Project 2: Modeling, Testing, and Predicting

SDS348 Fall 2019

Modeling

Instructions

A knitted R Markdown document (as a PDF) and the raw R Markdown file (as .Rmd) should both be submitted to Canvas by 11:59pm on 11/24/2019. These two documents will be graded jointly, so they must be consistent (i.e., don't change the R Markdown file without also updating the knitted document). Knit an html copy too, for later! In the .Rmd file for Project 2, you can copy the first code-chunk into your project .Rmd file to get better formatting. Notice that you can adjust the opts_chunk$set(…) above to set certain parameters if necessary to make the knitting cleaner (you can globally set the size of all plots, etc). I have gone ahead and set a few for you (such as disabling warnings and package-loading messges when knitting)!

Like before, I envision your written text forming something of a narrative structure around your code/output. All results presented must have corresponding code. Any answers/results/plots etc. given without the corresponding R code that generated the result will not be graded. Furthermore, all code contained in your final project document should work properly. Please do not include any extraneous code or code which produces error messages. (Code which produces warnings is acceptable, as long as you understand what the warnings mean).

Find data:

Find one dataset with at least 5 variables that wish to use to build model. At least one should be categorical (with 2-5 groups) and at least two should be numeric. Ideally, one of your variables will be binary (if not, you will need to create one by discretizing a numeric, which is workable but less than ideal). You will need a minimum of 40 observations (*at least* 10 observations for every explanatory variable you have, ideally 20+ observations/variable).

It is perfectly fine to use either dataset (or the merged dataset, or a subset of your variables) from Project 1. However, you could also diversify your portfolio a bit by choosing a different dataset to work with (particularly if the variables did not reveal interesting associations in Project 1). The only requirement/restriction is that you may not use data from any examples we have done in class or lab. It would be a good idea to pick more cohesive data this time around (i.e., variables that you actually thing might have a relationship you would want to test). Think more along the lines of your Biostats project.

Again, you can use data from anywhere you want (see bottom for resources)! If you want a quick way to see whether a built-in (R) dataset has binary and/or character (i.e., categorical) variables, check out this list: https://vincentarelbundock.github.io/Rdatasets/datasets.html (https://vincentarelbundock.github.io/Rdatasets/datasets.html).

Guidelines and Rubrc

- **0. (5 pts)** Introduce your dataset and each of your variables (or just your main variables if you have lots) in a paragraph.

The dataset I chose is is medical school admission statuses and information based on GPA and standardized test scores. The data was was collected from midwestern libreral college. The main variables I chose to focus on were: Accept: whether the person was accepted or denied, Sex: whether male or female, GPA: college grade point average, and MCAT: score on the MCAT exam.

- **1. (15 pts)** Perform a MANOVA testing whether any of your numeric variables (or a subset of them, if including them all doesn't make sense) show a mean difference across levels of one of your categorical variables (3). If they do, perform univariate ANOVAs to find response(s) showing a mean difference across groups (3), and perform post-hoc t tests to find which groups differ (3). Discuss the number of tests you have performed, calculate the probability of at least one type I error (if unadjusted), and adjust the significance level accordingly (bonferroni correction) before discussing significant differences (3). Briefly discuss assumptions and whether or not they are likely to have been met (2).

```
#Manova Test
MedGPA<-read_csv("MedGPA.csv")
man1<-manova(cbind(GPA,MCAT)~Accept, data=MedGPA)
summary(man1)
```

```
## Df Pillai approx F num Df den Df Pr(>F)
## Accept 1 0.31225 11.805 2 52 5.931e-05 ***
## Residuals 53
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
#Univariate ANOVAs
summary.aov(man1)
```

```
## Response GPA :
## Df Sum Sq Mean Sq F value Pr(>F)
## Accept 1 1.2947 1.29472 21.879 2.043e-05 ***
## Residuals 53 3.1363 0.05918
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Response MCAT :
## Df Sum Sq Mean Sq F value Pr(>F)
## Accept 1 212.4 212.402 10.819 0.001789 **
## Residuals 53 1040.5 19.632
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
#Post-Hoc T tests
MedGPA%>%group_by(Accept)%>%summarize(mean(GPA),mean(MCAT))
```

```
## # A tibble: 2 x 3
##   Accept `mean(GPA)` `mean(MCAT)`
##   <chr>        <dbl>        <dbl>
## 1 A             3.69         38.1
## 2 D             3.39         34.1
```

```
pairwise.t.test(MedGPA$MCAT,MedGPA$Accept,p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  MedGPA$MCAT and MedGPA$Accept
##
##   A
## D 0.0018
##
## P value adjustment method: none
```

```
pairwise.t.test(MedGPA$GPA,MedGPA$Accept,p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  MedGPA$GPA and MedGPA$Accept
##
##   A
## D 2e-05
##
## P value adjustment method: none
```

```
#1 MANOVA, 2 ANOVAs, 2 t tests
type1error=1-(1-.05)^5
type1error
```

```
## [1] 0.2262191
```

```
#Boneforroni
Boneforroni=(.05)/5
Boneforroni
```

```
## [1] 0.01
```

```
#adjusted
pairwise.t.test(MedGPA$MCAT,MedGPA$Accept,p.adj="bonf")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  MedGPA$MCAT and MedGPA$Accept
##
##   A
## D 0.0018
##
## P value adjustment method: bonferroni
```

```
pairwise.t.test(MedGPA$GPA,MedGPA$Accept,p.adj="bonf")
```

