Homework 2

Daniel Dulaney

September 6, 2020

Note to self (and the grader): Procrastination is bad! This homework is only (very) partially completed :(

```
library(tidyverse)
library(here)
library(ISLR)
library(skimr)
library(tidymodels)
library(discrim)
```

4.7

4.10

(a)

```
weekly <- ISLR::Weekly %>%
  as_tibble()

skim(weekly)
```

Data summary

Name	weekly
Number of rows	1089
Number of columns	9

Column type frequency:	
factor	1
numeric	8
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Direction	0	1	FALSE	2	Up: 605, Dow: 484

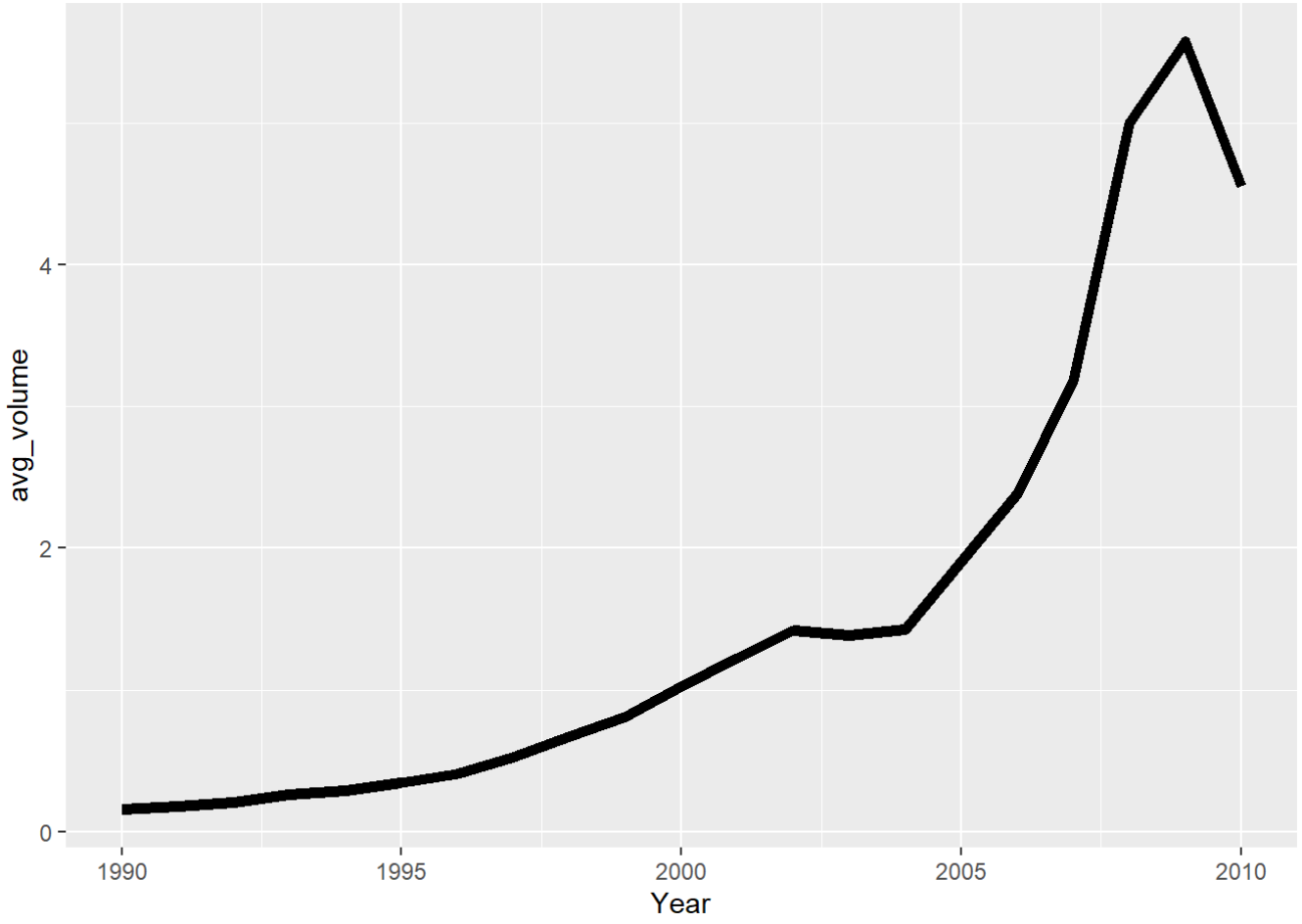
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Year	0	1	2000.05	6.03	1990.00	1995.00	2000.00	2005.00	2010.00	
Lag1	0	1	0.15	2.36	-18.20	-1.15	0.24	1.41	12.03	
Lag2	0	1	0.15	2.36	-18.20	-1.15	0.24	1.41	12.03	
Lag3	0	1	0.15	2.36	-18.20	-1.16	0.24	1.41	12.03	
Lag4	0	1	0.15	2.36	-18.20	-1.16	0.24	1.41	12.03	
Lag5	0	1	0.14	2.36	-18.20	-1.17	0.23	1.41	12.03	
Volume	0	1	1.57	1.69	0.09	0.33	1.00	2.05	9.33	
Today	0	1	0.15	2.36	-18.20	-1.15	0.24	1.41	12.03	

```
# how much has the volume of shares traded changed over time?
weekly %>%
  group_by(Year) %>%
  summarise(avg_volume = mean(Volume)) %>%
```

```
  ggplot(aes(Year, avg_volume)) +
  geom_path(size = 2)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



(b)

