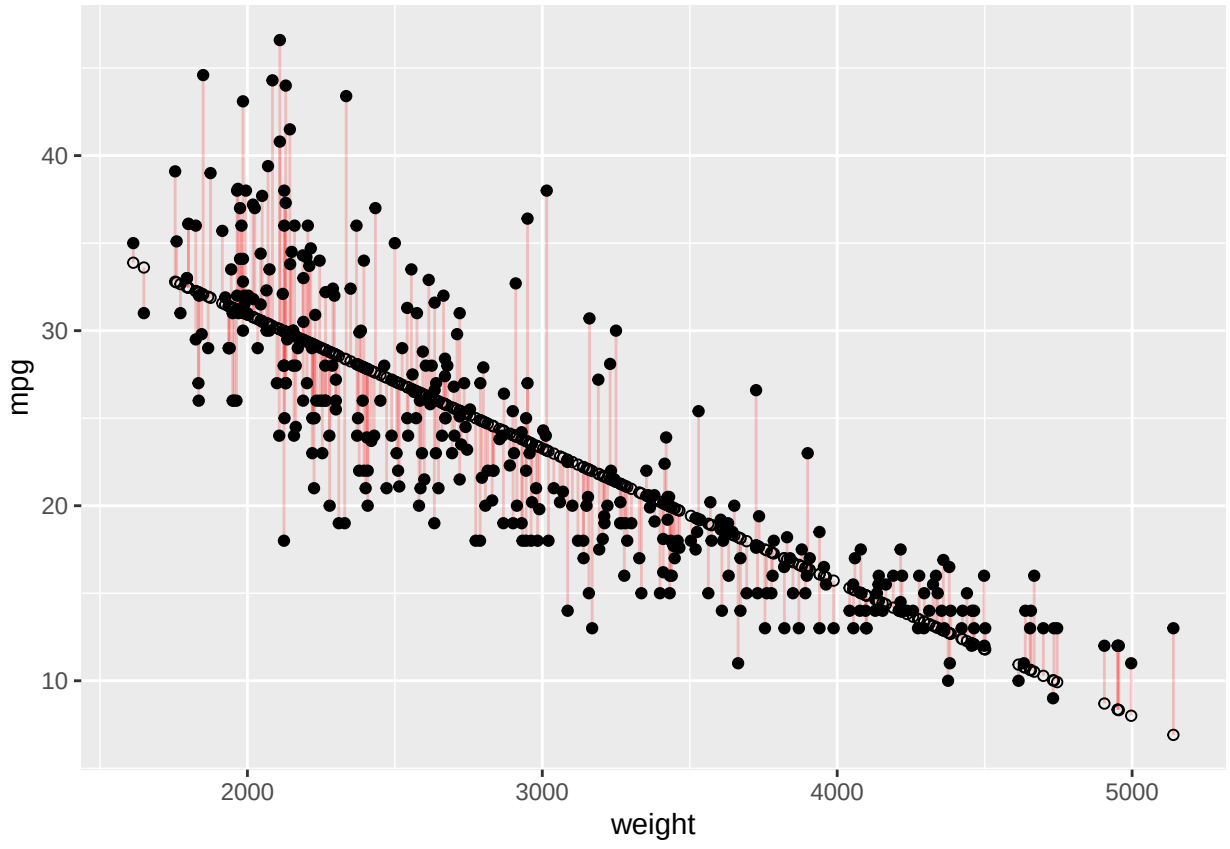

```
lm.fit = lm(mpg ~ weight, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -11.9736  -2.7556  -0.3358   2.1379  16.5194
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.216524   0.798673   57.87   <2e-16 ***
## weight      -0.007647   0.000258  -29.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.333 on 390 degrees of freedom
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6918
## F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.auto, aes(x = weight, y = mpg)) +
  geom_segment(aes(xend = weight, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



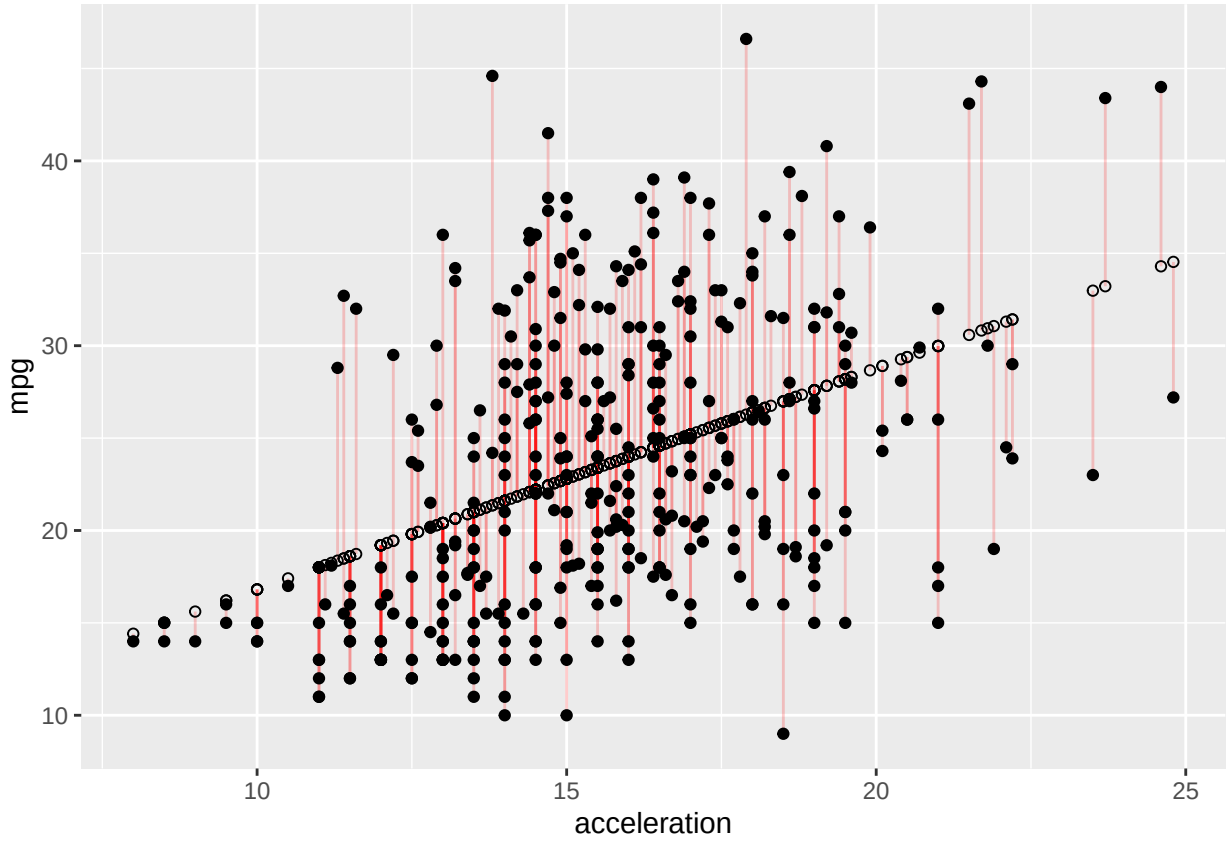

