500 Homework 1 Answer Sketch

Thomas E. Love

due 2018-01-25 (version: 2018-01-22)

Contents

Setup and Task 1

```
library(skimr)
library(broom)
library(simputation)
library(tidyverse)
```

```
-- Attaching packages ---------------------------------------------------------------------------
```

```
v ggplot2 2.2.1     v purrr   0.2.4
v tibble  1.4.1     v dplyr   0.7.4
v tidyr   0.7.2     v stringr 1.2.0
v readr   1.1.1     v forcats 0.2.0
```

```
-- Conflicts ------------------------------------------------------------------------------------
x dplyr::contains()    masks skimr::contains()
x dplyr::ends_with()   masks skimr::ends_with()
x dplyr::everything()  masks skimr::everything()
x dplyr::filter()      masks stats::filter()
x dplyr::lag()         masks stats::lag()
x dplyr::matches()     masks skimr::matches()
x dplyr::num_range()   masks skimr::num_range()
x dplyr::one_of()      masks skimr::one_of()
x dplyr::starts_with() masks skimr::starts_with()
```

```
dig1 <- read.csv("dig1.csv") %>% tbl_df
```

Task 1 requires you to request the DIG data.

Task 2

In task 2 you were asked to build a mock proposal for a DIG observational study. We'll discuss those in class, and so I have no real sketch here. I expect that some of Rosenbaum's writing will be of help. The questions you needed to answer were:

1. What comparison do you want to make? (Select a comparison different than the one made in the original DIG paper)
   - Did patients receiving "EXPOSURE A" have lower rates of "BAD OUTCOME" than those who received "EXPOSURE B"?
2. Why is this of interest?
   - "OUTCOME" is important because ...
   - "EXPOSURE" (A or B) is important because ...
   - (*Be sure to clearly indicate what you hypothesize the effect of EXPOSURE on OUTCOME to be.*)
3. What are the key measures - specifically, the exposure/treatment, the primary outcome, and important covariates that are available in the data to help address your question of interest?
   - Exposure/Treatment = A or B, and be sure to specify the way in which you will know which exposure someone receives, and whether the exposure / treatment is applied using a randomized approach, or not.
   - Outcome = ..., and be sure to specify the variables you will use to determine the outcome, as well as the *type* of outcome, be it continuous, categorical (and if categorical, binary or multi-categorical) or survival (and if survival, is censoring involved?)
   - Covariates of interest: We'd be interested in anything related to treatment choice or to outcome. You should provide a list of such variables of interest. Remember to include **ONLY** things which are measured prior to the exposure/treatment of interest, or which are not possibly changed by it.

Task 3

The Problem

Here, you were to build and evaluate a logistic regression model using the DIG data.

Your model should be fitted to a random training sample of 5,000 subjects (be sure to specify the seed you used to select that sample) and then tested on the remaining 1,800 subjects, but you'll probably want to check for and deal with missingness in the entire sample before splitting into training and test groups. Your model will predict the probability that a subject in the study will die, based on:

- the subject's assigned treatment (digoxin or placebo),
- the subject's age at randomization,
- race,
- sex,
- ejection fraction (percent),
- calculated body mass index,
- NYHA functional class, and
- whether or not the subject currently has angina.

The relevant variables in the `dig1.csv` data set are therefore: `subjectid`, `DEATH`, `TRTMT`, `AGE`, `RACE`, `SEX`, `EJF_PER`, `BMI`, `FUNCTCLS`, and `ANGINA`.

Be sure to treat the categorical variables (including NYHA class, angina status, race and sex) appropriately as factors (ideally with meaningful names), and account for missingness deliberately in an appropriate way.

Your final results should include:

1. a R Markdown file containing all of your code

2. an HTML file with the results from your Markdown, which describes:
    1. your sample preparation work including dealing with missingness and partitioning the data into training and test samples
    2. your fitted logistic regression model (to your training sample)
    3. the results of your application of your model to your test sample, which is best accomplished as a graph which shows the distribution of your model probability estimates in the "actually died" and "actually survived" groups within your test sample.

