HW2

Wei_Wang_ww5

March 15, 2019

Name: Wei Wang andrew ID: ww5

Part 1: Randomized trial for strokes (12 points)

Preliminaries

1a.

- **Y:** What was the definition of the primary outcome in this study?
- What is (are) the variable name(s) for the outcome?

Answer: The definition of the primary outcome is death within 14 days and death or dependency at 6 months The variable names of the outcome is: **DDEAD(Dead on discharge form)** and **FDEAD(Dead at six month follow-up (Y/N))**

- **U:** what is (are) the variable name(s) for the intervention, and what is (are) their possible values?

The intervention in this study include the what descibe below: Half the patients were allocated unfractionated heparin (5000 or 12500 IU bd [twice daily]) and half were allocated avoid heparin; and, in a factorial design, half were allocated aspirin 300 mg daily and half avoid aspirin?.

Therefore the variables include: - RHEP24 Heparin within 24 hours prior to randomisation (Y/N) - RASP3 Aspirin within 3 days prior to randomisation (Y/N) - DASP14 Aspirin given for 14 days or till death or discharge (Y/N) - DASPLT Discharged on long term aspirin (Y/N) - DLH14 Low dose heparin given for 14 days or till death/discharge (Y/N) - DMH14 Medium dose heparin given for 14 days or till death/discharge (Y/N) - DHH14 Medium dose heparin given for 14 days etc in pilot (combine with above)

- **V, W:** describe the covariates included and the population being studied. Be specific where possible.

Covariates can be the the patients' personal information including Age, sex, and the Final diagnosis of initial event DDIAGISC Ischaemic stroke; DDIAGHA Haemorrhagic stroke; DDIAGUN Indeterminate stroke; DNOSTRK Not a stroke; DNOSTRKX Comment on above.

The population of the study is: 19435 patients with suspected acute ischaemic stroke entering 467 hospitals in 36 countries were randomised within 48 hours of symptom onset.

1b. Provide descriptive statistics for in groups of {aspirin, no aspirin} use, including information on age, gender, systolic blood pressure, and conscious state. In clinical literature, this information is often referred to as "Table 1".

```
#set environment
loadlibs = function(libs) {
  for(lib in libs) {
    class(lib)
    if(!do.call(require,as.list(lib))) {install.packages(lib)}
    do.call(require,as.list(lib))
  }
}
libs = c("dplyr","mice","randomForest","adabag","gbm","pROC","gsubfn","chron")
loadlibs(libs)
```

```
## Loading required package: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##
## 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
```

```
## Loading required package: mice
```

```
## Warning: package 'mice' was built under R version 3.5.2
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'mice'
```

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

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 3.5.2
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## Loading required package: adabag
```

```
## Warning: package 'adabag' was built under R version 3.5.3
```

```
## Loading required package: rpart
```

```
## Warning: package 'rpart' was built under R version 3.5.2
```

```
## Loading required package: caret
```

```
## Warning: package 'caret' was built under R version 3.5.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
## Loading required package: foreach
```

```
## Loading required package: doParallel
```

```
## Warning: package 'doParallel' was built under R version 3.5.3
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
## Loading required package: gbm
```

```
## Warning: package 'gbm' was built under R version 3.5.3
```

```
## Loaded gbm 2.1.5
```

```
## Loading required package: pROC
```

```
## Warning: package 'pROC' was built under R version 3.5.2
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
## Loading required package: gsubfn
```

```
## Warning: package 'gsubfn' was built under R version 3.5.3
```

```
## Loading required package: proto
```

```
## Warning: package 'proto' was built under R version 3.5.3
```

```
## Loading required package: chron
```

```
## Warning: package 'chron' was built under R version 3.5.3
```

```
##
## Attaching package: 'chron'
```

```
## The following object is masked from 'package:foreach':
##
##     times
```

```
#read dataset
stroke=read.csv("IST_corrected.csv",header = TRUE)
stroke=stroke[stroke$DASP14!="n",]
stroke=stroke[stroke$DASP14!="y",]
```

statistics description group by

```
aggregate( . ~ RASP3, data=stroke[c("AGE","RCONSC","RSBP","RASP3")], FUN=range)
```

```
##   RASP3 AGE.1 AGE.2 RCONSC.1 RCONSC.2 RSBP.1 RSBP.2
## 1          23    99        1        3     90    240
## 2     N    16    98        1        3     70    290
## 3     Y    26    98        1        3     71    295
```

```
aggregate( . ~ RASP3, data=stroke[c("AGE","RSBP","RASP3")], FUN=mean)
```

```
##   RASP3      AGE     RSBP
## 1       69.96334 159.3564
## 2     N 71.41541 160.2930
## 3     Y 73.25863 159.8614
```

```
stroke %>%
  group_by(RASP3,SEX) %>%
  summarise(Freq = n()) %>%
  filter(RASP3!="")
```

```
## # A tibble: 4 x 3
## # Groups:   RASP3 [2]
##   RASP3 SEX    Freq
##   <fct> <fct> <int>
## 1 N     F      6879
## 2 N     M      7632
## 3 Y     F      1723
## 4 Y     M      2217
```

```
stroke %>%
  group_by(RASP3,RCONSC) %>%
  summarise(Freq = n()) %>%
  filter(RASP3!="")
```

```
## # A tibble: 6 x 3
## # Groups:   RASP3 [2]
##   RASP3 RCONSC  Freq
##   <fct> <fct>  <int>
## 1 N     D       3232
## 2 N     F      11066
## 3 N     U        213
## 4 Y     D        824
## 5 Y     F       3080
## 6 Y     U         36
```

Machine learning analysis

```
# split the dataset into 50-50
smp_size <- floor(0.5 * nrow(stroke))
## set the seed to make your partition reproducible
set.seed(123)
train_stroke <- sample(seq_len(nrow(stroke)), size = smp_size)
train <- stroke[train_stroke, ]
test <- stroke[-train_stroke, ]
```

1c. Let our outcome of interest be “dead or dependent at 6 months”, i.e. so that we have a binary classification problem. What percent of patients are dead or dependent at 6 months in your train set and test set?

The variable is chosen to be FDEAD