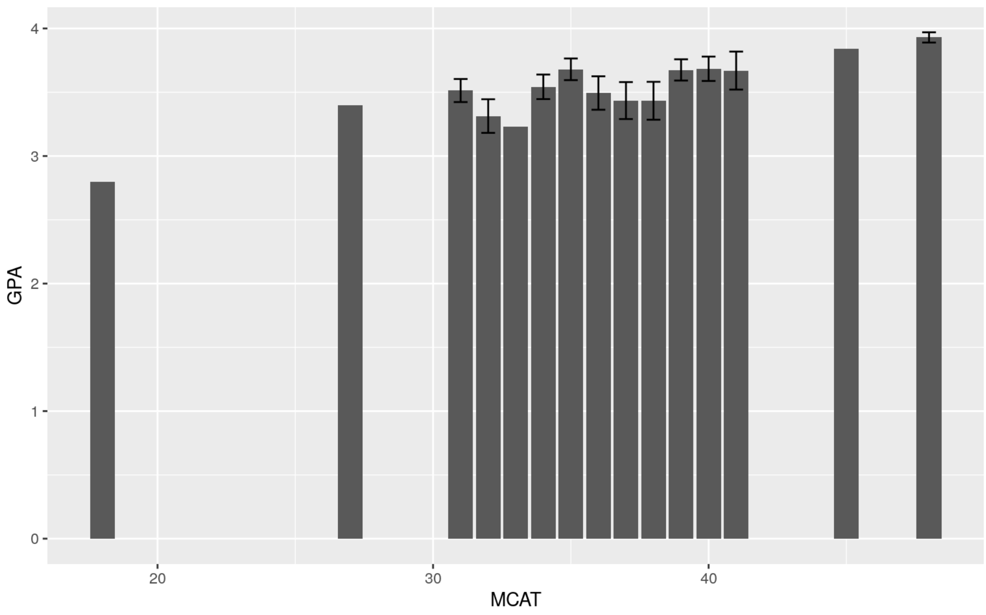
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  MedGPA$GPA and MedGPA$Accept
##
##   A
## D 2e-05
##
## P value adjustment method: bonferroni
```

I conducted a Manova Test, Univariate ANOVAS because my Manova test showed a mean different accross levels in my Accept categorical variable, and I also ran a post-hoc t test. Because my univariate ANOVAS spit out two tests, I divided .05 by the five total tests performed (1 MANOVA, 2 ANOVAS, and 2 t tests. - **2. (10 pts)** Perform some kind of randomization test on your data (that makes sense). This can be anything you want! State null and alternative hypotheses, perform the test, and interpret the results (7). Create a plot visualizing the null distribution and the test statistic (3).

