

CK2

Group 10

2022-11-03

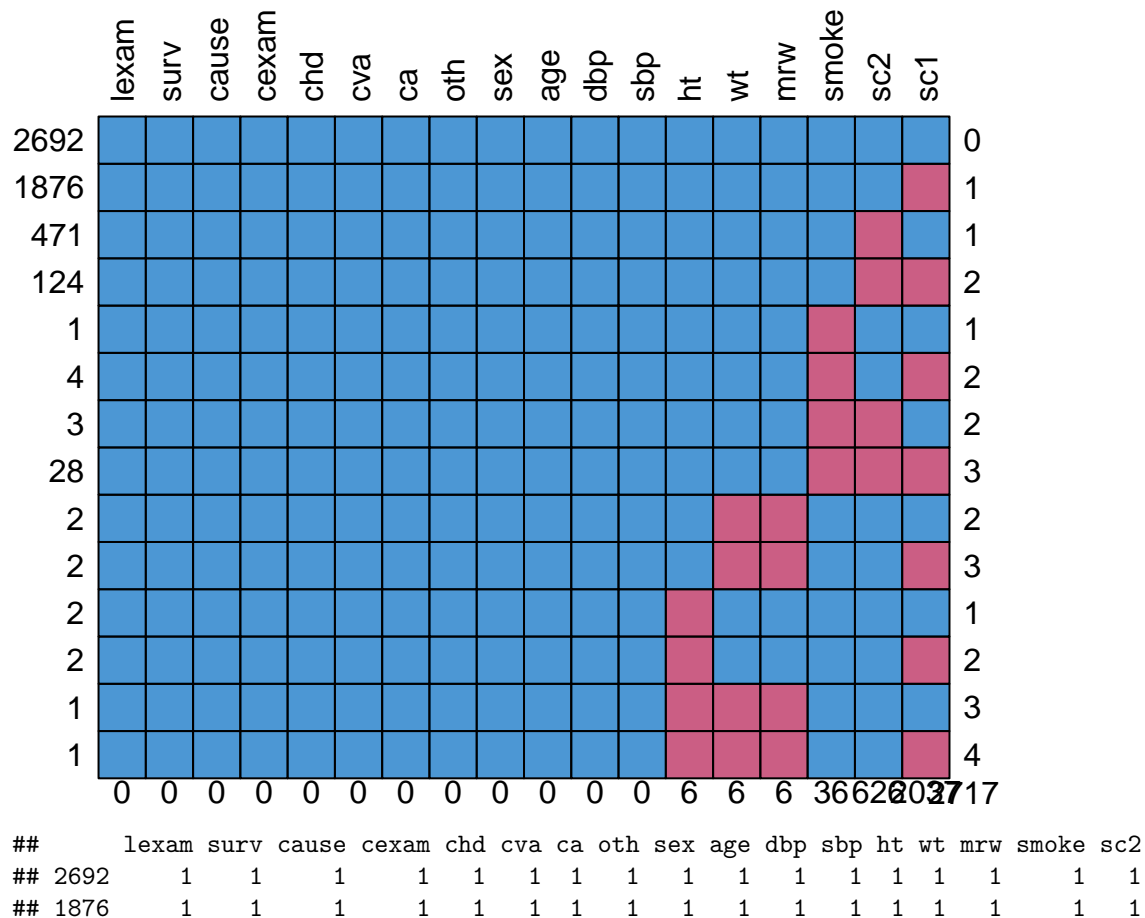
```
library(dplyr)
library(ggplot2)
library(foreign)
library(mice)
data <- read.dta("fram.dta")
```

Q4. Missing Data Analysis

Types of Missingness

From the plot below, we see 14 different patterns of missingness. Patterns with most missing values are **1. missing in sc1**, **2. missing in sc2**, and **3. missing in sc1 and sc2**.

```
md.pattern(data, rotate.names = TRUE)
```



```

## 471      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      0
## 124      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      0
## 1        1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      0      1
## 4        1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      0      1
## 3        1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      0      0
## 28       1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      0      0
## 2        1      1      1      1      1      1      1      1      1      1      1      1      1      0      0      1      1
## 2        1      1      1      1      1      1      1      1      1      1      1      1      1      0      0      1      1
## 2        1      1      1      1      1      1      1      1      1      1      1      1      0      1      1      1      1
## 2        1      1      1      1      1      1      1      1      1      1      1      1      0      1      1      1      1
## 1        1      1      1      1      1      1      1      1      1      1      1      1      0      0      0      1      1
## 1        1      1      1      1      1      1      1      1      1      1      1      1      0      0      0      1      1
##          0      0      0      0      0      0      0      0      0      0      0      0      0      6      6      6      36 626
##          sc1
## 2692     1      0
## 1876     0      1
## 471      1      1
## 124      0      2
## 1        1      1
## 4        0      2
## 3        1      2
## 28       0      3
## 2        1      2
## 2        0      3
## 2        1      1
## 2        0      2
## 1        1      3
## 1        0      4
##          2037 2717

```

By fitting the logistic regression of indicator R1 with the rest of the covariate, we identify significant predictors with p-values less than 0.05. These are `cexam`, `sex`, `dbp`, `smoke`, and `sc2`. Thus we say that missingness in `sc1` is NOT MCAR.

