Chapter 2 Questions

(7)

(a)

Obs 1: $\sqrt{(0-0)^2+(0-3)^2+(0-0)^2} = 9$

Obs 2: $\sqrt{(0-2)^2+(0-0)^2+(0-0)^2} = 4$

Obs 3: $\sqrt{(0-0)^2+(0-1)^2+(0-3)^2} = 10$

Obs 4: $\sqrt{(0-0)^2+(0-1)^2+(0-2)^2} = 5$

Obs 5: $\sqrt{(0--1)^2+(0-0)^2+(0-1)^2} = 2$

Obs 6: $\sqrt{(0-1)^2+(0-1)^2+(0-1)^2} = 3$

(b)

With K = 1, our prediction is Green because observation 5 is the closest to our new point and it is Green.

(c)

With K = 3, our prediction is Red because 2 of the 3 closest observations are Red.

(d)

To fit a highly non-linear function, we want our model to be very flexible. With a small K, the boundary will be less rigid and will be closer to fitting the non-linear f.

(10)

(a)

```
bos <- Boston
```

```
nrow(bos)
```

```
## [1] 506
```

```
ncol(bos)
```

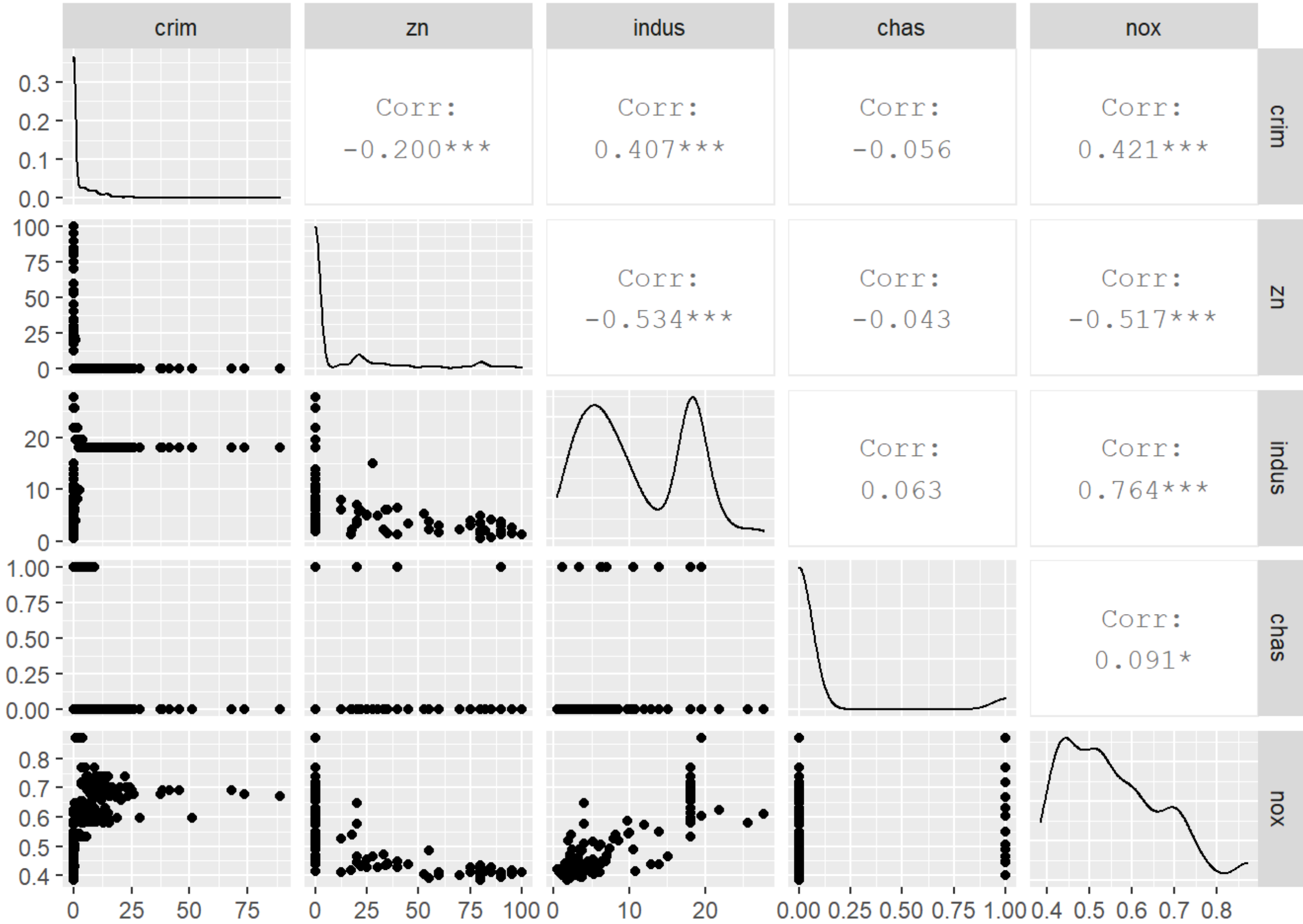
```
## [1] 14
```

14 columns, 506 rows.

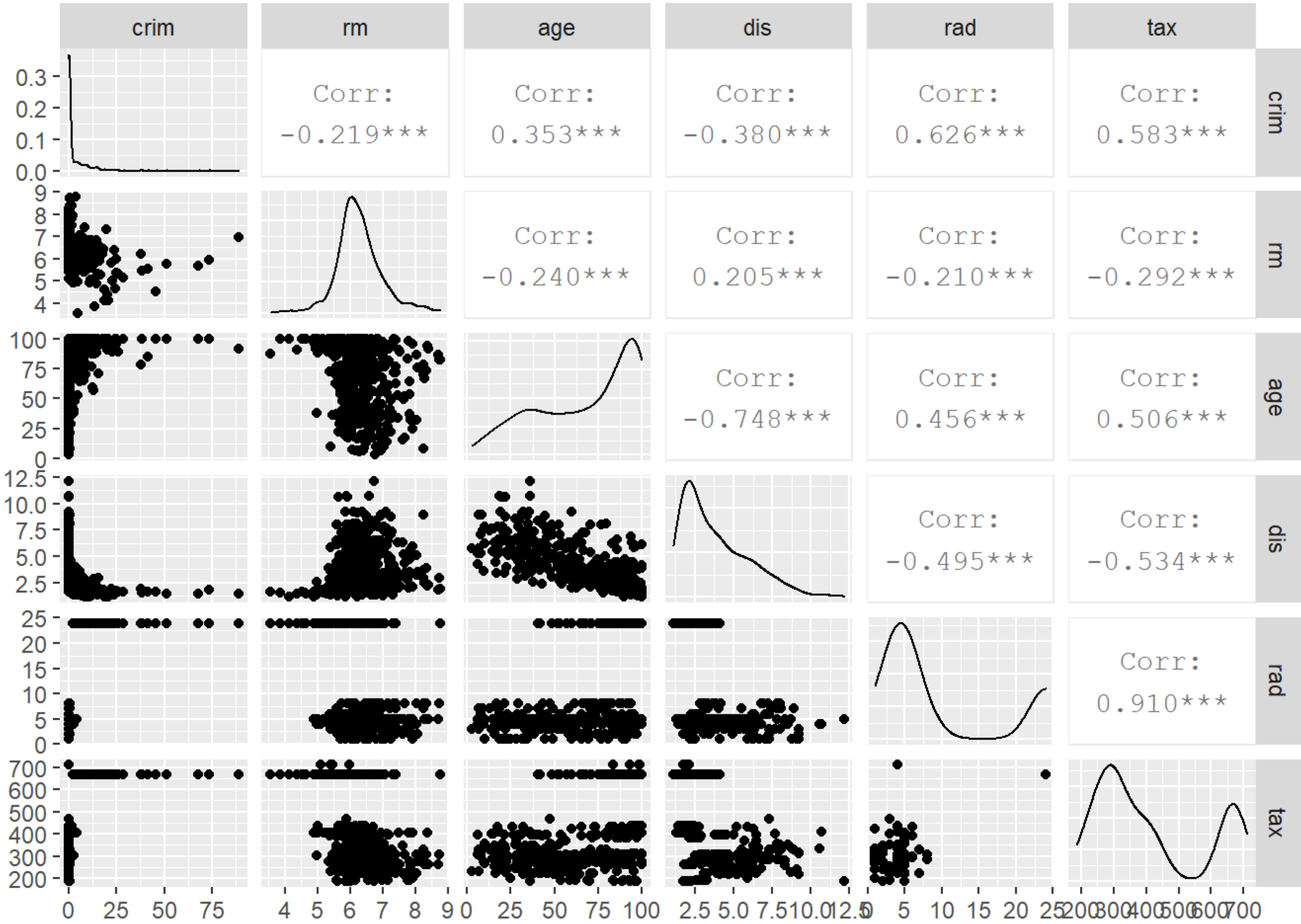
The columns represent various attributes about a house, every row is a different house.