```
log_fit_full <- logistic_reg() %>%
  set_engine("glm") %>%
  fit(Direction ~ Volume + Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = weekly)

summary(log_fit_full$fit)
```

```
##
## Call:
## stats::glm(formula = Direction ~ Volume + Lag1 + Lag2 + Lag3 +
##     Lag4 + Lag5, family = stats::binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Volume      -0.02274    0.03690  -0.616   0.5377
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

`Lag2` is the only statistically significant predictor.

(c)

(d)

```
weekly_train <- weekly %>%
  filter(Year <= 2008)

weekly_test <- weekly %>%
  filter(Year > 2008)
```

```
log_fit <- logistic_reg() %>%
  set_engine("glm") %>%
  fit(Direction ~ Volume + Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = weekly_train)

predict(log_fit, weekly_test) %>%
  cbind(weekly_test) %>%
  conf_mat(truth = "Direction", estimate = ".pred_class")
```

```
##           Truth
## Prediction Down Up
##       Down   31 44
##       Up     12 17
```

(e)

```
lda_fit <- discrim_linear() %>%
  set_engine("MASS") %>%
  fit(Direction ~ Volume + Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = weekly_train)

predict(lda_fit, weekly_test) %>%
  cbind(weekly_test) %>%
  conf_mat(truth = "Direction", estimate = ".pred_class")
```

```
##           Truth
## Prediction Down Up
##       Down   31 44
##       Up     12 17
```

(f)

```
# qda_fit <- discrim_regularized() %>%
#   set_engine("MASS") %>%
#   fit(Direction ~ Volume + Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = weekly_train)
#
# predict(qda_fit, weekly_test) %>%
#   cbind(weekly_test) %>%
#   conf_mat(truth = "Direction", estimate = ".pred_class")
```

(g)

```
knn_fit <- nearest_neighbor() %>%
  set_engine("kknn") %>%
  set_mode("classification") %>%
  fit(Direction ~ Volume + Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = weekly_train)

predict(knn_fit, weekly_test) %>%
  cbind(weekly_test) %>%
  conf_mat(truth = "Direction", estimate = ".pred_class")
```

```
##           Truth
## Prediction Down Up
##       Down   25 36
##       Up     18 25
```

(h)

Out of logistic regression, LDA, and KNN (couldn't get QDA running), the logistic and LDA models have the same confusion matrix and are both better than the KNN model.

(i)

5.5

5.8

(a)

```
set.seed(1)

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

df <- cbind(x, y) %>%
  as_tibble() %>%
  mutate(x2 = x^2,
         x3 = x^3,
         x4 = x^4)
```