By fitting the logistic regression of indicator R2 with the rest of the covariate, we identify significant predictors with p-values less than 0.05. These are `Lexam`, `dbp`, and `sbp`. Thus we say that missingness in `sc2` is NOT MCAR.

By fitting the logistic regression of indicator R3 with the rest of the covariate, we did NOT identify significant predictors with p-values less than 0.05. Thus we say that missingness in the (`sc1`, `sc2`) pair is MCAR.

```
### Create indicator variable R1 where R1=1 if sample experience missingness in sc1
```

```
data <- data %>% mutate(R1 = if_else(is.na(sc1), 1, 0))
```

```
data <- data %>% mutate(R2 = if_else(is.na(sc2), 1, 0))
```

```
data <- data %>% mutate(R3 = if_else(R1+R2==2, 1, 0))
```

```
### Test for MCAR by logistic regression
```

```
logit1 <- glm(R1 ~ leexam + surv + cause + cexam + cva + ca + oth + sex + age + dbp + sbp + ht + wt + mrw +
summary(logit1)
```

```
##
```

```
## Call:
```

```
## glm(formula = R1 ~ leexam + surv + cause + cexam + cva + ca +
##      oth + sex + age + dbp + sbp + ht + wt + mrw + smoke + sc2,
##      family = "binomial", data = data)
```

```
##
```

```
## Deviance Residuals:
```

```

##      Min      1Q   Median      3Q      Max
## -1.6299 -1.0342 -0.8277  1.2398  1.9569
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.0077235  2.9288414  -0.003  0.997896
## leexam      -0.0195046  0.0144085  -1.354  0.175837
## surv        0.1975997  0.1800836   1.097  0.272525
## cause      -0.1044127  0.0744720  -1.402  0.160904
## cexam       0.0255558  0.0073209   3.491  0.000482 ***
## cva         0.0258111  0.2799328   0.092  0.926535
## ca         -0.1990322  0.1915329  -1.039  0.298733
## oth         0.3257275  0.4075533   0.799  0.424159
## sex         0.6018454  0.1051644   5.723  1.05e-08 ***
## age         0.0045063  0.0044396   1.015  0.310091
## dbp        -0.0237534  0.0041418  -5.735  9.75e-09 ***
## sbp         0.0002085  0.0023192   0.090  0.928365
## ht          0.0010009  0.0460712   0.022  0.982668
## wt          0.0160012  0.0096609   1.656  0.097665 .
## mrw        -0.0188196  0.0120381  -1.563  0.117973
## smoke      -0.0164396  0.0029867  -5.504  3.71e-08 ***
## sc2         0.0026887  0.0007274   3.696  0.000219 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 6186.0  on 4567  degrees of freedom
## Residual deviance: 5993.2  on 4551  degrees of freedom
## (641 observations deleted due to missingness)
## AIC: 6027.2
##
## Number of Fisher Scoring iterations: 4
logit2 <- glm(R2 ~ leexam + surv + cause + cexam + cva + ca + oth + sex + age + dbp + sbp + ht + wt + mrw + smoke + sc1,
summary(logit2)

##
## Call:
## glm(formula = R2 ~ leexam + surv + cause + cexam + cva + ca +
##      oth + sex + age + dbp + sbp + ht + wt + mrw + smoke + sc1,
##      family = "binomial", data = data)
##
## Deviance Residuals:
##      Min      1Q   Median      3Q      Max
## -1.1191 -0.5889 -0.5266 -0.4483  2.4133
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.488881  4.546406   0.547  0.58408
## leexam      -0.103010  0.020639  -4.991  6.01e-07 ***
## surv       -0.525694  0.298507  -1.761  0.07823 .
## cause       0.098808  0.098838   1.000  0.31745
## cexam      -0.023889  0.014509  -1.647  0.09965 .
## cva        -0.077892  0.405591  -0.192  0.84771