(b)

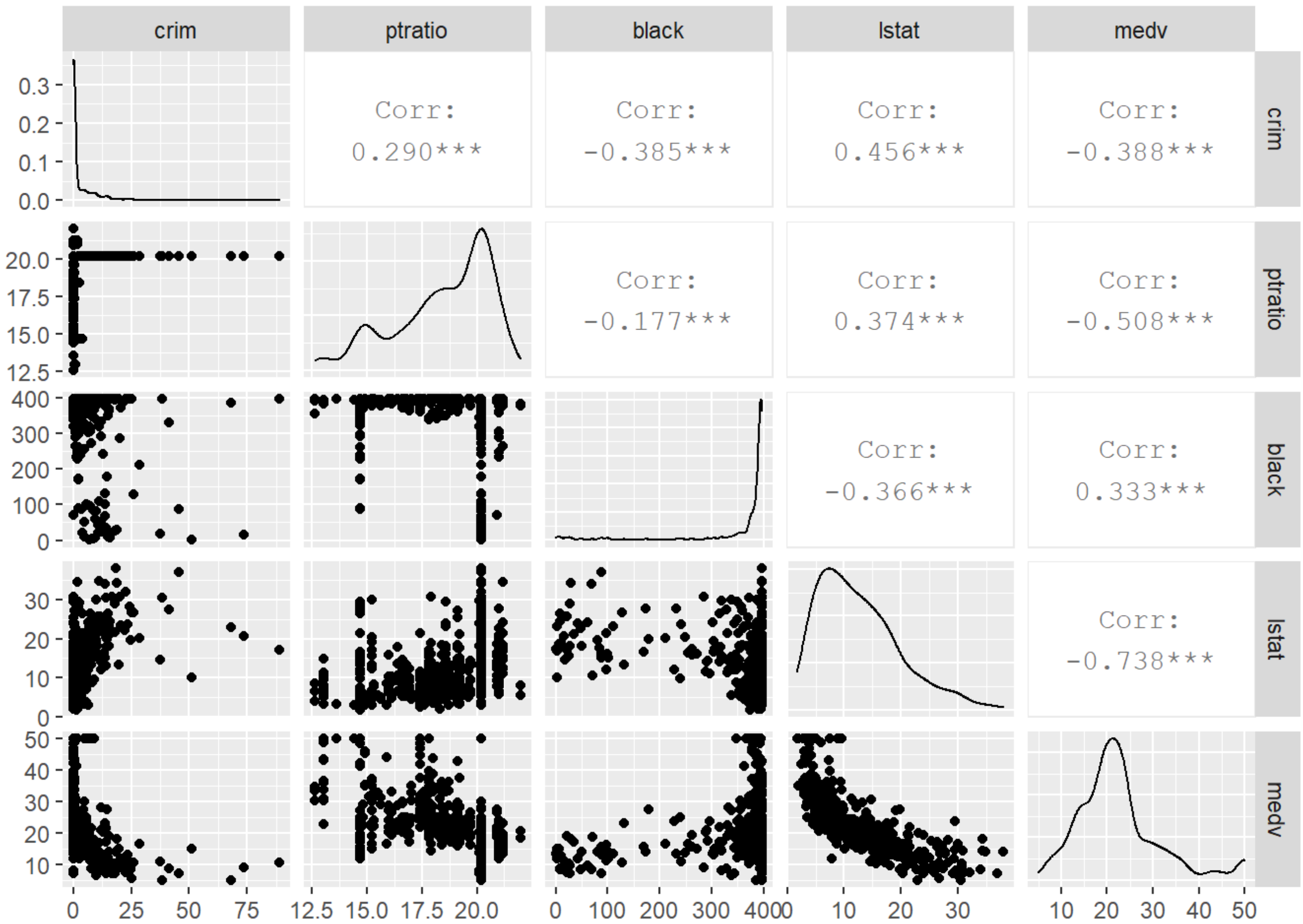
```
ggpairs(bos, columns = 1:5)
```




```
ggpairs(bos, columns = c(1, 6:10))
```




```
ggpairs(bos, columns = c(1, 11:14))
```



Many relationships look random, but some like `medv` vs `lstat` and `crim` vs `age` have strong non-linear relationship.

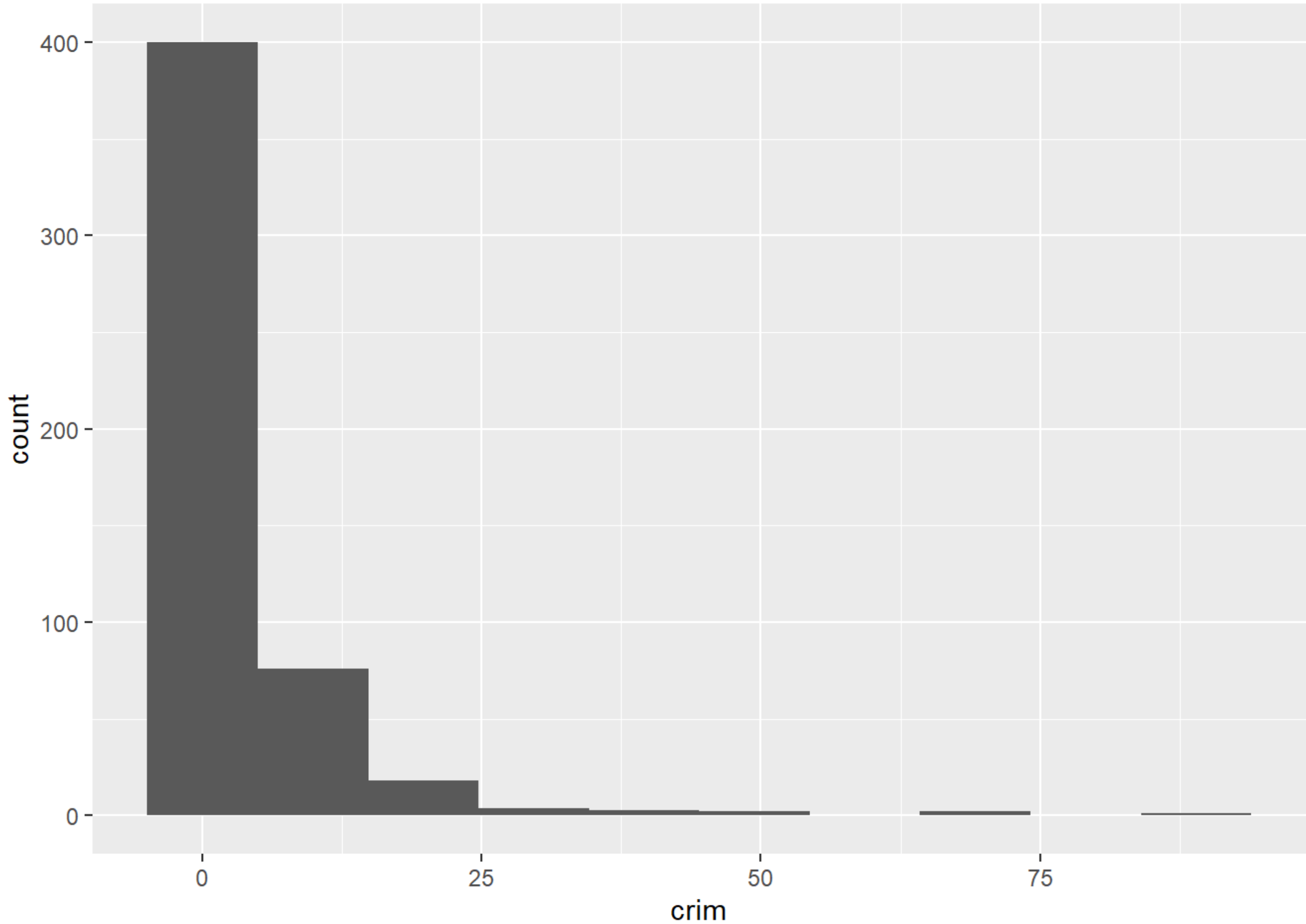
(c)

`rad` and `tax` are both strongly correlated to `crim`. Not having access to radial highways might suggest a more rural property, while higher taxes are robbing people and pushing them towards crime.