Preparing the Sample for Modeling

```
dig_hw1 <- dig1 %>%
  mutate(subject = as.character(subjectid),
         nyha_f = factor(FUNCTCLS),
         angina = ANGINA,
         female = SEX - 1,
         race_f = fct_recode(factor(RACE), White = "1", Nonwhite = "2"),
         tx_f = fct_recode(factor(TRTMT), Placebo = "0", Treatment = "1"),
         death_f = fct_recode(factor(DEATH), Died = "1", Survived = "0")) %>%
  select(subject, death_f, tx_f, AGE, race_f, female, EJF_PER, BMI, nyha_f, angina, FUNCTCLS)

skim(select(dig_hw1, -subject))
```

```
Skim summary statistics
 n obs: 6800
 n variables: 10

Variable type: factor
 variable missing complete    n n_unique                       top_counts
  death_f       0     6800 6800        2      Sur: 4425, Die: 2375, NA: 0
   nyha_f       6     6794 6800        4 2: 3664, 3: 2081, 1: 907, 4: 142
   race_f       0     6800 6800        2       Whi: 5809, Non: 991, NA: 0
     tx_f       0     6800 6800        2      Pla: 3403, Tre: 3397, NA: 0
 ordered
   FALSE
   FALSE
   FALSE
   FALSE

Variable type: integer
 variable missing complete    n  mean    sd p0 p25 median p75 p100
      AGE       0     6800 6800 63.48 10.92 21  57     65  71   94
   angina       2     6798 6800  0.27  0.44  0   0      0   1    1
  EJF_PER       0     6800 6800 28.54  8.85  3  22     29  35   45
 FUNCTCLS       6     6794 6800  2.21  0.69  1   2      2   3    4
     hist
 <U+2581><U+2581><U+2582><U+2583><U+2587><U+2587><U+2583><U+2581>
 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2583>
 <U+2581><U+2582><U+2585><U+2587><U+2587><U+2587><U+2586><U+2586>
 <U+2582><U+2581><U+2587><U+2581><U+2581><U+2585><U+2581><U+2581>

Variable type: numeric
 variable missing complete    n  mean   sd    p0   p25 median  p75  p100
      BMI       1     6799 6800 27.11 5.19 14.45 23.68   26.5 29.8 62.66
```

```
   female       0     6800 6800  0.22 0.42  0     0       0    0    1
     hist
 <U+2581><U+2587><U+2587><U+2582><U+2581><U+2581><U+2581><U+2581>
 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2582>
```

There are two subjects missing `angina_f`, 6 missing `nyha_f` (and `FUNCTCLS`) and 1 missing BMI.

- With so few missing values, a completely reasonable strategy would be to simply omit the missing data before splitting into training and test samples.
- A simple imputation would also work in this setting, I suppose, but I won't bother for now. I will come back to this later, and that's the reason I'm keeping the integer FUNCTCLS along with the factor `nyha_f`.

```
dig_hw1_noNA <- dig_hw1 %>% drop_na()

set.seed(20180125)
dig_hw1_train <- sample_n(dig_hw1_noNA, size = 5000)
dig_hw1_test <- anti_join(dig_hw1_noNA, dig_hw1_train)
```

```
Joining, by = c("subject", "death_f", "tx_f", "AGE", "race_f", "female", "EJF_PER", "BMI", "nyha_f", "an
```