```

```

## ca          -0.064811    0.281683   -0.230   0.81803
## oth          -0.363254    0.572896   -0.634   0.52604
## sex           0.189586    0.173215    1.095   0.27373
## age          -0.001429    0.007283   -0.196   0.84449
## dbp          -0.016891    0.006164   -2.740   0.00614 **
## sbp           0.011860    0.003368    3.521   0.00043 ***
## ht          -0.045596    0.072156   -0.632   0.52744
## wt           0.003807    0.015291    0.249   0.80338
## mrw          -0.003140    0.018675   -0.168   0.86649
## smoke         0.007306    0.004593    1.591   0.11163
## sc1          -0.001688    0.001221   -1.382   0.16690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2662.1 on 3162 degrees of freedom
## Residual deviance: 2593.0 on 3146 degrees of freedom
## (2046 observations deleted due to missingness)
## AIC: 2627
##
## Number of Fisher Scoring iterations: 4
logit3 <- glm(R3 ~ leexam + surv + cause + cexam + cva + ca + oth + sex + age + dbp + sbp + ht + wt + mrw + smoke, family = "binomial", data = data)
summary(logit3)

##
## Call:
## glm(formula = R3 ~ leexam + surv + cause + cexam + cva + ca + oth + sex + age + dbp + sbp + ht + wt + mrw + smoke, family = "binomial", data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6472  -0.2433  -0.2051  -0.1745   3.1445
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.254236    8.619116  -1.654   0.0982 .
## leexam       -0.029084    0.039157  -0.743   0.4576
## surv         0.508612    0.550016   0.925   0.3551
## cause       -0.091023    0.234104  -0.389   0.6974
## cexam       -0.003473    0.023085  -0.150   0.8804
## cva         0.036185    0.847738   0.043   0.9660
## ca          0.120798    0.566066   0.213   0.8310
## oth         0.023943    1.254766   0.019   0.9848
## sex         0.550062    0.325974   1.687   0.0915 .
## age        -0.009502    0.012849  -0.740   0.4596
## dbp        -0.013812    0.012401  -1.114   0.2654
## sbp        -0.005703    0.007002  -0.814   0.4154
## ht         0.168112    0.137565   1.222   0.2217
## wt        -0.033006    0.028627  -1.153   0.2489
## mrw         0.053646    0.034256   1.566   0.1173
## smoke       0.003375    0.008831   0.382   0.7024
## ---

```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 1169.8  on 5162  degrees of freedom
## Residual deviance: 1141.0  on 5147  degrees of freedom
##    (46 observations deleted due to missingness)
## AIC: 1173
##
## Number of Fisher Scoring iterations: 7
```

To argue for whether missingness in `sc1` is MAR or MNAR, we reason with context. `sc1` stands for serum cholesterol exam 1, which is the serum cholesterol level of each individual from their first exam. For `sc1` to be MNAR, it's missingness has to be dependent on `sc1` itself. This means those with higher or lower `sc1` may be less or more likely to take the examination. Since we see no logical explanation behind the former statement, we are more inclined to conclude that are data is MAR. Using the same logic, we say missingness `insc2` is also MAR.

Accomodation

We first examine the total number of missing values per column. The 5% threshold is 260. We see only missing values in `sc1` and `sc2` with count greater than the threshold. Thus, for columns with missing value count less than the threshold, one method would be to use the Complete Case Analysis.

```
colSums(is.na(data))

## lexam  surv  cause  cexam   chd    cva    ca   oth   sex   age   ht   wt   sc1
##      0      0      0      0      0      0      0      0      0      0      6      6  2037
##   sc2   dbp   sbp   mrw smoke   R1    R2    R3
##   626      0      0      6   36      0      0      0

sum(is.na(data$sc1)) / nrow(data)

## [1] 0.3910539

sum(is.na(data$sc2)) / nrow(data)

