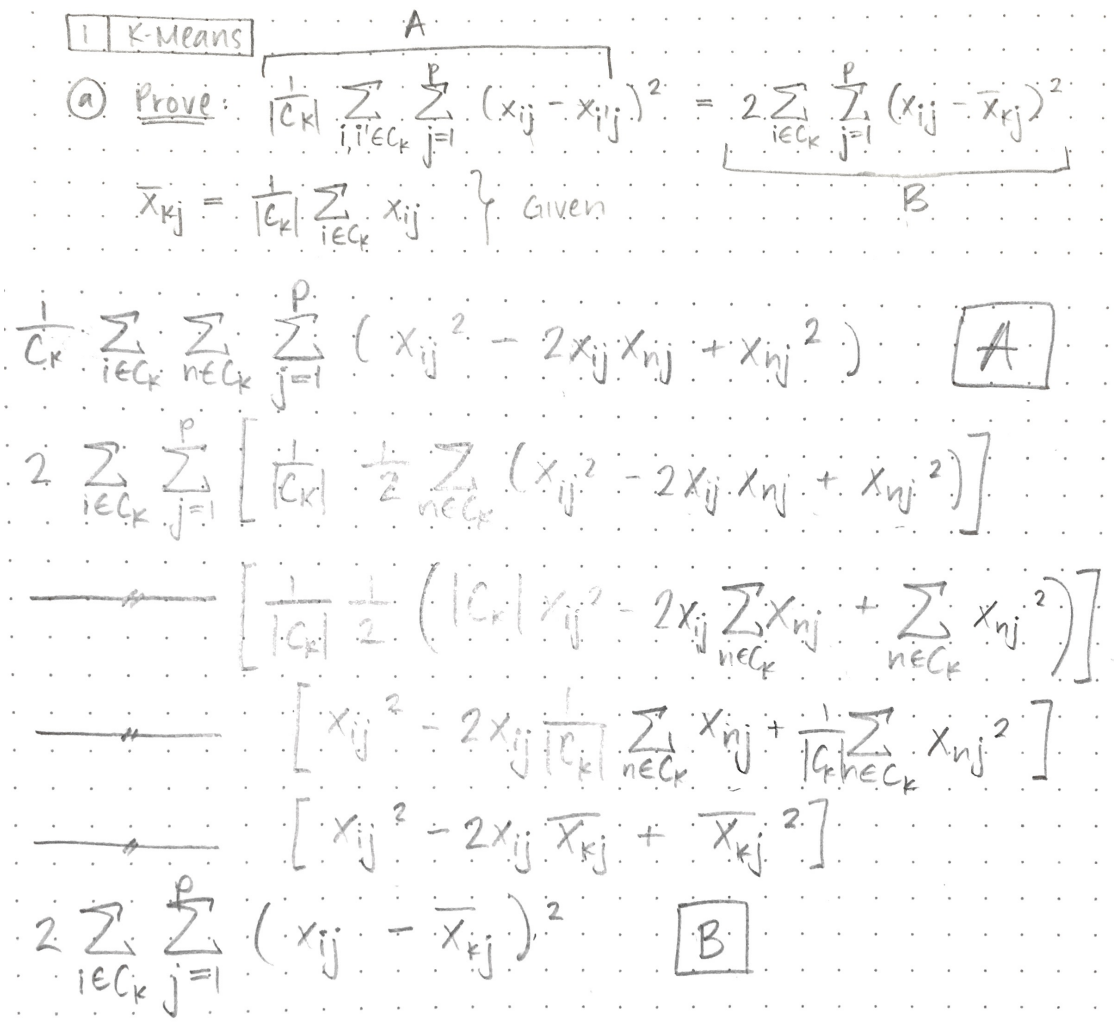
Devon Zuegel // 7 Oct 2015 // STATS 202

Problem Set 2

Problem 1

Part A



Part B

Theorem: The K-means clustering algorithm decreases the objective at each iteration (until it reaches a stable local minimum).

Proof: Let `c` be an arbitrary centroid. On each iteration, we update `c` for a particular cluster to be the mean of all the observations in that group (assigned to the nearest centroid earlier in the iteration). This update will change `c`'s value iff there is some possible `c'` that is on average closer to all of the observations in the group. Thus, the portion of the objective determined by `c` decreases with every iteration until the stopping point (the iteration at which no centroids are updated).

Problem 2

Misc

Common linkage types in hierarchical clustering

Complete – Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.

Single – Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.

Average – Mean intercluster dissimilarity. Compute all pairwise dissimilarities
between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.

Centroid – Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable inversions.

$$\frac{1}{C_k} \sum_{i \in C_k} \sum_{n \in C_k} \sum_{j=1}^{p} \left(x_{ij}^2 - 2 x_{ij} x_{nj} + x_{nj}^2 \right) \quad \boxed{A}$$

$$2 \sum_{i \in C_k} \sum_{j=1}^{p} \left[\frac{1}{|C_k|} \frac{1}{2} \sum_{n \in C_k} \left(x_{ij}^2 - 2 x_{ij} x_{nj} + x_{nj}^2 \right) \right]$$

$$\left[\frac{1}{|C_k|} \frac{1}{2} \left(|C_k| x_{ij}^2 - 2 x_{ij} \sum_{n \in C_k} x_{nj} + \sum_{n \in C_k} x_{nj}^2 \right) \right]$$

$$\left[x_{ij}^2 - 2 x_{ij} \frac{1}{|C_k|} \sum_{n \in C_k} x_{nj} + \frac{1}{|C_k|} \sum_{n \in C_k} x_{nj}^2 \right]$$

$$\left[x_{ij}^2 - 2 x_{ij} \overline{x}_{kj} + \overline{x}_{kj}^2 \right]$$

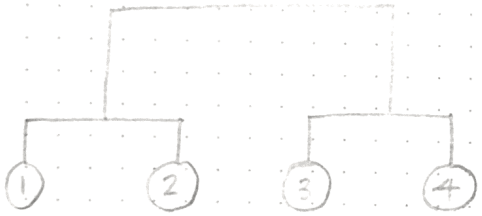
$$2 \sum_{i \in C_k} \sum_{j=1}^{p} \left(x_{ij} - \overline{x}_{kj} \right)^2 \quad \boxed{B}$$

2 | Dissimilarity

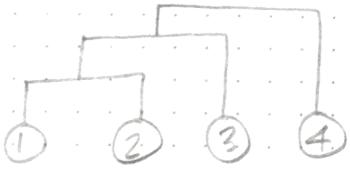
(a) ⟶ using COMPLETE LINKAGE

Dissimilarity Matrix

	1	2	3	4
1	X	X	X	X
2	30	X	X	X
3	40	50	X	X
4	70	80	45	X



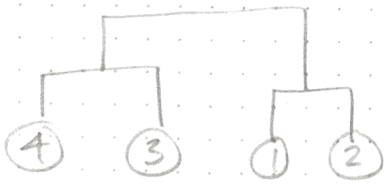
(b) SINGLE-LINKAGE



(c) Cluster A: (1) (2)
Cluster B: (3) (4)

(d) Cluster A: (1) (2) (3)
Cluster B: (4)

(e)



Problem 4

$$y_{\text{linear}} = \beta_0 + \beta_1 X + \epsilon$$

$$y_{\text{cubic}} = \beta_0 + \beta_1 X + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

Part A

Despite the fact that the true relationship between X and Y is linear, I would expect the **training** residual sum of squares (RSS) for the cubic regression to be lower than that of the linear regression. The relationship may be linear, but the cubic regression will do a better job of capturing the noise / random error in the training set, which will cause the error to be lower.

Part B

The **test** RSS for linear regression will likely be lower than that of the cubic regression, because it will not overtrain to that random error.

Part C

The cubic will be at least as good as the linear, because it can "train away" its β_2 and β_3 coefficients to a 0 value, effectively making it a linear function.

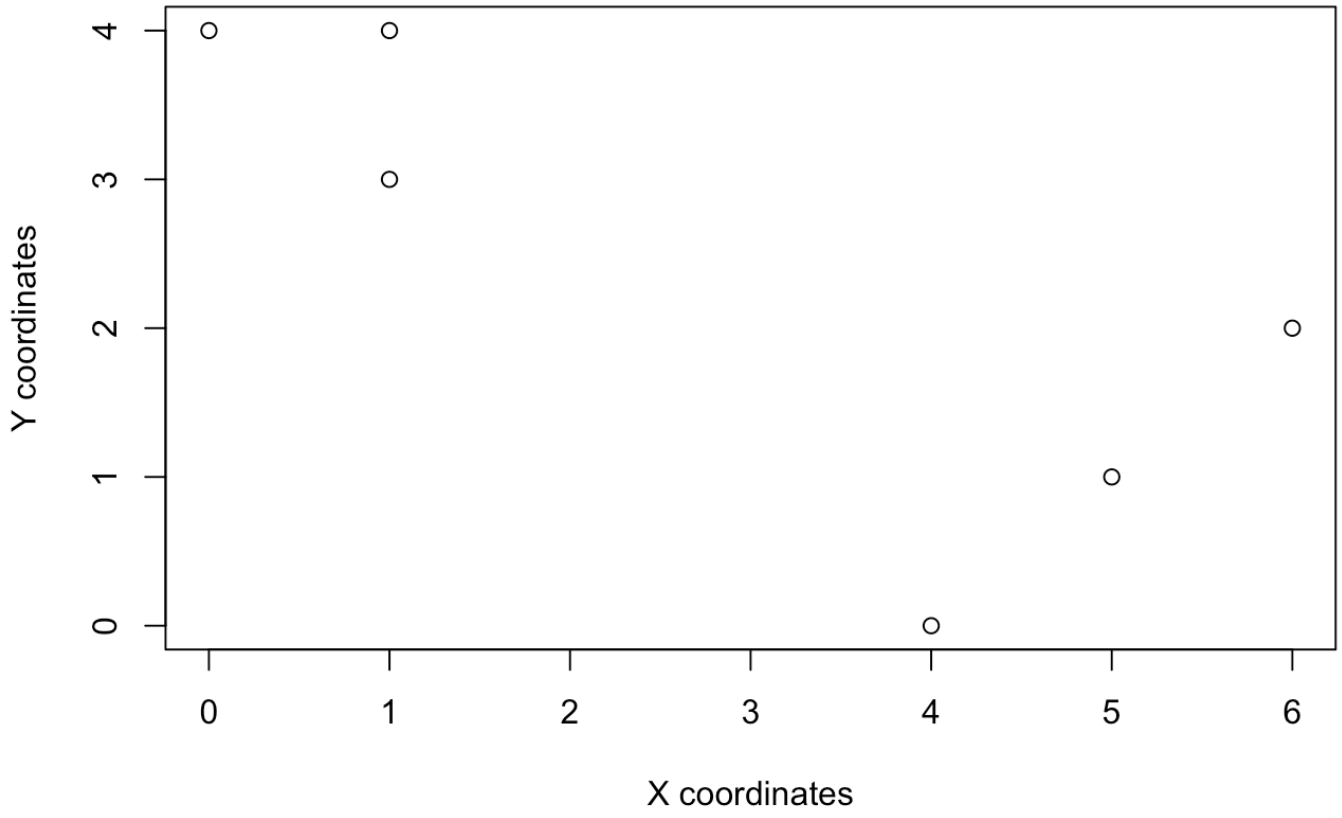
Part D

There is not enough information to tell. If the true relationship is some even-powered polynomial (e.g. x^2 , x^4, …) then linear is slightly better because it can at least generally increase/decrease in the same directions as $x \to \pm\infty$. However, if it's some odd-powered polynomial, then cubic is slightly better because *it* can at least generally increase/decrease in the same directions as $x \to \pm\infty$.

Misc

I accidentally did problem 3 from the textbook, so here's a scatter plot:

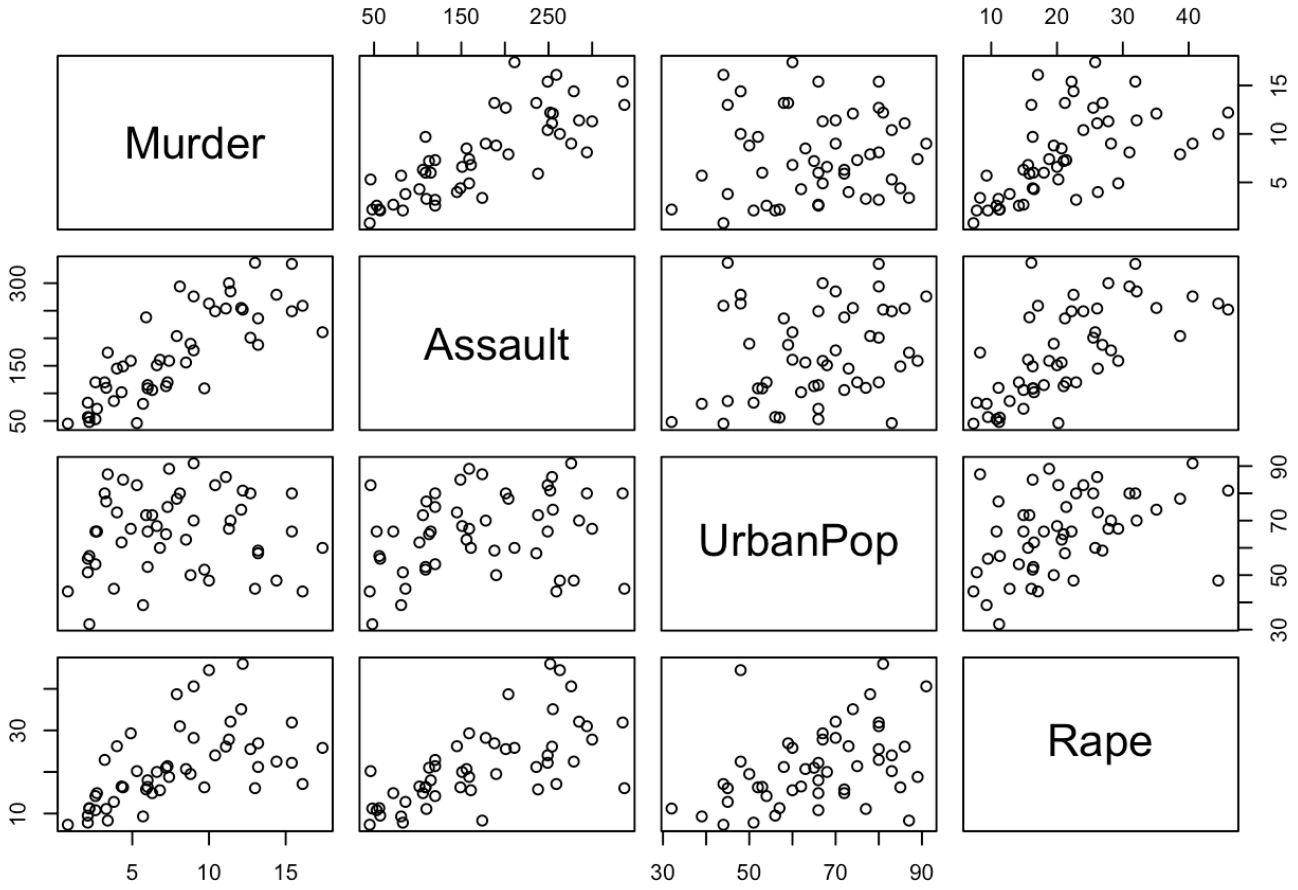
```
pts = matrix(
  c(1, 1, 0, 5, 6, 4, 4, 3, 4, 1, 2, 0),
  nrow = 6,
  ncol = 2
)
colnames(pts) = c('X coordinates', 'Y coordinates')
plot(pts)
```



