Homework 7

Logistic Regression

AUTHOR

Zeb Sorenson

Data and Description

Type 2 diabetes is a problem with the body that causes blood sugar levels to rise higher than normal (hyperglycemia) because the body does not use insulin properly. Specifically, the body cannot make enough insulin to keep blood sugar levels normal. Type 2 diabetes is associated with various health complications such as neuropathy (nerve damage), glaucoma, cataracts and various skin disorders. Early detection of diabetes is crucial to proper treatment so as to alleviate complications.

The data set contains information on 392 randomly selected women who are at risk for diabetes. The data set contains the following variables:

Variable	Description
pregnant	Number of times pregnant
glucose	Plasma glucose concentration at 2 hours in an oral glucose tolerance test
diastolic	Diastolic blood pressure (mm Hg)
triceps	Triceps skin fold thickness (mm)
insulin	2 hour serum insulin (mu U/ml)
bmi	Body mass index (kg/m^2, mass in kilograms divided by height in meters-squared)
pedigree	Numeric strength of diabetes in family line (higher numbers mean stronger history)
age	Age
diabetes	Does the patient have diabetes (0 if “No”, 1 if “Yes”)

The data can be found in the Diabetes data set on Canvas. Download Diabetes.txt, and put it in the same folder as this quarto file.

0. Replace the text "< PUT YOUR NAME HERE >" (above next to "author:") with your full name.

1. Read in the data set, call it "dia", remove the "row" column, and change the class of any categorical variables to a factor. Print a summary of the data and make sure the data makes sense.

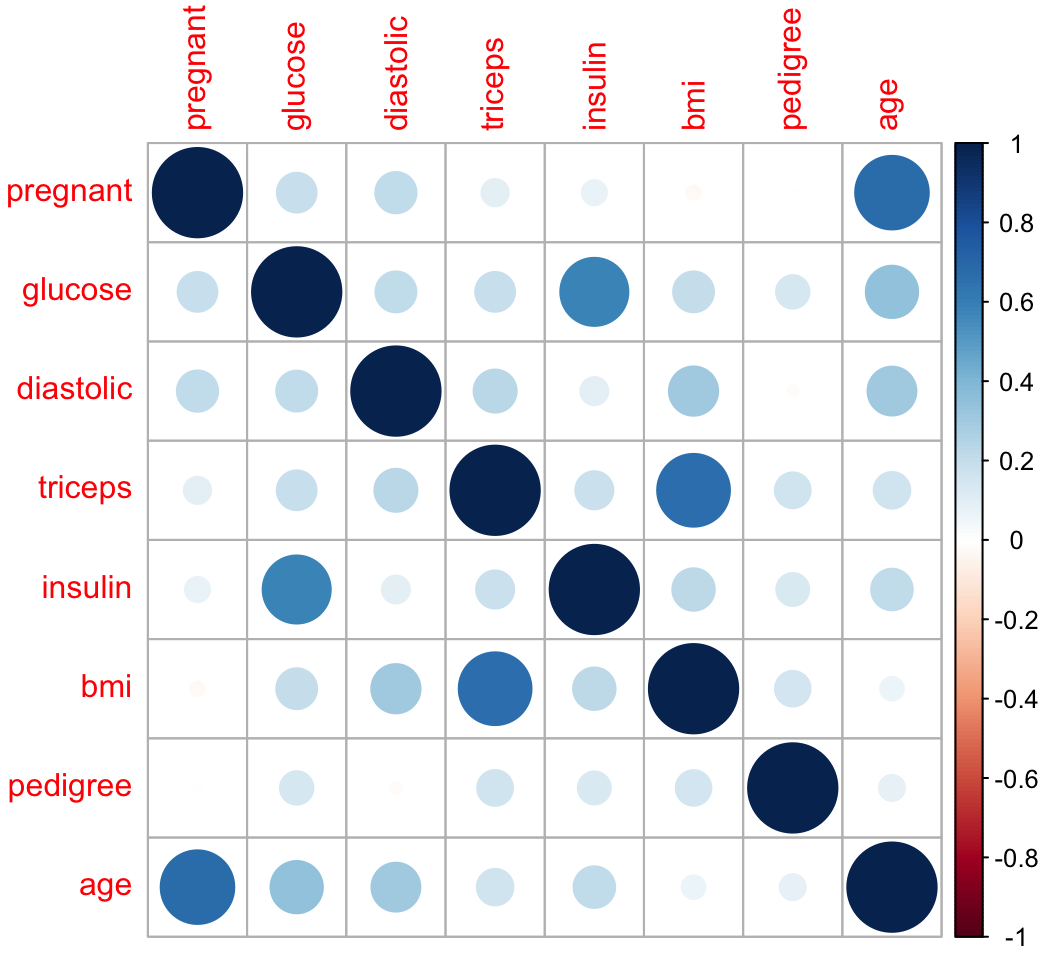
```
dia <- read.csv("~/Desktop/Stat 330/Diabetes.txt", sep="")

dia <- dia[, -which(names(dia) == "row")] #Remove row column

dia$diabetes <- as.factor(dia$diabetes) #Change diabetes to be a factor. Yes or No

summary(dia)
```

```
   pregnant         glucose        diastolic         triceps
Min.   : 0.000   Min.   : 56.0   Min.   : 24.00   Min.   : 7.00
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:21.00
Median : 2.000   Median :119.0   Median : 70.00   Median :29.00
Mean   : 3.301   Mean   :122.6   Mean   : 70.66   Mean   :29.15
3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00   3rd Qu.:37.00
Max.   :17.000   Max.   :198.0   Max.   :110.00   Max.   :63.00
   insulin            bmi           pedigree           age        diabetes
Min.   : 14.00   Min.   :18.20   Min.   :0.0850   Min.   :21.00   0:262
1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697   1st Qu.:23.00   1:130
Median :125.50   Median :33.20   Median :0.4495   Median :27.00
Mean   :156.06   Mean   :33.09   Mean   :0.5230   Mean   :30.86
3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870   3rd Qu.:36.00
Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
```

```
view(dia)
```

2. Explore the data. Create a correlation matrix (or correlation plot) for the covariates. *Comment on why or why not you think multicollinearity may be a problem for this data set.*

```
corrplot(cor(dia[,-9]))
```