## [1] 0.1201766

nrow(data) * 0.05

## [1] 260.45
```

From the pattern plot above, among the 14 patterns of missing data, only the top 4 has significant count of missing values. Hence, we use Multivariate Imputation with 4 imputations.

To ensure our imputation did not add new information to the data, we drew correlation plots for data before (only include complete cases) and after imputation. We can see from below that the plot looks very similar.

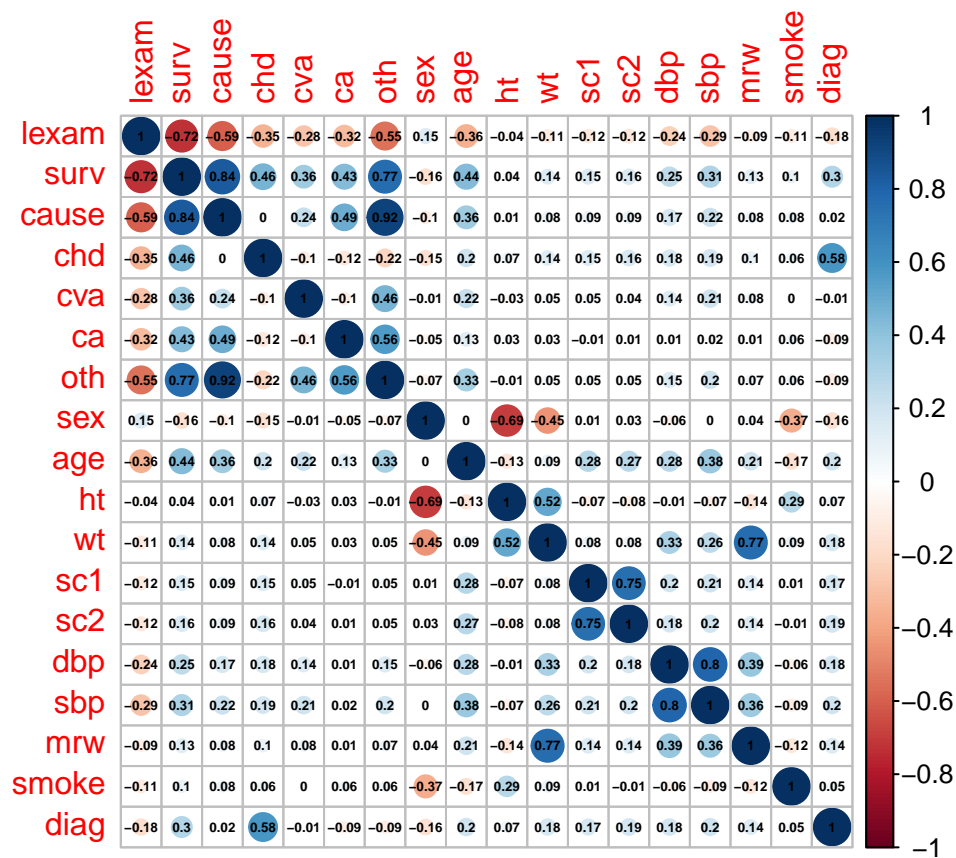
```
library(corrplot)

## corrplot 0.92 loaded

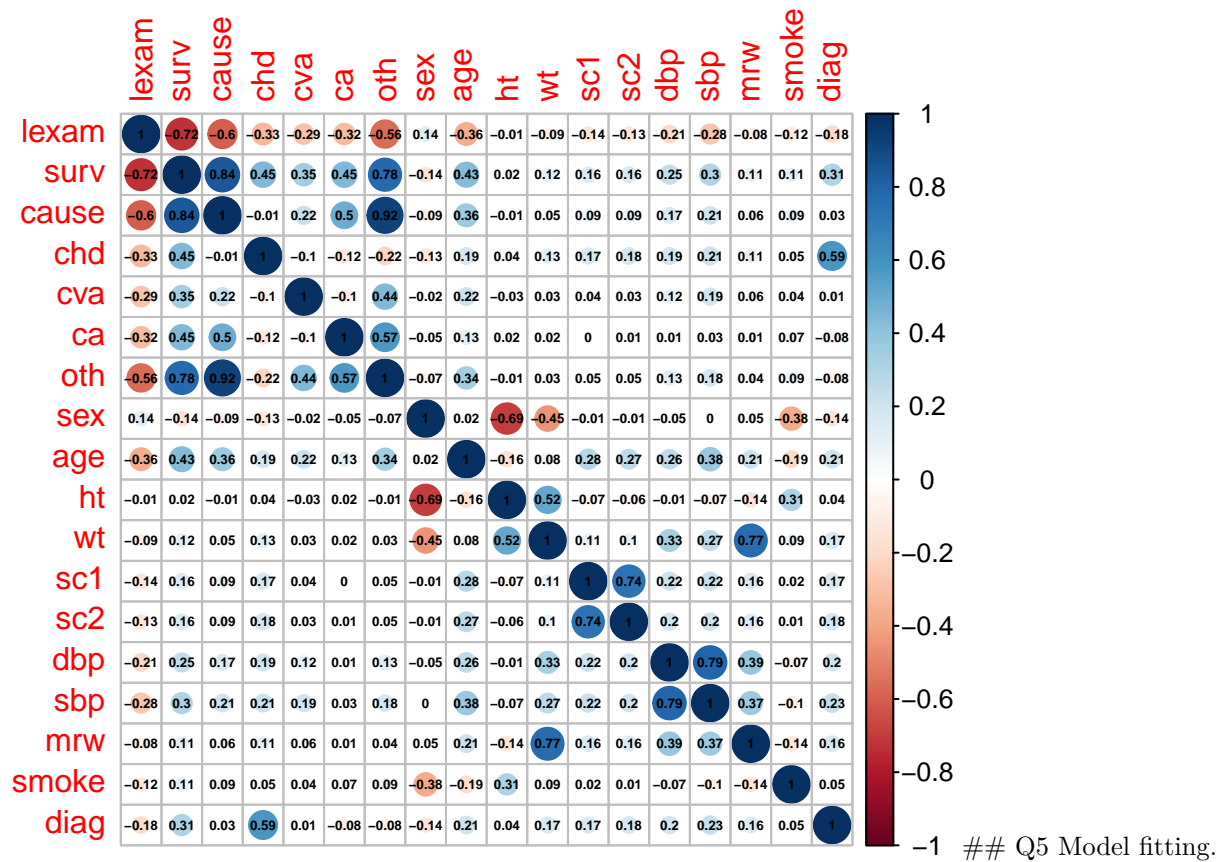
data <- data %>% mutate(diag = if_else(cexam == 0, 0, 1))
tempData = mice(data, m =4, "pmm", printFlag = FALSE, seed = 2022)