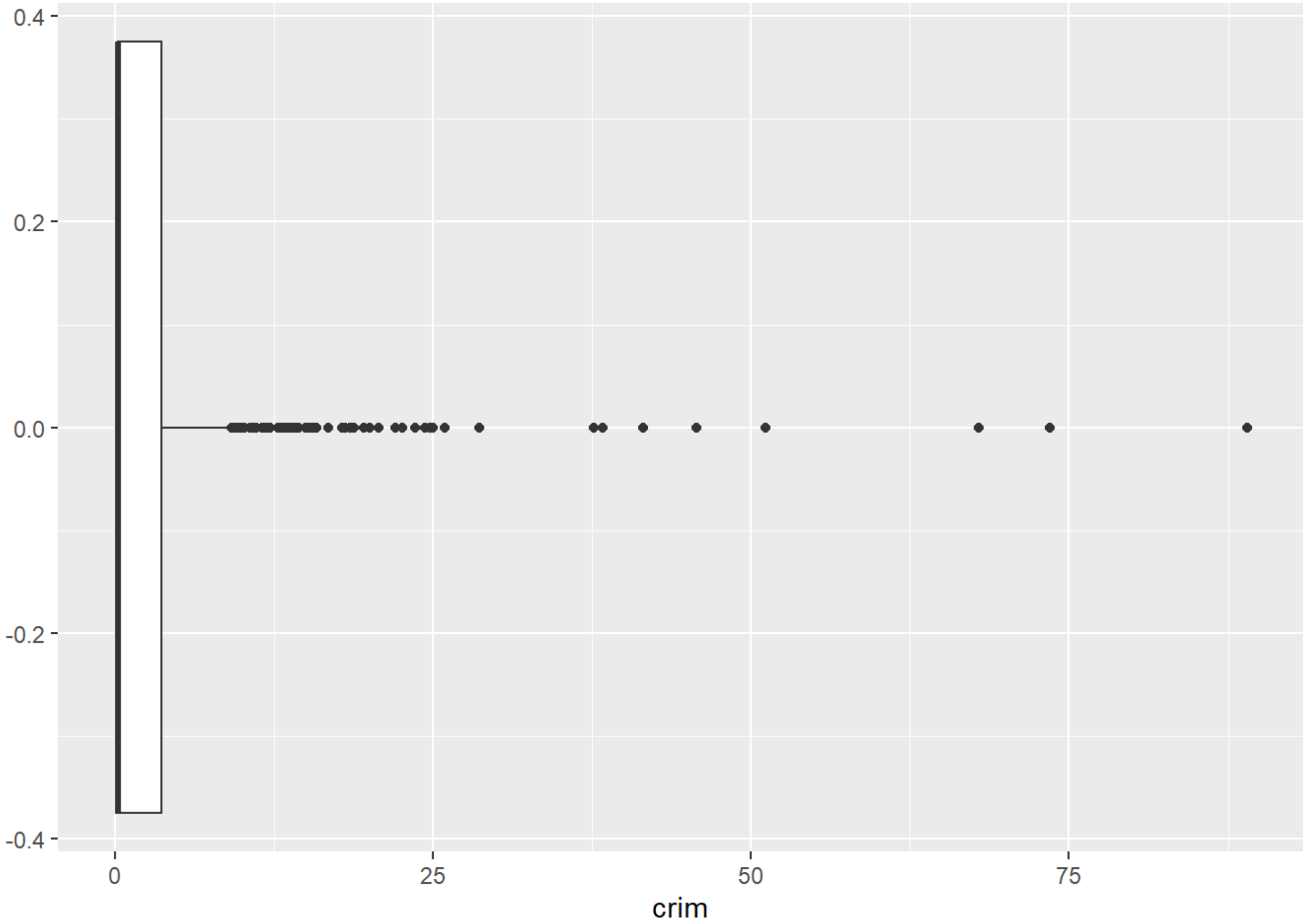
(d)

Crime rate

```r
bos %>%
  ggplot(aes(crim)) +
  geom_histogram(bins = 10)
```



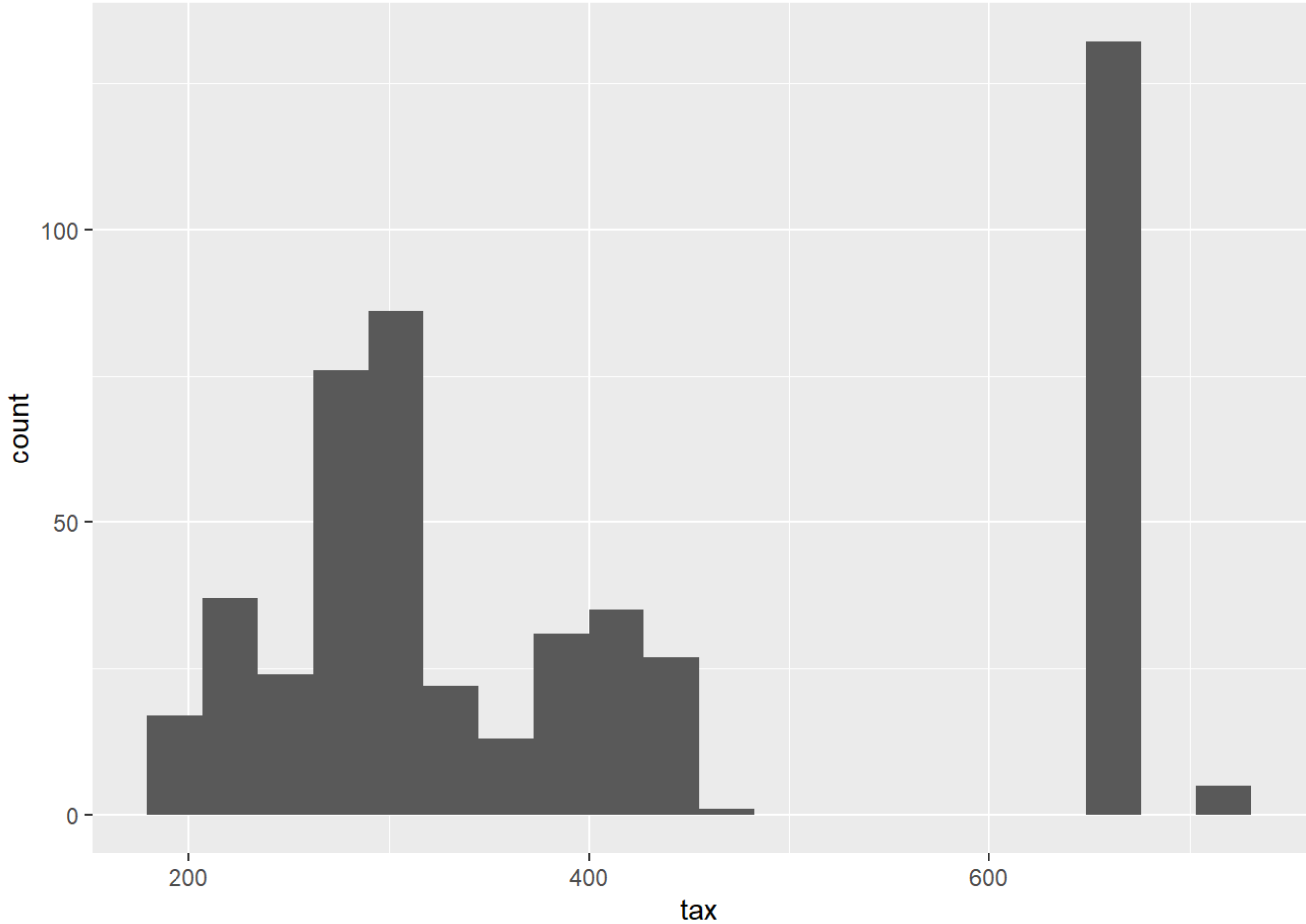

```r
bos %>%
  ggplot(aes(crim)) +
  geom_boxplot()
```



