432 Homework 4 Answer Sketch and Grading Rubric

432 TAs

Due 2020-03-25. Version: 2020-03-20

Contents

Setup and Data Ingest

```
knitr::opts_chunk$set(comment = NA)

library(here); library(janitor); library(magrittr)

library(mice)
library(naniar)
library(caret)
library(simputation)
library(car)
library(rms)
library(broom)
library(knitr)

library(tidyverse)
```

```
phr <- read_csv(here("data", "phr.csv")) %>%
    clean_names()
```

Question 1 (5 points)

Create a `phr1` tibble which includes

- only those patients who have data on systolic blood pressure, and
- only the following 9 variables: `id`, `sbp`, `age`, `female`, `caucasian`, `insurance`, `hba1c`, `ldl`, and `bmi`.

In addition to annotating your R code, specify (in a sentence) how the size of your tibble changes from the original `phr` to this new `phr1`.

For this task, you will use the `filter` and `select` functions to generate the required data set `phr1`. After selecting the requested variables and filtering, we have 3 fewer rows (observations) and 11 fewer columns (variables).

```
phr1 <- phr %>% filter(complete.cases(sbp)) %>%
    select(id, sbp, age, female, caucasian,
           insurance, hba1c, ldl, bmi)

dim(phr)
```

```
[1] 10746    20
```

```
dim(phr1)
```

```
[1] 10743     9
```

Question 1 Rubric (5 points):

- Give 5 points for correctly doing all of the following
  - Loading the data set
  - Creating the `phr1` data set
  - Stating the difference in size between the two data sets
- If the student does something incorrectly that will affect their answers for future questions, they should only be penalized here and not for those questions as long as their answers are correct given these errors.

Question 2 (5 points)

Use the `caret` package to help you accomplish a validation split, to help you build up a model for `sbp` in the `phr1` data. Specifically, split the `phr1` tibble into a training sample containing 75% of the data, and a test sample containing the remaining 25%. Use the number `2020` as your seed for random number generation.

After setting the seed to `2020`, we will use the `createDataPartition` function to create a data frame containing 75% of the observations. The rest of the observations will be used to create the test sample.

```
set.seed(2020)

training.samples <- phr1$sbp %>%
    createDataPartition(p = 0.75, list = FALSE)

phr1_train <- phr1[training.samples,]
phr1_test <- phr1[-training.samples,]
```

Question 2 Rubric (5 points):

- Give 5 points for correctly doing all of the following
  - Choosing the correct seed
  - Creating a training sample containing 75% of the data
  - Creating a test sample containing 25% of the data
- If the student does something incorrectly that will affect their answers for future questions, they should only be penalized here and not for those questions as long as their answers are correct given these errors.

Question 3 (5 points)

Next, you will build a regression model to predict a patient's systolic blood pressure on the basis of seven predictors, specifically the patient's hemoglobin A1c, LDL cholesterol, body-mass index, age, sex, race and insurance status. How many observations are missing for each of the predictors in your training sample?

For this question, we are just checking for missing variables. There are many ways to do this, but we will use the `miss_var_summary` function.

```
phr1_train %>% select(hba1c, ldl, bmi, age,
                      female, caucasian, insurance) %>%
    miss_var_summary(.)
```

```
# A tibble: 7 x 3
  variable  n_miss pct_miss
  <chr>      <int>    <dbl>
1 ldl          797     9.89
2 bmi          450     5.58
3 hba1c        418     5.19
4 age            0     0
5 female         0     0
6 caucasian      0     0
7 insurance      0     0
```

Question 3 Rubric (5 points):

- Give full credit if the student checked missingness in all variables.
- Deduct 1 point for each variable that was missed.

Question 4 (10 points)

We'd like to use a Box-Cox procedure to evaluate whether a transformation of the outcome is necessary, within your training sample, but to accomplish this, you'll need to do something about those missing values.

So use simple imputation with a robust linear model including the other 6 predictors and the systolic blood pressure to predict any missing values you identified in Question 3. Verify that there is no missing data remaining after you do this imputation.

To perform a simple imputation with a robust linear model, you will have to use the `impute_rlm` function from the `simputation` package.

```
phr1_train_imp <- phr1_train %>%
    impute_rlm(., ldl + bmi + hba1c ~
                   age + female + caucasian + insurance + sbp)

n_case_complete(phr1_train_imp) # sanity check
```

```
[1] 8058
```

```
n_case_miss(phr1_train_imp) # sanity check
```

```
[1] 0
```

After a sanity check, we are sure that all the variables now have no missing values.