## Warning: Number of logged events: 120

data_imputed <- complete(tempData, action=1)
M = cor(data_imputed[, -c(4, 19, 20, 21)])
corrplot(M, addCoef.col = 'black', number.cex= 7/(ncol(data) - 4))
```



```
N = cor(data[complete.cases(data), -c(4, 19, 20, 21)])
corrplot(N, addCoef.col = 'black', number.cex= 7/(ncol(data) - 4))
```



We first include a BMI column and drop wt, ht, and mrw since BMI is a linear combination of the dropped variables.

```
data_imputed$bmi = 703 * data_imputed$wt / (data_imputed$ht)^2
data_imputed$bmi_sq = (data_imputed$bmi)^2
```

Logistic Lasso

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

```
y = as.matrix(data_imputed$diag)
```

```
x = as.matrix(dplyr::select(data_imputed, c('sex', 'age', 'sc1', 'sc2', 'dbp', 'sbp', 'smoke', 'bmi', 'bmi_sq')))
```

```
cv_model <- cv.glmnet(x, y, alpha = 1)
```

```
#find optimal lambda value that minimizes test MSE
```

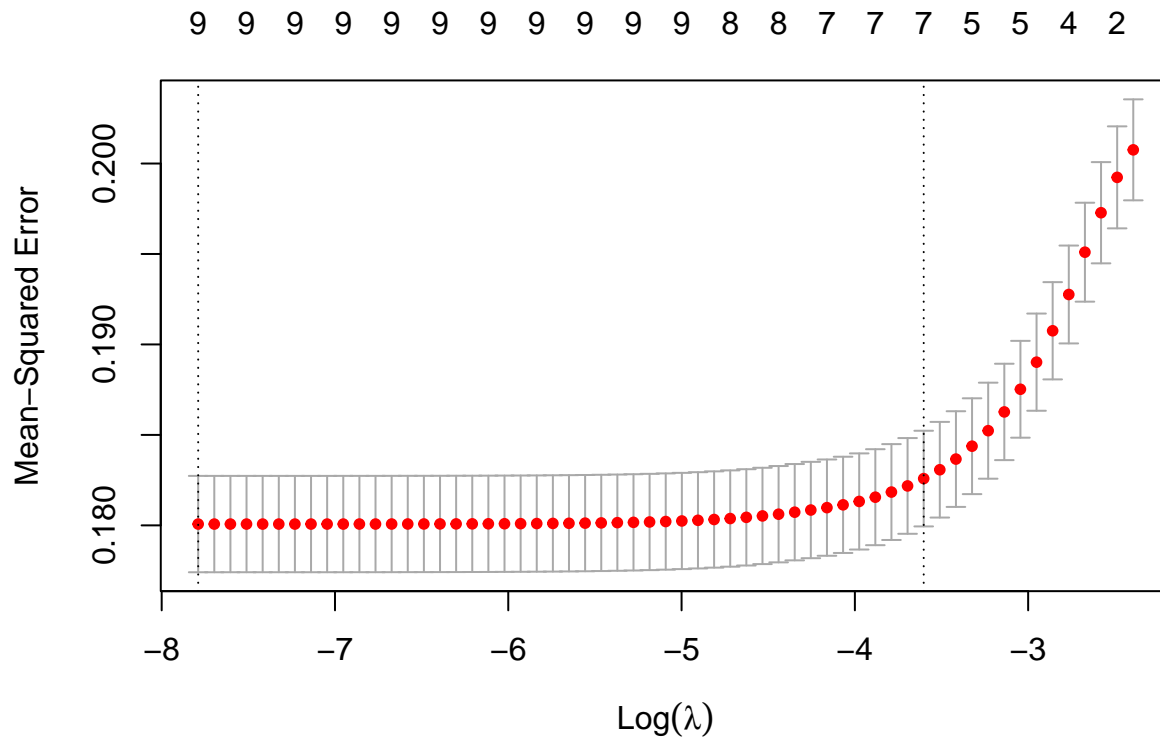
```
best_lambda <- cv_model$lambda.min
```