```
length(train[train$FDEAD=="Y",])/length(train$FDEAD)
```

```
## [1] 0.01152738
```

```
length(test[test$FDEAD=="Y",])/length(test$FDEAD)
```

```
## [1] 0.01152619
```

1d. Choose which variables to include in your model.

The below variables should be removed:

DDEAD, DDEADD, DDEADC, DDEADX, DALIVE, DALIVED, DPLACE, these are all the 14 days variables.

FDEADX, comment on death is unrelated

```
#delete columns useless
delete=c("FDEADX","DDEAD", "DDEADD", "DDEADC", "DDEADX", "DALIVE", "DALIVED", "DPLACE","COUNTRY"
,
         "CNTRYNUM","NCCODE","RDATE","DSIDEX","DNOSTRKX","DMAJNCHX",
         "HOURLOCAL","MINLOCAL","DAYLOCAL","FSOURCE","ID","EXPDD","EXPD14","EXPD6","SET14D",
         "ID14","OCCODE","SETASPLT","ID","DIED","DEAD1","DEAD2","DEAD3","DEAD4","DEAD5","DEAD6",
"DEAD7","DEAD8",
         "FDEADC","FDENNIS","FPLACE",
         "RDEF1","RDEF2","RDEF3","RDEF4","RDEF5","RDEF6","RDEF7","RDEF8")

train=train[, !(colnames(train) %in% delete), drop=FALSE]
test=test[, !(colnames(test) %in% delete), drop=FALSE]
```

1e. Of the remaining variables, decide whether to exclude variables with missing data, impute them, and/or use indicator variables.

data clean and imputate

```
#check number of Nans in the dataset
null=sapply(train, function(x) sum(is.na(x)))
null=null[null>0]
null
```

```
##   ONDRUG DMAJNCHD   DSIDED  DRSISCD    DRSHD  DRSUNKD     DPED   FLASTD
##        1     9636     9386     9503     9671     9598     9650     9687
##   FDEADD FU1_RECD FU2_DONE FU1_COMP       TD
##     7554        8       48      237        1
```

As we can see from the result, the above columns have high number of NA values, For the date that is missing mostly, I remove the columns. For column **FDEADC**, it relates to the cause of death, for NA record, I replace them with 0, which refers to “unknown”. Column **FU1_RECD, FU2_DONE, FU1_COMP, TD** can be impute since it shows some correlation with the outcome

```
delete2=c("DMAJNCHD", "DSIDED","DRSISCD","DRSHD","DRSUNKD","DPE","DPED","FLASTD","FDEADD","DRSUN
K",
          "FAP","FAP","FRECOVER","FOAC","CMPLHEP","DRSH","DNOSTRKX","TD")
train=train[, !(colnames(train) %in% delete2), drop=FALSE]
test=test[, !(colnames(test) %in% delete2), drop=FALSE]
#train$FDEADC <- replace(train$FDEADC,is.na(train$FDEADC),0)
#test$FDEADC <- replace(test$FDEADC,is.na(test$FDEADC),0)
train$ONDRUG <- replace(train$ONDRUG,is.na(train$ONDRUG),0)
test$ONDRUG <- replace(test$ONDRUG,is.na(test$ONDRUG),0)

#imputate train data
mtrain = mice(train %>%
              select(c("FU1_RECD","FU2_DONE","FU1_COMP")) %>%
              mutate_if(is.character, as.factor),m=1,maxit = 1)
```

```
##
##  iter imp variable
##   1   1  FU1_RECD  FU2_DONE  FU1_COMP
```

```
train_tmp = mice::complete(mtrain) %>% as_tibble()
# Rename columns to indicate imputation
names(train_tmp) = lapply(names(train_tmp), paste0, "_imputed")
```

```
## Warning: Must use a character vector as names.
## This warning is displayed once per session.
```

```
#imputate test data
mtest = mice(test %>%
                select(c("FU1_RECD","FU2_DONE","FU1_COMP")) %>%
                mutate_if(is.character, as.factor),m=1,maxit = 1)
```

```
##
##  iter imp variable
##   1   1  FU1_RECD  FU2_DONE  FU1_COMP
```

```
test_tmp = mice::complete(mtest) %>% as_tibble()
# Rename columns to indicate imputation
names(test_tmp) = lapply(names(test_tmp), paste0, "_imputed")

#reform the two dataset
delete3=c("FU1_RECD","FU2_DONE","FU1_COMP","TD","ONDRUG")
train=train[, !(colnames(train) %in% delete3), drop=FALSE]
test=test[, !(colnames(test) %in% delete3), drop=FALSE]
train=cbind(train,train_tmp)
test=cbind(test,test_tmp)

#clean dependent variable
train=train[train$FDEAD!="U",]
train=train[train$FDEAD!="",]
train$FDEAD=droplevels(train$FDEAD)

test=test[test$FDEAD!="U",]
test=test[test$FDEAD!="",]
test$FDEAD=droplevels(test$FDEAD)
```

logistic regression

```
mylogit <- glm(FDEAD ~., data = train,family = "binomial")
```

random forest

```
rf <- randomForest(FDEAD ~ ., data = train, ntree = 500, mtry = 6, importance = TRUE)
summary(rf)
```

