Project 2

Manav Mandhani, mm58926

Instructions

Both this completed Rmarkdown document and its knitted HTML, converted to PDF, should be handed in on Canvas **no later than 12:30 pm on March 31st, 2015**. These two documents will be graded jointly, so they must be consistent (as in, don't change the Rmarkdown file without also updating the knitted HTML!).

All results presented **must** have corresponding code. Any answers/results given without the corresponding R code that generated the result will be considered absent. All code reported in your final project document should work properly. Please bear in mind that **you will lose points** for the following:

- printing entire data-frames
- an R-code chunk with no comments
- code that produces error messages
- results without corresponding R code
- extraneous code which does not contribute to the question (if code raises errors or you decide not to use it, you should delete it rather than simply commenting it out!)

For this project, you will work with a dataset collected from Pima Native American women. Studies have shown that Pima women have a much higher incidence of Type II Diabetes than the general population. Since the 1960s, NIH researchers have periodically asked Pima women to undergo various medical tests in order to assess possible diabetes risk factors. Consequently, data on Pima women has proven useful for predicting how likely an individual is to develop diabetes. (Source: J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Symposium on Computer Applications in Medical Care, 261–265.)

```
##### Use these datasets for part 1 (described below) #####
# Dataset to use specifically for training in Part 1
pima_training <- read.csv("http://wilkelab.org/classes/SDS348/2015_spring_projects/project2/pima_training.csv")
# Dataset to use specifically for testing your model in Part 1
pima_test <- read.csv("http://wilkelab.org/classes/SDS348/2015_spring_projects/project2/pima_test.csv")

##### Use this dataset for part 2 (described below) #####
# Complete Pima data, with a single observation per individual
pima_full <- read.csv("http://wilkelab.org/classes/SDS348/2015_spring_projects/project2/pima.csv")

head(pima_full)
```

```
##   npreg glucose dbp skin insulin  bmi pedigree age diabetic
## 1     6     148  72   35       0 33.6    0.627  50      Yes
## 2     1      85  66   29       0 26.6    0.351  31       No
## 3     8     183  64    0       0 23.3    0.672  32      Yes
## 4     1      89  66   23      94 28.1    0.167  21       No
## 5     0     137  40   35     168 43.1    2.288  33      Yes
## 6     5     116  74    0       0 25.6    0.201  30       No
```

The column contents are as follows:

- **npreg**: number of times pregnant
- **glucose**: plasma glucose concentration at 2 hours in an oral glucose tolerance test (units: mg/dL)
- **dpb**: diastolic blood pressure (units: mm Hg)
- **skin**: triceps skin-fold thickness (units: mm)
- **insulin**: 2-hour serum insulin level (units: μU/mL)
- **bmi**: Body Mass Index
- **age**: age in years
- **diabetic**: whether or not the individual has diabetes

Your goal for this project is to analyze the Pima women dataset using several statistical approaches we have learned, in two parts:

Part 1 (60 points). We have divided the dataset, which consists of observations from 768 individuals, into a training and a test data set. Fit a logistic regression model (to predict diabetes incidence) on the training data set. When building your model, use backwards selection to choose predictors which are significant at your chosen significance level (be sure to report your chosen value!). Your code should be appropriately commented with high-level statements about the code's function.

Using your final model, predict the outcome on the test data set, and plot and discuss your results. You should have two final plots: a plot with two ROC curves for the training and test data each, and a plot of the fitted probability of diabetes incidence as a function of the predictors, colored by diabetes, on the test data. Your discussion should, at least, cover the differences and similarities in model performance on the training vs. test data (including AUC) as well as a clear interpretation of each plot. Please limit your discussion to a maximum of 8 sentences.

Part 2 (40 points). Think of two questions to ask about this data set (for this, you are welcome to use either the training, test, or full data set). For each question, perform an exploratory statistical analysis (PCA, k-means, logistic regression, linear model, etc.) with a corresponding figure. Discuss your findings, in particular how your analysis' results reveal the trend of interest. Please limit each question's discussion to a maximum of 5 sentences.

