Final_project EES6690 Columbia University – Haoran Guo (hg2461@columbia.edu), Hao Fang (hg2345@columbia.edu)

Notice: This work is the final project of course EECS E6690 TPC: Statistical Learning in Bio & Info System. Prof. Predrag R. Jelenkovi ??c. Instructors: Ludwig Zhao, Tingkai Liu.

This work aims to reproduce the result in article: Beata Strack,1 Jonathan P. DeShazo Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records, 2014. Print. Major part includes training, evaluating and improving a logistic model. Some statistical graphs are then generated for analysis purpose.

Data set: Diabetes 130-US hospitals for years 1999-2008 Data Set

Loading libraries

```r
library(GGally)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(lasso2)
```

```
## R Package to solve regression problems while imposing
##   an L1 constraint on the parameters. Based on S-plus Release 2.1
## Copyright (C) 1998, 1999
## Justin Lokhorst   <jlokhors@stats.adelaide.edu.au>
## Berwin A. Turlach <bturlach@stats.adelaide.edu.au>
## Bill Venables     <wvenable@stats.adelaide.edu.au>
##
## Copyright (C) 2002
## Martin Maechler <maechler@stat.math.ethz.ch>
```

```
##
## Attaching package: 'lasso2'
```

```
## The following object is masked from 'package:psych':
##
##     tr
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:GGally':
```

```
##
##     nasa
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(proto)
library(RSQLite)
library(gsubfn)
library(sqldf)
library(lattice)
library(survival)
library(grid)
library(Matrix)
library(survey)
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```r
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```r
library(rpart)
library(randomForest)
```

```
## 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
```

```
## The following object is masked from 'package:psych':
##
##     outlier
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(e1071)
library(nnet)
library(CORElearn)
```

```
##
## Attaching package: 'CORElearn'

## The following object is masked from 'package:survey':
##
##     calibrate
```

Loading Original Data

```r
filename = 'C:\\Users\\guohr\\Desktop\\readm_pre\\diabetic_data.csv'
filename2 = 'C:\\Users\\Mice\\Desktop\\Statistic_pj\\diabetic_data.csv'
data = read.table(filename, sep = ",", header = T, na.strings = "?")
nrow(data)
```

```
## [1] 101766
```

Dropping some columns

```r
data = select(data, -encounter_id,  -weight,-payer_code, -(25:41),-(43:47))
```

Adding missing value

```r
any(is.na(data$race)) # true
```

```
## [1] TRUE
```

```r
any(is.na(data$medical_specialty)) # true
```

```
## [1] TRUE
```

```r
levels <- levels(data$race)
levels[length(levels) + 1] <- "Missing"

# refactor Species to include "Missing" as a factor level
# and replace NA with "None"
data$race <- factor(data$race, levels = levels)
data$race[is.na(data$race)] <- "Missing"

levels <- levels(data$medical_specialty)
levels[length(levels) + 1] <- "Missing"

data$medical_specialty <- factor(data$medical_specialty, levels = levels)
data$medical_specialty[is.na(data$medical_specialty)] <- "Missing"

any(is.na(data$race)) # false
```

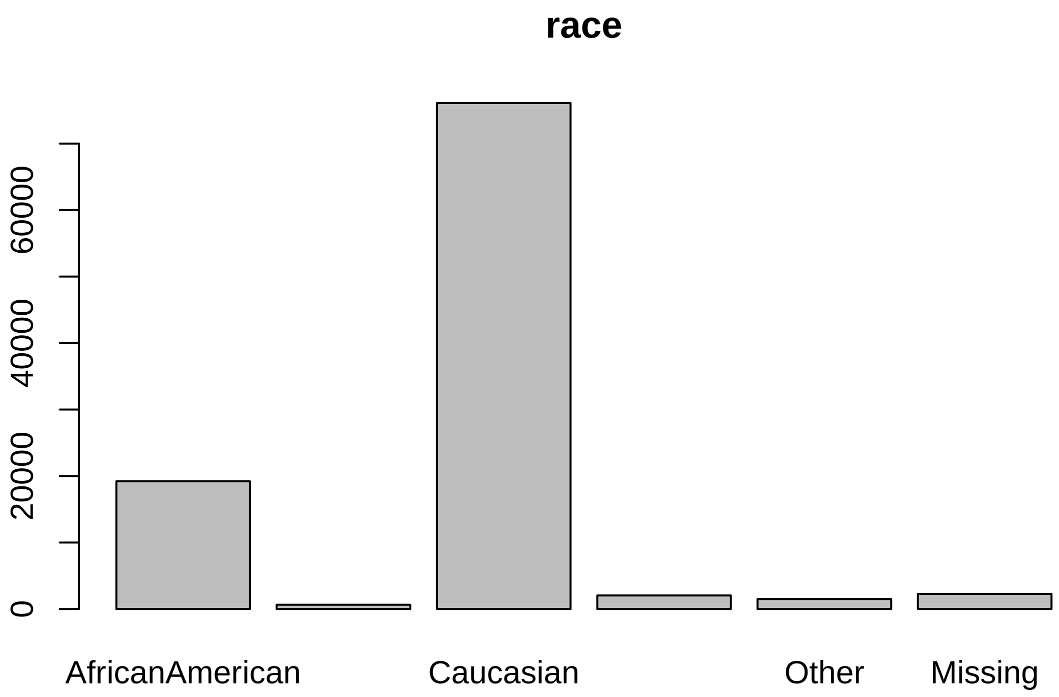
```
## [1] FALSE
```

```r
any(is.na(data$medical_specialty)) # false
```

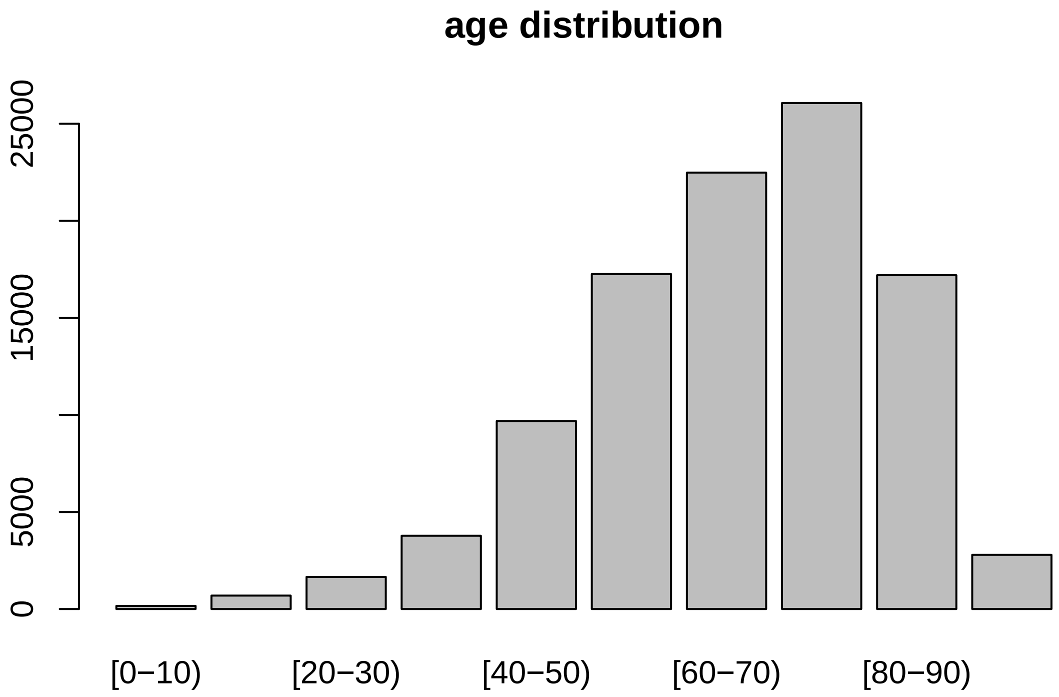
```
## [1] FALSE
```

Preliminary data analysis

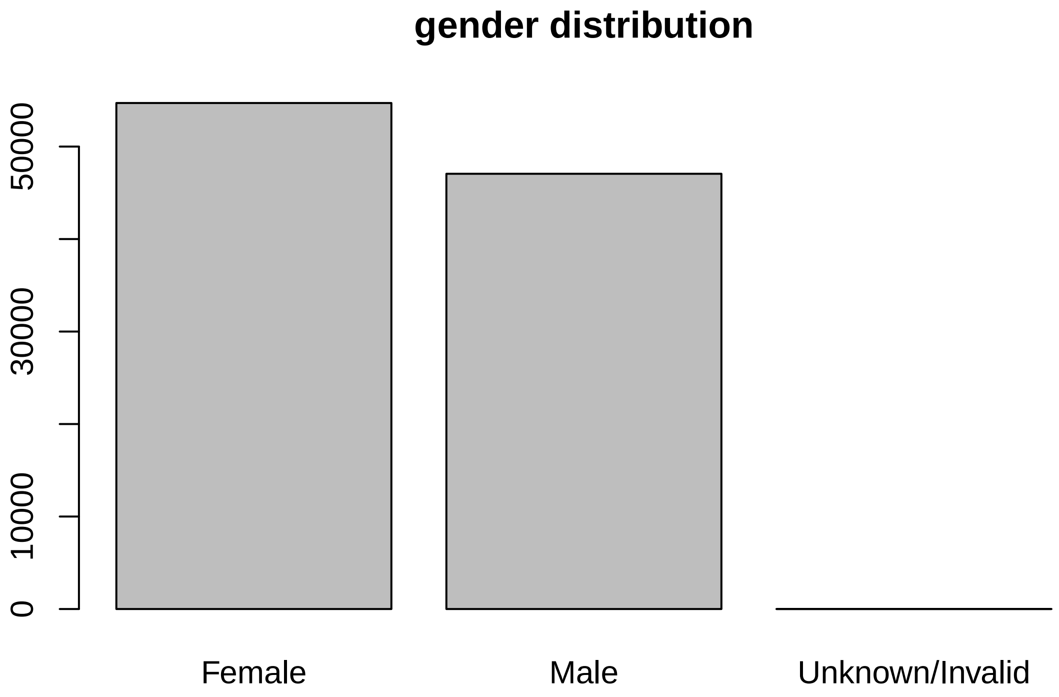
```r
plot(data$race, main = "race")
```



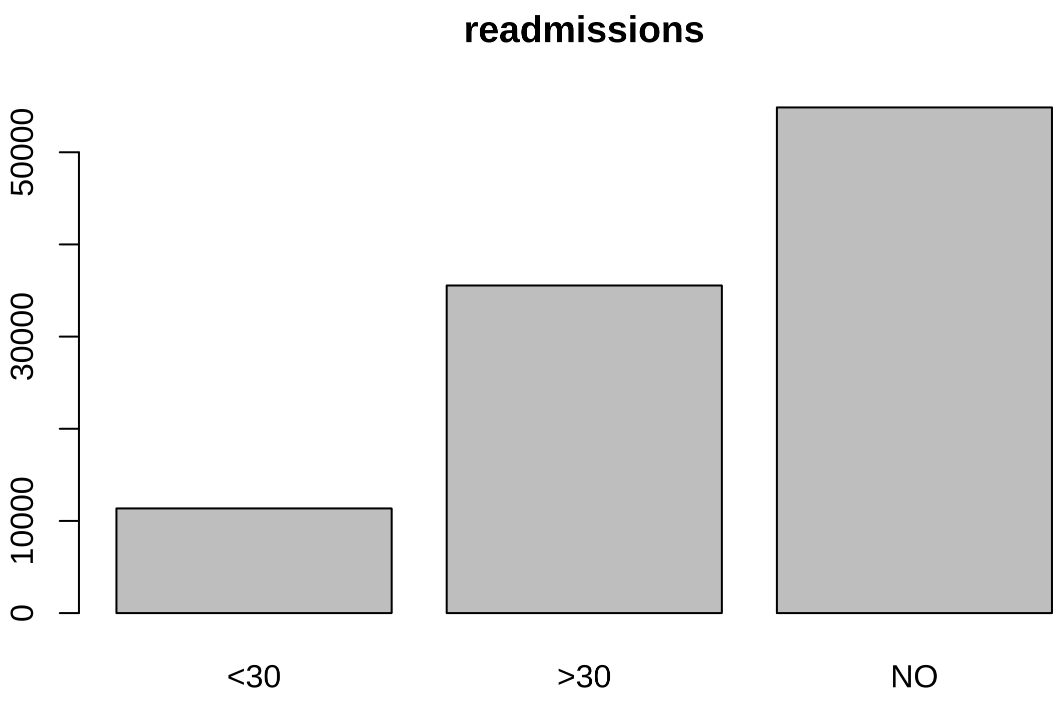
```
# PLOTS

# variable distributions
plot(data$age, main = "age distribution")
```



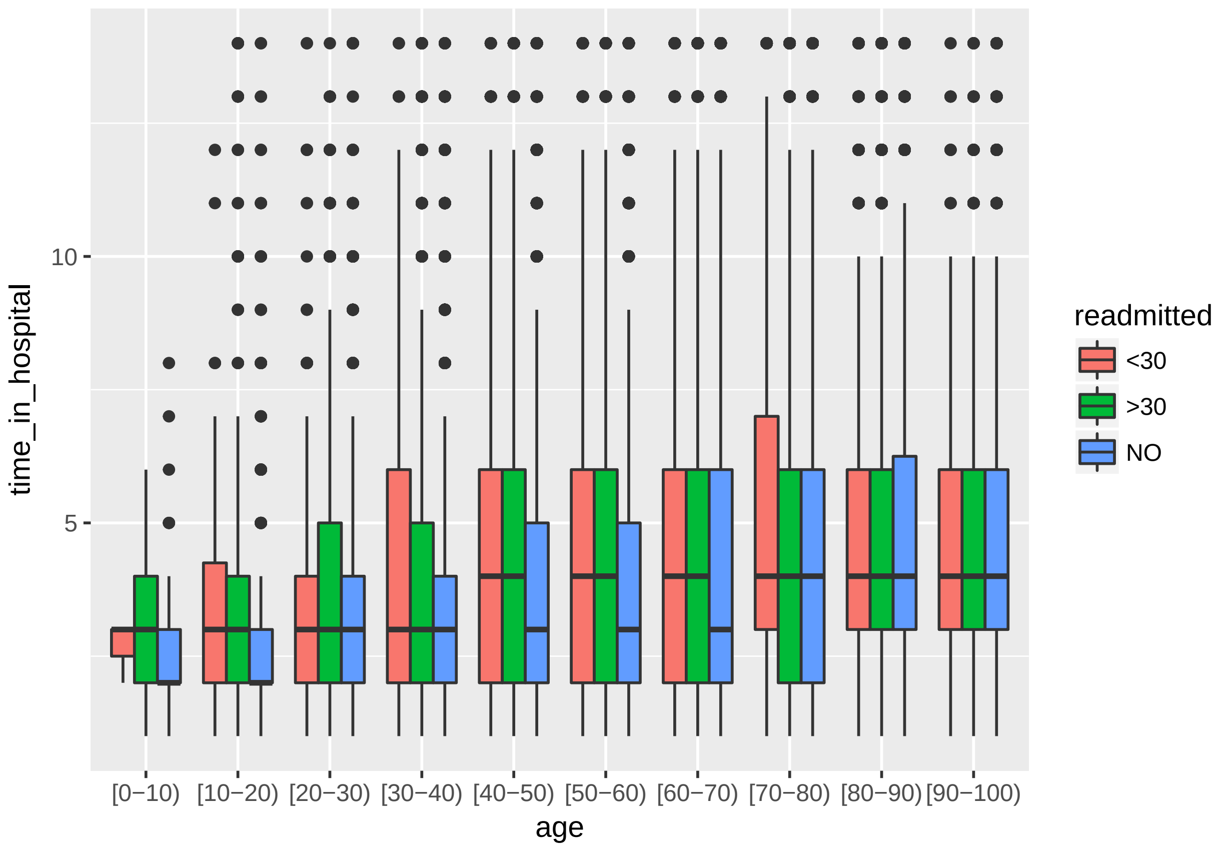
```r
plot(data$gender, main = "gender distribution")
```



