ISLR Chapter 3 Exercises

Ryan Heslin

2/16/21

Conceptual

Linear regression really uses linear combination of covariance matrix to project - which is why perfect fit is achievable only with perfect correlation.

1.

The null hypothesis for a regression coefficient is that the predictor has no linear relationship with he response. In this table, TV and radio ads have such a relationship, but newspaper adds do not becasue the coefficient's confidence interval contains 0. β_0 is also significant, meaning the average effect of *omitted* variables is significant as well (since $E(Y) = 0$)

##2.

KNN classification assigns each observation to the most common class among the K nearest points. the. KNN regression *predicts* for each observation the mean of the K closest observations. ## 3.

The model here is

$$Y = 50 + 20X_1 + .07X_2 + 35X_3 + .01X_1X_2 - 10X_1X_3$$

a. i., ii. It depends; see below.

iii. True. Females outearn males only if their GPA is below 3.5:

$$35 > 10X_2X_2 < 3.5$$

So a male with GPA above 3.5 beats the value of the female indicator variable.

iv. False by the above

b. $50 + 20(4.0) + .07(110) + 35(1) + .01(4)(110)) - 10(4)(1) = 137.1$ That comes to $137,100.

c. False. Since this is an interaction coefficient, it refers to the product of the units of GPA and IQ, not the individual scales.

4.

Even if the relationship is linear, the cubic model will have lower training RSS because adding information to the model will always improve it, even if it is irrelevant to the true relationship. b. Test RSS will most likely increase because the model is biased; it does not reflect the true form of the function.

c, d. The more flexible cubic fit would outperform a truly nonlinear $f(x)$ on both test and training sets, though the improvement might be marginal depending on the form of the function.

Applied

11.

This problem concerns no-intercept regression.

```
set.seed(1)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
x <- rnorm(100)
y <- 2*x + rnorm(100)

mod <- lm(y~x-1)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

We see the model estimates the correct coefficient with high probability.

  b.

```
mod2 <- lm(x ~y-1)
summary(mod2)
```

```
##
## Call:
## lm(formula = x ~ y - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

c. The coefficient is .4, not the correct .5 (as $Y = 2X + \epsilon$. The models have the same t values and R^2, though standard error is different (as the coefficients differ)

The numerator expresses β, the denominator its standard error.

d. This was a fun proof. I'll simplify the sum notation. First clean up the fraction:

$$\frac{\sum x_i y_i \sqrt{(n-1)\sum x_i^2}}{\sum x_i^2 \sqrt{\sum (y_i - x_i\beta)^2}}$$

then cancel the $\sum x_i^2$:

$$\frac{\sum x_i y_i \sqrt{(n-1)}}{\sqrt{\sum x_i^2 \sum (y_i - x_i\beta)^2}}$$

then complete the square:

$$\frac{\sum x_i y_i \sqrt{(n-1)}}{\sqrt{\sum x_i^2 (\sum y_i^2 + \sum x_i \beta^2 - \sum 2x_i y_i \beta)}}$$

and bring out the constants with respect to the sums. Substituting the definition of beta makes everything cancel nicely, leaving:

$$\frac{\sum x_i y_i \sqrt{(n-1)}}{\sqrt{\sum x_i^2 \sum y_i^2 - \sum x_i y_i}}$$

e. From the equation, it is obvious by the commutative property of multiplication that substituting x for y gives the same result.

f. From the model summaries above, the t-values are identical.

12.

Here we are asked to continue no-intercept regression.

a. For simple no-intercept regression,

$$\beta = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} xi^2}$$

The denominator is the same whether we regress y on x or vice versa, and by wrapping the values we see that:

$$\beta = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} yi^2}$$

Obviously these two are only equal if x and y have equal sums of squares.

b. In almost every real case this property will not hold. Here randomness prevents the coefficients from quite matching

```
y <- rnorm(100)
x <- rnorm(100)

lm(y~x-1)
```

```
##
## Call:
## lm(formula = y ~ x - 1)
##
## Coefficients:
##      x
## 0.1168
```

```
lm(x~y-1)
```

```
##
## Call:
## lm(formula = x ~ y - 1)
##
## Coefficients:
##      y
## 0.1076
```

c. But if the vectors were opposite-signed…

```
y <- rnorm(100)
x <- -y
sum((x-mean(x))^2) == sum((y-mean(y))^2)
```

```
## [1] TRUE
```

```r
lm(x~y-1)
```

```
##
## Call:
## lm(formula = x ~ y - 1)
##
## Coefficients:
##  y
## -1
```

```r
lm(y~x-1)
```

```
##
## Call:
## lm(formula = y ~ x - 1)
##
## Coefficients:
##  x
## -1
```

13.