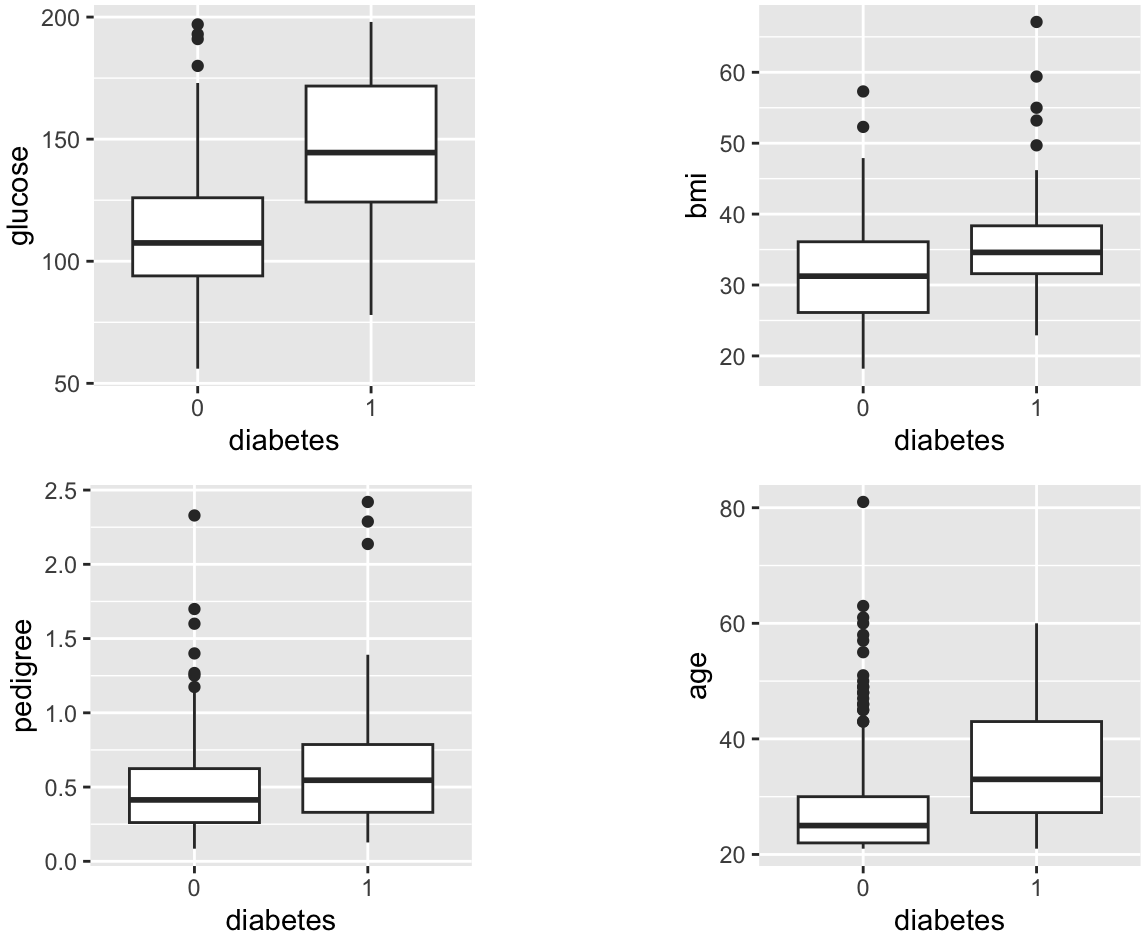


it appears that there may be an issue of multicolinearity with a few of the variables. However, this is not uncommon when dealing with many variables and is something we can address with variables selection methods. Right now, before doing any type of selection, it looks like pregnant & age, along with BMI & Insulin may be causing issues and we will investigate further.

3. Explore the data. Create boxplots of the response against the following predictors: glucose, bmi, pedigree, and age (4 plots in total. You may want to use the grid.arrange function from the gridExtra package to display them in a 2x2 grid). *Briefly comment on one interesting trend you observe.*

```
gluc <- ggplot(data = dia) +
  geom_boxplot(mapping = aes(y = glucose, x = diabetes)) +
  theme(aspect.ratio = 1)

bodymassIndex <- ggplot(data = dia) +
  geom_boxplot(mapping = aes(y = bmi, x = diabetes)) +
  theme(aspect.ratio = 1)

ped <- ggplot(data = dia) +
  geom_boxplot(mapping = aes(y = pedigree, x = diabetes)) +
  theme(aspect.ratio = 1)

age_dia <- ggplot(data = dia) +
  geom_boxplot(mapping = aes(y = age, x = diabetes)) +
  theme(aspect.ratio = 1)
```

```
grid.arrange(gluc, bodymassIndex, ped, age_dia,   nrow = 2)
```

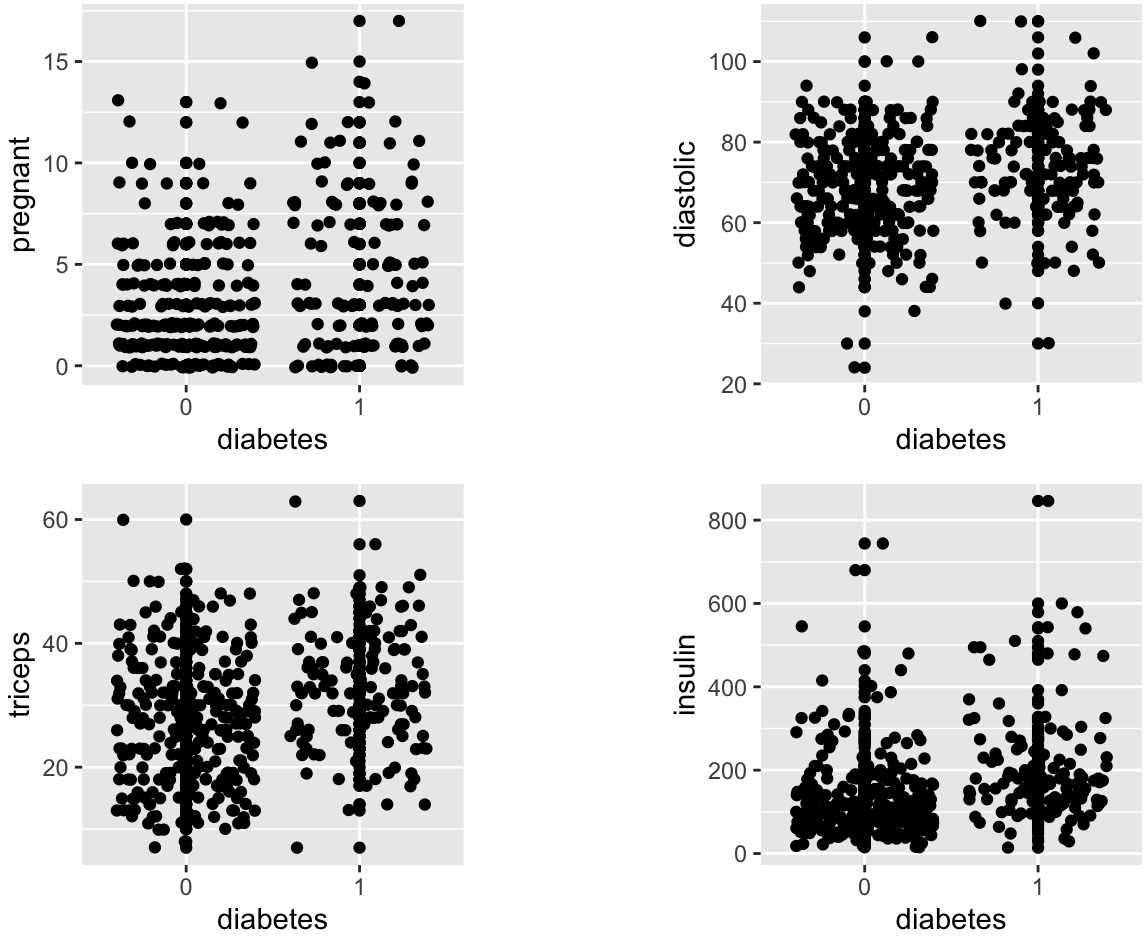


What catches my attention first is that those with and without diabetes have a very similar spread with the pedigree variable and age seems to be playing more of a role than I would have initially anticipated.

4. Explore the data. Create jittered scatterplots of the response against the following predictors: pregnant, diastolic, triceps, insulin (4 plots in total. You may want to use the grid.arrange function from the gridExtra package to display them in a 2x2 grid). *Briefly comment on one interesting trend you observe.*

```
jitter_preg <- ggplot(data = dia) +
  geom_point(mapping = aes(y = pregnant, x = diabetes)) +
  geom_jitter(mapping = aes(y = pregnant, x = diabetes),
              height = 0.1) +
  theme(aspect.ratio = 1)

jitter_diastolic <- ggplot(data = dia) +
  geom_point(mapping = aes(y = diastolic, x = diabetes)) +
  geom_jitter(mapping = aes(y = diastolic, x = diabetes),
              height = 0.1) +
  theme(aspect.ratio = 1)

jitter_triceps <- ggplot(data = dia) +
  geom_point(mapping = aes(y = triceps, x = diabetes)) +
```

```
  geom_jitter(mapping = aes(y = triceps, x = diabetes),
              height = 0.1) +
  theme(aspect.ratio = 1)

jitter_insulin <- ggplot(data = dia) +
  geom_point(mapping = aes(y = insulin, x = diabetes)) +
  geom_jitter(mapping = aes(y = insulin, x = diabetes),
              height = 0.1) +
  theme(aspect.ratio = 1)

grid.arrange(jitter_preg, jitter_diastolic, jitter_triceps, jitter_insulin,   nrow = 2)
```



Giving us a more clear view of the datapoints, pregnancy appears to be less of a predictor that it appeared with the previous plots. diastolic also almost appears identical which is interesting.