```
best_lambda
```

```
## [1] 0.000414107
```

```
#produce plot of test MSE by lambda value
```

```
plot(cv_model)
```



```
best_model <- glmnet(x, y, alpha = 1, family = "binomial", lambda = best_lambda)
coef(best_model)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                s0
## (Intercept) -5.796651540
## sex         -0.762997966
## age          0.035753203
## sc1          0.001181410
## sc2          0.006203233
## dbp          0.003834676
## sbp          0.008116909
## smoke        0.006443167
## bmi          0.044576783
## bmi_sq       .
```

```
library(glmnet)
y = as.matrix(data_imputed$chd)
```

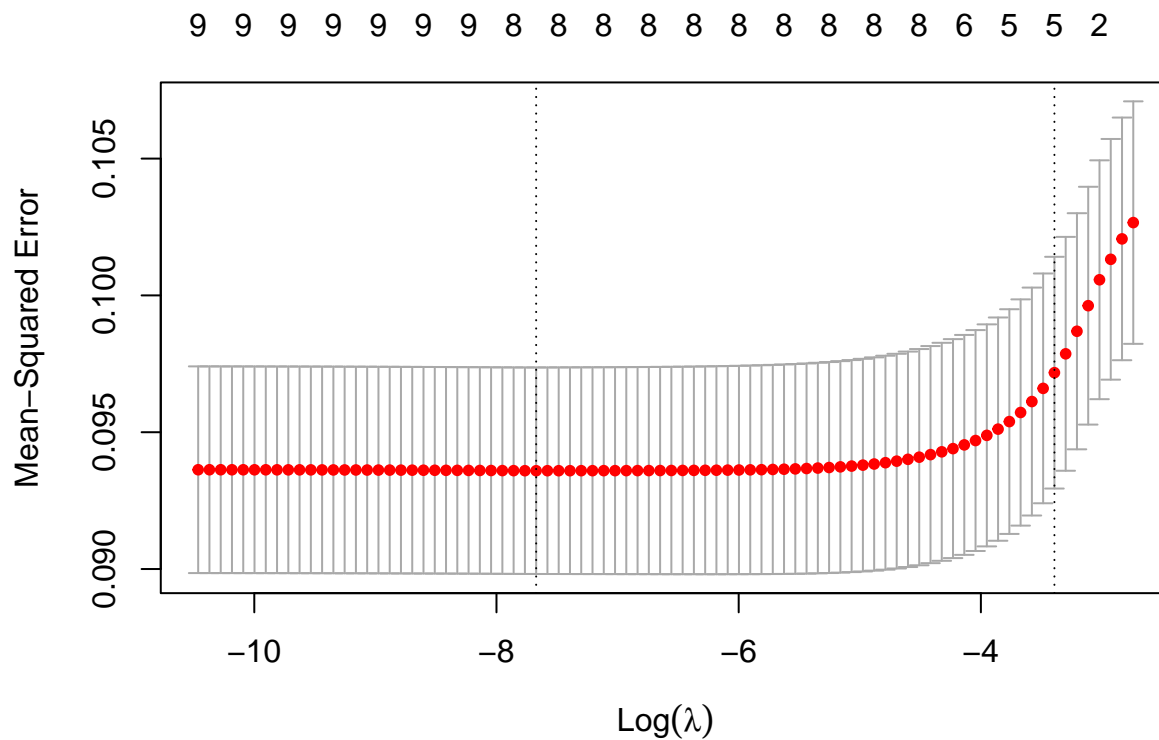
```
x = as.matrix(dplyr::select(data_imputed, c('sex', 'age', 'sc1', 'sc2', 'dbp', 'sbp', 'smoke', 'bmi', 'bmi_sq')))
```

```
cv_model <- cv.glmnet(x, y, alpha = 1)
```

```
#find optimal lambda value that minimizes test MSE
best_lambda <- cv_model$lambda.min
best_lambda
```