Question 4 Rubric (10 points):

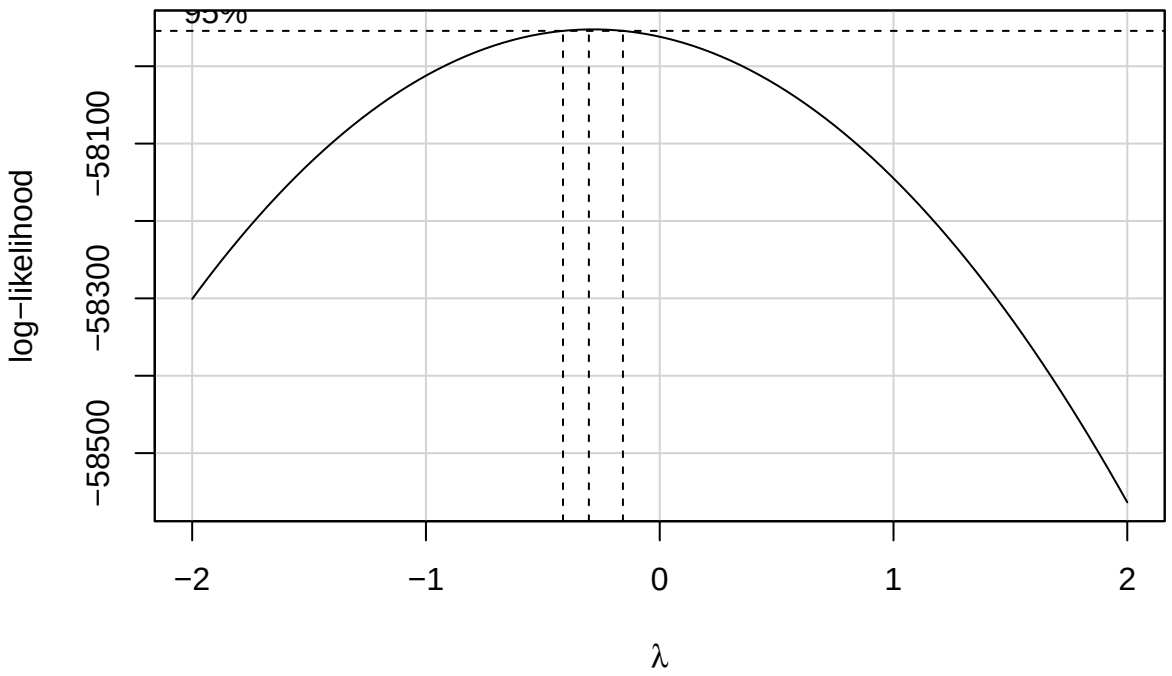
- Give full credit if the student performed simple imputation with a robust linear model approach.
- Deduct 2 points if the student performed imputation using a different method or didn't include the required variables for imputation.

Question 5 (10 points)

Generate appropriate Box-Cox procedure results for assessing potential transformations of the outcome using the model described in Question 3, and describe your conclusions. Please restrict yourself to consideration of only five possible transformations: (a) squaring the outcome, (b) the untransformed outcome, (c) the square root of the outcome, (d) the natural logarithm of the outcome, or (e) the inverse of the outcome.

The model in Question 3 asks you to predict `sbp` based on `hba1c`, `bmi`, `ldl`, `age`, `female`, `caucasian`, and `insurance`. This is the Box-Cox Plot result for this model.

```
phr1_train_imp %$%
    boxCox(sbp ~ hba1c + bmi + ldl + age + female +
               caucasian + insurance)
```



We will use a logarithm here, as the suggested result is far closer to 0 than to -1.

Question 5 Rubric (10 points):

- Give 6 points if the students generates the appropriate Box-Cox plot with the correct model.
- Give 4 points if the student interprets the plot appropriately and chooses the correct outcome transformation.

Question 6 (10 points)

The planned model for the outcome you decided on in Question 5 includes seven predictors, all of which will be included in your model as main effects. Run this "main effects" model in your training sample using the `lm` function, and obtain the nominal R-squared, adjusted R-squared, AIC and BIC results.

We will run a linear regression model after transforming our outcome variable `sbp` and using the main effects of the 7 predictors we chose earlier.