```
##                Length Class  Mode
## call               6  -none- call
## type               1  -none- character
## predicted       9640  factor numeric
## err.rate        1500  -none- numeric
## confusion          6  -none- numeric
## votes          19280  matrix numeric
## oob.times       9640  -none- numeric
## classes            2  -none- character
## importance       200  -none- numeric
## importanceSD     150  -none- numeric
## localImportance    0  -none- NULL
## proximity          0  -none- NULL
## ntree              1  -none- numeric
## mtry               1  -none- numeric
## forest            14  -none- list
## y               9640  factor numeric
## test               0  -none- NULL
## inbag              0  -none- NULL
## terms              3  terms  call
```

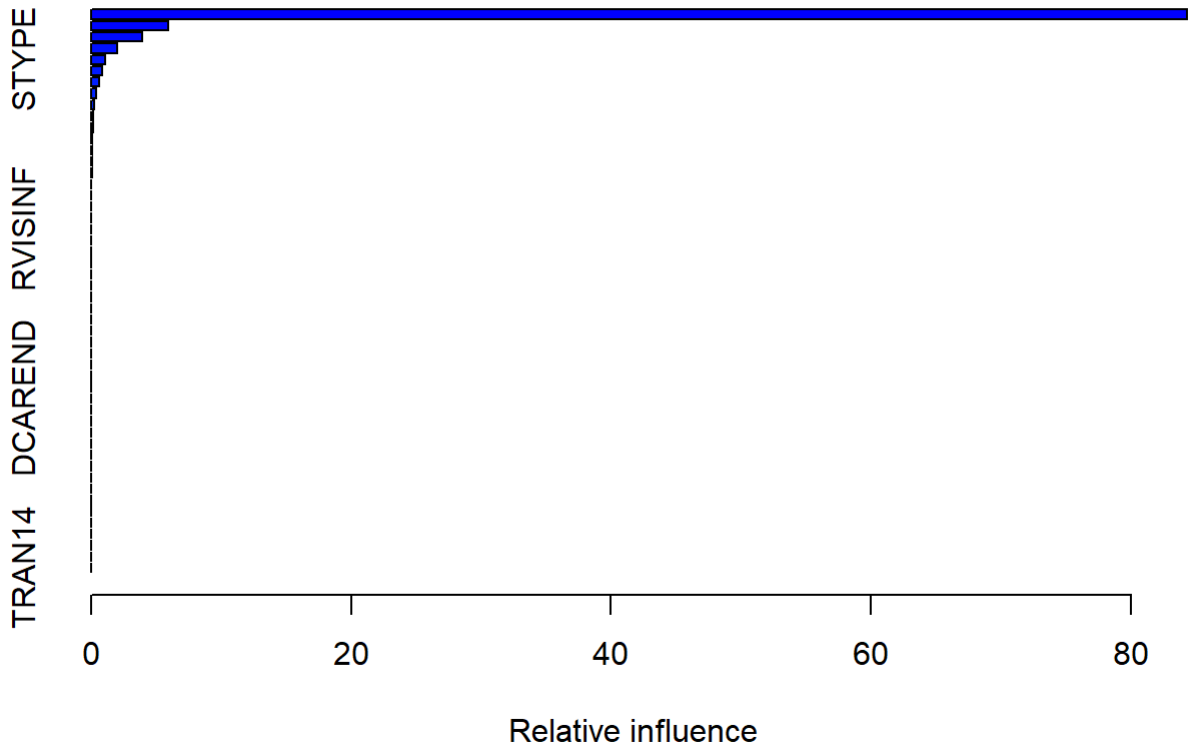
boosting

```
boost <- boosting(FDEAD~., data=train, boos=TRUE, mfinal=3)
summary(boost)
```

```
##            Length Class   Mode
## formula         3  formula call
## trees           3  -none-  list
## weights         3  -none-  numeric
## votes       19280  -none-  numeric
## prob        19280  -none-  numeric
## class        9640  -none-  character
## importance     50  -none-  numeric
## terms           3  terms   call
## call            5  -none-  call
```

gradient boosting

```
gbm <- gbm(FDEAD~., data=train,distribution = "gaussian")
#n.trees = 10000,shrinkage = 0.01, interaction.depth = 4
summary(gbm) #Summary gives a table of Variable Importance and a plot of Variable Importance
```


STYPE
RVISINF
DCAREND
TRAN14
0
20
40
60
80
Relative influence