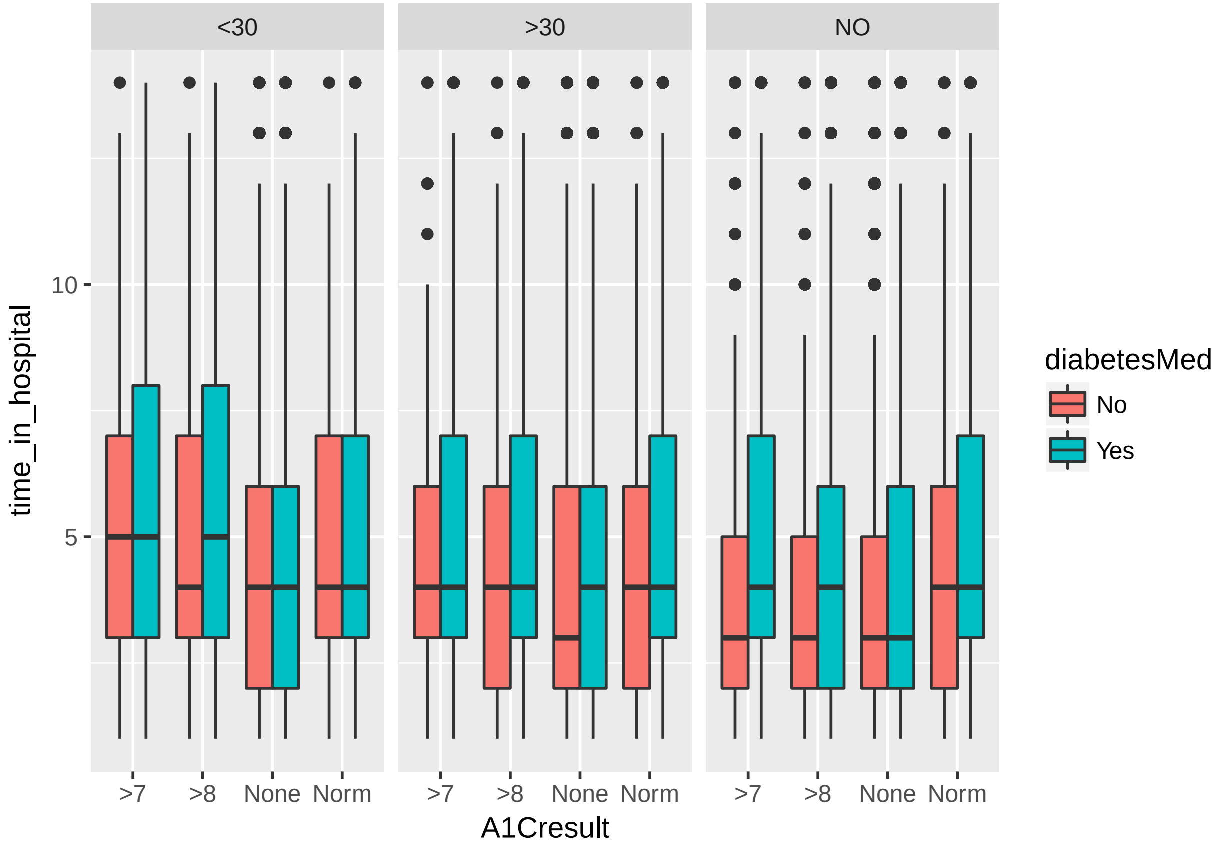
```
plot(data$readmitted, main = "readmissions")
```



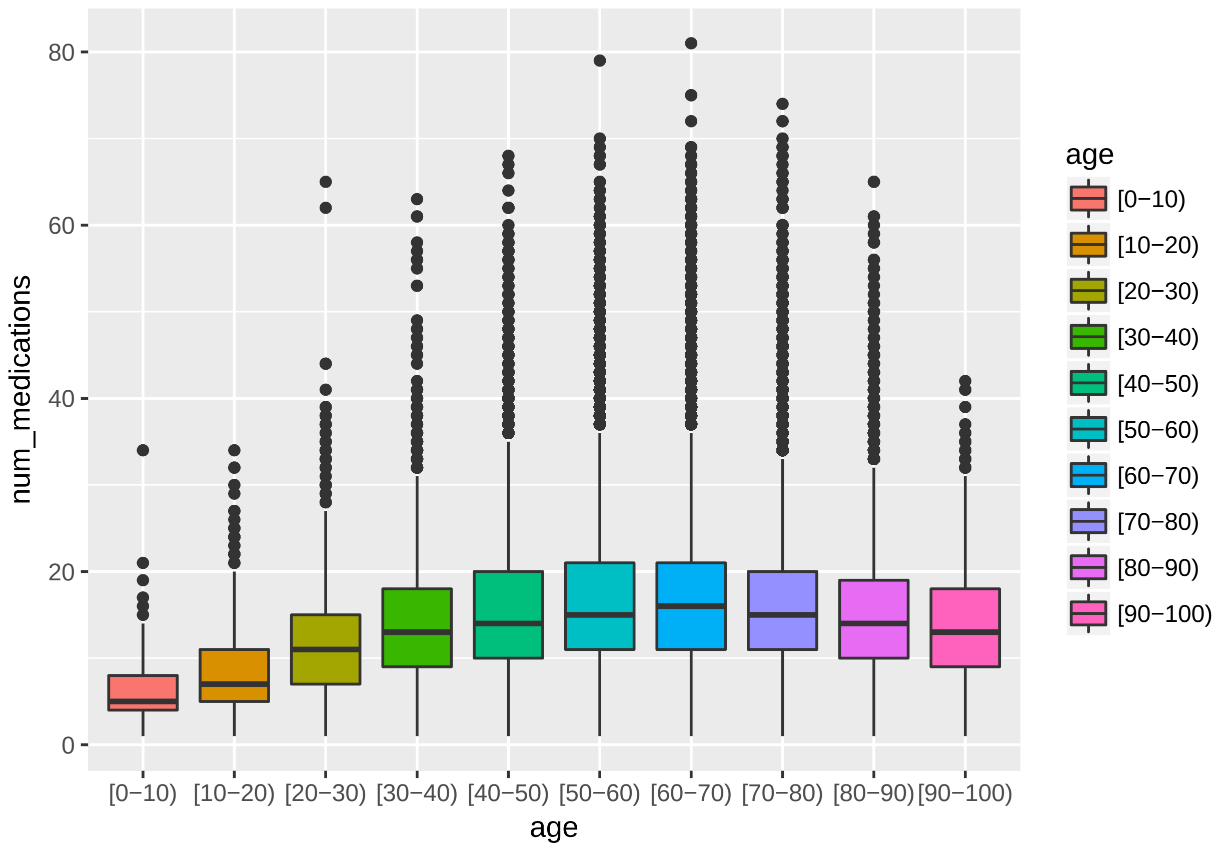
```r
g <- ggplot(data, aes(x=age, y=time_in_hospital))
g + geom_boxplot(aes(fill=readmitted))
```



```r
g <- ggplot(data,aes(x=A1Cresult, y=time_in_hospital))
g + geom_boxplot(aes(fill=diabetesMed)) + facet_grid(. ~ readmitted)
```



```
g <- ggplot(data,aes(x=age, y=num_medications))
g + geom_boxplot(aes(fill=age))
```



Preprocessing data with RSQL, merging classes of A1Cresult and readmitted, also filter out istances with discharge_disposition_id

```
data = sqldf("select *, count(distinct patient_nbr) as a from data where discharge_disposition_id not i

data = sqldf("select case A1Cresult when '>7' then 'Norm' when '>8' then 'High' else A1Cresult end as H

data = select(data, -readmitted, -patient_nbr, -a, -A1Cresult, -num_lab_procedures)
```

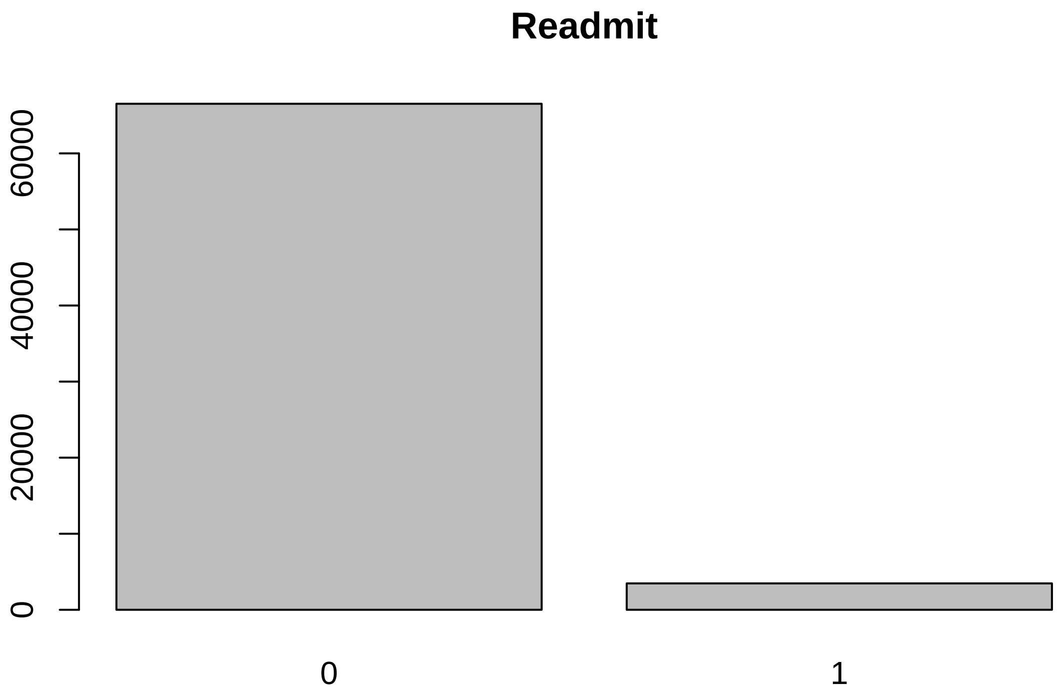
Preprocessing data

Handling with 3 times' diganosis and also medical_speciality(Surgery)

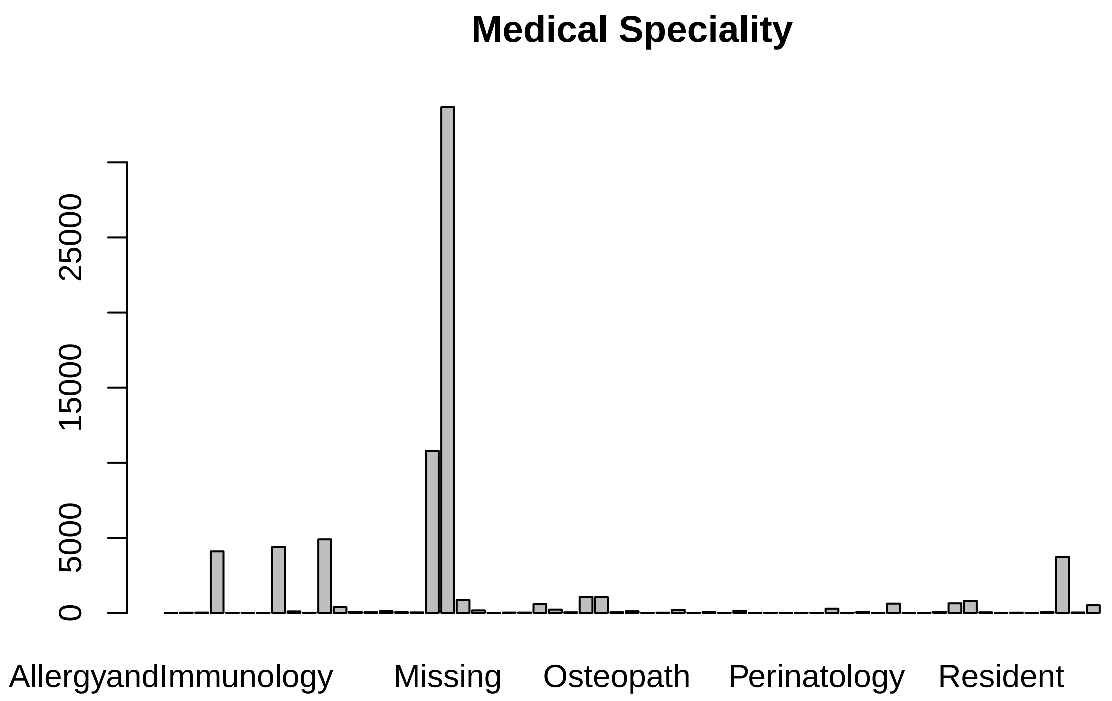
```
data = sqldf("select case when diag_3 between 250 and 251 then 'Diabetes' when diag_3 between 390 and 4

data = sqldf("select case when diag_2 between 250 and 251 then 'Diabetes' when diag_2 between 390 and 4

data = sqldf("select case when diag_1 between 250 and 251 then 'Diabetes' when diag_1 between 390 and 4

data = sqldf("select case when medical_specialty like 'Surgery%' then 'Surgery' else medical_specialty 

data$MedicalSpeciality <- as.factor(data$MedicalSpeciality)
data = select(data, -medical_specialty)

# Make a copy of dataframe
data_full = data
```

Further statistical exploration of dataset

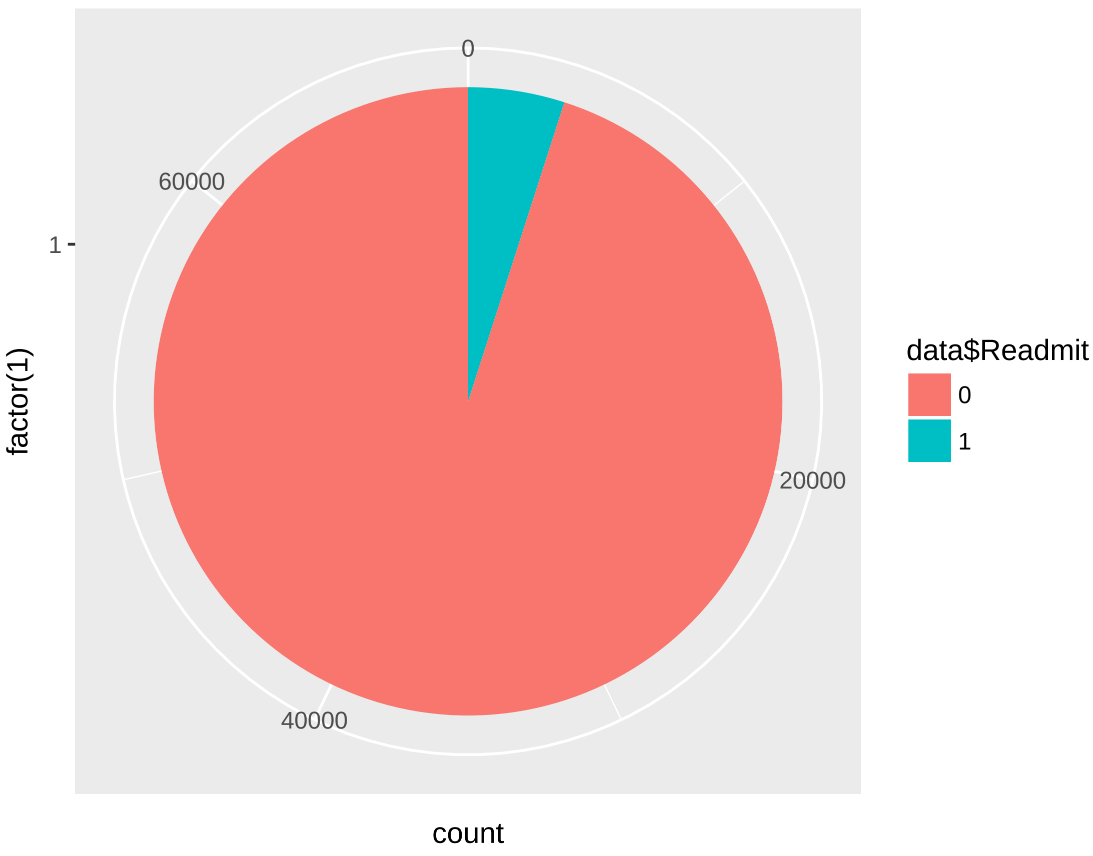
```r
# Factorize some columns in data
data$Readmit <- as.factor(data$Readmit)
data$race <- as.factor(data$race)
data$Diag1 <- as.factor(data$Diag1)
data$Diag2 <- as.factor(data$Diag2)
data$Diag3 <- as.factor(data$Diag3)
data$HbA1c <- as.factor(data$HbA1c)
data$MedicalSpeciality <-as.factor(data$MedicalSpeciality)

plot(data$Readmit, main="Readmit")
```