Using `glance` is an easy way to get the R-squared, AIC, and BIC results.

```
mod_main <- phr1_train %$%
    lm(log(sbp) ~ hba1c + bmi + ldl + age +
           female + caucasian + insurance)

glance(mod_main) %>%
    mutate(model = "Main Effects") %>%
    select(model, r.squared, adj.r.squared, AIC, BIC, df) %>%
    kable(digits = c(0, 4, 4, 1, 1, 0))
```

model	r.squared	adj.r.squared	AIC	BIC	df
Main Effects	0.0517	0.0508	-9966	-9904.6	8

Question 6 Rubric (10 points):

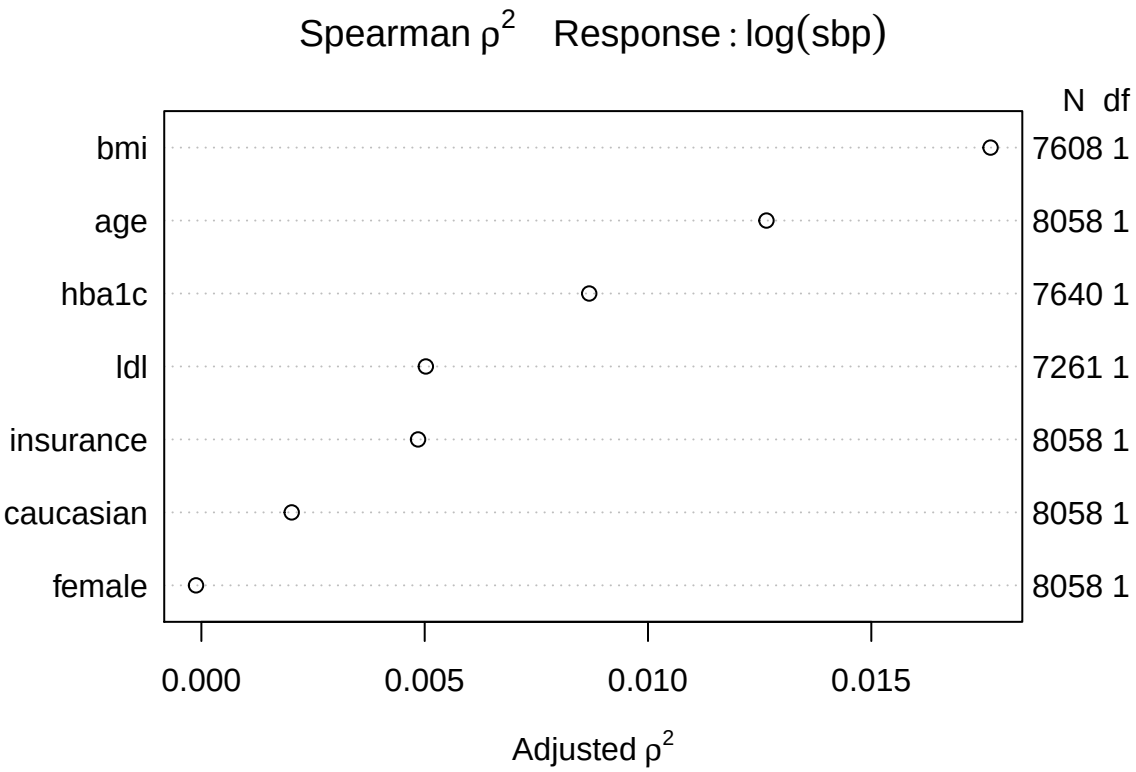
- Give 6 points for creating the appropriate model using the `lm` function.
- Give 1 point for obtaining each of the requested regression metrics.

Question 7 (10 points)

Use an appropriate strategy to determine how best to spend exactly 5 additional degrees of freedom (beyond the main effects model) on exactly two terms involving restricted cubic splines. Describe your conclusions and reasoning carefully.

We will use the the Spearman ρ^2 plot to figure out how to incorporate non-linear predictor terms in the model.

```
phr1_train %$%
    plot(Hmisc::spearman2(log(sbp) ~ hba1c + bmi + ldl + age +
                          female + caucasian + insurance))
```



We'll spend 3 additional df (beyond what we used in the main effects model) using a restricted cubic spline with 5 knots in `bmi` and then 2 additional df using a restricted cubic spline with 4 knots in `age`.

Question 7 Rubric (10 points):

- Give 4 points for generating the Spearman ρ^2 plot.
- Give 3 points for correctly identifying the most important non-linear term as an RCS with 5 knots for `bmi`.
- Give 3 points for correctly identifying the most important non-linear term as an RCS with 4 knots for `age`.