5. Briefly explain why traditional multiple linear regression methods are not suitable for *this* data set. (*your reasons should refer to this data set (i.e. be specific, not general)*)

Because our response variables is either the subject has diabetes or does not have diabetes, there is no in between, it wouldn't make sure to use traditional methods. Can we predict specific numeric diabetes levels for someone in between diabetic and not? Maybe, but not with the current knowledge we have in this class. We're not longer dealing with a continuous variable as our response.

Because of this we need to rely on Logistic Regression.

6. Use a variable selection procedure to help you decide which, if any, variables to omit from the logistic regression model you will soon fit. You may choose which selection method to use (best subsets, backward, sequential replacement, LASSO, or elastic net) and which metric/criteria to use (AIC, BIC, or CV/PMSE). *Briefly justify (in a few sentences) why you chose the **method** and **metric** that you did.*

```
diabetes_best_subsets_bic <- bestglm(as.data.frame(dia),
                                     IC = "BIC",
                                     method = "exhaustive",
                                     TopModels = 1,
                                     family = binomial)
```

```
Morgan-Tatar search since family is non-gaussian.
```

```
summary(diabetes_best_subsets_bic$BestModel)
```

```
Call:
glm(formula = y ~ ., family = family, data = Xi, weights = weights)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.092018   1.080251  -9.342  < 2e-16 ***
glucose       0.036189   0.004982   7.264 3.76e-13 ***
bmi           0.074449   0.020267   3.673 0.000239 ***
pedigree      1.087129   0.419408   2.592 0.009541 **
age           0.053012   0.013439   3.945 8.00e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 498.10  on 391  degrees of freedom
Residual deviance: 347.23  on 387  degrees of freedom
AIC: 357.23

Number of Fisher Scoring iterations: 5
```

I decided to use the best subset method because we are not dealing with a high number of co-variates and specified BIC because it will help us to avoid overfitting our model.

7. Write out the logistic regression model for this data set using the covariates that you have chosen. You should use parameters/Greek letters (NOT the "fitted" model using numbers...since you have not fit a model yet).

$$\log(\pi_i/1-\pi_i) = \beta_0 + \beta_1 Glucose_i + \beta_2 BMI_i + \beta_3 Pedigree_i + \beta_4 Age_i$$

Where

$$\log(\pi_i) = Prob(Diabetes_i = 1 | Glucose_i, BMI_i, Pedigree_i, Age_i)$$

And

$$\text{Diabetes}_i \overset{ind}{\sim} \text{Bern}(\pi_i)$$

8. Fit a logistic regression model using the covariates you chose. Print a summary of the results.

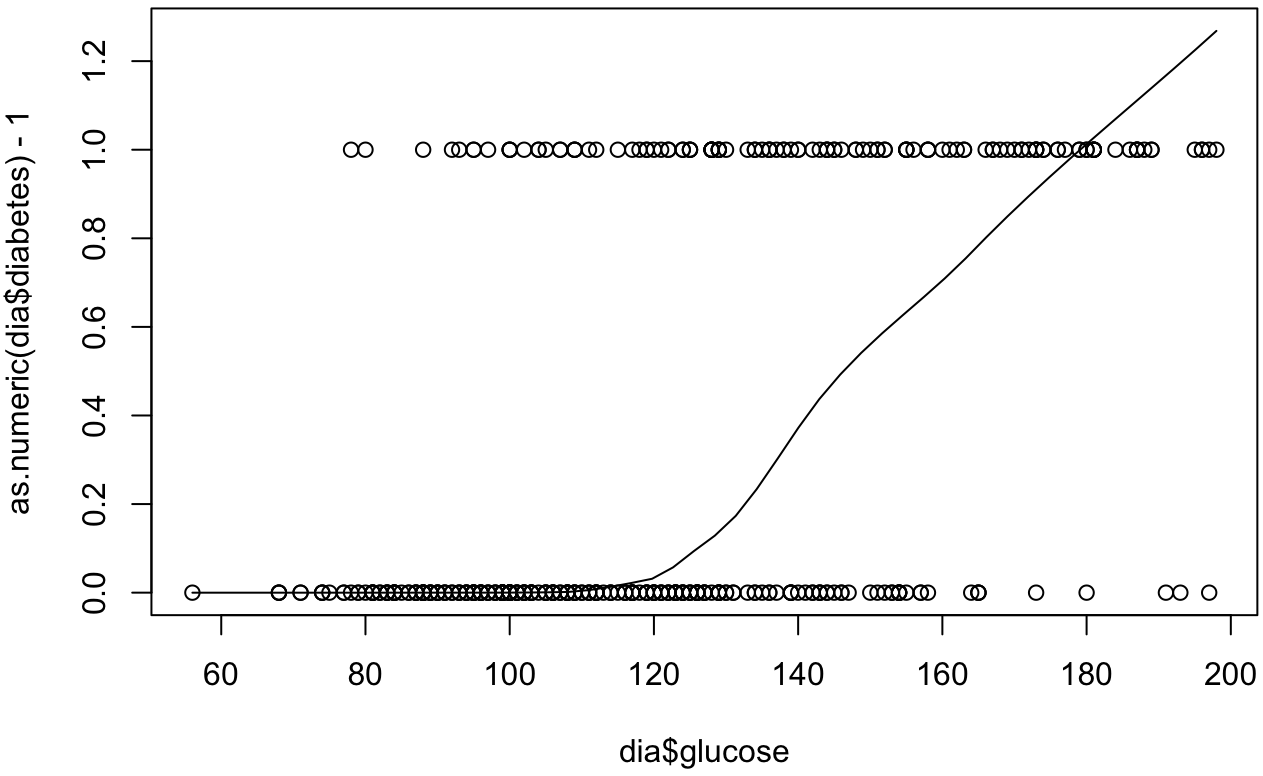
```
dia_logistic <- glm(diabetes ~ glucose + bmi + pedigree + age,
                    data = dia,
                    family = binomial(link = "logit"))
summary(dia_logistic)
```

```
Call:
glm(formula = diabetes ~ glucose + bmi + pedigree + age, family = binomial(link = "logit"),
    data = dia)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.092018   1.080251  -9.342  < 2e-16 ***
glucose       0.036189   0.004982   7.264 3.76e-13 ***
bmi           0.074449   0.020267   3.673 0.000239 ***
pedigree      1.087129   0.419408   2.592 0.009541 **
age           0.053012   0.013439   3.945 8.00e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 498.10  on 391  degrees of freedom
Residual deviance: 347.23  on 387  degrees of freedom
AIC: 357.23

Number of Fisher Scoring iterations: 5
```