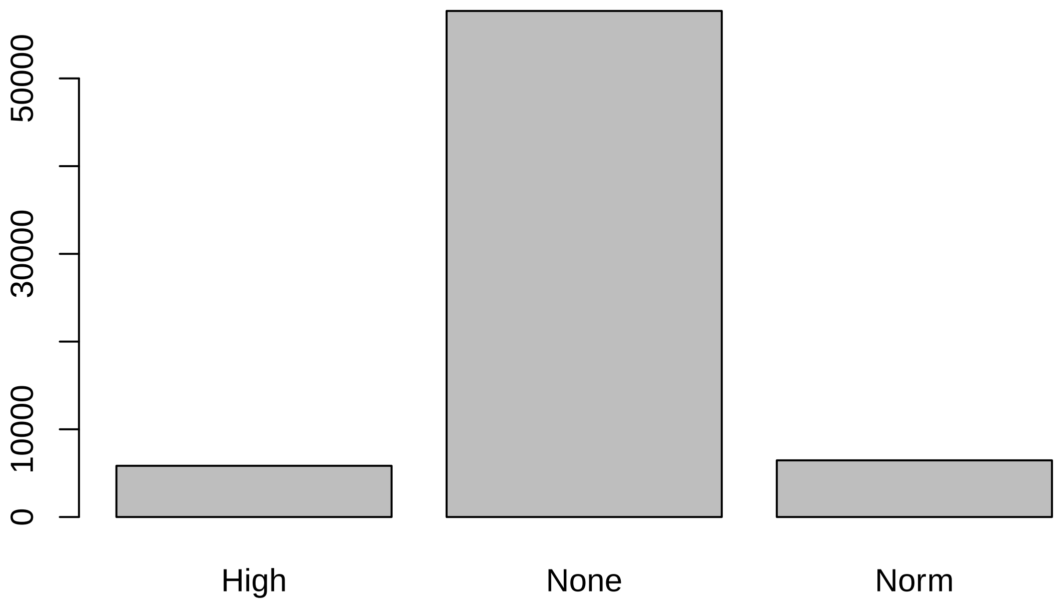
```r
plot(data$MedicalSpeciality, main="Medical Speciality")
```



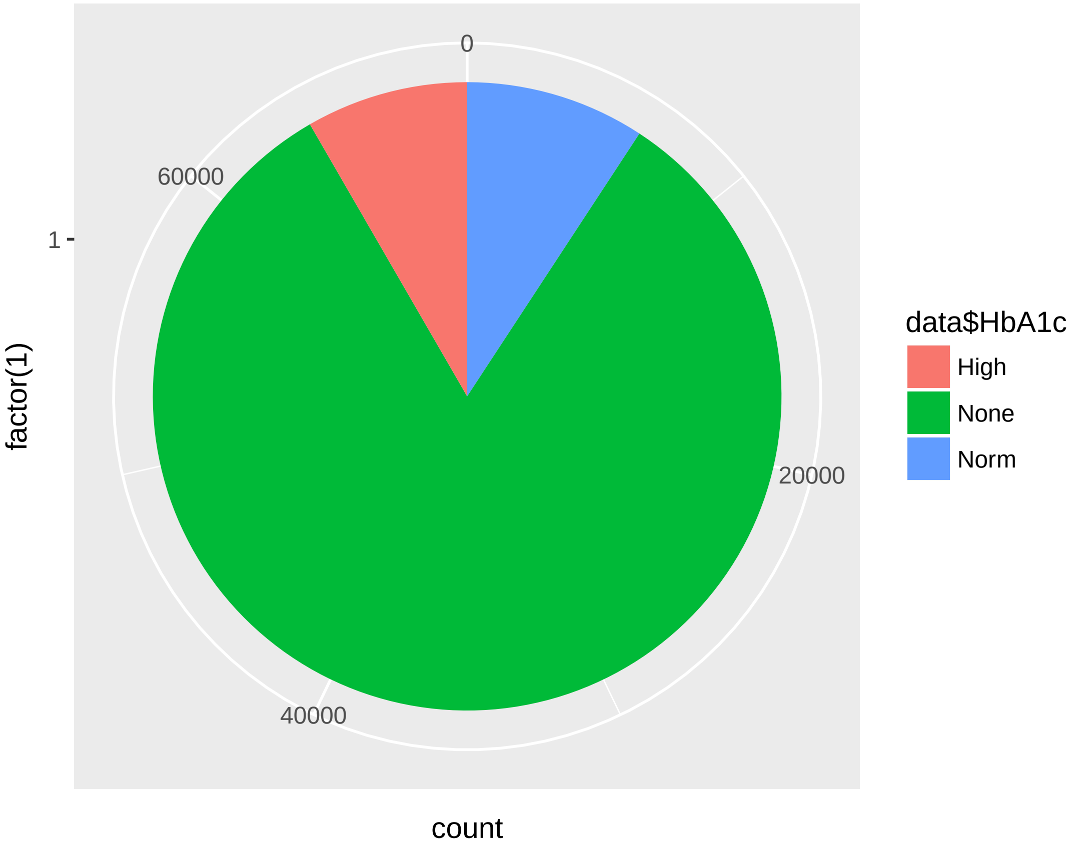
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Readmit))+coord_polar(theta="y")
```



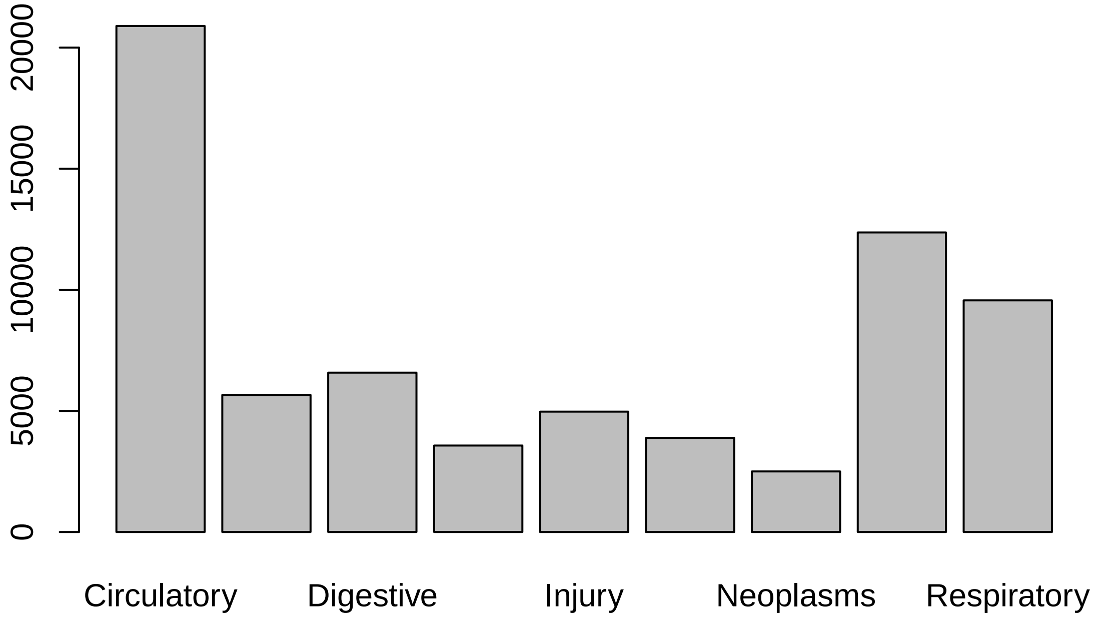
```
plot(data$HbA1c)
```



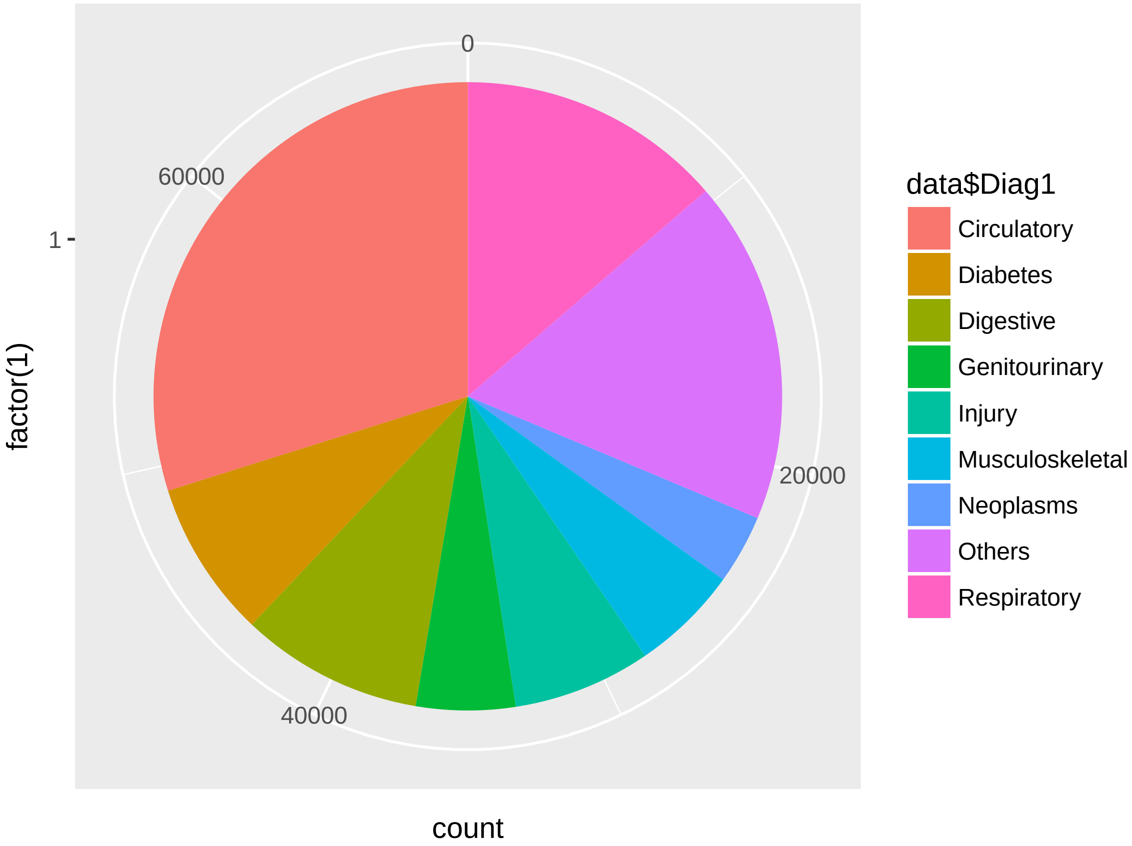
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$HbA1c))+coord_polar(theta="y")
```



