432 Homework 3 Answer Sketch and Grading Rubric

432 TAs

Due 2020-02-18. Version: 2020-02-19

Contents

Setup and Data Ingest

```r
library(skimr)
library(broom)
library(magrittr)
library(janitor)
library(caret)
library(naniar)
library(leaps)
library(knitr)
library(patchwork)
library(here)
library(tidyverse)

theme_set(theme_bw())

hbp432 <- read_csv(here("data/hbp432.csv")) %>%
    clean_names()
```

Question 1 (30 points)

Again, consider the `hbp432` data used in Homework 1. Build your best model for the prediction of body-mass index, considering the following 14 predictors: `practice`, `age`, `race`, `eth_hisp`, `sex`, `insurance`, `income`, `hsgrad`, `tobacco`, `depdiag`, `sbp`, `dbp`, `statin` and `bpmed`. Use an appropriate best subsets procedure to aid in your search, and use a cross-validation strategy to assess and compare potential models.

- Feel free to omit the cases with missing values in the variables you are considering (these 14 predictors, plus the `bmi` outcome) before proceeding. This should not materially affect your sample size very much. In the answer sketch, we will use a complete cases analysis.
- Use the `nvmax = 7` command within your call to `regsubsets` to limit your investigation to models containing no more than seven of these candidate predictors.
- Do not transform any variables, and consider models with main effects only so that no product terms are used.
- A 5-fold cross-validation strategy would be very appropriate. Another reasonable choice would involve partitioning the data once (prior to fitting any models) into training and test samples, as we did in 431.

Be sure to provide a written explanation of your conclusions and specify the variables in your final model, in complete sentences.

Data Preparation

We'll need to manage the data a bit. Specifically, we'll…

1. Calculate the outcome, `bmi`.
2. Express all multi-categorical variables in `hbp432` as factors, with `type.convert()`, except for the subject identifier (`subject`)
3. Restrict ourselves to complete cases, so as to avoid problems with missing data.
4. Use only the variables we're considering as predictors, plus the outcome (`bmi`) and `subject` code.

```
hw3q1 <- hbp432 %>%
    mutate(bmi = weight / (height^2)) %>%
    type.convert() %>%
    mutate(subject = as.character(subject)) %>%
    drop_na() %>%
    select(subject, bmi,
           practice, age, race, eth_hisp,
           sex, insurance, income, hsgrad, tobacco,
           depdiag, sbp, dbp, statin, bpmed)
```

Sanity Check

Let's check to be sure all predictors are either a factor or numeric, and that we now have no missing values.

```
skim_without_charts(hw3q1)
```

Table 1: Data summary

Name	hw3q1
Number of rows	387
Number of columns	16
Column type frequency:	
character	1
factor	7
numeric	8
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
subject	0	1	4	4	0	387	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
practice	0	1	FALSE	4	B: 115, A: 102, D: 95, C: 75
race	0	1	FALSE	4	Bla: 266, Whi: 103, Asi: 9, Mul: 9
eth_hisp	0	1	FALSE	2	No: 361, Yes: 26
sex	0	1	FALSE	2	M: 204, F: 183
insurance	0	1	FALSE	4	Med: 188, Com: 103, Med: 79, Uni: 17
tobacco	0	1	FALSE	3	For: 140, Nev: 134, Cur: 113
depdiag	0	1	FALSE	2	No: 313, Yes: 74

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
bmi	0	1	31.36	7.96	15.1	26.17	29.7	35.35	61.3
age	0	1	62.01	12.60	29.0	54.00	61.0	70.00	89.0
income	0	1	34995.09	16184.50	7200.0	24500.00	33000.0	42100.00	102500.0
hsgrad	0	1	80.95	8.64	38.4	74.20	82.6	87.60	96.5
sbp	0	1	135.21	18.45	94.0	123.50	134.0	144.00	210.0
dbp	0	1	77.39	11.40	48.0	70.00	78.0	83.00	136.0
statin	0	1	0.56	0.50	0.0	0.00	1.0	1.00	1.0
bpmed	0	1	0.72	0.45	0.0	0.00	1.0	1.00	1.0

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100

OK. This looks reasonable. Now, we could partition the data first into training and test samples at this point, but instead, we'll do the exhaustive search first and then do 5-fold cross-validation later.

Performing an exhaustive search with `regsubsets`

```
q1_best <- regsubsets(bmi ~ practice + age + race +
                     eth_hisp + sex + insurance + income +
                     hsgrad + tobacco + depdiag + sbp + dbp +
                     statin + bpmed,
                   data = hw3q1, nvmax = 7, nbest = 1)

q1_summ <- summary(q1_best)
```