Question 8 (10 points)

Fit the "augmented" model you selected in Question 7 again using the `lm` function, and compare the training sample's R-squared, adjusted R-squared, AIC and BIC results to what you saw in Question 6. Then run an ANOVA to compare the two models. What conclusions do you draw about which model to prefer, based on the training sample?

First, we will fit this new model then compare it with the main effects model.

```
mod_aug <- phr1_train %$%
    lm(log(sbp) ~ hba1c + rcs(bmi,5) + ldl + rcs(age, 4) +
           female + caucasian + insurance)

glance(mod_aug) %>%
    mutate(model = "Augmented") %>%
    select(model, r.squared, adj.r.squared, AIC, BIC, df) %>%
    kable(digits = c(0, 4, 4, 1, 1, 0))
```

model	r.squared	adj.r.squared	AIC	BIC	df
Augmented	0.0548	0.0531	-9977.8	-9882.3	13

The augmented model has slightly better results than the main effects model in terms of adjusted R^2, and AIC, but it displays slightly worse results in terms of BIC.

In terms of the ANOVA comparison, the augmented model does appear to provide additional highly detectable predictive value in the training sample, based on the very small p value below.

```
anova(mod_aug, mod_main)
```

```
Analysis of Variance Table

Model 1: log(sbp) ~ hba1c + rcs(bmi, 5) + ldl + rcs(age, 4) + female +
    caucasian + insurance
Model 2: log(sbp) ~ hba1c + bmi + ldl + age + female + caucasian + insurance
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   6759 90.482
2   6764 90.774 -5  -0.29172 4.3583 0.0005813 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 8 Rubric (10 points):

- Give 1 point for providing each of the regression metrics requested in the question.
- Give 3 points for a proper comparison between the models with and without non-linear terms.
- Give 1 point for correctly running an ANOVA comparing the two models.
- Give 2 points for properly interpreting the ANOVA output.

Question 9 (15 points)

Now use each model (the one from Question 6 and the one from Question 8) to make predictions in the test sample you developed way back in Question 2 and make a decision on the basis of the usual summary statistics (specifically the validation R-square, root mean squared prediction error and mean absolute prediction error) as to which model produces better predictions for systolic blood pressure.

First, drop all observations with missing values in the test sample.

```
phr1_test_cc <- phr1_test %>% drop_na()
```

Note that this drops quite a few rows from our test sample, but that's OK for this Homework.

```
dim(phr1_test)
```

```
[1] 2685    9
```

```
dim(phr1_test_cc)
```

```
[1] 2291    9
```

Main Effects Model

Obtain predicted `log(sbp)` in the test sample, using the main effects model `mod_main`

```
main_pred_logs <- mod_main %>% predict(phr1_test_cc)
```

Exponentiate to get predicted `sbp` values in the test sample.

```
main_pred_sbp <- exp(main_pred_logs)
```

Create key summaries of the quality of predictions we made for `sbp` using `mod_main`...

```
main_summaries <- tibble(
    model = "Main Effects",
    R2 = R2(main_pred_sbp, phr1_test_cc$sbp),
    RMSE = RMSE(main_pred_sbp, phr1_test_cc$sbp),
    MAE = MAE(main_pred_sbp, phr1_test_cc$sbp)
)
```

Augmented Model

Obtain predicted `log(sbp)` in the test sample, using the augmented model `mod_aug`

```
aug_pred_logs <- mod_aug %>% predict(phr1_test_cc)
```

Exponentiate to get predicted `sbp` values in the test sample.

```
aug_pred_sbp <- exp(aug_pred_logs)
```

Create key summaries of the quality of predictions we made for `sbp` using `mod_main`...

```
aug_summaries <- tibble(
    model = "Augmented",
    R2 = R2(aug_pred_sbp, phr1_test_cc$sbp),
    RMSE = RMSE(aug_pred_sbp, phr1_test_cc$sbp),
    MAE = MAE(aug_pred_sbp, phr1_test_cc$sbp)
)
```

And now we can produce the comparison