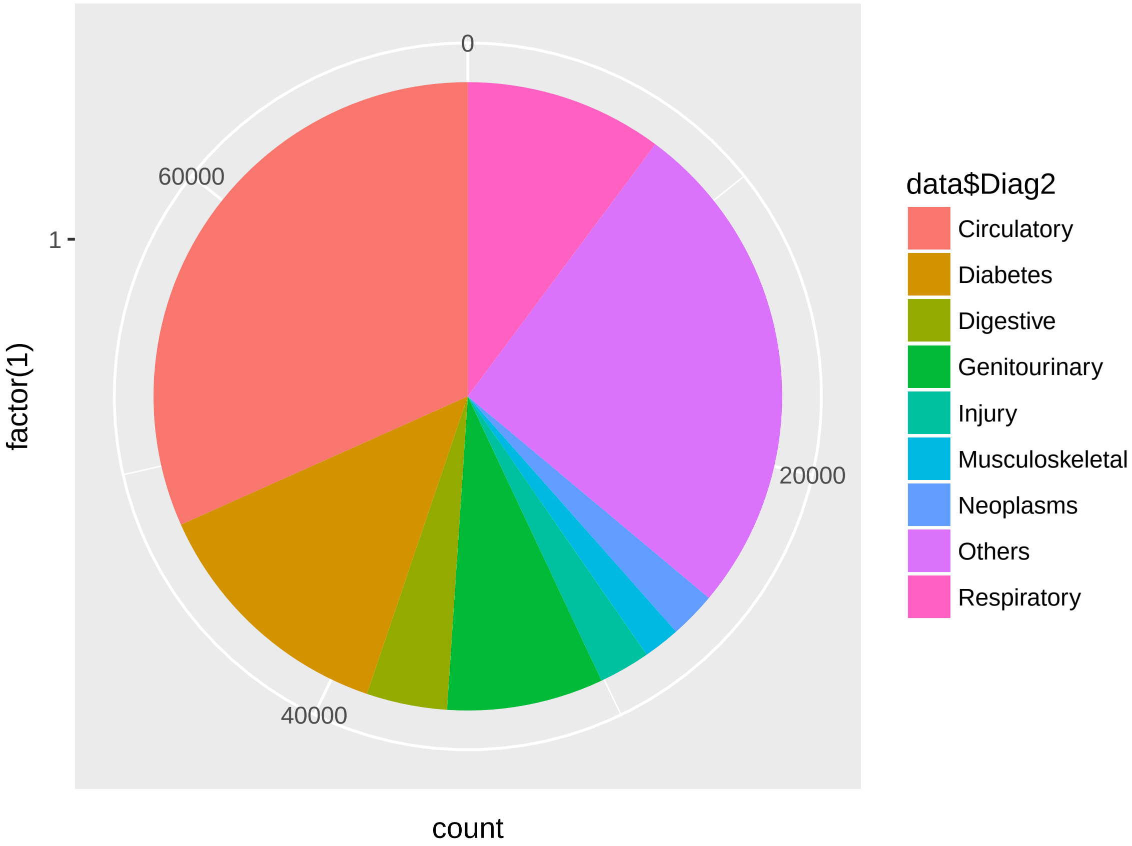
```r
plot(data$Diag1)
```



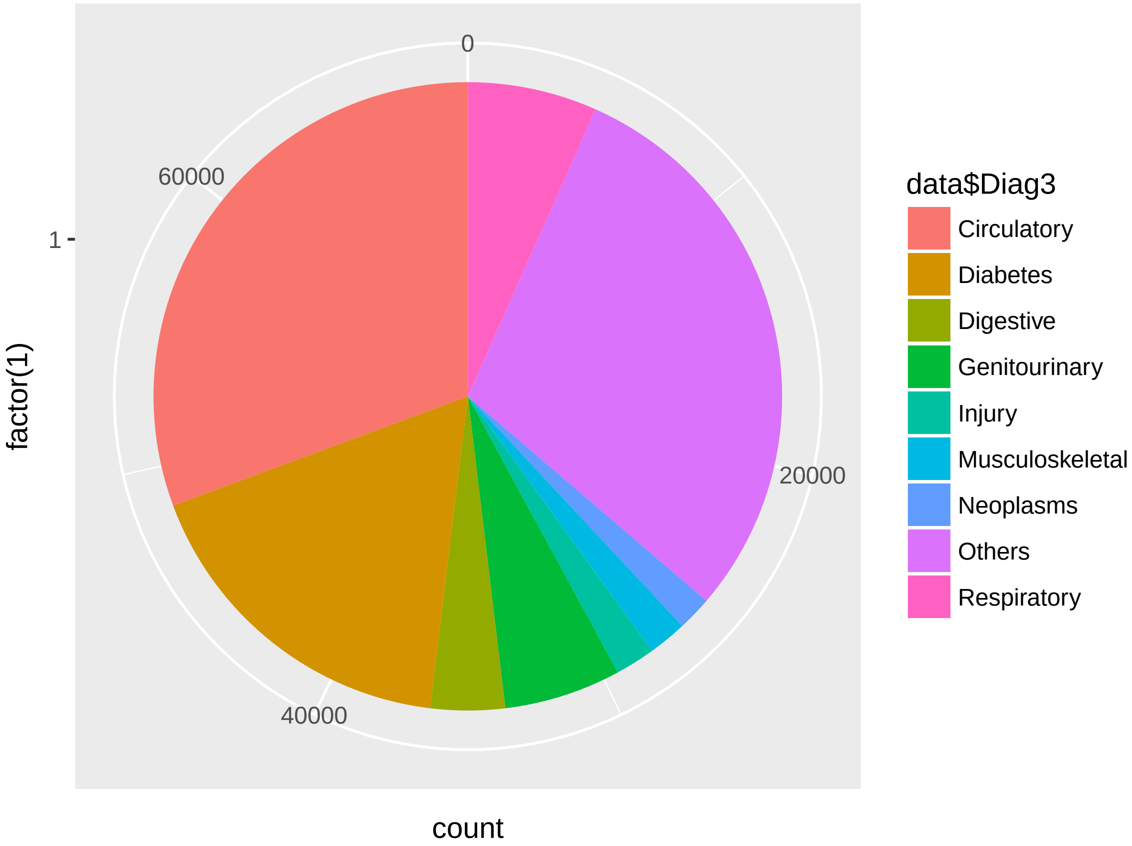
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Diag1))+coord_polar(theta="y")
```



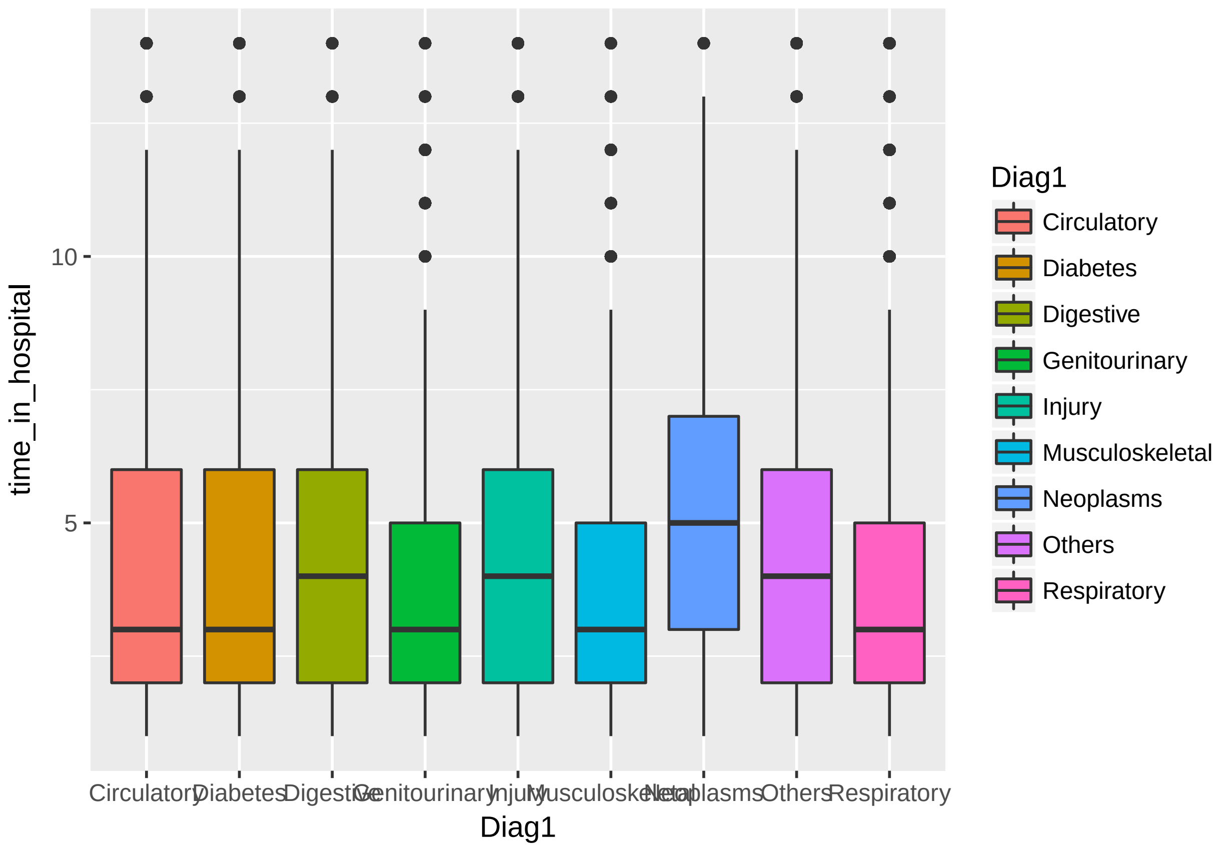
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Diag2))+coord_polar(theta="y")
```



```r
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data$Diag3))+coord_polar(theta="y")
```



```
g <- ggplot(data,aes(x=Diag1, y=time_in_hospital))
g + geom_boxplot(aes(fill=Diag1))
```



Further data cleaning, remove some columns, merge column HbA1c & change to 'Reaction'

```
data = select(data, -diag_1, -diag_2, -diag_3, -admission_source_id, -num_procedures, -num_medications,

data = select(data, -Diag2, -Diag3, -number_diagnoses, -max_glu_serum, -insulin, -diabetesMed)

data = sqldf("select case when HbA1c is 'High' and change is 'Ch' then 'High&Ch' when HbA1c is 'High' ar

# Factorize
data$Reaction <- as.factor(data$Reaction)
data$race <- as.factor(data$race)
data$discharge_disposition_id <- as.factor(data$discharge_disposition_id)
data$admission_type_id <- as.factor(data$admission_type_id)

# Remove duplicates
data = select(data, -Diag1, -HbA1c, -change)
```

Correlation of numeric factors

```
#data[is.na(data)] <- 0
#summary(data)
#data[] <- lapply(data, function(x) {
#    if(is.factor(x)) as.numeric(as.character(x)) else x
#})
#sapply(data, class)

#c <- cor(data[], use= "pairwise.complete.obs")
```

```r
#corrplot(c)

# p = ggpairs(data_simp)
# p_ <- GGally::print_if_interactive
# data_simp[] <- lapply(data_simp, function(x) {
#     if(is.factor(x)) as.numeric(as.character(x)) else x
# })
# sapply(data_simp, class)
# p_(p)

# Bad result!
```

Quick PCA with numeric variables

```r
# y <- select(data, Readmit)
# X <- select(data, time_in_hospital, num_procedures,
#              number_outpatient, number_emergency, number_inpatient, number_diagnoses,
#              max_glu_serum, insulin, change)
# # no rotation
# pca_noRot <- principal(X, nfactors = 5, rotate = "none")
# rotation2_noRot <- data.frame(cbind(pca_noRot$score, y))
# head(rotation2_noRot)
# pca_noRot$loadings
```

pc1 number of medications and time in hosptial pc2 number of in-patient visits and emergency pc3 number of procedures pc4 number of out-patient visits pc5 number of diagnoses

Logit regression

```r
logisticPseudoR2s <- function(LogModel) {
  dev <- LogModel$deviance
  nullDev <- LogModel$null.deviance
  modelN <-  length(LogModel$fitted.values)
  R.l <-  1 -  dev / nullDev
  R.cs <- 1- exp ( -(nullDev - dev) / modelN)
  R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN))))
  cat("Pseudo R^2 for logistic regression\n")
  cat("Hosmer and Lemeshow R^2  ", round(R.l, 3), "\n")
  cat("Cox and Snell R^2        ", round(R.cs, 3), "\n")
  cat("Nagelkerke R^2           ", round(R.n, 3),    "\n")
}
```

Chisq test, select important factors (Gender could be dropped)

```r
#anova(linModel_noRot, test="Chisq")
#
#                          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
# NULL                                     69989      27608
# Reaction                  3    10.73     69986      27598 0.0132927 *
# Pri_diag                  8    31.23     69978      27566 0.0001279 ***
# MedicalSpeciality        58   244.94     69920      27321 < 2.2e-16 ***
# race                      5    88.91     69915      27233 < 2.2e-16 ***
# gender                    2     0.69     69913      27232 0.7099052
# age                       9   160.53     69904      27071 < 2.2e-16 ***
# admission_type_id         7   174.54     69897      26897 < 2.2e-16 ***
# discharge_disposition_id 20   701.57     69877      26195 < 2.2e-16 ***
# time_in_hospital          1    21.16     69876      26174 4.218e-06 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data Merging - Select the attributes and groups we use to train the model.

```
#summary(data2)
data2 = data
#library(sqldf)
data2$race<- as.factor(data2$race)
data2 = select(data2, -gender)
data2 = sqldf("select case discharge_disposition_id when 1 then 'Home' else 'Other' end as Discharge, *
data2 = sqldf("select case admission_type_id when 1 or 2 then 'Emergency' when 7 then 'referral' else '
data2 = sqldf("select case race when 'AfricanAmerican' then 'AfricanAmerican' when 'Caucasian' then 'Ca
data2 = sqldf("select case age when '[30-40)' then '[30, 60)' when '[40-50)' then '[30, 60)' when '[50-
data2 = sqldf("select case MedicalSpeciality when 'Missing' then 'Missing' when 'Cardiology' then 'Card
data2$race_<-as.factor(data2$race_)
data2$Admission<-as.factor(data2$Admission)
data2$Discharge<-as.factor(data2$Discharge)
data2$Pri_diag <- as.factor(data2$Pri_diag)
data2$Age_ <- as.factor(data2$Age_)
data2$Medical_speciality <- as.factor(data2$Medical_speciality)
data2 = select(data2, -race, -admission_type_id, -discharge_disposition_id, -age, -MedicalSpeciality)
summary(data2)
```

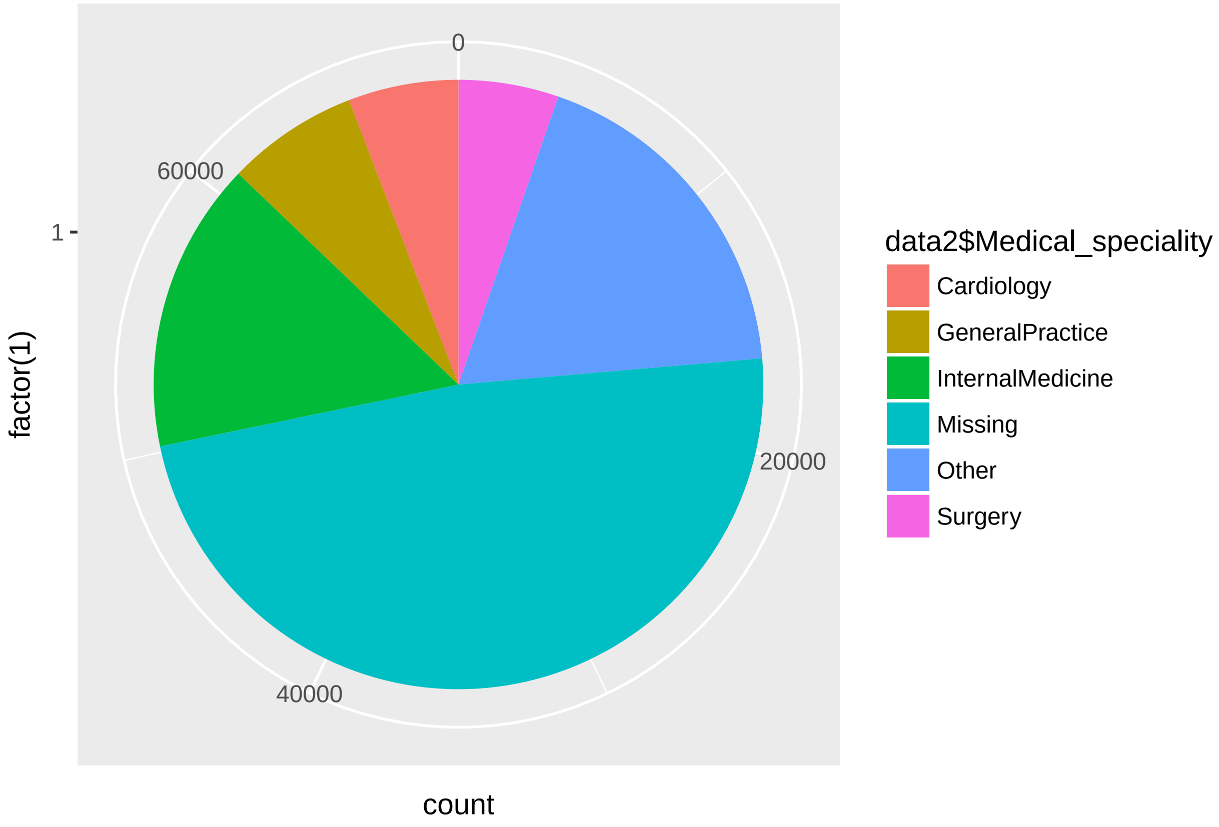
```
##          Medical_speciality        Age_                  race_
##  Cardiology      : 4094    [30, 60) :21587   AfricanAmerican:12656
##  GeneralPractice : 4894    [60, 100):47727   Caucasian      :52352
##  InternalMedicine:10788    <30      :  676   Missing        : 1850
##  Missing         :33676                      Other          : 3132
##  Other           :12821
##  Surgery         : 3717
##
##      Admission     Discharge         Reaction           Pri_diag
##  Emergency:36098   Home :43721   High&Ch : 3784   Circulatory:20894
##  Other    :33876   Other:26269   High&Not: 2053   Others     :12368
##  referral :   16                 None    :57691   Respiratory: 9564
##                                  Norm    : 6462   Digestive  : 6579
##                                                   Diabetes   : 5660
##                                                   Injury     : 4969
##                                                   (Other)    : 9956
##  Readmit   time_in_hospital
##  0:66521   Min.   : 1.000
##  1: 3469   1st Qu.: 2.000
##            Median : 4.000
##            Mean   : 4.302
##            3rd Qu.: 6.000
##            Max.   :14.000
##
```