Fitting a Logistic Regression Model to the training sample

```
model1 <- glm(death_f ~ tx_f + AGE + race_f + female + EJF_PER +
                BMI + nyha_f + angina,
              family = binomial(link = "logit"),
              data = dig_hw1_train)

summary(model1)
```

```
Call:
glm(formula = death_f ~ tx_f + AGE + race_f + female + EJF_PER +
    BMI + nyha_f + angina, family = binomial(link = "logit"),
    data = dig_hw1_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5253  -0.9103  -0.7385   1.2566   2.0566

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.0642269  0.2831803   0.227  0.82057
tx_fTreatment   0.0104649  0.0613321   0.171  0.86452
AGE             0.0002092  0.0028137   0.074  0.94073
race_fNonwhite -0.0033702  0.0860726  -0.039  0.96877
female         -0.2166571  0.0755308  -2.868  0.00412 **
EJF_PER        -0.0358246  0.0035532 -10.082  < 2e-16 ***
BMI            -0.0050209  0.0059280  -0.847  0.39701
nyha_f2         0.3023445  0.1025639   2.948  0.00320 **
nyha_f3         0.8981808  0.1077956   8.332  < 2e-16 ***
nyha_f4         1.3017880  0.2226787   5.846 5.03e-09 ***
angina         -0.0714840  0.0695653  -1.028  0.30415
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

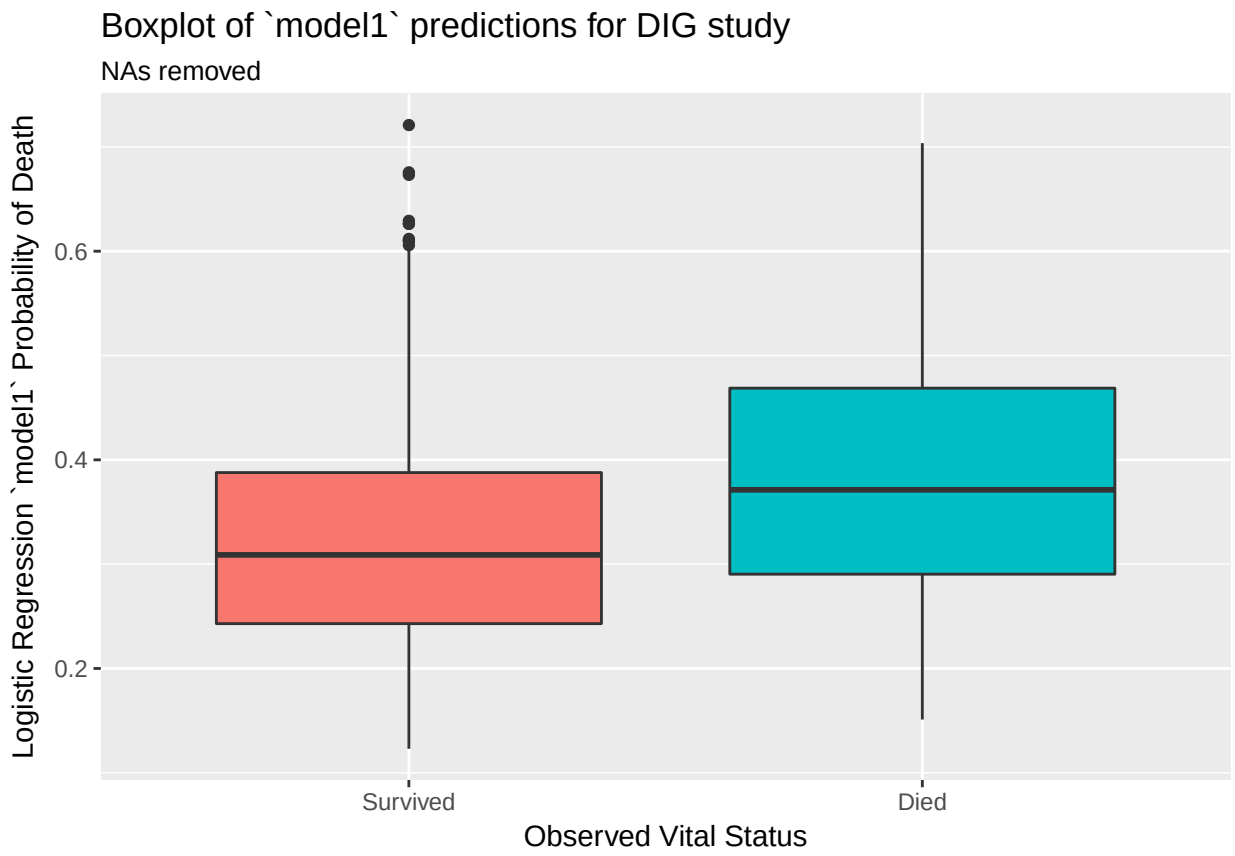
```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6441.7  on 4999  degrees of freedom
Residual deviance: 6154.4  on 4989  degrees of freedom
AIC: 6176.4

Number of Fisher Scoring iterations: 4
```

```r
glance(model1)
```

```
  null.deviance df.null    logLik      AIC      BIC deviance df.residual
1      6441.681    4999 -3077.188 6176.377 6248.066 6154.377        4989
```

Applying the model to a test sample, and producing a graph