```
lm.fit = lm(mpg ~ acceleration, data = my.auto)
summary(lm.fit)
```

```
##
```

```
## Call:
## lm(formula = mpg ~ acceleration, data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.989  -5.616  -1.199   4.801  23.239
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.8332     2.0485   2.359   0.0188 *
## acceleration   1.1976     0.1298   9.228   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.08 on 390 degrees of freedom
## Multiple R-squared:  0.1792, Adjusted R-squared:  0.1771
## F-statistic: 85.15 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.auto, aes(x = acceleration, y = mpg)) +
  geom_segment(aes(xend = acceleration, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```

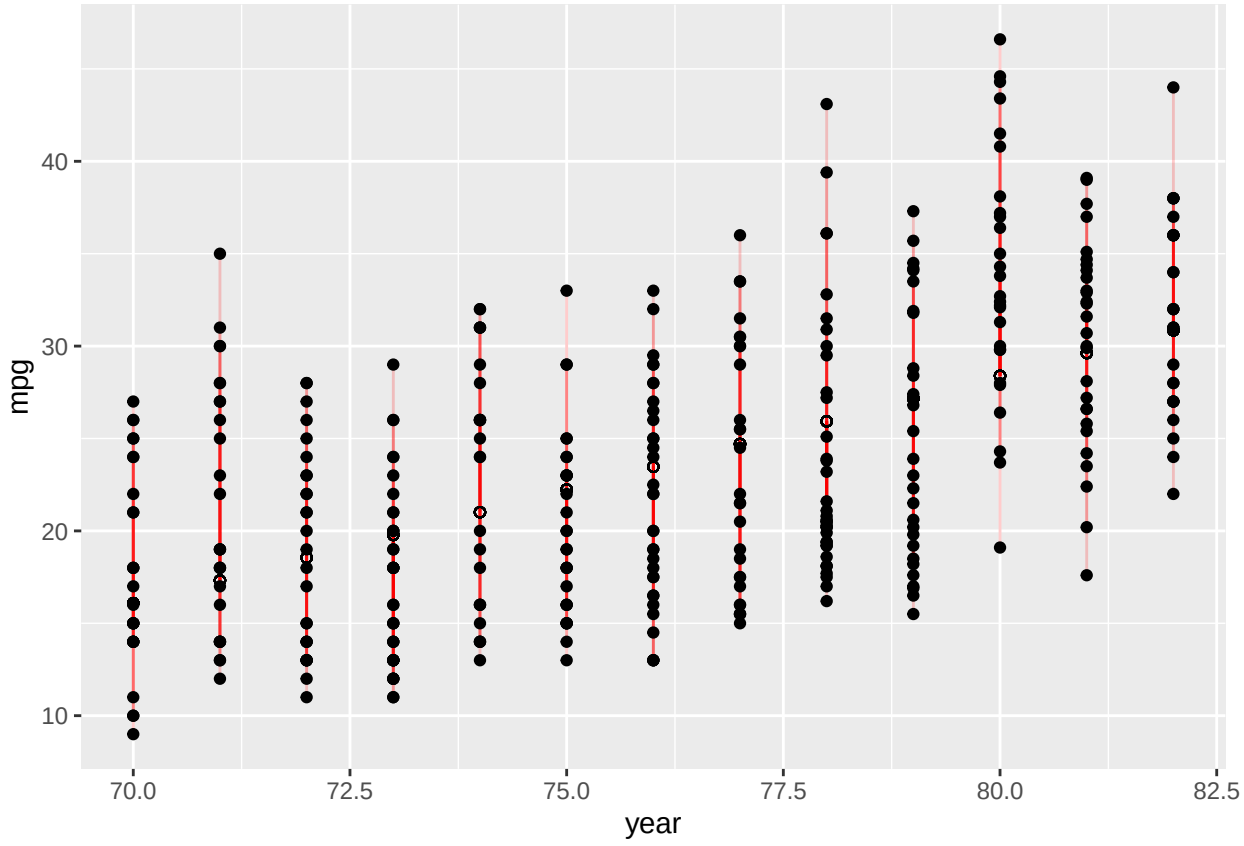


```r
lm.fit = lm(mpg ~ year, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ year, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0212  -5.4411  -0.4412   4.9739  18.2088
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70.01167    6.64516  -10.54   <2e-16 ***
## year          1.23004    0.08736   14.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.363 on 390 degrees of freedom
## Multiple R-squared:  0.337,  Adjusted R-squared:  0.3353
## F-statistic: 198.3 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values

ggplot(my.auto, aes(x = year, y = mpg)) +
  geom_segment(aes(xend = year, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```

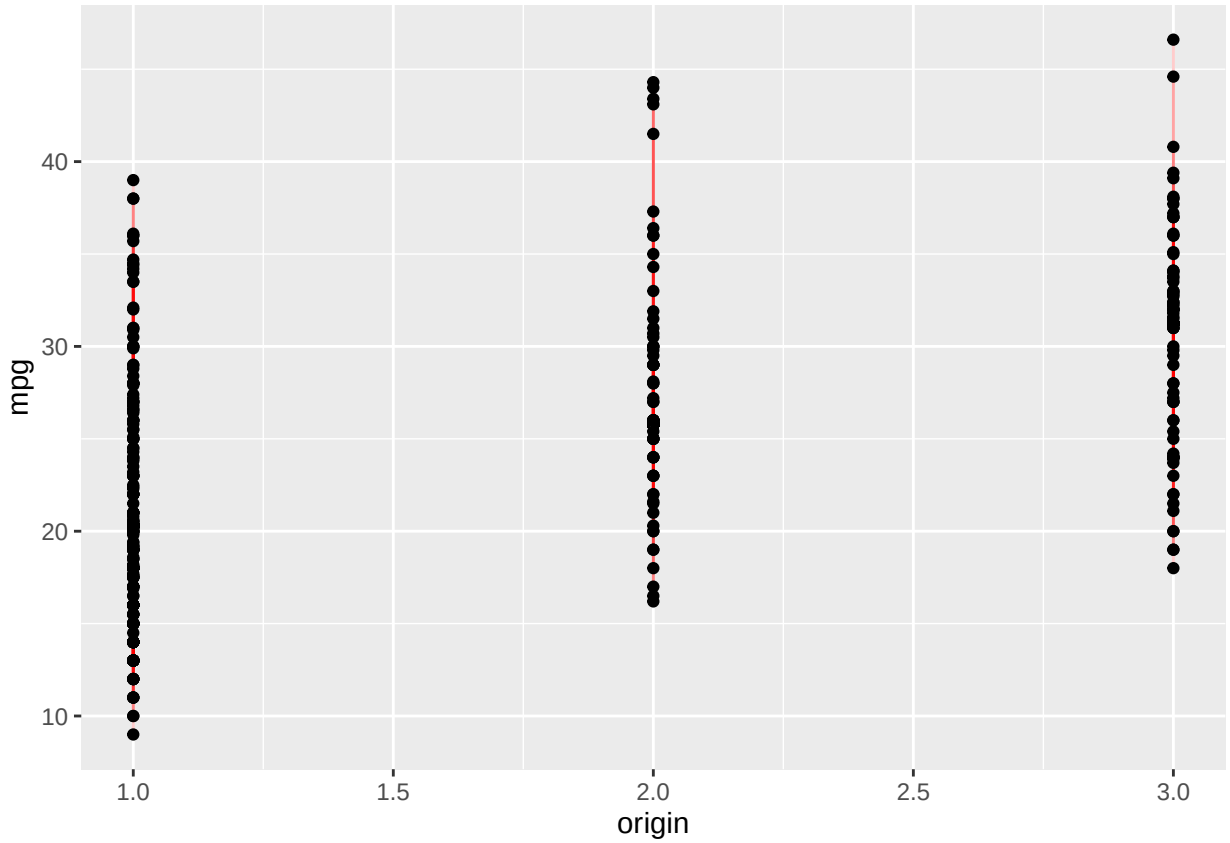