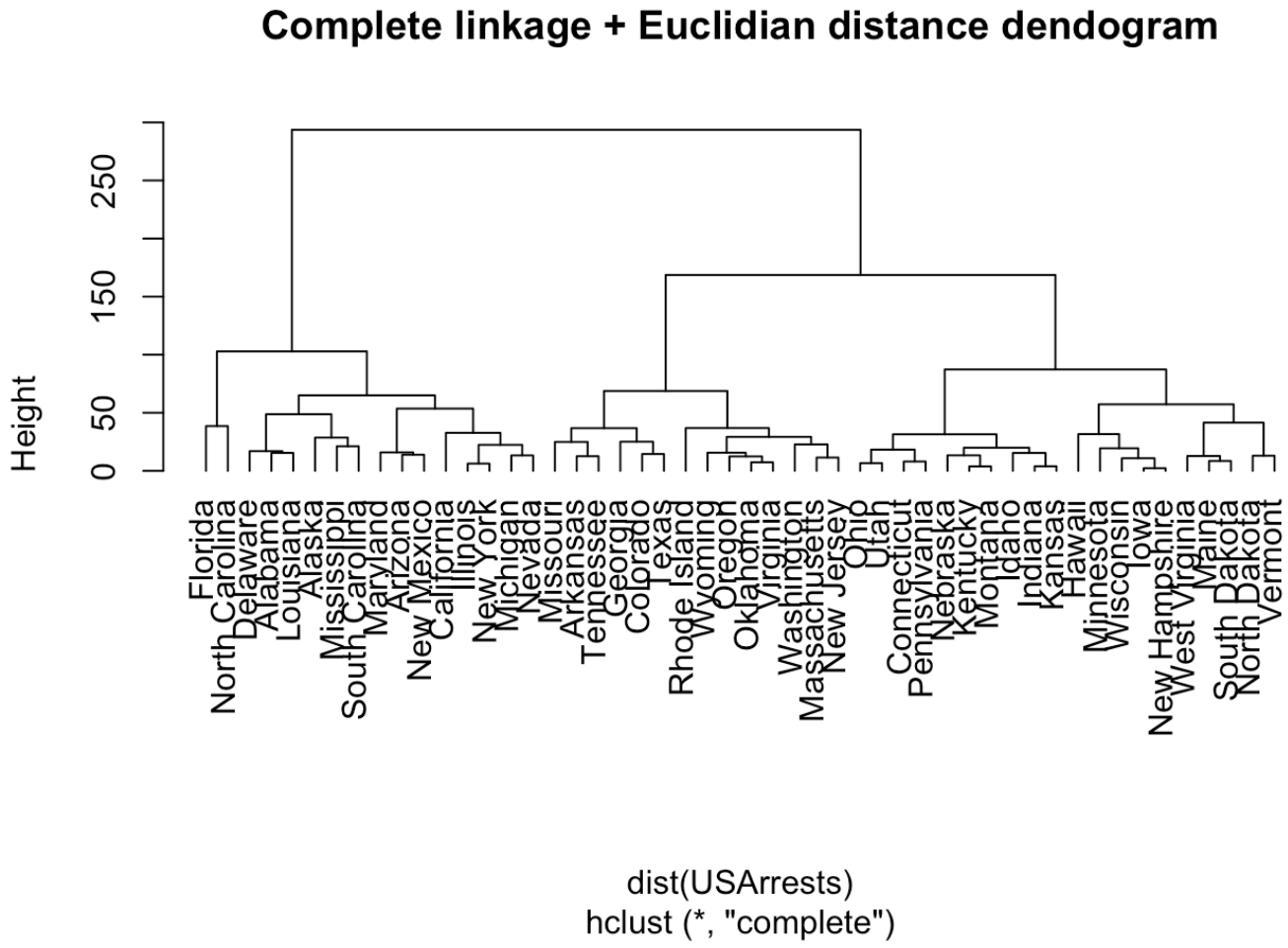
Problem 3

Part A

```
arrests=USArrests
plot(USArrests)
```




```
hc <- hclust(dist(USArrests), "complete")
plot(hc, hang = -1, main = 'Complete linkage + Euclidian distance dendogram')
```



Part B

Note: I relied heavily on this website (https://www.biostars.org/p/86563/) to solve this problem.

```
# cutree returns a vector of cluster membership in the order of the original data rows examine it
clusters <- cutree(hc, k=10)

## to grab a cluster:
# cluster1 <- USArrests[clusters == 1,]

# to add the cluster ID to your data:
all_clusters <- cbind(USArrests, clusterID=clusters)

# examine the data with cluster ids attached, ordered by the 'clusterID' column
all_clusters[order(all_clusters[['clusterID']]), ]
```

```
##                Murder Assault UrbanPop Rape clusterID
## Alabama          13.2     236       58 21.2         1
## Delaware          5.9     238       72 15.8         1
## Louisiana        15.4     249       66 22.2         1
## Alaska           10.0     263       48 44.5         2
## Mississippi      16.1     259       44 17.1         2
## South Carolina   14.4     279       48 22.5         2
## Arizona           8.1     294       80 31.0         3
## Maryland         11.3     300       67 27.8         3
## New Mexico       11.4     285       70 32.1         3
## Arkansas          8.8     190       50 19.5         4
## Colorado          7.9     204       78 38.7         4
## Georgia          17.4     211       60 25.8         4
## Missouri          9.0     178       70 28.2         4
## Tennessee        13.2     188       59 26.9         4
## Texas            12.7     201       80 25.5         4
## California        9.0     276       91 40.6         5
## Illinois         10.4     249       83 24.0         5
## Michigan         12.1     255       74 35.1         5
## Nevada           12.2     252       81 46.0         5
## New York         11.1     254       86 26.1         5
## Connecticut       3.3     110       77 11.1         6
## Idaho             2.6     120       54 14.2         6
## Indiana           7.2     113       65 21.0         6
## Kansas            6.0     115       66 18.0         6
## Kentucky          9.7     109       52 16.3         6
## Montana           6.0     109       53 16.4         6
## Nebraska          4.3     102       62 16.5         6
## Ohio              7.3     120       75 21.4         6
## Pennsylvania      6.3     106       72 14.9         6
## Utah              3.2     120       80 22.9         6
## Florida          15.4     335       80 31.9         7
## North Carolina   13.0     337       45 16.1         7
## Hawaii            5.3      46       83 20.2         8
## Iowa              2.2      56       57 11.3         8
## Minnesota         2.7      72       66 14.9         8
## New Hampshire     2.1      57       56  9.5         8
## Wisconsin         2.6      53       66 10.8         8
## Maine             2.1      83       51  7.8         9
## North Dakota      0.8      45       44  7.3         9
## South Dakota      3.8      86       45 12.8         9
## Vermont           2.2      48       32 11.2         9
## West Virginia     5.7      81       39  9.3         9
## Massachusetts     4.4     149       85 16.3        10
## New Jersey        7.4     159       89 18.8        10
## Oklahoma          6.6     151       68 20.0        10
## Oregon            4.9     159       67 29.3        10
## Rhode Island      3.4     174       87  8.3        10
## Virginia          8.5     156       63 20.7        10
## Washington        4.0     145       73 26.2        10
```

```
## Wyoming            6.8     161       60 15.6        10
```

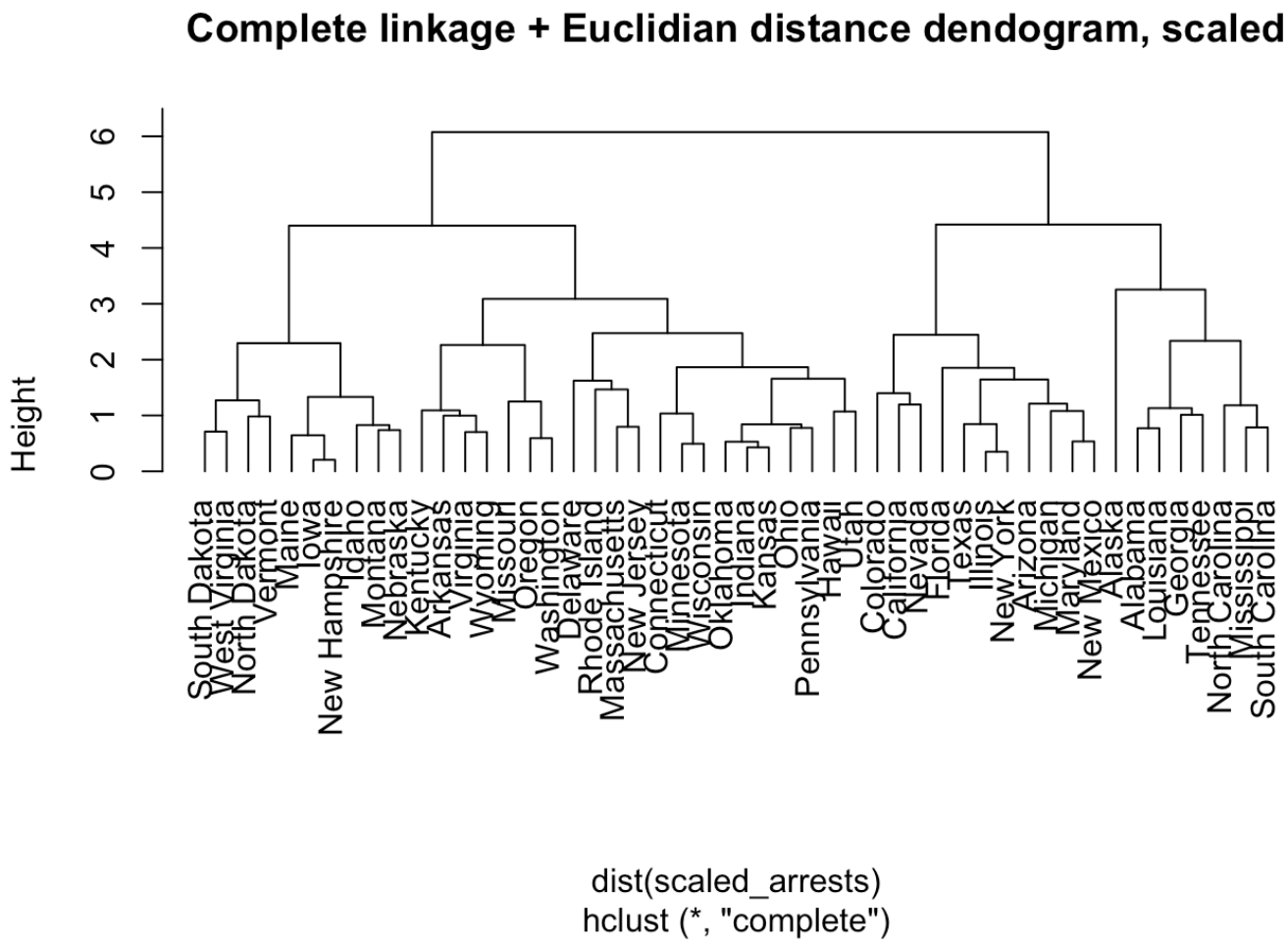
Part C

```
scaled_arrests = scale(USArrests)
scaled_arrests
```

```
##                    Murder     Assault    UrbanPop         Rape
## Alabama         1.24256408  0.78283935 -0.52090661 -0.003416473
## Alaska          0.50786248  1.10682252 -1.21176419  2.484202941
## Arizona         0.07163341  1.47880321  0.99898006  1.042878388
## Arkansas        0.23234938  0.23086801 -1.07359268 -0.184916602
## California      0.27826823  1.26281442  1.75892340  2.067820292
## Colorado        0.02571456  0.39885929  0.86080854  1.864967207
## Connecticut    -1.03041900 -0.72908214  0.79172279 -1.081740768
## Delaware       -0.43347395  0.80683810  0.44629400 -0.579946294
## Florida         1.74767144  1.97077766  0.99898006  1.138966691
## Georgia         2.20685994  0.48285493 -0.38273510  0.487701523
## Hawaii         -0.57123050 -1.49704226  1.20623733 -0.110181255
## Idaho          -1.19113497 -0.60908837 -0.79724965 -0.750769945
## Illinois        0.59970018  0.93883125  1.20623733  0.295524916
## Indiana        -0.13500142 -0.69308401 -0.03730631 -0.024769429
## Iowa           -1.28297267 -1.37704849 -0.58999237 -1.060387812
## Kansas         -0.41051452 -0.66908525  0.03177945 -0.345063775
## Kentucky        0.43898421 -0.74108152 -0.93542116 -0.526563903
## Louisiana       1.74767144  0.93883125  0.03177945  0.103348309
## Maine          -1.30593210 -1.05306531 -1.00450692 -1.434064548
## Maryland        0.80633501  1.55079947  0.10086521  0.701231086
## Massachusetts  -0.77786532 -0.26110644  1.34440885 -0.526563903
## Michigan        0.99001041  1.01082751  0.58446551  1.480613993
## Minnesota      -1.16817555 -1.18505846  0.03177945 -0.676034598
## Mississippi     1.90838741  1.05882502 -1.48810723 -0.441152078
## Missouri        0.27826823  0.08687549  0.30812248  0.743936999
## Montana        -0.41051452 -0.74108152 -0.86633540 -0.515887425
## Nebraska       -0.80082475 -0.82507715 -0.24456358 -0.505210947
## Nevada          1.01296983  0.97482938  1.06806582  2.644350114
## New Hampshire  -1.30593210 -1.36504911 -0.65907813 -1.252564419
## New Jersey     -0.08908257 -0.14111267  1.62075188 -0.259651949
## New Mexico      0.82929443  1.37080881  0.30812248  1.160319648
## New York        0.76041616  0.99882813  1.41349461  0.519730957
## North Carolina  1.19664523  1.99477641 -1.41902147 -0.547916860
## North Dakota   -1.60440462 -1.50904164 -1.48810723 -1.487446939
## Ohio           -0.11204199 -0.60908837  0.65355127  0.017936483
## Oklahoma       -0.27275797 -0.23710769  0.16995096 -0.131534211
## Oregon         -0.66306820 -0.14111267  0.10086521  0.861378259
## Pennsylvania   -0.34163624 -0.77707965  0.44629400 -0.676034598
## Rhode Island   -1.00745957  0.03887798  1.48258036 -1.380682157
## South Carolina  1.51807718  1.29881255 -1.21176419  0.135377743
## South Dakota   -0.91562187 -1.01706718 -1.41902147 -0.900240639
## Tennessee       1.24256408  0.20686926 -0.45182086  0.605142783
## Texas           1.12776696  0.36286116  0.99898006  0.455672088
## Utah           -1.05337842 -0.60908837  0.99898006  0.178083656
## Vermont        -1.28297267 -1.47304350 -2.31713632 -1.071064290
## Virginia        0.16347111 -0.17711080 -0.17547783 -0.056798864
## Washington     -0.86970302 -0.30910395  0.51537975  0.530407436
## West Virginia  -0.47939280 -1.07706407 -1.83353601 -1.273917376
## Wisconsin      -1.19113497 -1.41304662  0.03177945 -1.113770203
```

```
## Wyoming        -0.22683912 -0.11711392 -0.38273510 -0.601299251
## attr(,"scaled:center")
##   Murder  Assault UrbanPop     Rape
##    7.788  170.760   65.540   21.232
## attr(,"scaled:scale")
##    Murder   Assault  UrbanPop      Rape
##  4.355510 83.337661 14.474763  9.366385
```

```
hc <- hclust(dist(scaled_arrests), "complete")
plot(hc, hang = -1, main = 'Complete linkage + Euclidian distance dendogram, scale
d')
```