```
##                                 var     rel.inf
## FU2_DONE_imputed FU2_DONE_imputed 84.31356287
## AGE                             AGE  5.96605271
## RCONSC                       RCONSC  3.95355940
## DASPLT                       DASPLT  1.99219721
## STYPE                         STYPE  1.07983852
## FU1_RECD_imputed FU1_RECD_imputed  0.81934554
## RATRIAL                     RATRIAL  0.64841877
## DSTER                         DSTER  0.39546211
## NCB14                         NCB14  0.22689773
## DOAC                           DOAC  0.18157632
## STRK14                       STRK14  0.14911278
## HOSPNUM                     HOSPNUM  0.09451329
## CMPLASP                     CMPLASP  0.07504924
## DLH14                         DLH14  0.05239971
## FU1_COMP_imputed FU1_COMP_imputed  0.05201380
## RDELAY                       RDELAY  0.00000000
## SEX                             SEX  0.00000000
## RSLEEP                       RSLEEP  0.00000000
## RCT                             RCT  0.00000000
## RVISINF                     RVISINF  0.00000000
## RHEP24                       RHEP24  0.00000000
## RASP3                         RASP3  0.00000000
## RSBP                           RSBP  0.00000000
## RXASP                         RXASP  0.00000000
## RXHEP                         RXHEP  0.00000000
## DASP14                       DASP14  0.00000000
## DMH14                         DMH14  0.00000000
## DHH14                         DHH14  0.00000000
## DSCH                           DSCH  0.00000000
## DIVH                           DIVH  0.00000000
## DAP                             DAP  0.00000000
## DGORM                         DGORM  0.00000000
## DCAA                           DCAA  0.00000000
## DHAEMD                       DHAEMD  0.00000000
## DCAREND                     DCAREND  0.00000000
## DTHROMB                     DTHROMB  0.00000000
## DMAJNCH                     DMAJNCH  0.00000000
## DSIDE                         DSIDE  0.00000000
## DDIAGISC                   DDIAGISC  0.00000000
## DDIAGHA                     DDIAGHA  0.00000000
## DDIAGUN                     DDIAGUN  0.00000000
## DNOSTRK                     DNOSTRK  0.00000000
## DRSISC                       DRSISC  0.00000000
## H14                             H14  0.00000000
## ISC14                         ISC14  0.00000000
## NK14                           NK14  0.00000000
## HTI14                         HTI14  0.00000000
## PE14                           PE14  0.00000000
## DVT14                         DVT14  0.00000000
## TRAN14                       TRAN14  0.00000000
```

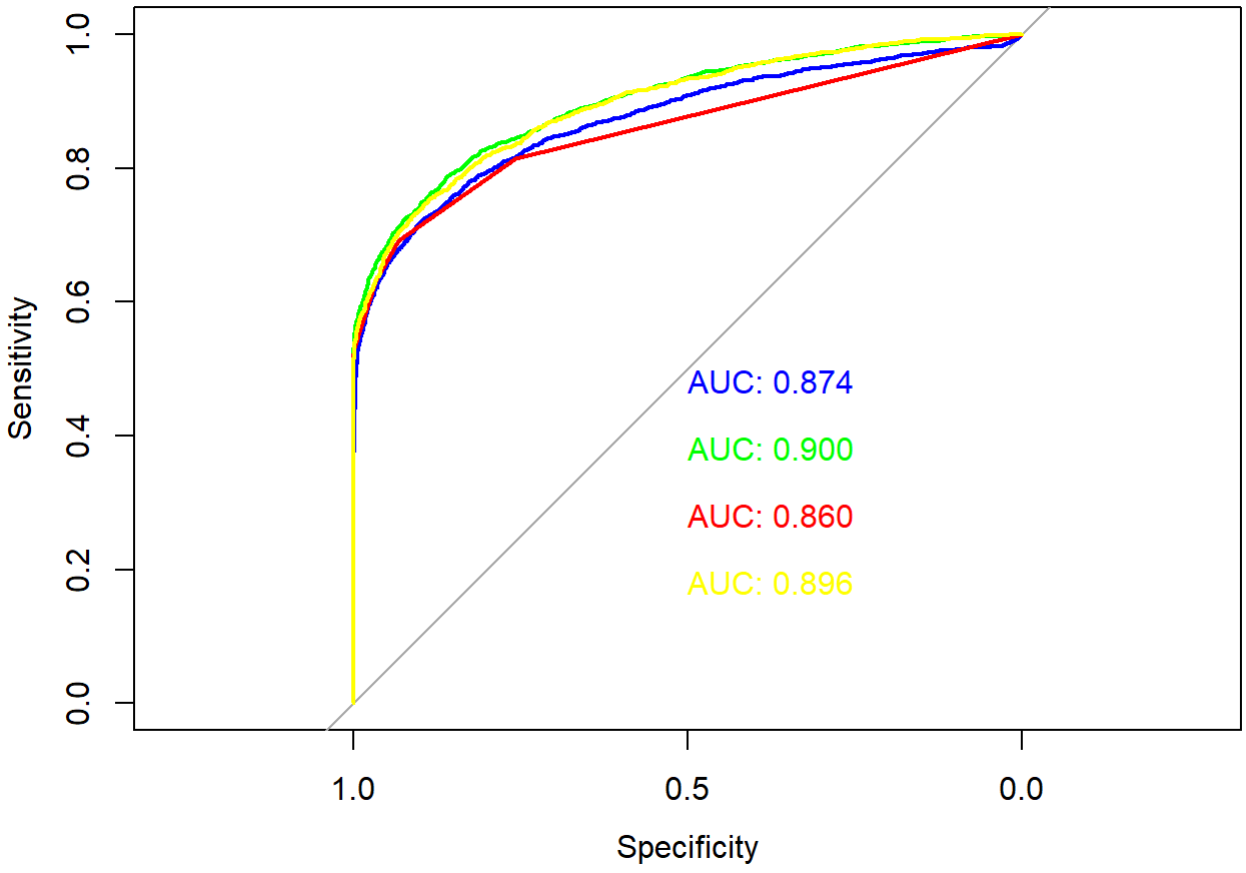
1f. Construct an ROC

```
#ROC for multiple model
rf_pred=data.frame(predict(rf, test, type="prob"))
lg_pre=predict(mylogit,newdata=test,type=c("response"))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
boost_pred=predict(boost,test,type=c("response"))
boost_pred=data.frame(boost_pred$prob)
boost_pred=boost_pred$X2
gbm_pred=predict(gbm,test,n.trees = 100,type = c("response"))

roc_rose <- plot(roc(test$FDEAD,lg_pre), print.auc = TRUE, col = "blue")
roc_rose <- plot(roc(test$FDEAD, rf_pred$Y), print.auc = TRUE,
                 col = "green", print.auc.y = .4, add = TRUE)
roc_rose <- plot(roc(test$FDEAD,boost_pred), print.auc = TRUE,
                 col = "red", print.auc.y = .3, add = TRUE)
roc_rose <- plot(roc(test$FDEAD,gbm_pred), print.auc = TRUE,
                 col = "yellow", print.auc.y = .2, add = TRUE)
```



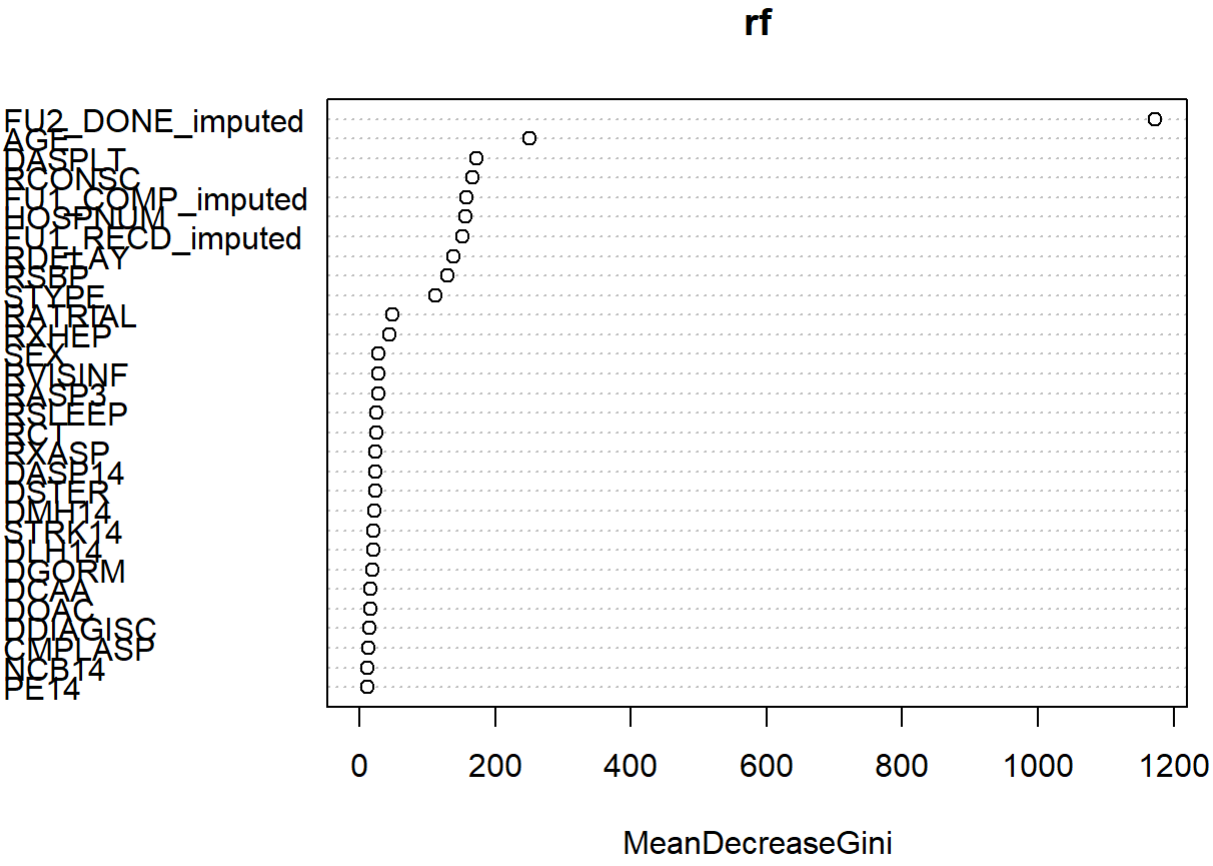
1g. Report the variable importance