```
library(dplyr)
install.packages("vegan")
library(vegan)
dists<-MedGPA%>%dist()
adonis(dists~Accept,data=MedGPA)
```

```
##
## Call:
## adonis(formula = dists ~ Accept, data = MedGPA)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
## Df SumsOfSqs MeanSqs F.Model R2 Pr(>F)
## Accept 1 2594.3 2594.28 7.6749 0.12649 0.004 **
## Residuals 53 17915.1 338.02 0.87351
## Total 54 20509.4 1.00000
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
MedGPA%>%ggplot(aes(x=MCAT,y=GPA))+geom_bar(stat="summary")+geom_errorbar
(stat="summary", width=.5)
```

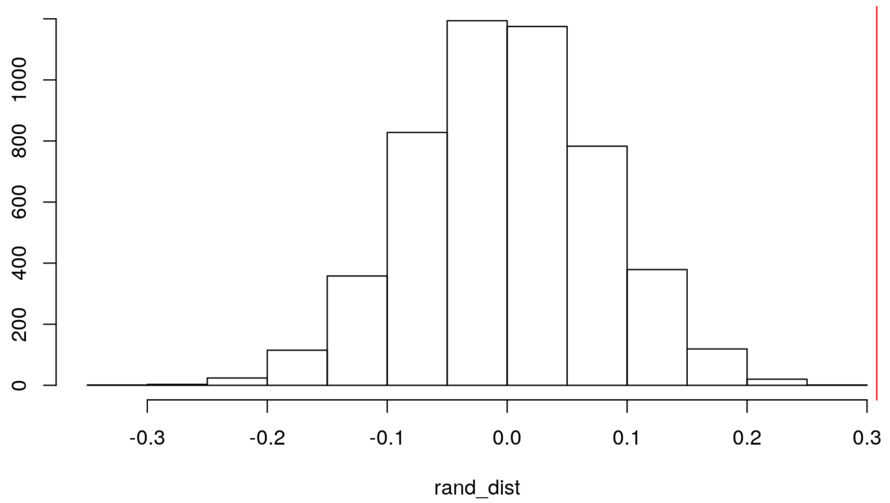


```
#Randomization Test
mean_diff<-mean(MedGPA[MedGPA$Accept=="A",]$GPA)-mean(MedGPA[MedGPA$Accep
t=="D",]$GPA)
#Permutation Loop
rand_dist<-vector()
for(i in 1:5000){
  new<-data.frame(GPA=sample(MedGPA$GPA),Accept=MedGPA$Accept)
  rand_dist[i]<-mean(new[new$Accept=="D",]$GPA)-mean(new[new$Accept=="A",
]$GPA)}

#P-value for Permutation Test
mean(rand_dist>mean_diff)*2
```

```
## [1] 0
```

```
#Plot Visualizing Null Distribution and Test Statistic
{hist(rand_dist,main="",ylab="");abline(v=mean_diff,col="red")}
```



The p-value from the randomization is 2 being above .05 we cannot reject the null hypothesis and conclude the mean Accept is not different different between acceptances or denies. - **3. (35 pts)** Build a linear regression model predicting one of your response variables from at least 2 other variables, including their interaction. Mean-center any numeric variables involved in the interaction.

```
library(dplyr)
library(MASS)
library(ggplot2)
library(lmtest)
library(tidyverse)
library(sandwich)
#linear regression with interaction
fit1<-lm(GPA~MCAT,data=MedGPA)
coef(fit1)
```