Part D

A major effect of scaling the variables is that the cluster sizes are more evenly sized. I prefer it with more evenly distributed clusters, because with uneven group sizes you get some really large groups that can contain more variance than they suggest. For instance, before scaling the variables the sheer magnitude (rather than per-capita) of crime committed in each state likely played a role in determining the clusters. As a result, the groups tended to indicate the approximate population of the states. For instance, cluster 6 in

part b tended to contain states with a middle- to low-level population, the main outliers being PA and OH. State population isn't what we're trying to measure here, so it confounds our groups. Rather, we'd rather cluster states by their crime rates relative to each other.

```
clusters <- cutree(hc, k=10)
all_clusters <- cbind(USArrests, clusterID=clusters)
all_clusters[order(all_clusters[['clusterID']]), ]
```

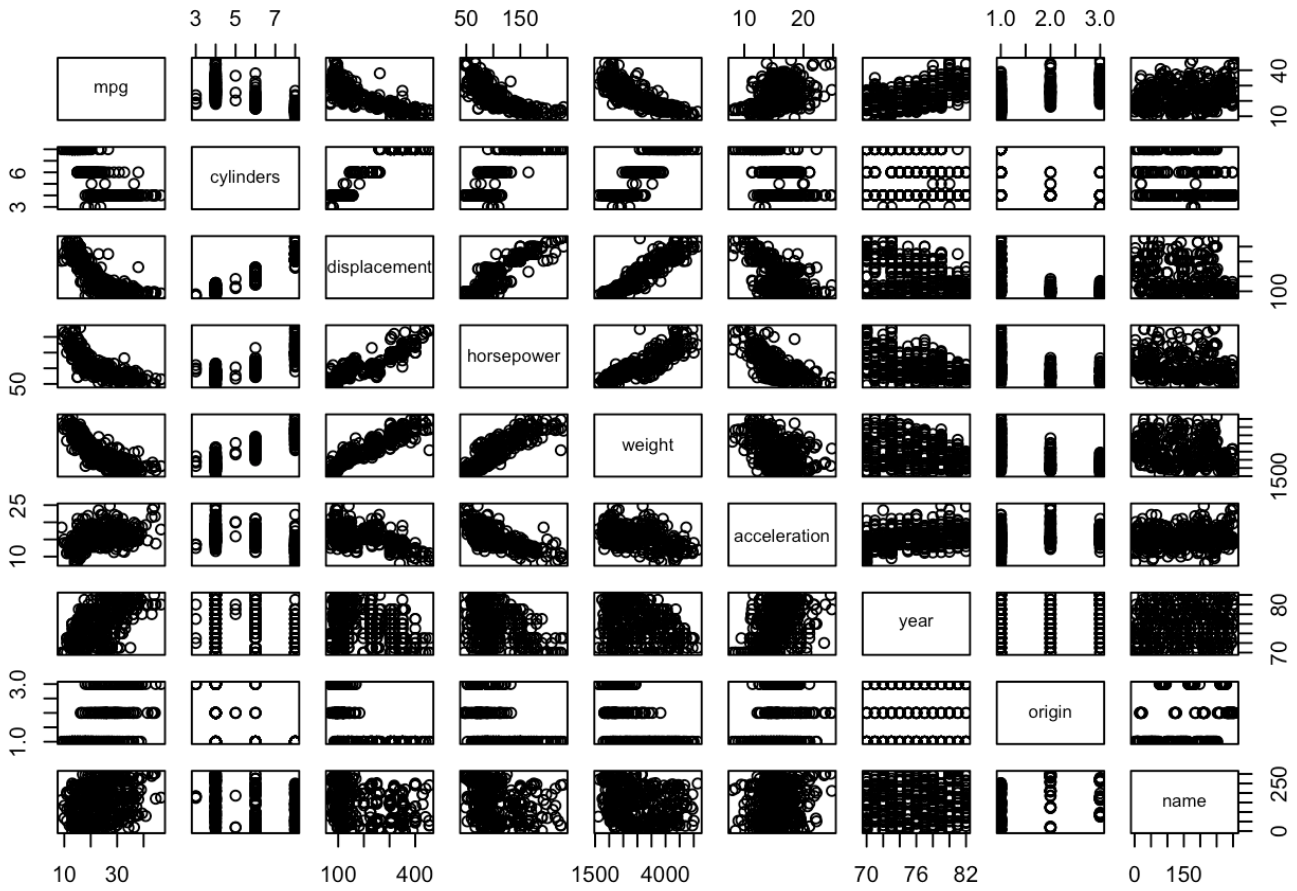
```
##                Murder Assault UrbanPop Rape clusterID
## Alabama          13.2     236       58 21.2         1
## Georgia          17.4     211       60 25.8         1
## Louisiana        15.4     249       66 22.2         1
## Tennessee        13.2     188       59 26.9         1
## Alaska           10.0     263       48 44.5         2
## Arizona           8.1     294       80 31.0         3
## Florida          15.4     335       80 31.9         3
## Illinois         10.4     249       83 24.0         3
## Maryland         11.3     300       67 27.8         3
## Michigan         12.1     255       74 35.1         3
## New Mexico       11.4     285       70 32.1         3
## New York         11.1     254       86 26.1         3
## Texas            12.7     201       80 25.5         3
## Arkansas          8.8     190       50 19.5         4
## Kentucky          9.7     109       52 16.3         4
## Missouri          9.0     178       70 28.2         4
## Oregon            4.9     159       67 29.3         4
## Virginia          8.5     156       63 20.7         4
## Washington        4.0     145       73 26.2         4
## Wyoming           6.8     161       60 15.6         4
## California        9.0     276       91 40.6         5
## Colorado          7.9     204       78 38.7         5
## Nevada           12.2     252       81 46.0         5
## Connecticut       3.3     110       77 11.1         6
## Hawaii            5.3      46       83 20.2         6
## Indiana           7.2     113       65 21.0         6
## Kansas            6.0     115       66 18.0         6
## Minnesota         2.7      72       66 14.9         6
## Ohio              7.3     120       75 21.4         6
## Oklahoma          6.6     151       68 20.0         6
## Pennsylvania      6.3     106       72 14.9         6
## Utah              3.2     120       80 22.9         6
## Wisconsin         2.6      53       66 10.8         6
## Delaware          5.9     238       72 15.8         7
## Massachusetts     4.4     149       85 16.3         7
## New Jersey        7.4     159       89 18.8         7
## Rhode Island      3.4     174       87  8.3         7
## Idaho             2.6     120       54 14.2         8
## Iowa              2.2      56       57 11.3         8
## Maine             2.1      83       51  7.8         8
## Montana           6.0     109       53 16.4         8
## Nebraska          4.3     102       62 16.5         8
## New Hampshire     2.1      57       56  9.5         8
## Mississippi      16.1     259       44 17.1         9
## North Carolina   13.0     337       45 16.1         9
## South Carolina   14.4     279       48 22.5         9
## North Dakota      0.8      45       44  7.3        10
## South Dakota      3.8      86       45 12.8        10
## Vermont           2.2      48       32 11.2        10
```

```
## West Virginia      5.7       81        39   9.3         10
```

Problem 5

Part A

```
library("ISLR")
plot(Auto)
```




```
Auto.quantonly = Auto[, !names(Auto) %in% c("name")]
```

Part B

```
cor(Auto.quantonly)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

Part C

Note: For this problem, I relied heavily on this site (http://www.r-tutor.com/elementary-statistics/simple-linear-regression/significance-test-linear-regression).