```
lm.fit = lm(mpg ~ origin, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ origin, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2416  -5.2533  -0.7651   3.8967  18.7115
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.8120     0.7164   20.68   <2e-16 ***
## origin        5.4765     0.4048   13.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.447 on 390 degrees of freedom
## Multiple R-squared:  0.3195, Adjusted R-squared:  0.3177
## F-statistic: 183.1 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
my.predicted <- predict(lm.fit)   # Save the predicted values
my.residuals <- residuals(lm.fit) # Save the residual values
```

```
ggplot(my.auto, aes(x = origin, y = mpg)) +
  geom_segment(aes(xend = origin, yend = my.predicted), color='red', alpha=0.2) +
  geom_point() +
  geom_point(aes(y = my.predicted), shape = 1)
```



Forward selection.

```
lm.fit1 = lm(mpg ~ weight, data = my.auto)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9736  -2.7556  -0.3358   2.1379  16.5194
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.216524   0.798673   57.87   <2e-16 ***
## weight      -0.007647   0.000258  -29.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.333 on 390 degrees of freedom
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6918
## F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16
```

Choose next feature

```
lm.fit = lm(mpg ~ weight + year, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ weight + year, data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8505 -2.3014 -0.1167  2.0367 14.3555
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.435e+01  4.007e+00  -3.581 0.000386 ***
## weight      -6.632e-03  2.146e-04 -30.911  < 2e-16 ***
## year         7.573e-01  4.947e-02  15.308  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.427 on 389 degrees of freedom
## Multiple R-squared:  0.8082, Adjusted R-squared:  0.8072
## F-statistic: 819.5 on 2 and 389 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(mpg ~ weight + cylinders, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ weight + cylinders, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6469  -2.8282  -0.2905   2.1606  16.5856
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.2923105  0.7939685  58.305   <2e-16 ***
## weight      -0.0063471  0.0005811 -10.922   <2e-16 ***
## cylinders   -0.7213779  0.2893780  -2.493   0.0131 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.304 on 389 degrees of freedom
## Multiple R-squared:  0.6975, Adjusted R-squared:  0.6959
## F-statistic: 448.4 on 2 and 389 DF,  p-value: < 2.2e-16
```

```r
lm.fit = lm(mpg ~ weight + origin, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ weight + origin, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0698  -2.7888  -0.3122   2.4489  15.4816
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.4908175  1.3266161   32.03  < 2e-16 ***
## weight      -0.0070071  0.0003136  -22.34  < 2e-16 ***
## origin       1.1540278  0.3306915    3.49 0.000539 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.272 on 389 degrees of freedom
## Multiple R-squared:  0.702,  Adjusted R-squared:  0.7004
## F-statistic: 458.1 on 2 and 389 DF,  p-value: < 2.2e-16
```