```
## [1] 0.0004655286
```

```
#produce plot of test MSE by lambda value
plot(cv_model)
```

```
best_model <- glmnet(x, y, alpha = 1, family = "binomial", lambda = best_lambda)
coef(best_model)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept) -8.0646590804
## sex         -1.0591548869
## age          0.0611263096
## sc1           0.0012213348
## sc2           0.0074131655
## dbp           0.0129357675
## sbp           0.0074065919
## smoke         0.0125882569
## bmi           0.0007988262
## bmi_sq        0.0005198748
```

Multinomial Logistic

```
library(magrittr) # needs to be run every time you start R and want to use %>%
library(dplyr)

data_imputed %<>%
  mutate(cexam_comb = case_when(
    cexam %in% 0 ~ "0",
    cexam %in% 1 ~ "1",
    cexam %in% 2:6 ~ "2",
    cexam %in% 7:11 ~ "3",
    cexam %in% 12:16 ~ "4"
  ))
```

```
library(nnet)
x = as.matrix(dplyr::select(data_imputed, c('sex', 'age', 'sc1', 'sc2', 'dbp', 'sbp', 'smoke', 'bmi',
y = as.matrix(data_imputed$cexam_comb)
mod = multinom(y~x)
```

```
## # weights: 55 (40 variable)
## initial value 8383.562086
## iter 10 value 5328.701603
## iter 20 value 5041.251567
## iter 30 value 4994.474194
## iter 40 value 4452.560821
## iter 50 value 4441.194877
## final value 4441.193762
## converged
```

```
summary(mod)
```

```
## Call:
## multinom(formula = y ~ x)
##
## Coefficients:
## (Intercept)      xsex      xage      xsc1      xsc2      xdbp
## 1 -10.557744 -1.3418182  0.108082410 -0.0070284505 0.011646888 -0.0023793691
## 2 -15.098703 -1.2182716  0.093318259 -0.0014675038 0.012102944  0.0149901100
## 3 -8.118530 -0.7362341  0.037877882  0.0007015824 0.006210548  0.0045770123
## 4 -3.167041 -0.5397982 -0.002433658  0.0047223697 0.002567896 -0.0009698932
##
##      xsbp      xsmoke      xbmi      xbmi_sq
## 1 0.016430806  0.013414629 -0.04139630  0.001748408
## 2 0.009932578  0.024883585  0.26503033 -0.003435446
## 3 0.010004858  0.008767162  0.11183598 -0.001201018
## 4 0.002522616 -0.007547208 -0.01341692  0.000896759
##
## Std. Errors:
## (Intercept)      xsex      xage      xsc1      xsc2      xdbp
## 1 0.007202607  0.26720364  0.015480707  0.003672284 0.003606170 0.012741771
## 2 0.003187545  0.14156216  0.008276447  0.001895220 0.001876982 0.007117022
## 3 0.002719138  0.10449630  0.006267475  0.001559617 0.001571171 0.005936186
## 4 0.003085312  0.09941138  0.006293489  0.001562875 0.001600162 0.006269294
##
##      xsbp      xsmoke      xbmi      xbmi_sq
## 1 0.006559182  0.009306620  0.07006048  0.0011569759
## 2 0.003813146  0.004779483  0.04219195  0.0007991296
## 3 0.003221024  0.004116348  0.03092815  0.0005798658
## 4 0.003546118  0.004386767  0.03018362  0.0005513686
##
## Residual Deviance: 8882.388
## AIC: 8962.388
```

