1. Logistic regression on college students data

The college students data had 108 rows out of which some variables had about 21 missing values. In statistics, it is not advisable to keep variables which have more than 15% missing values.

I started out by viewing the summary of data and then counting the missing values. I then found that only 2 columns had 21 missing values and they occurred together. Moreover, 2 missing values in other variables were also lumped with these 21 rows. So I deleted the two columns and dropped the 21 missing rows

Next I created plots of all the features and found some outliers. SAT score had outliers above 1100, F17.GPA had outliers below 1 and number of credits earned had outliers below 10. I deleted these rows as well.

Then I have divided the data into 80-20 train test split and fitted the first model. The first model showed coefficients of two variables as NA. So I fit another model with these 2 variables removed

The accuracy of this second model is 10/13 = 76.92% which is fairly good. We can then extract the important features in the model using varimp function of the caret package. The important features are-

1. SAT Score
2. Federal Ethnic Group
3. Pell Grant Eligible
4. Attended Orientation
5. Resident Commuter
6. Athlete
7. Receptivity to social engagement
8. Receptivity to career guidance
9. Desire to transfer
10. Peer Mentor
11. Completed community service

CODE:

```
library(dplyr)
library(caret)
library(ggplot2)
college_data = read.csv("232408682483120_File.csv")

#Have a quick view of the data
summary(college_data) #Data has up to 21 NA values per feature

length(which(is.na(college_data))) #Total missing values = 230
#Check character variables
length(which(is.na(college_data$Federal.Ethnic.Group))) #0
```

```
length(which(is.na(college_data$Gender))) #0
length(which(is.na(college_data$Peer.Mentor))) #0
length(which(is.na(college_data$Reason.not.Retained))) #0
length(which(is.na(college_data$Reason.for.not.Completing.Connect))) #0

#Check the rows where 21 values are missing in S18.GPA or CUM.GPA
View(as.matrix(college_data[is.na(college_data$S18.GPA),]))
#Observation 1: S18.GPA and CUM.GPA are missing for the same rows
#Observation 2: Most other missing values are also in the same rows

#Conclusion - Drop S18.GPA and CUM.GPA
#Also remove rows where S18.GPA or CUM.GPA were missing
college_data = college_data %>% filter(!is.na(S18.GPA))
college_data = college_data %>% select(-S18.GPA)
college_data = college_data %>% select(-CUM.GPA)

#Let's plot data
ggplot(data=college_data, aes(x=High.School.GPA)) + geom_bar()
ggplot(data=college_data, aes(x=SAT.Score)) + geom_bar() #values above 1100
are outliers
ggplot(data=college_data,
aes(x=Dropout.Proneness..percentile.score.before.start.of.semester.)) +
geom_bar()
ggplot(data=college_data,
aes(x=Predicted.Academic.Difficulty..percentile.score.before.start.of.semester
.)) + geom_bar()
ggplot(data=college_data,
aes(x=Educational.Stress..percentile.score.before.start.of.semester.)) +
geom_bar()
ggplot(data=college_data,
aes(x=Receptivity.to.Institutional.Help..percentile.score.before.start.of.seme
ster.)) + geom_bar()
ggplot(data=college_data,
aes(x=Receptivity.to.Personal.Counseling..percentile.score.before.start.of.sem
ester.)) + geom_bar()
ggplot(data=college_data,
aes(x=Receptivity.to.Academic.Assistance..percentile.score.before.start.of.sem
ester.)) + geom_bar()
ggplot(data=college_data,
aes(x=Receptivity.to.Social.Engagement..percentile.score.before.start.of.semes
ter.)) + geom_bar()
ggplot(data=college_data,
aes(x=Receptivity.to.Career.Guidance...percentile.score.before.start.of.semest
er.)) + geom_bar()
ggplot(data=college_data,
aes(x=Receptivity.to.Financial.Guidance..percentile.score.before.start.of.seme
ster.)) + geom_bar()
```

```r
ggplot(data=college_data,
aes(x=Desire.to.Transfer..percentile.score.before.start.of.semester.)) +
geom_bar()
ggplot(data=college_data, aes(x=Peer.Mentor)) + geom_bar()
ggplot(data=college_data, aes(x=Number.of.Faculty.Advisor.Meetings.Attended))
+ geom_bar()
ggplot(data=college_data, aes(x=Number.of.Peer.Mentor.Meetings.Attended)) +
geom_bar()
ggplot(data=college_data, aes(x=Number.of.Workshops.Attended)) + geom_bar()
ggplot(data=college_data, aes(x=F17.GPA)) + geom_bar() #Less than 1 are
outliers
ggplot(data=college_data, aes(x=Number.of.Credits.Earned)) + geom_bar() #Less
than 10 are outliers
ggplot(data=college_data, aes(x=Reason.for.not.Completing.Connect)) +
geom_bar()
ggplot(data=college_data, aes(x=Reason.not.Retained)) + geom_bar()

#Remove outliers
college_data = college_data %>% filter(SAT.Score<1100)
college_data = college_data %>% filter(F17.GPA>1)
college_data = college_data %>% filter(Number.of.Credits.Earned>10)

#Time for our first logistic model

#Divide data into train and test
set.seed(100)
train <- college_data %>% sample_frac(0.80)
test  <- anti_join(college_data, train)

College_model = glm(Retained.F17.F18...1.yes..0.no. ~ .,  data =
train,  family = "binomial")
summary(College_model)
#Need to build another model with Reason.for.not.Completing.Connectacademic
dismissal after S18 semester , Reason.not.Retainedacademic dismissal after S18
semester  and Reason.not.Retainedmilitary leave    removed
train = train %>% select(-Reason.for.not.Completing.Connect)
train = train %>% select(-Reason.not.Retained)
College_model = glm(Retained.F17.F18...1.yes..0.no. ~ .,  data =
train,  family = "binomial")
summary(College_model)

test$predictions = predict(College_model, test, type = "response")
test$predictions[test$predictions>0.5] = 1
test$predictions[test$predictions<0.5] = 0
table(test$predictions,test$Retained.F17.F18...1.yes..0.no.) #10/13 correct

#Most important features
varImp(College_model, scale = FALSE)
```

```
#Important features are SAT Score, Federal Ethnic, Pell Grant Eligible,
Attended Orientation, Resident Commuter, Athlete, Receptivity to social
engagement, Receptivity to career guidance, Desire to transfer, Peer Mentor,
Completed community service
```