```
 #       Medical_speciality        Age_                  race_          Admission     Discharge
 # Cardiology      : 4094    [30, 60) :21587   AfricanAmerican:12656   Emergency:36098   Home :43721
 # GeneralPractice : 4894    [60, 100):47727   Caucasian      :52352   Other    :33876   Other:26269
 # InternalMedicine:10788    <30      :  676   Missing        : 1850   referral :   16
 # Missing         :33676                      Other          : 3132
```

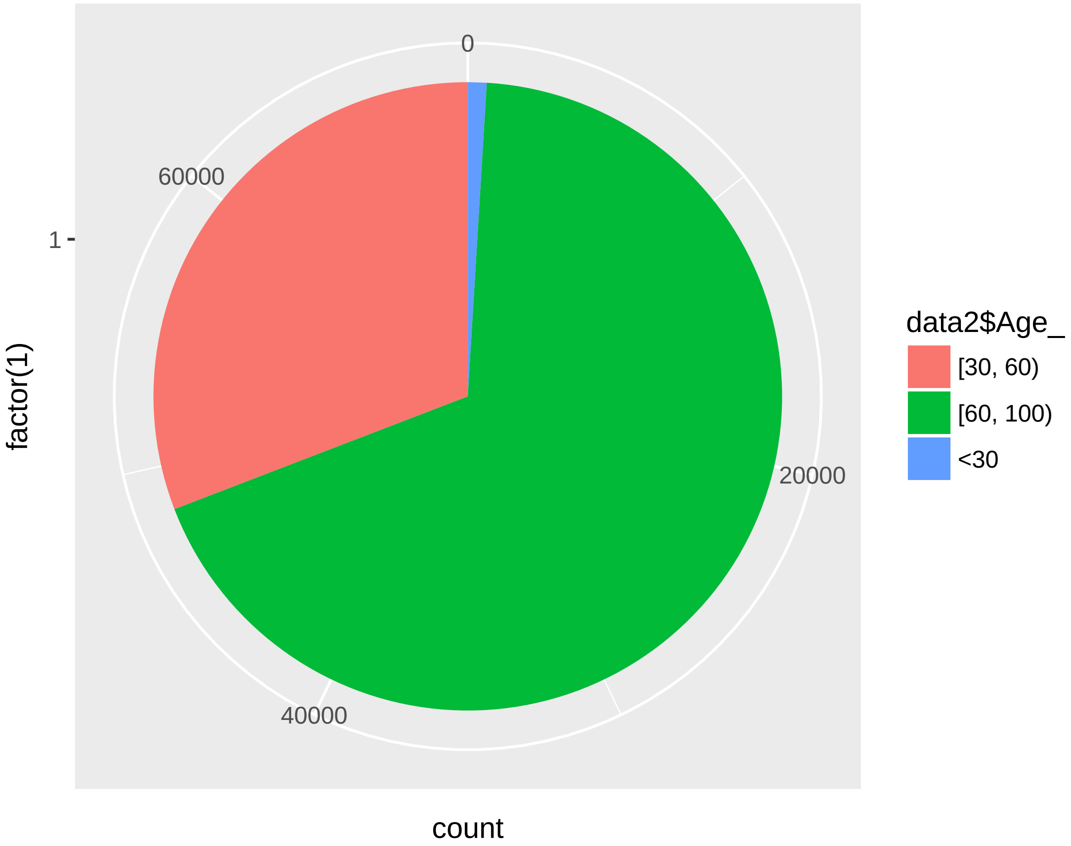
```
# Other            :12821
# Surgery          : 3717
#
#        Pri_diag     Readmit   time_in_hospital
# Circulatory:20894   0:66521   Min.   : 1.000
# Others     :12368   1: 3469   1st Qu.: 2.000
# Respiratory: 9564             Median : 4.000
# Digestive  : 6579             Mean   : 4.302
# Diabetes   : 5660             3rd Qu.: 6.000
# Injury     : 4969             Max.   :14.000
# (Other)    : 9956
```

Data Analysis

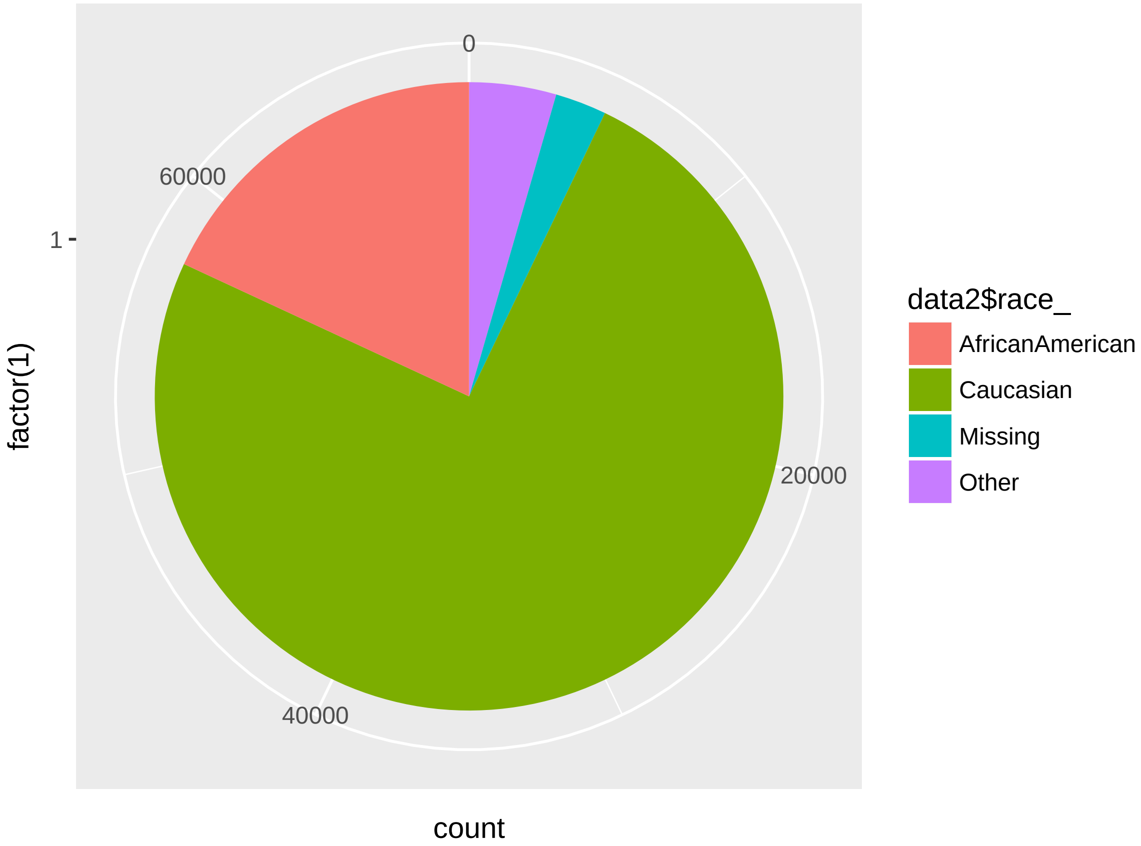
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Medical_speciality))+coord_polar(theta="y")
```



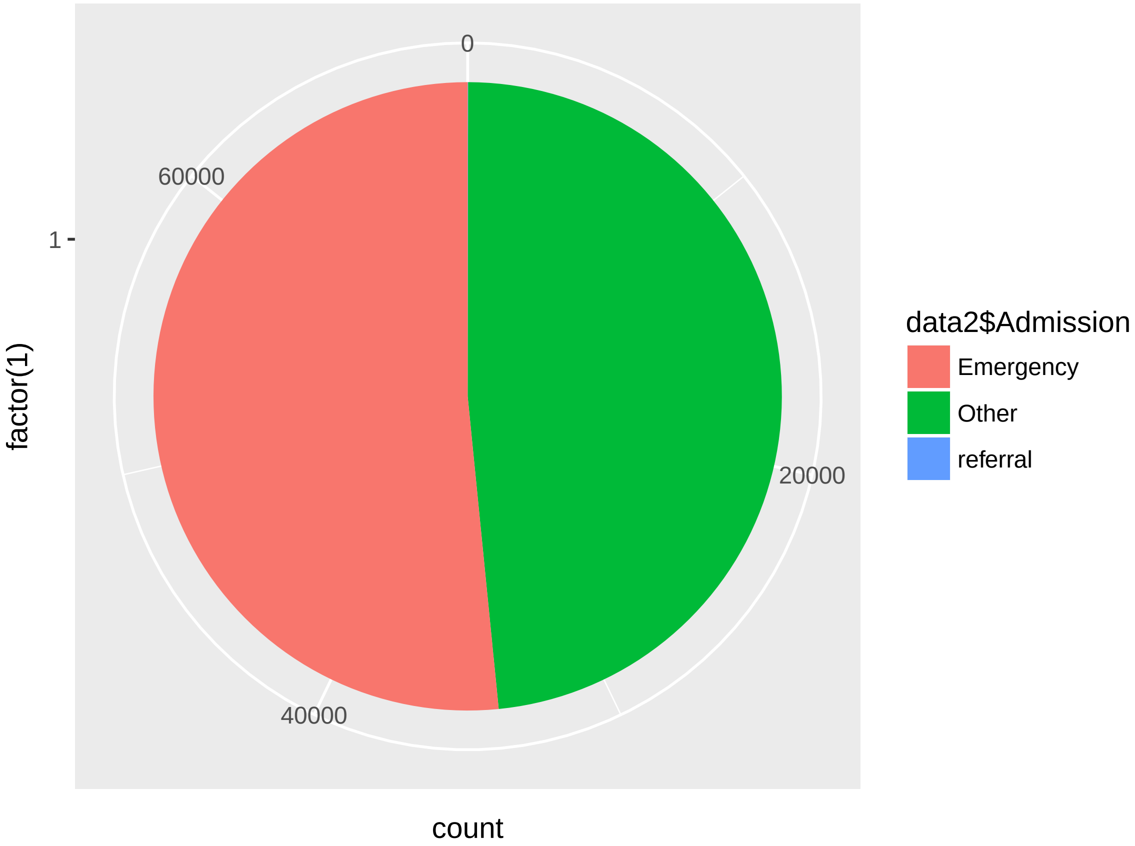
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Age_))+coord_polar(theta="y")
```



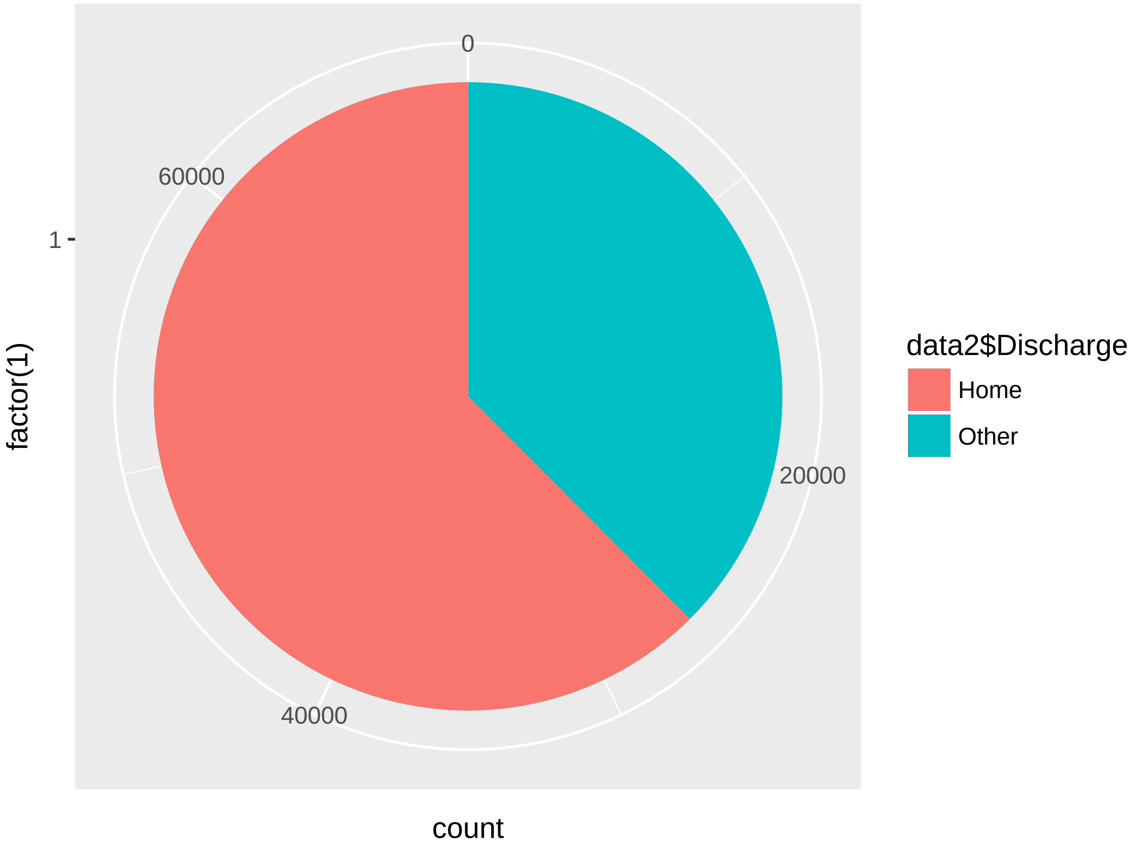
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$race_))+coord_polar(theta="y")
```



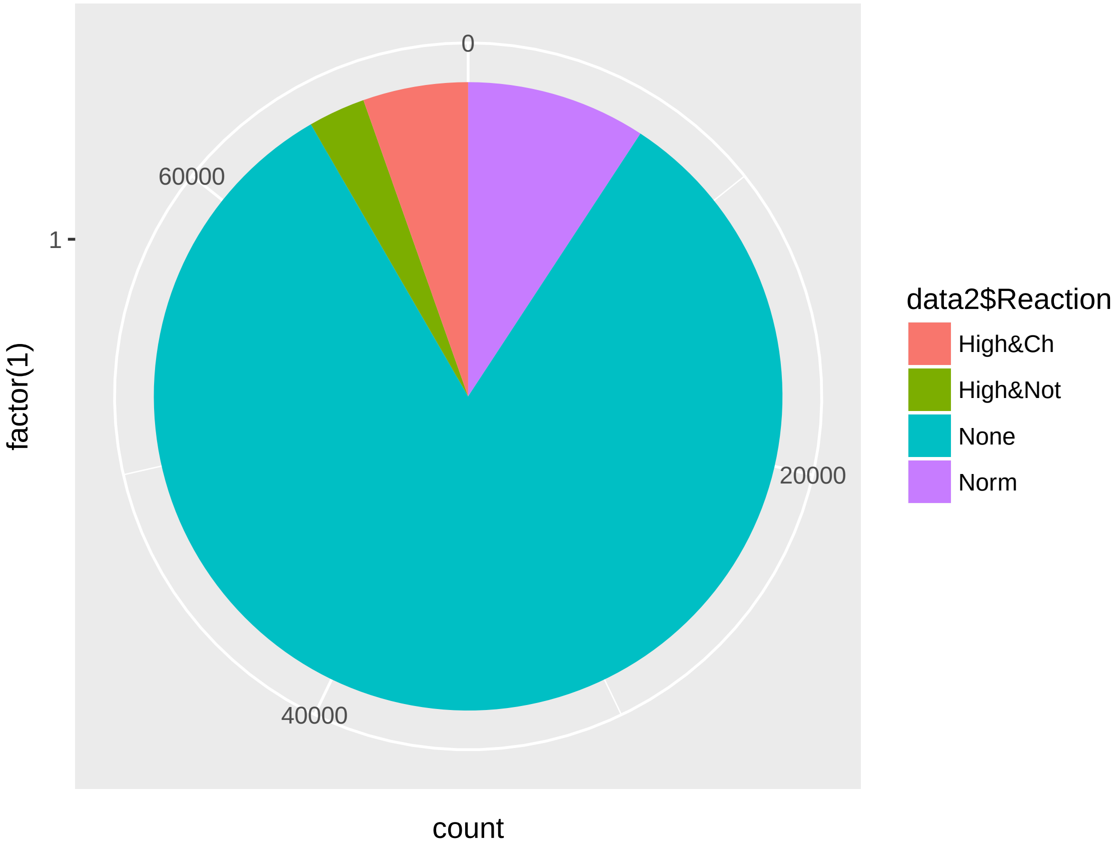
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Admission))+coord_polar(theta="y")
```



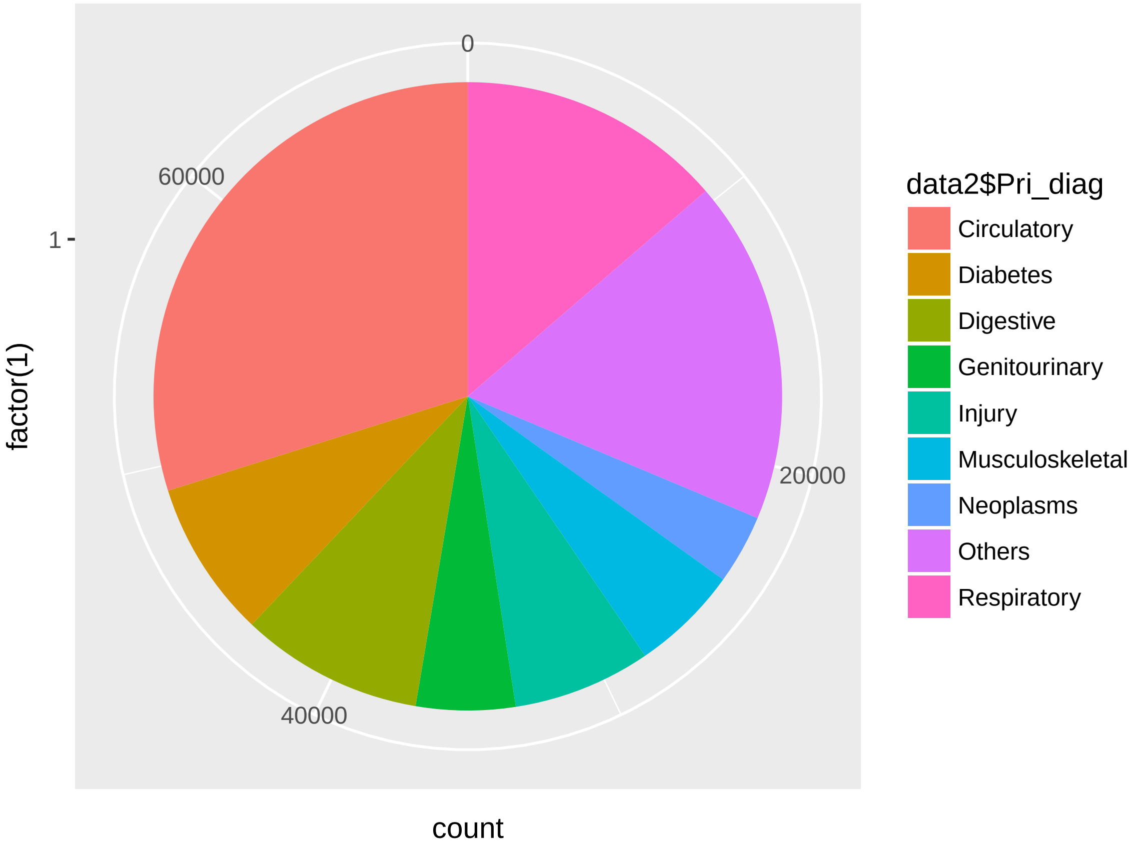
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Discharge))+coord_polar(theta="y")
```