Project responses should be entered below.

```
# This R code chunk contains the calc_ROC function.
calc_ROC <- function(probabilities, known_truth, model.name=NULL)
  {
  outcome <- as.numeric(factor(known_truth))-1
  pos <- sum(outcome) # total known positives
  neg <- sum(1-outcome) # total known negatives
  pos_probs <- outcome*probabilities # probabilities for known positives
  neg_probs <- (1-outcome)*probabilities # probabilities for known negatives
  true_pos <- sapply(probabilities,
                     function(x) sum(pos_probs>=x)/pos) # true pos. rate
  false_pos <- sapply(probabilities,
                      function(x) sum(neg_probs>=x)/neg)
  if (is.null(model.name))
    result <- data.frame(true_pos, false_pos)
  else
    result <- data.frame(true_pos, false_pos, model.name)
  result %>% arrange(false_pos, true_pos)
  }
```

Part 1

```
# Perform first logistical regression using all the parameters
glm.out <- glm(diabetic ~ npreg + glucose + dbp + skin + insulin + bmi + pedigree + age + diabetic, data = pima_training,
               family = binomial)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 9 in
## model.matrix: no columns are assigned
```

```
summary(glm.out)
```

```
##
## Call:
## glm(formula = diabetic ~ npreg + glucose + dbp + skin + insulin +
##     bmi + pedigree + age + diabetic, family = binomial, data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8315  -0.7146  -0.3881   0.6895   2.5156
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.831485   1.022475  -9.615  < 2e-16 ***
## npreg        0.053773   0.041507   1.295  0.19515
## glucose      0.039778   0.004801   8.284  < 2e-16 ***
## dbp         -0.016043   0.007325  -2.190  0.02850 *
## skin        -0.008628   0.008842  -0.976  0.32920
## insulin     -0.001582   0.001128  -1.403  0.16058
## bmi          0.118569   0.021625   5.483 4.18e-08 ***
## pedigree     1.143502   0.391533   2.921  0.00349 **
## age          0.027106   0.011975   2.264  0.02360 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 633.38  on 487  degrees of freedom
## Residual deviance: 442.52  on 479  degrees of freedom
## AIC: 460.52
##
## Number of Fisher Scoring iterations: 5
```

```
# Remove skin because it has the highest p-value
glm.out <- glm(diabetic ~ npreg + glucose + dbp + insulin + bmi + pedigree + age + diabetic, data = pima_training,
               family = binomial)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 8 in
## model.matrix: no columns are assigned
```

```
summary(glm.out)
```

```
##
## Call:
## glm(formula = diabetic ~ npreg + glucose + dbp + insulin + bmi +
##     pedigree + age + diabetic, family = binomial, data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7619  -0.7146  -0.3763   0.6813   2.4806
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.798695   1.019453  -9.612  < 2e-16 ***
## npreg        0.054210   0.041679   1.301  0.19337
## glucose      0.040435   0.004779   8.460  < 2e-16 ***
## dbp         -0.016828   0.007264  -2.317  0.02052 *
## insulin     -0.002031   0.001024  -1.984  0.04722 *
## bmi          0.111954   0.020356   5.500  3.8e-08 ***
## pedigree     1.109725   0.388744   2.855  0.00431 **
## age          0.027961   0.011971   2.336  0.01951 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 633.38  on 487  degrees of freedom
## Residual deviance: 443.47  on 480  degrees of freedom
## AIC: 459.47
##
## Number of Fisher Scoring iterations: 5
```

```
# Remove npreg
glm.out <- glm(diabetic ~ glucose + dbp + insulin + bmi + pedigree + age + diabetic, data = pima_training,
               family = binomial)
```

