Analysis of Big Healthcare Databases - Exercises

Introduction

The goal of these exercises is to explore the structure of an electronic health records (EHR)-derived data set and some of the common challenges encountered in working with healthcare-derived data. We will also practice implementing some statistical methods introduced throughout the short course to address these issues. To do this we will use a synthetic data set simulated to mimic the structure of a real EHR-derived data set. **Please note that the data we will be working with are simulated and intended for instructional purposes only.** Real EHR data generally have access restrictions due to privacy/confidentiality issues and HIPAA protections. At the end of the course I will provide links for a few public access repositories that provide real EHR data. However, these generally do require a data use agreement and thus a few steps are involved in getting access.

The synthetic data we will be working with are based on the PEDSnet study of pediatric type 2 diabetes described in class. The data are divided into four files which can be downloaded from GitHub. The four files contain data from 9,930 patients age 10-20 years who had at least one outpatient encounter between 2001 and 2019. The four files can be linked using the variable patientid.

Encounter data

This data set includes one row per outpatient encounter.

```
encounter = read.csv("https://raw.githubusercontent.com/rhubb/ASA_EHR_ShortCourse/master/data/encounter.csv", head=T)
```

patientid: Patient ID

servicedate: Date of the encounter

age: Age in years

race: Provider-reported patient race

proc: CPT codes for procedure performed; codes 99211-99215 are for evaluation and management of established patients in an outpatient setting

diag: ICD-9 (on or before 10/31/2015) or ICD-10 (after 10/31/2015) for primary diagnosis; a few diagnosis codes of interest are T2DM ICD-9 = “250.00”; T2DM ICD-10 = “E11.9”; T1DM ICD-9 = “250.01”; T1DM ICD-10 = “E10.9”; Depression ICD-9 = “296.2”,“296.9”,“296.3”,“300.4”; Depression ICD-10 = “F32.9”,“F41.8”,“F33.9”

prov: CMS provider specialty code; a few provider codes of potential interest are General Practice = “1”, General Dermatology = “7”, Family practice = “8”, Internal Medicine = “11”, Neurology = “13”, Ophthalmology = “18”, Psychiatry = “26”, Pediatric Medicine = “37”, Endocrinology = “46”

Prescription medication data

This data set includes one row per prescription recorded on the date of an outpatient encounter included in the encounters file.

```
meds = read.csv("https://raw.githubusercontent.com/rhubb/ASA_EHR_ShortCourse/master/data/meds.csv", head=T)
```

patientid: Patient ID

presdate: Date of the prescription

drug: Drug class

Measures data

This data set includes one row per anthropometric or laboratory measurement recorded on the date of an outpatient encounter included in the encounters file.

```
measures = read.csv("https://raw.githubusercontent.com/rhubb/ASA_EHR_ShortCourse/master/data/measures.csv", head=T)
```

patientid: Patient ID

service: Date of measurement

measurement: Numeric value of the measurement

measuretype: Description of the laboratory or anthropometric test (height in cm, weight in kg, glucose in mg/dl, hemoglobin A1c (hba1c) in %)

Validation data

This data set includes one row per patient for 998 patients randomly selected for manual chart review to determine gold-standard type 2 diabetes status.

```
validation = read.csv("https://raw.githubusercontent.com/rhubb/ASA_EHR_ShortCourse/master/data/validation.csv", head=T)
```

patientid: Patient ID

T2DMv: Type 2 diabetes status based on manual chart review (1 = T2DM, 0 = no evidence of T2DM)

Install R packages

- For these exercises you will need the *rpart*, *pROC*, *boot*, and *gee* packages.
- If you have not already, please install these packages now.

```
install.packages("rpart")
install.packages("pROC")
install.packages("boot")
install.packages("gee")
library(rpart)
library(pROC)
library(boot)
library(gee)
```

Exercises

1. **Data Quality Evaluation.** The first task in analysis of EHR data is data exploration and visualization to identify and resolve data errors. Using the measures data set, we will carry out a descriptive analysis. Are there any observations that seem likely to be errors? What are some techniques we can use to identify errors? What are some options for

handling possibly erroneous data points once they have been identified?

```
## Use summary statistics and plots to investigate basic characteristics of the data

summary(measures)
```

```
##    patientid          servicedate     measurement      measuretype
##  Min.   :100172   2015-10-04:   40   Min.   : -0.10   chol   : 2543
##  1st Qu.:320683   2010-01-23:   39   1st Qu.: 88.27   glucose: 6387
##  Median :546440   2005-09-30:   37   Median :131.50   hba1c  : 4500
##  Mean   :545570   2015-04-19:   37   Mean   :127.82   height :30484
##  3rd Qu.:773212   2009-06-04:   35   3rd Qu.:163.68   weight :30484
##  Max.   :999973   2009-08-03:   35   Max.   :935.96
##                   (Other)   :74175
```

```
# check types of measures available
unique(measures$measuretype)
```

```
## [1] height  weight  hba1c   chol    glucose
## Levels: chol glucose hba1c height weight
```

```
# separate variables by measurement type
height <- measures[measures$measuretype == "height",-4]
names(height) <- c("patientid","servicedate","height")

weight <- measures[measures$measuretype == "weight",-4]
names(weight) <- c("patientid","servicedate","weight")

glucose <- measures[measures$measuretype == "glucose",-4]
names(glucose) <- c("patientid","servicedate","glucose")

hba1c <- measures[measures$measuretype == "hba1c",-4]
names(hba1c) <- c("patientid","servicedate","hba1c")

chol <- measures[measures$measuretype == "chol",-4]
names(chol) <- c("patientid","servicedate","chol")

# explore number of observations available per patient for each measurement type

summary(c(table(factor(height$patientid, levels = unique(encounter$patientid)))))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    3.07    4.00   38.00
```

```
summary(c(table(factor(weight$patientid, levels = unique(encounter$patientid)))))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    3.07    4.00   38.00
```

```
summary(c(table(factor(glucose$patientid, levels = unique(encounter$patientid)))))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.6432  1.0000 10.0000
```

```
summary(c(table(factor(hba1c$patientid, levels = unique(encounter$patientid)))))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4532  1.0000  6.0000
```

```
summary(c(table(factor(chol$patientid, levels = unique(encounter$patientid)))))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.2561  0.0000  4.0000
```

```
# number of children with no measures available

sum(c(table(factor(height$patientid, levels = unique(encounter$patientid)))) == 0)
```

```
## [1] 33
```

```
sum(c(table(factor(weight$patientid, levels = unique(encounter$patientid)))) == 0)
```

```
## [1] 33
```

```
sum(c(table(factor(glucose$patientid, levels = unique(encounter$patientid)))) == 0)
```

```
## [1] 5141
```

```
sum(c(table(factor(hba1c$patientid, levels = unique(encounter$patientid)))) == 0)
```

```
## [1] 6235
```

```
sum(c(table(factor(chol$patientid, levels = unique(encounter$patientid)))) == 0)
```

```
## [1] 7644
```

```
# summarize distribution of variables across all patients, looking for values outside the p
lausible range

summary(height$height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   50.82  151.69  162.80  162.56  173.50  225.06
```

```
summary(weight$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -0.10   77.87   96.10   99.92  116.03  935.96
```

```
summary(glucose$glucose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   28.06   79.03  100.75  115.97  128.67  481.01
```

```
summary(hba1c$hba1c)
```