```
## (Intercept)        MCAT
##  2.38537092  0.03219779
```

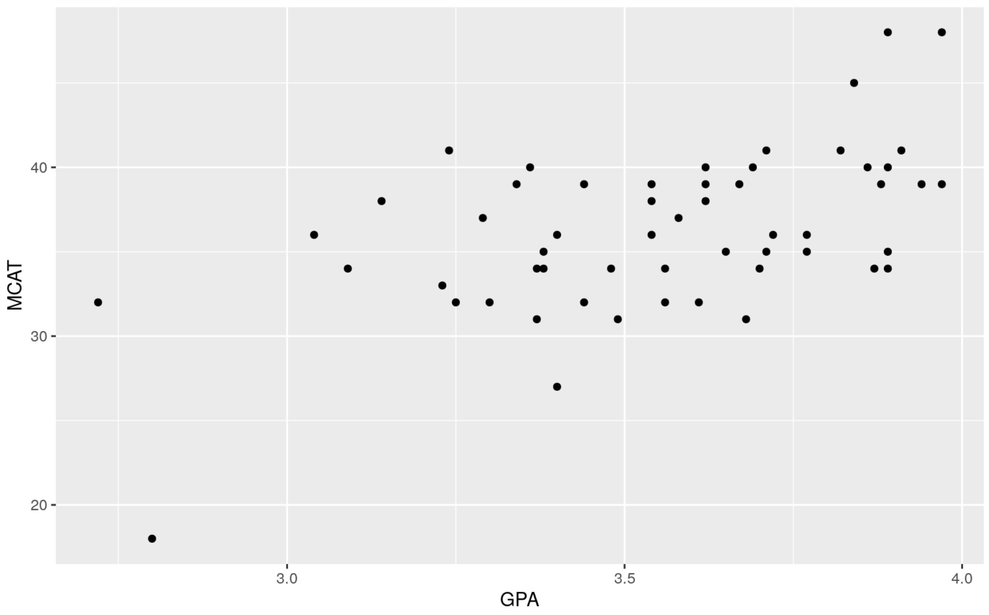
```
summary(fit1)
```

```
##
## Call:
## lm(formula = GPA ~ MCAT, data = MedGPA)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.69570 -0.12400 0.00573 0.19051 0.40990
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.385371 0.251263 9.494 5.02e-13 ***
## MCAT 0.032198 0.006868 4.688 1.97e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Residual standard error: 0.2431 on 53 degrees of freedom
## Multiple R-squared: 0.2931, Adjusted R-squared: 0.2798
## F-statistic: 21.98 on 1 and 53 DF, p-value: 1.969e-05
```

```
cov(MedGPA$MCAT,MedGPA$GPA)/var(MedGPA$MCAT)
```

```
## [1] 0.03219779
```

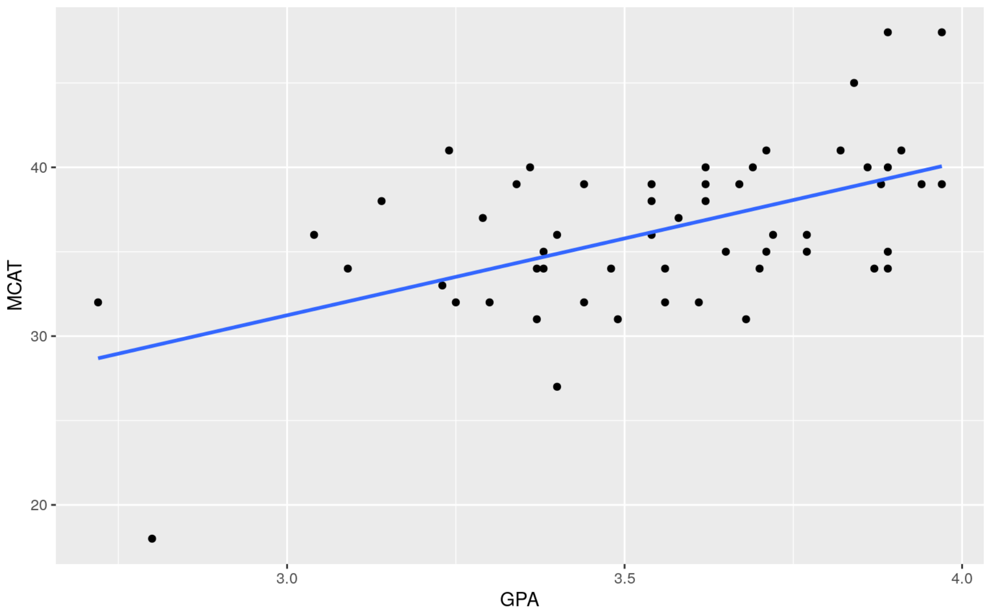
```
library(tidyverse)
MedGPA$Accept<-factor(MedGPA$Accept,labels = c("D","A"))
MedGPA$Sex<-factor(MedGPA$Sex,labels = c("F","M"))
MedGPA%>%ggplot(aes(GPA,MCAT))+geom_point()
```



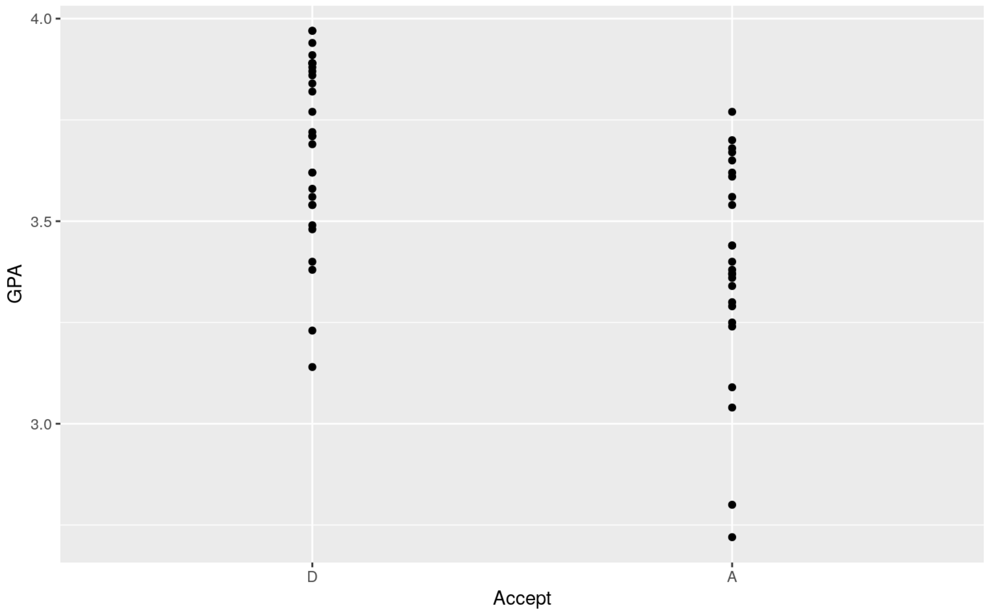
```
fit<-lm(MCAT~GPA,data=MedGPA)
#Hypothesis test
summary(fit)
```