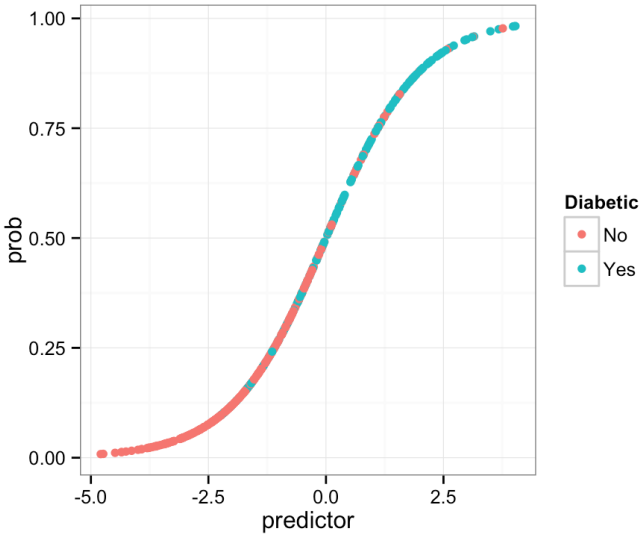
```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 7 in
## model.matrix: no columns are assigned
```

```
summary(glm.out)
```

```
##
## Call:
## glm(formula = diabetic ~ glucose + dbp + insulin + bmi + pedigree +
##     age + diabetic, family = binomial, data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7512  -0.7003  -0.3802   0.6866   2.4830
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.844364   1.018857  -9.662  < 2e-16 ***
## glucose      0.040197   0.004777   8.415  < 2e-16 ***
## dbp         -0.016102   0.007255  -2.219 0.026458 *
## insulin     -0.002113   0.001030  -2.051 0.040313 *
## bmi          0.111613   0.020258   5.510 3.59e-08 ***
## pedigree     1.090654   0.385865   2.827 0.004706 **
## age          0.035913   0.010355   3.468 0.000524 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 633.38  on 487  degrees of freedom
## Residual deviance: 445.17  on 481  degrees of freedom
## AIC: 459.17
##
## Number of Fisher Scoring iterations: 5
```

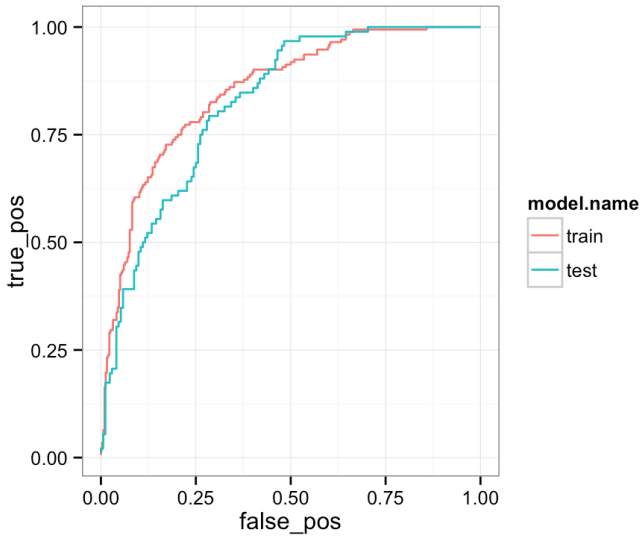
```
# All attributes have p-value of less than 0.1

# Create a data-frame using the fitted values and linear predictors
lr_data <- data.frame(predictor=glm.out$linear.predictors, prob=glm.out$fitted.values, Diabetic=pima_training$diabetic)
ggplot(lr_data, aes(x=predictor, y=prob, color=Diabetic)) + geom_point()
```




```
# Generate ROC data for both the training and test data sets
test_pred <- predict(glm.out, pima_test, type='response')
```

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

```
ROC.train <- calc_ROC(probabilities=glm.out$fitted.values,
                known_truth=pima_training$diabetic,
                model.name="train")
ROC.test <- calc_ROC(probabilities=test_pred,
                known_truth=pima_test$diabetic,
                model.name="test")
# Bind the train and test ROC curves together and then plot
ROCs <- rbind(ROC.train, ROC.test)
ggplot(data=ROCs, aes(x=false_pos, y=true_pos, color=model.name)) +
  geom_line()
```



```
# Calculate AUC values for test and train datasets
ROCs %>% group_by(model.name) %>%
  mutate(delta=false_pos-lag(false_pos)) %>%
  summarize(AUC=sum(delta*true_pos, na.rm=T)) %>%
  arrange(desc(AUC))
```

```
## Source: local data frame [2 x 2]
##
##   model.name       AUC
## 1      train 0.8516522
## 2       test 0.8236223
```

Discussion for part 1 goes here.