```
# variable importance
importance(rf)
```

```
##                           N           Y MeanDecreaseAccuracy
## HOSPNUM           12.2279584   0.1850410           11.1099578
## RDELAY             7.1411730  -1.5067412            5.3082822
## RCONSC            34.0506511  22.5751904           40.0627698
## SEX                1.8679826   4.6228200            3.8407049
## AGE               30.9330645  41.1864641           47.7634565
## RSLEEP             2.4280368  -0.4270501            1.8653519
## RATRIAL           19.3733298   5.2236866           19.8703742
## RCT                2.0936229   5.3910411            4.8315323
## RVISINF            5.2113845   1.7848778            5.6058717
## RHEP24             4.2023382  -1.1145011            3.3037225
## RASP3              3.3429257   0.1177898            3.0212864
## RSBP               7.2336053   2.9784306            7.4755802
## STYPE             21.7444110   8.7096844           23.1400454
## RXASP              6.7784216  -1.1269709            5.9885583
## RXHEP              9.2388762  -2.6926328            7.9722607
## DASP14             5.6125150  -1.4783907            4.8843067
## DASPLT            23.1956924  26.1407932           32.4278481
## DLH14              8.7476914   0.2355522            8.8053785
## DMH14              5.6060832  -0.4837211            5.4167011
## DHH14              3.6575119   1.9410155            4.2115200
## DSCH               5.3619374   1.3592752            5.6592254
## DIVH               7.9306661  -1.5779393            7.0987573
## DAP                0.1589292   1.5974170            0.8902654
## DOAC              17.9892238   8.4211391           18.3978770
## DGORM              6.7530493  -0.1150492            6.1521469
## DSTER             12.5819108   7.6089452           14.3921243
## DCAA               2.2255686  -0.1368826            1.9973161
## DHAEMD             3.3007849   1.1212639            3.4355556
## DCAREND            3.8541336   0.3748510            3.8654188
## DTHROMB            1.5573549  -0.4816543            1.2710603
## DMAJNCH            5.4891457  -2.7526594            4.6850150
## DSIDE              8.2024467  -1.9052299            7.4681135
## DDIAGISC           2.1415873   1.3954996            2.6354260
## DDIAGHA            6.1330119  -2.2682072            4.9100659
## DDIAGUN           -0.3890229   1.4134316            0.5023799
## DNOSTRK            0.8164026   6.6539994            4.6501371
## DRSISC             9.1938318  -4.0552842            8.2336049
## CMPLASP            0.7907160   2.5486294            1.9704884
## H14                5.5512914  -1.3211230            4.7161937
## ISC14              7.1084700  -0.9924735            5.9992698
## NK14               7.9299684  -1.2469046            6.8882488
## STRK14            10.1003814   0.3261618           10.3587582
## HTI14              1.6723738  -0.6370346            1.1619665
## PE14              12.0565381   1.9522433           12.2006570
## DVT14              1.6115940  -0.8716831            1.1066048
## TRAN14            11.7287071   3.0880453           12.4713805
## NCB14             11.1733332   1.8012520           11.5412471
## FU1_RECD_imputed  19.6975837   1.8621542           19.7854445
## FU2_DONE_imputed 102.7271640 109.9301463          117.5243114
## FU1_COMP_imputed  23.4417308   5.0252253           24.8078678
##                  MeanDecreaseGini
## HOSPNUM                155.820112
```

```
## RDELAY                137.758996
## RCONSC                165.804265
## SEX                    28.291380
## AGE                   250.890221
## RSLEEP                 25.327855
## RATRIAL                48.510181
## RCT                    24.805335
## RVISINF                27.323369
## RHEP24                  9.082253
## RASP3                  27.122339
## RSBP                  129.151219
## STYPE                 111.210348
## RXASP                  23.704629
## RXHEP                  44.572879
## DASP14                 23.128638
## DASPLT                171.963370
## DLH14                  20.115006
## DMH14                  22.274594
## DHH14                   4.064133
## DSCH                   10.413337
## DIVH                    7.790697
## DAP                     6.814847
## DOAC                   15.396989
## DGORM                  19.164278
## DSTER                  22.943997
## DCAA                   15.686168
## DHAEMD                 10.330789
## DCAREND                 3.535490
## DTHROMB                 1.568336
## DMAJNCH                 3.505411
## DSIDE                   9.874953
## DDIAGISC               14.856341
## DDIAGHA                 8.031292
## DDIAGUN                 9.574251
## DNOSTRK                 7.140119
## DRSISC                  6.315314
## CMPLASP                12.807783
## H14                     4.524180
## ISC14                   4.924060
## NK14                    9.135068
## STRK14                 20.578361
## HTI14                   1.339043
## PE14                   11.835076
## DVT14                   1.441072
## TRAN14                  6.633501
## NCB14                  11.992205
## FU1_RECD_imputed      152.362931
## FU2_DONE_imputed     1172.662908
## FU1_COMP_imputed      158.004605
```