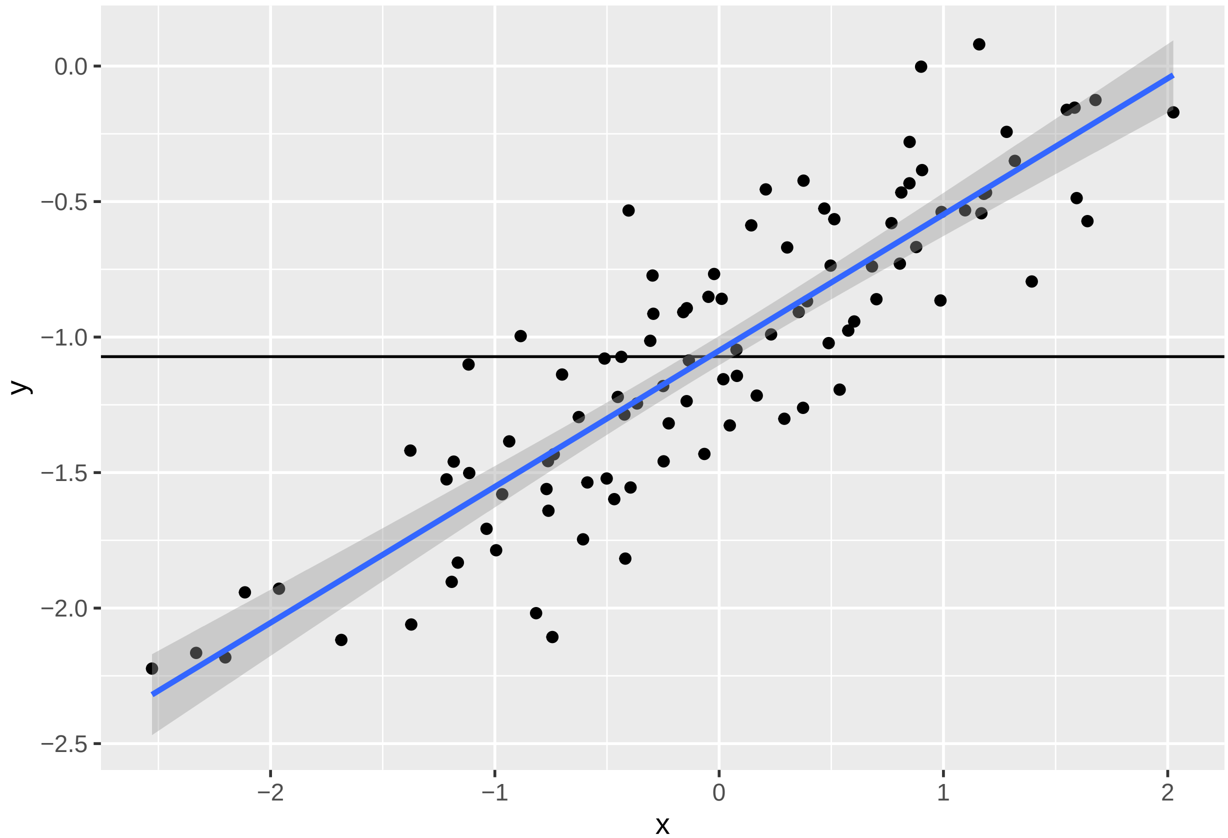
a.

```r
x <- rnorm(100)
eps <- rnorm(100, sd = .25)
y <- -1 +.5*x +eps
```

In this implied model, $\beta_0 = -1$ and $\beta_1 = .5$.

```r
tibble(x, y) %>%
  ggplot(aes(x,y))+
  geom_point() +
  geom_hline(yintercept = mean(y)) +
  geom_smooth(method = "lm")
```