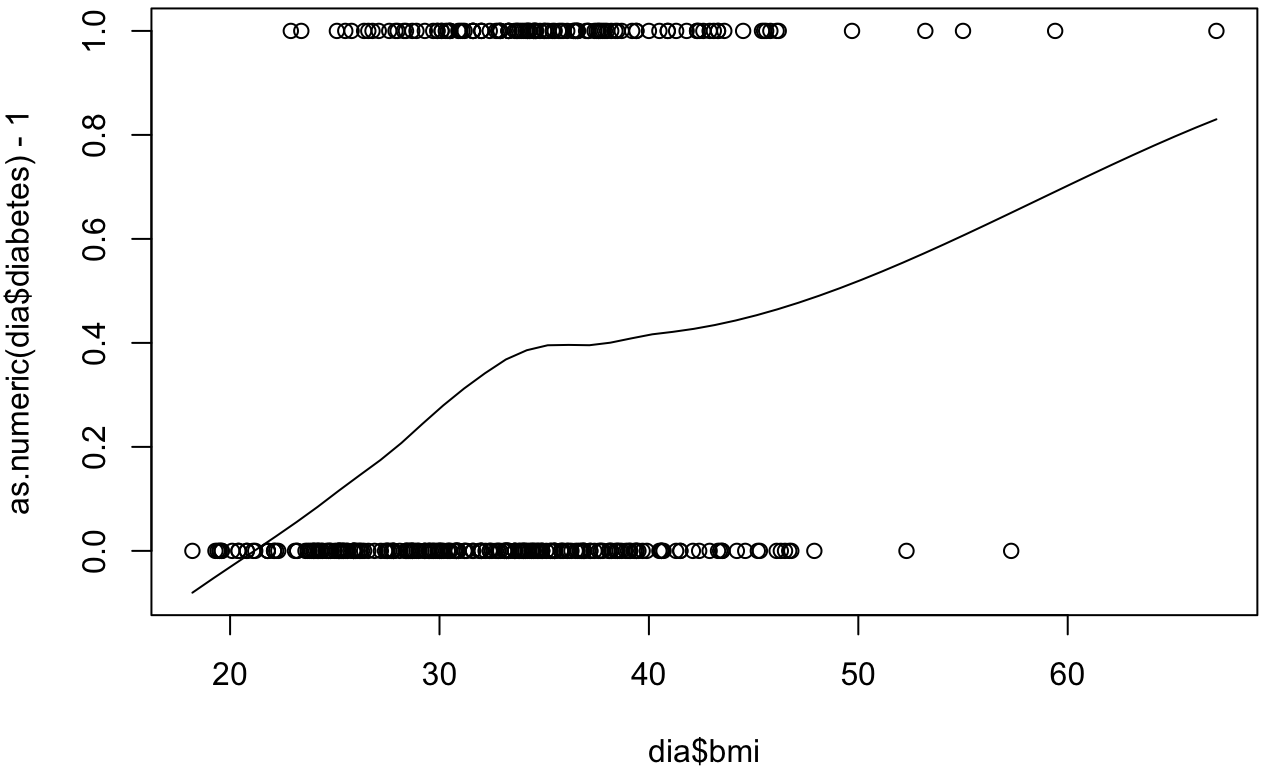
Questions 9-12 involve using diagnostics to check the logistic regression model assumptions. For each assumption, (1) include the relevant code for the diagnostic(s), and (2) explain whether or not you think the assumption is violated and why you think that.

9. The X's vs log odds are linear (monotone in probability) (Use scatterplots with smoothers)

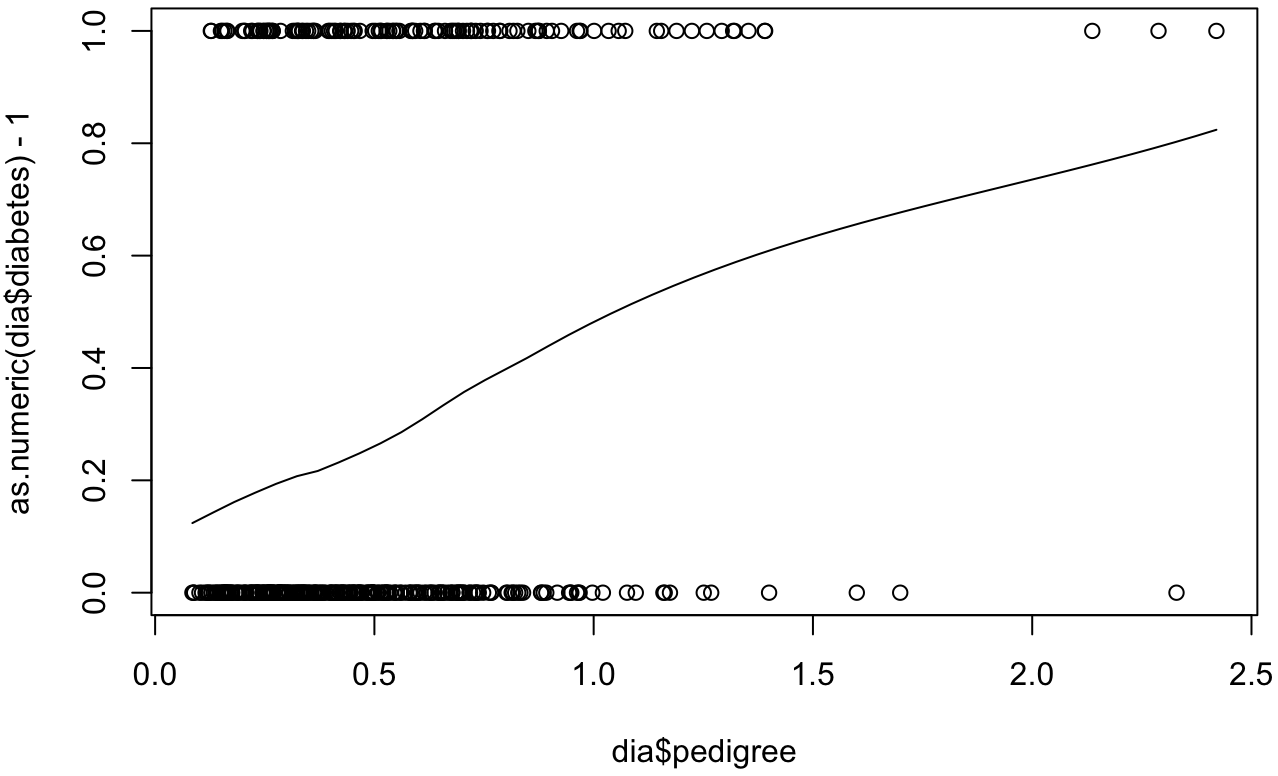
```
scatter.smooth(x = dia$glucose, y = as.numeric(dia$diabetes) - 1)
```



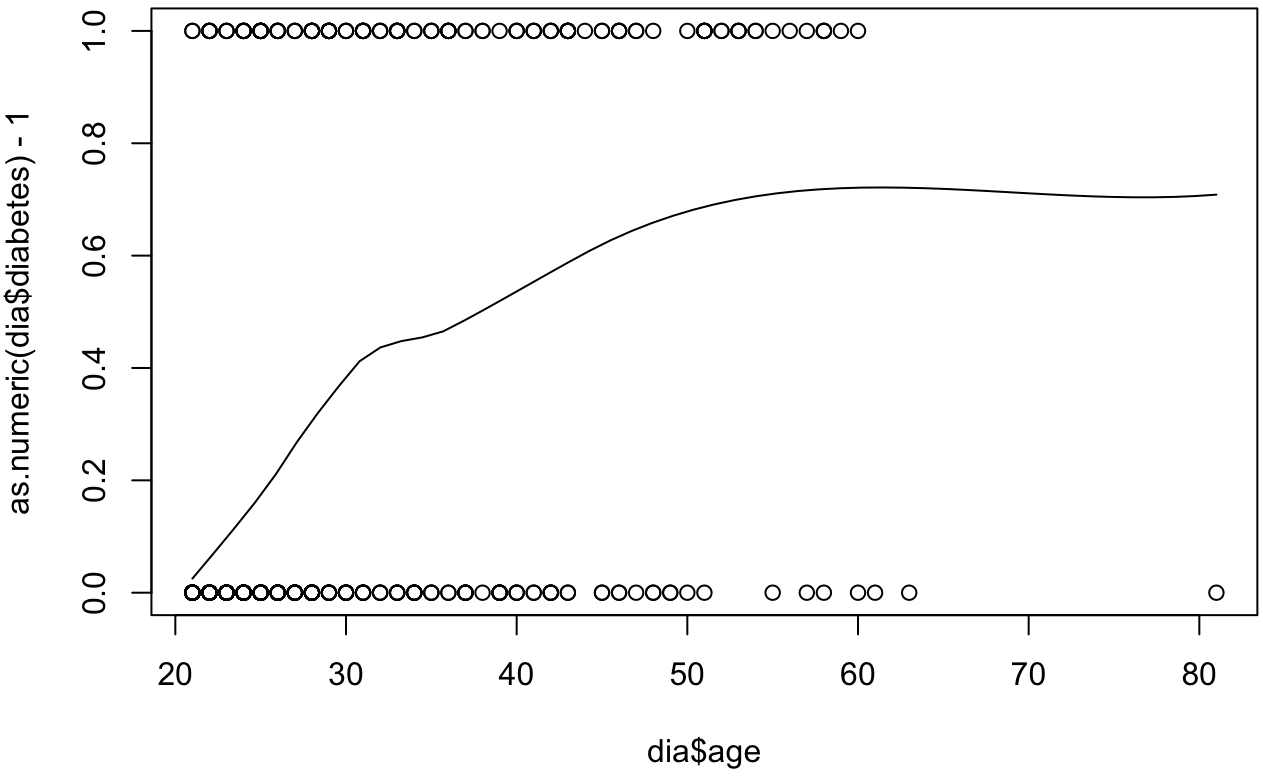

```r
scatter.smooth(x = dia$bmi, y = as.numeric(dia$diabetes) - 1)
```




```
scatter.smooth(x = dia$pedigree, y = as.numeric(dia$diabetes) - 1)
```




```
scatter.smooth(x = dia$age, y = as.numeric(dia$diabetes) - 1)
```