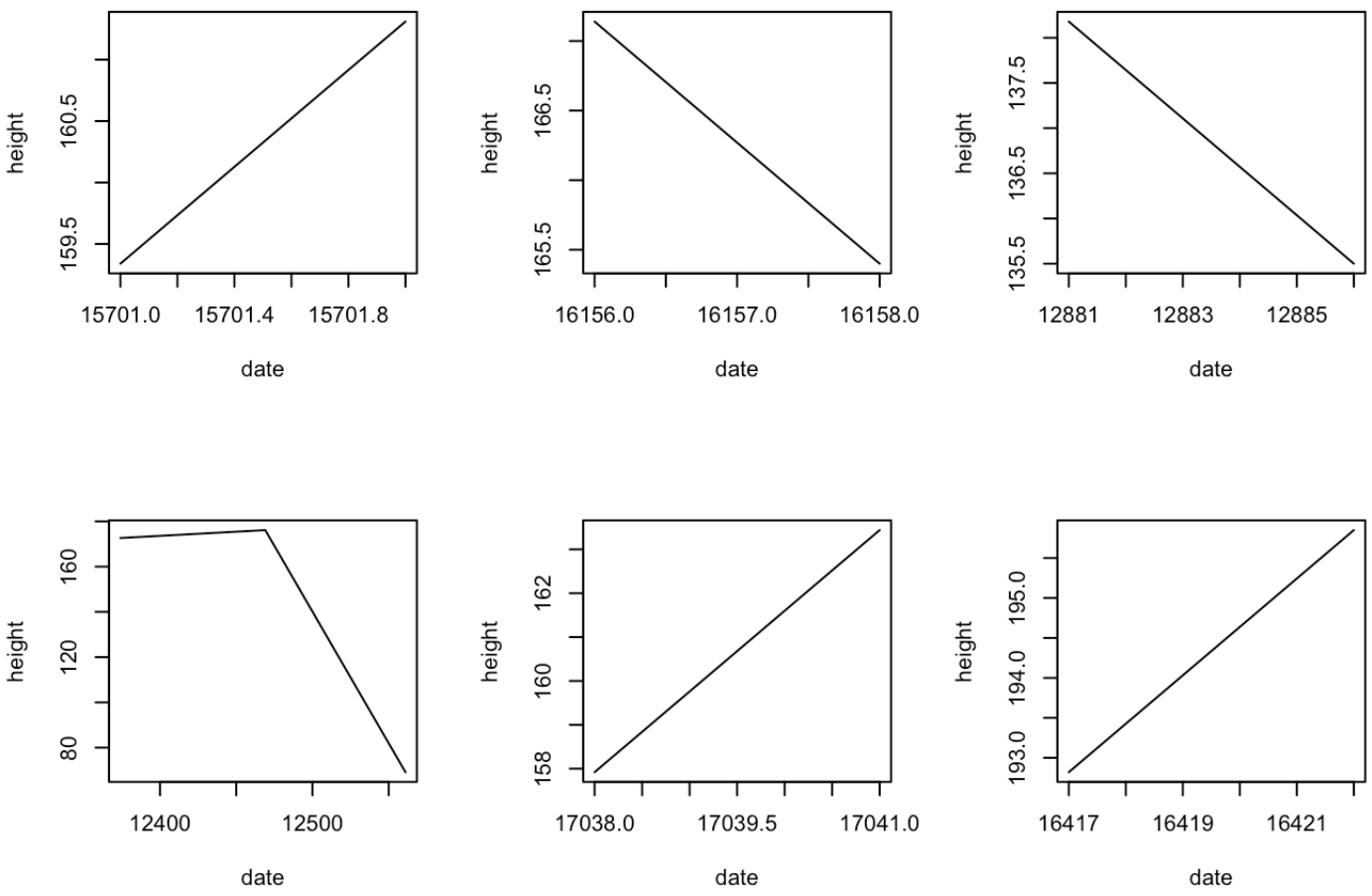
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.180   5.230   6.160   6.351   7.330  13.620
```

```
summary(chol$chol)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   41.31  187.36  328.48  290.60  373.36  519.34
```

```
# values that are clearly outside the plausible range can be eliminated, those that seem un
likely should be
# noted for discussion with clinical collaborators

# remove negative weights as clearly lying outside the plausible range

weight$weight <- ifelse(weight$weight < 0, NA, weight$weight)
```

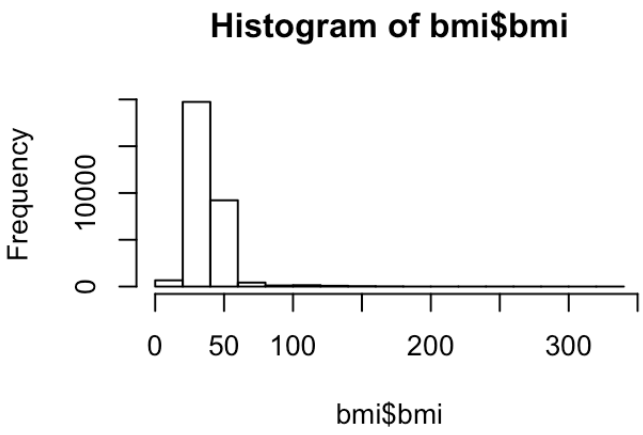
```
# identify patients with extreme heights and weights

extreme.heights <- height$patientid[height$height < 100] # flag patients with height < 1 m
extreme.weights <- weight$patientid[weight$weight > 200] # flag patients with weight > 200
kg

# implausible patterns in longitudinal measurements provide an additional means of identify
ing data errors

height.s <- split(data.frame(height$servicedate,height$height),height$patientid)
weight.s <- split(data.frame(as.Date(weight$servicedate),weight$weight),weight$patientid)
glucose.s <- split(data.frame(as.Date(glucose$servicedate),glucose$glucose),glucose$patient
id)
hba1c.s <- split(data.frame(as.Date(hba1c$servicedate),hba1c$hba1c),hba1c$patientid)

# summarize rate of change and within-patient variability

# function to estimate rate of change and residual variability for each child's data
longrate <- function(x){
  days <- as.numeric(as.Date(x[,1]))
  measure <- x[,2]
  mod <- lm(measure ~ days)
  rate <- mod$coef[2]
  residsd <- summary(mod)$sigma
  return(c(rate,residsd))
}

height.lm <- t(sapply(height.s,longrate))

# take a look at a few patients with implausible trajectories

height.change.ind <- which(abs(height.lm[,1]) > 0.5)
par(mfrow = c(2,3))
for (i in 1:6){
  plot(as.numeric(as.Date(height.s[[height.change.ind[i]]][,1])),height.s[[height.change.in
d[i]]][,2], xlab = "date", ylab = "height", type = "l")
}
```

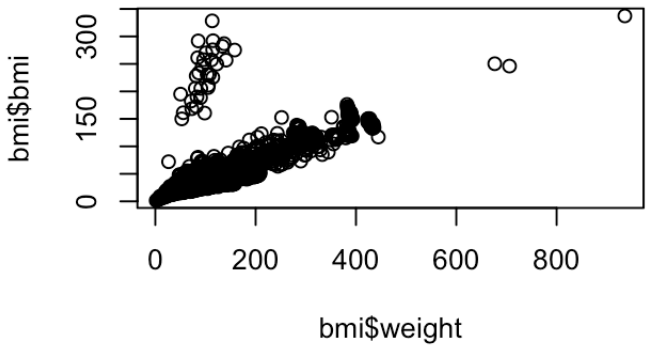


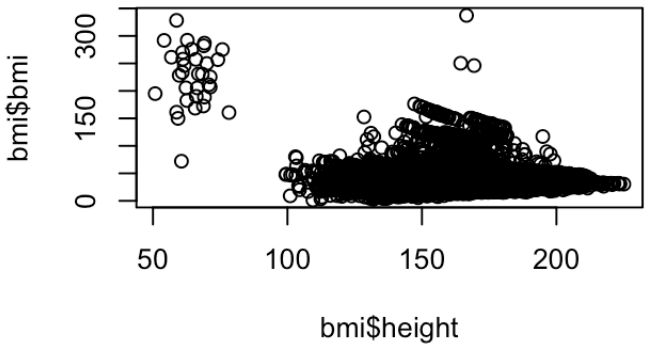

```
# a few of these measures look very suspicious, as if one measurement is about 2.5 times th
e other
# take a closer look at an example case

height[height$patientid == names(height.change.ind[4]),]
```

```
##       patientid servicedate    height
## 22268    363685  2003-11-18 172.66000
## 22271    363685  2004-02-21 176.16000
## 22273    363685  2004-05-23  69.14254
```

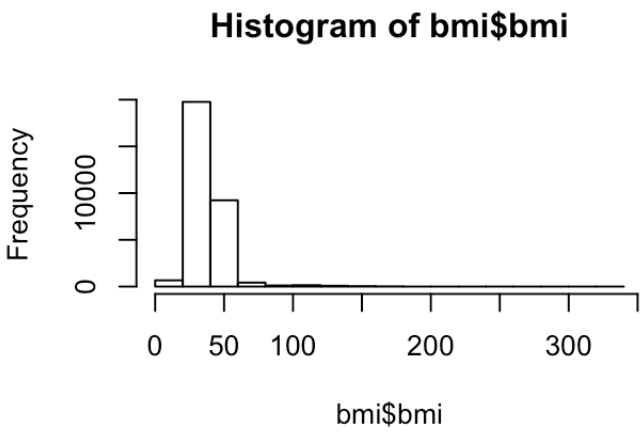
```
# generate BMI and look for implausible values

height$iddate <- paste(height$patientid,height$servicedate)
weight$iddate <- paste(weight$patientid,weight$servicedate)
bmi <- merge(height,weight,by = "iddate") # merge height and weight data
bmi$bmi <- bmi$weight/(bmi$height/100)^2

par(mfrow = c(2,2))
hist(bmi$bmi)
plot(bmi$weight,bmi$bmi)
plot(bmi$height,bmi$bmi)