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



e. Fitting a model, we see that the estimates are very close

```
mod <- lm(y~x)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68355 -0.20026 -0.01447  0.18656  0.71960
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.04980    0.02719  -38.61   <2e-16 ***
## x            0.50215    0.02825   17.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2716 on 98 degrees of freedom
## Multiple R-squared:  0.7633, Adjusted R-squared:  0.7608
## F-statistic:   316 on 1 and 98 DF,  p-value: < 2.2e-16
```

g. Now we add a polynomial model:

```r
mod_poly <- lm(y~poly(x, degree = 2))
summary(mod_poly)
```

```
##
## Call:
## lm(formula = y ~ poly(x, degree = 2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68266 -0.20015 -0.01476  0.18780  0.72092
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.07215    0.02730 -39.278   <2e-16 ***
## poly(x, degree = 2)1  4.82733    0.27297  17.685   <2e-16 ***
## poly(x, degree = 2)2  0.01819    0.27297   0.067    0.947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.273 on 97 degrees of freedom
## Multiple R-squared:  0.7633, Adjusted R-squared:  0.7584
## F-statistic: 156.4 on 2 and 97 DF,  p-value: < 2.2e-16
```

The coefficients are completely different, but the fit has hardly changed, since the true relationship is linear.

```r
x <- rnorm(100)
eps <- rnorm(100, sd = .1)
y <- -1 +.5*x +eps
params <- map(c(.1, .15, .4), ~tibble(x = rnorm(100), y = -1 + x + rnorm(100, sd = .x))) %>%
  transpose() %>%
  as_tibble()

mods <- pmap(params, ~list(linear = lm(.y ~.x),
                           poly = lm(.y~poly(.x,2))))

frame <- map_depth(mods, 2, broom::glance) %>% map(bind_rows, .id = "Type") %>% bind_rows(.id = "Subset
```

As expected, R squared declines as the variance increases,a nd confidence intervals greatly widen.

```r
map_depth(mods, 2, confint)
```

```
## [[1]]
## [[1]]$linear
##                  2.5 %     97.5 %
## (Intercept) -1.0215725 -0.9805154
## .x           0.9834584  1.0225780
##
## [[1]]$poly
##                    2.5 %     97.5 %
## (Intercept)  -1.01745878 -0.9764746
## poly(.x, 2)1 10.32197364 10.7318153
## poly(.x, 2)2 -0.08393056  0.3259111
```

```
##
##
## [[2]]
## [[2]]$linear
##                   2.5 %     97.5 %
## (Intercept) -0.9971699 -0.9319223
## .x           0.9610220  1.0267299
##
## [[2]]$poly
##                    2.5 %    97.5 %
## (Intercept)  -1.1478839 -1.083205
## poly(.x, 2)1  9.4295431 10.076337
## poly(.x, 2)2 -0.4290106  0.217783
##
##
## [[3]]
## [[3]]$linear
##                   2.5 %     97.5 %
## (Intercept) -1.0602035 -0.8917984
## .x           0.8625449  1.0547984
##
## [[3]]$poly
##                   2.5 %    97.5 %
## (Intercept)  -1.224846 -1.059136
## poly(.x, 2)1  7.403279  9.060384
## poly(.x, 2)2 -1.050862  0.606243
```

14.