```
library(caret)
varImpPlot(rf,type=2)
```



Part 2. Basis functions and regularization for daily glucoses [8 points]

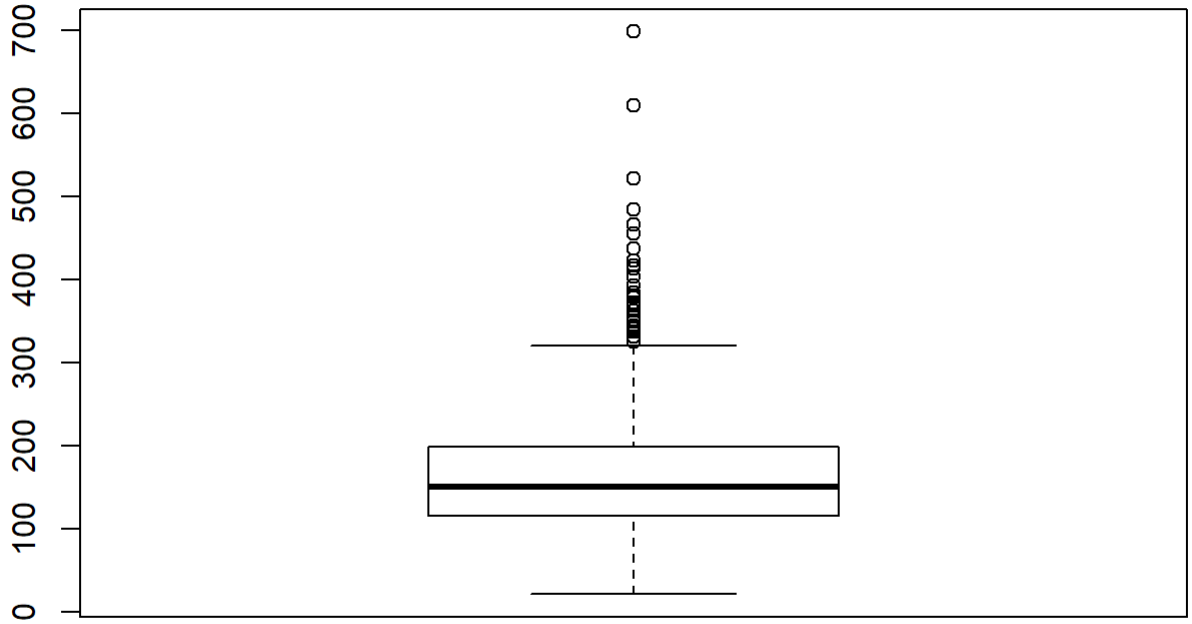
read dataset

```
#read data
item=read.csv("d_labitems.csv",header = TRUE)
events=read.csv("labevents.csv",header = TRUE)
data=events[events$subject_id=="13033",]
data=data[data$itemid=="50112",]

summary(data$valuenum)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    22.0   116.0   151.5   163.5   199.0   698.0
```

```
boxplot(data$valuenum)
```



2a. Extract the glucose data

```
library(ggplot2)
library(scales)
library(lubridate)
```

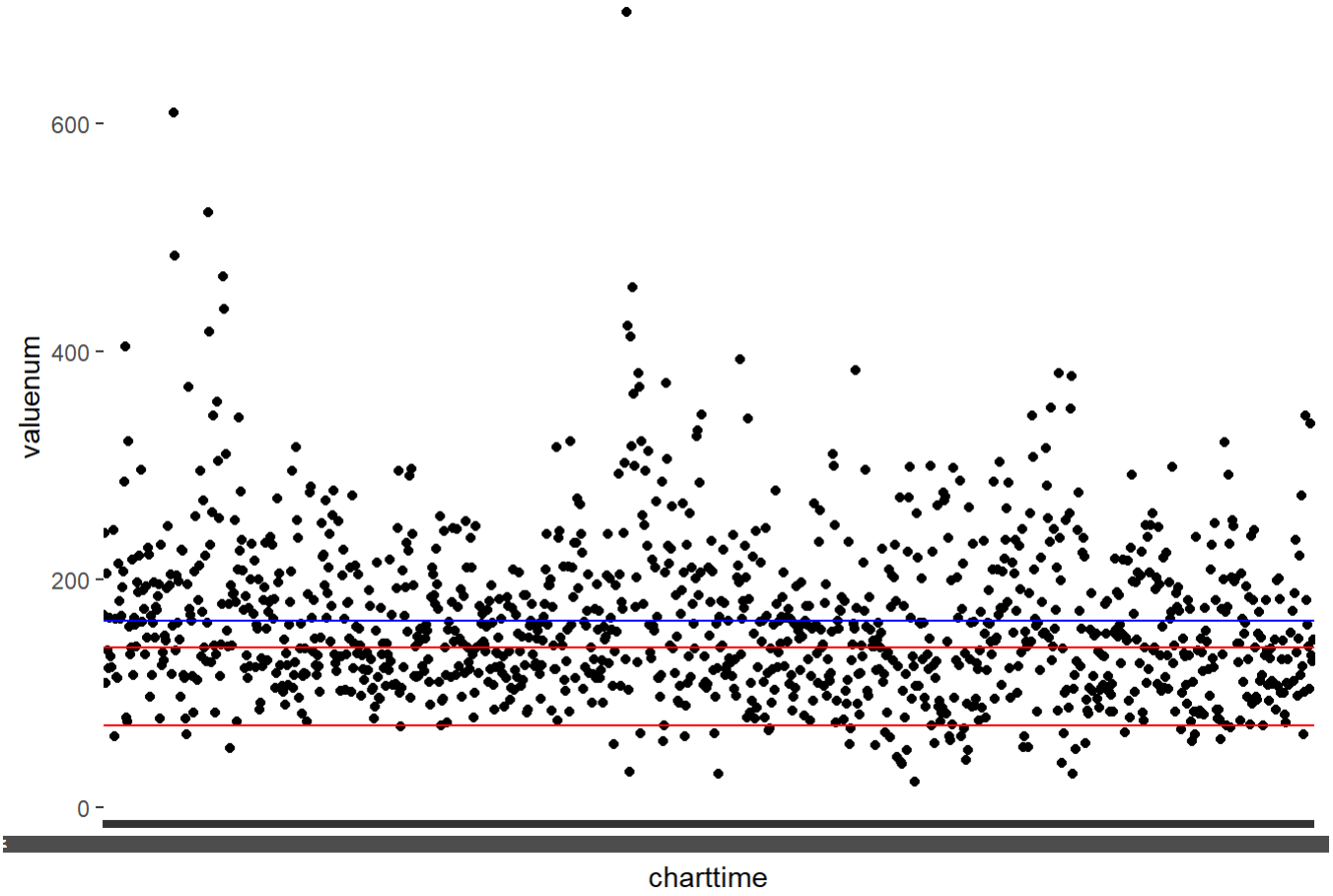
```
## Warning: package 'lubridate' was built under R version 3.5.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:chron':
##
##     days, hours, minutes, seconds, years
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
data$Date <- as.Date(data$charttime)
data$time <- times(strftime(data$charttime,"%H:%M:%S"))
ggplot(data, aes(charttime, valuenum)) +
  geom_point() +
  #scale_x_datetime(breaks=date_breaks("2 month"), labels="%b-%d-%Y") +
  #theme(axis.text.x=element_text(angle=90))+
  geom_hline(yintercept = mean(data$valuenum), color="blue")+
  geom_hline(yintercept = 72, color="red")+
  geom_hline(yintercept = 140, color="red")
```



2b. Split the data into training set and test set

I use period = 0.5, which means 12 hours (24*0.5)