# unusual groupings in BMI plots suggest patients with wrong units for height or weight
# select a rule for eliminating these heights or weights

bmi$height <- ifelse(bmi$height < 100 & bmi$bmi > 100, bmi$height*2.54, bmi$height)
bmi$bmi <- bmi$weight/(bmi$height/100)^2

par(mfrow = c(2,2))
```

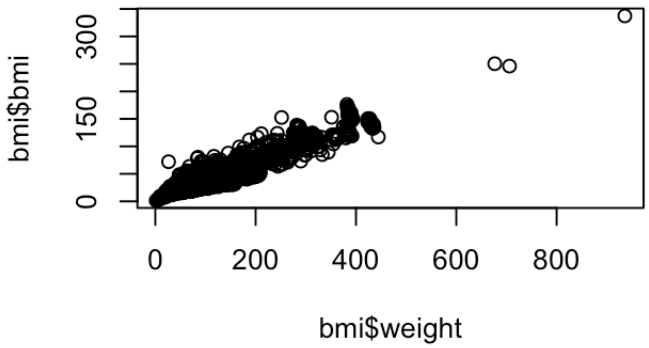


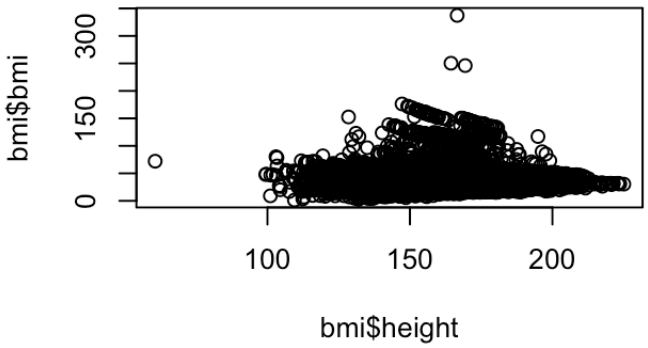





```
hist(bmi$bmi)
plot(bmi$weight,bmi$bmi)
plot(bmi$height,bmi$bmi)
```







2. **Phenotype Extraction.** We will next explore a few alternative approaches to deriving a type 2 diabetes (T2DM) phenotype from this data set. To do so, we first need to reduce the data to one observation per patient considering what data elements might be of use at the patient-level. Next we will use the validation data to develop a prediction model for T2DM using logistic regression and CART. Finally, we will apply the eMERGE T2DM rule to these data. How do the sensitivity, specificity, PPV, and NPV of these approaches compare?

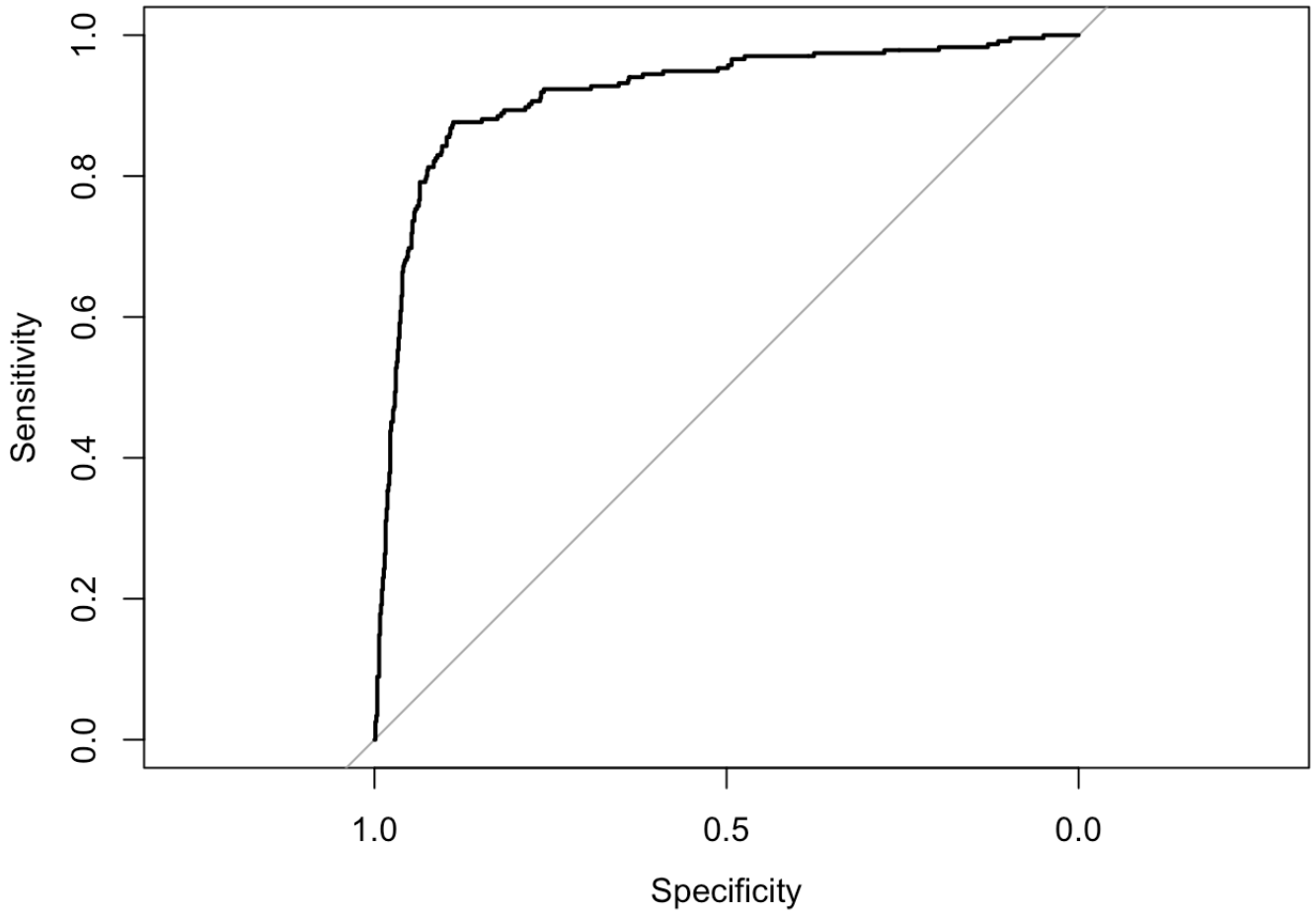
```
## Aggregate data to the patient level

# aggregate numeric measurements using the earliest, mean and maximum observed values

bmi$bmimean <- unsplit(sapply(split(bmi$bmi,bmi$patientid.x),mean,na.rm = T),bmi$patientid.
x)
bmi$bmimax <- unsplit(sapply(split(bmi$bmi,bmi$patientid.x),max,na.rm = T),bmi$patientid.x)
bmi$firstbmi <- unsplit(sapply(split(bmi$bmi,bmi$patientid.x),function(x){x[1]}),bmi$patien
tid.x)

glucose$glucosemean <- unsplit(sapply(split(glucose$glucose,glucose$patientid),mean,na.rm =
T),glucose$patientid)
```