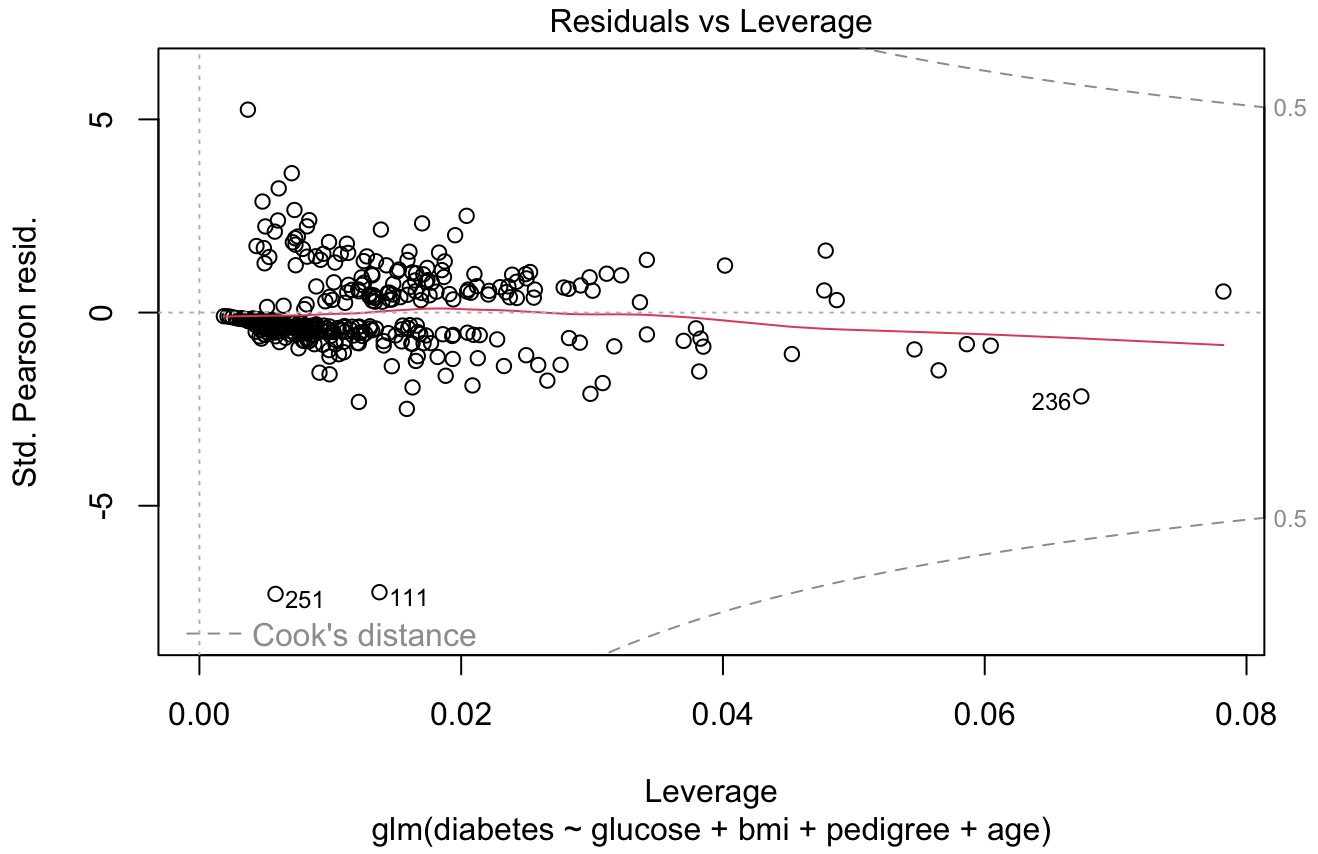
In our final graph inspecting age, we can see our line flattening out around the 80 years old mark. This appears to be caused by a single piece of data. Although this graph is reason for concern and should certainly be investigated and kept in mind during our analysis, I feel confident that the log odds are linear and this assumption is met.

10. The observations are independent (no diagnostic tools or code needed - just think about how the data was collected and briefly write your thoughts)

We are told that the data is collected randomly from 392 randomly selected women. Which is promising, however, we do not know if these women are independent from one another and if one peice of data effects another for certain. Although we are probably okay with this assumption, I will see that it is violated due to the lack of further information.

11. The model describes all observations (i.e., there are no influential points) (Use residuals vs. leverage plot)

```
plot(dia_logistic, which = 5, cook.levels = .5) # Residuals by leverage
```



Looking at the graph we can clearly see that there are no influential points that violates our set bounds of 0.5

This assumption is met!

12. No multicollinearity (Use variance inflation factors)

```
dia_lm <- lm(as.numeric(diabetes) ~glucose + bmi + pedigree + age,
             data = dia)
vif(dia_lm)
```

```
 glucose      bmi pedigree      age 
1.190042 1.065001 1.040289 1.135755 
```

```
max(vif(dia_lm))# < 10
```

```
[1] 1.190042
```

```
mean(vif(dia_lm)) # < 5
```

```
[1] 1.107772
```

All individual VIF's are below 10 and the mean is below 5, I am confident that this assumption is met!

13. Briefly comment on if all assumptions are met. If there is anything you would like to do before proceeding to statistical inference, do that here.

```
#Nothing to do here.
```

As the only concerns are a single data point and lack of information regarding data collection, I am confident that we can proceed with our analysis and acheive a fair understanding of the data with our model. Beyond the two mentioned issues, the assumptions appear to be met and we can proceed.

14. For the coefficient for bmi, compute (and output) the log odds ratio (β_{bmi}, pull this value from the model output), odds ratio ($\exp\{\beta_{bmi}\}$), and the odds ratio converted to a percentage ($100 \times (\exp\{\beta_{bmi}\} - 1)$). (If you cannot view the math used in this question (and subsequent), you can see it by rendering the document.)

```
log_odds_ratio <- coef(dia_logistic)["bmi"]
print("Log odds Ratio")
```

```
[1] "Log odds Ratio"
```

```
print(log_odds_ratio)
```

```
       bmi
0.07444854
```

```
print("Exp Log odds Ratio")
```

```
[1] "Exp Log odds Ratio"
```

```
odds_ratio <- exp(log_odds_ratio)
print(odds_ratio)
```

```
    bmi
1.07729
```

```
print("Percentage")
```

```
[1] "Percentage"
```

```
odds_ratio_percentage <- (100)*(odds_ratio-1)
print(odds_ratio_percentage)
```

```
     bmi
7.728991
```

15. Interpret the coefficient for bmi based on the FOUR different ways we discussed in class.

Interpretation 1: For those of similar glucose levels, pedigree and age, for every one unit increase in BMI, the log odds having diabetes increase by 0.074449

Interpretation 2: For those of similar glucose levels, pedigree and age, as BMI increase by one, the odds of having diabetes is 1.07729 times more likely

Interpretation 3: For those of similar glucose levels, pedigree and age, as BMI increase by one, the odds of having diabetes increases by 7.72%

Directionality Interpretation: Since 0.074449 is greater than 0, the probability of developing diabetes increases as BMI increases, for those of similar glucose levels, pedigree and age.