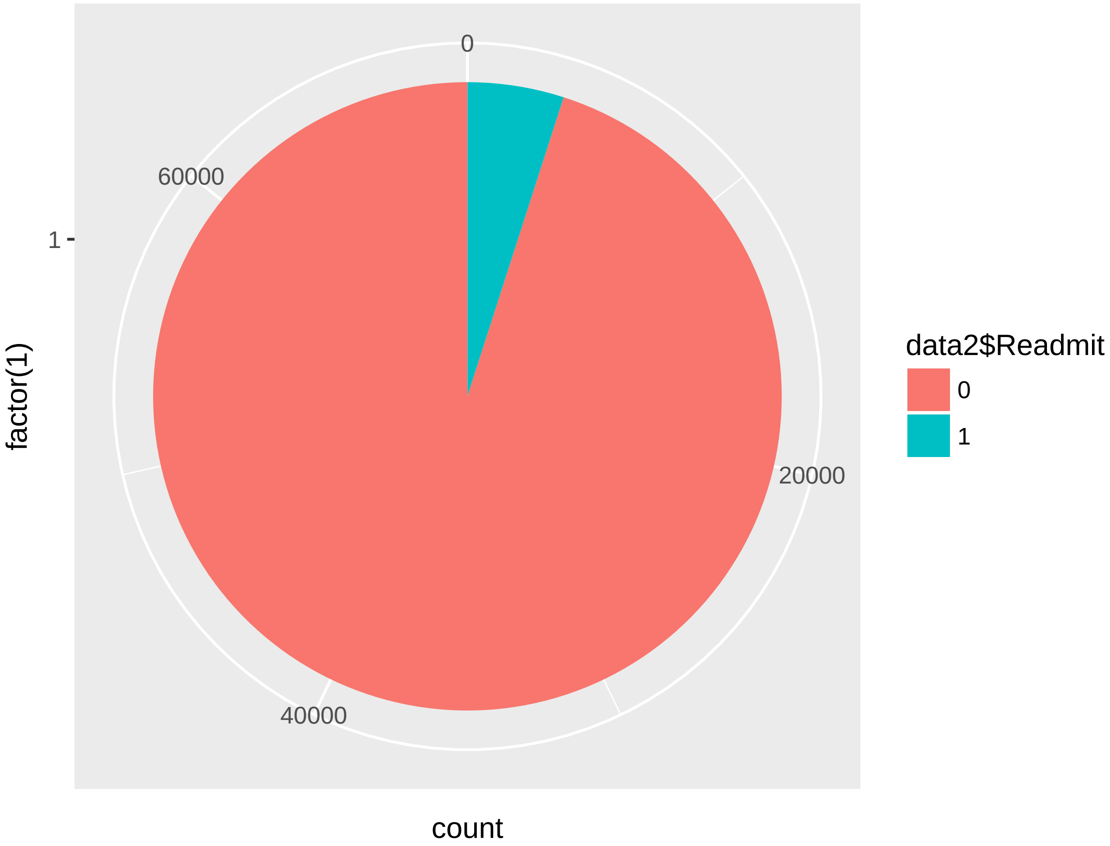
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Reaction))+coord_polar(theta="y")
```



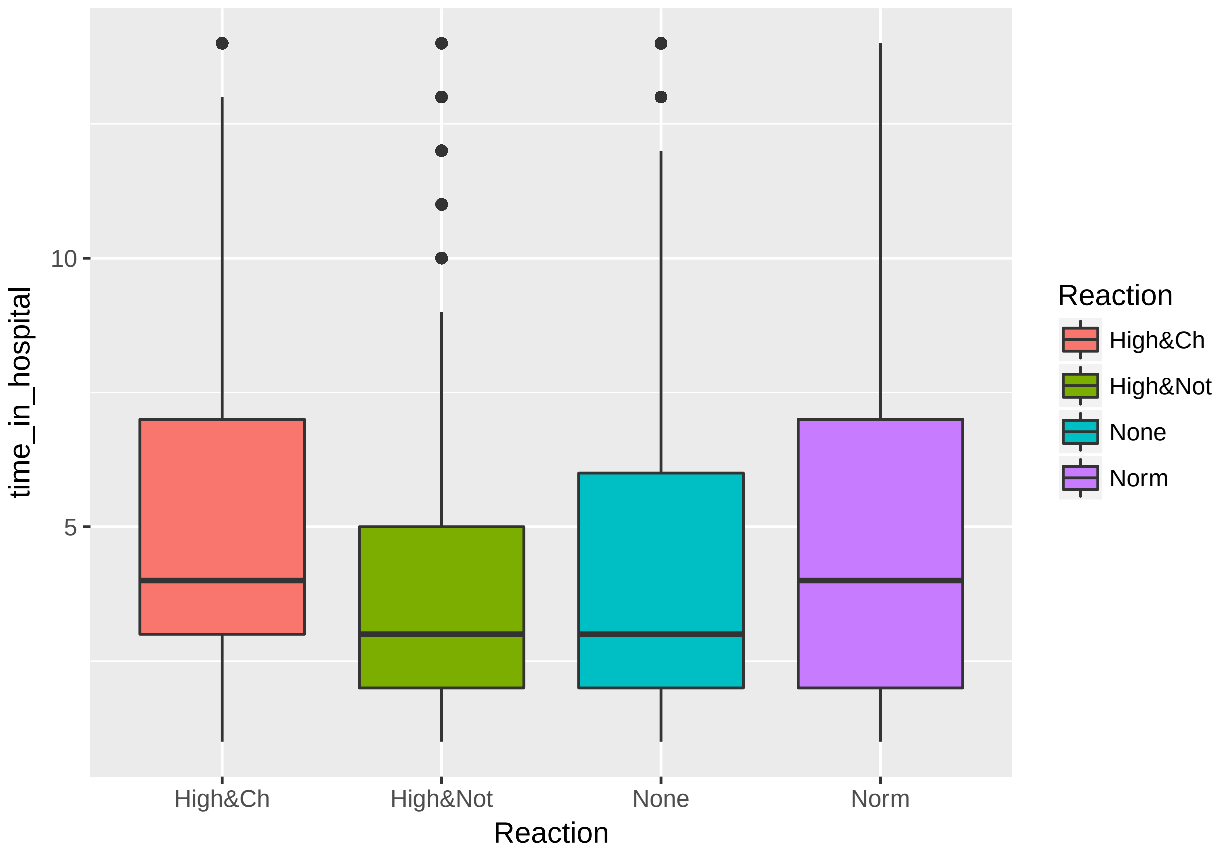
```r
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Pri_diag))+coord_polar(theta="y")
```



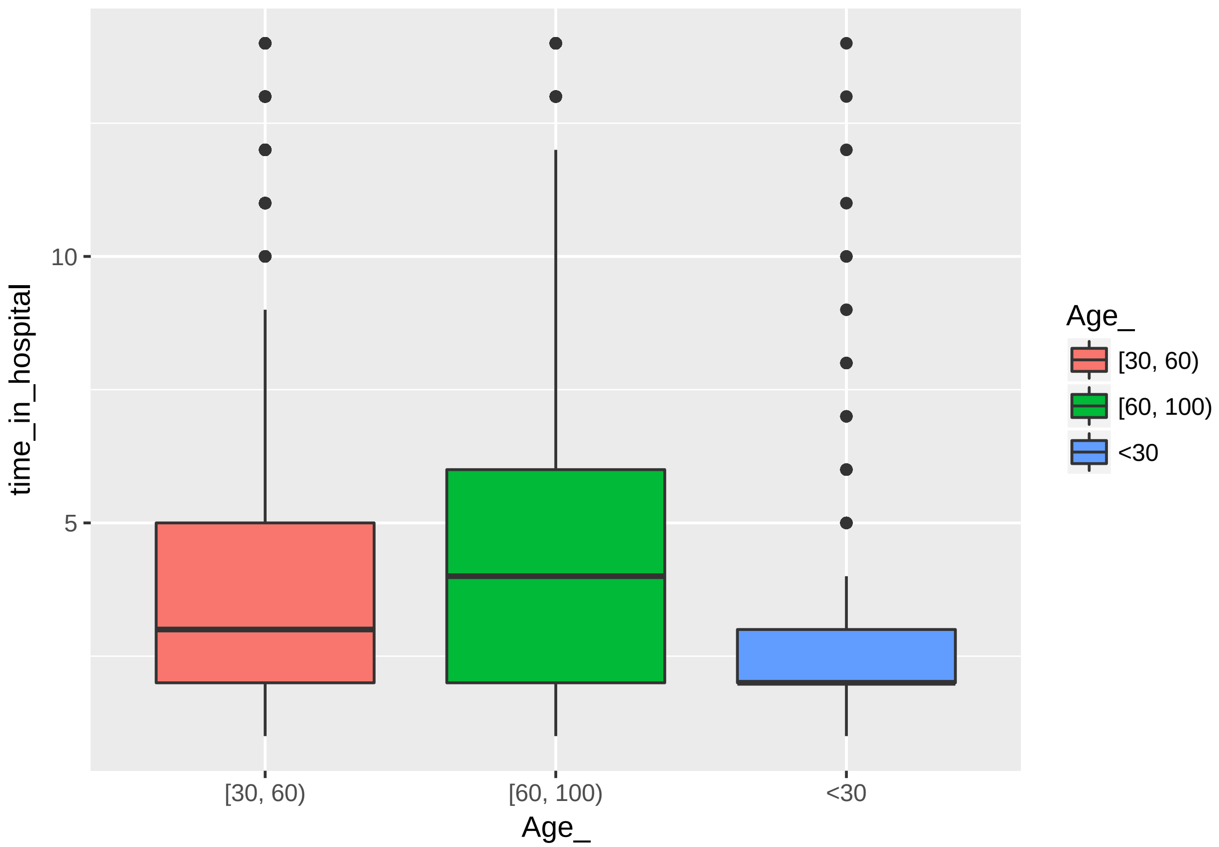
```
ggplot(data)+geom_bar(width=1, aes(x=factor(1),fill=data2$Readmit))+coord_polar(theta="y")
```



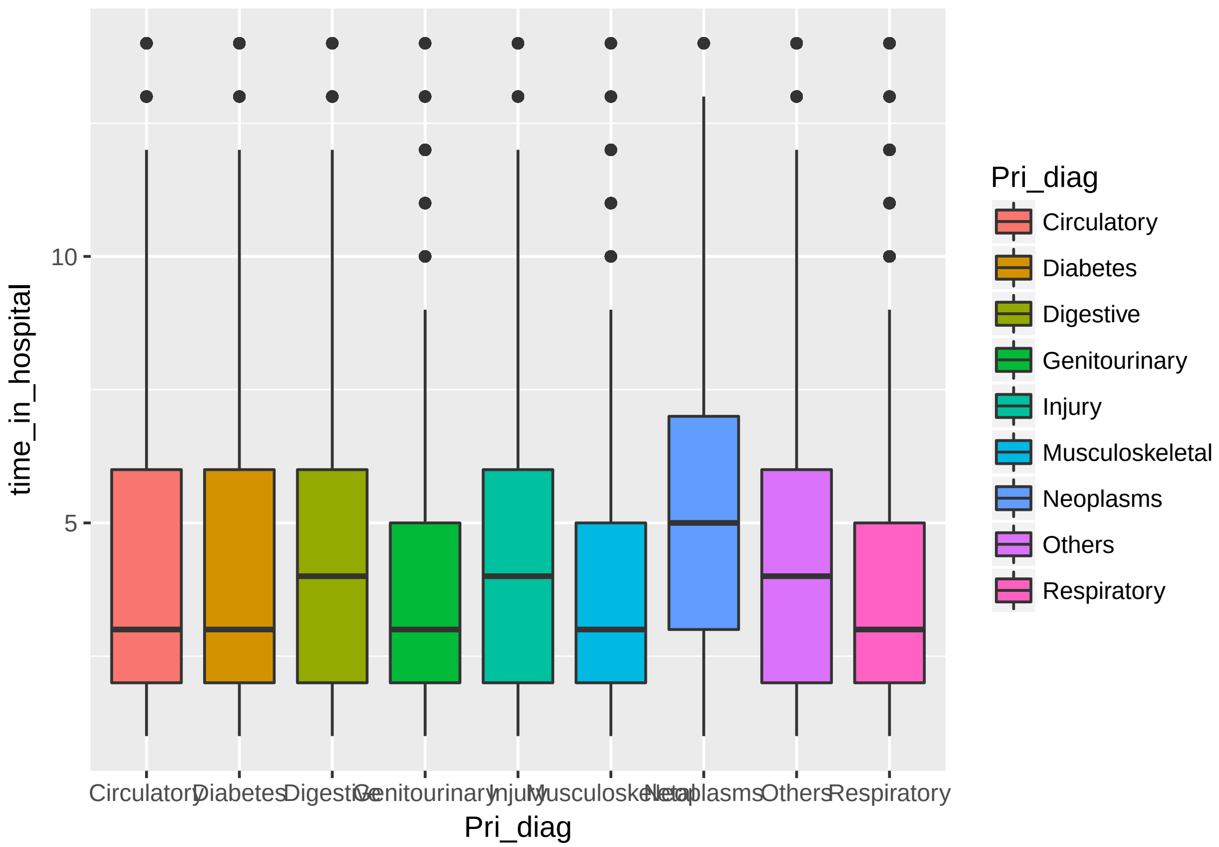
```
g <- ggplot(data2,aes(x=Reaction, y=time_in_hospital))
g + geom_boxplot(aes(fill=Reaction))
```



```
g <- ggplot(data2,aes(x=Age_, y=time_in_hospital))
g + geom_boxplot(aes(fill=Age_))
```



```
g <- ggplot(data2,aes(x=Pri_diag, y=time_in_hospital))
g + geom_boxplot(aes(fill=Pri_diag))
```



Relevel for references

```
data2$race_ <- relevel(data2$race_, ref = 'AfricanAmerican')
data2$Reaction <- relevel(data2$Reaction, ref = 'None')
data2$Pri_diag <- relevel(data2$Pri_diag, ref = 'Diabetes')
```

Basic model, order-one

```
linModel_no2 <- glm(Readmit ~ . , data2, family = binomial)
#summary(linModel_no2)

# Coefficients:
#                                     Estimate Std. Error z value Pr(>|z|)
# (Intercept)                        -3.720905   0.116511 -31.936  < 2e-16 ***
# Medical_specialityGeneralPractice   0.037285   0.102993   0.362 0.717339
# Medical_specialityInternalMedicine  0.299026   0.089461   3.343 0.000830 ***
# Medical_specialityMissing           0.014635   0.083602   0.175 0.861037
# Medical_specialityOther            -0.285336   0.093223  -3.061 0.002207 **
# Medical_specialitySurgery          -0.079796   0.112335  -0.710 0.477490
# Age_[60, 100)                       0.348403   0.044247   7.874 3.43e-15 ***
# Age_<30                            -0.314197   0.277667  -1.132 0.257819
# race_Caucasian                      0.344032   0.052194   6.591 4.36e-11 ***
# race_Missing                       -0.114495   0.137427  -0.833 0.404768
# race_Other                          0.371437   0.096633   3.844 0.000121 ***
# AdmissionOther                     -0.006926   0.038174  -0.181 0.856024
# Admissionreferral                  -9.825065  80.300762  -0.122 0.902619
# DischargeOther                      0.511007   0.038326  13.333  < 2e-16 ***
# ReactionHigh&Ch                    -0.088169   0.082880  -1.064 0.287413
# ReactionHigh&Not                   -0.181984   0.118067  -1.541 0.123230
# ReactionNorm                       -0.127005   0.062829  -2.021 0.043234 *
# Pri_diagCirculatory                -0.153162   0.070793  -2.164 0.030500 *
# Pri_diagDigestive                  -0.272778   0.087827  -3.106 0.001897 **
# Pri_diagGenitourinary              -0.297167   0.102871  -2.889 0.003868 **
# Pri_diagInjury                     -0.031670   0.087207  -0.363 0.716487
# Pri_diagMusculoskeletal            -0.238246   0.102064  -2.334 0.019581 *
# Pri_diagNeoplasms                  -0.104240   0.110581  -0.943 0.345857
# Pri_diagOthers                     -0.135494   0.074824  -1.811 0.070168 .
# Pri_diagRespiratory                -0.268682   0.079986  -3.359 0.000782 ***
# time_in_hospital                    0.039181   0.005803   6.752 1.46e-11 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Select all pairwise attributes to train a logit model

```
#linModel_no4 <- glm(Readmit ~ .^2 , data2, family = binomial)
#summary(linModel_no4)
```