```
Auto.fit <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration
+ year + origin, data=Auto.quantonly)
summary(Auto.fit)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = Auto.quantonly)
##
## 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
```

(i)

Yes, there is a clear relationship between the predictors and the response. The p-value is much less than `0.05`, so we reject the null hypothesis.

(ii)

We look at the `Pr(>|t|)` column to determine which predictors appear to have a statistically significant relationship. Here we find that the following predictors have a p-value < 0.05:

- displacement
- weight
- year
- origin

(iii)

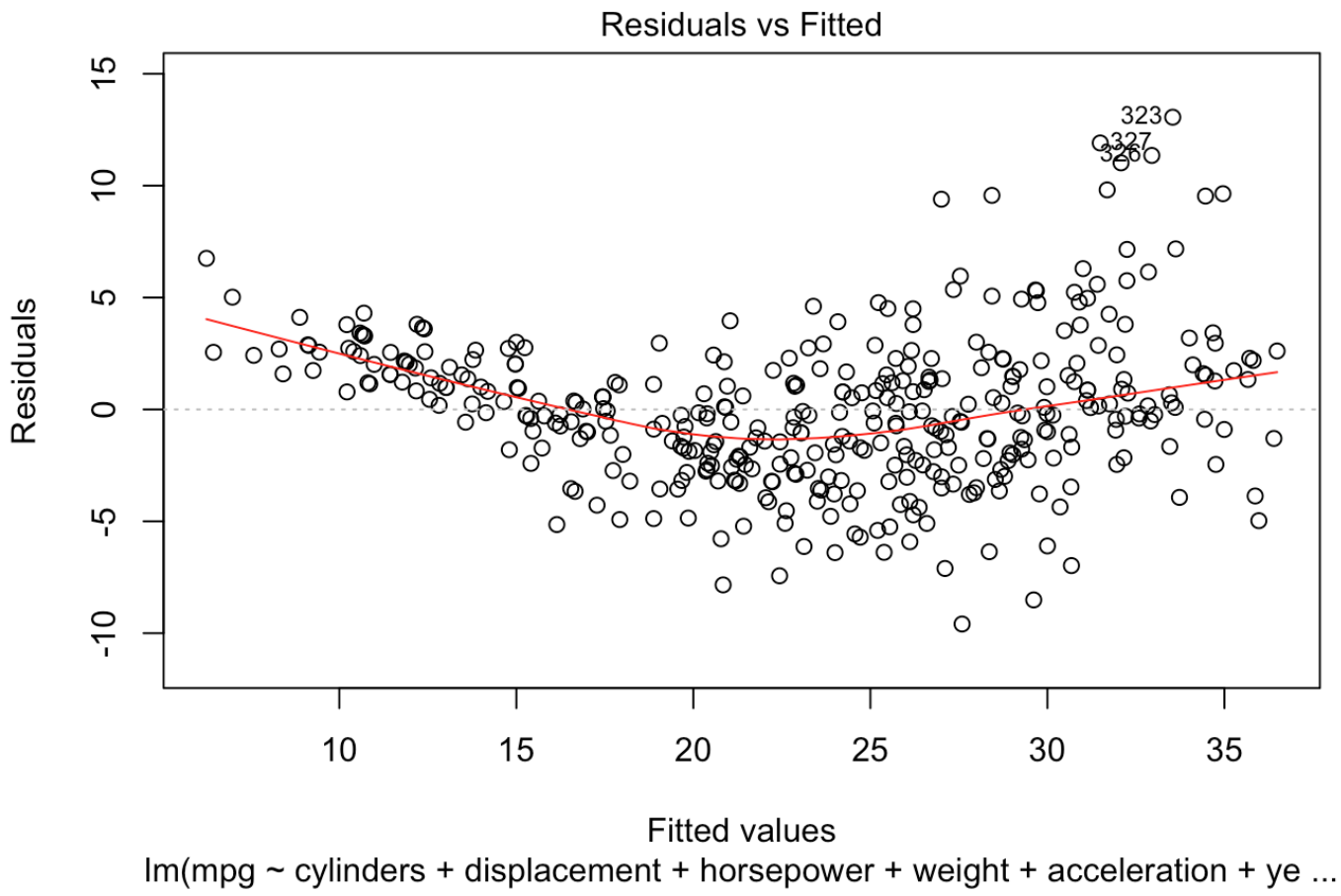
The year the car was made is one of the most important predictors for its mpg. This makes sense intuitively as a sanity check, because older cars don't have catalytic converters and they generally weren't built with efficiency in mind.

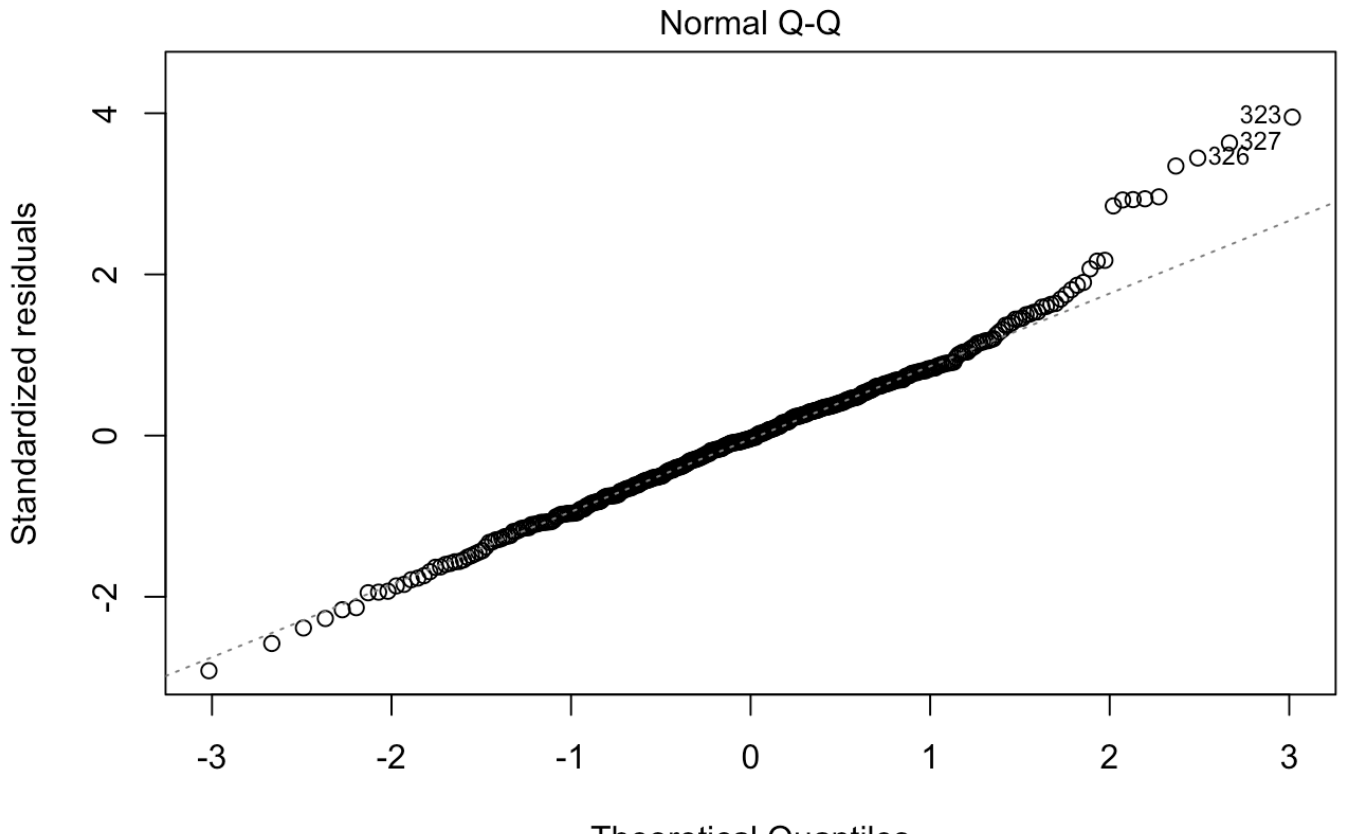
Part D

As you can see from the plot below, there are a few issues with the fit:

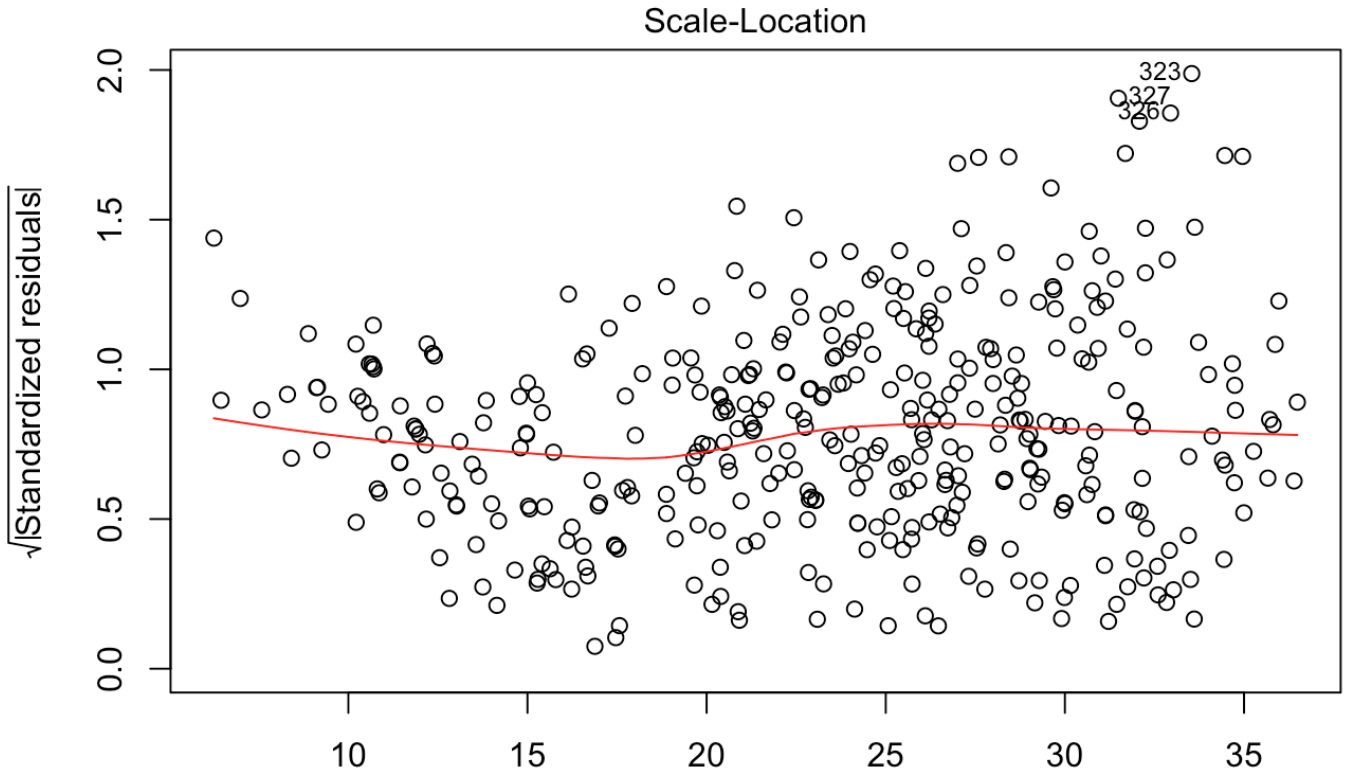
- There are some unusually large outliers in the residuals vs. the fitted (up in the top right hand corner and bottom right). The model fits our data well on the left, but the variance spreads out as we move rightwards.
- Nearly all observations are clustered on the left, but then there's one all the way off to the right with an extremely high leverage. This smells fishy…

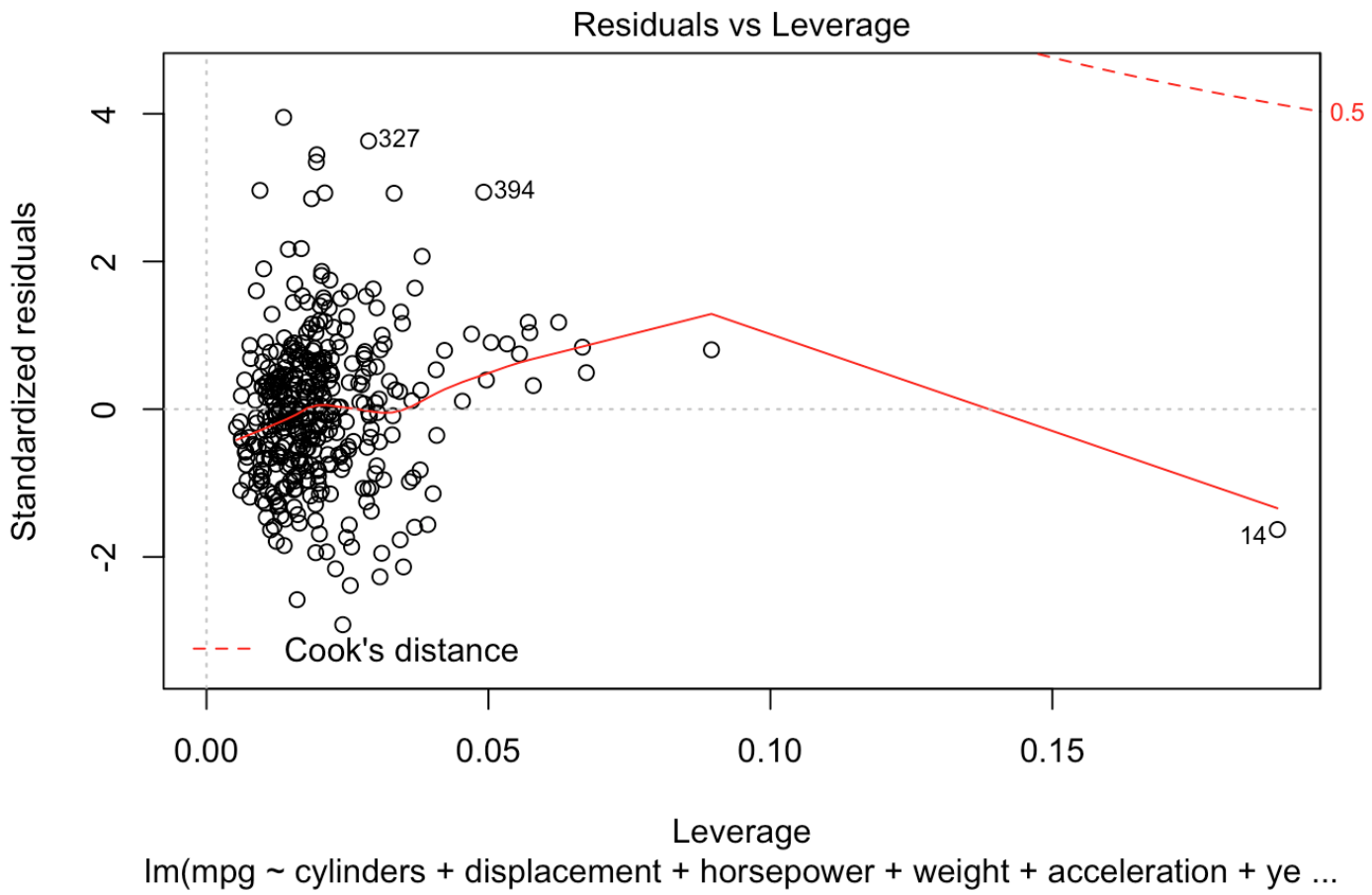
```
plot(Auto.fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + ye ...
323
327
326
15
10
5
0
-5
-10
10
15
20
25
30
35


Normal Q-Q
Standardized residuals
Theoretical Quantiles
323
327
326
4
2
0
-2
-3
-2
-1
0
1
2
3

Theoretical Quantiles
lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + ye ...




Residuals vs Leverage
Standardized residuals
4
2
0
-2
327
394
14
0.5
Cook's distance
0.00
0.05
0.10
0.15
Leverage
lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + ye ...

Problem 6

```
set.seed(1)
x1 = runif(100)
x2 = (0.5 * x1) + rnorm(100)/10

# Create a linear model in which y is a function of x1 and x2.
y = 2 + (2 * x1) + (0.3 * x2) + rnorm(100)
```

Part A

The form of the linear model is

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \\ &= 2 + 2 \cdot x_1 + 0.3 \cdot x_2 + rnorm(100) \end{aligned}$$

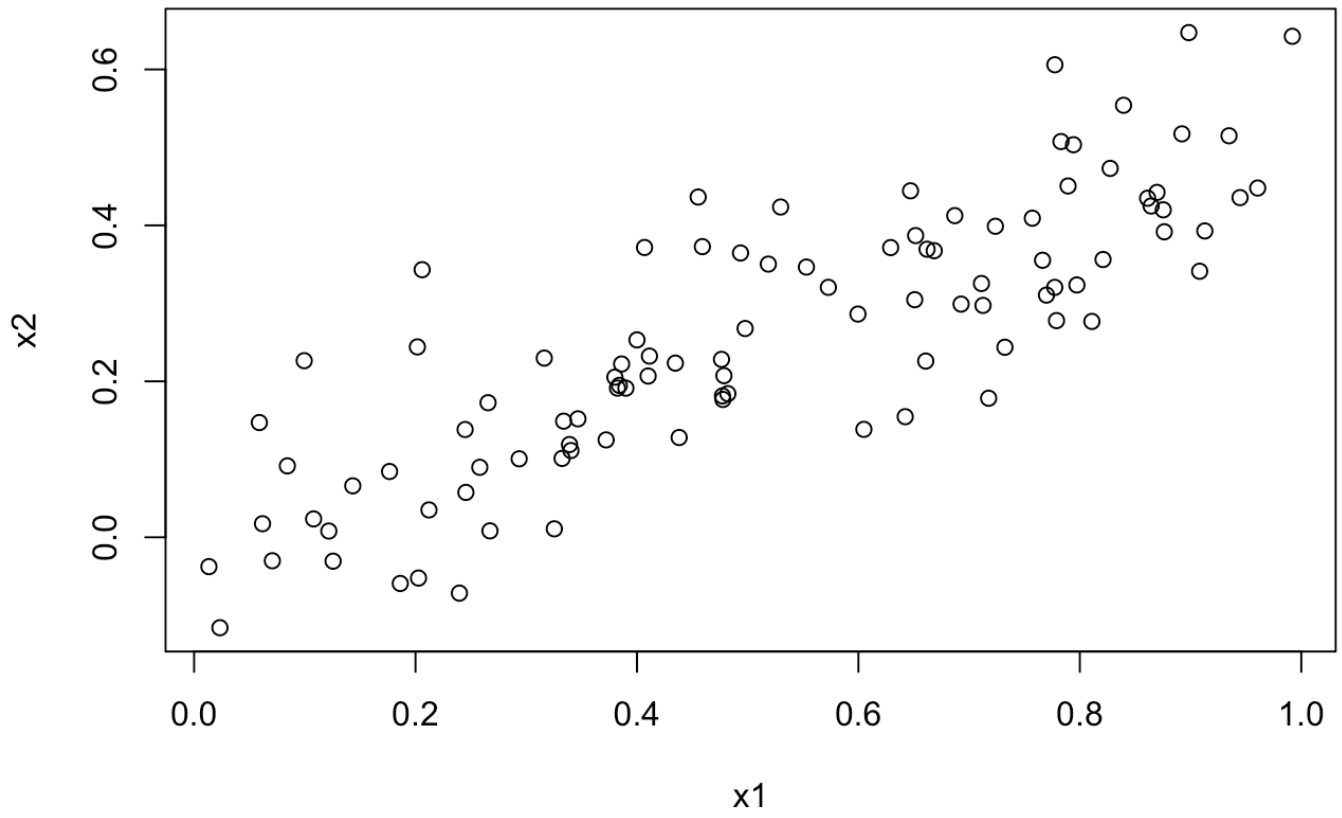
$$\boxed{\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3, \epsilon = rnorm(100)}$$

Part B

```
cor(x1, x2)
```

```
## [1] 0.8351212
```

```
plot(x1, x2)
```



Part C

```
fit <- lm(y ~ x1 + x2)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

The estimated coefficients are $\beta_0 = 2.1305$, $\beta_1 = 1.4396$, and $\beta_2 = 1.0097$, which are at least in the ballpark of the true coefficients (`2, 2, 0.3`). β_2 is smaller than both β_0 and β_1 in both the true and estimated coefficients.

We can reject the null hypothesis $H_0 : \beta_1 = 0$, because the `p = 0.0487 < 0.05` . However, we cannot reject $H_0 : \beta_2 = 0$, because `p = 0.3754 > 0.05` .

Part D

```
fit <- lm(y ~ x1)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

We can reject the null hypothesis $H_0 : \beta_1 = 0$, because the `p = 2.661e-06 < 0.05`. When we throw out x_2, we get a much more impressive `p` value than when we included both x_1 and x_2 in the linear regression.