16. Create (and output) 95% confidence intervals for β_k, $\exp\{\beta_k\}$, and $100 \times (\exp\{\beta_k\} - 1)$ for all predictors using the `confint` function.

```
dia_conf_int_LOR <- confint(dia_logistic)[-1,] # omit intercept row
```

```
Waiting for profiling to be done...
```

```
exp(dia_conf_int_LOR)
```

```
              2.5 %   97.5 %
glucose  1.027127 1.047445
bmi      1.036290 1.122320
pedigree 1.327091 6.871195
age      1.027527 1.083311
```

```
100 * (exp(dia_conf_int_LOR) - 1) # Most straigtforward
```

```
               2.5 %     97.5 %
glucose   2.712692   4.744534
bmi       3.628959  12.231960
pedigree 32.709070 587.119461
age       2.752739   8.331088
```

17. Interpret the 95% confidence interval for $100 \times (\exp\{\beta_{bmi}\} - 1)$.

Interpretation using $100 \times (\exp\{\beta_{bmi}\} - 1)$*:* We are 95% confident that for every additional increase in one unit of BMI, the odds of developing diabetes increase between 3.63% and 12.23%. For those with similar glucose, pedigree and age. (Holding all else constant)

18. Calculate a 95% confidence interval for the predicted probability that a patient has diabetes where pregnant = 1, glucose = 90, diastolic = 62, triceps = 18, insulin = 59, bmi = 25.1, pedigree = 1.268 and age = 25. Note that you may not need to use all of these values depending on the variables you chose to include in your model. *Do you think this patient will develop diabetes? Why or why not?*

```r
new_patient <- data.frame(glucose = 90,
                          bmi = 25.1,
                          pedigree= 1.268 ,
                          age = 25)

log_odds <- predict(dia_logistic,
                    newdata = new_patient,
                    se.fit = TRUE)

# compute the margin of error

moe <- qnorm(p = .975, lower.tail = TRUE) * log_odds$se.fit

# compute the 95% confidence interval (and point estimate) for the log odds

pred_interval <- log_odds$fit + c(-1, 0, 1) * moe

# compute the 95% confidence interval (and point estimate) for the predicted
# probability

exp(pred_interval) / (1 + exp(pred_interval))
```

```
[1] 0.04256397 0.09427071 0.19593629
```

We are 95% confident that the log odds of diabetes for the patient is between 0.04256397 and 0.19593629. Meaning, it's fairly safe to assume that this patient will not develop diabetes.

19. Compute the likelihood ratio test statistic (aka deviance, aka model chi-squared test) for the model, and compute the associated p-value. Print out the test statistic and the p-value. *Based on the results, what do you conclude?*

```r
# Likelihood ratio test statistic
like_ratio <- dia_logistic$null.deviance - dia_logistic$deviance

# Likelihood ratio p-value
pvalLikely <- pchisq(q = like_ratio,
       df = length(coef(dia_logistic)) - 1,
       lower.tail = FALSE)

print(like_ratio)
```

```
[1] 150.8628
```

```r
print(pvalLikely)
```

```
[1] 1.329928e-31
```

These values together suggest that the probability that the test results from our model is correct is extremely high compared to the probability of our model being incorrect. Our model is extremely strong.

20. Compute (and output) the pseudo R^2 value for the model.

```
1 - dia_logistic$deviance/dia_logistic$null.deviance
```

```
[1] 0.3028779
```

21. What is the best cutoff value for the model that minimizes the percent misclassified (or equivalently maximizes accuracy)? Show your code and output the best cutoff value.

```
dia_preds <- predict(dia_logistic,
                     type = "response")

dia_preds
```

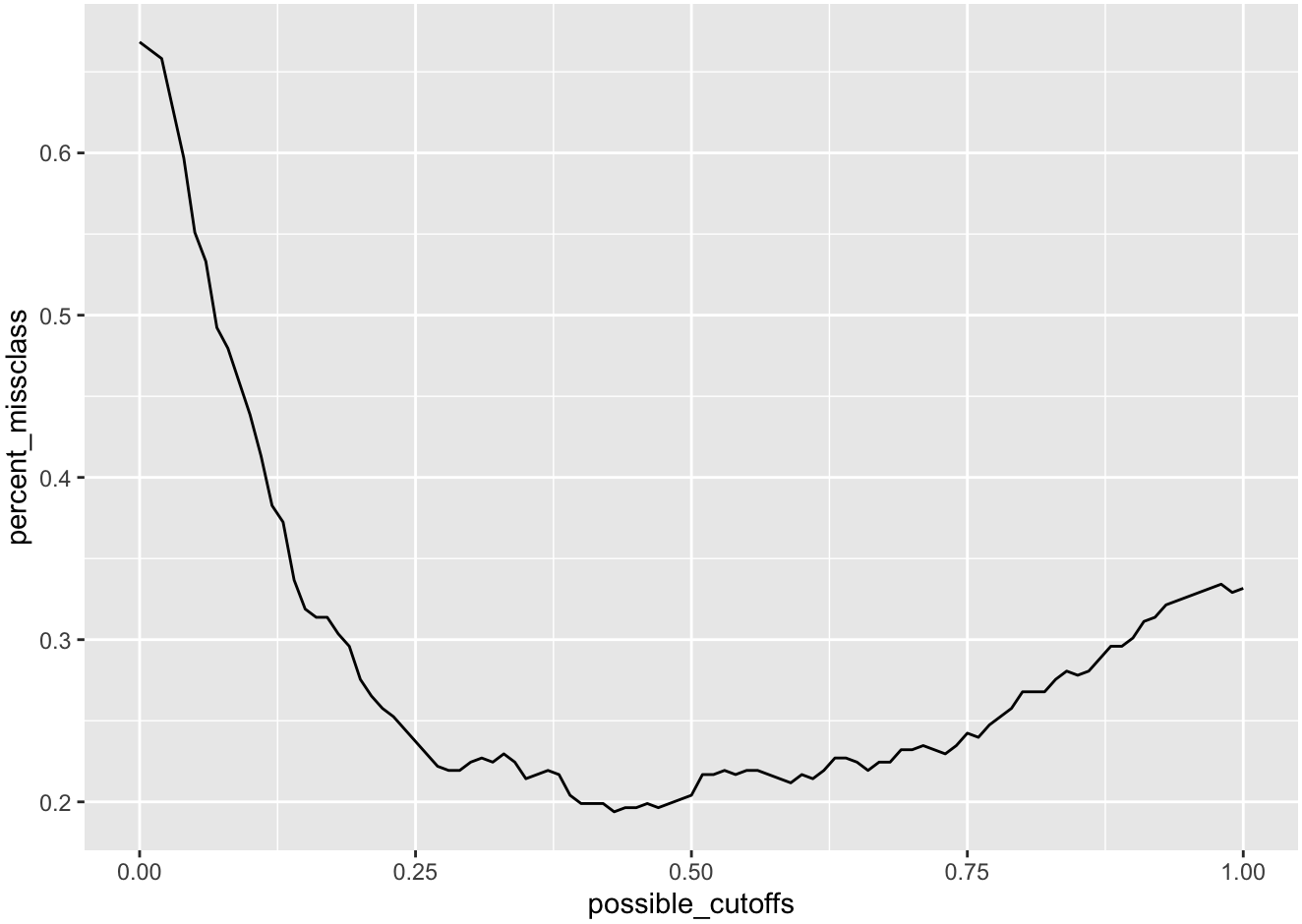
```
          1           2           3           4           5           6
0.029759656 0.909813017 0.035112629 0.907986894 0.927505689 0.764567520
          7           8           9          10          11          12
0.457703266 0.232796457 0.252959940 0.398980834 0.687309300 0.298031611
         13          14          15          16          17          18
0.040746599 0.524025743 0.595777153 0.026479851 0.144522716 0.762563389
         19          20          21          22          23          24
0.651621853 0.957820821 0.038412416 0.063856396 0.041939505 0.914397317
         25          26          27          28          29          30
0.716417436 0.873432840 0.425622111 0.136081906 0.256129228 0.029128696
         31          32          33          34          35          36
0.265013809 0.168417597 0.248356530 0.207495184 0.102962970 0.181093716
         37          38          39          40          41          42
0.132813645 0.509392584 0.274002185 0.236184001 0.236178831 0.510174160
         43          44          45          46          47          48
0.011135252 0.048178088 0.504955327 0.026585817 0.195999323 0.398643739
         49          50          51          52          53          54
0.054983454 0.088759347 0.516010433 0.746000012 0.042273682 0.662756581
         55          56          57          58          59          60
0.031479932 0.867805450 0.113061095 0.290040434 0.383728381 0.137103002
         61          62          63          64          65          66
0.325011241 0.627533567 0.647019692 0.049017187 0.211986536 0.081822767
         67          68          69          70          71          72
0.054932507 0.131491890 0.095172155 0.354326623 0.340534223 0.336286700
         73          74          75          76          77          78
0.835641791 0.606743227 0.053549119 0.104662545 0.034450914 0.903041334
         79          80          81          82          83          84
0.264260975 0.256742005 0.241595814 0.131790081 0.382100638 0.115247535
         85          86          87          88          89          90
0.046748951 0.819114262 0.785324002 0.235083033 0.927465810 0.526978993
         91          92          93          94          95          96
0.150438287 0.270631766 0.343111905 0.628521774 0.071852588 0.232465089
         97          98          99         100         101         102
0.324061539 0.035442529 0.428016352 0.969887639 0.061875988 0.471727762
        103         104         105         106         107         108
0.213891926 0.786771951 0.168250872 0.243400750 0.762784723 0.763421380
```