Reserve the important pairwise attributes and train model We found Age_*Medical_speciality useless, dropped.

```
linModel_core_num <- glm(Readmit ~ Discharge + race_ + Admission + Medical_speciality + time_in_hospita
                      Pri_diag*Discharge + race_*Discharge
                     + Discharge*time_in_hospital + Medical_speciality*Discharge
                     + time_in_hospital*Medical_speciality
                     + time_in_hospital* Pri_diag + Reaction*Pri_diag,
                       data = data2, family = binomial)
# summary(linModel_core_num)

# Coefficients:
#                                                       Estimate Std. Error z value Pr(>|z|)
# (Intercept)                                          -4.145212   0.197483 -20.990  < 2e-16 ***
# DischargeOther                                        0.901349   0.244866   3.681 0.000232 ***
# race_Caucasian                                        0.437519   0.074212   5.895 3.74e-09 ***
# race_Missing                                         -0.296718   0.210110  -1.412 0.157891
# race_Other                                            0.302575   0.135545   2.232 0.025596 *
# AdmissionOther                                       -0.005596   0.038397  -0.146 0.884123
# Admissionreferral                                    -9.784584  80.554871  -0.121 0.903323
# Medical_specialityGeneralPractice                     0.401829   0.188392   2.133 0.032930 *
# Medical_specialityInternalMedicine                    0.500129   0.157556   3.174 0.001502 **
# Medical_specialityMissing                             0.363802   0.144066   2.525 0.011562 *
# Medical_specialityOther                              -0.297803   0.169253  -1.760 0.078490 .
# Medical_specialitySurgery                             0.256724   0.213291   1.204 0.228732
# time_in_hospital                                      0.136690   0.032557   4.198 2.69e-05 ***
# Age_[60, 100)                                         0.339596   0.044287   7.668 1.75e-14 ***
# Age_<30                                              -0.157528   0.282762  -0.557 0.577455
# Pri_diagCirculatory                                  -0.037004   0.138808  -0.267 0.789787
# Pri_diagDigestive                                    -0.342643   0.171152  -2.002 0.045287 *
# Pri_diagGenitourinary                                -0.322777   0.202376  -1.595 0.110725
# Pri_diagInjury                                       -0.009665   0.185387  -0.052 0.958422
# Pri_diagMusculoskeletal                              -0.703889   0.238734  -2.948 0.003194 **
# Pri_diagNeoplasms                                     0.221054   0.216383   1.022 0.306976
# Pri_diagOthers                                       -0.027303   0.148768  -0.184 0.854386
# Pri_diagRespiratory                                 -0.425242   0.157461  -2.701 0.006921 **
# ReactionHigh&Ch                                     -0.750748   0.215940  -3.477 0.000508 ***
# ReactionHigh&Not                                    -0.531529   0.295624  -1.798 0.072178 .
# ReactionNorm                                        -0.207005   0.211000  -0.981 0.326560
# DischargeOther:Pri_diagCirculatory                  -0.051205   0.145939  -0.351 0.725688
# DischargeOther:Pri_diagDigestive                     0.139439   0.183223   0.761 0.446636
# DischargeOther:Pri_diagGenitourinary                -0.243134   0.213656  -1.138 0.255133
# DischargeOther:Pri_diagInjury                        0.095687   0.187207   0.511 0.609261
# DischargeOther:Pri_diagMusculoskeletal               0.453188   0.234151   1.935 0.052935 .
# DischargeOther:Pri_diagNeoplasms                    -0.125321   0.228869  -0.548 0.583991
# DischargeOther:Pri_diagOthers                        0.348167   0.154750   2.250 0.024457 *
# DischargeOther:Pri_diagRespiratory                   0.253834   0.165773   1.531 0.125716
# DischargeOther:race_Caucasian                       -0.186434   0.103542  -1.801 0.071773 .
# DischargeOther:race_Missing                          0.331627   0.278537   1.191 0.233809
# DischargeOther:race_Other                            0.190659   0.193892   0.983 0.325447
# DischargeOther:time_in_hospital                     -0.036059   0.011828  -3.049 0.002299 **
# DischargeOther:Medical_specialityGeneralPractice    -0.121850   0.223700  -0.545 0.585959
# DischargeOther:Medical_specialityInternalMedicine   -0.121562   0.196055  -0.620 0.535230
# DischargeOther:Medical_specialityMissing            -0.362101   0.184227  -1.966 0.049355 *
# DischargeOther:Medical_specialityOther              -0.007635   0.205228  -0.037 0.970325
# DischargeOther:Medical_specialitySurgery             0.503798   0.245500   2.052 0.040157 *
# Medical_specialityGeneralPractice:time_in_hospital  -0.071874   0.033733  -2.131 0.033116 *
# Medical_specialityInternalMedicine:time_in_hospital -0.037413   0.028476  -1.314 0.188901
# Medical_specialityMissing:time_in_hospital          -0.041631   0.026728  -1.558 0.119327
# Medical_specialityOther:time_in_hospital            -0.007006   0.029872  -0.235 0.814562
# Medical_specialitySurgery:time_in_hospital          -0.133988   0.038119  -3.515 0.000440 ***
# time_in_hospital:Pri_diagCirculatory                -0.041732   0.021838  -1.911 0.056013 .
# time_in_hospital:Pri_diagDigestive                  -0.017480   0.027729  -0.630 0.528445
# time_in_hospital:Pri_diagGenitourinary               0.011861   0.033335   0.356 0.721985
# time_in_hospital:Pri_diagInjury                     -0.030957   0.027655  -1.119 0.262977
# time_in_hospital:Pri_diagMusculoskeletal             0.009490   0.036182   0.262 0.793098
# time_in_hospital:Pri_diagNeoplasms                  -0.072089   0.033778  -2.134 0.032824 *
# time_in_hospital:Pri_diagOthers                     -0.080533   0.022926  -3.513 0.000443 ***
# time_in_hospital:Pri_diagRespiratory                -0.009011   0.024637  -0.366 0.714570
# Pri_diagCirculatory:ReactionHigh&Ch                  1.044042   0.250548   4.167 3.09e-05 ***
# Pri_diagDigestive:ReactionHigh&Ch                    0.286802   0.507700   0.565 0.572139
# Pri_diagGenitourinary:ReactionHigh&Ch                0.819908   0.433084   1.893 0.058333 .
# Pri_diagInjury:ReactionHigh&Ch                       0.190152   0.509654   0.373 0.709075
# Pri_diagMusculoskeletal:ReactionHigh&Ch              0.880868   0.566935   1.554 0.120247
# Pri_diagNeoplasms:ReactionHigh&Ch                    0.249116   0.761235   0.327 0.743477
# Pri_diagOthers:ReactionHigh&Ch                       0.636597   0.299036   2.129 0.033268 *
# Pri_diagRespiratory:ReactionHigh&Ch                  0.791565   0.316789   2.499 0.012465 *
# Pri_diagCirculatory:ReactionHigh&Not                 0.441651   0.351824   1.255 0.209364
# Pri_diagDigestive:ReactionHigh&Not                   0.135432   0.591980   0.229 0.819042
# Pri_diagGenitourinary:ReactionHigh&Not               0.383801   0.666438   0.576 0.564684
# Pri_diagInjury:ReactionHigh&Not                      0.671458   0.555651   1.208 0.226888
# Pri_diagMusculoskeletal:ReactionHigh&Not             1.039674   0.610377   1.703 0.088506 .
# Pri_diagNeoplasms:ReactionHigh&Not                   1.523366   0.694452   2.194 0.028263 *
# Pri_diagOthers:ReactionHigh&Not                      0.010094   0.467974   0.022 0.982792
# Pri_diagRespiratory:ReactionHigh&Not                 0.387420   0.454992   0.851 0.394498
# Pri_diagCirculatory:ReactionNorm                     0.040582   0.238436   0.170 0.864851
# Pri_diagDigestive:ReactionNorm                       0.152586   0.322777   0.473 0.636408
# Pri_diagGenitourinary:ReactionNorm                   0.104212   0.366184   0.285 0.775959
# Pri_diagInjury:ReactionNorm                          0.015613   0.319722   0.049 0.961051
# Pri_diagMusculoskeletal:ReactionNorm                 0.133092   0.375174   0.355 0.722779
# Pri_diagNeoplasms:ReactionNorm                       0.347555   0.419031   0.829 0.406864
# Pri_diagOthers:ReactionNorm                          0.266480   0.249662   1.067 0.285807
# Pri_diagRespiratory:ReactionNorm                    -0.161596   0.278172  -0.581 0.561294
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chisq test

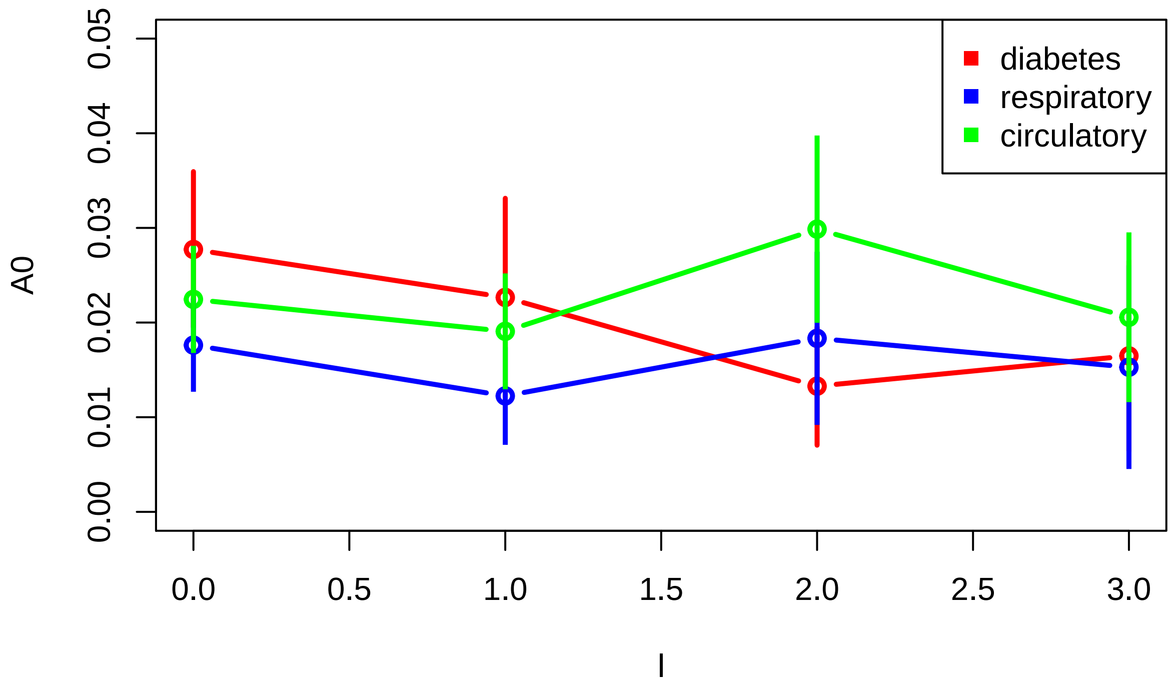
```
# anova(linModel_core_num, test="Chisq")
#
#                                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
# NULL                                                69989      27608
# Discharge                            1   365.64     69988      27243 < 2.2e-16 ***
# race_                                3    56.68     69985      27186 3.001e-12 ***
# Admission                            2     6.11     69983      27180  0.047043 *
# Medical_speciality                   5    99.47     69978      27080 < 2.2e-16 ***
# time_in_hospital                     1    50.42     69977      27030 1.243e-12 ***
# Age_                                 2    69.32     69975      26961 8.844e-16 ***
# Pri_diag                             8    22.74     69967      26938  0.003709 **
# Reaction                             3     7.00     69964      26931  0.071785 .
# Discharge:Pri_diag                   8    25.27     69956      26906  0.001400 **
# Discharge:race_                      3    11.16     69953      26894  0.010881 *
# Discharge:time_in_hospital           1     9.36     69952      26885  0.002214 **
# Discharge:Medical_speciality         5    33.35     69947      26852 3.214e-06 ***
# Medical_speciality:time_in_hospital  5    19.96     69942      26832  0.001271 **
# time_in_hospital:Pri_diag            8    25.81     69934      26806  0.001132 **
# Pri_diag:Reaction                   24    33.91     69910      26772  0.086257 .

# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Graph-probability of readmission based on reference value create reference data frame & calculate fit/confidence interval

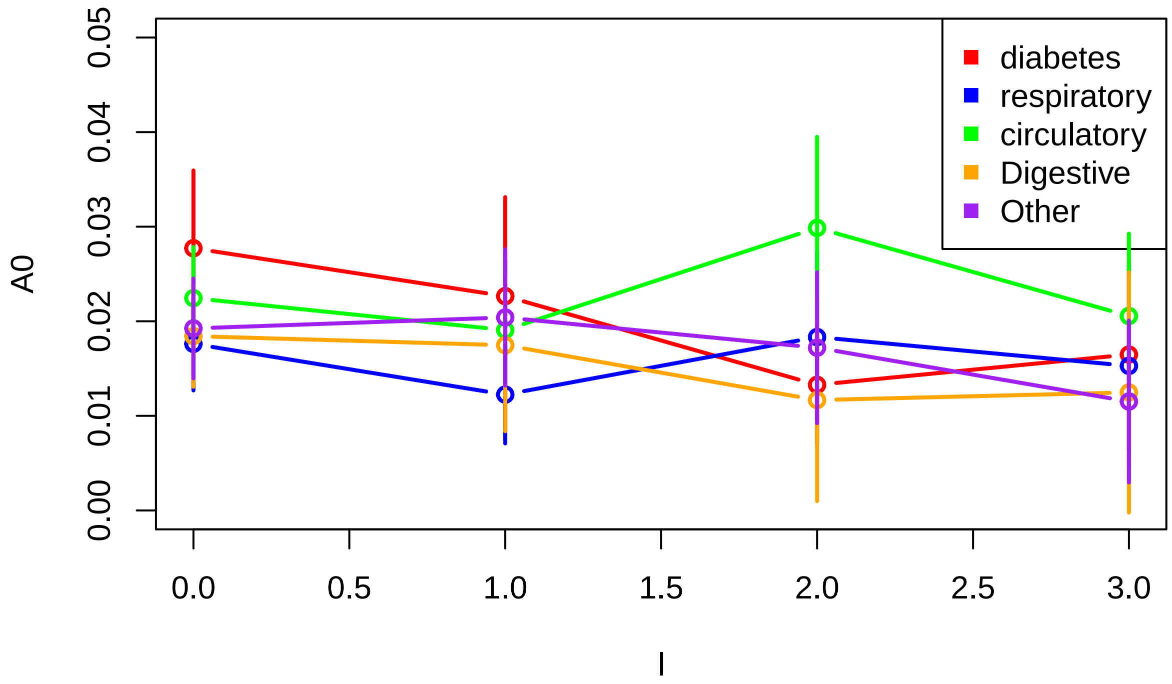
```
time_hos = 4.3
Diag_list = list("Diabetes","Respiratory","Circulatory","Digestive","Others")
Diag_list2 = list()
Readmit_list = list("None","Norm","High&Ch","High&Not")

TEST = list(list(),list(),list(),list(),list())
TEST2 = list(list(),list(),list(),list(),list())
for (i in 1:5) {
  for (j in 1:4){
    temp = data.frame( "Medical_speciality" = "Cardiology" , "Age_" = as.factor("[30, 60)"), "race_" =
    TEST[[i]][[j]] = temp
    colnames(TEST[[i]][[j]]) <- c("Medical_speciality","Age_","race_","Admission","Discharge","Reaction
  }
}

critval <- 1.96 ## approx 95% CI
plot_data = list(c(),c(),c(),c(),c())
upr_data = list(c(),c(),c(),c(),c())
lwr_data = list(c(),c(),c(),c(),c())
for (i in 1:5 ){
  for (j in 1:4 ){
    temp = predict(linModel_core_num, TEST[[i]][[j]], type="response",se.fit = TRUE)
    plot_data[[i]][j] = temp$fit
    upr_data[[i]][j] = temp$fit + (critval * temp$se.fit)
    lwr_data[[i]][j] = temp$fit - (critval * temp$se.fit)
  }
}
```

plot graph

```
A0 = plot_data[[1]]
B0 = plot_data[[2]]
C0 = plot_data[[3]]
D0 = plot_data[[4]]
E0 = plot_data[[5]]
l = c(0,1,2,3)
plot(l,A0, type="b", ylim = c(0, 0.05),col="red",lwd=2.5)
lines(l,B0,type = "b",col="blue",lwd=2.5)
lines(l,C0,type = "b",col="green",lwd=2.5)
legend("topright", c("diabetes", "respiratory", "circulatory"), col = c("red", "blue", "green"), pch =
segments(x0=l, y0=lwr_data[[1]], y1= upr_data[[1]], col="red",lwd=2.5,lend=0)
segments(x0=l, y0=lwr_data[[2]], y1=upr_data[[2]],col="blue",lwd=2.5,lend=1)
segments(x0=l, y0=lwr_data[[3]], y1=upr_data[[3]],col="green",lwd=2.5,lend=2)
```



```
plot( l,A0, type="b", ylim = c(0, 0.05),col="red",lwd=2)
lines(l,B0,type = "b",col="blue",lwd=2)
lines(l,C0,type = "b",col="green",lwd=2)
lines(l,D0,type = "b",col="orange",lwd=2)
lines(l,E0,type = "b",col="purple",lwd=2)
legend("topright", c("diabetes", "respiratory", "circulatory","Digestive","Other"), col = c("red", "blu
segments(x0=l, y0=lwr_data[[1]], y1=upr_data[[1]],col="red",lwd=2)
segments(x0=l, y0=lwr_data[[2]], y1=upr_data[[2]],col="blue",lwd=2)
segments(x0=l, y0=lwr_data[[3]], y1=upr_data[[3]],col="green",lwd=2)
segments(x0=l, y0=lwr_data[[4]], y1=upr_data[[4]],col="orange",lwd=2)
segments(x0=l, y0=lwr_data[[5]], y1=upr_data[[5]],col="purple",lwd=2)
```