```
bind_rows(main_summaries, aug_summaries) %>%
    kable(digits = c(0, 4, 3, 3))
```

model	R2	RMSE	MAE
Main Effects	0.0522	15.536	11.612
Augmented	0.0514	15.524	11.629

The main effects model has the better R^2 and MAE, but worse RMSE than the augmented model. So we'll

go with the main effects model.

Question 9 Rubric (15 points):

- Give 4 points for creating appropriate predictions for the main effects model.
- Give 4 points for creating appropriate predictions for the augmented model.
- Give 4 points for generating the prediction quality metrics.
- Give 3 points for appropriately choosing the correct models based on the key summaries.

Question 10 (10 points)

Return to the `phr1` tibble you created back in Question 1 and fit the regression model you preferred in Question 9 to predict your (potentially transformed) `sbp` using all of the **complete cases** in that tibble.

- Provide an attractive table containing the coefficients, standard errors, and lower and upper bounds for a 95% confidence interval for each coefficient.

The code for this question is straightforward, just refitting the main effects model with the original `phr1` data. We then use `tidy` to produce our attractive table of model outputs.

```
mod_cc <- phr1 %$% lm(log(sbp) ~ hba1c + bmi + ldl + age +
           female + caucasian + insurance)

tidy(mod_cc, conf.int = TRUE, conf.level = 0.95) %>%
    select(term, estimate, std.error, conf.low, conf.high)
```

```
# A tibble: 8 x 5
  term         estimate std.error  conf.low conf.high
  <chr>           <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)  4.59     0.0151     4.56      4.62
2 hba1c        0.00644  0.000908   0.00466   0.00822
3 bmi          0.00254  0.000171   0.00221   0.00288
4 ldl          0.000295 0.0000399  0.000217  0.000373
5 age          0.00180  0.000151   0.00150   0.00209
6 female      -0.00812  0.00251   -0.0130   -0.00319
7 caucasian   -0.0165   0.00280   -0.0220   -0.0110
8 insurance   -0.00334  0.00302   -0.00926   0.00259
```

Question 10 Rubric (10 points):

- Give 8 points if student correctly creates uses the model selected in Question 9 and fit with the `phr1` data.
- Give 2 points for generating the table with the requested components of the model output, including the correct confidence interval.

Question 11 (15 points)

Return to the `phr1` tibble you created back in Question 1 and fit the regression model you preferred in Question 9, to predict your (potentially transformed) `sbp` but this time, use **multiple imputation**.

- Include all 6 other predictors and the outcome in the imputation model for each predictor, and set the random seed to 2020432 and run 10 imputations.
- As in Question 10, provide an attractive table containing the coefficients, standard errors, and lower and upper bounds for a 95% confidence interval for each coefficient after this model has been fit.

First, we create the 10 imputations.

```
set.seed(2020432)
phr1_mi10 <- mice(phr1, m = 10, print = FALSE)
```

Next we fit the main effects model on each imputed data frame.

```
mi_mods <- with(phr1_mi10,
                lm(log(sbp) ~ hba1c + bmi + ldl + age +
                       female + caucasian + insurance))
```

Then we pool the models.

```
mi_pool <- pool(mi_mods)
```

I'll summarize the two models in reasonably manageable tables.

First, we repeat the complete cases result.

```
tidy(mod_cc, conf.int = TRUE, conf.level = 0.95) %>%
    select(term, estimate, std.error, p.value, conf.low, conf.high) %>%
    kable(digits = 5)
```

term	estimate	std.error	p.value	conf.low	conf.high
(Intercept)	4.59288	0.01509	0.00000	4.56330	4.62246
hba1c	0.00644	0.00091	0.00000	0.00466	0.00822
bmi	0.00254	0.00017	0.00000	0.00221	0.00288
ldl	0.00030	0.00004	0.00000	0.00022	0.00037
age	0.00180	0.00015	0.00000	0.00150	0.00209
female	-0.00812	0.00251	0.00124	-0.01304	-0.00319
caucasian	-0.01651	0.00280	0.00000	-0.02199	-0.01103
insurance	-0.00334	0.00302	0.26964	-0.00926	0.00259

And now, the multiple imputation result.