```
#split data
bld_test=data[data$Date>"3417-5-1",]
bld_train=data[data$Date<="3417-5-1",]

#sine consine basic function
period = 0.5
K = 5

# Function for basis expansion of {sin(kx),cos(kx)} for i = 1 to K
sincos = function(dat, variable="time", period=2*pi, K=10) {
  data = dat
  for(i in 1:K) {
    data[[paste0("sin_",i)]] = sin(data[[variable]]*i*2*pi/period)
  }
  for(i in 1:K) {
    data[[paste0("cos_",i)]] = cos(data[[variable]]*i*2*pi/period)
  }
  # data$intercept = 1  # not necessary if using with models that use intercepts
  data
}
```

2c. Use cv.glmnet to learn a daily trend for the individual on the training set.

Plot the coefficient profile with lambda on the x-axis. Report the lambda that performed best in CV (use lambda.min for this exercise).

```
bld_train1 = sincos(bld_train, period=period, K=K)
bld_train1 = bld_train1 %>% dplyr::select(-charttime)%>% dplyr::select(-time)%>% dplyr::select(-
Date)

# Learn a linear model, regularized
library(glmnet)  # Does regularization with (generalized) linear models
```

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

```
## Loaded glmnet 2.0-16
```

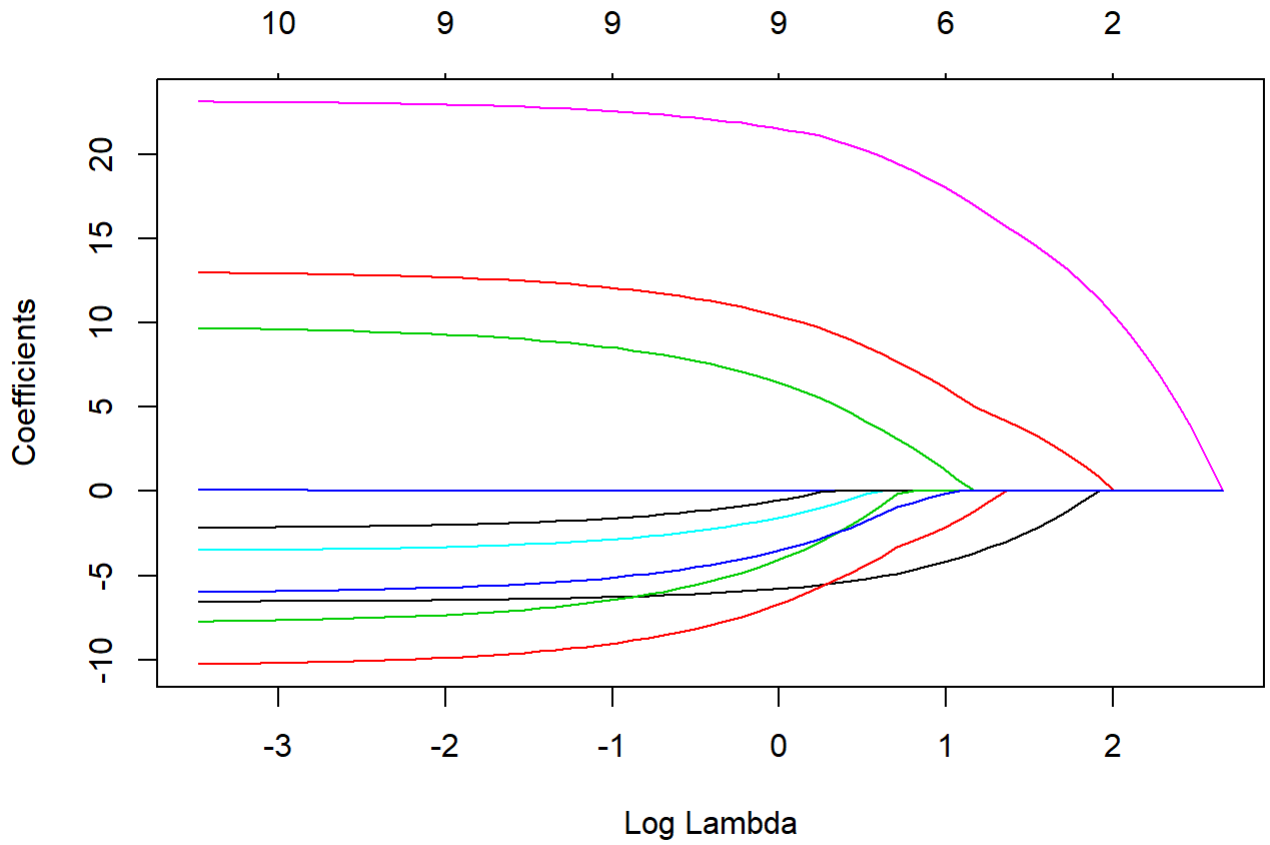
```
##
## Attaching package: 'glmnet'
```