```r
dig_hw1_test$.fit <- predict(model1, newdata = dig_hw1_test, type = "response")

ggplot(dig_hw1_test, aes(x = death_f, y = .fit, fill = death_f)) +
  geom_boxplot() +
  guides(fill = FALSE) +
  labs(y = "Logistic Regression `model1` Probability of Death", x = "Observed Vital Status",
      title = "Boxplot of `model1` predictions for DIG study",
      subtitle = "NAs removed")
```



What if we did a simple imputation instead?

We had some missing values earlier. What if instead of removing them, we imputed them? I'll show you a simple imputation approach, making use of the `simputation` package, which is a good tool for simple imputation, and has a nice vignette here.

As mentioned earlier, the `dig_hw1` data set has some missing values.

```
colSums(is.na(dig_hw1))
```

```
 subject  death_f     tx_f      AGE   race_f   female  EJF_PER      BMI
       0        0        0        0        0        0        0        1
  nyha_f   angina FUNCTCLS
       6        2        6
```

It's easier to impute the multi-categorical variable contained in `FUNCTCLS` (as a number) and in `nyha_f` (as a factor) in its numeric form, so we'll do that, then recreate `nyha_f` from the imputed `FUNCTCLS`.

```
set.seed(500)
dig_imp <- dig_hw1 %>%
  impute_pmm(FUNCTCLS ~ EJF_PER) %>%
  impute_lm(BMI ~ AGE + race_f + female) %>%
  impute_pmm(angina ~ EJF_PER) %>%
  mutate(nyha_f = factor(FUNCTCLS))

colSums(is.na(dig_imp))
```

```
 subject  death_f     tx_f      AGE   race_f   female  EJF_PER      BMI
       0        0        0        0        0        0        0        0
  nyha_f   angina FUNCTCLS
       0        0        0
```

```
set.seed(20180125)
dig_imp_train <- sample_n(dig_imp, size = 5000)
dig_imp_test <- anti_join(dig_imp, dig_imp_train)
```

```
Joining, by = c("subject", "death_f", "tx_f", "AGE", "race_f", "female", "EJF_PER", "BMI", "nyha_f", "a
```

```
colSums(is.na(dig_imp_train))
```

```
 subject  death_f     tx_f      AGE   race_f   female  EJF_PER      BMI
       0        0        0        0        0        0        0        0
  nyha_f   angina FUNCTCLS
       0        0        0
```

and now we can follow the earlier commands to fit the logistic regression model in the training set, and then assess its results in the test set.

```
model2 <- glm(death_f ~ tx_f + AGE + race_f + female + EJF_PER +
                BMI + nyha_f + angina,
              family = binomial(link = "logit"),
              data = dig_imp_train)