```
tidy(mod, exponentiate=FALSE, conf.int=TRUE) %>%
  knitr::kable(digits=3, format="markdown")
```

y.level	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1	(Intercept)	-10.558	0.007	-1465.823	0.000	-10.572	-10.544
1	xsex	-1.342	0.267	-5.022	0.000	-1.866	-0.818
1	xage	0.108	0.015	6.982	0.000	0.078	0.138

y.level	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1	xsc1	-0.007	0.004	-1.914	0.056	-0.014	0.000
1	xsc2	0.012	0.004	3.230	0.001	0.005	0.019
1	xdbp	-0.002	0.013	-0.187	0.852	-0.027	0.023
1	xbp	0.016	0.007	2.505	0.012	0.004	0.029
1	xsmoke	0.013	0.009	1.441	0.149	-0.005	0.032
1	xbmi	-0.041	0.070	-0.591	0.555	-0.179	0.096
1	xbmi_sq	0.002	0.001	1.511	0.131	-0.001	0.004
2	(Intercept)	-15.099	0.003	-4736.781	0.000	-15.105	-15.092
2	xsex	-1.218	0.142	-8.606	0.000	-1.496	-0.941
2	xage	0.093	0.008	11.275	0.000	0.077	0.110
2	xsc1	-0.001	0.002	-0.774	0.439	-0.005	0.002
2	xsc2	0.012	0.002	6.448	0.000	0.008	0.016
2	xdbp	0.015	0.007	2.106	0.035	0.001	0.029
2	xbp	0.010	0.004	2.605	0.009	0.002	0.017
2	xsmoke	0.025	0.005	5.206	0.000	0.016	0.034
2	xbmi	0.265	0.042	6.282	0.000	0.182	0.348
2	xbmi_sq	-0.003	0.001	-4.299	0.000	-0.005	-0.002
3	(Intercept)	-8.119	0.003	-2985.700	0.000	-8.124	-8.113
3	xsex	-0.736	0.104	-7.046	0.000	-0.941	-0.531
3	xage	0.038	0.006	6.044	0.000	0.026	0.050
3	xsc1	0.001	0.002	0.450	0.653	-0.002	0.004
3	xsc2	0.006	0.002	3.953	0.000	0.003	0.009
3	xdbp	0.005	0.006	0.771	0.441	-0.007	0.016
3	xbp	0.010	0.003	3.106	0.002	0.004	0.016
3	xsmoke	0.009	0.004	2.130	0.033	0.001	0.017
3	xbmi	0.112	0.031	3.616	0.000	0.051	0.172
3	xbmi_sq	-0.001	0.001	-2.071	0.038	-0.002	0.000
4	(Intercept)	-3.167	0.003	-1026.490	0.000	-3.173	-3.161
4	xsex	-0.540	0.099	-5.430	0.000	-0.735	-0.345
4	xage	-0.002	0.006	-0.387	0.699	-0.015	0.010
4	xsc1	0.005	0.002	3.022	0.003	0.002	0.008
4	xsc2	0.003	0.002	1.605	0.109	-0.001	0.006
4	xdbp	-0.001	0.006	-0.155	0.877	-0.013	0.011
4	xbp	0.003	0.004	0.711	0.477	-0.004	0.009
4	xsmoke	-0.008	0.004	-1.720	0.085	-0.016	0.001
4	xbmi	-0.013	0.030	-0.445	0.657	-0.073	0.046
4	xbmi_sq	0.001	0.001	1.626	0.104	0.000	0.002

Poisson / Negative Binomial

```
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```

library(MASS)

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##      select

p_data <- dplyr::select(data_imputed, c('sex', 'age', 'sc1', 'sc2', 'dbp', 'sbp', 'smoke', 'bmi', 'cexam'))
p_data <- p_data %>% mutate(count = if_else(cexam!=0, 1, 0))
### calculate exposure time for offset, average exam interval is 2 years
p_data <- p_data %>% mutate(years = case_when(cexam == 0 ~ 15 * 2,
                                             cexam == 1 ~ 0.1,
                                             cexam != 0 & cexam != 1 ~ (cexam - 1) * 2))
p_mod <- glm(formula = count ~ sex + age + sc1 + sc2 + dbp + sbp + smoke + bmi + offset(years), data = p_data)

## Warning: glm.fit: algorithm did not converge
deviance(p_mod)/p_mod$df.residual

## [1] 7.953426

```

Diagnosis Fitting the Poisson model show overdispersion. In combination with the fact that our outcome is dichotomous, we switch to negative binomial to address the problem.

```

negative_binomial <- glm.nb(count ~ sex + age + sc1 + sc2 + dbp + sbp + smoke + bmi, data = p_data)
summary(negative_binomial)

```

```

##
## Call:
## glm.nb(formula = count ~ sex + age + sc1 + sc2 + dbp + sbp +
##      smoke + bmi, data = p_data, init.theta = 6325.893044, link = log)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5514  -0.7407  -0.5657   0.5128   1.8529
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.4828422  0.2616649 -17.132  < 2e-16 ***
## sex          -0.5292607  0.0580283  -9.121  < 2e-16 ***
## age           0.0257415  0.0033956   7.581 3.44e-14 ***
## sc1           0.0008401  0.0008385   1.002  0.31636
## sc2           0.0038956  0.0008274   4.708 2.50e-06 ***
## dbp           0.0029950  0.0031807   0.942  0.34639
## sbp           0.0043589  0.0016917   2.577  0.00998 **
## smoke         0.0041837  0.0022392   1.868  0.06171 .
## bmi           0.0304647  0.0064338   4.735 2.19e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(6325.893) family taken to be 1)
##
##      Null deviance: 3707.9  on 5208  degrees of freedom
## Residual deviance: 3301.0  on 5200  degrees of freedom

```

```
## AIC: 6219.2
##
## Number of Fisher Scoring iterations: 1
##
##
##          Theta: 6326
##        Std. Err.: 12102
## Warning while fitting theta: iteration limit reached
##
## 2 x log-likelihood: -6199.204
exp(0.0257415)
## [1] 1.026076
```