```
##
## Call:
## lm(formula = MCAT ~ GPA, data = MedGPA)
##
## Residuals:
## Min 1Q Median 3Q Max
## -11.4148 -2.5168 -0.1519 2.6653 8.6616
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.923 6.922 0.567 0.573
## GPA 9.104 1.942 4.688 1.97e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Residual standard error: 4.088 on 53 degrees of freedom
## Multiple R-squared: 0.2931, Adjusted R-squared: 0.2798
## F-statistic: 21.98 on 1 and 53 DF, p-value: 1.969e-05
```

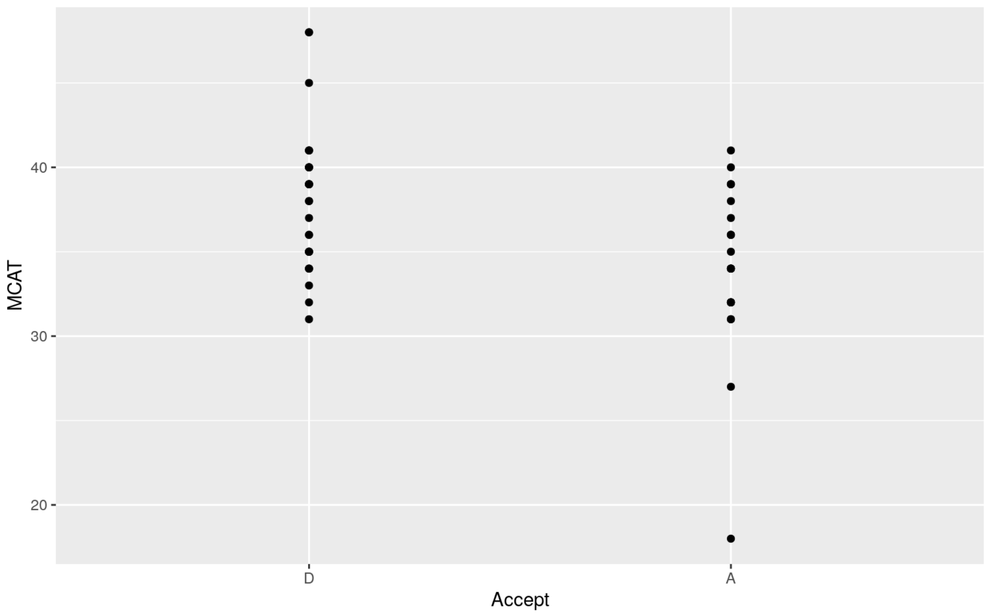
```
MedGPA%>%ggplot(aes(GPA,MCAT))+geom_point()+geom_smooth(method='lm',se=F)
```