The `outmat` section of the summary output has the listing of fitted models that we want. Note that the multi-categorical variables, like `race`, `practice`, `insurance`, and `tobacco` are split into their indicators for each level.

```
q1_summ$outmat
```

```
         practiceB practiceC practiceD age raceBlack/AA raceMultiracial
1  ( 1 ) " "       " "       " "       "*" " "          " "
2  ( 1 ) " "       " "       " "       "*" " "          " "
3  ( 1 ) " "       " "       " "       "*" " "          " "
4  ( 1 ) " "       " "       " "       "*" " "          " "
5  ( 1 ) " "       " "       " "       "*" " "          " "
6  ( 1 ) "*"       " "       " "       "*" " "          " "
7  ( 1 ) "*"       " "       " "       "*" " "          " "
         raceWhite eth_hispYes sexM insuranceMedicaid insuranceMedicare
1  ( 1 ) " "       " "         " "  " "               " "
2  ( 1 ) " "       " "         "*"  " "               " "
3  ( 1 ) " "       " "         "*"  " "               " "
4  ( 1 ) " "       " "         "*"  " "               " "
5  ( 1 ) " "       " "         "*"  " "               " "
6  ( 1 ) " "       " "         "*"  " "               " "
7  ( 1 ) "*"       " "         "*"  " "               " "
         insuranceUninsured income hsgrad tobaccoFormer tobaccoNever depdiagYes
1  ( 1 ) " "                " "    " "    " "           " "          " "
2  ( 1 ) " "                " "    " "    " "           " "          " "
3  ( 1 ) " "                " "    " "    " "           " "          " "
4  ( 1 ) " "                " "    " "    " "           "*"          " "
5  ( 1 ) " "                " "    " "    "*"           "*"          " "
6  ( 1 ) " "                " "    " "    "*"           "*"          " "
7  ( 1 ) " "                " "    " "    "*"           "*"          " "
         sbp dbp statin bpmed
1  ( 1 ) " " " " " "    " "
2  ( 1 ) " " " " " "    " "
3  ( 1 ) " " " " "*"    " "
4  ( 1 ) " " " " "*"    " "
5  ( 1 ) " " " " "*"    " "
6  ( 1 ) " " " " "*"    " "
```

```
7  ( 1 ) " " " " "*"    " "
```

So, here are our “best subsets” models:

Inputs	Predictors Included (in addition to Intercept)
1	`age`
2	`age`, `sex`
3	Model 2 + `statin`
4	Model 3 + `tobaccoNever`
5	Model 4 + `tobaccoFormer`
6	Model 5 + `insuranceUninsured`
7	**Model 5** + `raceWhite` and `practiceB`

Notice that Model 7 doesn’t include `insuranceUninsured` like Model 6 does.

Fit Quality Statistics

```
q1_winners <- tbl_df(q1_summ$which) %>%
    mutate(inputs = 1:(q1_best$nvmax - 1),
           r2 = q1_summ$rsq,
           adjr2 = q1_summ$adjr2,
           cp = q1_summ$cp,
           bic = q1_summ$bic,
           rss = q1_summ$rss) %>%
    select(inputs, adjr2, cp, bic, everything())

q1_winners %>%
  select(inputs, adjr2, cp, bic) %>%
  kable(digits = c(0, 3, 1, 1))
```

inputs	adjr2	cp	bic
1	0.104	52.6	-31.8
2	0.176	18.9	-58.9
3	0.193	11.4	-62.2
4	0.197	10.8	-58.9
5	0.208	6.2	-59.6
6	0.211	5.8	-56.1
7	0.215	5.1	-52.9

Comparing Best Subsets Models on Summary Measures

To make it easier to compare, we’ll create separate graphs for adjusted R^2, Mallows’ C_p, and BIC (Bayes Information Criterion).

Code for Adjusted R^2 plot

```
p1 <- ggplot(q1_winners, aes(x = inputs, y = adjr2,
                           label = round(adjr2,3))) +
    geom_line() +
    geom_label() +
    geom_label(data = subset(q1_winners,
                             adjr2 == max(adjr2)),
               aes(x = inputs, y = adjr2,
                   label = round(adjr2,3)),
               fill = "yellow", col = "blue", size = 6) +
    scale_y_continuous(expand = expand_scale(mult = .1)) +
    labs(x = "# of regression inputs",
         y = "Adjusted R-squared")
```

Code for Mallows' C_p plot

```
p2 <- ggplot(q1_winners, aes(x = inputs, y = cp,
                       label = round(cp,1))) +
    geom_line() +
    geom_label() +
    geom_label(data = subset(q1_winners,
                             cp == min(cp)),
               aes(x = inputs, y = cp,
                   label = round(cp,1)),
               fill = "navy", col = "white", size = 6) +
    scale_y_continuous(expand = expand_scale(mult = .1)) +
    labs(x = "# of regression inputs",
         y = "Mallows' Cp")
```

Code for BIC plot

```
p3 <- ggplot(q1_winners, aes(x = inputs, y = bic,
                       label = round(bic, 1))) +
    geom_line() +
    geom_label() +
    geom_label(data = subset(q1_winners,
                             bic == min(bic)),
               aes(x = inputs, y = bic, label = round(bic,1)),
               fill = "red", col = "white", size = 6) +
    scale_y_continuous(expand = expand_scale(mult = .1)) +
    labs(x = "# of regression inputs",
         y = "Bayes Information Criterion")
```

Which looks best?