2. Logistic regression on Health data

The health data had 406 rows with all numerical data. There were only 5 missing values and they were non-overlapping. Since 5 missing rows is about 1% of the data, the rows can be dropped without much change in accuracy

After seeing the summary and counting the missing values, I moved on to outlier detection by plotting graphs of each of the features. There were outliers in ID where it was more than 400 and Ethnicity where it was 5. These rows were also deleted and I am left with 388 rows of data

I then jumped to building 3 models – one where we try to predict if a person needs medical treatment in 2 days or less, the second in which we try to predict if a person needs medical treatment in mean of delaydays or less and the third in which we try to predict if a person needs medical treatment in 1 day or less.

The top important features in the three models are very different from each other. Below comparison table shows the results

Model 1	Model 2	Model 3
Cough	Nausea	Orthopnea
Age	DOE	Dyspnea
DOE	Edema	Cough
Weightgain	Age	Age
Palpitations	Education	Marital
Fatigue	Fatigue	DOE
Education	Weightgain	Palpitations
PND	Livewith	Education
Marital	Marital	Chestpain

If I disregard the order of importance, then I see that Age, DOE, Education and Marital are common features in all the three models. However these are only 4 out of the top 9 important features and hence, the delay in number of days vastly changes the important factors

CODE:

```
library(readxl)
library(dplyr)
library(ggplot2)
```

```r
health_data = read_excel("789968165805283_File.xls")

#Have a quick view of the data
summary(health_data) #2 columns - one has 2 NA and another has 3 NA values

length(which(is.na(health_data))) #Total missing values = 5

#Check the rows where 5 values are missing in delaydays or Livewith
View(as.matrix(health_data[is.na(health_data$delaydays),]))
View(as.matrix(health_data[is.na(health_data$Livewith),]))
#Sadly there is no overlap

#5 missing values out of 406 means about 1% of data missing. Drop rows
health_data = health_data %>% filter(!is.na(delaydays))
health_data = health_data %>% filter(!is.na(Livewith))

#Time to check for outliers
#Let's plot data
ggplot(data=health_data, aes(x=ID)) + geom_bar() #Outlier above 400
ggplot(data=health_data, aes(x=Age)) + geom_bar()
ggplot(data=health_data, aes(x=Gender)) + geom_bar()
ggplot(data=health_data, aes(x=Ethnicity)) + geom_bar() #5 is an outlier
ggplot(data=health_data, aes(x=Marital)) + geom_bar()
ggplot(data=health_data, aes(x=Livewith)) + geom_bar()
ggplot(data=health_data, aes(x=Education)) + geom_bar()
ggplot(data=health_data, aes(x=palpitations)) + geom_bar()
ggplot(data=health_data, aes(x=orthopnea)) + geom_bar()
ggplot(data=health_data, aes(x=chestpain)) + geom_bar()
ggplot(data=health_data, aes(x=nausea)) + geom_bar()
ggplot(data=health_data, aes(x=cough)) + geom_bar()
ggplot(data=health_data, aes(x=fatigue)) + geom_bar()
ggplot(data=health_data, aes(x=dyspnea)) + geom_bar()
ggplot(data=health_data, aes(x=edema)) + geom_bar()