```
summary(mi_pool, conf.int = TRUE, conf.level = 0.95) %>%
    select(-df, -statistic) %>% kable(digits = 5)
```

term	estimate	std.error	p.value	2.5 %	97.5 %
(Intercept)	4.58687	0.01374	0.00000	4.55993	4.61381
hba1c	0.00629	0.00085	0.00000	0.00463	0.00796
bmi	0.00244	0.00016	0.00000	0.00212	0.00276
ldl	0.00029	0.00004	0.00000	0.00022	0.00037
age	0.00191	0.00013	0.00000	0.00165	0.00218

term	estimate	std.error	p.value	2.5 %	97.5 %
female	-0.00601	0.00233	0.00981	-0.01057	-0.00145
caucasian	-0.01466	0.00259	0.00000	-0.01974	-0.00957
insurance	0.00021	0.00278	0.93973	-0.00524	0.00566

Question 11 Rubric (15 points):

- Give 5 points for performing multiple imputation correctly.
- Give 8 points if student creates the correct model.
- Give 2 points for generating the output table with the requested summaries.

Question 12 (10 points)

Specify the `insurance` effect size in the models you obtained in Questions 10 and 11. Provide a confidence interval and point estimate, and describe what the point estimate means in this context. Does the use of multiple imputation as opposed to a complete case analysis appear have a large impact on this estimate?

For the complete cases model, the `insurance` point estimate is -0.00334, with 95% confidence interval (-0.00926, 0.00259). This means that if we have two patients named Harry and Sally, where Harry has commercial insurance while Sally has non-commercial, then the log of Harry's systolic blood pressure is estimated to be 0.00334 mm Hg lower than Sally's. Since the 95% confidence interval contains 0, the log of Harry's systolic blood pressure is not detectably less than Sally's.

For the multiple imputation model, the `insurance` point estimate is 0.00021, with 95% confidence interval (-0.00524, 0.00566). This means that if we have two patients named Harry and Sally, where Harry has commercial insurance while Sally has non-commercial, then the log of Harry's systolic blood pressure is estimated to be 0.00021 mm Hg higher than Sally's. Since the 95% confidence interval contains 0, the log of Harry's systolic blood pressure is not detectably greater than Sally's.

The impact of the imputation is pretty small, generally, but you could certainly note that the confidence intervals are a bit different, and the p value for `insurance` is 0.27 for the complete case analysis and 0.94 for the multiply imputed version.

Question 12 Rubric (10 points):

- Give 3 points for appropriately describing the effects of insurance in the first (complete cases) model.
  - Subtract 2 points if the student doesn't correctly describe the insurance variable as the effect of moving from non-commercial to commercial insurance.
- Give 3 points for appropriately describing the effects of insurance in the second (multiple imputation) model.
  - Subtract 2 points if the student doesn't correctly describe the insurance variable as the effect of moving from non-commercial to commercial insurance.
- Give 4 points for appropriately comparing the results and describing the effect of multiple imputation on the summary results.

Overall Grading Note

We will subtract 5 points from the total score if there are more than two typographical/spelling or grammatical errors in the document.

Session Information

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

```
- Session info -------------------------------------------------------------
 setting  value
 version  R version 3.6.3 (2020-02-29)
 os       macOS Catalina 10.15.3
 system   x86_64, darwin15.6.0
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       America/New_York
 date     2020-03-20

- Packages -----------------------------------------------------------------
 package      * version    date       lib source
 abind          1.4-5      2016-07-21 [1] CRAN (R 3.6.0)
 acepack        1.4.1      2016-10-29 [1] CRAN (R 3.6.0)
 assertthat     0.2.1      2019-03-21 [1] CRAN (R 3.6.0)
 backports      1.1.5      2019-10-02 [1] CRAN (R 3.6.0)
 base64enc      0.1-3      2015-07-28 [1] CRAN (R 3.6.0)
 broom        * 0.5.5      2020-02-29 [1] CRAN (R 3.6.0)
 car          * 3.0-7      2020-03-11 [1] CRAN (R 3.6.0)
 carData      * 3.0-3      2019-11-16 [1] CRAN (R 3.6.0)
 caret        * 6.0-86     2020-03-20 [1] CRAN (R 3.6.3)
 cellranger     1.1.0      2016-07-27 [1] CRAN (R 3.6.0)
 checkmate      2.0.0      2020-02-06 [1] CRAN (R 3.6.0)
 class          7.3-15     2019-01-01 [1] CRAN (R 3.6.3)
 cli            2.0.2      2020-02-28 [1] CRAN (R 3.6.0)
 cluster        2.1.0      2019-06-19 [1] CRAN (R 3.6.3)
 codetools      0.2-16     2018-12-24 [1] CRAN (R 3.6.3)
 colorspace     1.4-1      2019-03-18 [1] CRAN (R 3.6.0)
 crayon         1.3.4      2017-09-16 [1] CRAN (R 3.6.0)
 curl           4.3        2019-12-02 [1] CRAN (R 3.6.1)
 data.table     1.12.8     2019-12-09 [1] CRAN (R 3.6.0)
 DBI            1.1.0      2019-12-15 [1] CRAN (R 3.6.0)
 dbplyr         1.4.2      2019-06-17 [1] CRAN (R 3.6.0)
 digest         0.6.25     2020-02-23 [1] CRAN (R 3.6.0)
 dplyr        * 0.8.5      2020-03-07 [1] CRAN (R 3.6.0)
 ellipsis       0.3.0      2019-09-20 [1] CRAN (R 3.6.0)
 evaluate       0.14       2019-05-28 [1] CRAN (R 3.6.0)
 fansi          0.4.1      2020-01-08 [1] CRAN (R 3.6.0)
 forcats      * 0.5.0      2020-03-01 [1] CRAN (R 3.6.0)
 foreach        1.4.8      2020-02-09 [1] CRAN (R 3.6.2)
 foreign        0.8-76     2020-03-03 [1] CRAN (R 3.6.0)
 Formula      * 1.2-3      2018-05-03 [1] CRAN (R 3.6.0)
 fs             1.3.2      2020-03-05 [1] CRAN (R 3.6.0)
 generics       0.0.2      2018-11-29 [1] CRAN (R 3.6.0)
 ggplot2      * 3.3.0      2020-03-05 [1] CRAN (R 3.6.0)
 glue           1.3.2      2020-03-12 [1] CRAN (R 3.6.0)
```