Here we simulate the model

$$Y = 2 + 2X_1 + 0.3X_2 + \epsilon$$

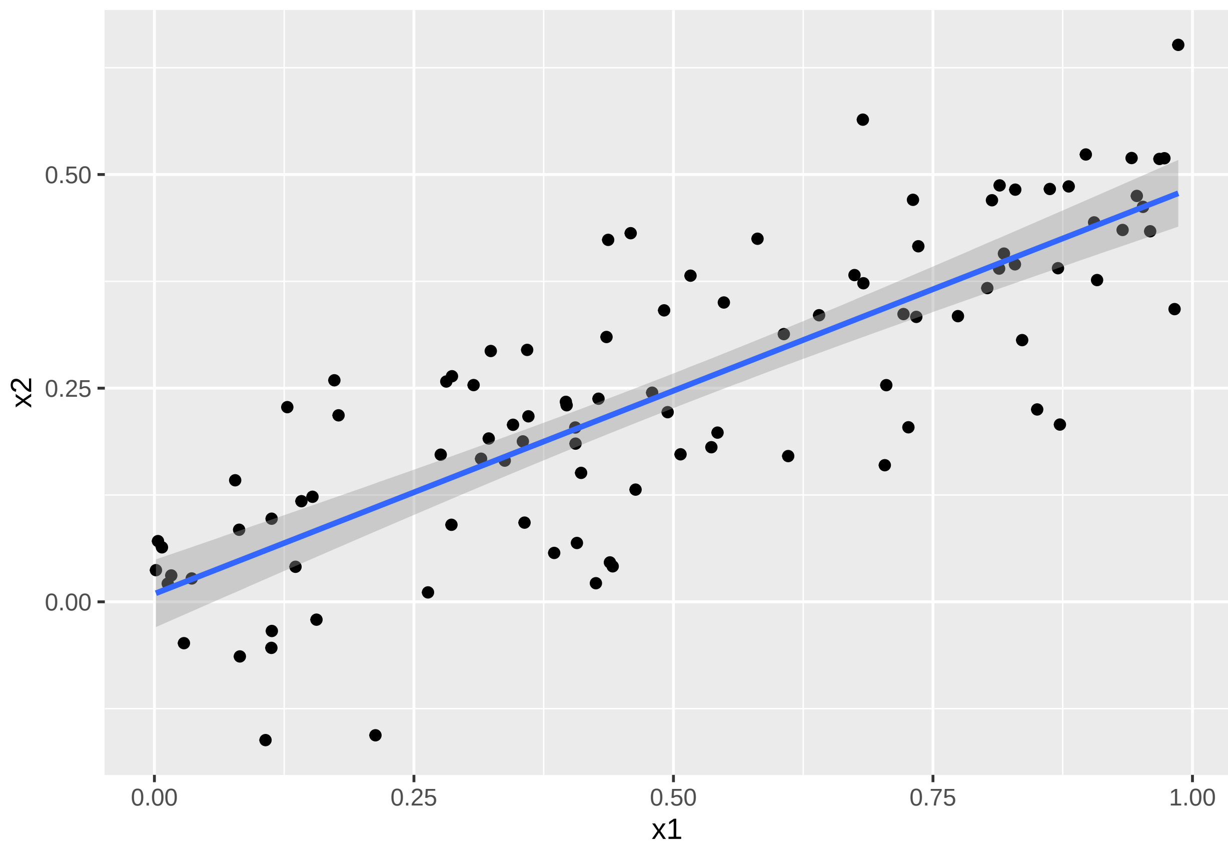
```
dat <- tibble(x1=runif(100), x2= .5 *x1 + (rnorm(100)/10), y = 2 +2 *x1 +.3*x2 + rnorm(100) )
```

b.

X_1 and X_2 are strongly correlated, since X_2 is just a multiple of X_1 with noise added. For that reason, X_1 has more correlation with Y.

```
dat %>% ggplot(aes(x1, x2)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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




```
cor(dat)
```

```
##           x1        x2         y
## x1 1.0000000 0.8121399 0.4412833
## x2 0.8121399 1.0000000 0.3858594
## y  0.4412833 0.3858594 1.0000000
```

Fitting multiple models, we see the first reduces X_1 but boosts X_2, , the second gets X_1 about right but overstates the intercept, and the second gets the intercept about right but grossly overstates x_2. This is consistent with both variables being correlated with each other but only X_1 strongly correlated with Y.

```
mods <- list(both = lm(data = dat, y ~.),
             x1 <- lm(data = dat, y~ x1),
             x2 = lm(data = dat, y~x2))

map(mods, summary)
```

```
## $both
##
## Call:
## lm(formula = y ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.40977 -0.75052 -0.06067  0.92880  3.05732
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0882     0.2236   9.340 3.58e-15 ***
## x1            1.5904     0.6601   2.409   0.0179 *
## x2            0.5839     1.1281   0.518   0.6060
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.141 on 97 degrees of freedom
## Multiple R-squared:  0.1969, Adjusted R-squared:  0.1804
## F-statistic: 11.89 on 2 and 97 DF,  p-value: 2.399e-05
##
##
## [[2]]
##
## Call:
## lm(formula = y ~ x1, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.41466 -0.74713 -0.09575  0.96716  3.11331
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0936     0.2225   9.410 2.31e-15 ***
## x1            1.8678     0.3837   4.868 4.32e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 98 degrees of freedom
## Multiple R-squared:  0.1947, Adjusted R-squared:  0.1865
## F-statistic:  23.7 on 1 and 98 DF,  p-value: 4.317e-06
##
##
## $x2
##
## Call:
## lm(formula = y ~ x2, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74142 -0.83249 -0.00239  0.98551  2.61790
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3374     0.2030   11.51  < 2e-16 ***
## x2            2.7913     0.6742    4.14 7.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.169 on 98 degrees of freedom
## Multiple R-squared:  0.1489, Adjusted R-squared:  0.1402
```

```
## F-statistic: 17.14 on 1 and 98 DF,  p-value: 7.347e-05
```

g. This observation was tragically mismeasured. The X_1 point has higher leverage because X_2 comes partially from a normal rather than a uniform distribution, making extreme values less likely. We refit the models and see that X_2 now has a negative coefficient in the two-variable model, while the coefficients for the two single-variable models are inflated. The high-leverage point increases TSS and thus the magnitude of the betas needed to project the data.