```
#Accepted by GPA
MedGPA%>%ggplot(aes(Accept,GPA))+geom_point()+geom_smooth(method='lm',se=F)
```



```
#Accepted by MCAT Score
MedGPA%>%ggplot(aes(Accept,MCAT))+geom_point()+geom_smooth(method='lm',se=F)
```



```
#interpret the coefficient estimates
#Recompute regression results
coeftest(fit,vcov=vcovHC(fit))
```

```
##
## t test of coefficients:
##
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.9229 10.7738 0.3641 0.717218
## GPA 9.1042 2.9852 3.0497 0.003571 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
#Proportion of variation
SST<-sum((MedGPA$GPA-mean(MedGPA$GPA))^2)
SSR<-sum((fit$fitted.values-mean(MedGPA$GPA))^2)
SSE<-sum(fit$residuals^2)
SSR/SST
```

```
## [1] 13371.27
```

The intercept estimate 3.923 is the average GPA when there is no interaction between sex and MCAT score. 9.104 is the coefficent estimate of GPA, as the Acceptance increases by GPA. - Interpret the coefficient estimates (do not discuss significance) (10) - Plot the regression using `ggplot()` . If your interaction is numeric by numeric, refer to code near the end of WS15 to make the plot. If you have 3 or more predictors, just chose two to plot for convenience. (7) - Check assumptions of linearity, normality, and homoskedasticity either graphically or using a hypothesis test (3) - Regardless, recompute regression results with robust standard errors via `coeftest(..., vcov=vcovHC(...))` . Discuss significance of results, including any changes from before/after robust SEs if applicable. (7) - What proportion of the variation in the outcome does your model explain? (3) - Finally, rerun the regression but without interactions (just main effects); compare this with the interaction model and the null model using a likelihood ratio test (4)********

- **4. (5 pts)** Rerun same regression model (with interaction), but this time compute bootstrapped standard errors. Discuss any changes you observe in SEs and p-values using these SEs compared to the original SEs and the robust SEs)

```
#bootstrapped SE
MedGPA_dat<-MedGPA[sample(nrow(MedGPA),replace=TRUE),]
library(sandwich); library(lmtest)
samp_distn<-replicate(5000, {
     MedGPA_dat<-MedGPA[sample(nrow(MedGPA),replace=TRUE),]
     fit<-lm(MCAT~GPA,data=MedGPA_dat)
     coef(fit)
 })

samp_distn%>%t%>%as.data.frame%>%summarize_all(sd)
```

```
##   (Intercept)      GPA
## 1    9.557743 2.655536
```

```
fit10<-lm(Accept~Sex*MCAT*GPA,data=MedGPA)
resids<-fit10$residuals
fitted<-fit10$fitted.values
```

The bootstrap SE was 2.675029, being less than the original standard error of 3.923. - **5. (40 pts)** Perform a logistic regression predicting a binary categorical variable (if you don't have one, make/get one) from at least two explanatory variables (interaction not necessary).

```
library(dplyr)
library(MASS)
library(ggplot2)
library(lmtest)
library(sandwich)
library(Matrix)
library(plotROC)
install.packages("vegan")

#Logistic Regression
fit2<-glm(Sex~GPA+Accept,data=MedGPA,family=binomial(link="logit"))
coeftest(fit2)
```

```
## 
## z test of coefficients:
## 
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.20034    4.27592 -0.2807   0.7789
## GPA          0.21516    1.15267  0.1867   0.8519
## AcceptA      0.87776    0.66027  1.3294   0.1837
```