```
glucose$glucosemax <- unsplit(sapply(split(glucose$glucose,glucose$patientid),max,na.rm =
T),glucose$patientid)

hba1c$hba1cmean <- unsplit(sapply(split(hba1c$hba1c,hba1c$patientid),mean,na.rm = T),hba1c$
patientid)
hba1c$hba1cmax <- unsplit(sapply(split(hba1c$hba1c,hba1c$patientid),max,na.rm = T),hba1c$pa
tientid)

chol$cholmean <- unsplit(sapply(split(chol$chol,chol$patientid),mean,na.rm = T),chol$patien
tid)
chol$cholmax <- unsplit(sapply(split(chol$chol,chol$patientid),max,na.rm = T),chol$patienti
d)

encounter$agemean <- unsplit(sapply(split(encounter$age,encounter$patientid),mean,na.rm =
T),encounter$patientid)
encounter$firstage <- unsplit(sapply(split(encounter$age,encounter$patientid),min,na.rm =
T),encounter$patientid)

# look for any occurence of diabetes diagnosis codes, insulin, metformin,
# or visit to an endocrinologist within the period of interest
# T2DM ICD-9 = "250.00", T2DM ICD-10 = "E11.9", T1DM ICD-9 = "250.01", T1DM ICD-10 = "E10.9
"
# Endocrinologist Medicare specialty code = 46

anycode <- function(x,code){
  code.present <- x %in% code
  return(sum(code.present)>0)
}

# Count number of occurences of code

sumcode <- function(x,code){
  code.present <- x %in% code
  return(sum(code.present))
}

# Determine whether metformin prescription precedes insulin prescription
# Returns 1 if only metformin prescribed or metformin prescribed before insulin
# otherwise returns 0

codeorder <- function(x){
```

```
  metdates <- as.Date(x$dates[x$drugs == "metformin"])
  insdates <- as.Date(x$dates[x$drugs == "insulin"])
  if (length(metdates) == 0) metfirst <- 0
  else if (length(metdates) > 0 & length(insdates) == 0) metfirst <- 1
  else if (length(metdates) == 0 & length(insdates) == 0) metfirst <- 0
  else metfirst <- suppressMessages(1*(min(metdates) < min(insdates)))
  return(metfirst)
}

# any T2DM code
encounter$T2DM <- unsplit(sapply(split(encounter$diag,encounter$patientid),anycode,code = c
("250.00","E11.9")),encounter$patientid)

# number of T2DM codes
encounter$T2DMnum <- unsplit(sapply(split(encounter$diag,encounter$patientid),sumcode,code
= c("250.00","E11.9")),encounter$patientid) # number of occurence of T2DM code

# any T1DM code
encounter$T1DM <- unsplit(sapply(split(encounter$diag,encounter$patientid),anycode,code = c
("250.01","E10.9")),encounter$patientid)

# any visit to an endocrinologist
encounter$endo <- unsplit(sapply(split(encounter$prov,encounter$patientid),anycode,code = "
46"),encounter$patientid)

# any depression diagnosis
encounter$dep <- unsplit(sapply(split(encounter$diag,encounter$patientid),anycode,code = c(
"296.2","296.9","296.3","300.4","F32.9","F41.8","F33.9")),encounter$patientid)

# any insulin prescription
meds$anyinsulin <- unsplit(sapply(split(meds$drug,meds$patientid),anycode,code = "insulin")
,meds$patientid)

# any metformin prescription
meds$anymetformin <- unsplit(sapply(split(meds$drug,meds$patientid),anycode,code = "metform
in"),meds$patientid)

# metformin prescription precedes insulin prescription
meds$metforminfirst <- unsplit(sapply(split(data.frame(dates=meds$presdate,drugs=meds$drug)
,
                        meds$patientid),codeorder),meds$patientid)
```

```
## Create merged dataset with one observation per patient and aggregate variables

encounter1 <- encounter[!duplicated(encounter$patientid),c("patientid","agemean","firstage"
,"race","gender","T2DM","T1DM","endo","T2DMnum","dep")]
bmi1 <- bmi[!duplicated(bmi$patientid.x),c("patientid.x","bmimean","bmimax","firstbmi")]
names(bmi1) <- c("patientid","bmimean","bmimax","firstbmi")
glucose1 <- glucose[!duplicated(glucose$patientid),c("patientid","glucosemean","glucosemax"
)]
hba1c1 <- hba1c[!duplicated(hba1c$patientid),c("patientid","hba1cmean","hba1cmax")]
chol1 <- chol[!duplicated(chol$patientid),c("patientid","cholmean","cholmax")]
meds1  <- meds[!duplicated(meds$patientid),c("patientid","anyinsulin","anymetformin","metfo
rminfirst")]

data1 <- Reduce(function(x,y){merge(x,y, all = T)},list(encounter1,bmi1,glucose1,hba1c1,cho
l1,meds1,validation))

# create indicators for availability of any glucose or HbA1c measures

data1$anyglucose <- !is.na(data1$glucosemean)
data1$anyhba1c   <- !is.na(data1$hba1cmean)

# set insulin and metformin to false for patients with no medication data

data1$anyinsulin <- ifelse(is.na(data1$anyinsulin),FALSE,data1$anyinsulin)
data1$anymetformin <- ifelse(is.na(data1$anymetformin),FALSE,data1$anymetformin)

## Phenotyping models using gold standard labels from validation data set to construct pred
iction models for T2DM

## Logistic regression

mod.glm <- glm(T2DMv ~ T2DM + T1DM + bmimean + anyglucose + anyhba1c + anyinsulin + anymetf
ormin, data = data1, family = "binomial")

# logistic regression-based phenotype
data1$T2DMglm <- predict(mod.glm, newdata = data1)

# evaluate performance of logistic regression phenotype
pred.glm <- na.omit(data.frame(pred = data1$T2DMglm,true = data1$T2DMv))
```

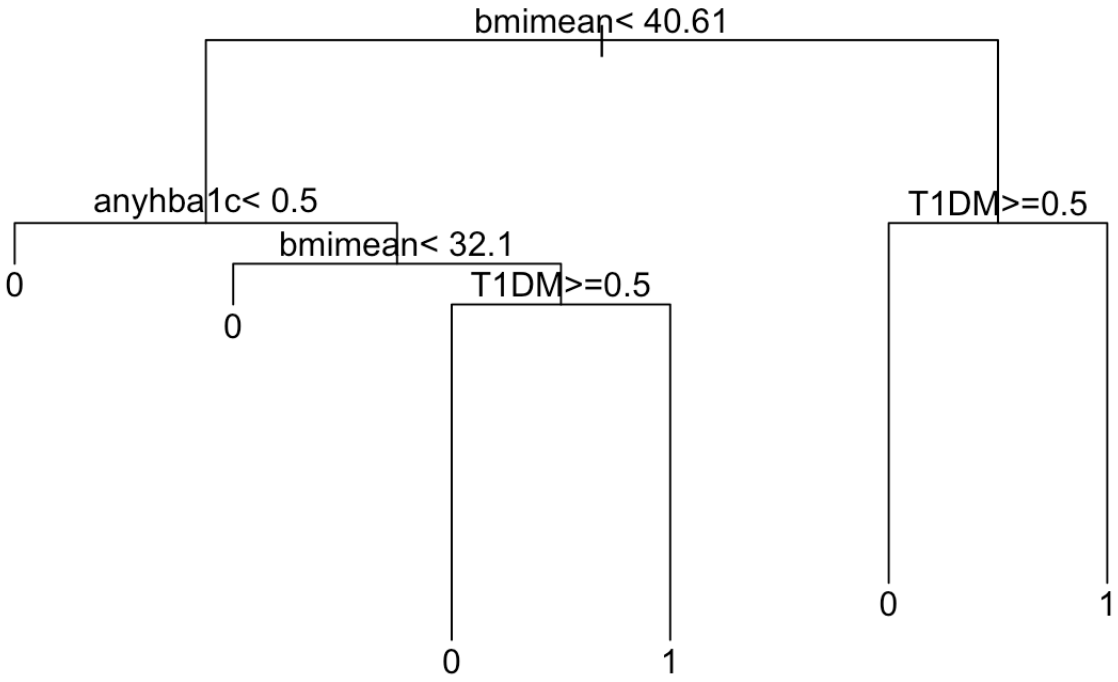
```
perf.glm <- roc(pred.glm$true, pred.glm$pred, auc = TRUE, print.auc = TRUE, show.thres = TR
UE)