Most crime rates close to 0, but also many outliers at 20 and 25%+

Tax rate

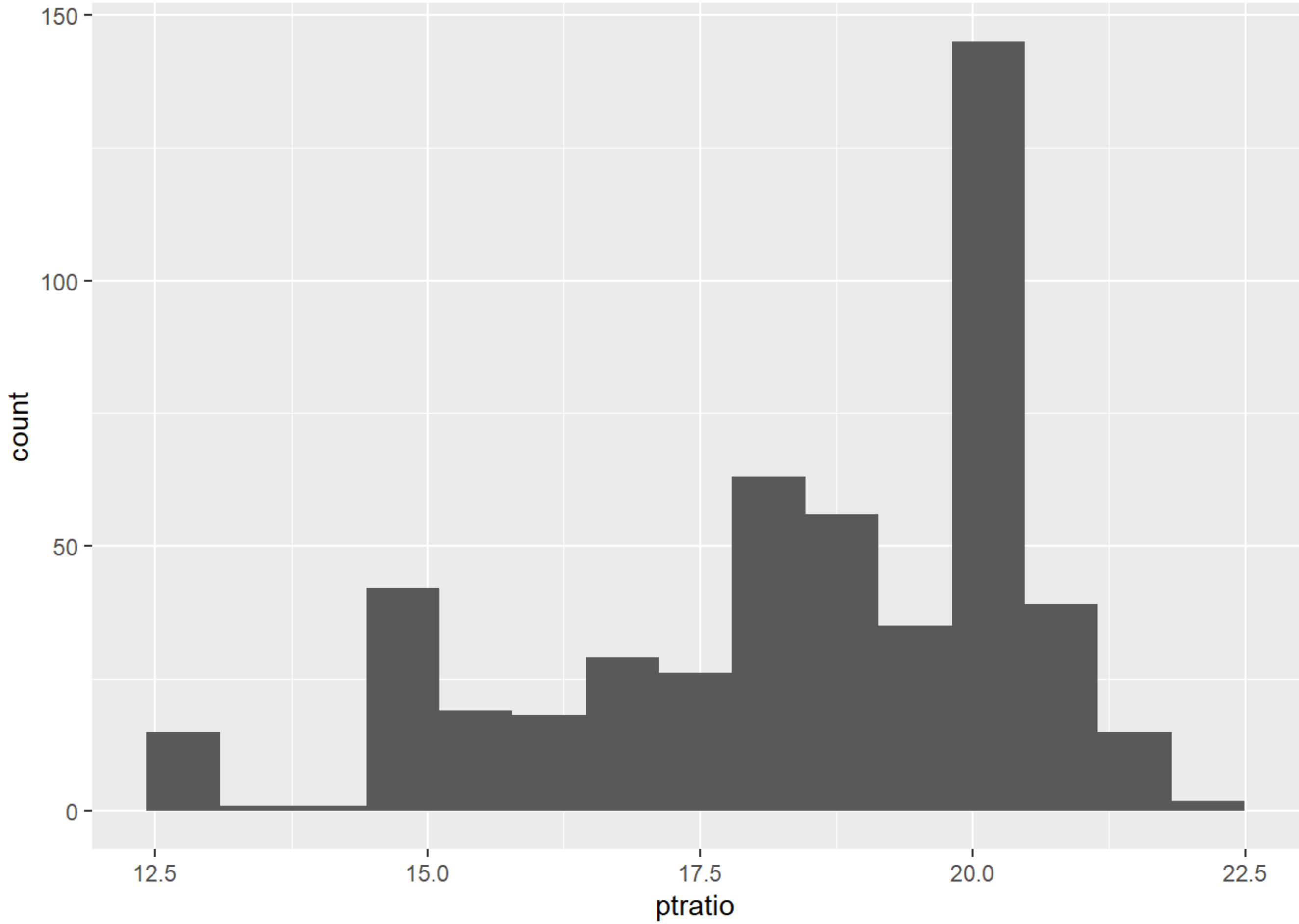
```
bos %>%
  ggplot(aes(tax)) +
  geom_histogram(bins = 20)
```



Normal cluster around 300 (from 200 - 440), then a few much further up above 600.

Pupil-teacher ratio

```
bos %>%
  ggplot(aes(ptratio)) +
  geom_histogram(bins = 15)
```



Skewed left normal curve around 19, from 15 to 21.5

(e)

```
bos %>%
  count(chas)
```

```
##   chas   n
## 1    0 471
```

```
## 2     1  35
```

35 suburbs are river-bound, 471 are not.

(f)

```
median(bos$ptratio)
```

```
## [1] 19.05
```

19.05

(g)

```
bos %>%
  arrange(medv) %>%
  slice(1)
```

```
##       crim zn indus chas   nox    rm age    dis rad tax ptratio black lstat medv
## 1 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.9 30.59    5
```

This suburb has high crime, high taxes, and a high pupil-teacher ratio.

(h)

```
bos %>%
  filter(rm > 7) %>%
  nrow()
```

```
## [1] 64
```

64 / 506, or 12.6%

```
bos %>%
  filter(rm > 8) %>%
```

```
  nrow()
```

```
## [1] 13
```

13 / 506, or 2.6%

```
bos_roomy <- bos %>%
  filter(rm > 8)
```

```
bos_roomy %>%
  summarise_all(mean)
```

```
##        crim       zn    indus      chas       nox       rm      age      dis
## 1 0.7187954 13.61538 7.078462 0.1538462 0.5392385 8.348538 71.53846 3.430192
##        rad      tax  ptratio    black lstat medv
## 1 7.461538 325.0769 16.36154 385.2108  4.31 44.2
```

Very high crime, the highest in the datset.