```
exp(-8.1331e-11)
```

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

```
#Confussion Matrix
prob<-predict(fit,type="response")
#table(predict=as.numeric(MedGPA$prob>.5),truth=MedGPA$Accept)%>%addmargi
ns
#Accuracy
30/55
```

```
## [1] 0.5454545
```

```
#Sensitivity
1/30
```

```
## [1] 0.03333333
```

```
#Specificity
30/55
```

```
## [1] 0.5454545
```

```
#Density of Log-Odds Plot
#MedGPA$odds<-(MedGPA$prob)/(1-MedGPA$prob)
#MedGPA$logit<-log(MedGPA$odds)

#ggplot(MedGPA)+geom_density(aes(logit,fill=Sex),alpha=.3)

#ROC Curve and Plot
#ROCplot<-ggplot(MedGPA)+geom_roc(aes(d=y,m=prob),n.cuts=0)+geom_segment(
aes(x=0,xend=1,y=0,yend=1),lty=2)
#view(ROCplot)
#ROCplot
#calc_auc(ROCplot)


fit7<-lm(GPA~Accept,data=MedGPA)
summary(fit7)
```

```
##
## Call:
## lm(formula = GPA ~ Accept, data = MedGPA)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.66520 -0.13427 0.01667 0.19667 0.38480
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.69333 0.04441 83.159 < 2e-16 ***
## AcceptA -0.30813 0.06588 -4.678 2.04e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Residual standard error: 0.2433 on 53 degrees of freedom
## Multiple R-squared: 0.2922, Adjusted R-squared: 0.2788
## F-statistic: 21.88 on 1 and 53 DF, p-value: 2.043e-05
```

```
#auc(MedGPA$GPA,prob)



#class diagnostic function
#class_diag<-function(probs,truth){ tab<-table(factor(probs>.5,levels=c("
FALSE","TRUE")),truth) acc=sum(diag(tab))/sum(tab)
#sens=tab[2,2]/colSums(tab)[2]
#spec=tab[1,1]/colSums(tab)[1]
#ppv=tab[2,2]/rowSums(tab)[2]
#if(is.numeric(truth)==FALSE&is.logical(truth)== FALSE)truth<-as.numeric(
truth)-1 ord<-order(probs, decreasing=TRUE)
#probs <- probs[ord]; truth <- truth[ord]
#TPR=cumsum(truth)/max(1,sum(truth))
#FPR=cumsum(!truth)/max(1,sum(!truth))
#dup<-c(probs[-1]>=probs[-length(probs)], FALSE)
#TPR<-c(0,TPR[!dup],1); FPR<-c(0,FPR[!dup],1)
#n <- length(TPR)
#auc<- sum( ((TPR[-1]+TPR[-n])/2) * (FPR[-1]-FPR[-n]) ) data.frame(acc,se
ns,spec,ppv,auc)
#}

#10 fold
#set.seed(1234)
#k=10
#data1<-MedGPA[sample(nrow(MedGPA)),] folds<-cut(seq(1:nrow(MedGPA)),brea
ks=k,labels=F) diags<-NULL
#for(i in 1:k){
#train<-data1[folds!=i,]
#test<-data1[folds==i,]
#truth<-test$GPA
#fit<-glm(y~GPA+MCAT,data=train,family="binomial") probs<-predict(fit,new
data = test,type="response") diags<-rbind(diags,class_diag(probs,truth))
#}
```

```
- Interpret coefficient estimates in context (10)
- Report a confusion matrix for your logistic regression (2)
- Compute and discuss the Accuracy, Sensitivity (TPR), Specificity (TNR),
and Recall (PPV) of your model (5)
- Using ggplot, plot density of log-odds (logit) by your binary outcome v
ariable (3)
- Generate an ROC curve (plot) and calculate AUC (either manually or with
a package); interpret (10)
- Perform 10-fold (or repeated random sub-sampling) CV and report average
out-of-sample Accuracy, Sensitivity, and Recall (10)
```

*There is a significant effect of Acceptance based on GPA, a 1.0 increase as GPA increases. As the GPA increases, as will the chance for Acceptance.The confusion matrix, for accuracy is .5454545 meaning that the number of predicted denials that are actually denied. The sensitivity is because there is .0333 perdicted Acceptance. The specificity is .5454545 because there was only predicted denials. And the precision of acceptances that are actually acceptances is .01818182. According to the 1-fold CV the out of sample accuracy is NaN. - **6. (10 pts)** Choose one variable you want to predict (can be one you used from before; either binary or continuous) and run a LASSO regression inputting all the rest of your variables as predictors. Choose lambda to give the simplest model whose accuracy is near that of the best (i.e., `lambda.1se`). Discuss which variables are retained. Perform 10-fold CV using this model: if response in binary, compare model’s out-of-sample accuracy to that of your logistic regression in part 5; if response is numeric, compare the residual standard error (at the bottom of the summary output, aka RMSE): lower is better fit!