```r
lm.fit = lm(mpg ~ weight + displacement, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ weight + displacement, data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.407  -2.928  -0.357   2.320  16.376
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43.7776194  1.1630993  37.639  < 2e-16 ***
## weight       -0.0057511  0.0007103  -8.097 7.31e-15 ***
## displacement -0.0164971  0.0057653  -2.861  0.00444 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.293 on 389 degrees of freedom
## Multiple R-squared:  0.699,  Adjusted R-squared:  0.6974
## F-statistic: 451.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

```r
lm.fit2 = lm(mpg ~ weight + year, data = my.auto)
summary(lm.fit2)
```

```
##
## Call:
```

```
## lm(formula = mpg ~ weight + year, data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8505 -2.3014 -0.1167  2.0367 14.3555
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.435e+01  4.007e+00  -3.581 0.000386 ***
## weight      -6.632e-03  2.146e-04 -30.911  < 2e-16 ***
## year         7.573e-01  4.947e-02  15.308  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.427 on 389 degrees of freedom
## Multiple R-squared:  0.8082, Adjusted R-squared:  0.8072
## F-statistic: 819.5 on 2 and 389 DF,  p-value: < 2.2e-16
```

And so on.

Qualitative predictors

Transform categorical variables into factors. Remove any variables which cannot be used as predictors.

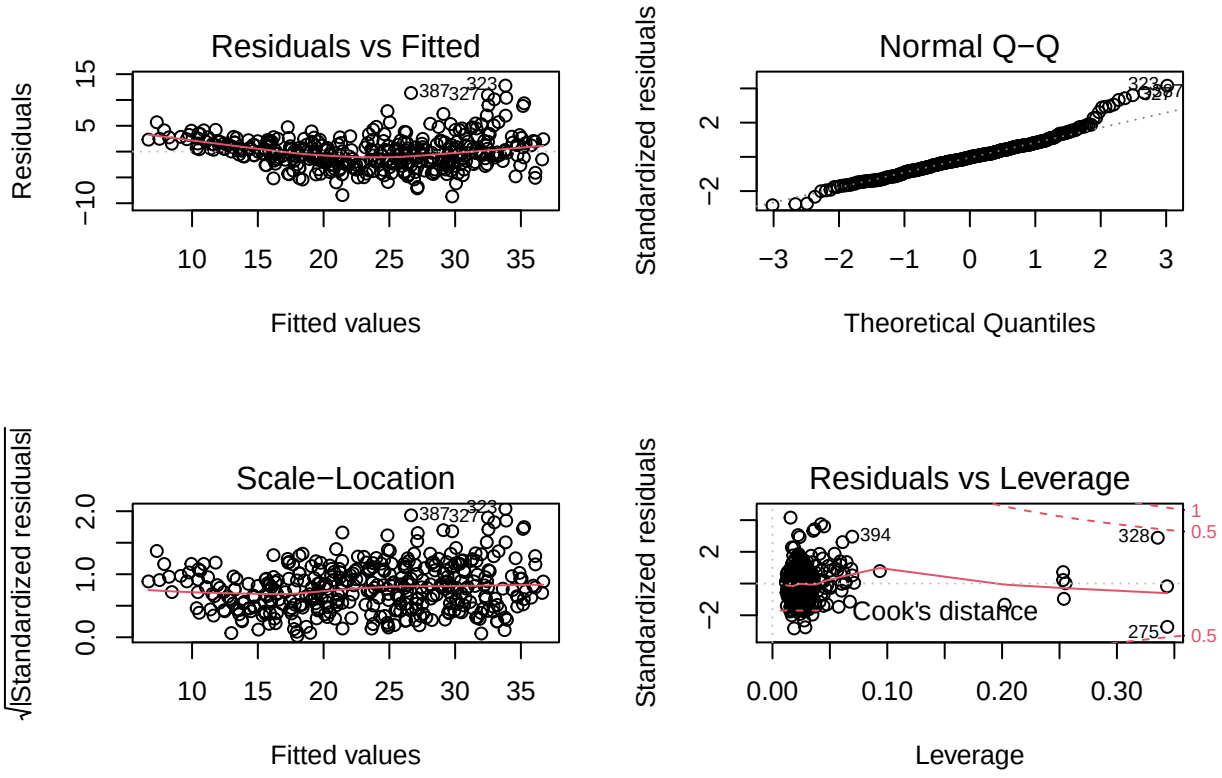
```
my.auto$cylinders = as.factor(my.auto$cylinders)
my.auto$origin = as.factor(my.auto$origin)

lm.all = lm(mpg ~ . - name, data = my.auto)
summary(lm.all)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6797 -1.9373 -0.0678  1.6711 12.7756
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.208e+01  4.541e+00  -4.862 1.70e-06 ***
## cylinders4    6.722e+00  1.654e+00   4.064 5.85e-05 ***
## cylinders5    7.078e+00  2.516e+00   2.813  0.00516 **
## cylinders6    3.351e+00  1.824e+00   1.837  0.06701 .
## cylinders8    5.099e+00  2.109e+00   2.418  0.01607 *
## displacement  1.870e-02  7.222e-03   2.590  0.00997 **
## horsepower   -3.490e-02  1.323e-02  -2.639  0.00866 **
## weight       -5.780e-03  6.315e-04  -9.154  < 2e-16 ***
## acceleration  2.598e-02  9.304e-02   0.279  0.78021
## year          7.370e-01  4.892e-02  15.064  < 2e-16 ***
## origin2       1.764e+00  5.513e-01   3.200  0.00149 **
## origin3       2.617e+00  5.272e-01   4.964 1.04e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.098 on 380 degrees of freedom
## Multiple R-squared:  0.8469, Adjusted R-squared:  0.8425
## F-statistic: 191.1 on 11 and 380 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(lm.all)
```



