

# Model Complete Cases

*Martin Skarzynski*

*2018-04-20*

```
library(here)

## here() starts at /Users/marskar/gdrive/nhanes
library(readr)
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
library(tidyr)
library(survey)

## Loading required package: grid
## Loading required package: methods
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##   expand
## Loading required package: survival
##
## Attaching package: 'survey'
## The following object is masked from 'package:graphics':
##
##   dotchart
library(purrr)

# this function takes in two integers as arguments
# the seed and the number of random variables
# and returns a dataframe
get_modelstats <- function(seed, n_random_vars){

  set.seed(seed)
  #Variables included in the model
  chosen_vars <- c(
    "PERMTH_INT",
```

```

        "canc_mort",
        "SDPPSU6",
        "SDPSTRA6",
        "WTPFQX6"
    )

remove_vars <- c(
    "HAN9", #remove age variables
    "HAQ7",
    "HAT29",
    "HAJO",
    "WTPXRP2", #remove unneed weight variables
    "WTPQRP21",
    "WTPQRP27",
    "WTPQRP43"
)

#remove variables,
#move chosen variables to the beginning, and
#sample from the variables to be randomized
read_rds(here('dat/3-clean-complete-cases.rds')) %>%
    select(-one_of(remove_vars)) %>%
    select(one_of(chosen_vars),
            everything()[sample(seq(ncol(.)),
                                n_random_vars)]) ->
dat

# create survey design object
svydesign(ids = ~SDPPSU6,
          strata = ~SDPSTRA6,
          weights = ~WTPFQX6,
          nest = TRUE,
          data = dat) ->
des

# create left sides of equations
form <- as.formula(Surv(PERMTH_INT, canc_mort) ~ x1)
# create right sides of equations
n_params <- ncol(dat) # the total number of parameters
# n_params should equal the number of chosen variables
# plus the number of random variables
r_cols <- (length(chosen_vars)+1):n_params
if(n_random_vars==1){
    vrs <- as.name(names(dat)[n_params])
    vrs2 <- as.name(names(dat)[n_params])
} else{
    vrs <- as.name(paste(names(dat)[r_cols], collapse=' + '))
    vrs2 <- as.name(paste(names(dat)[r_cols], collapse=', '))
}

set.seed(seed)
#train <- sample(x = seq(nrow(dat)),
#
#               n_random_vars = round(nrow(dat)*.7))

```

```

# generate cox models without and with penalties

cox <- try(svycoxph(update(form,
  paste("~ ", vrs)),
  design = des, data = dat))

rid <- try(svycoxph(update(form,
  paste("~ ridge(", vrs2, ')')),
  design = des, data = dat))

# define functions needed to create first table
get_con <- function(x) {
  signif(summary(x)$concordance[1]*100, digits = 2)
}
get_HR <- function(x) {
  summary(x)$conf.int[, "exp(coef)"]
}
get_HR_CI_lower <- function(x) {
  summary(x)$conf.int[, "lower .95"]
}
get_HR_CI_upper <- function(x) {
  summary(x)$conf.int[, "upper .95"]
}
get_coef_pvalue <- function(x) {
  coefs <- summary(x)$coef
  coefs[, ncol(coefs)]
}
model_list <- try(list(cox, rid))

try(data_frame(seed = rep(seed, 2),
  n_random_vars = n_random_vars,
  type = c('coxph', 'ridge'),
  aic = AIC(cox, rid)[, "AIC"],
  concordance = map_dbl(model_list,
    get_con),
  hazard_ratio = map(model_list,
    get_HR),
  HR_CI_lower = map(model_list,
    get_HR_CI_lower),
  HR_CI_upper = map(model_list,
    get_HR_CI_upper),
  coef_pvalue = map(model_list,
    get_coef_pvalue)))
}

#save an object with 1000 models
map_seeds <- function(seed){
  map2_dfr(.x = seed,
    .y = seq(10), #max number of variables to randomize
    get_modelstats)
}
map_dfr(seq(50), map_seeds) %>%

```









[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]







[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]











[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]















[illegible]

[illegible]







[illegible]





[illegible]



[illegible]





[illegible]









[illegible]





[illegible]

[illegible]

[illegible]

[illegible]









[illegible]



[illegible]











[illegible]

[illegible]

[illegible]



[illegible]







[illegible]

[illegible]



[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]





[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]



[illegible]

[illegible]

[illegible]



[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



[illegible]



[illegible]

[illegible]

[illegible]







[illegible]

[illegible]



[illegible]



[illegible]





[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



[illegible]



[illegible]



[illegible]

[illegible]





[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]









































[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]





























[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

##      nest = TRUE, data = dat)
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (98) clusters.
## svydesign(ids = ~SDPPSU6, strata = ~SDPSTRA6, weights = ~WTPFQX6,
##      nest = TRUE, data = dat)
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (98) clusters.
## svydesign(ids = ~SDPPSU6, strata = ~SDPSTRA6, weights = ~WTPFQX6,
##      nest = TRUE, data = dat)
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (98) clusters.
## svydesign(ids = ~SDPPSU6, strata = ~SDPSTRA6, weights = ~WTPFQX6,
##      nest = TRUE, data = dat)
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (98) clusters.
## svydesign(ids = ~SDPPSU6, strata = ~SDPSTRA6, weights = ~WTPFQX6,
##      nest = TRUE, data = dat)
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (98) clusters.
## svydesign(ids = ~SDPPSU6, strata = ~SDPSTRA6, weights = ~WTPFQX6,
##      nest = TRUE, data = dat)
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (98) clusters.
## svydesign(ids = ~SDPPSU6, strata = ~SDPSTRA6, weights = ~WTPFQX6,
##      nest = TRUE, data = dat)

```