plot(perf.glm)
```



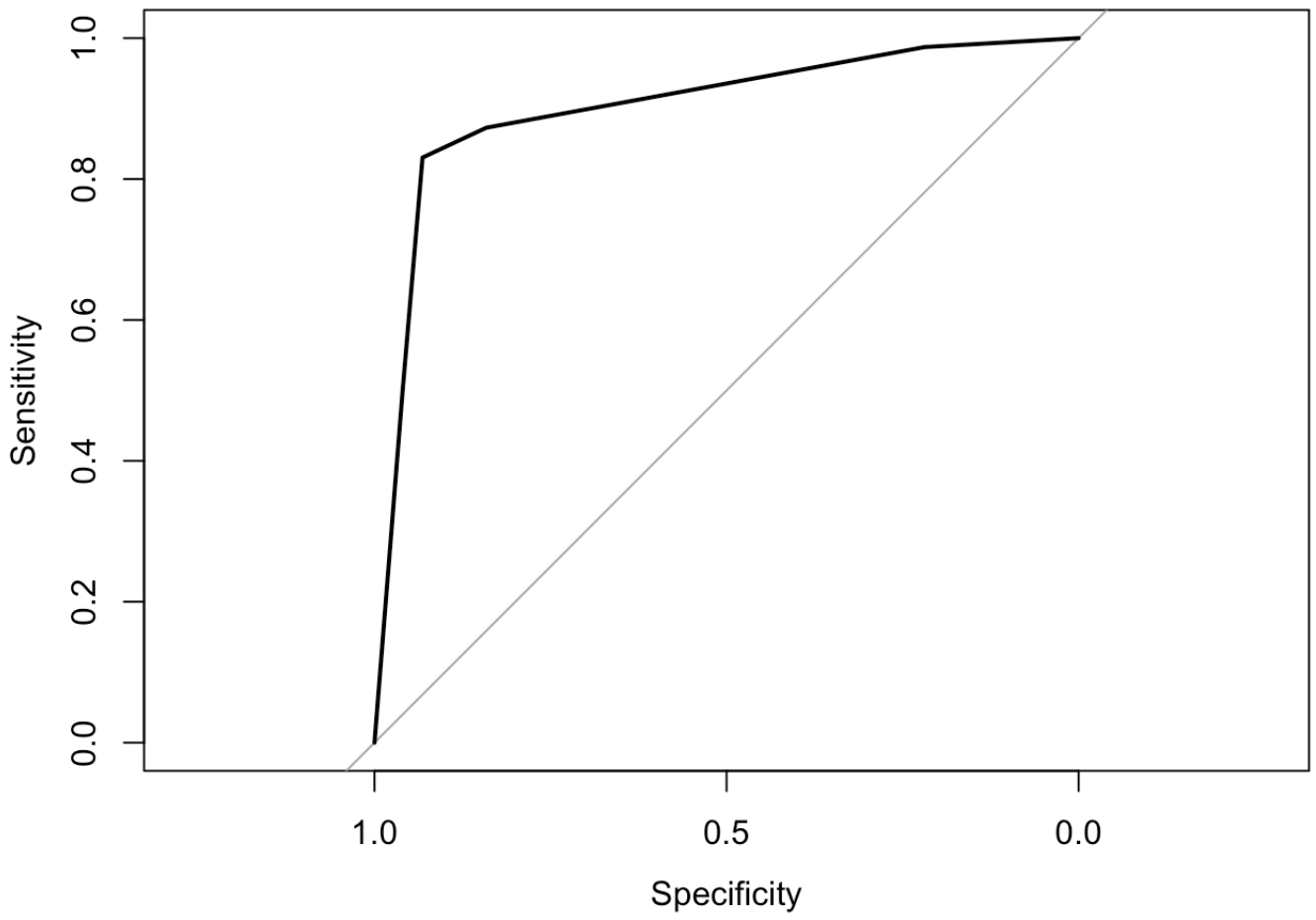

```
# Logistic regression AUC
perf.glm$auc
```

```
## Area under the curve: 0.9182
```

```
## CART

set.seed(20190805)
mod.cart <- rpart(T2DMv ~ T2DM + T1DM  + bmimean + anyglucose + anyhba1c, data = data1, met
hod = "class")
mod.pruned<- prune(mod.cart, cp= mod.cart$cptable[which.min(mod.cart$cptable[,"xerror"]),"C
P"])

par(xpd = NA) # prevent text labels from being cut off
plot(mod.pruned)
text(mod.pruned)
```



```
# predicted probabilities of T2DM based on CART
data1$T2DMcart <- predict(mod.pruned, newdata = data1, type = "prob")

# binary T2DM phenotype based on CART
data1$T2DMcart.class <- as.numeric(as.character(predict(mod.pruned, newdata = data1, type =
"class")))

# evaluate performance of continuous CART phenotype
pred.cart <- na.omit(data.frame(pred = data1$T2DMcart[,2],true = data1$T2DMv))
perf.cart <- roc(pred.cart$true, pred.cart$pred, auc = TRUE, print.auc = TRUE, show.thres =
TRUE)

par(xpd = FALSE)
plot(perf.cart)
```




```
# CART AUC
perf.cart$auc
```

```
## Area under the curve: 0.9022
```

```
## eMERGE T2DM rule

T2DM.rule <- function(x){
  if (x$T1DM ==1) T2DM <- 0
  else{
    if (x$T2DM ==1){
      if (x$anyinsulin == 1){
        if (x$anymetformin == 0){
          if (x$T2DMnum < 2){
            T2DM <- 0
          } else{
            T2DM <- 1
          }
        } else{
          if (x$metforminfirst == 0){
            T2DM <- 0
          } else{
            T2DM <- 1
          }
        }
      } else{
        if (x$anymetformin == 1){
          T2DM <- 1
        } else{
          if ((!is.na(x$glucosemax) & x$glucosemax > 200) | (!is.na(x$hba1cmax) & x$hba1cma
x > 6.5)){
            T2DM <- 1
          } else{
            T2DM <- 0
          }
        }
      }
    } else{
      if (x$anymetformin == 0){
        T2DM <- 0
      } else{
        if ((!is.na(x$glucosemax) & x$glucosemax > 200) | (!is.na(x$hba1cmax) & x$hba1cmax
> 6.5)){
```

```
          T2DM <- 1
        } else{
          T2DM <- 0
        }
      }
    }
  }
  return(T2DM)
}

data1$T2DMemerge <- unsplit(sapply(split(data1,data1$patientid),T2DM.rule),data1$patientid)

# eMERGE specificity
1-mean(data1$T2DMemerge[data1$T2DMv == 0 & !is.na(data1$T2DMv)])
```

```
## [1] 0.9685039
```

```
# eMERGE sensitivity
mean(data1$T2DMemerge[data1$T2DMv == 1 & !is.na(data1$T2DMv)])
```

```
## [1] 0.1059322
```

```
# eMERGE PPV
mean(data1$T2DMv[data1$T2DMemerge == 1],na.rm = T)
```

```
## [1] 0.5102041
```

```
# eMERGE NPV
1 - mean(data1$T2DMv[data1$T2DMemerge == 0],na.rm = T)
```

```
## [1] 0.7776607
```

3. **Missing Data.** Next we will explore missing data in an EHR-derived data set. Suppose we want to use our T2DM phenotype from exercise 2 to explore the relationship between total cholesterol and T2DM diagnosis. How might we define total cholesterol? Using this definition, how much missingness is there? Is missingness related to any other factors in the data set? Use IPW with a single module or multiple modules to account for missingness in your analysis of the association between total cholesterol and T2DM.

```
## For most real examples we would want to define our exposure (cholesterol) in a window ar
ound
## cohort entry. For this toy example we will just use all available data.