Part E

```
fit <- lm(y ~ x2)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

We can reject the null hypothesis $H_0 : \beta_2 = 0$, because the `p = 1.366e-06 < 0.05`. When we throw out x_1, we get a much more impressive `p` value than when we included both x_1 and x_2 in the linear regression.

Part F

In a way, yes, I expected Part E to show a non-significant `p`-value, but instead it was even lower than in Part D!

Part G

```
x1 = c(x1, 0.1)
x2 = c(x2, 0.8)
y  = c(y,  6)
```

```
fit <- lm(y ~ x1 + x2)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
fit <- lm(y ~ x1)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

```
fit <- lm(y ~ x2)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

This new observation flips the relationship we saw before. Previously, we saw a more significant p value for x_1 when we fit the lm to both x_1 and x_2, while now we have a more significant p value for x_2. However, they still both indicate a low p value in the lms where we fit the two variables independently.

Problem 7

Part A

We can see a statistically significant association between crim and zn, indus, nox, rm, age, dis, rad, tax, ptratio, black, and lstat, all of which have p values `< 0.05`.

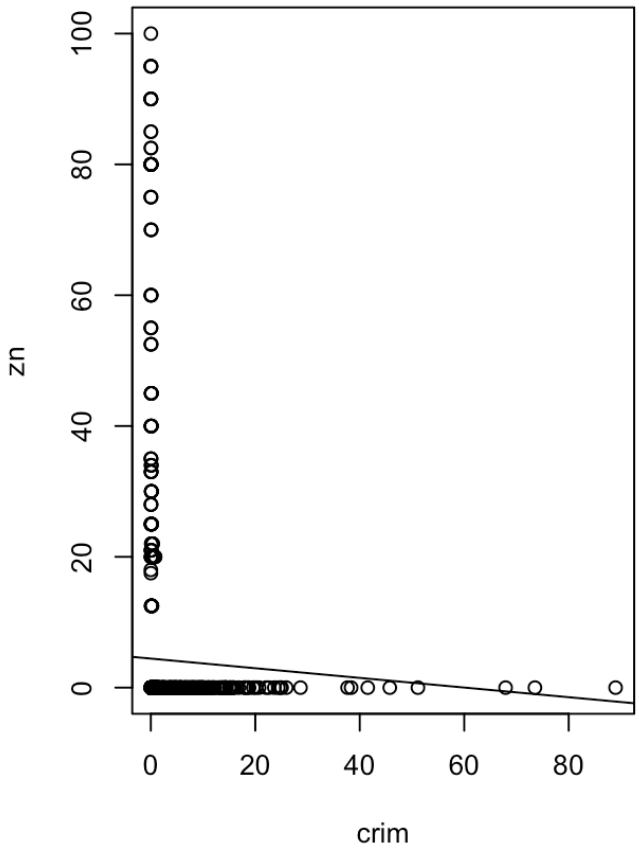
```
library(MASS)
summary(Boston)
```

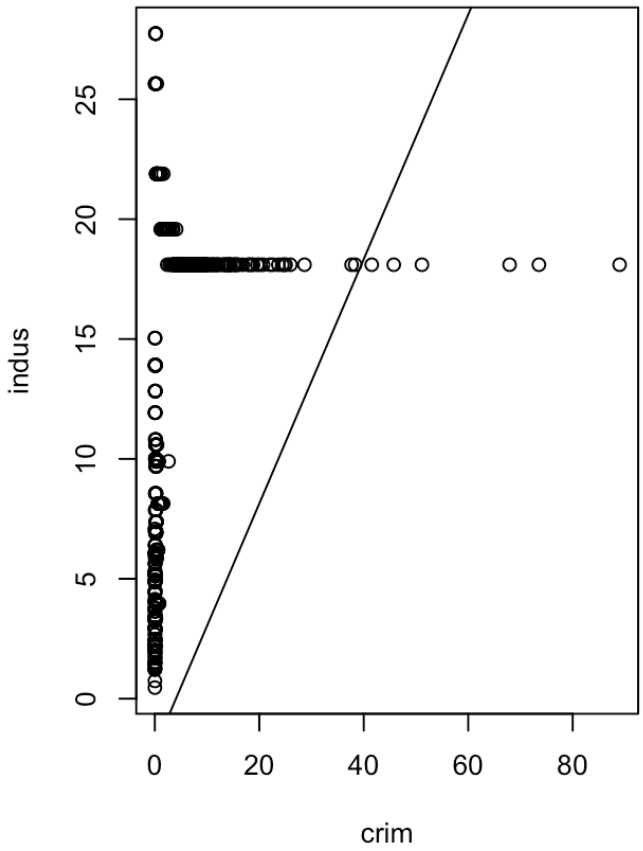
```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm              age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

```
layout(matrix(c(1,2), 2, 2, byrow = TRUE))

for (i in 3:length(Boston) - 1) {
  cat(colnames(Boston)[i])
  cat('\n==================')
  fit = lm(crim ~ Boston[,i], data=Boston)
  print(summary(fit))
  plot(Boston[,c(1, i)])
  abline(fit)
}
```

```
## zn
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## Boston[, i] -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

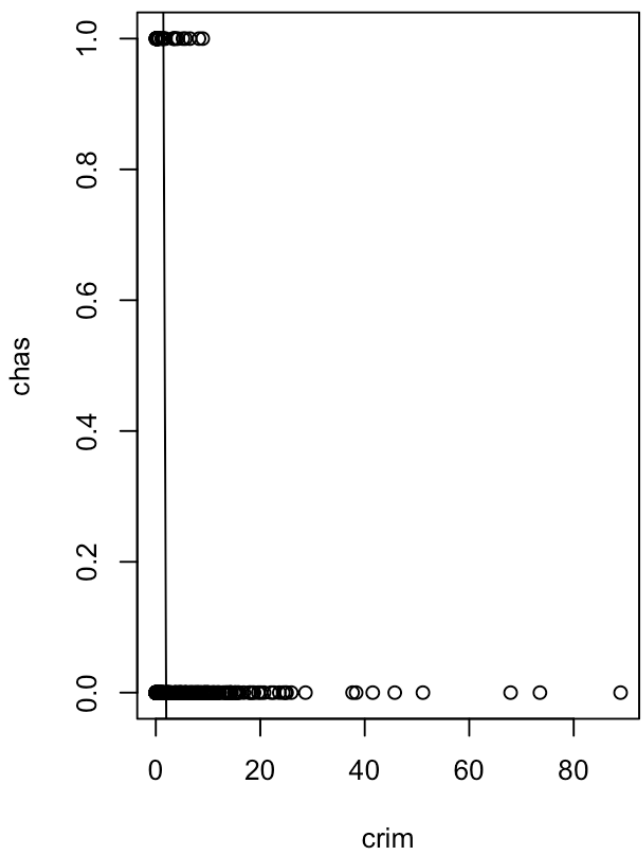
```
## indus
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## Boston[, i]  0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

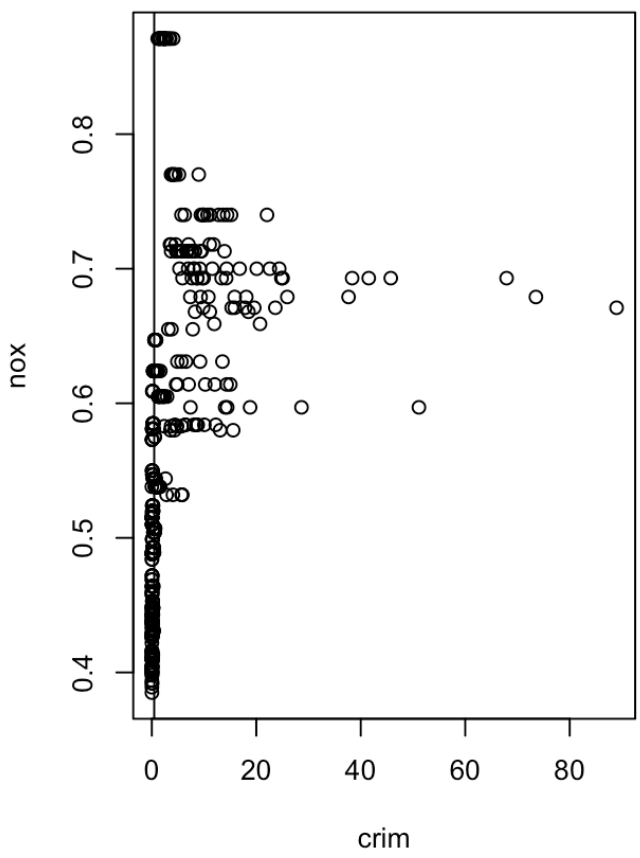




```
## chas
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## Boston[, i]  -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

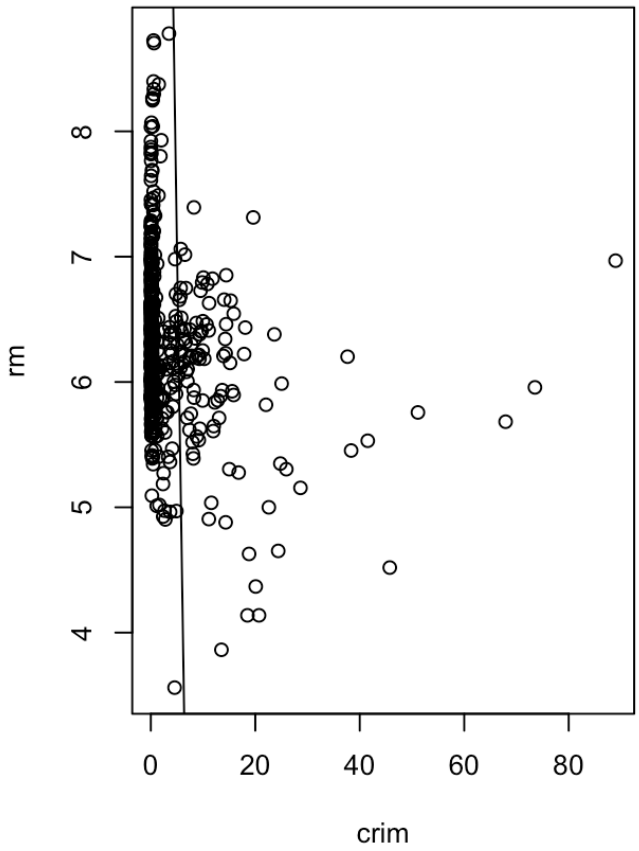
```
## nox
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## Boston[, i]   31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

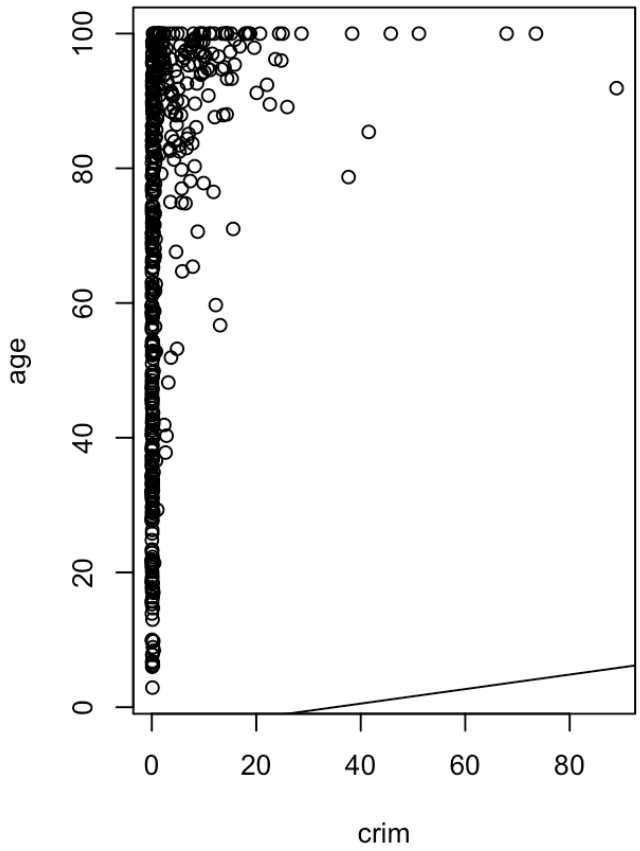





```
## rm
## ===================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## Boston[, i]   -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

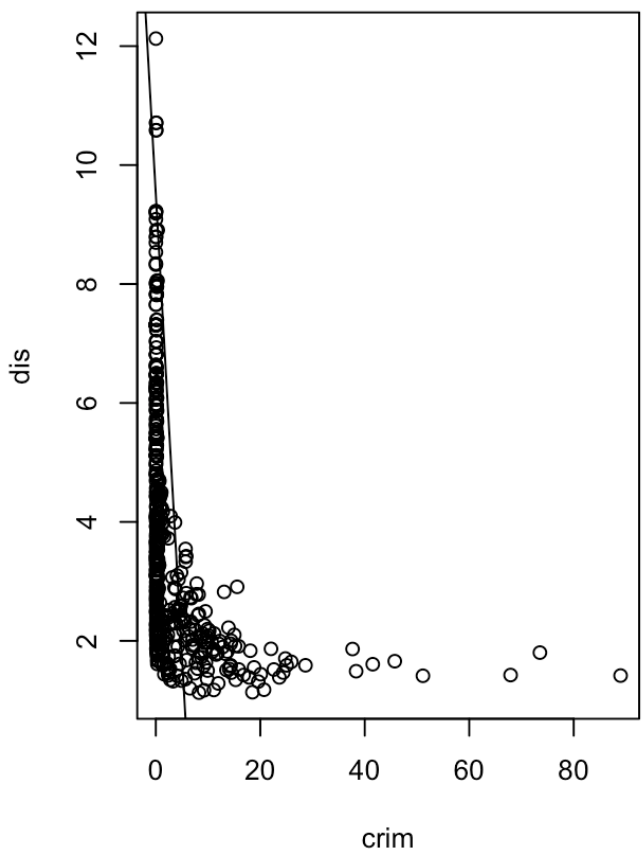
```
## age
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## Boston[, i]  0.10779    0.01274   8.463 2.85e-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:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