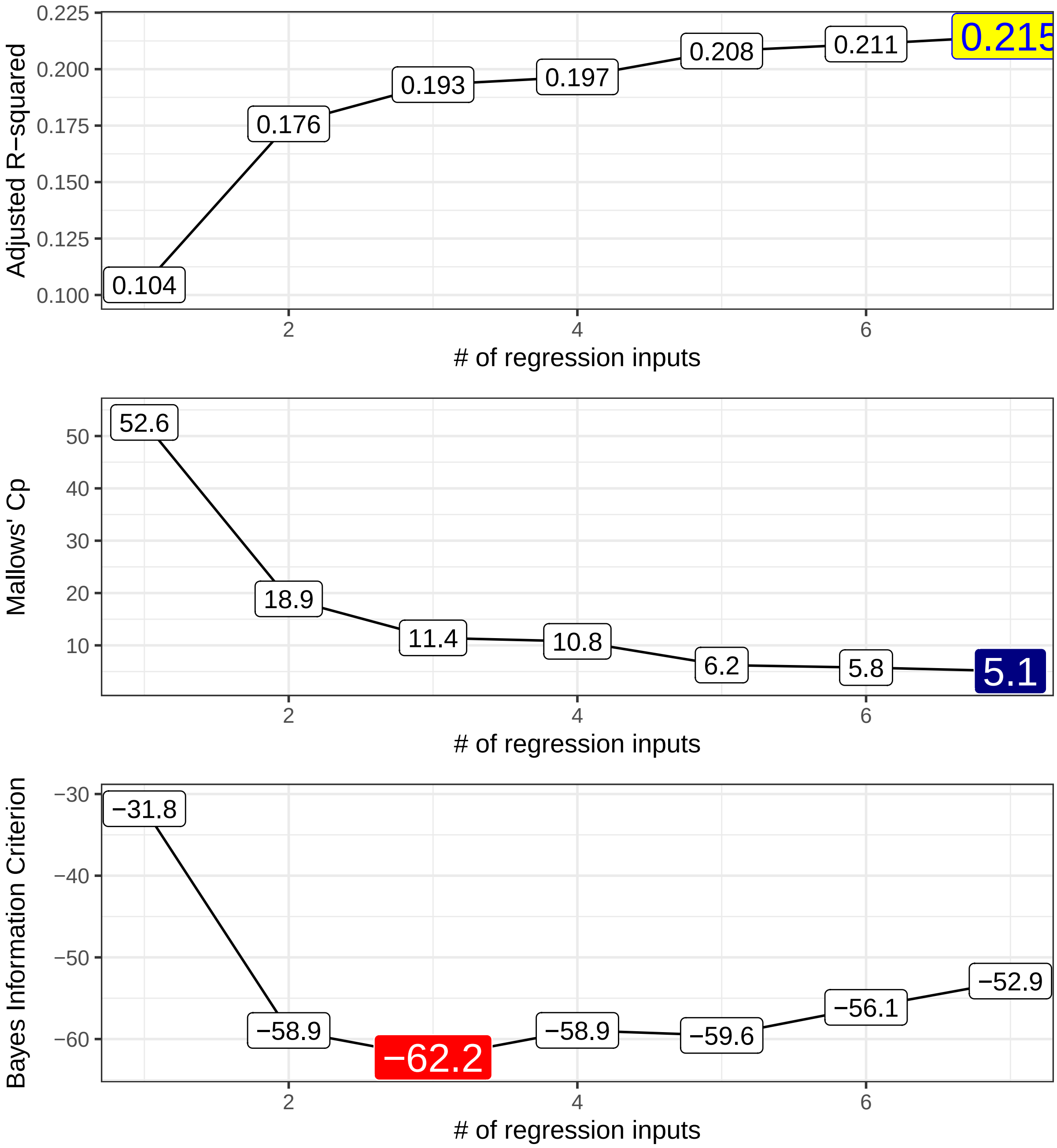
Remember that we want to maximize adjusted R^2 and minimize Mallows' C_p and BIC.

```
tibble(AdjR2 = which.max(q1_winners$adjr2),
       Cp = which.min(q1_winners$cp),
       BIC = which.min(q1_winners$bic))
```

```
# A tibble: 1 x 3
  AdjR2    Cp   BIC
  <int> <int> <int>
1     7     7     3
```

The Plots

```
p1 / p2 / p3
```



Selecting a Winner

The models we'll consider are:

Inputs	Predictors Included	Reason
3	`age`, `sex`, `statin`	lowest BIC
7	Model 3 + `tobaccoNever`, `tobaccoFormer`, `raceWhite`, and `practiceB`	highest adjusted R^2 and lowest C_p

We'll fit each of these models in turn, and then perform a 5-fold cross validation for each, then compare results. In each case, we'll calculate the root mean squared error of the predictions, the R^2, and the mean absolute prediction error across the complete samples.

5-fold cross-validation of model 3

```
set.seed(4322020)

train_c <- trainControl(method = "cv", number = 5)

model3_cv <- train(bmi ~ age + sex + statin,
                   data = hw3q1, method = "lm",
                   trControl = train_c)

model3_cv
```

```
Linear Regression

387 samples
  3 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 308, 311, 309, 310, 310
Resampling results:

  RMSE      Rsquared   MAE
  7.214811  0.1985182  5.526208

Tuning parameter 'intercept' was held constant at a value of TRUE
```

5-fold cross-validation of model 7

```
set.seed(2020432)

train_c <- trainControl(method = "cv", number = 5)

model7_cv <- train(bmi ~ age + sex + statin + tobacco +
                     (race == "White") + (practice == "B"),
                   data = hw3q1, method = "lm",
                   trControl = train_c)
```

```
model7_cv
```

```
Linear Regression

387 samples
  6 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 310, 309, 309, 310, 310
Resampling results:

  RMSE      Rsquared  MAE
  7.170072  0.204714  5.487901

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Which model looks better?