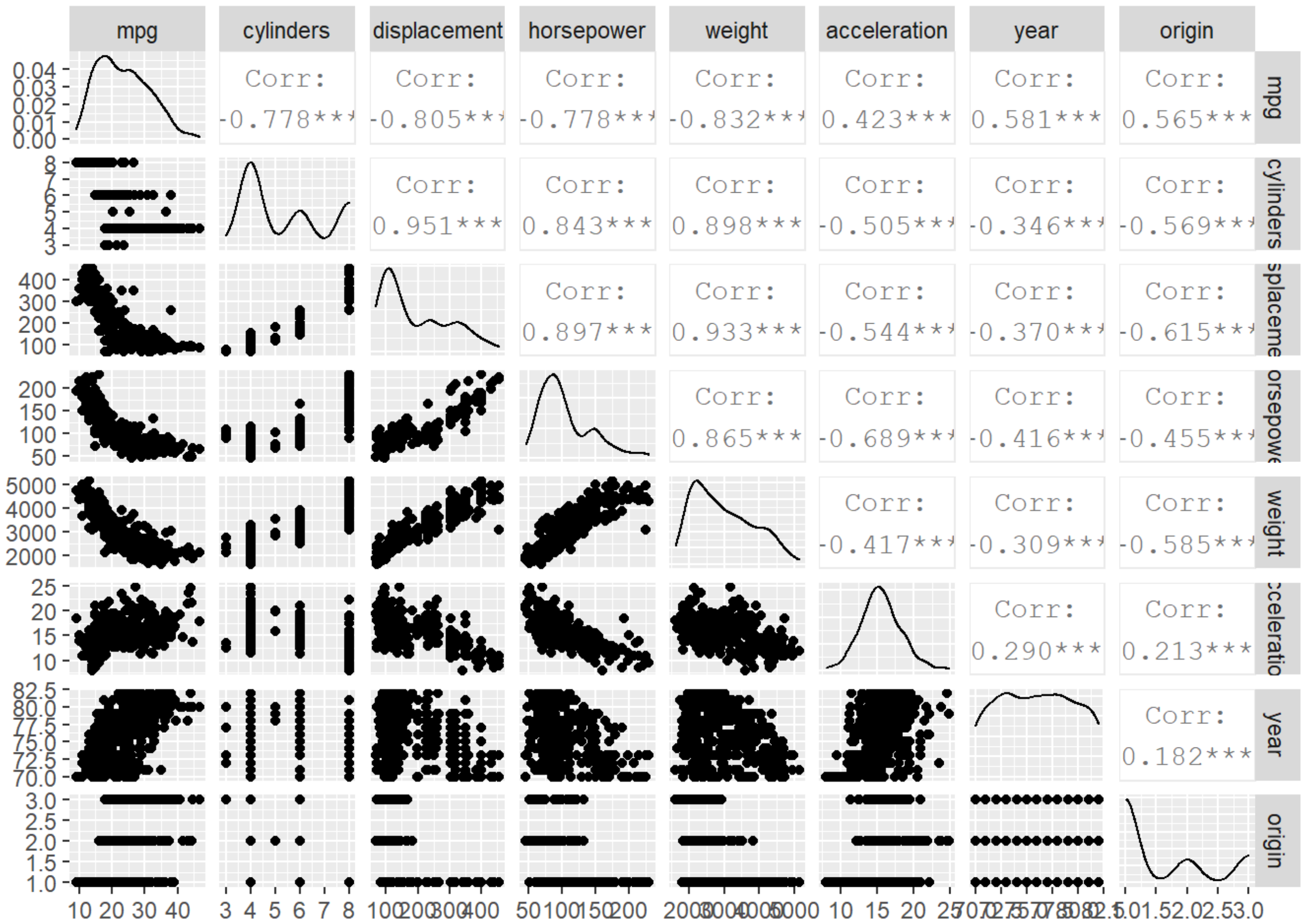
Chapter 3 Questions

(9)

(a)

```
auto <- ISLR::Auto
auto_num <- auto %>%
  select(-name)
```

```
ggpairs(auto_num)
```



(b)

```
auto_num %>%
  cor()
```

```
##                     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
```

(c)

```
fit <- lm(mpg ~ ., data = auto_num)

summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = auto_num)
##
## 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. `mpg` relates very strongly to `origin`, `year`, `weight` and strongly to `displacement`.

ii. The 4 mentioned in i.

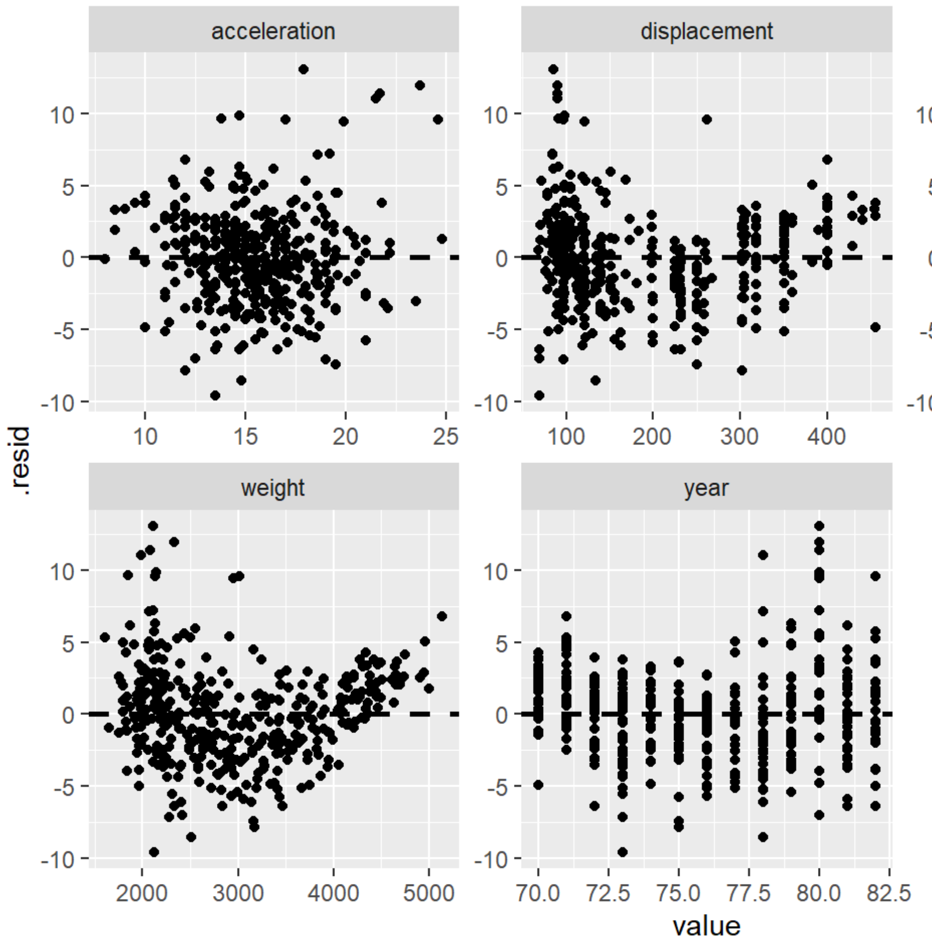
iii. This suggests that when holding all other predictors constant, an increase in year still leads to higher mpg. This suggests to me that there are other factors besides those covered in this dataset that are leading to cars with better mpg.

(d)

```
fit_tidy <- fit %>%
  broom::augment()
```

Looking at residuals vs predictor values to detect non-randomness in the residuals-- this would indicate the model is not fitting the data well.

```
fit_tidy %>%
  pivot_longer(cols = displacement:year,
               names_to = "var",
               values_to = "value") %>%
  ggplot(aes(value, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", size = 1) +
  facet_wrap(~ var, scales = "free")
```



Residuals look non-linear for both `weight` and `displacement`, while heteroskedasticity may be present for `horsepower` and `year`.

(e)

```
fit_int <- lm(mpg ~ weight*cylinders + weight, data = auto_num)

summary(fit_int)
```

```
##
## Call:
## lm(formula = mpg ~ weight * cylinders + weight, data = auto_num)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4916  -2.6225  -0.3927   1.7794  16.7087
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       65.3864559  3.7333137  17.514  < 2e-16 ***
## weight            -0.0128348  0.0013628  -9.418  < 2e-16 ***
## cylinders         -4.2097950  0.7238315  -5.816 1.26e-08 ***
## weight:cylinders   0.0010979  0.0002101   5.226 2.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.165 on 388 degrees of freedom
## Multiple R-squared:  0.7174, Adjusted R-squared:  0.7152
## F-statistic: 328.3 on 3 and 388 DF,  p-value: < 2.2e-16
```

In this simple model, the interaction of weight and cylinders is significant.

(f)

```
fit_log <- lm(mpg ~ cylinders + displacement + log(horsepower) + weight + acceleration + year, data = auto_num)