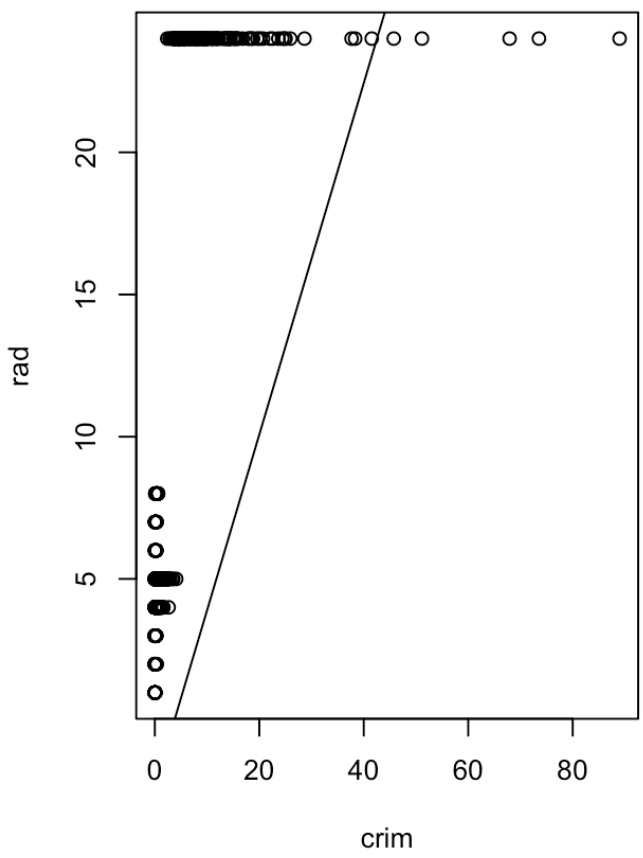




```
## dis
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## Boston[, i]  -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

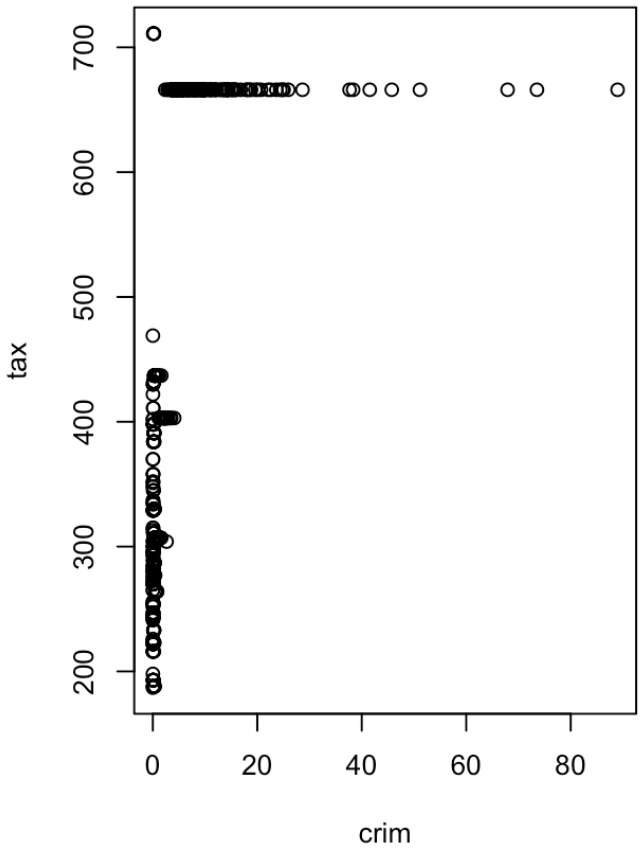
```
## rad
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## Boston[, i]  0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

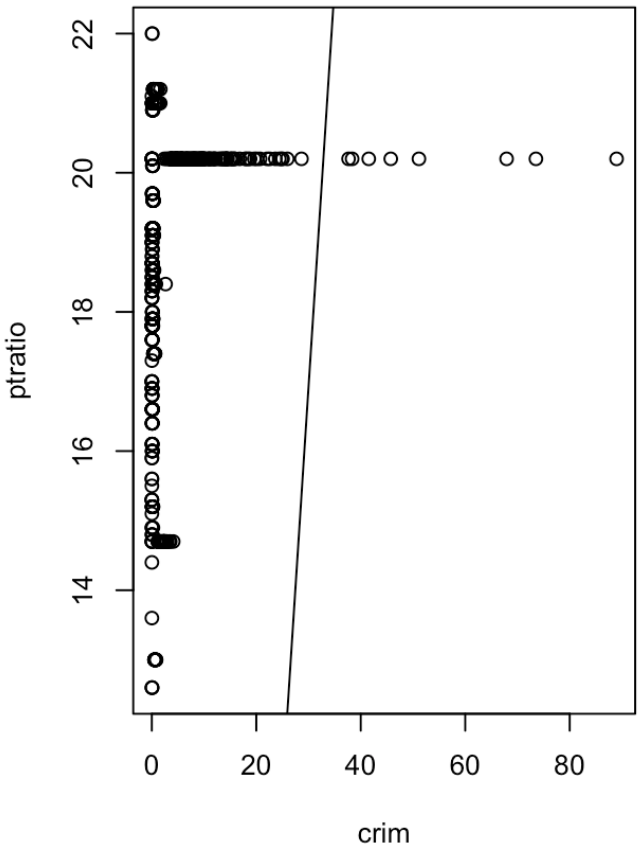





```
## tax
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## Boston[, i]  0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

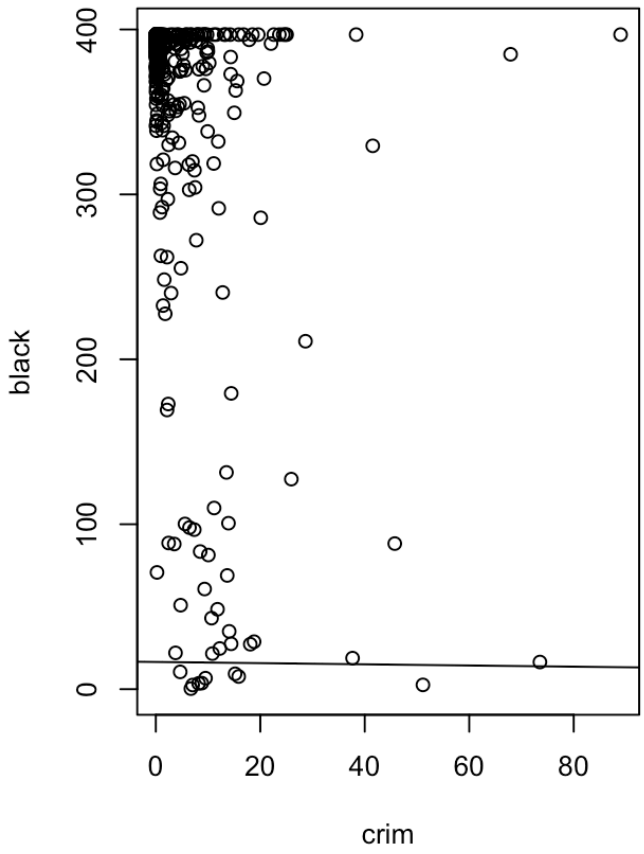
```
## ptratio
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## Boston[, i]   1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

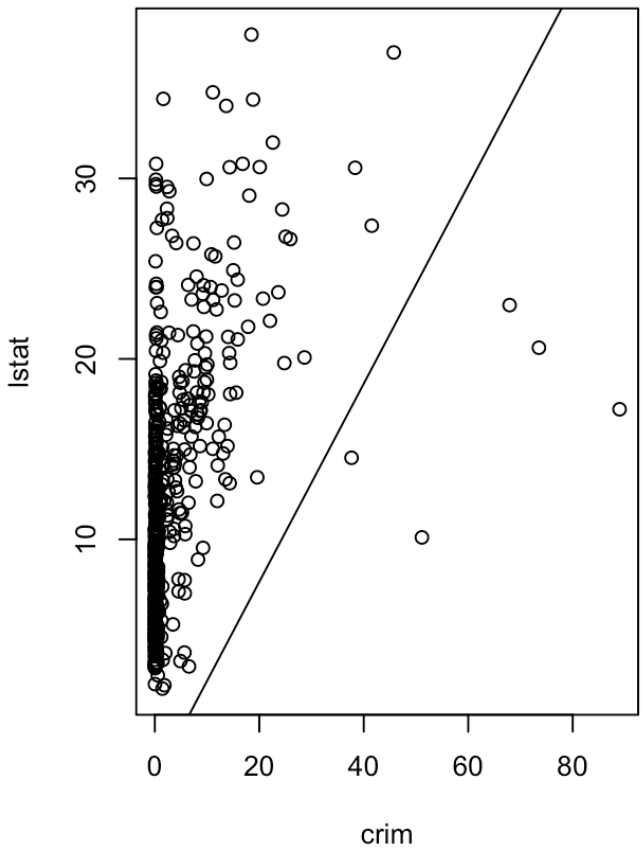




```
## black
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## Boston[, i] -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
## lstat
## ==================
## Call:
## lm(formula = crim ~ Boston[, i], data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## Boston[, i]  0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```





Part B

We can reject the null hypothesis for:

- zn
- dis
- rad
- black
- medv

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

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

Part C

Far fewer relationships are marked as “statistically significant” in Part B.

```
univ_coefficnts = c()
for (i in 3:length(Boston) - 1) {
  univ_coefficnts[i - 1] <- summary( lm(crim ~ Boston[,i], data=Boston) )$coeffici
ents[2]
}
print(univ_coefficnts)
```

```
##  [1] -0.07393498  0.50977633 -1.89277655 31.24853120 -2.68405122
##  [6]  0.10778623 -1.55090168  0.61791093  0.02974225  1.15198279
## [11] -0.03627964  0.54880478
```

```
mult_coefficnts = summary(fit)$coefficients[2:13, 1]
print(mult_coefficnts)
```

```
##            zn         indus          chas           nox            rm
##   0.044855215  -0.063854824  -0.749133611 -10.313534912   0.430130506
##           age           dis           rad           tax       ptratio
##   0.001451643  -0.987175726   0.588208591  -0.003780016  -0.271080558
##         black         lstat
##  -0.007537505   0.126211376
```

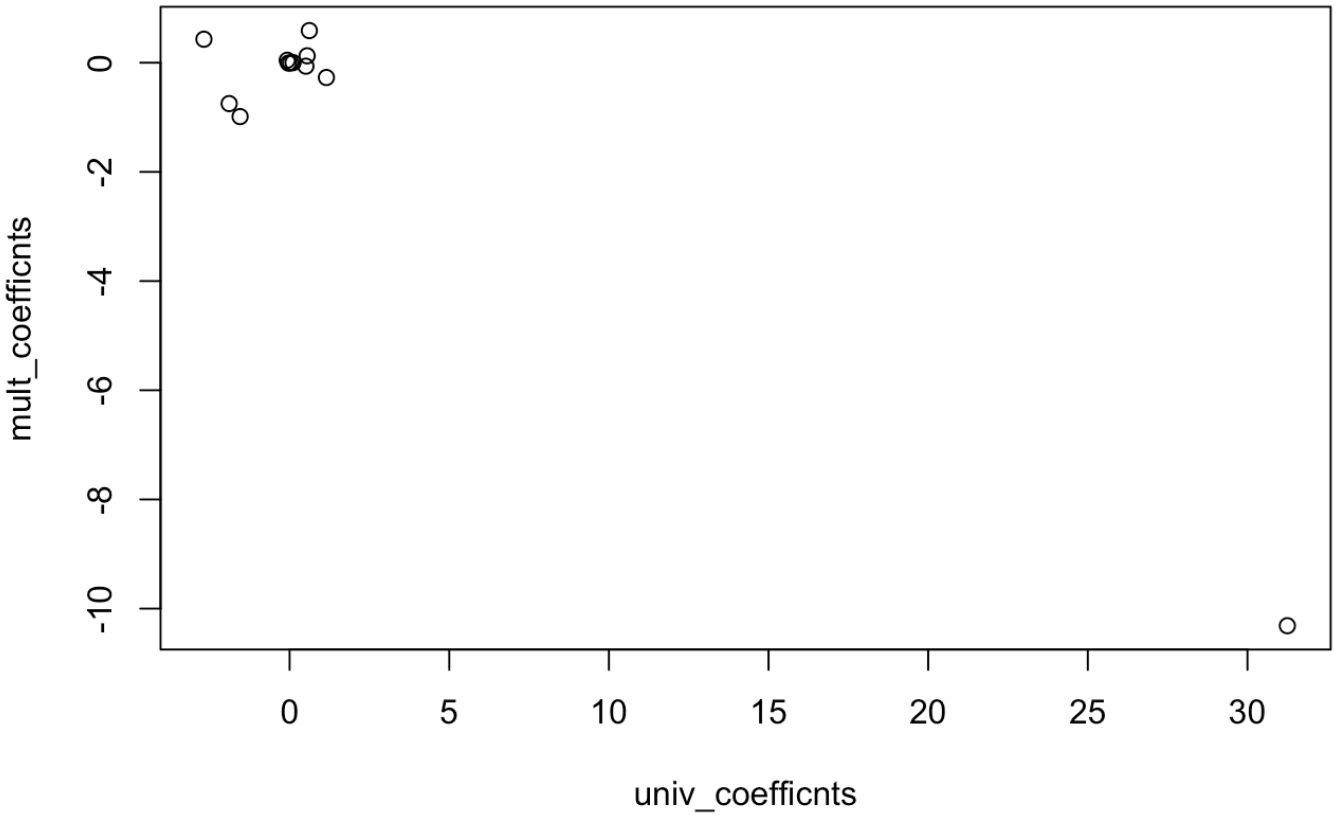
```
cat(length(univ_coefficnts))
```

```
## 12
```

```
cat(length(mult_coefficnts))
```

```
## 12
```

```
plot(univ_coefficnts, mult_coefficnts)
```



Part D

```
layout(matrix(c(1,2), 2, 2, byrow = TRUE))

for (i in 3:length(Boston) - 1) {
  cat(colnames(Boston)[i])
  cat('\n==================')
  fit = lm(crim ~ poly(Boston[,i], degree=3, raw=TRUE), data=Boston)
  print(summary(fit))
  plot(Boston[,c(1, i)])
  abline(fit)
  predict(fit, data.frame(crim = 0:80))
}
```

```
## zn
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 4.846e+00  4.330e-01  11.192
## poly(Boston[, i], degree = 3, raw = TRUE)1 -3.322e-01  1.098e-01  -3.025
## poly(Boston[, i], degree = 3, raw = TRUE)2  6.483e-03  3.861e-03   1.679
## poly(Boston[, i], degree = 3, raw = TRUE)3 -3.776e-05  3.139e-05  -1.203
##                                            Pr(>|t|)
## (Intercept)                                 < 2e-16 ***
## poly(Boston[, i], degree = 3, raw = TRUE)1  0.00261 **
## poly(Boston[, i], degree = 3, raw = TRUE)2  0.09375 .
## poly(Boston[, i], degree = 3, raw = TRUE)3  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

```
## indus
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 3.6625683  1.5739833   2.327
## poly(Boston[, i], degree = 3, raw = TRUE)1 -1.9652129  0.4819901  -4.077
## poly(Boston[, i], degree = 3, raw = TRUE)2  0.2519373  0.0393221   6.407
## poly(Boston[, i], degree = 3, raw = TRUE)3 -0.0069760  0.0009567  -7.292
##                                            Pr(>|t|)
## (Intercept)                                  0.0204 *
## poly(Boston[, i], degree = 3, raw = TRUE)1 5.30e-05 ***
## poly(Boston[, i], degree = 3, raw = TRUE)2 3.42e-10 ***
## poly(Boston[, i], degree = 3, raw = TRUE)3 1.20e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

zn
crim
0
20
40
60
80
100
0
20
40
60
80

indus
crim
0
5
10
15
20
25
0
20
40
60
80