```
bind_rows(model3_cv$results, model7_cv$results) %>%
    mutate(model = c("model3", "model7")) %>%
    select(model, Rsquared, RMSE, MAE)
```

```
   model  Rsquared     RMSE      MAE
1 model3 0.1985182 7.214811 5.526208
2 model7 0.2047140 7.170072 5.487901
```

Model 7 has a larger cross-validated R^2 and smaller RMSE and MAE, so it looks like the stronger model.

So, we select the model with seven inputs.

Moving forward with the 7-input model

Refitting this model to the complete case sample of people without missing values on the variables we decided to use at the beginning, we have the following summary results. Notice that since our model includes indicator variables for tobacco = Former and tobacco = Never, and we only have three levels of tobacco (Current, Former and Never) we can simply include the tobacco factor to show this model. As with any included binary variable, we include the `sex` and `statin` factors as usual, too. For the `race` and `practice` multi-categorical variables, we instead need to isolate the indicator variable that best subsets selected.

```
summary(lm(bmi ~ age + sex + statin + tobacco +
             (race == "White") + (practice == "B"),
        data = hw3q1))
```

```
Call:
lm(formula = bmi ~ age + sex + statin + tobacco + (race == "White") +
    (practice == "B"), data = hw3q1)

Residuals:
     Min       1Q   Median       3Q      Max
-22.7152  -4.4963  -0.8724   3.9138  24.8475
```

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        44.27208    2.16840  20.417  < 2e-16 ***
age                -0.24231    0.03075  -7.880 3.50e-14 ***
sexM               -3.69047    0.76717  -4.810 2.18e-06 ***
statin              2.31139    0.73500   3.145  0.00179 **
tobaccoFormer       2.46501    0.91833   2.684  0.00759 **
tobaccoNever        2.66305    0.92469   2.880  0.00420 **
race == "White"TRUE  1.45986    0.88999   1.640  0.10177
practice == "B"TRUE  1.87835    0.90043   2.086  0.03764 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.052 on 379 degrees of freedom
Multiple R-squared:  0.2288,    Adjusted R-squared:  0.2146
F-statistic: 16.07 on 7 and 379 DF,  p-value: < 2.2e-16
```

The model appears to account for about 22% of the variation in `bmi`, and includes information on age, sex, statin and tobaco usage, plus indicator variables for white race/ethnicity and practice B.

Question 1 (Rubric: 30 points)

To receive 30 points, the students should:

- (3 points) correctly set up the data to run regsubsets
- (5 points) successfully perform the exhaustive search and identify seven models
- (5 points) correctly plot the summaries of those models for adjusted R^2, BIC and Mallows' C_p
- (4 points) use their plots to identify candidate models appropriately
- (4 points) perform 5-fold cross-validation correctly on each of those candidate models
- (3 points) come to an appropriate conclusion based on their RMSE, MAE, and R^2 and select a model
- (6 points) identify the final choice of model explicitly, as part of a written explanation of their conclusions.
- Subtract 3 points off of their total score if they fail to deal with the missing data in a sensible way.
- Subtract 3 points if they fail to treat the multi-categorical variables as factors.
- A reasonable but not completely successful attempt should receive points for all of the pieces above that are correct. If they made a mistake early on, but then did everything else correctly in light of their early mistake, they should receive credit for the later pieces.
- A completely successful effort will thus receive the full 30 points.
- Provide comments to all students who score less than 30 for any reason other than typos.

Question 2-5 (40 points, total)

Using the `hbp432` data, you will build models to predict whether or not the subject has a statin prescription based on the subject's current LDL cholesterol and which of the four practices they receive care from. Fit logistic regression models both with and without an interaction term between the two practice (factor) and LDL level.

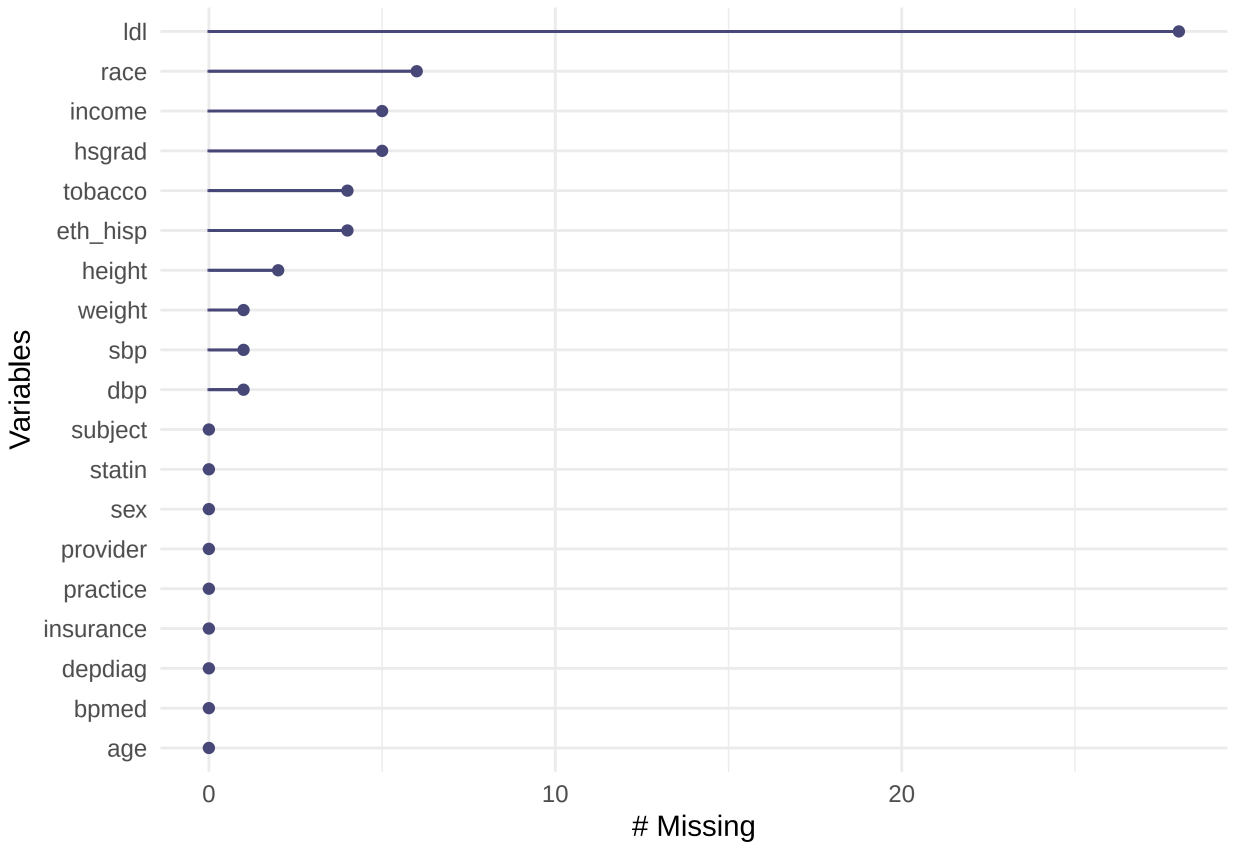
Question 2 (10 points)