```
gower          0.2.1      2019-05-14 [1] CRAN (R 3.6.0)
gridExtra      2.3        2017-09-09 [1] CRAN (R 3.6.0)
gtable         0.3.0      2019-03-25 [1] CRAN (R 3.6.0)
haven          2.2.0      2019-11-08 [1] CRAN (R 3.6.0)
here         * 0.1        2017-05-28 [1] CRAN (R 3.6.0)
highr          0.8        2019-03-20 [1] CRAN (R 3.6.0)
Hmisc        * 4.3-1      2020-02-07 [1] CRAN (R 3.6.0)
hms            0.5.3      2020-01-08 [1] CRAN (R 3.6.0)
htmlTable      1.13.3     2019-12-04 [1] CRAN (R 3.6.1)
htmltools      0.4.0      2019-10-04 [1] CRAN (R 3.6.0)
htmlwidgets    1.5.1      2019-10-08 [1] CRAN (R 3.6.0)
httr           1.4.1      2019-08-05 [1] CRAN (R 3.6.0)
ipred          0.9-9      2019-04-28 [1] CRAN (R 3.6.0)
iterators      1.0.12     2019-07-26 [1] CRAN (R 3.6.0)
janitor      * 1.2.1      2020-01-22 [1] CRAN (R 3.6.0)
jpeg           0.1-8.1    2019-10-24 [1] CRAN (R 3.6.0)
jsonlite       1.6.1      2020-02-02 [1] CRAN (R 3.6.0)
knitr        * 1.28       2020-02-06 [1] CRAN (R 3.6.0)
lattice      * 0.20-40    2020-02-19 [1] CRAN (R 3.6.0)
latticeExtra   0.6-29     2019-12-19 [1] CRAN (R 3.6.0)
lava           1.6.7      2020-03-05 [1] CRAN (R 3.6.0)
lifecycle      0.2.0      2020-03-06 [1] CRAN (R 3.6.0)
lubridate      1.7.4      2018-04-11 [1] CRAN (R 3.6.0)
magrittr     * 1.5        2014-11-22 [1] CRAN (R 3.6.0)
MASS           7.3-51.5   2019-12-20 [1] CRAN (R 3.6.3)
Matrix         1.2-18     2019-11-27 [1] CRAN (R 3.6.3)
MatrixModels   0.4-1      2015-08-22 [1] CRAN (R 3.6.0)
mice         * 3.8.0      2020-02-21 [1] CRAN (R 3.6.0)
ModelMetrics   1.2.2.2    2020-03-17 [1] CRAN (R 3.6.0)
modelr         0.1.6      2020-02-22 [1] CRAN (R 3.6.2)
multcomp       1.4-12     2020-01-10 [1] CRAN (R 3.6.2)
munsell        0.5.0      2018-06-12 [1] CRAN (R 3.6.0)
mvtnorm        1.1-0      2020-02-24 [1] CRAN (R 3.6.0)
naniar       * 0.5.0      2020-02-28 [1] CRAN (R 3.6.0)
nlme           3.1-145    2020-03-04 [1] CRAN (R 3.6.0)
nnet           7.3-13     2020-02-25 [1] CRAN (R 3.6.0)
openxlsx       4.1.4      2019-12-06 [1] CRAN (R 3.6.0)
pillar         1.4.3      2019-12-20 [1] CRAN (R 3.6.0)
pkgconfig      2.0.3      2019-09-22 [1] CRAN (R 3.6.0)
plyr           1.8.6      2020-03-03 [1] CRAN (R 3.6.0)
png            0.1-7      2013-12-03 [1] CRAN (R 3.6.0)
polspline      1.1.17     2019-11-07 [1] CRAN (R 3.6.0)
pROC           1.16.2     2020-03-19 [1] CRAN (R 3.6.3)
prodlim        2019.11.13 2019-11-17 [1] CRAN (R 3.6.0)
purrr        * 0.3.3      2019-10-18 [1] CRAN (R 3.6.0)
quantreg       5.54       2019-12-13 [1] CRAN (R 3.6.0)
R6             2.4.1      2019-11-12 [1] CRAN (R 3.6.0)
RColorBrewer   1.1-2      2014-12-07 [1] CRAN (R 3.6.0)
Rcpp           1.0.4      2020-03-17 [1] CRAN (R 3.6.0)
readr        * 1.3.1      2018-12-21 [1] CRAN (R 3.6.0)
readxl         1.3.1      2019-03-13 [1] CRAN (R 3.6.0)
recipes        0.1.10     2020-03-18 [1] CRAN (R 3.6.0)
reprex         0.3.0      2019-05-16 [1] CRAN (R 3.6.0)
reshape2       1.4.3      2017-12-11 [1] CRAN (R 3.6.0)
```