```
## The following object is masked from 'package:pROC':
##
##     auc
```

```
lasso = cv.glmnet(x = bld_train1 %>% select(-c(subject_id,hadm_id,icustay_id,itemid,value,valuenum,valueuom,flag))
                  %>% as.matrix(),
                  y=bld_train1$valuenum, family="gaussian")
lambda = lasso$lambda.min
lambda
```

```
## [1] 0.5505187
```

```
plot(lasso$glmnet.fit, "lambda")
```



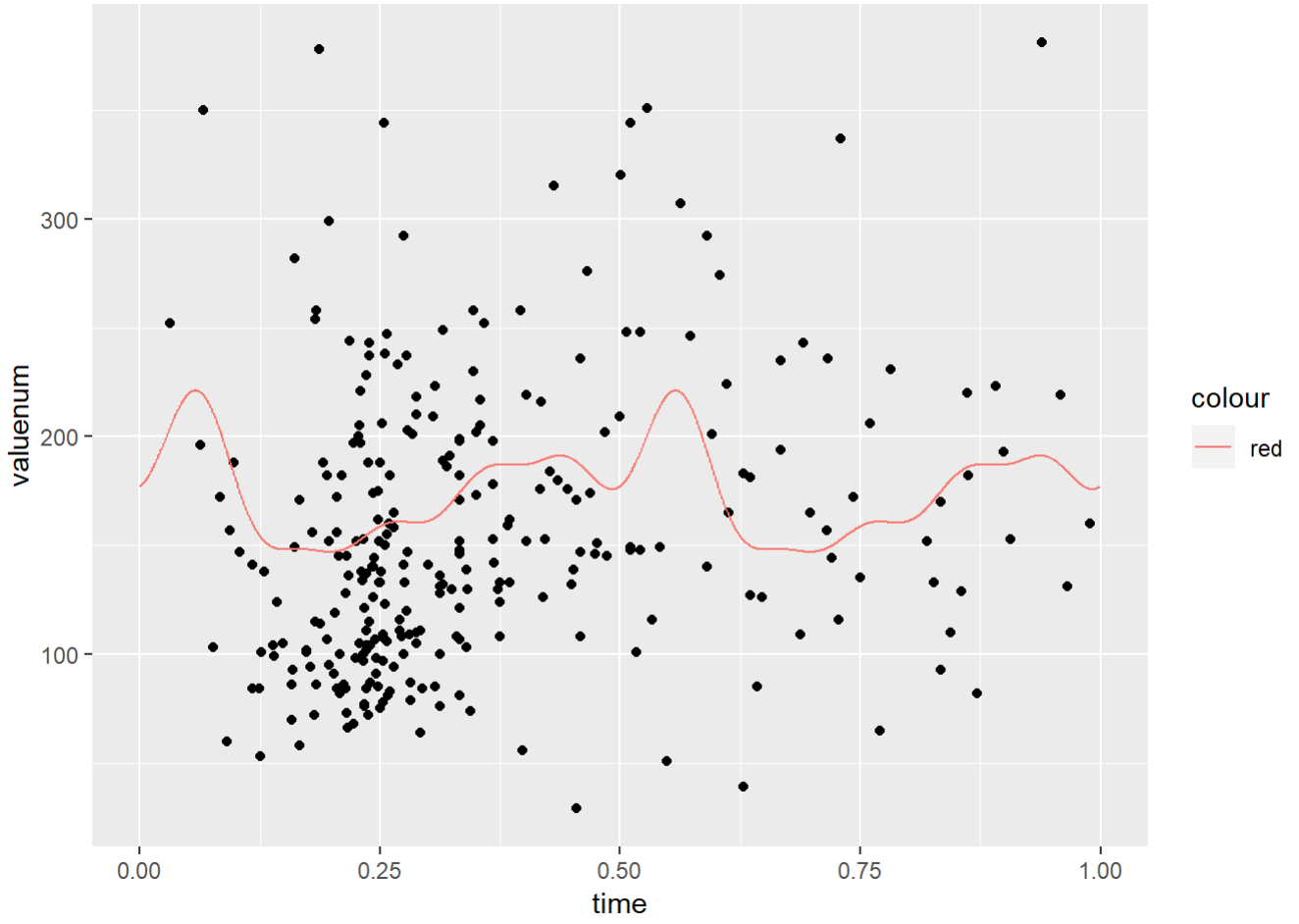
2d. plot

```
#predict
bld_test1 = sincos(bld_test, period=period, K=K)
bld_test2 = bld_test1 %>% dplyr::select(-charttime)%>% dplyr::select(-time)%>% dplyr::select(-Da
te)

self_data=data.frame(seq(ISOdate(2000,1,31), by = "min", length.out = 1440))
self_data$time=times(strftime(self_data$seq.ISOdate.2000..1..31...by....min...length.out...144
0.,"%H:%M:%S"))
self_data1=sincos(self_data, period=period, K=K)
self_data2=self_data1%>% dplyr::select(-time)%>% dplyr::select(-seq.ISOdate.2000..1..31...by....
min...length.out...1440.)

self_data1[["ylasso"]] = predict(lasso,
                                 self_data2
                                #%>% select(-c(subject_id,hadm_id,icustay_id,itemid,value,valuen
um,valueuom,flag))
                                 %>% as.matrix(),
                               s = c(lasso$lambda.min))
# Plot
ggplot(data=bld_test1, aes(time,valuenum)) + geom_point()+
  geom_line(data = self_data1, aes(x=time,y=ylasso,color="red"))
```

```
## Don't know how to automatically pick scale for object of type times. Defaulting to continuou
s.
```



2e. What percent of the variation is explained by your model compared to using the training set mean?

Based on the claculation below, around 94%

```
train_mean=mean(bld_train$valuenum)
#sum of square error
ssq=sum((bld_test$valuenum-train_mean)^2)
rsq=sum((bld_test1$valuenum-bld_test1$ylasso)^2)
r2=rsq/ssq
r2
```

```
## [1] 0
```

2f. Make a statement about the daily variation of glucose in this individual. In particular, according to your model, when is it lowest? When is it highest?

It is high in the mid night, and in the morning about 6am, it is low. Until 12:00pm, the value is slowly increasing, and drop to lowest at 6pm.