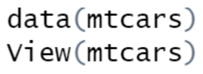
MTH 325, Lab #5, Spring 2023   Name ____

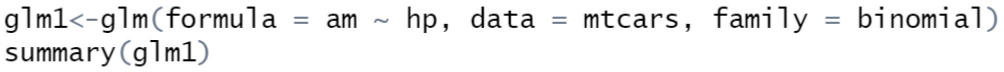
Instructions: Follow along with the tutorial portion of the lab. Replicate the code examples in R on your own, along with the demonstration. Then use those examples as a model to answer the questions/perform the tasks that follow. Copy and paste the results of your code to answer questions where directed. Submit your response file and the code used (both for the tutorial and part two). Your code file and your lab response file should each include your name inside. Be sure to follow the write-up directions in the Lab Directions file.

As we did with our examples in lecture, we are going to apply the logistic regression model using the mtcars dataset, to predict the am variable (whether the vehicle is American made or not). Begin by loading the mtcars data.



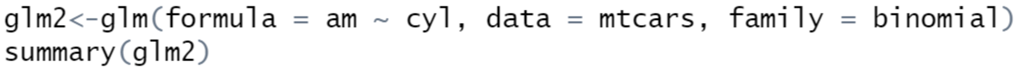
```
data(mtcars)
View(mtcars)
```

We are going to start with a single variable, and then look at a more complete model. Let's start by using hp (horsepower) to predict am.



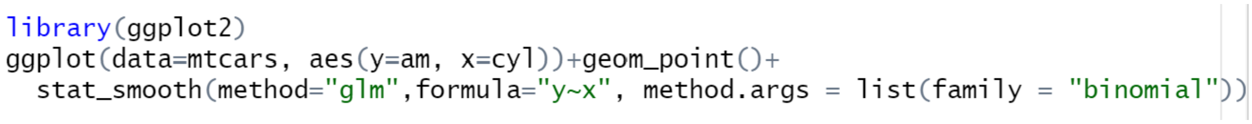
```
glm1<-glm(formula = am ~ hp, data = mtcars, family = binomial)
summary(glm1)
```

In this case, we find the neither the intercept nor the slope achieves significance. Using forward selection, we would eliminate this variable and try another.



```
glm2<-glm(formula = am ~ cyl, data = mtcars, family = binomial)
summary(glm2)
```

For one-variable models, we can create a plot of the variables and the logistic curve.

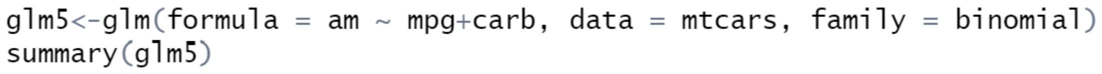


```
library(ggplot2)
ggplot(data=mtcars, aes(y=am, x=cyl))+geom_point()+
  stat_smooth(method="glm",formula="y~x", method.args = list(family = "binomial"))
```

Our graph here will look a little strange because cyl (cylinders) is a discrete variable, and the dots plot on top of each other, so it's hard to see how the probabilities change. Experiment in your graph with other variables, such as mpg.

Compare the results of your model now to another version of your model after converting your am variable to a factor. Is there a difference? It will matter later on when we do predictions.

Continue with the selection process until you can build a model with two variables and where all coefficients are significant. After some experimentation, I came to the model below.



```
glm5<-glm(formula = am ~ mpg+carb, data = mtcars, family = binomial)
summary(glm5)
```

The more variables you use, the higher the p-values get in this data, so a two variable model seems to work the best.

Consider the residual diagnostic plots for the final model.

```
plot(glm5)
```

Note any potential outliers. While we won't address them in this lab, it is something we should be aware of and address it in any final model.

To test the accuracy of the model, we'll create the confusion matrix. For that, we'll need the {caret} package, which you'll need to install the first time you use it.

```
library(caret)
predicted <- as.factor(round(predict(glm5, mtcars, type="response")))
mtcars$am<-as.factor(mtcars$am)
caret::confusionMatrix(mtcars$am, predicted)
```

In addition to the confusion matrix, there are a number of performance plots we can use to optimize our model. We can find several in the ROCR package.

```
library(ROCR)

preds<-predict(glm5, mtcars, type="response")
pred = prediction(preds, mtcars$am)
perf = performance(pred, "acc")
plot(perf)
```

Some additional graphs and metrics:

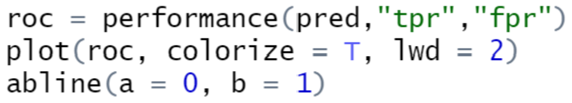
```
perf_cost = performance(pred, "cost")
perf_err = performance(pred, "err")
perf_tpr = performance(pred, "tpr")
perf_sn_sp = performance(pred, "sens", "spec")

plot(perf_cost)
plot(perf_err)
plot(perf_tpr)
plot(perf_sn_sp)
```

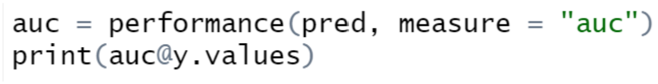
We can use the graph to estimate the value, or let R calculate the optimal cut-off value.

```
max_ind = which.max(slot(perf, "y.values")[[1]] )
acc = slot(perf, "y.values")[[1]][max_ind]
cutoff = slot(perf, "x.values")[[1]][max_ind]
print(c(accuracy= acc, cutoff = cutoff))
```

Finally, we can create an ROC curve, and the AUC (area under the curve).



```
roc = performance(pred,"tpr","fpr")
plot(roc, colorize = T, lwd = 2)
abline(a = 0, b = 1)
```



```
auc = performance(pred, measure = "auc")
print(auc@y.values)
```

Tasks:

1. Use the data in **325lab5data.xlsx** to create a predictive model for admission status (admitted or not). Start by finding the best one-variable model. Create appropriate plots of the original data and logistic regression curve, and diagnostic plots, including the AUC and confusion matrix. Then try to build a better model using additional variables. Use the model selection procedure of your choice. Then create appropriate diagnostic plots, the AUC and confusion matrix. Describe the improvement of the new model over the one-variable model, including accuracy and change in residual deviance.

Resources:

1. https://www.tutorialspoint.com/r/r_logistic_regression.htm
2. https://stats.oarc.ucla.edu/r/dae/logit-regression/
3. https://www.r-bloggers.com/2015/09/how-to-perform-a-logistic-regression-in-r/
4. https://www.statology.org/confusion-matrix-in-r/
5. https://cran.r-project.org/web/packages/ConfusionTableR/vignettes/ConfusionTableR.html
6. https://www.datatechnotes.com/2019/03/how-to-create-roc-curve-in-r.html