```
rio            0.5.16     2018-11-26 [1] CRAN (R 3.6.0)
rlang          0.4.5      2020-03-01 [1] CRAN (R 3.6.0)
rmarkdown      2.1        2020-01-20 [1] CRAN (R 3.6.2)
rms          * 5.1-4      2019-11-17 [1] CRAN (R 3.6.0)
rpart          4.1-15     2019-04-12 [1] CRAN (R 3.6.3)
rprojroot      1.3-2      2018-01-03 [1] CRAN (R 3.6.0)
rstudioapi     0.11       2020-02-07 [1] CRAN (R 3.6.0)
rvest          0.3.5      2019-11-08 [1] CRAN (R 3.6.0)
sandwich       2.5-1      2019-04-06 [1] CRAN (R 3.6.0)
scales         1.1.0      2019-11-18 [1] CRAN (R 3.6.0)
sessioninfo    1.1.1      2018-11-05 [1] CRAN (R 3.6.0)
simputation  * 0.2.4      2020-03-13 [1] CRAN (R 3.6.0)
snakecase      0.11.0     2019-05-25 [1] CRAN (R 3.6.0)
SparseM      * 1.78       2019-12-13 [1] CRAN (R 3.6.0)
stringi        1.4.6      2020-02-17 [1] CRAN (R 3.6.0)
stringr      * 1.4.0      2019-02-10 [1] CRAN (R 3.6.0)
survival     * 3.1-11     2020-03-07 [1] CRAN (R 3.6.0)
TH.data        1.0-10     2019-01-21 [1] CRAN (R 3.6.0)
tibble       * 2.1.3      2019-06-06 [1] CRAN (R 3.6.0)
tidyr        * 1.0.2      2020-01-24 [1] CRAN (R 3.6.0)
tidyselect     1.0.0      2020-01-27 [1] CRAN (R 3.6.2)
tidyverse    * 1.3.0      2019-11-21 [1] CRAN (R 3.6.0)
timeDate       3043.102   2018-02-21 [1] CRAN (R 3.6.0)
utf8           1.1.4      2018-05-24 [1] CRAN (R 3.6.0)
vctrs          0.2.4      2020-03-10 [1] CRAN (R 3.6.0)
visdat         0.5.3      2019-02-15 [1] CRAN (R 3.6.0)
withr          2.1.2      2018-03-15 [1] CRAN (R 3.6.0)
xfun           0.12       2020-01-13 [1] CRAN (R 3.6.0)
xml2           1.2.5      2020-03-11 [1] CRAN (R 3.6.0)
yaml           2.2.1      2020-02-01 [1] CRAN (R 3.6.0)
zip            2.0.4      2019-09-01 [1] CRAN (R 3.6.0)
zoo            1.8-7      2020-01-10 [1] CRAN (R 3.6.0)

[1] /Library/Frameworks/R.framework/Versions/3.6/Resources/library
```