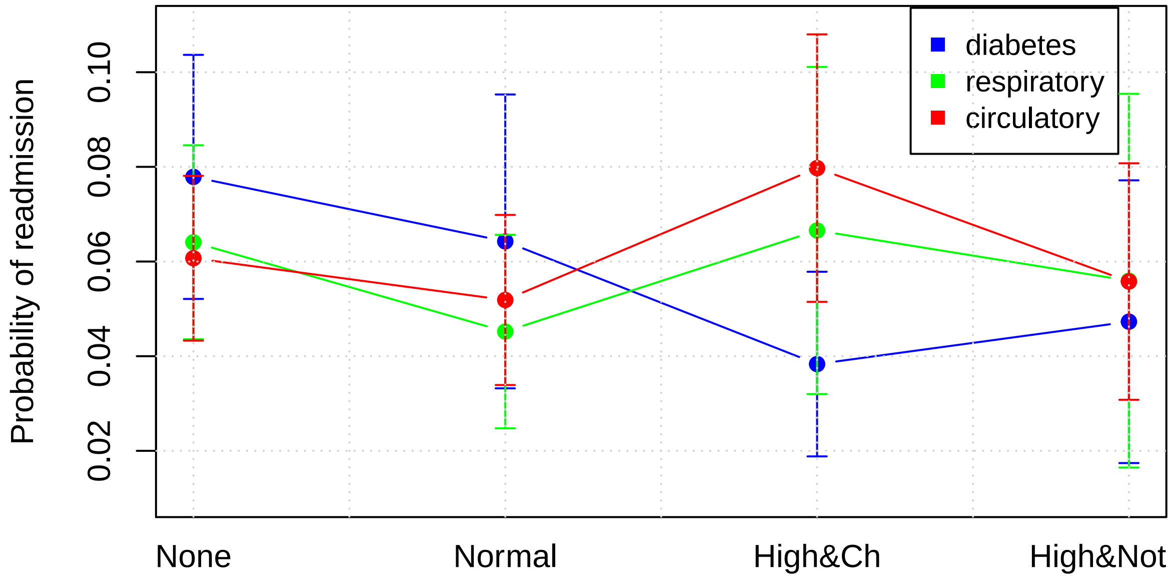


age: 30-60 -> 60-100 Home -> Other change in probability

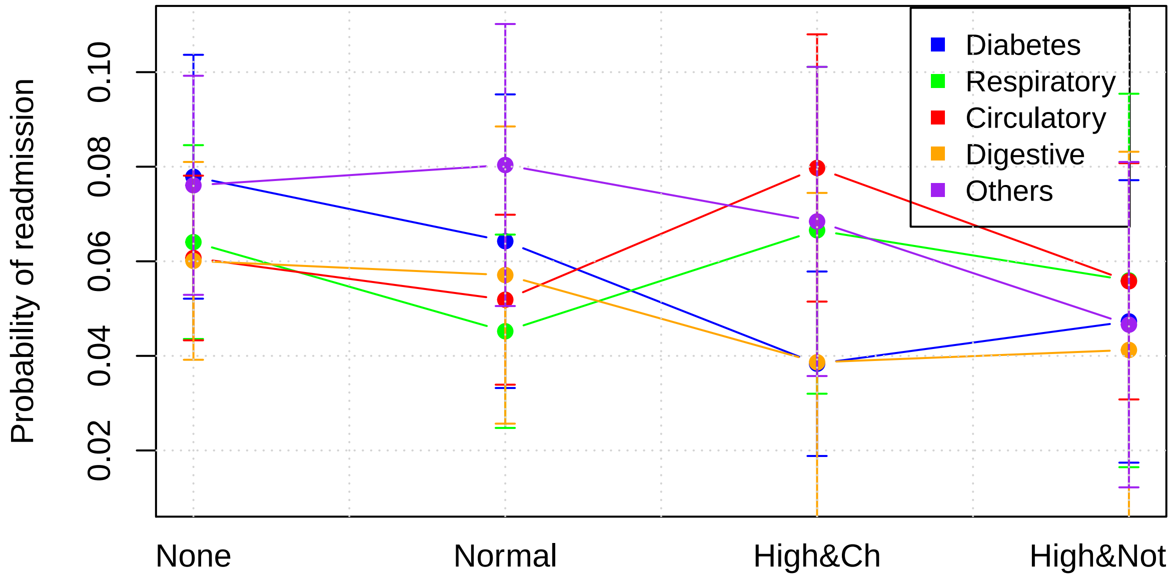
```
time_hos = 4.3
Diag_list = list("Diabetes","Respiratory","Circulatory","Digestive","Others")
Readmit_list = list("None","Norm","High&Ch","High&Not")

TEST2 = list(list(),list(),list(),list(),list())
for (i in 1:5) {
  for (j in 1:4){
    temp = data.frame( "Medical_speciality" = "Cardiology" , "Age_" = as.factor("[60, 100)"), "race_" =
    TEST2[[i]][[j]] = temp
    colnames(TEST2[[i]][[j]]) <- c("Medical_speciality","Age_","race_","Admission","Discharge","Reactio
  }
}

critval <- 1.96 ## approx 95% CI
plot_data2 = list(c(),c(),c(),c(),c())
```

```
upr_data2 = list(c(),c(),c(),c(),c())
lwr_data2 = list(c(),c(),c(),c(),c())
for (i in 1:5 ){
  for (j in 1:4 ){
    temp = predict(linModel_core_num, TEST2[[i]][[j]], type="response",se.fit = TRUE)
    plot_data2[[i]][j] = temp$fit
    upr_data2[[i]][j] = temp$fit + (critval * temp$se.fit)
    lwr_data2[[i]][j] = temp$fit - (critval * temp$se.fit)
  }
}
```

```
A02 = plot_data2[[1]]
B02 = plot_data2[[2]]
C02 = plot_data2[[3]]
D02 = plot_data2[[4]]
E02 = plot_data2[[5]]
lr = 1

plot( l,A02,ann=F , type="b", ylim = c(0.01, 0.11),col="blue",lwd=lr,pch=20,cex=1.5, xaxt="n")
title(ylab = 'Probability of readmission')
mtext("None",side=1,at=0,line=0.5)
mtext("Normal",side=1,at=1,line=0.5)
mtext("High&Ch",side=1,at=2,line=0.5)
mtext("High&Not",side=1,at=2.9,line=0.5)
lines(l,B02,type = "b",col="green",lwd=lr,pch=20,cex=1.5)
lines(l,C02,type = "b",col="red",lwd=lr,pch=20,cex=1.5)
legend(2.3,0.1136, c("diabetes", "respiratory", "circulatory"), col = c("blue", "green", "red"), pch = c
arrows(l,lwr_data2[[1]],l,upr_data2[[1]],code=3,col="blue",lwd=lr,length= 0.05,angle=90)
arrows(l,lwr_data2[[2]],l,upr_data2[[2]],code=3,col="green",lwd=lr,length= 0.05,angle=90)
arrows(l,lwr_data2[[3]],l,upr_data2[[3]],code=3,col="red",lwd=lr,length= 0.05,angle=90)
grid()
```



```
A02 = plot_data2[[1]]
B02 = plot_data2[[2]]
C02 = plot_data2[[3]]
D02 = plot_data2[[4]]
E02 = plot_data2[[5]]
lr = 1

plot( l,A02,ann=F , type="b", ylim = c(0.01, 0.11),col="blue",lwd=lr,pch=20,cex=1.5, xaxt="n")
title(ylab = 'Probability of readmission')
mtext("None",side=1,at=0,line=0.5)
mtext("Normal",side=1,at=1,line=0.5)
mtext("High&Ch",side=1,at=2,line=0.5)
mtext("High&Not",side=1,at=2.9,line=0.5)
lines(l,B02,type = "b",col="green",lwd=lr,pch=20,cex=1.5)
lines(l,C02,type = "b",col="red",lwd=lr,pch=20,cex=1.5)
lines(l,D02,type = "b",col="orange",lwd=lr,pch=20,cex=1.5)
lines(l,E02,type = "b",col="purple",lwd=lr,pch=20,cex=1.5)

legend(2.3,0.1136, c("Diabetes","Respiratory","Circulatory","Digestive","Others"), col = c("blue", "gree
arrows(l,lwr_data2[[1]],l,upr_data2[[1]],code=3,col="blue",lwd=lr,length= 0.05,angle=90)
arrows(l,lwr_data2[[2]],l,upr_data2[[2]],code=3,col="green",lwd=lr,length= 0.05,angle=90)
arrows(l,lwr_data2[[3]],l,upr_data2[[3]],code=3,col="red",lwd=lr,length= 0.05,angle=90)
arrows(l,lwr_data2[[4]],l,upr_data2[[4]],code=3,col="orange",lwd=lr,length= 0.05,angle=90)
arrows(l,lwr_data2[[5]],l,upr_data2[[5]],code=3,col="purple",lwd=lr,length= 0.05,angle=90)

grid()
```



Test and Other Methods

Comment: This data set is a super-unbalanced one, therefore accuracy of all these methods are meaningless. However, good to learn!

Split data_full into train and test set

```
data_full$Diag1 = as.factor(data_full$Diag1)
data_full$Diag2 = as.factor(data_full$Diag2)
data_full$Diag3 = as.factor(data_full$Diag3)
data_full$HbA1c = as.factor(data_full$HbA1c)

data_full = data2
data_full$Readmit = as.factor(data_full$Readmit)
colnames(data_full)
```

```
## [1] "Medical_speciality" "Age_"               "race_"
## [4] "Admission"          "Discharge"          "Reaction"
## [7] "Pri_diag"           "Readmit"            "time_in_hospital"
```

```
set.seed(17)
inTrain <- createDataPartition(y = data_full$Readmit, p = .60,list = FALSE)
train <- data_full[inTrain,]
test <- data_full[-inTrain,]
nrow(train)
```

```
## [1] 41995
```

```
nrow(test)
```

```
## [1] 27995
```

Test of logistic model

```
test$pred_readmit <- predict(linModel_core_num, test, type="response")
# test$pred_readmit = as.numeric(test$pred_readmit)
# test$pred_readmit[test$pred_readmit>=0.5] <- 1
# test$pred_readmit[test$pred_readmit<0.5] <- 0
# test$pred_readmit = as.factor(as.integer(test$pred_readmit))
summary(test$pred_readmit)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 3.080e-06 3.093e-02 4.696e-02 4.949e-02 6.439e-02 1.922e-01
```

```
# confusionMatrix(test$pred_readmit, test$Readmit)

#Accuracy : 0.9505

#     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#1.600e-07 3.117e-02 4.695e-02 4.949e-02 6.434e-02 1.900e-01
```

Random Forest

```
#
#Rf_fit<-randomForest(formula=Readmit ~ Medical_speciality+Age_+race_+Admission+Discharge+Reaction+Pri_
# Rf_fit<-randomForest(formula=Readmit ~ .,
#                       data=train, ntree=500)
#
# print(Rf_fit)
#
# test$pred_readmit <- predict(Rf_fit, test, type = "response")
# table(test$Readmit, test$pred_readmit)
# prop.table(table(test$Readmit, test$pred_readmit),1)

# #importance(Rf_fit)
```

Naive Bayes

```
# # build decision tree with naive Bayes in the leaves
# nbModel <- CoreModel("Readmit", train, model="tree", modelType=4)
# #print(nbModel)
#
# # prediction on test set
# pred <- predict(nbModel, test, type="class")
# # mEval <- modelEval(nbModel, test$Readmit, pred)
# # print(mEval) # evaluation of the model
#
# test$pred_readmit <- pred
# prop.table(table(test$Readmit, test$pred_readmit),1)
# confusionMatrix(test$pred_readmit, test$Readmit)

#Accuracy : 0.949
```

SVM

```
# SVMmodel <- svm(Readmit~ .,
#                  data=train, kernel = "linear")
# print(SVMmodel)
# summary(SVMmodel)
# x <- select(test, -Readmit)
# y <- select(test, Readmit)
# pred <- predict(SVMmodel, x)
# test$pred_readmit <- pred
# prop.table(table(test$Readmit, test$pred_readmit),1)
# confusionMatrix(test$pred_readmit, test$Readmit)

# Accuracy : 0.9505
```

Neural Networks

```
# nnet_model <- nnet(formula = Readmit ~ ., data=train, size = 10, maxit = 100)
#
# test$pred_readmit <- predict(nnet_model, test, type = "class")
# test$pred_readmit = as.factor(test$pred_readmit)
# prop.table(table(test$Readmit, test$pred_readmit),1)
#
# #summary(nnet_model)
#
# confusionMatrix(test$pred_readmit, test$Readmit)
# Accuracy : 0.9499
```

Rpart Tree

```
# rpart_tree <- rpart(formula = Readmit ~ .,
#                     data=train, method = 'class')
# summary(rpart_tree)
#
# test$pred_readmit <- predict(rpart_tree, test, type="class")
# table(predict(rpart_tree, test, type="class"), test$Readmit)
# prop.table(table(test$Readmit, test$pred_readmit),1)
#
# confusionMatrix(test$pred_readmit, test$Readmit)

# Accuracy : 0.9505
```

KNN

```
# KNN
# knnModel <- CoreModel(Readmit ~ .,
#                       data=train, model="knn", kInNN = 5)
# print(knnModel)
#
# pred <- predict(knnModel, test, type="class")
# mEval <- modelEval(knnModel, test$Readmit, pred)
# print(mEval) # evaluation of the model
#
# test$pred_readmit <- pred
# prop.table(table(test$Readmit, test$pred_readmit),1)
# confusionMatrix(test$pred_readmit, test$Readmit)

#Accuracy : 0.9474
```

Referrence:

[1] Beata Strack,1 Jonathan P. DeShazo Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records, 2014. Print. https://www.hindawi.com/journals/bmri/2014/781670/ [2] Diabetes 130-US hospitals for years 1999-2008 Data Set https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008# [3] https://github.com/tobiolatunji/Readmission_Prediction [4] https://github.com/swengzju/Predicting-Diabetes-Patient-Readmission [5] http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html [6] https://www.r-bloggers.com/evaluating-logistic-regression-models/