```
## chas
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients: (2 not defined because of singularities)
##                                             Estimate Std. Error t value
## (Intercept)                                   3.7444     0.3961   9.453
## poly(Boston[, i], degree = 3, raw = TRUE)1  -1.8928     1.5061  -1.257
## poly(Boston[, i], degree = 3, raw = TRUE)2       NA         NA      NA
## poly(Boston[, i], degree = 3, raw = TRUE)3       NA         NA      NA
##                                             Pr(>|t|)
## (Intercept)                                   <2e-16 ***
## poly(Boston[, i], degree = 3, raw = TRUE)1    0.209
## poly(Boston[, i], degree = 3, raw = TRUE)2       NA
## poly(Boston[, i], degree = 3, raw = TRUE)3       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

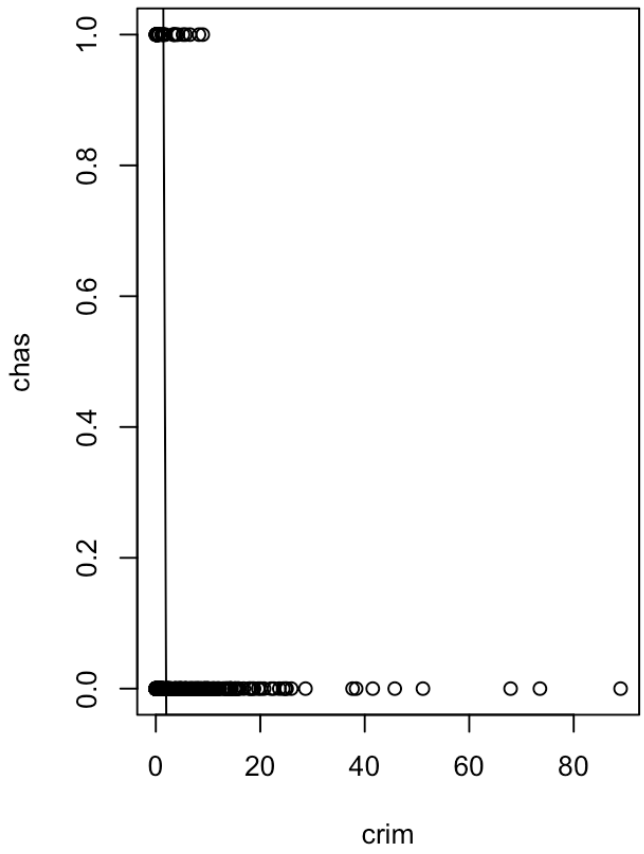
```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

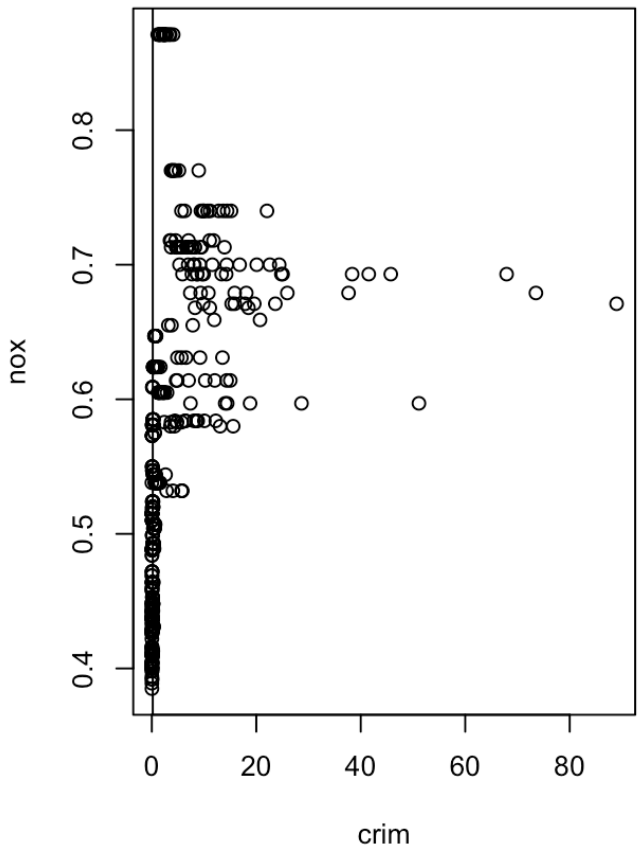
```
## Warning in predict.lm(fit, data.frame(crim = 0:80)): prediction from a
## rank-deficient fit may be misleading
```

```
## nox
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##                                            Estimate Std. Error t value
## (Intercept)                                  233.09      33.64   6.928
## poly(Boston[, i], degree = 3, raw = TRUE)1 -1279.37     170.40  -7.508
## poly(Boston[, i], degree = 3, raw = TRUE)2  2248.54     279.90   8.033
## poly(Boston[, i], degree = 3, raw = TRUE)3 -1245.70     149.28  -8.345
##                                            Pr(>|t|)
## (Intercept)                                1.31e-11 ***
## poly(Boston[, i], degree = 3, raw = TRUE)1 2.76e-13 ***
## poly(Boston[, i], degree = 3, raw = TRUE)2 6.81e-15 ***
## poly(Boston[, i], degree = 3, raw = TRUE)3 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```


chas
crim
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80


nox
crim
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80

```
## rm
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                 112.6246    64.5172   1.746
## poly(Boston[, i], degree = 3, raw = TRUE)1 -39.1501    31.3115  -1.250
## poly(Boston[, i], degree = 3, raw = TRUE)2   4.5509     5.0099   0.908
## poly(Boston[, i], degree = 3, raw = TRUE)3  -0.1745     0.2637  -0.662
##                                             Pr(>|t|)
## (Intercept)                                   0.0815 .
## poly(Boston[, i], degree = 3, raw = TRUE)1    0.2118
## poly(Boston[, i], degree = 3, raw = TRUE)2    0.3641
## poly(Boston[, i], degree = 3, raw = TRUE)3    0.5086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

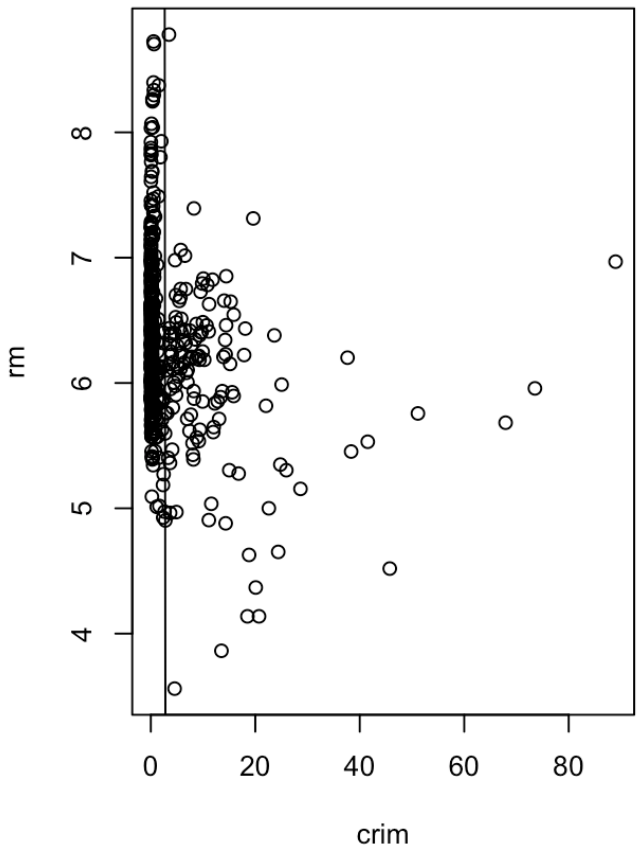
```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

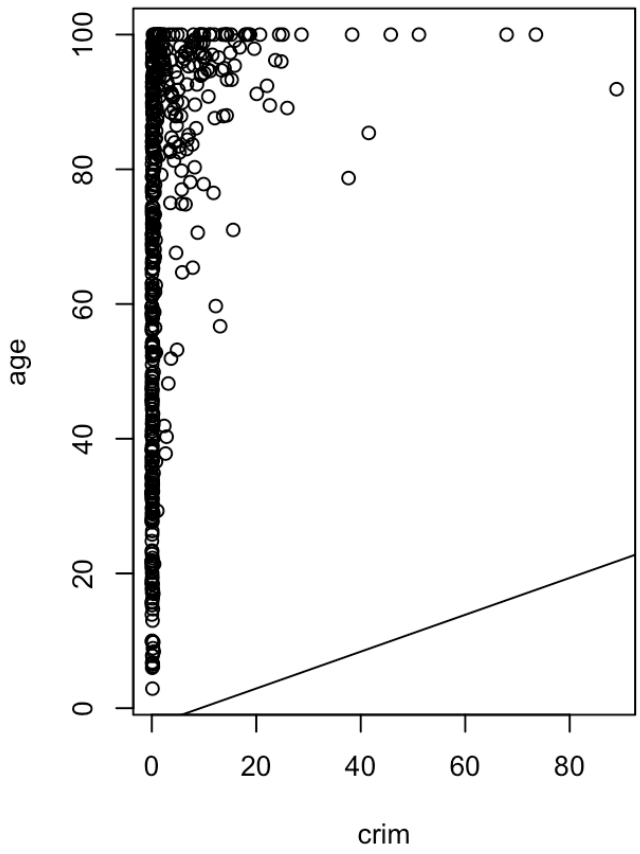
```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

```
## age
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 -2.549e+00  2.769e+00  -0.920
## poly(Boston[, i], degree = 3, raw = TRUE)1  2.737e-01  1.864e-01   1.468
## poly(Boston[, i], degree = 3, raw = TRUE)2 -7.230e-03  3.637e-03  -1.988
## poly(Boston[, i], degree = 3, raw = TRUE)3  5.745e-05  2.109e-05   2.724
##                                             Pr(>|t|)
## (Intercept)                                  0.35780
## poly(Boston[, i], degree = 3, raw = TRUE)1   0.14266
## poly(Boston[, i], degree = 3, raw = TRUE)2   0.04738 *
## poly(Boston[, i], degree = 3, raw = TRUE)3   0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```


rm
crim
0
20
40
60
80
4
5
6
7
8


age
crim
0
20
40
60
80
100

```
## dis
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                  30.0476     2.4459  12.285
## poly(Boston[, i], degree = 3, raw = TRUE)1 -15.5543     1.7360  -8.960
## poly(Boston[, i], degree = 3, raw = TRUE)2   2.4521     0.3464   7.078
## poly(Boston[, i], degree = 3, raw = TRUE)3  -0.1186     0.0204  -5.814
##                                            Pr(>|t|)
## (Intercept)                                 < 2e-16 ***
## poly(Boston[, i], degree = 3, raw = TRUE)1  < 2e-16 ***
## poly(Boston[, i], degree = 3, raw = TRUE)2 4.94e-12 ***
## poly(Boston[, i], degree = 3, raw = TRUE)3 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

```
## rad
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 -0.605545   2.050108  -0.295
## poly(Boston[, i], degree = 3, raw = TRUE)1  0.512736   1.043597   0.491
## poly(Boston[, i], degree = 3, raw = TRUE)2 -0.075177   0.148543  -0.506
## poly(Boston[, i], degree = 3, raw = TRUE)3  0.003209   0.004564   0.703
##                                             Pr(>|t|)
## (Intercept)                                    0.768
## poly(Boston[, i], degree = 3, raw = TRUE)1    0.623
## poly(Boston[, i], degree = 3, raw = TRUE)2    0.613
## poly(Boston[, i], degree = 3, raw = TRUE)3    0.482
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

dis
crim
0
20
40
60
80
2
4
6
8
10
12

rad
crim
0
20
40
60
80
5
10
15
20

```
## tax
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 1.918e+01  1.180e+01   1.626
## poly(Boston[, i], degree = 3, raw = TRUE)1 -1.533e-01  9.568e-02  -1.602
## poly(Boston[, i], degree = 3, raw = TRUE)2  3.608e-04  2.425e-04   1.488
## poly(Boston[, i], degree = 3, raw = TRUE)3 -2.204e-07  1.889e-07  -1.167
##                                            Pr(>|t|)
## (Intercept)                                   0.105
## poly(Boston[, i], degree = 3, raw = TRUE)1    0.110
## poly(Boston[, i], degree = 3, raw = TRUE)2    0.137
## poly(Boston[, i], degree = 3, raw = TRUE)3    0.244
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

```
## ptratio
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 477.18405  156.79498   3.043
## poly(Boston[, i], degree = 3, raw = TRUE)1 -82.36054   27.64394  -2.979
## poly(Boston[, i], degree = 3, raw = TRUE)2   4.63535    1.60832   2.882
## poly(Boston[, i], degree = 3, raw = TRUE)3  -0.08476    0.03090  -2.743
##                                            Pr(>|t|)
## (Intercept)                                 0.00246 **
## poly(Boston[, i], degree = 3, raw = TRUE)1  0.00303 **
## poly(Boston[, i], degree = 3, raw = TRUE)2  0.00412 **
## poly(Boston[, i], degree = 3, raw = TRUE)3  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

tax
200
300
400
500
600
700
0
20
40
60
80
crim

ptratio
14
16
18
20
22
0
20
40
60
80
crim

```
## black
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 1.826e+01  2.305e+00   7.924
## poly(Boston[, i], degree = 3, raw = TRUE)1 -8.356e-02  5.633e-02  -1.483
## poly(Boston[, i], degree = 3, raw = TRUE)2  2.137e-04  2.984e-04   0.716
## poly(Boston[, i], degree = 3, raw = TRUE)3 -2.652e-07  4.364e-07  -0.608
##                                            Pr(>|t|)
## (Intercept)                                 1.5e-14 ***
## poly(Boston[, i], degree = 3, raw = TRUE)1    0.139
## poly(Boston[, i], degree = 3, raw = TRUE)2    0.474
## poly(Boston[, i], degree = 3, raw = TRUE)3    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

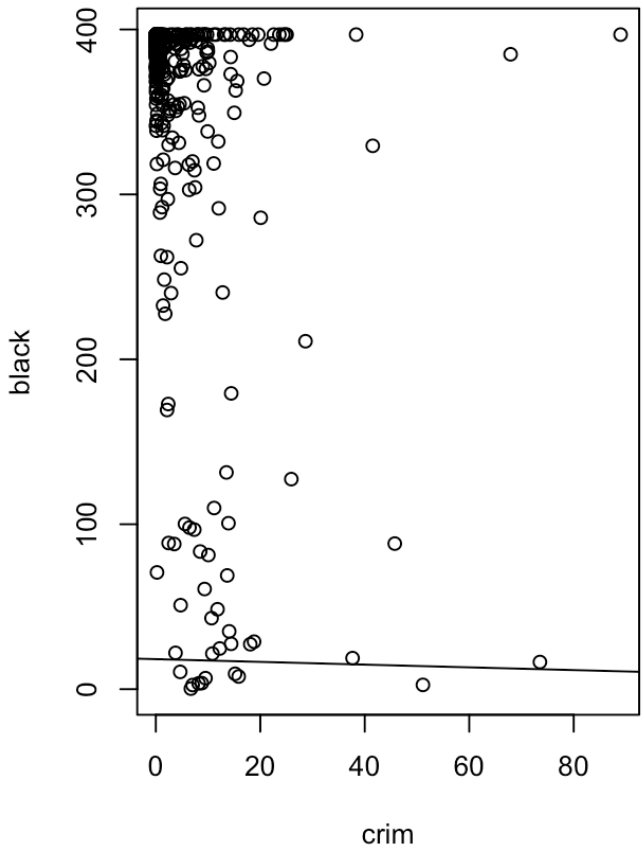
```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