Use a likelihood ratio test to compare the models, and describe its conclusions.

Check for missing data

First, let's check on missingness in the `hbp432` data.

```
gg_miss_var(hbp432)
```



As it turns out, we've got complete data on `statin` and `practice`, but we're missing 28 `ldl` observations, so we have to decide what to do about that. The simplest thing is to omit those cases, and then build our models on the remaining 404 observations. Another approach would have been to impute the missing `ldl` values.

```
hw3q2 <- hbp432 %>%
    filter(complete.cases(ldl)) %>%
    select(subject, statin, ldl, practice)
```

Building logistic regression models with and without interaction

We'll fit the models using `glm`. Our initial model predicts `statin` based on `ldl` and `practice` without an interaction and our second model includes an interaction between the predictors. I'm using 90% confidence intervals here anticipating Question 5.

```
model_without <- hbp432 %$%
    glm(statin ~ ldl + practice, family = binomial)

tidy(model_without, exponentiate = TRUE,
     conf.int=TRUE, conf.level=0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	3.488	0.347	1.984	6.220
ldl	0.994	0.003	0.989	0.999
practiceB	0.712	0.282	0.446	1.131
practiceC	0.608	0.311	0.363	1.013
practiceD	0.579	0.289	0.359	0.929

```
model_with <- hbp432 %$%
    glm(statin ~ ldl * practice, family = binomial)

tidy(model_with, exponentiate = TRUE,
     conf.int=TRUE, conf.level=0.9) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	63.517	0.880	16.050	294.823
ldl	0.964	0.009	0.950	0.978
practiceB	0.027	1.058	0.004	0.144
practiceC	0.011	1.140	0.002	0.066
practiceD	0.023	1.036	0.004	0.118
ldl:practiceB	1.035	0.010	1.018	1.053
ldl:practiceC	1.042	0.011	1.024	1.062
ldl:practiceD	1.035	0.010	1.018	1.053

Perform Likelihood Ratio Test

Now, we will compare the two models using the Model Likelihood Ratio Test.

```
anova(model_without, model_with, test= "LRT")
```

```
Analysis of Deviance Table

Model 1: statin ~ ldl + practice
Model 2: statin ~ ldl * practice
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       399     541.37
2       396     523.14  3   18.235 0.0003933 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because we have a small p-value, we conclude that adding the interaction term adds statistically detectable predictive value to the model.

Question 2 (Rubric: 10 points)

- Award up to 5 points for generating appropriate logistic regression models with and without interaction.
- Award 5 more points for correctly interpreting the usefulness of adding the interaction term using the Model Likelihood Test.

If students don't explicitly notice the missing `ldl` data and address it, they should lose 2 points. (Note that the default approach for `glm` is to omit those 28 cases.)

Question 3 (10 points)

Compare the confusion matrix produced by the two models (using a 0.5 cut point). Produce an attractively formatted table comparing the models in terms of prediction accuracy, sensitivity, specificity, as well as PPV and NPV.

Building the Confusion Matrix

To do this, we will use the `augment` function and then the `confusionMatrix` function from the `caret` package. We'll do this for each model separately, and then compare the results.

```
model_without_aug <- augment(model_without)

confuse_without <-
  model_without_aug %$% confusionMatrix(
    data= factor(.fitted >= 0.5),
    reference = factor(statin==1),
    positive ="TRUE")

confuse_without
```

```
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE   146  158
     TRUE     25   75

               Accuracy : 0.547
                 95% CI : (0.4971, 0.5963)
    No Information Rate : 0.5767
    P-Value [Acc > NIR] : 0.8957

                  Kappa : 0.1592

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.3219
            Specificity : 0.8538
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.4803
             Prevalence : 0.5767
         Detection Rate : 0.1856
   Detection Prevalence : 0.2475
```