summary(fit_log)
```

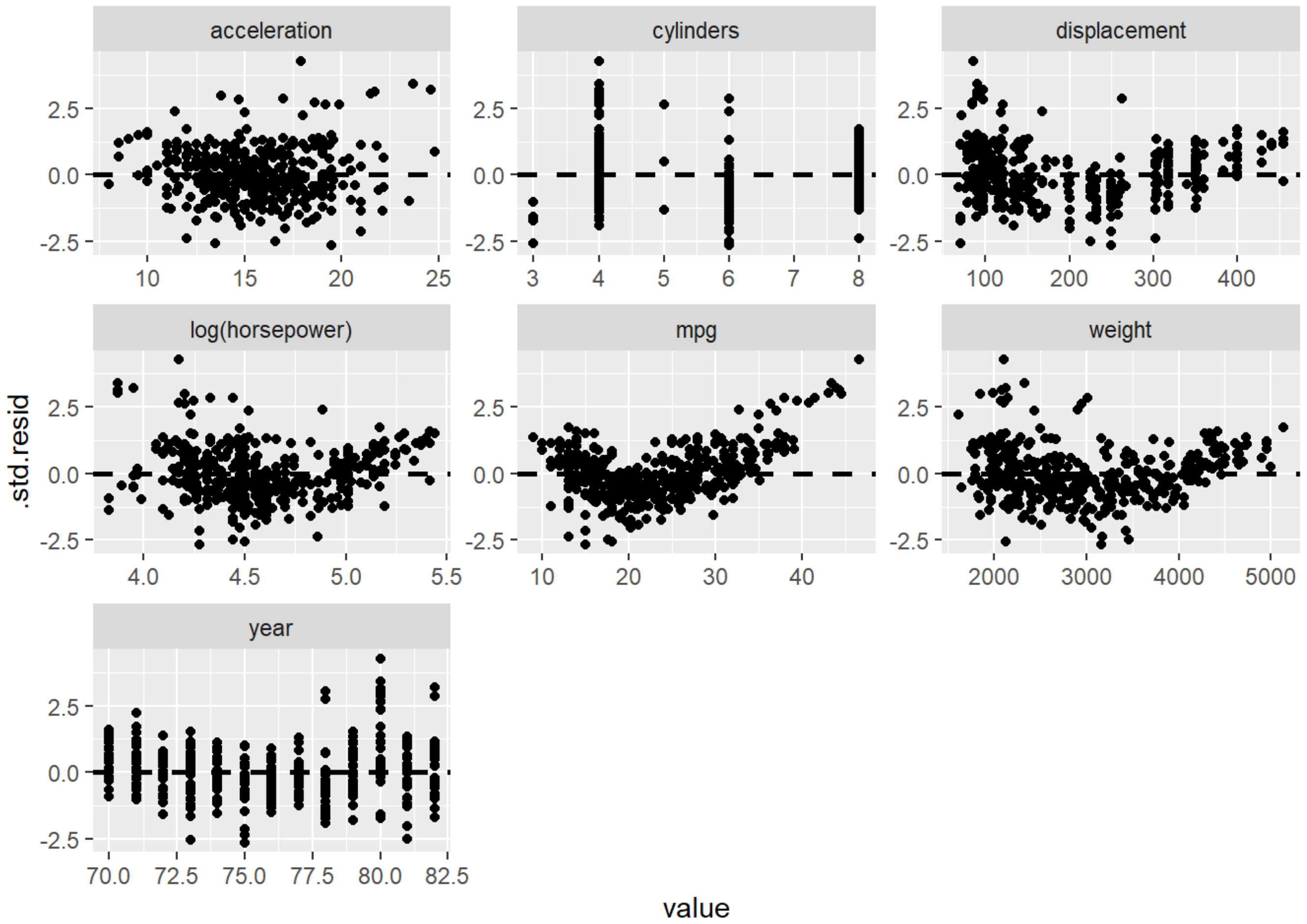
```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + log(horsepower) +
##     weight + acceleration + year, data = auto_num)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -8.6778 -2.0080 -0.3142  1.9262 14.0979
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     29.1713000  8.9291383   3.267  0.00118 **
## cylinders       -0.3563199  0.3181815  -1.120  0.26347
## displacement     0.0088277  0.0068866   1.282  0.20066
## log(horsepower) -8.7568129  1.5958761  -5.487 7.42e-08 ***
## weight          -0.0044304  0.0007213  -6.142 2.03e-09 ***
## acceleration    -0.3317439  0.1077476  -3.079  0.00223 **
## year             0.6979715  0.0503916  13.851  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.308 on 385 degrees of freedom
## Multiple R-squared:  0.8231, Adjusted R-squared:  0.8203
## F-statistic: 298.5 on 6 and 385 DF,  p-value: < 2.2e-16
```

```
fit_log_tidy <- fit_log %>% augment()
```

I fit log(horsepower) because the residuals for horsepower looked heteroskedastic.

```
fit_log_tidy %>%
  pivot_longer(cols = mpg:year,
               names_to = "var",
               values_to = "value") %>%
  ggplot(aes(value, .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", size = 1) +
  facet_wrap(~ var, scales = "free")
```



Difficult to tell whether this helped. The residuals for horsepower might be slightly more random, less non-linear?

(13)

(a)

```
x <- rnorm(n = 100, mean = 0, sd = 1)
```

(b)

```
eps <- rnorm(n = 100, mean = 0, sd = .25)
```

(c)

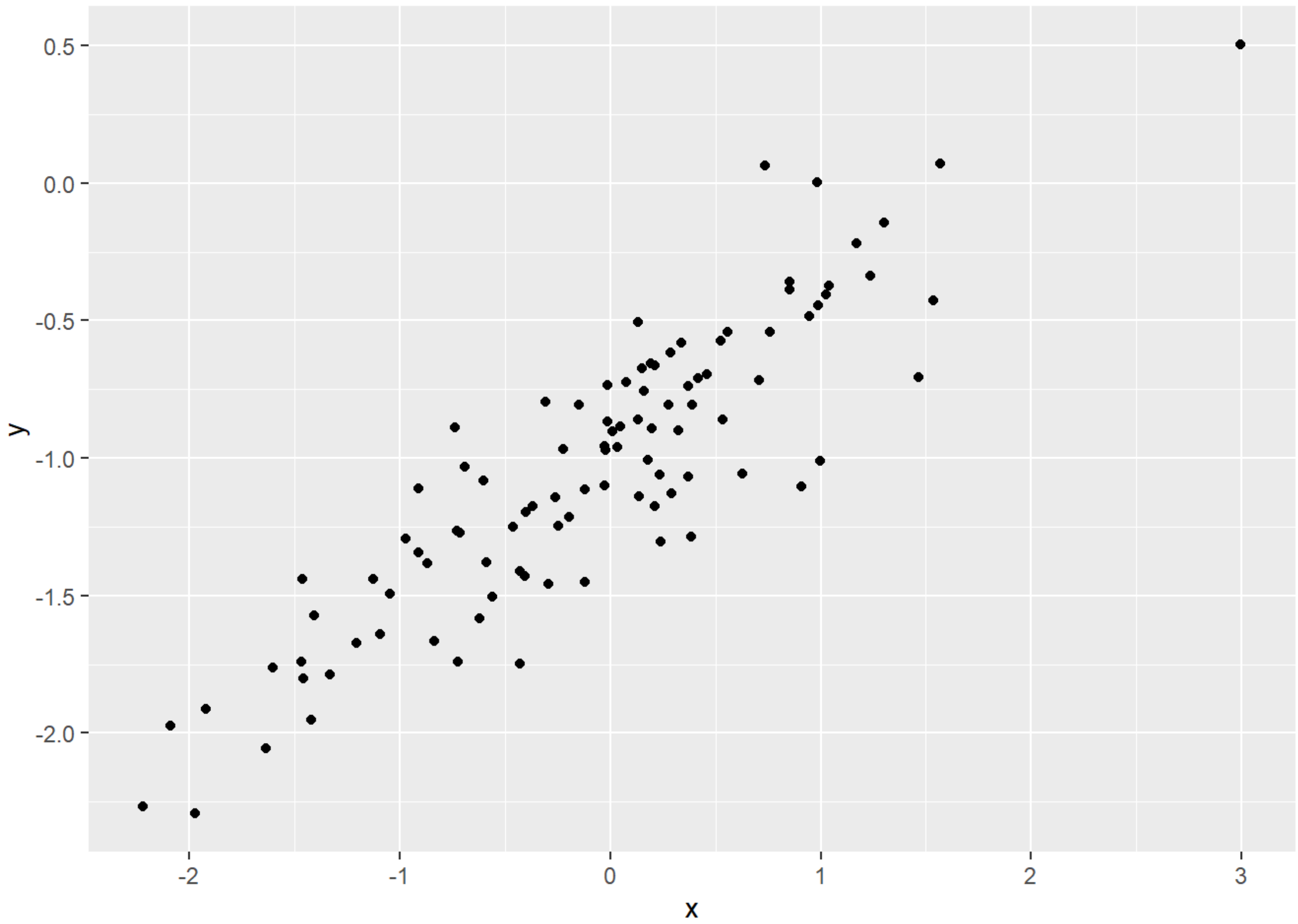
```
y <- -1 + (.5 * x) + eps
```

`y` is length 100 (same as `x` and `eps`). In this model, $\hat{\beta}_0 = -1$ and $\hat{\beta}_1 = .5$

(d)