# Percent missing cholesterol
mean(is.na(data1$cholmean))
```

```
## [1] 0.7697885
```

```
# Number of encounters per patient
encounter$numvisit <- rep(c(table(encounter$patientid)), times = c(table(encounter$patienti
d)))
summary(c(table(encounter$patientid)))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.164   4.000  39.000
```

```
# Merge number of encounters onto data set with one observation per patient
numvisit <- encounter[!duplicated(encounter$patientid),c("patientid","numvisit")]
data1 <- merge(data1,numvisit)

# Look for factors associated with missing cholesterol
data1$misschol <- is.na(data1$cholmean)
misschol.mod <- glm(misschol ~ firstage + race + gender + firstbmi, data = data1, family =
binomial)
summary(misschol.mod)
```

```
##
## Call:
## glm(formula = misschol ~ firstage + race + gender + firstbmi,
##     family = binomial, data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2087   0.5729   0.6787   0.7421   1.9901
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.738631   0.228746   7.601 2.95e-14 ***
## firstage      0.031373   0.010799   2.905  0.00367 **
## raceBlack    -0.041769   0.144388  -0.289  0.77236
## raceHispanic -0.047558   0.167469  -0.284  0.77642
## raceOther     0.364683   0.203119   1.795  0.07259 .
## raceUnknown  -0.160360   0.191397  -0.838  0.40212
## raceWhite    -0.025952   0.143299  -0.181  0.85629
## genderMale   -0.056812   0.048235  -1.178  0.23887
## firstbmi     -0.025593   0.002325 -11.007  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10690  on 9895  degrees of freedom
## Residual deviance: 10538  on 9887  degrees of freedom
##   (34 observations deleted due to missingness)
## AIC: 10556
##
## Number of Fisher Scoring iterations: 4
```

```r
# Generate probability of missingness from this model
data1$pmisschol1[!is.na(data1$firstbmi)] <- 1-predict(misschol.mod, type = "response", data
= data1)

# Estimate probability of missingness using a two stage model
# first estimate probability of missingness conditional on making an endocrinologist visit
misschol.mod.2 <- glm(misschol ~ endo, data = data1, family = binomial)
summary(misschol.mod.2)
```

```
##
## Call:
## glm(formula = misschol ~ endo, family = binomial, data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8516   0.6302   0.7611   0.7611   0.7611
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.51571    0.04788  31.656  < 2e-16 ***
## endoTRUE    -0.42472    0.05526  -7.685 1.53e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10715  on 9929  degrees of freedom
## Residual deviance: 10653  on 9928  degrees of freedom
## AIC: 10657
##
## Number of Fisher Scoring iterations: 4
```

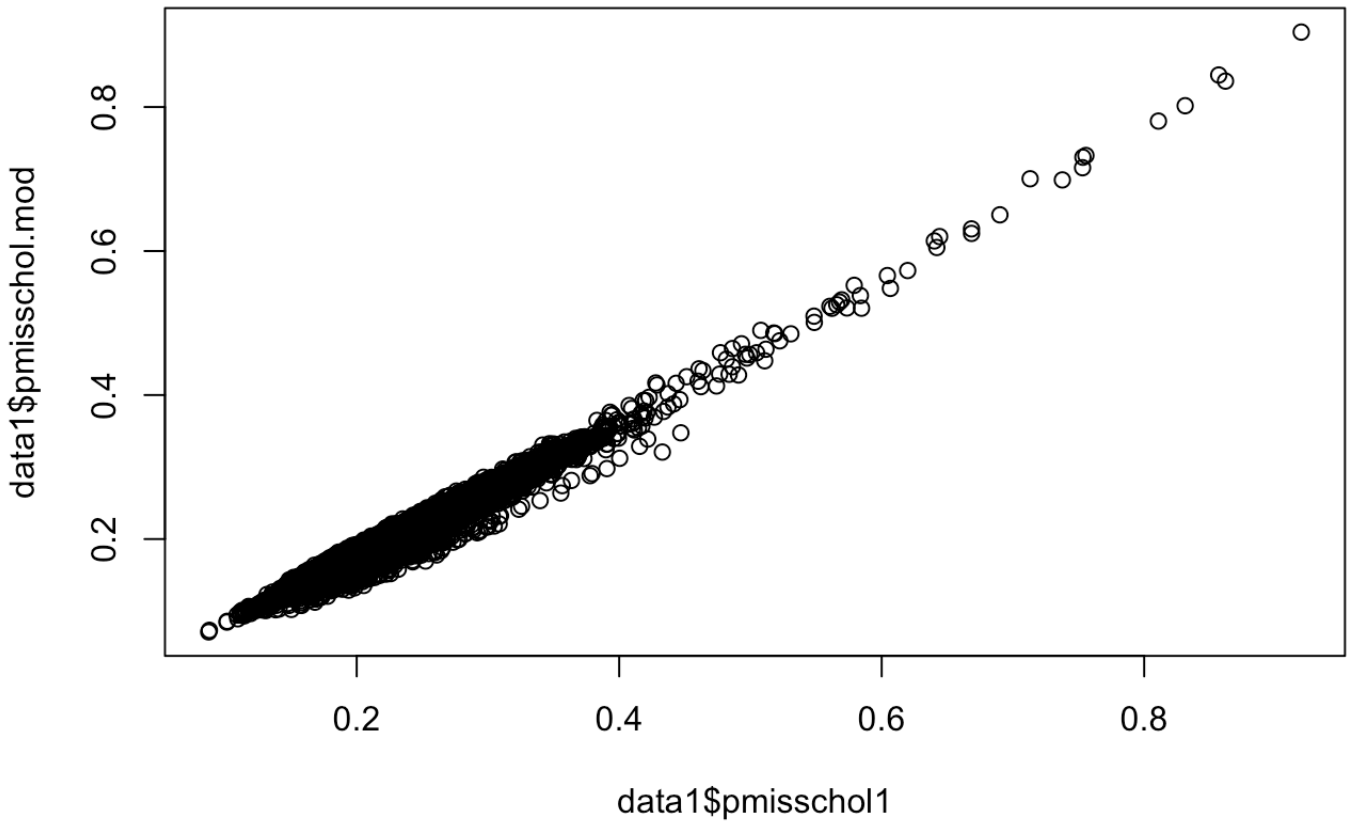
```r
data1$pmisschol2 <- 1-predict(misschol.mod.2, type = "response")

# next estimate probability of missingness among those with and without endocrinologist vis
it
misschol.mod.20 <- glm(misschol ~ firstage + race + gender + firstbmi, data = data1[data1$e
ndo == 0,], family = binomial)
summary(misschol.mod.20)
```

```
##
## Call:
## glm(formula = misschol ~ firstage + race + gender + firstbmi,
##     family = binomial, data = data1[data1$endo == 0, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3423   0.5269   0.5998   0.6539   1.3881
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.701678   0.482984   5.594 2.22e-08 ***
## firstage      0.010246   0.021138   0.485    0.628
## raceBlack    -0.348744   0.334825  -1.042    0.298
## raceHispanic -0.303760   0.377361  -0.805    0.421
## raceOther     0.079812   0.448537   0.178    0.859
## raceUnknown  -0.569856   0.420526  -1.355    0.175
## raceWhite    -0.409527   0.332578  -1.231    0.218
## genderMale   -0.118909   0.096894  -1.227    0.220
## firstbmi     -0.026743   0.004823  -5.545 2.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2774.1  on 2931  degrees of freedom
## Residual deviance: 2737.3  on 2923  degrees of freedom
##   (22 observations deleted due to missingness)
## AIC: 2755.3
##
## Number of Fisher Scoring iterations: 4
```

```
misschol.mod.21 <- glm(misschol ~ firstage + race + gender + firstbmi, data = data1[data1$e
ndo == 1,], family = binomial)
summary(misschol.mod.21)
```

```
##
## Call:
## glm(formula = misschol ~ firstage + race + gender + firstbmi,
##     family = binomial, data = data1[data1$endo == 1, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0309  -0.9697   0.7147   0.7777   1.9343
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.424344   0.263077   5.414 6.16e-08 ***
## firstage      0.034603   0.012644   2.737   0.0062 **
## raceBlack     0.030477   0.161875   0.188   0.8507
## raceHispanic  0.010982   0.189058   0.058   0.9537
## raceOther     0.437645   0.229879   1.904   0.0569 .
## raceUnknown  -0.051194   0.217098  -0.236   0.8136
## raceWhite     0.072651   0.160643   0.452   0.6511
## genderMale   -0.032985   0.055801  -0.591   0.5544
## firstbmi     -0.023421   0.002665  -8.788  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7856.3  on 6963  degrees of freedom
## Residual deviance: 7755.2  on 6955  degrees of freedom
##   (12 observations deleted due to missingness)
## AIC: 7773.2
##
## Number of Fisher Scoring iterations: 4
```

```
data1$pmisschol20[!is.na(data1$firstbmi)] <- 1-predict(misschol.mod.20, type = "response",
newdata = data1[!is.na(data1$firstbmi),])
data1$pmisschol21[!is.na(data1$firstbmi)] <- 1-predict(misschol.mod.21, type = "response",
newdata = data1[!is.na(data1$firstbmi),])