```
      Balanced Accuracy : 0.5878

       'Positive' Class : TRUE
```

```
model_with_aug <- augment(model_with)

confuse_with <-
  model_with_aug %$% confusionMatrix(
    data= factor(.fitted >= 0.5),
    reference = factor(statin==1),
    positive="TRUE")

confuse_with
```

```
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE   157  174
     TRUE     14   59

               Accuracy : 0.5347
                 95% CI : (0.4847, 0.5841)
    No Information Rate : 0.5767
    P-Value [Acc > NIR] : 0.9606

                  Kappa : 0.1524

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.2532
            Specificity : 0.9181
         Pos Pred Value : 0.8082
         Neg Pred Value : 0.4743
             Prevalence : 0.5767
         Detection Rate : 0.1460
   Detection Prevalence : 0.1807
      Balanced Accuracy : 0.5857

       'Positive' Class : TRUE
```

Building an Attractively Formatted Table of Key Summaries

We asked you to build a table of these key summaries. Here's how Dr. Love would do it.

```
cwo <- tidy(confuse_without) %>%
  select(term, model_without = estimate) %>%
  filter(term %in%
           c("accuracy", "sensitivity", "specificity",
             "pos_pred_value", "neg_pred_value"))

cw <- tidy(confuse_with) %>%
  select(term, model_with = estimate) %>%
  filter(term %in%
           c("accuracy", "sensitivity", "specificity",
             "pos_pred_value", "neg_pred_value"))

left_join(cwo, cw, by = "term")
```

```
# A tibble: 5 x 3
  term           model_without model_with
  <chr>                  <dbl>      <dbl>
1 accuracy               0.547      0.535
2 sensitivity            0.322      0.253
3 specificity            0.854      0.918
4 pos_pred_value         0.750      0.808
5 neg_pred_value         0.480      0.474
```

If you just used the summaries printed previously, that would be OK, but you'd want to clean that up in practical work going forward using an approach like this.

As for interpreting these results, the model without the interaction has . . .

- a **stronger** performance in terms of predictive **accuracy** (54.7% of predictions are correct, as compared to 53.5% of predictions made by the model including the interaction.)
- a **stronger** (but still weak) performance in terms of **sensitivity** (if the subject actually has a statin prescription, the model without interaction detects this 32.2% of the time, as compared to 25.3% of the time for the model with interaction.)
- a **weaker** performance in terms of **specificity** (if the subject actually doesn't have a statin prescription, the model without interaction gets this right 85.4% of the time, as compared to 91.8% of the time for the model with interaction.)
- a **weaker positive predictive value** (our predictions from the model without the interaction that a subject has a statin prescription are correct 75% of the time, while for the model with the interaction such predictions are correct 80.8% of the time.)
- a **stronger negative preditive value** (our predictions from the model without the interaction that a subject does not have a statin prescription are correct 48% of the time, while for the model with the interaction such predictions are correct 47.4% of the time.)

Depending on which of these summaries were more important to you, you could wind up favoring either model here, on the basis of these confusion matrices.

Question 3: Rubric (10 points)

- Award 5 points for correctly creating the two confusion matrices, which may look a little different if they failed to deal with the missingness before fitting the models.
- Award 5 points for correctly comparing each of the five requested summary characteristics given their confusion matrices, and associating the correct direction (stronger/weaker) with each.

Question 4 (10 points)

Based on your general assessment of each model's quality of fit, select the model (interaction or no interaction) that seems more appropriate, and justify that selection.

Assessing Fit Quality with AIC and BIC

We will use the `glance` function to evaluate the quality of fit for our models.

```r
bind_rows(glance(model_without), glance(model_with)) %>%
  mutate(model= c("Without interaction", "With interaction"),
         deviance_diff= null.deviance - deviance,
         df_diff = df.null - df.residual) %>%
  select(model, AIC, BIC, deviance_diff, df_diff) %>%
  kable(digits = 1)
```

model	AIC	BIC	deviance_diff	df_diff
Without interaction	551.4	571.4	9.1	4
With interaction	539.1	571.1	27.4	7

- The model with interaction has smaller values for both AIC and BIC, so it looks like the more appropriate choice.
- Note that the confusion matrix summaries aren't really what we were looking for in this question, and present a somewhat mixed bag of results.

Question 4 (Rubric: 10 points)

- Award 5 points if the student generates appropriate quality of fit measures.
- Award 5 points for proper interpretation of the results.

Question 5 (10 points)

For the model you selected in Question 4, interpret the odds ratio associated with the `ldl` main effect carefully, specifying a 90% uncertainty interval and what we can conclude from the results.

```r
tidy(model_with, exponentiate = TRUE,
     conf.int=TRUE, conf.level=0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	63.517	0.880	16.050	294.823
ldl	0.964	0.009	0.950	0.978
practiceB	0.027	1.058	0.004	0.144
practiceC	0.011	1.140	0.002	0.066
practiceD	0.023	1.036	0.004	0.118
ldl:practiceB	1.035	0.010	1.018	1.053
ldl:practiceC	1.042	0.011	1.024	1.062
ldl:practiceD	1.035	0.010	1.018	1.053

The `ldl` odds ratio is estimated to be 0.964, with 90% uncertainty interval (0.950, 0.978). This means that if we have two patients named Harry and Sally who are seen at practice A, where Harry's LDL cholesterol is 1 point higher than Sally's, then the odds of Harry having a statin prescription are 96.4% as high as the odds for Sally. Since the 90% confidence interval is below 1, Harry's odds are detectably smaller (with 90% confidence) than Sally's.