summary(model2)
```

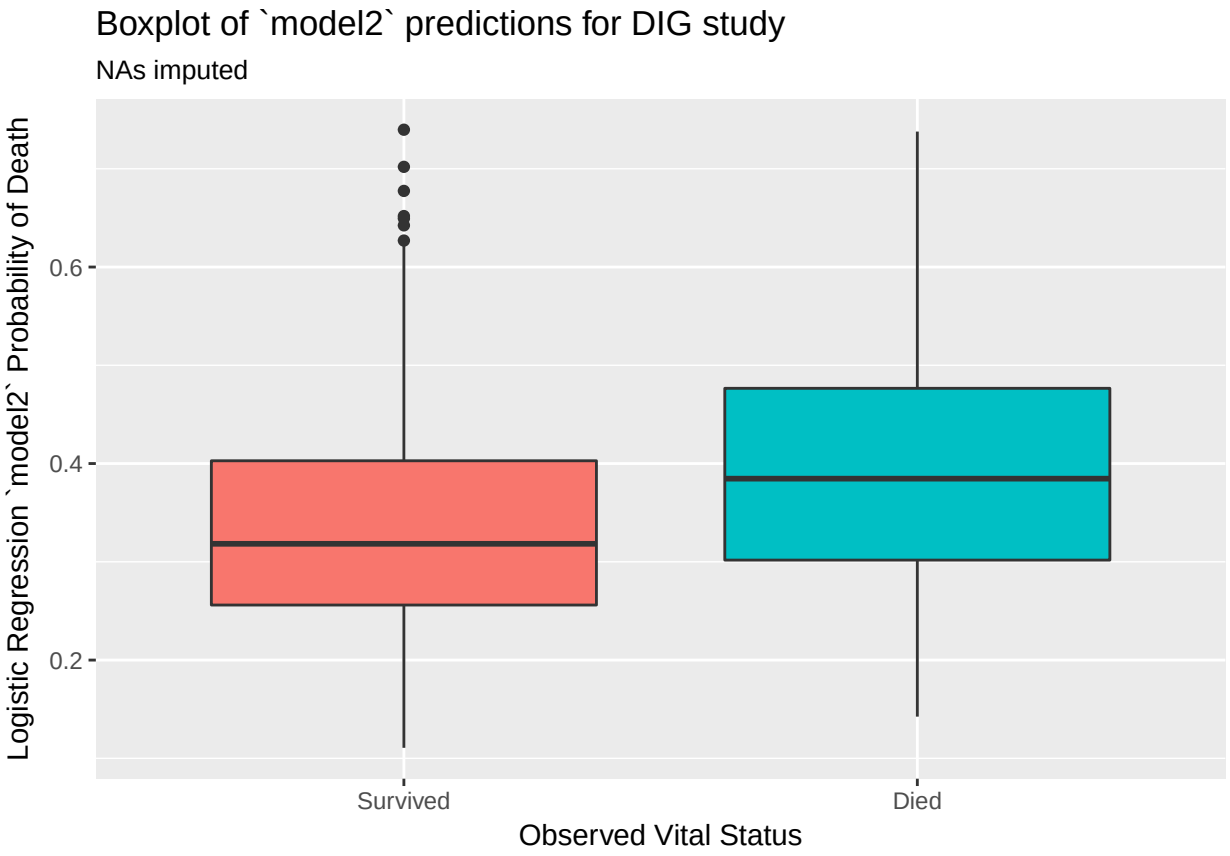
```
Call:
glm(formula = death_f ~ tx_f + AGE + race_f + female + EJF_PER +
    BMI + nyha_f + angina, family = binomial(link = "logit"),
```

```
    data = dig_imp_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5631  -0.9205  -0.7446   1.2629   2.0897

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.087715   0.282130   0.311 0.755876
tx_fTreatment  -0.024932   0.061197  -0.407 0.683703
AGE            -0.001798   0.002812  -0.639 0.522562
race_fNonwhite -0.051116   0.087306  -0.585 0.558223
female         -0.287127   0.075752  -3.790 0.000150 ***
EJF_PER        -0.034961   0.003566  -9.804  < 2e-16 ***
BMI            -0.001815   0.005961  -0.304 0.760802
nyha_f2         0.380598   0.100796   3.776 0.000159 ***
nyha_f3         0.940238   0.106411   8.836  < 2e-16 ***
nyha_f4         1.431948   0.233472   6.133 8.61e-10 ***
angina         -0.078933   0.069197  -1.141 0.253999
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6464.5  on 4999  degrees of freedom
Residual deviance: 6177.6  on 4989  degrees of freedom
AIC: 6199.6

Number of Fisher Scoring iterations: 4
```

```
glance(model2)
```

```
  null.deviance df.null    logLik      AIC      BIC deviance df.residual
1      6464.505    4999 -3088.781 6199.563 6271.252 6177.563        4989
```

```
dig_imp_test$.fit2 <- predict(model2, newdata = dig_imp_test, type = "response")

ggplot(dig_imp_test, aes(x = death_f, y = .fit2, fill = death_f)) +
  geom_boxplot() +
  guides(fill = FALSE) +
  labs(y = "Logistic Regression `model2` Probability of Death", x = "Observed Vital Status",
      title = "Boxplot of `model2` predictions for DIG study",
      subtitle = "NAs imputed")
```


Boxplot of `model2` predictions for DIG study
NAs imputed
Logistic Regression `model2` Probability of Death
0.6
0.4
0.2
Survived
Died
Observed Vital Status