109	110	111	112	113	114
0.068256033	0.043455453	0.981011875	0.245309284	0.709739797	0.028139622
115	116	117	118	119	120
0.024962409	0.922124164	0.063957115	0.357613062	0.482637937	0.810966930
121	122	123	124	125	126
0.297313315	0.030018008	0.202839757	0.691050016	0.897606809	0.776953524
127	128	129	130	131	132
0.320164021	0.044997881	0.029918979	0.198345781	0.072705117	0.101923438
133	134	135	136	137	138
0.406351384	0.349673265	0.542977174	0.705530920	0.506268413	0.034844025
139	140	141	142	143	144
0.188359990	0.050351740	0.206406867	0.676517618	0.376140579	0.562686749
145	146	147	148	149	150
0.305730906	0.196385711	0.296952847	0.322095172	0.068166953	0.264511586
151	152	153	154	155	156
0.617468689	0.117213121	0.412982499	0.351232292	0.182614359	0.354265649
157	158	159	160	161	162
0.141779724	0.144377632	0.040181217	0.190558115	0.238664955	0.617308309
163	164	165	166	167	168
0.250028973	0.327137012	0.160301073	0.129534252	0.047275044	0.031498043
169	170	171	172	173	174
0.751220132	0.662633542	0.144322629	0.110487122	0.577144563	0.259642421
175	176	177	178	179	180
0.038988959	0.051799546	0.363755160	0.210180156	0.900874831	0.775849556
181	182	183	184	185	186
0.458824429	0.136435266	0.026225482	0.450898906	0.935595307	0.064095056
187	188	189	190	191	192
0.106784553	0.351947563	0.823686569	0.039999110	0.079111820	0.370518108
193	194	195	196	197	198
0.146545160	0.107776105	0.094270709	0.100314244	0.067221478	0.744146226
199	200	201	202	203	204
0.167350501	0.200756321	0.113983318	0.143810450	0.408762270	0.156398071
205	206	207	208	209	210
0.401726723	0.362884070	0.822971133	0.171005768	0.640063874	0.171521457
211	212	213	214	215	216
0.304017770	0.567886817	0.149849999	0.381420920	0.046024748	0.151143404
217	218	219	220	221	222
0.738029984	0.777598823	0.797044021	0.378284620	0.180118415	0.119782012
223	224	225	226	227	228
0.038262897	0.048130103	0.134250795	0.991840636	0.083945920	0.099784063
229	230	231	232	233	234
0.108496529	0.143009776	0.020316902	0.110187669	0.136427485	0.044641845
235	236	237	238	239	240
0.864718760	0.814509120	0.297471944	0.124324211	0.084945813	0.020121092
241	242	243	244	245	246
0.134190907	0.724169758	0.186493121	0.114596916	0.204174826	0.622026805
247	248	249	250	251	252
0.041915296	0.060903352	0.404141128	0.416844989	0.981388997	0.083743934
253	254	255	256	257	258
0.552897908	0.047432747	0.872845821	0.655205399	0.073899378	0.225251465
259	260	261	262	263	264
0.791856483	0.198981079	0.067652997	0.111239046	0.040554136	0.483768036

```
        265         266         267         268         269         270
0.742497802 0.462317970 0.014183952 0.194191095 0.022130535 0.072595534
        271         272         273         274         275         276
0.134391981 0.180545181 0.202491695 0.100286799 0.331766673 0.508752857
        277         278         279         280         281         282
0.367783875 0.265117917 0.068289223 0.069727559 0.836020309 0.979265618
        283         284         285         286         287         288
0.226402839 0.786876970 0.062418625 0.047275669 0.090411866 0.172263219
        289         290         291         292         293         294
0.898274975 0.080462271 0.093642167 0.061138883 0.111605651 0.135953124
        295         296         297         298         299         300
0.534998029 0.233351918 0.095774911 0.070396421 0.380578076 0.180508303
        301         302         303         304         305         306
0.160086522 0.508892652 0.940897119 0.162051609 0.138190110 0.368181545
        307         308         309         310         311         312
0.731363676 0.043915864 0.068696822 0.828049083 0.877611527 0.030425185
        313         314         315         316         317         318
0.555562884 0.051223539 0.085818529 0.769825225 0.859085987 0.699682956
        319         320         321         322         323         324
0.009602513 0.193370593 0.123331348 0.109348950 0.091020292 0.107052602
        325         326         327         328         329         330
0.082197662 0.290395156 0.033084294 0.115830715 0.110578773 0.564381338
        331         332         333         334         335         336
0.482253344 0.702843391 0.364876221 0.030797167 0.197242626 0.209937576
        337         338         339         340         341         342
0.083999398 0.496161648 0.029910417 0.641860169 0.140051431 0.770246432
        343         344         345         346         347         348
0.687824579 0.125664825 0.219726199 0.585496336 0.840801716 0.100938057
        349         350         351         352         353         354
0.678757518 0.060171364 0.008692296 0.145603949 0.272068663 0.247511294
        355         356         357         358         359         360
0.794785608 0.349840829 0.549685133 0.432594166 0.546033497 0.311503204
        361         362         363         364         365         366
0.249578428 0.080373450 0.171165930 0.125056325 0.387372383 0.325019941
        367         368         369         370         371         372
0.144139057 0.869835699 0.799460595 0.139765854 0.154132474 0.521835630
        373         374         375         376         377         378
0.389933348 0.207183726 0.246093243 0.816438344 0.078161324 0.139530999
        379         380         381         382         383         384
0.101673320 0.689505603 0.091574423 0.067780850 0.859514998 0.219156404
        385         386         387         388         389         390
0.304520444 0.850335207 0.260845465 0.786081780 0.590996318 0.057646960
        391         392
0.386531825 0.129443275
```

```r
possible_cutoffs <- seq(0, 1, by = .01)

percent_missclass <- rep(NA, length(possible_cutoffs))

for(i in 1:length(possible_cutoffs)){
  classify <- ifelse(dia_preds > possible_cutoffs[i], 1, 0)
  percent_missclass[i] <- mean(classify != dia$diabetes)
```

```
}

missclass_df <- as.data.frame(cbind(percent_missclass, possible_cutoffs))

ggplot(data = missclass_df) +
  geom_line(aes(x = possible_cutoffs, y = percent_missclass))
```




```
cutoff_best <- possible_cutoffs[which.min(percent_missclass)]

percent_missclass == min(percent_missclass)
```

```
  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
 [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [97] FALSE FALSE FALSE FALSE FALSE
```

```
print(cutoff_best)
```

```
[1] 0.43
```