Question 5 (Rubric: 10 points)

- Award 5 points if the student generates the appropriate odds ratios and uncertainty intervals.
- Award 3 points for proper interpretation of the `ldl` value but not recognizing that this only applies if the practice is A (so 3 points if they suggest only that their version of Harry and Sally need to be at the same practice.)
- Award the final 2 points if they recognize that the main effect of `ldl` only applies to practice A in this interaction model.
- If they mistakenly chose the model without the interaction in Question 4, then their answer should reflect that choice.

Question 6 (30 points)

- First, in 2-4 complete English sentences, please specify, using your own words and complete English sentences, the most useful and relevant piece of advice you took away from reading the chapters in David Spiegelhalter's **The Art of Statistics** that you have read so far.
  - Please provide a reference to the section of the book that provides this good advice.
- Then, in an essay of 4-8 additional sentences, describe why this particular piece of advice was meaningful or useful for you, personally, and how it will affect the way you move forward.
  - You are strongly encouraged to provide a specific example of a past or current scientific experience of yours that would have been (or is being) helped by this new approach or idea.
  - After reading your work, we want to be able to easily specify what this idea is, and why it is important and worth sharing.

We don't write sketches for essay questions. We hope to share a few of the more interesting responses with you after they've been graded.

Question 6 (Rubric: 30 points)

- Award up to 12 points for the initial little essay, giving full credit if they write down an actual piece of advice that makes sense to you, assuming they provide a clear indication of where it came from.
  - A reasonable piece of advice with no citation should get 9/12 on this part.
- Award up to 18 additional points for the second little essay, awarding 14-15 points for most students who do this in a reasonable way, but 17-18 points for the top 5 or so essays overall.
- Provide comments to all students who score below 24/30 here for reasons other than just typos or grammatical issues.