# create combined probability of having an observed cholesterol value given these two modul
es
data1$pmisschol.mod <- data1$pmisschol2*data1$pmisschol21+(1-data1$pmisschol2)*data1$pmissc
hol20

## Compare one module and two module probabilities of being observed
plot(data1$pmisschol1, data1$pmisschol.mod)
```



```
## Fit regression models using IPW to account for missingness in cholesterol

# Model using 1 step weights
data1$w1 <- 1/data1$pmisschol1
data1$w1 <- sum(!is.na(data1$w1))*data1$w1/sum(data1$w1,na.rm = T) # normalize weights to m
aintain sample size
chol.mod1 <- glm(T2DMcart.class~ firstage + factor(race) + gender + cholmean, data = data1,
weights = w1, family = "binomial")
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(chol.mod1)
```

```
##
## Call:
## glm(formula = T2DMcart.class ~ firstage + factor(race) + gender +
##     cholmean, family = "binomial", data = data1, weights = w1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1342  -0.7431  -0.6623   1.2864   2.4898
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.8726370  0.4752341  -1.836  0.06632 .
## firstage             -0.0018165  0.0231983  -0.078  0.93759
## factor(race)Black     0.1530503  0.3144667   0.487  0.62647
## factor(race)Hispanic  0.2368799  0.3625274   0.653  0.51349
## factor(race)Other    -0.2602421  0.4354981  -0.598  0.55012
## factor(race)Unknown   0.3012915  0.4117013   0.732  0.46428
## factor(race)White     0.1456695  0.3120705   0.467  0.64065
## genderMale           -0.3130456  0.1036234  -3.021  0.00252 **
## cholmean             -0.0011645  0.0005107  -2.280  0.02259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2321.2  on 2282  degrees of freedom
## Residual deviance: 2304.4  on 2274  degrees of freedom
##   (7647 observations deleted due to missingness)
## AIC: 2627.5
##
## Number of Fisher Scoring iterations: 4
```

```
# Model using 2 step weights
data1$w.mod <- 1/data1$pmisschol.mod
data1$w.mod <- sum(!is.na(data1$w.mod))*data1$w.mod/sum(data1$w.mod,na.rm = T) # normalize
weights to maintain sample size
chol.mod2 <- glm(T2DMcart.class~ firstage + factor(race) + gender + cholmean, data = data1,
weights = w.mod, family = "binomial")
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(chol.mod2)
```

```
##
## Call:
## glm(formula = T2DMcart.class ~ firstage + factor(race) + gender +
##     cholmean, family = "binomial", data = data1, weights = w.mod)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1319  -0.7417  -0.6555   1.2853   2.4789
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.9006749  0.4616654  -1.951  0.05107 .
## firstage             -0.0006169  0.0232998  -0.026  0.97888
## factor(race)Black     0.1479875  0.2922172   0.506  0.61256
## factor(race)Hispanic  0.2346305  0.3416190   0.687  0.49220
## factor(race)Other    -0.2716552  0.4165533  -0.652  0.51430
## factor(race)Unknown   0.2960725  0.3992965   0.741  0.45840
## factor(race)White     0.1445054  0.2900153   0.498  0.61829
## genderMale           -0.3139550  0.1041307  -3.015  0.00257 **
## cholmean             -0.0011469  0.0005128  -2.237  0.02531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2308.3  on 2282  degrees of freedom
## Residual deviance: 2291.6  on 2274  degrees of freedom
##   (7647 observations deleted due to missingness)
## AIC: 2620.6
##
## Number of Fisher Scoring iterations: 4
```

4. **Confounding by Utilization Intensity.** Accounting for the intensity of healthcare utilization in analyses. How much variability is there in the intensity of utilization in this data set? Use a measure of intensity of utilization to account for informed presence bias in an analysis of the association between depression diagnosis and T2DM.

```r
## Analyze association between depression and T2DM with and without conditioning on visit i
ntensity
dep.glm1 <- glm(T2DMcart.class ~ firstage + factor(race) + gender + dep, data = data1, fami
ly = "binomial")
summary(dep.glm1)
```

```
##
## Call:
## glm(formula = T2DMcart.class ~ firstage + factor(race) + gender +
##     dep, family = "binomial", data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9230  -0.7613  -0.6643  -0.5759   2.0423
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.852465   0.206500  -4.128 3.66e-05 ***
## firstage             -0.028439   0.010789  -2.636  0.00839 **
## factor(race)Black    -0.091164   0.140435  -0.649  0.51624
## factor(race)Hispanic -0.110312   0.164607  -0.670  0.50276
## factor(race)Other    -0.331234   0.194232  -1.705  0.08813 .
## factor(race)Unknown   0.007052   0.189507   0.037  0.97032
## factor(race)White    -0.104223   0.139289  -0.748  0.45431
## genderMale           -0.257161   0.048274  -5.327 9.98e-08 ***
## depTRUE               0.496894   0.048576  10.229  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10679  on 9929  degrees of freedom
## Residual deviance: 10528  on 9921  degrees of freedom
## AIC: 10546
##
## Number of Fisher Scoring iterations: 4
```

```
dep.glm2 <- glm(T2DMcart.class ~ firstage + factor(race) + gender + dep + numvisit, data =
data1, family = "binomial")
summary(dep.glm2)
```

```
##
## Call:
## glm(formula = T2DMcart.class ~ firstage + factor(race) + gender +
##     dep + numvisit, family = "binomial", data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2376  -0.7622  -0.6116  -0.4727   2.2161
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.956791   0.220952  -8.856  < 2e-16 ***
## firstage             -0.005544   0.011112  -0.499    0.618
## factor(race)Black    -0.103840   0.142270  -0.730    0.465
## factor(race)Hispanic -0.117429   0.167000  -0.703    0.482
## factor(race)Other    -0.328713   0.196876  -1.670    0.095 .
## factor(race)Unknown   0.023632   0.192137   0.123    0.902
## factor(race)White    -0.110363   0.141125  -0.782    0.434
## genderMale           -0.264333   0.049062  -5.388 7.14e-08 ***
## depTRUE               0.249812   0.051381   4.862 1.16e-06 ***
## numvisit              0.278357   0.017115  16.263  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10679  on 9929  degrees of freedom
## Residual deviance: 10239  on 9920  degrees of freedom
## AIC: 10259
##
## Number of Fisher Scoring iterations: 4
```

```
# compare odds ratios before and after adjustment
cbind(c(exp(dep.glm1$coef),NA),exp(dep.glm2$coef))
```

```
##                            [,1]      [,2]
## (Intercept)           0.4263628 0.1413111
## firstage              0.9719613 0.9944712
## factor(race)Black     0.9128683 0.9013694
## factor(race)Hispanic  0.8955548 0.8892039
## factor(race)Other     0.7180368 0.7198495
## factor(race)Unknown   1.0070765 1.0239131
## factor(race)White     0.9010242 0.8955093
## genderMale            0.7732437 0.7677181
## depTRUE               1.6436087 1.2837843
##                              NA 1.3209573
```

5. **Outcome Misclassification.** Use the classic Magder and Hughes approach to accounting for outcome misclassification to account for phenotyping error in an analysis of the association between having a depression diagnosis code and T2DM using the CART-derived T2DM phenotype.