```r
dat <- add_row(dat, x1 =.1, x2 = .8, y = 6)
cor(dat)
```

```
##           x1        x2         y
## x1 1.0000000 0.7269879 0.3952211
## x2 0.7269879 1.0000000 0.4274806
## y  0.3952211 0.4274806 1.0000000
```

```r
mods <- map(mods, update)

map(mods, summary)
```

```
## $both
##
## Call:
## lm(formula = y ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48901 -0.83019  0.00398  0.89376  2.85041
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1423     0.2271   9.431 2.08e-15 ***
## x1            0.7721     0.5679   1.359   0.1771
## x2            2.1058     0.9333   2.256   0.0263 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.166 on 98 degrees of freedom
## Multiple R-squared:  0.1979, Adjusted R-squared:  0.1815
## F-statistic: 12.09 on 2 and 98 DF,  p-value: 2.033e-05
##
##
## [[2]]
##
## Call:
## lm(formula = y ~ x1, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3966 -0.7579 -0.0992  0.9379  3.6180
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2116     0.2297   9.630 7.02e-16 ***
## x1            1.7036     0.3980   4.281 4.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.19 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: 4.308e-05
##
##
## $x2
##
## Call:
## lm(formula = y ~ x2, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7499 -0.8322  0.0155  0.9861  2.6297
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2919     0.1995  11.486  < 2e-16 ***
## x2            3.0281     0.6436   4.705 8.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.171 on 99 degrees of freedom
## Multiple R-squared:  0.1827, Adjusted R-squared:  0.1745
## F-statistic: 22.14 on 1 and 99 DF,  p-value: 8.25e-06
```

15.

We are asked to predict crime in Boston. Some more quasiquotation silliness.

a. It turns out only crim is a nonsiginficant predictor.

```r
boston <- MASS::Boston

preds <- boston %>% select(-crim) %>%
  names() %>%
  map(as.symbol)

call <- quote(lm(data =boston))

bos_mods <- map2(preds, list(call), function(.x, .y){
  .y$formula <- bquote(crim ~ .(.x))
  eval(.y)
})
```

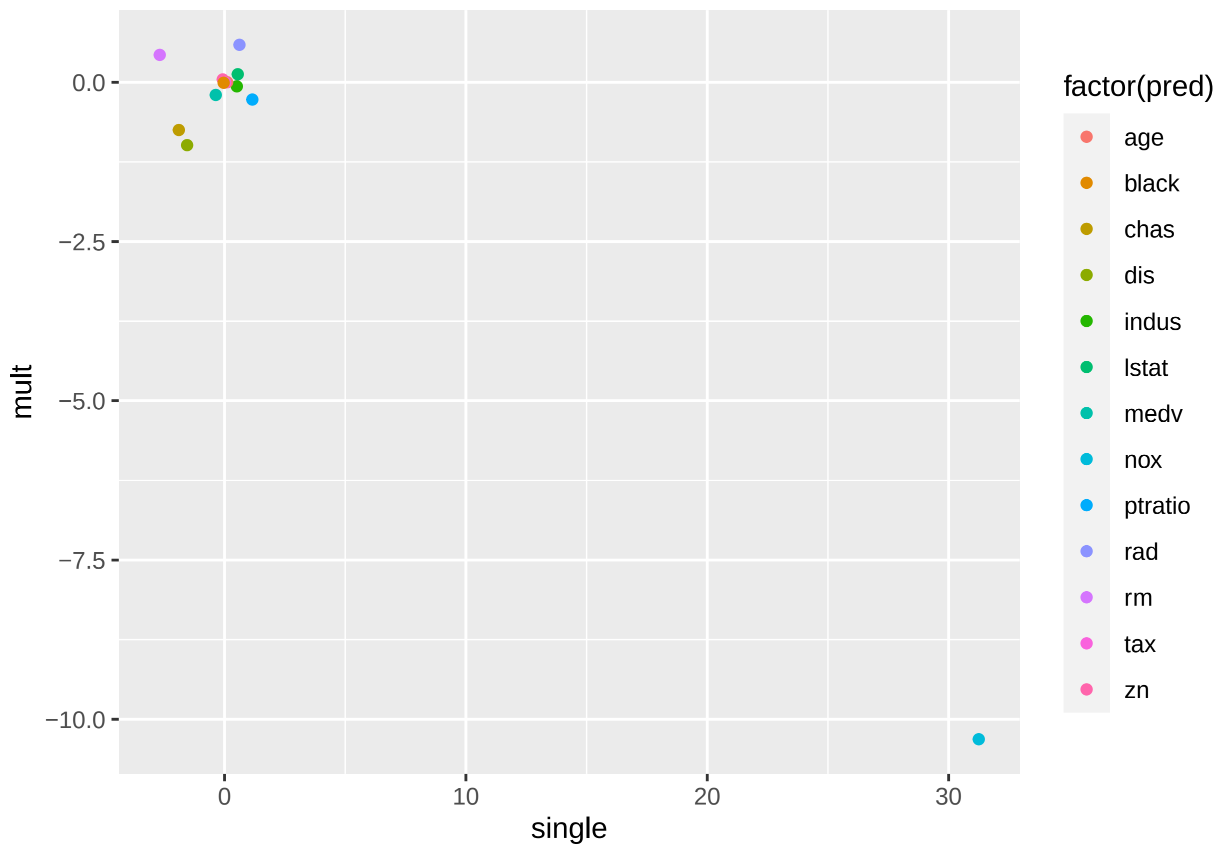
b. But when we fit a model on the whole data, only a few predictors are significant, dis and rad.