Session Information

```r
sessioninfo::session_info()
```

```
- Session info ---------------------------------------------------------------
 setting  value
 version  R version 3.6.2 (2019-12-12)
 os       Windows 10 x64
 system   x86_64, mingw32
 ui       RTerm
 language (EN)
 collate  English_United States.1252
 ctype    English_United States.1252
 tz       America/New_York
 date     2020-02-19

- Packages -------------------------------------------------------------------
 package      * version    date       lib source
 assertthat     0.2.1      2019-03-21 [1] CRAN (R 3.6.1)
 backports      1.1.5      2019-10-02 [1] CRAN (R 3.6.1)
 base64enc      0.1-3      2015-07-28 [1] CRAN (R 3.6.0)
 broom        * 0.5.4      2020-01-27 [1] CRAN (R 3.6.2)
 caret        * 6.0-85     2020-01-07 [1] CRAN (R 3.6.2)
 cellranger     1.1.0      2016-07-27 [1] CRAN (R 3.6.1)
 class          7.3-15     2019-01-01 [2] CRAN (R 3.6.2)
 cli            2.0.1      2020-01-08 [1] CRAN (R 3.6.2)
 codetools      0.2-16     2018-12-24 [2] CRAN (R 3.6.2)
 colorspace     1.4-1      2019-03-18 [1] CRAN (R 3.6.1)
 crayon         1.3.4      2017-09-16 [1] CRAN (R 3.6.1)
 data.table     1.12.8     2019-12-09 [1] CRAN (R 3.6.1)
 DBI            1.1.0      2019-12-15 [1] CRAN (R 3.6.1)
 dbplyr         1.4.2      2019-06-17 [1] CRAN (R 3.6.1)
 digest         0.6.23     2019-11-23 [1] CRAN (R 3.6.1)
 dplyr        * 0.8.3      2019-07-04 [1] CRAN (R 3.6.1)
 e1071          1.7-3      2019-11-26 [1] CRAN (R 3.6.1)
 ellipsis       0.3.0      2019-09-20 [1] CRAN (R 3.6.1)
 evaluate       0.14       2019-05-28 [1] CRAN (R 3.6.1)
 fansi          0.4.1      2020-01-08 [1] CRAN (R 3.6.2)
 farver         2.0.3      2020-01-16 [1] CRAN (R 3.6.2)
 forcats      * 0.4.0      2019-02-17 [1] CRAN (R 3.6.1)
 foreach        1.4.7      2019-07-27 [1] CRAN (R 3.6.2)
 fs             1.3.1      2019-05-06 [1] CRAN (R 3.6.1)
 generics       0.0.2      2018-11-29 [1] CRAN (R 3.6.1)
 ggplot2      * 3.2.1      2019-08-10 [1] CRAN (R 3.6.1)
 glue           1.3.1      2019-03-12 [1] CRAN (R 3.6.2)
 gower          0.2.1      2019-05-14 [1] CRAN (R 3.6.1)
 gtable         0.3.0      2019-03-25 [1] CRAN (R 3.6.1)
 haven          2.2.0      2019-11-08 [1] CRAN (R 3.6.1)
 here         * 0.1        2017-05-28 [1] CRAN (R 3.6.2)
 highr          0.8        2019-03-20 [1] CRAN (R 3.6.1)
 hms            0.5.3      2020-01-08 [1] CRAN (R 3.6.2)
 htmltools      0.4.0      2019-10-04 [1] CRAN (R 3.6.1)
 httr           1.4.1      2019-08-05 [1] CRAN (R 3.6.1)
 ipred          0.9-9      2019-04-28 [1] CRAN (R 3.6.2)
 iterators      1.0.12     2019-07-26 [1] CRAN (R 3.6.2)
 janitor      * 1.2.1      2020-01-22 [1] CRAN (R 3.6.2)
 jsonlite       1.6        2018-12-07 [1] CRAN (R 3.6.1)
 knitr        * 1.27       2020-01-16 [1] CRAN (R 3.6.2)
 labeling       0.3        2014-08-23 [1] CRAN (R 3.6.0)
 lattice      * 0.20-38    2018-11-04 [1] CRAN (R 3.6.2)
 lava           1.6.6      2019-08-01 [1] CRAN (R 3.6.2)
 lazyeval       0.2.2      2019-03-15 [1] CRAN (R 3.6.1)
 leaps        * 3.1        2020-01-16 [1] CRAN (R 3.6.2)
 lifecycle      0.1.0      2019-08-01 [1] CRAN (R 3.6.1)
 lubridate      1.7.4      2018-04-11 [1] CRAN (R 3.6.1)
 magrittr     * 1.5        2014-11-22 [1] CRAN (R 3.6.2)
 MASS           7.3-51.5   2019-12-20 [1] CRAN (R 3.6.2)
 Matrix         1.2-18     2019-11-27 [2] CRAN (R 3.6.2)
 ModelMetrics   1.2.2.1    2020-01-13 [1] CRAN (R 3.6.2)
 modelr         0.1.5      2019-08-08 [1] CRAN (R 3.6.1)
 munsell        0.5.0      2018-06-12 [1] CRAN (R 3.6.1)
 naniar       * 0.4.2      2019-02-15 [1] CRAN (R 3.6.1)
 nlme           3.1-142    2019-11-07 [2] CRAN (R 3.6.2)
 nnet           7.3-12     2016-02-02 [2] CRAN (R 3.6.2)
 patchwork    * 1.0.0      2019-12-01 [1] CRAN (R 3.6.1)
 pillar         1.4.3      2019-12-20 [1] CRAN (R 3.6.2)
 pkgconfig      2.0.3      2019-09-22 [1] CRAN (R 3.6.1)
 plyr           1.8.5      2019-12-10 [1] CRAN (R 3.6.1)
 pROC           1.16.1     2020-01-14 [1] CRAN (R 3.6.2)
 prodlim        2019.11.13 2019-11-17 [1] CRAN (R 3.6.2)
 purrr        * 0.3.3      2019-10-18 [1] CRAN (R 3.6.1)
 R6             2.4.1      2019-11-12 [1] CRAN (R 3.6.1)
 Rcpp           1.0.3      2019-11-08 [1] CRAN (R 3.6.1)
 readr        * 1.3.1      2018-12-21 [1] CRAN (R 3.6.1)
 readxl         1.3.1      2019-03-13 [1] CRAN (R 3.6.1)
 recipes        0.1.9      2020-01-07 [1] CRAN (R 3.6.2)
 repr           1.1.0      2020-01-28 [1] CRAN (R 3.6.2)
 reprex         0.3.0      2019-05-16 [1] CRAN (R 3.6.1)
 reshape2       1.4.3      2017-12-11 [1] CRAN (R 3.6.1)
 rlang          0.4.4      2020-01-28 [1] CRAN (R 3.6.2)
 rmarkdown      2.1        2020-01-20 [1] CRAN (R 3.6.2)
 rpart          4.1-15     2019-04-12 [2] CRAN (R 3.6.2)
 rprojroot      1.3-2      2018-01-03 [1] CRAN (R 3.6.1)
 rstudioapi     0.10       2019-03-19 [1] CRAN (R 3.6.1)
 rvest          0.3.5      2019-11-08 [1] CRAN (R 3.6.1)
 scales         1.1.0      2019-11-18 [1] CRAN (R 3.6.1)
 sessioninfo    1.1.1      2018-11-05 [1] CRAN (R 3.6.1)
 skimr        * 2.0.2      2019-11-26 [1] CRAN (R 3.6.2)
 snakecase      0.11.0     2019-05-25 [1] CRAN (R 3.6.1)
 stringi        1.4.5      2020-01-11 [1] CRAN (R 3.6.2)
 stringr      * 1.4.0      2019-02-10 [1] CRAN (R 3.6.1)
 survival       3.1-8      2019-12-03 [1] CRAN (R 3.6.2)
 tibble       * 2.1.3      2019-06-06 [1] CRAN (R 3.6.1)
 tidyr        * 1.0.2      2020-01-24 [1] CRAN (R 3.6.2)
 tidyselect     1.0.0      2020-01-27 [1] CRAN (R 3.6.2)
 tidyverse    * 1.3.0      2019-11-21 [1] CRAN (R 3.6.2)
 timeDate       3043.102   2018-02-21 [1] CRAN (R 3.6.2)
 utf8           1.1.4      2018-05-24 [1] CRAN (R 3.6.1)
 vctrs          0.2.2      2020-01-24 [1] CRAN (R 3.6.2)
 visdat         0.5.3      2019-02-15 [1] CRAN (R 3.6.1)
 withr          2.1.2      2018-03-15 [1] CRAN (R 3.6.1)
 xfun           0.12       2020-01-13 [1] CRAN (R 3.6.2)
 xml2           1.2.2      2019-08-09 [1] CRAN (R 3.6.1)
 yaml           2.2.0      2018-07-25 [1] CRAN (R 3.6.0)

[1] C:/Users/Thomas/Documents/R/win-library/3.6
[2] C:/Program Files/R/R-3.6.2/library
```