```
## Analysis without additional adjustment variables

# first compute sensitivity and specificity using validation data
sens <- mean(as.numeric(as.character(data1$T2DMcart.class[data1$T2DMv == 1 & !is.na(data1$T2DMv)])))
sens
```

```
## [1] 0.8305085
```

```
spec <- 1-mean(as.numeric(as.character(data1$T2DMcart.class[data1$T2DMv == 0 & !is.na(data1$T2DMv)])))
spec
```

```
## [1] 0.9317585
```

```
# compute odds ratios based on 2x2 table
a <- sum(data1$T2DMcart.class == 1 & data1$dep == 1)
b <- sum(data1$T2DMcart.class == 0 & data1$dep == 1)
c <- sum(data1$T2DMcart.class == 1 & data1$dep == 0)
d <- sum(data1$T2DMcart.class == 0 & data1$dep == 0)

or.std <- a*d/(b*c) # standard odds ratio
or.mh <- (a/(a+b)-(1-spec))/(c/(c+d)-(1-spec))*(sens-c/(c+d))/(sens-a/(a+b)) # Magder and H
ughes adjusted odds ratio

or.std
```

```
## [1] 1.658142
```

```
or.mh
```

```
## [1] 2.035416
```

```
## Adjusted analysis via logistic regression using EM algorithm

# posterior probability of Y
post.prob <- function(phat,S,sens,spec){
  post.probY <- ifelse(S== 1, sens*phat/(sens*phat+(1-spec)*(1-phat)),
                       (1-spec)*phat/((1-spec)*phat+sens*(1-phat)))
  return(post.probY)
}

# EM algorithm proposed by Magder and Hughes
mh.EM <- function(fmla, sens, spec, tol = 10^-4, maxit = 10){
  data1$Y <- data1$T2DMcart.class
  or1 <- glm(fmla, data = data1, family = "binomial")
  p0 <- predict(or1, type = "response")
  dif <- 1
  j <- 0
  while (dif > tol & j < maxit){
    w <- post.prob(p0,data1$T2DMcart.class,sens,spec)
    data2 <- rbind(data1, data1)
    data2$w <- c(w,1-w)
    data2$Y <- c(rep(1,nrow(data1)),rep(0,nrow(data1)))
    suppressWarnings(or2 <- glm(fmla, data = data2, family = "binomial", weights = w))
    p0 <- predict(or2, type = "response", newdata = data1)
    dif <- max(abs(or1$coef-or2$coef))
    or1 <- or2
    j <- j+1
  }
  if (dif > tol) return("Did not converge")
  else return(or2)
}

# fit model
fmla.dep <- formula("Y ~ firstage + factor(race) + gender + dep")
mod.MH <- mh.EM(fmla.dep, sens, spec, maxit = 100)
summary(mod.MH)
```

```
##
## Call:
## glm(formula = fmla, family = "binomial", data = data2, weights = w)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.86442  -0.54827  -0.04711   0.25356   1.62844
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.08217    0.22600  -4.788 1.68e-06 ***
## firstage             -0.03699    0.01188  -3.114  0.00184 **
## factor(race)Black    -0.11723    0.15257  -0.768  0.44227
## factor(race)Hispanic -0.14631    0.17947  -0.815  0.41493
## factor(race)Other    -0.49643    0.21861  -2.271  0.02316 *
## factor(race)Unknown  -0.00404    0.20593  -0.020  0.98435
## factor(race)White    -0.13049    0.15130  -0.862  0.38843
## genderMale           -0.36210    0.05322  -6.804 1.02e-11 ***
## depTRUE               0.70615    0.05411  13.051  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9369.6  on 19859  degrees of freedom
## Residual deviance: 9123.3  on 19851  degrees of freedom
## AIC: 10693
##
## Number of Fisher Scoring iterations: 4
```

```
# naive model for comparison
mod.cart <- glm(T2DMcart.class ~ firstage + factor(race) + gender + dep, data = data1, fami
ly = "binomial")
summary(mod.cart)
```

```
##
## Call:
## glm(formula = T2DMcart.class ~ firstage + factor(race) + gender +
##     dep, family = "binomial", data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9230  -0.7613  -0.6643  -0.5759   2.0423
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.852465   0.206500  -4.128 3.66e-05 ***
## firstage             -0.028439   0.010789  -2.636  0.00839 **
## factor(race)Black    -0.091164   0.140435  -0.649  0.51624
## factor(race)Hispanic -0.110312   0.164607  -0.670  0.50276
## factor(race)Other    -0.331234   0.194232  -1.705  0.08813 .
## factor(race)Unknown   0.007052   0.189507   0.037  0.97032
## factor(race)White    -0.104223   0.139289  -0.748  0.45431
## genderMale           -0.257161   0.048274  -5.327 9.98e-08 ***
## depTRUE               0.496894   0.048576  10.229  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10679  on 9929  degrees of freedom
## Residual deviance: 10528  on 9921  degrees of freedom
## AIC: 10546
##
## Number of Fisher Scoring iterations: 4
```

```
# model based on validation data only
mod.valid <- glm(T2DMv ~ firstage + factor(race) + gender + dep, data = data1, family = "bi
nomial")
summary(mod.valid)
```

```
##
## Call:
## glm(formula = T2DMv ~ firstage + factor(race) + gender + dep,
##     family = "binomial", data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0839  -0.8063  -0.6264  -0.5104   2.1072
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.54052    0.61432  -0.880    0.379
## firstage             -0.04287    0.03339  -1.284    0.199
## factor(race)Black    -0.37359    0.42683  -0.875    0.381
## factor(race)Hispanic -0.36823    0.49387  -0.746    0.456
## factor(race)Other    -0.30481    0.62390  -0.489    0.625
## factor(race)Unknown  -0.87872    0.63833  -1.377    0.169
## factor(race)White    -0.24339    0.42270  -0.576    0.565
## genderMale           -0.23062    0.15202  -1.517    0.129
## depTRUE               0.74528    0.15426   4.831 1.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1091.8  on 997  degrees of freedom
## Residual deviance: 1059.9  on 989  degrees of freedom
##   (8932 observations deleted due to missingness)
## AIC: 1077.9
##
## Number of Fisher Scoring iterations: 4
```

```
# Compare odds ratios from all three models
data.frame(MH = exp(mod.MH$coef), Naive = exp(mod.cart$coef), Validation = exp(mod.valid$co
ef))
```

```
##                             MH     Naive Validation
## (Intercept)          0.3388594 0.4263628  0.5824481
## firstage             0.9636815 0.9719613  0.9580315
## factor(race)Black    0.8893788 0.9128683  0.6882597
## factor(race)Hispanic 0.8638881 0.8955548  0.6919576
## factor(race)Other    0.6087010 0.7180368  0.7372609
## factor(race)Unknown  0.9959681 1.0070765  0.4153126
## factor(race)White    0.8776640 0.9010242  0.7839635
## genderMale           0.6962144 0.7732437  0.7940388
## depTRUE              2.0261800 1.6436087  2.1070248
```

6. **Using Probabilistic Phenotypes.** Using predicted probabilities from your logistic regression-based phenotype derived in exercise 2, estimate the association between having a depression diagnosis code and T2DM. How do your results change if you use the bias correction approach described in lecture vs the uncorrected results?

```
# Function for bias correction with known values for mu0 and mu1
# link can take values "ident", "log", or "logistic"
bias.adjust.prob <- function(fmla,mu0,mu1,p0,link = "ident"){

  # regress probabilistic phenotype on predictors
  fitp = lm(fmla, data = data1)

  # make bias correctioon
  betastar = fitp$coef/(mu1 - mu0)

  if (link == "ident"){
    betastar = betastar
  } else if (link == "log"){
    betastar = betastar/p0
  } else if (link == "logit"){
    betastar <- betastar/(p0*(1-p0))
  } else return("unsupported link function")

  # return association parameters (drop intercept)
  return(betastar[-1])
}

# use validation data to compute mean phenotype probability among true cases and controls
data1$prob <- inv.logit(data1$T2DMglm)

mu0 <- mean(data1$prob[data1$T2DMv == 0 & !is.na(data1$T2DMv)],na.rm = T)
mu1 <- mean(data1$prob[data1$T2DMv == 1 & !is.na(data1$T2DMv)], na.rm = T)

# use mean of predicted probabilities to estimate prevalence
p0 <- mean(inv.logit(data1$T2DMglm),na.rm = T)

# fit model
fmla.prob <- formula("prob ~ firstage + factor(race) + gender + dep")

mod.prob <- bias.adjust.prob(fmla.prob, mu0, mu1, p0, link = "logit")

# compare with results using validation data
data.frame(Adj = exp(mod.prob), Validation = exp(mod.valid$coef)[-1])
```

```
##                            Adj Validation
## firstage              0.9451832  0.9580315
## factor(race)Black     0.8197881  0.6882597
## factor(race)Hispanic  0.7845029  0.6919576
## factor(race)Other     0.5518600  0.7372609
## factor(race)Unknown   0.9176800  0.4153126
## factor(race)White     0.8023517  0.7839635
## genderMale            0.6716613  0.7940388
## depTRUE               2.3050130  2.1070248
```