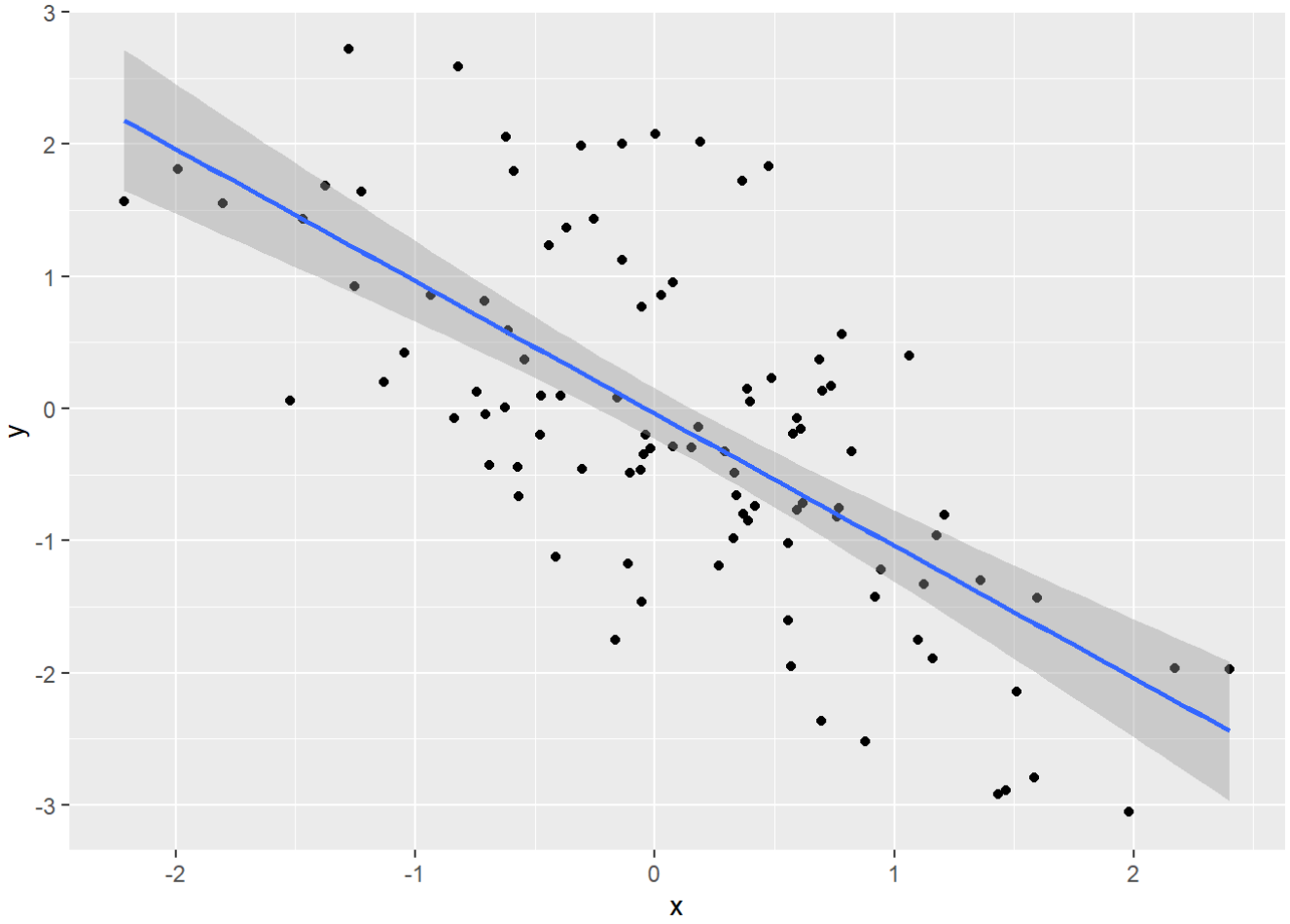
n = 100 and p = 1

(b)

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

```
  geom_point() +
  geom_smooth(method = "lm")
```

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



Looks like a strong negative linear relationship between `x` and `y`.

(c)

```
set.seed(20)

loo_cv(df)
```

```
## # Leave-one-out cross-validation
## # A tibble: 100 x 2
##    splits         id
##    <list>         <chr>
##  1 <split [99/1]> Resample1
##  2 <split [99/1]> Resample2
##  3 <split [99/1]> Resample3
##  4 <split [99/1]> Resample4
##  5 <split [99/1]> Resample5
##  6 <split [99/1]> Resample6
##  7 <split [99/1]> Resample7
##  8 <split [99/1]> Resample8
##  9 <split [99/1]> Resample9
## 10 <split [99/1]> Resample10
## # ... with 90 more rows
```

```
mod_1 <- lm(y ~ x, data = df)
mod_2 <- lm(y ~ x + x2, data = df)
mod_3 <- lm(y ~ x + x2 + x3, data = df)
mod_4 <- lm(y ~ x + x2 + x3 + x4, data = df)
```

(d)

(e)

(f)

5.9

(a)

```
boston <- MASS::Boston
```

```
mean_medv <- mean(boston$medv)

mean_medv
```

```
## [1] 22.53281
```

$\hat{\mu}$ = 22.5

(b)

```
sd_medv <- sd(boston$medv)

# 95% confidence interval (mu +- 1.96(SE(mu)))
lower <- mean_medv - (1.96 * (sd_medv / nrow(boston)))
upper <- mean_medv + (1.96 * (sd_medv / nrow(boston)))

lower
```

```
## [1] 22.49718
```

```
upper
```

```
## [1] 22.56843
```

95% confidence interval for μ: [22.50, 22.57], and the standard error is **.018.**

(c)

```
boston_bootstraps <- bootstraps(boston, times = 100)

medv_means <- boston_bootstraps$splits %>%
  map_dbl(function(x) {
    dat <- as.data.frame(x)$medv
    mean(dat)
  })

quantile(medv_means, probs = c(.05, .95))
```

```
##       5%      95%
## 21.88010 23.32763
```

The 95% bootstrap confidence interval for μ is [21.85, 23.20]

(d)

(e)

(f)

(g)

(h)

6.10