```
df <- cbind(x, eps, y) %>%
  as_tibble()
```

```
df %>%
  ggplot(aes(x, y)) +
  geom_point()
```



Can see there's a very strong linear relationship

(e)

```
fit_l <- lm(y ~ x, data = df)

summary(fit_l)
```

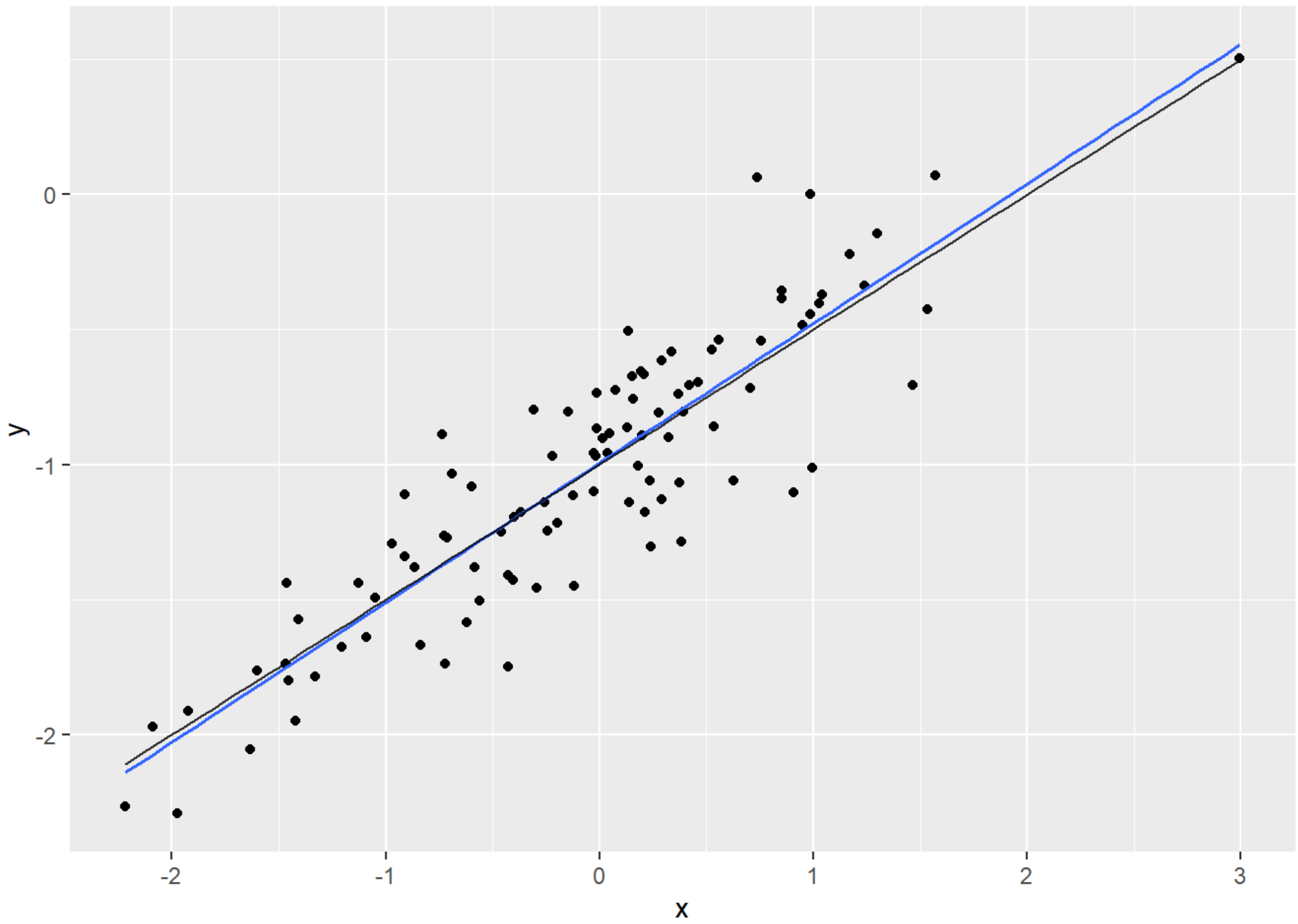
```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57837 -0.13049  0.03656  0.14982  0.67667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99366    0.02405  -41.32   <2e-16 ***
## x            0.51691    0.02651   19.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2394 on 98 degrees of freedom
## Multiple R-squared:  0.7951, Adjusted R-squared:  0.793
## F-statistic: 380.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_0 = -1.003$ (very close to $\beta_0 = -1$) and $\hat{\beta}_1 = .485$ (very close to $\beta_1 = .5$)

(f)

```
df %>%
  ggplot(aes(x, y)) +
  geom_point() +
  geom_smooth(method = "lm", se = F, alpha = .3, size = .7) +
  stat_function(fun = function(x) -1 + (0.5 * x), alpha = .8)
```

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




```
# legend(x = 1, y = 1, legend = "", col = "blue")
```

Very similar lines, the population line in black/grey and the fit to this dataset in blue.

(g)

```
fit_q <- lm(y ~ x + I(x^2), data = df)

summary(fit_q)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57716 -0.13624  0.03322  0.14967  0.67539
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.987012   0.028495  -34.64   <2e-16 ***
## x            0.515677   0.026766   19.27   <2e-16 ***
## I(x^2)      -0.008213   0.018676   -0.44    0.661
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2403 on 97 degrees of freedom
## Multiple R-squared:  0.7955, Adjusted R-squared:  0.7913
## F-statistic: 188.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

The quadratic term is not significant, and is not necessary. This makes sense after seeing the linear relationship visually.

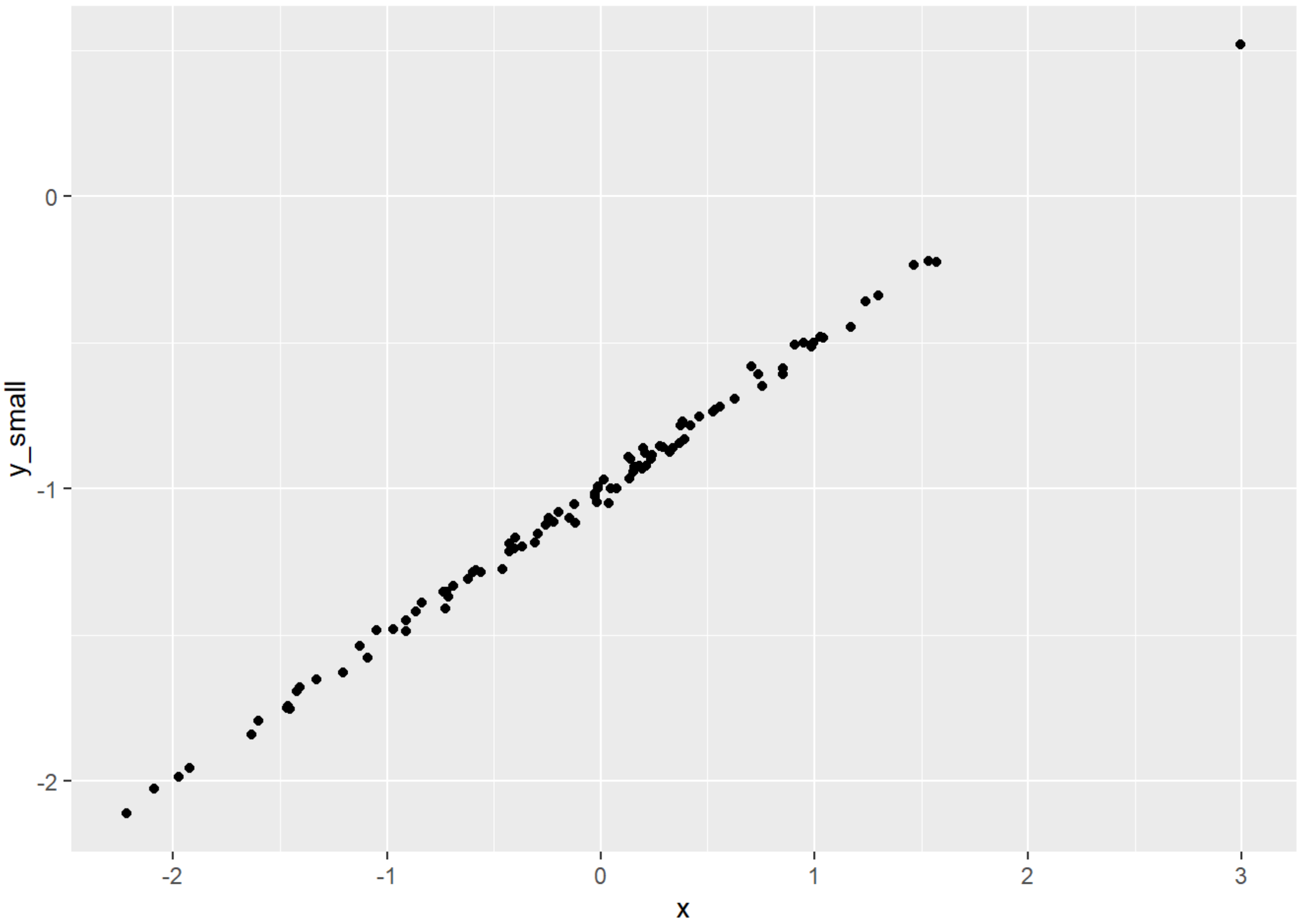
(h)