Interaction terms

Try some combinations of features.

```
lm.fit = lm(mpg ~ horsepower * weight, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower * weight, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7725  -2.2074  -0.2708   1.9973  14.7314
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.356e+01  2.343e+00  27.127  < 2e-16 ***
## horsepower        -2.508e-01  2.728e-02  -9.195  < 2e-16 ***
## weight            -1.077e-02  7.738e-04 -13.921  < 2e-16 ***
## horsepower:weight  5.355e-05  6.649e-06   8.054 9.93e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.93 on 388 degrees of freedom
## Multiple R-squared:  0.7484, Adjusted R-squared:  0.7465
## F-statistic: 384.8 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(mpg ~ acceleration * displacement, data = my.auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ acceleration * displacement, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1540  -2.2872  -0.2687   2.0308  20.4099
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                23.0532678  2.9221224   7.889 3.13e-14 ***
## acceleration                0.8303377  0.1815300   4.574 6.44e-06 ***
## displacement                0.0031393  0.0113352   0.277    0.782
## acceleration:displacement -0.0045805  0.0007899  -5.799 1.38e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.456 on 388 degrees of freedom
## Multiple R-squared:  0.6766, Adjusted R-squared:  0.6741
## F-statistic: 270.5 on 3 and 388 DF,  p-value: < 2.2e-16
```

Non-linear transformations

```
lm.fit = lm(mpg ~ horsepower + cylinders + I(weight^2), data = my.auto)
summary (lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + cylinders + I(weight^2), data = my.auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8647 -2.3891 -0.5468  1.8671 16.1531
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.078e+01  2.315e+00  13.293  < 2e-16 ***
## horsepower  -7.028e-02  1.223e-02  -5.748 1.84e-08 ***
## cylinders4   7.054e+00  2.083e+00   3.386 0.000781 ***
## cylinders5   7.828e+00  3.170e+00   2.470 0.013948 *
## cylinders6   2.141e+00  2.140e+00   1.000 0.317743
## cylinders8   4.916e+00  2.311e+00   2.127 0.034038 *
## I(weight^2) -5.606e-07  9.579e-08  -5.853 1.04e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.097 on 385 degrees of freedom
## Multiple R-squared:  0.7287, Adjusted R-squared:  0.7245
## F-statistic: 172.4 on 6 and 385 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(mpg ~ I(acceleration^2) + year, data = my.auto)
summary (lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ I(acceleration^2) + year, data = my.auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9241  -4.9758  -0.4817   4.9014  18.2030
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -64.428227   6.364051  -10.12  < 2e-16 ***
## I(acceleration^2)   0.023636   0.003571    6.62  1.2e-10 ***
## year                1.079049   0.086004   12.55  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.04 on 389 degrees of freedom
## Multiple R-squared:  0.4041, Adjusted R-squared:  0.4011
## F-statistic: 131.9 on 2 and 389 DF,  p-value: < 2.2e-16
```