ggplot(data=health_data, aes(x=PND)) + geom_bar()
ggplot(data=health_data, aes(x=tightshoes)) + geom_bar()
ggplot(data=health_data, aes(x=weightgain)) + geom_bar()
ggplot(data=health_data, aes(x=DOE)) + geom_bar()

#Remove outliers
health_data = health_data %>% filter(ID<400)
health_data = health_data %>% filter(Ethnicity!=5)

#Model fitting

#Model 1 for 2 days or less
health_data$medical_treatment = 0
health_data$medical_treatment[health_data$delaydays<=2] = 1
```

```r
set.seed(100)
train <- health_data %>% sample_frac(0.80)
train <- train %>% select(-delaydays)
test  <- anti_join(health_data, train)

health_data_model = glm(medical_treatment ~ .,  data = train,  family =
"binomial")
summary(health_data_model) #ID is important, Age, Palpitations and cough are
important

#Build another model with ID removed
train = train %>% select(-ID)
health_data_model = glm(medical_treatment ~ .,  data = train,  family =
"binomial")
summary(health_data_model) #Age, Palpitations and cough are important

test$predictions = predict(health_data_model, test, type = "response")
test$predictions[test$predictions>0.5] = 1
test$predictions[test$predictions<0.5] = 0
table(test$predictions,test$medical_treatment) #35/72 correct

#Most important features
varImp(health_data_model, scale = FALSE) #Cough, Age, DOE, Weightgain,
Palpitations, Fatigue, Education ,PND, Marital are top important features


#Model 2 for cohort average
health_data$medical_treatment = 0
health_data$medical_treatment[health_data$delaydays<=mean(health_data$delayday
s)] = 1

set.seed(100)
train <- health_data %>% sample_frac(0.80)
train <- train %>% select(-delaydays)
train = train %>% select(-ID)
test  <- anti_join(health_data, train)

health_data_model2 = glm(medical_treatment ~ .,  data = train,  family =
"binomial")
summary(health_data_model2) #Only Nausea is important

test$predictions = predict(health_data_model2, test, type = "response")
test$predictions[test$predictions>0.5] = 1
test$predictions[test$predictions<0.5] = 0
table(test$predictions,test$medical_treatment) #51/68 correct

#Most important features
varImp(health_data_model2, scale = FALSE) #Nausea, DOE, Edema, Age, Education,
Fatigue, Weightgain, Livewith, Marital are top features
```

```r
#Model 3 for 1 days or less
health_data$medical_treatment = 0
health_data$medical_treatment[health_data$delaydays<=1] = 1

set.seed(100)
train <- health_data %>% sample_frac(0.80)
train <- train %>% select(-delaydays)
train = train %>% select(-ID)
test  <- anti_join(health_data, train)

health_data_model3 = glm(medical_treatment ~ .,  data = train,  family =
"binomial")
summary(health_data_model3) #Age, Marital, Orthopnea, cough and dyspnea are
important

test$predictions = predict(health_data_model3, test, type = "response")
test$predictions[test$predictions>0.5] = 1
test$predictions[test$predictions<0.5] = 0
table(test$predictions,test$medical_treatment) #40/68 correct

#Most important features
varImp(health_data_model3, scale = FALSE) #Orthopnea, Dyspnea, Cough, Age,
Marital, DOE, Palpitations, Education, Chestpain are top features
```