```r
mod2 <- lm(data = boston, crim ~ .)
summary(mod2)
```

```
##
## 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
```

c. We see a tight cluster of points with just one coef, zn, clearly differentiated. This is the classic thin ellipse of multicollinearity.

```r
tibble(single = map(bos_mods, broom::tidy) %>%
         map_dbl(~`[`(.x, 2, "estimate", drop = TRUE) %>% unlist()), mult = broom::tidy(mod2)$estimate[-
  ggplot(aes(x =single, y =mult, color = factor(pred))) + geom_point()
```



d.

```
map(names(boston %>% select(-c(chas, crim))), as.symbol) %>%
  map(~bquote(crim ~poly(.(.x), degree = 3))) %>%
  map(~lm(formula = .x, data = boston)) %>%
  map(summary)
```

```
## [[1]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, degree = 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, degree = 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, degree = 3)3 -10.0719     8.3722  -1.203  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
##
##
## [[2]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3.614      0.330  10.950  < 2e-16 ***
## poly(indus, degree = 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, degree = 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, degree = 3)3  -54.130      7.423  -7.292  1.2e-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
##
##
## [[3]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, degree = 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, degree = 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, degree = 3)3 -60.3619     7.2336  -8.345 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
##
##
## [[4]]
##
## Call:
```

```
## lm(formula = .x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, degree = 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, degree = 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, degree = 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## 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
##
##
## [[5]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, degree = 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, degree = 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, degree = 3)3  21.3532     7.8397   2.724  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
##
##
## [[6]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)              3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, degree = 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, degree = 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, degree = 3)3 -42.6219     7.3315  -5.814 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
##
##
## [[7]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, degree = 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, degree = 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, degree = 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## 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
##
##
## [[8]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, degree = 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, degree = 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, degree = 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## 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
##
##
## [[9]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, degree = 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, degree = 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, degree = 3)3  -22.280      8.122  -2.743  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
##
##
## [[10]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.6135     0.3536  10.218   <2e-16 ***
## poly(black, degree = 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, degree = 3)2   5.9264     7.9546   0.745    0.457
## poly(black, degree = 3)3  -4.8346     7.9546  -0.608    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
##
##
## [[11]]
##
## Call:
## lm(formula = .x, data = boston)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, degree = 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, degree = 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, degree = 3)3 -11.5740     7.6294  -1.517   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
##
##
## [[12]]
##
## Call:
## lm(formula = .x, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.614      0.292  12.374  < 2e-16 ***
## poly(medv, degree = 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, degree = 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, degree = 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

We see strong nonlinear relationships for zn, indus, dis, tax, and medv, most of which are measured on very different scales.