```
eps_small <- rnorm(n = 100, mean = 0, sd = .025)
```

```
y_small <- -1 + (.5 * x) + eps_small
```

```
df_small <- cbind(x, eps_small, y_small) %>%
  as_tibble()
```

```
df_small %>%
  ggplot(aes(x, y_small)) +
  geom_point()
```




```
fit_small <- lm(y_small ~ x, data = df_small)

summary(fit_small)
```

```
##
## Call:
## lm(formula = y_small ~ x, data = df_small)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.068436 -0.016265  0.001615  0.016880  0.064995
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.000831   0.002453  -408.0   <2e-16 ***
## x            0.502038   0.002704   185.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02441 on 98 degrees of freedom
## Multiple R-squared:  0.9972, Adjusted R-squared:  0.9971
## F-statistic: 3.448e+04 on 1 and 98 DF,  p-value: < 2.2e-16
```

Extremely high R^2, almost perfect linear fit when the amount of noise is small.

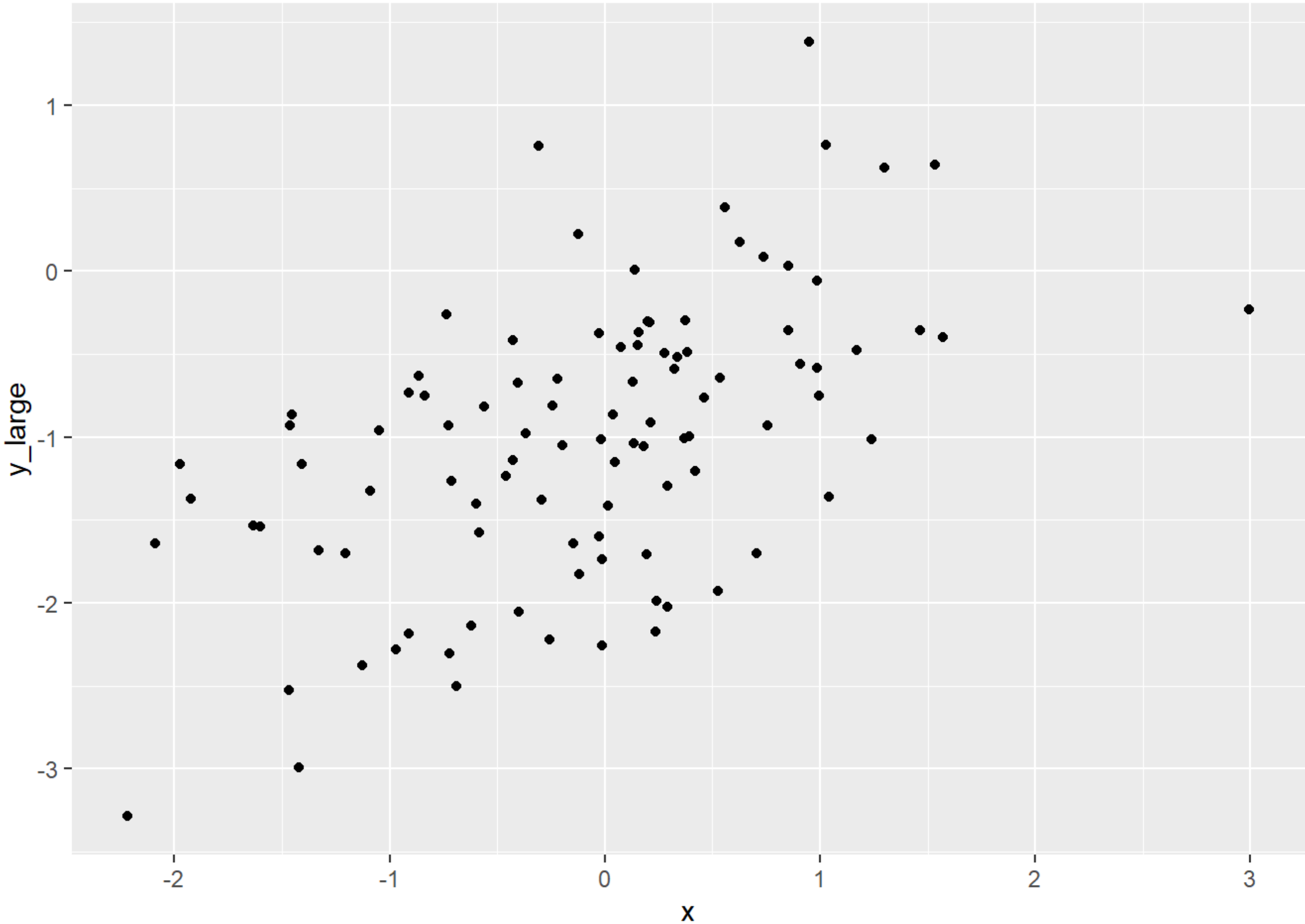
(i)

```
eps_large <- rnorm(n = 100, mean = 0, sd = .75)
```

```
y_large <- -1 + (.5 * x) + eps_large
```

```
df_large <- cbind(x, eps_large, y_large) %>%
  as_tibble()
```

```
df_large %>%
  ggplot(aes(x, y_large)) +
  geom_point()
```




```
fit_large <- lm(y_large ~ x, data = df_large)

summary(fit_large)
```

```
##
## Call:
## lm(formula = y_large ~ x, data = df_large)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.3274 -0.5104  0.0238  0.5083  1.8708
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.96249    0.07212 -13.345  < 2e-16 ***
## x            0.50014    0.07949   6.292 8.82e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7178 on 98 degrees of freedom
## Multiple R-squared:  0.2877, Adjusted R-squared:  0.2804
## F-statistic: 39.58 on 1 and 98 DF,  p-value: 8.824e-09
```

Lower R^2 when noise is greater.

(j)

Using Estimate +- 2 SE's from the `summary(fit)` outputs scattered above.

Original

$\beta_0 : [-.98, -1.02]$ $\beta_1 : [.465, .505]$

Small E

$\beta_0 : [-.999, 1.0001]$ $\beta_1 : [.4999, .5001]$

Large E

$\beta_0 : [-.86, 1.00]$ $\beta_1 : [.47, .59]$

Chapter 4 Questions

(6)

(a)

Plugging values of the beta hats into the logit equation (e^Bo+B1x) / (1 + e^Bo+B1x+..), we get P(student gets A) = .38

(b)

Setting the solved-for equation above equal to 0.5 (50% chance) and solving for hours to study x, you get 50 hours.

(9)

(a)

With our odds at .37, we can set that equal to $\frac{p(x)}{1-p(x)}$ and we end up with p(x) = .27. So there's a 27% chance of default.

(b)

Just divide .16 by (1 - .16) and we get .16 / .84 = .19