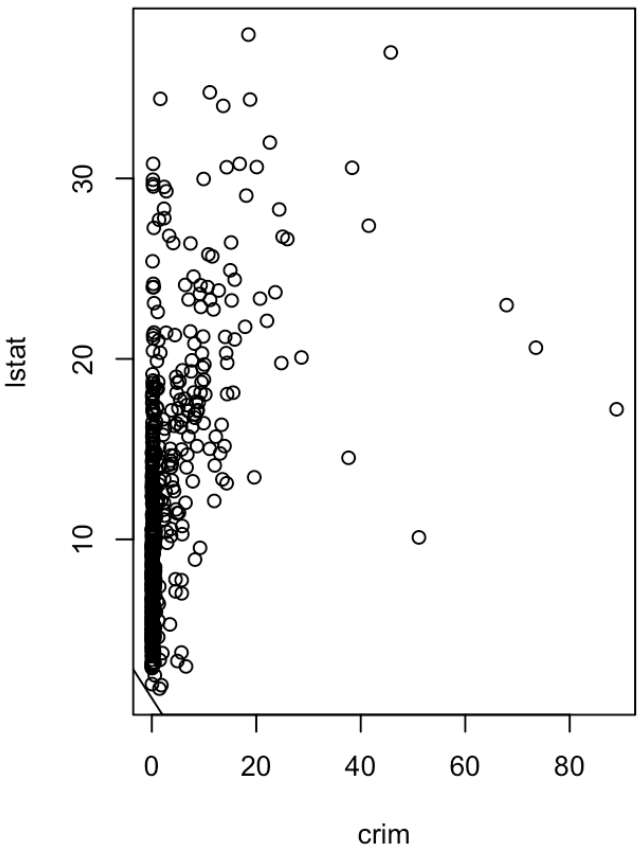
```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```

```
## lstat
## ==================
## Call:
## lm(formula = crim ~ poly(Boston[, i], degree = 3, raw = TRUE),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 1.2009656  2.0286452   0.592
## poly(Boston[, i], degree = 3, raw = TRUE)1 -0.4490656  0.4648911  -0.966
## poly(Boston[, i], degree = 3, raw = TRUE)2  0.0557794  0.0301156   1.852
## poly(Boston[, i], degree = 3, raw = TRUE)3 -0.0008574  0.0005652  -1.517
##                                            Pr(>|t|)
## (Intercept)                                  0.5541
## poly(Boston[, i], degree = 3, raw = TRUE)1   0.3345
## poly(Boston[, i], degree = 3, raw = TRUE)2   0.0646 .
## poly(Boston[, i], degree = 3, raw = TRUE)3   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
## Warning in abline(fit): only using the first two of 4 regression
## coefficients
```

```
## Warning: 'newdata' had 81 rows but variables found have 506 rows
```


black
0
100
200
300
400
0
20
40
60
80
crim


lstat
10
20
30
0
20
40
60
80
crim

Problem 8

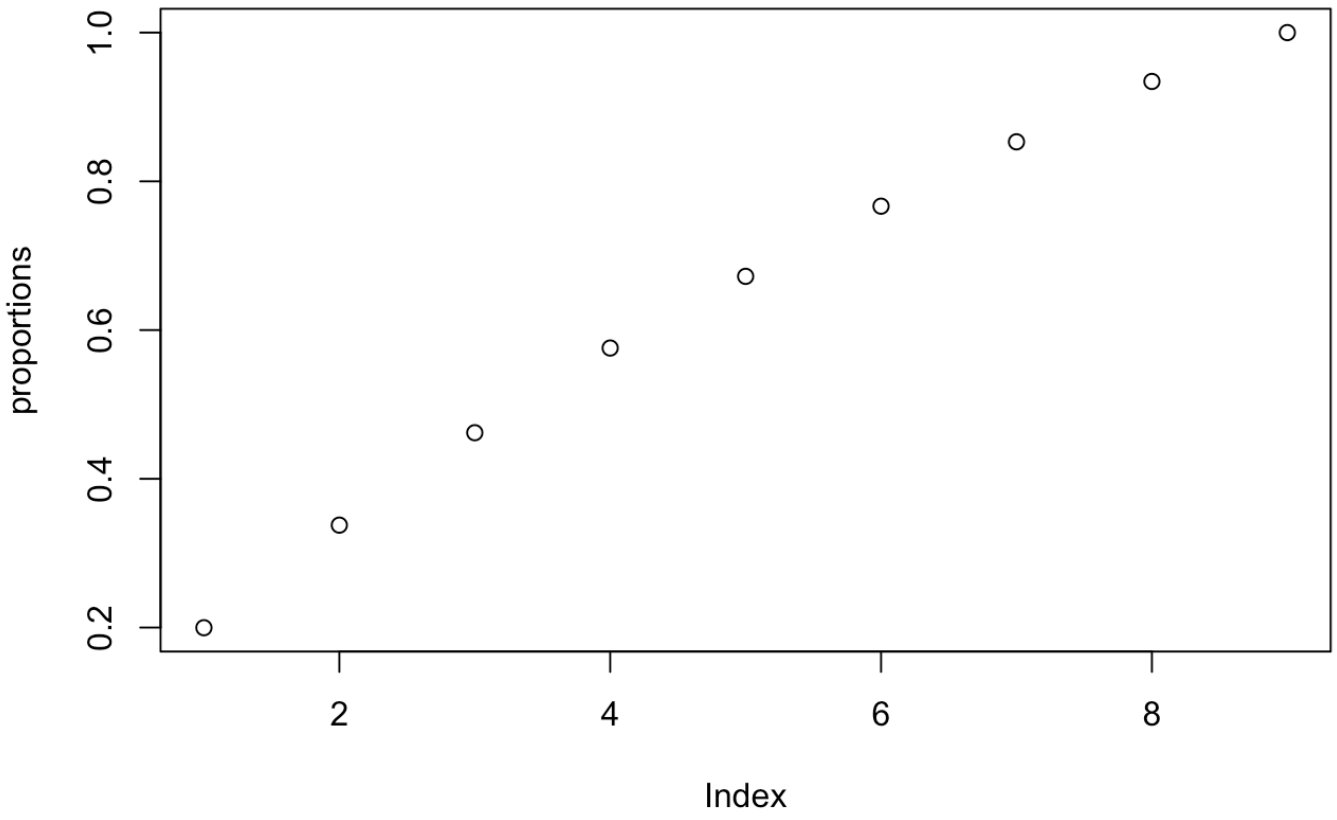
Part 1

```
training_features <- read.csv('training_features.csv')

# Replace data with a new data frame with all NAs imputed with column medians
training_features <- as.data.frame(lapply(training_features, function(x) {x[is.n
a(x)] <- median(x, na.rm=TRUE); x}))

# Exclude subject.id 525450 from the analysis
training_features <- training_features[training_features$subject.id != 525450,]

training_features <- training_features[,c("q1_speech.slope", "q2_salivation.slop
e", "q3_swallowing.slope", "q4_handwriting.slope", "q5a_cutting_without_gastrostom
y.slope", "q6_dressing_and_hygiene.slope", "q7_turning_in_bed.slope", "q8_walkin
g.slope", "q9_climbing_stairs.slope")]

pca = prcomp(training_features, scale=TRUE)
```

Part 2

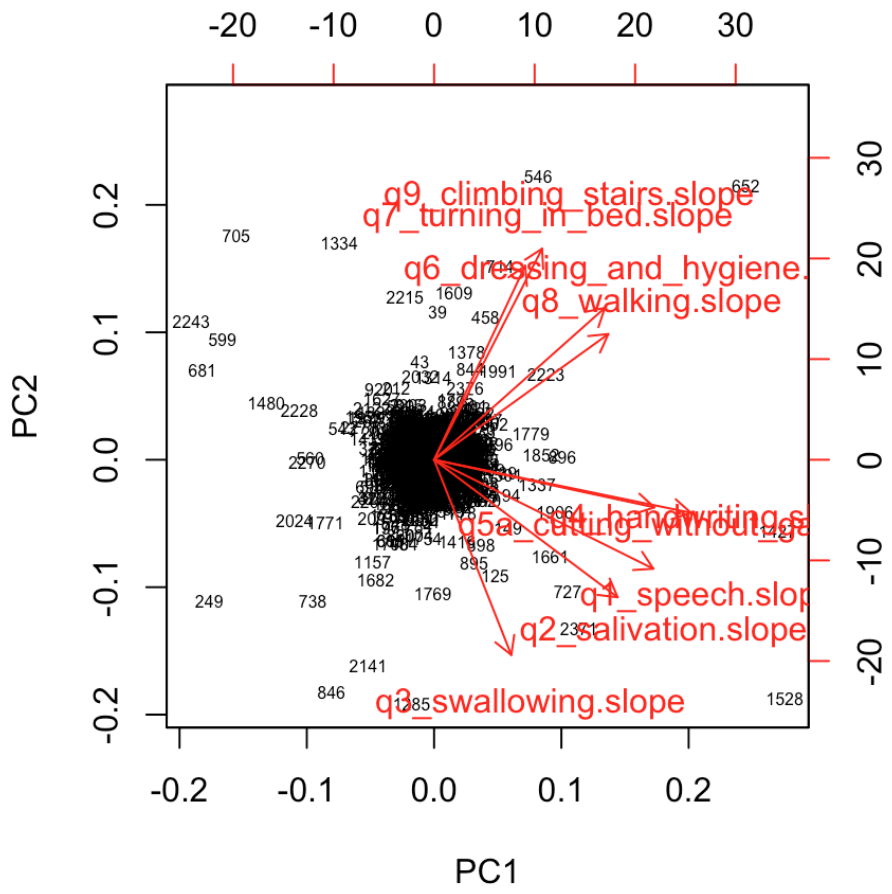
No, the top 2 principal components do not capture most of the variance.

```
vars <- apply(pca$x, 2, var)
props <- vars / sum(vars)
proportions = cumsum(props)
plot(proportions)
```



Part 3

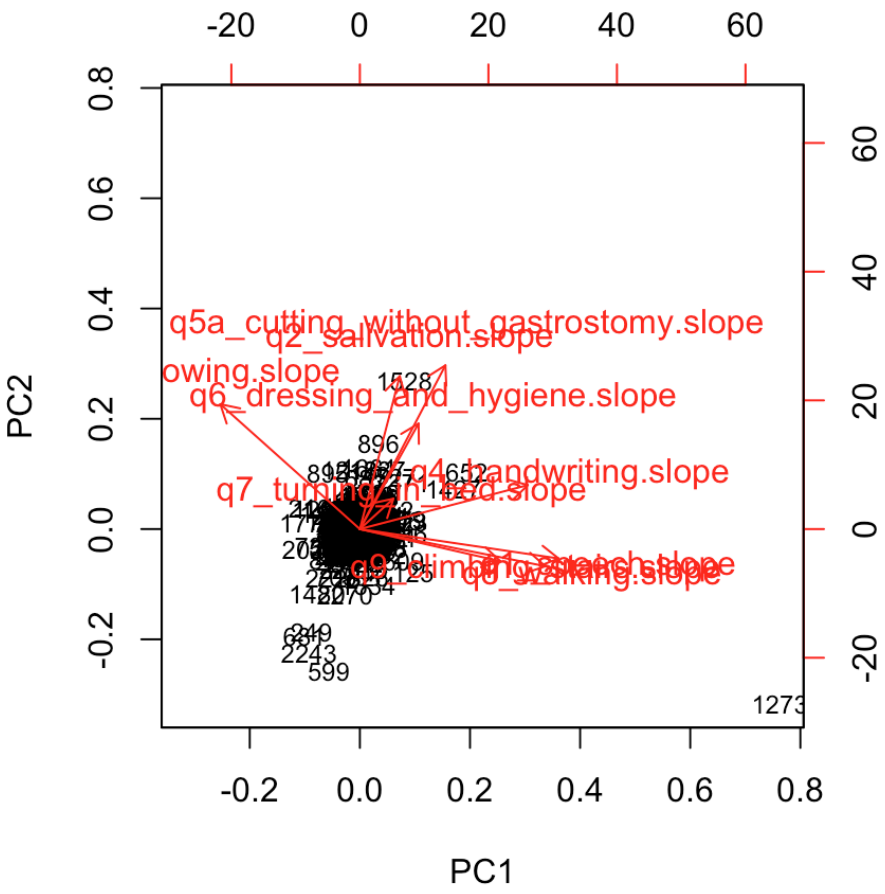
```
biplot(pca, scale=1, cex=c(1/2, 1))
```



Part 4

Patient #525450 is a major outlier from the rest of the data. It is so different that it completely changes the direction of the principal component!

```
training_features <- read.csv('training_features.csv')

# Replace data with a new data frame with all NAs imputed with column medians
training_features <- as.data.frame(lapply(training_features, function(x) {x[is.n
a(x)] <- median(x, na.rm=TRUE); x}))

training_features <- training_features[,c("q1_speech.slope", "q2_salivation.slop
e", "q3_swallowing.slope", "q4_handwriting.slope", "q5a_cutting_without_gastrostom
y.slope", "q6_dressing_and_hygiene.slope", "q7_turning_in_bed.slope", "q8_walkin
g.slope", "q9_climbing_stairs.slope")]

pca = prcomp(training_features, scale=TRUE)
biplot(pca, scale=1, cex=c(3/4, 1))
```


-20
0
20
40
60
0.8
0.6
0.4
0.2
0.0
-0.2
PC2
60
40
20
0
-20
q5a_cutting_without_gastrostomy.slope
q2_salivation.slope
owing.slope
q6_dressing_and_hygiene.slope
q4_handwriting.slope
q7_turning_in_bed.slope
q8_walking.slope
1528
896
652
1273
599
-0.2
0.0
0.2
0.4
0.6
0.8
PC1