0.43 Appears to be the best cutoff value

22. Create (and output) a confusion matrix using the best cutoff value you found above.

```
preds <- dia_preds > cutoff_best

conf_mat <- table("truth" = dia$diabetes, "predicted" = preds)

addmargins(conf_mat)
```

```
     predicted
truth FALSE TRUE Sum
  0     230   32 262
  1      44   86 130
  Sum   274  118 392
```

23. Based on the confusion matrix, what is the value for the specificity, and what does the specificity measure? Print the specificity.

```
specificity <- 230/(230+32)
print(specificity)
```

```
[1] 0.8778626
```

Specificity is the percent of true negatives. Therefore, the specificity of this model is roughly 87.79% accurately identify those without diabetes from those that do have diabetes.

24. Based on the confusion matrix, what is the value for the sensitivity, and what does the sensitivity measure? Print the sensitivity.

```
sensitvity<-86/(44+86)
print(sensitvity)
```

```
[1] 0.6615385
```

Sensitivity is the percent of true positives. Meaning that our model is roughly 66.15% accurate in accurately identifying those that truly have diabetes among those that indeed have diabetes.

25. Based on the confusion matrix, what is the percent correctly classified (accuracy), and what does the percent correctly classified measure? Print the percent correctly classified.

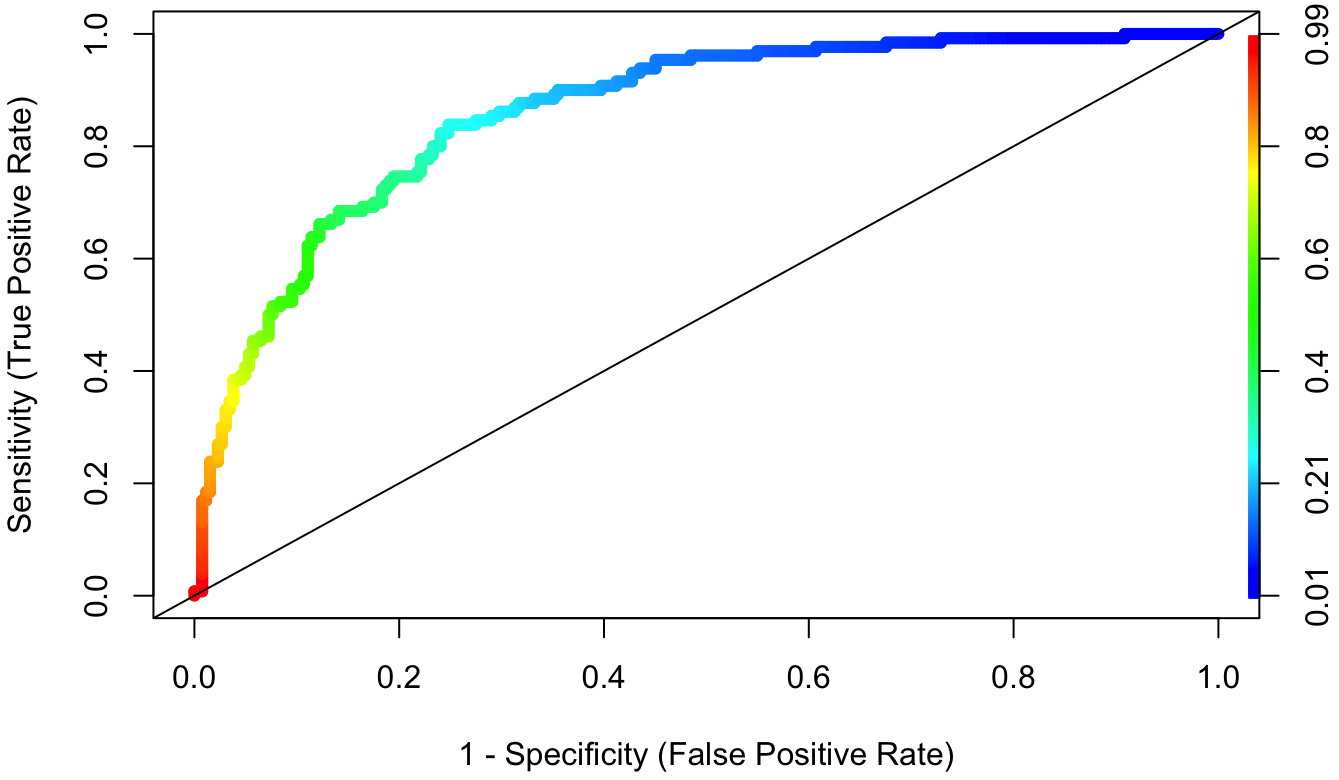
```
accuracy <-(230+86)/(230+32+44+86)
print(accuracy)
```

```
[1] 0.8061224
```

Accuracy is the sum of our true negative and true positives weighted by all of our results in the matrix. Meaning, the percentage of correctly clasified diabetes (has or does not have) cases is roughly 80.61%

26. Plot (and output) the ROC curve for the model (either using the `pROC` package or the `ROCR` package).

```
pred <- prediction(dia_preds, dia$diabetes)
perf <- performance(pred, "tpr", "fpr")  # tpr = true positive rate,

plot(perf,
     colorize = TRUE,
     lwd = 6,
     xlab = "1 - Specificity (False Positive Rate)",
     ylab = "Sensitivity (True Positive Rate)")
abline(a = 0, b = 1)
```



27. What is the AUC for the ROC curve plotted above? Print the value of the AUC.

```
auc <- performance(pred, measure = "auc")
auc@y.values[[1]]
```

```
[1] 0.8604521
```

```r
str(auc)
```

```
Formal class 'performance' [package "ROCR"] with 6 slots
  ..@ x.name      : chr "None"
  ..@ y.name      : chr "Area under the ROC curve"
  ..@ alpha.name  : chr "none"
  ..@ x.values    : list()
  ..@ y.values    :List of 1
  .. ..$ : num 0.86
  ..@ alpha.values: list()
```

28. Briefly summarize what you learned, personally, from this analysis about the statistics, model fitting process, etc.

Very interesting to go into more depth into how we can use regression with categorical variables, specifically those that are yes or no. To see what we need to do differently when dealing with this type of data versus continuous data. I can see that this could come in very handy in the real world. One of the most interesting parts was to see how we can investigate how accurately our model is predicting outcomes and how often it is wrong. This could also be very harmful in the real world if we are not careful.

29. Briefly summarize what you learned from this analysis *to a non-statistician*. Write a few sentences about (1) the purpose of this data set and analysis and (2) what you learned about this data set from your analysis. Write your response as if you were addressing a business manager (avoid using statistics jargon) and just provide the main take-aways.

There are many things in a person's life that could help us predict whether or not they are at risk of diabetes. Information was gathered on almost 400 random women, along with additional details about them such as age, their glucose levels, etc.

We found that using a woman's glucose, BMI, pedigree and age were the best items to help us determine if the woman was at risk of diabetes or not. We were also able to use statistics to help us create a method that was accurate in determining whether or not a woman was at risk, with an accuracy 80% of the time, only by using these aspects of the woman's health.