For the ROC curves, the training data set seems to perform better than the test data set until the 0.5 false positive rate. Overall, the training data set seems to perform better which is also evident by the Area under curve values. The AUC for train data is 0.8516522 but for the test data is only 0.8236223.

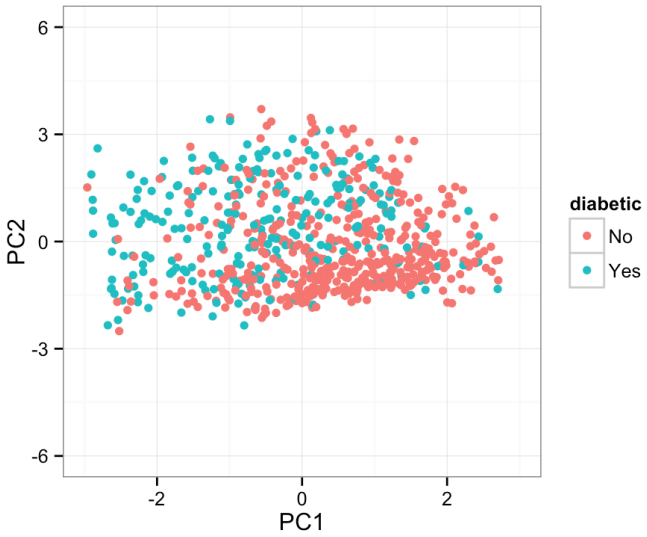
In the logistical curve, the linear predictor is able to quite clearly separate diabetic and non-diabetic women, besides a few exceptions.

Part 2

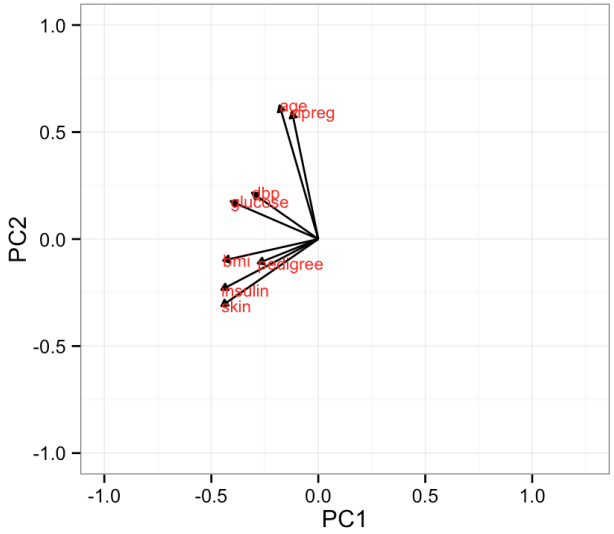
Question 1 goes here.

```
# Perform PCA analysis on pima full dataset
pima_full %>% select(-diabetic) %>%
  scale() %>%
  prcomp() ->
  pca

# Insert PCA data into a dataframe and plot
pima_full.pca <- data.frame(pima_full, pca$x)
ggplot(pima_full.pca, aes(x=PC1, y=PC2, color=diabetic)) + xlim(-3,3) + ylim(-6,6) + geom_point()
```

```
## Warning: Removed 29 rows containing missing values (geom_point).
```



```
# Create rotation matrix
rotation_data <- data.frame(pca$rotation, variable=row.names(pca$rotation))
arrow_style <- arrow(length = unit(0.05, "inches"),
                     type = "closed")
# now plot, using geom_segment() for arrows and geom_text for labels
ggplot(rotation_data) +
  geom_segment(aes(xend=PC1, yend=PC2), x=0, y=0, arrow=arrow_style) +
  geom_text(aes(x=PC1, y=PC2, label=variable), hjust=0, size=3, color='red') +
  xlim(-1.,1.25) +
  ylim(-1.,1.) +
  coord_fixed() # fix aspect ratio to 1:1
```



Discussion for question 1 goes here.

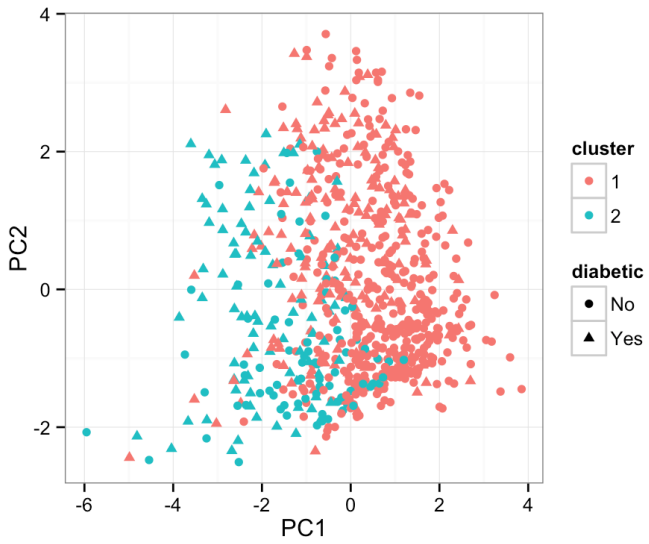
My first question is: Does PCA reveal any important predictors for diabetes incidence?

Age and npreg seem to contribute positively to PC2 whereas all the other attributes contribute negatively to PC1.

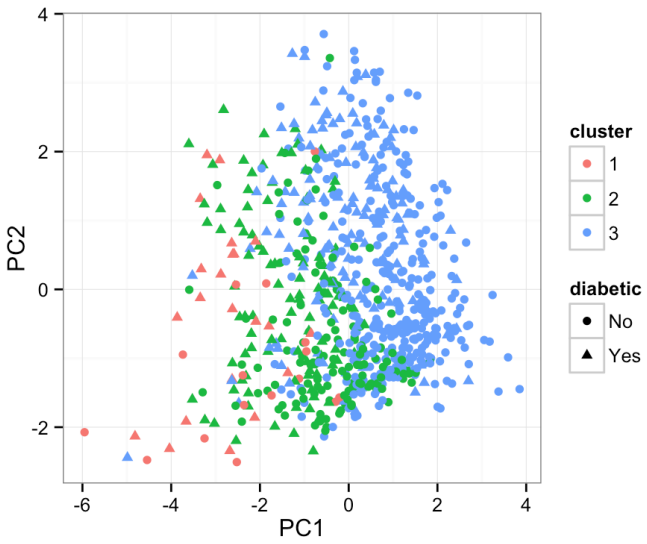
PC1 also seems to slightly measure the difference between diabetic and non-diabetic women. Non-diabetic women seem to score positively while diabetic women seem to score negatively. However, the difference doesn't seem significant enough.

Question 2 goes here.

```
# First, perform a pca analysis on the dataset
pima_full %>% select(-diabetic) %>% scale() %>% prcomp() -> pca

# Next, do a k-means with 2 clusters
pima_full %>% select(-diabetic) %>% # remove Species column
  kmeans(centers=2, nstart=10) ->        # do k-means clustering with 3 centers
  km
pima_clustered <- data.frame(pca$x, cluster=factor(km$cluster), diabetic=pima_full$diabetic)
ggplot(pima_clustered, aes(x=PC1, y=PC2, color=cluster, shape=diabetic)) + geom_point()
```




```
# Performing k-means using 3 clusters
pima_full %>% select(-diabetic) %>% # remove Species column
  kmeans(centers=3, nstart=10) ->        # do k-means clustering with 3 centers
  km
pima_clustered <- data.frame(pca$x, cluster=factor(km$cluster), diabetic=pima_full$diabetic)
ggplot(pima_clustered, aes(x=PC1, y=PC2, color=cluster, shape=diabetic)) + geom_point()
```



Discussion for question 2 goes here. My second question is: Do diabetic and non-diabetic women cluster distinctly in PCA space?

The data was clustered using 2 and 3 different centers. However, there doesn't seem to be any discernable clustering between diabetic and non-diabetic women using the primary principal components.

In the cluster with 2 centers, all the trianglular and circular points (diabetic and non-diabetic respectively) are evenly spread out. There is no clear distinction where all diabetic women belong to a certain cluster and non-diabetic women belong to another one.