```
install.packages("glmnet")
library(glmnet)
library(dplyr)
library(shiny)
library(ggplot2)
#dummy-coding the accept variable
MedGPA<-MedGPA%>%mutate(y=as.numeric(Accept=="A"))
#linear regression
fit6<-lm(y~.,data=MedGPA,family="binomial")
#mean squared error
yhat<-predict(fit6)
data(MedGPA)
#Lasso
y<-as.matrix(MedGPA$GPA)
x<-model.matrix(fit6)
x
```

```
## (Intercept) X1 AcceptA Acceptance SexM BCPM GPA VR PS WS
BS MCAT Apps
## 1 1 1 1 0 0 3.59 3.62 11 9 9 9 38 5
## 2 1 2 0 1 1 3.75 3.84 12 13 8 12 45 3
## 3 1 3 0 1 0 3.24 3.23 9 10 5 9 33 19
## 4 1 4 0 1 0 3.74 3.69 12 11 7 10 40 5
## 5 1 5 0 1 0 3.53 3.38 9 11 4 11 35 11
## 6 1 6 0 1 1 3.59 3.72 10 9 7 10 36 5
## 7 1 7 0 1 1 3.85 3.89 11 12 6 11 40 5
## 8 1 8 1 0 1 3.26 3.34 11 11 8 9 39 7
## 9 1 9 0 1 0 3.74 3.71 8 10 6 11 35 5
## 10 1 10 0 1 0 3.86 3.89 9 9 6 10 34 11
## 11 1 11 0 1 0 4.00 3.97 11 9 8 11 39 6
## 12 1 12 0 1 0 3.35 3.49 11 8 4 8 31 9
## 13 1 13 0 1 1 3.77 3.77 8 10 7 10 35 5
## 14 1 14 1 0 1 3.60 3.61 9 9 4 10 32 8
## 15 1 15 1 0 1 3.29 3.30 11 8 6 7 32 15
## 16 1 16 0 1 0 3.26 3.54 12 8 8 10 38 6
## 17 1 17 1 0 1 3.75 3.65 8 8 8 11 35 6
## 18 1 18 0 1 1 3.51 3.54 9 10 9 11 39 1
## 19 1 19 1 0 1 3.27 3.25 8 9 5 10 32 5
## 20 1 20 0 1 1 3.95 3.89 13 14 8 13 48 5
## 21 1 21 0 1 0 3.71 3.71 13 10 8 10 41 6
## 22 1 22 1 0 1 3.73 3.77 8 10 8 10 36 7
## 23 1 23 0 1 1 4.00 3.91 10 13 6 12 41 17
## 24 1 24 0 1 1 3.98 3.88 9 10 8 12 39 17
## 25 1 25 1 0 1 3.76 3.68 7 9 6 9 31 10
## 26 1 26 0 1 1 3.51 3.56 6 10 6 10 32 5
## 27 1 27 1 0 0 3.38 3.44 11 9 4 8 32 10
## 28 1 28 0 1 0 3.41 3.58 11 11 6 9 37 14
## 29 1 29 0 1 0 3.15 3.40 10 10 6 10 36 1
## 30 1 30 0 1 0 3.78 3.82 10 11 10 10 41 5
## 31 1 31 0 1 1 3.54 3.62 10 12 6 11 39 7
## 32 1 32 1 0 0 3.14 3.09 7 10 8 9 34 24
## 33 1 33 0 1 0 3.89 3.89 8 9 8 10 35 7
## 34 1 34 1 0 1 3.67 3.70 7 10 8 9 34 11
## 35 1 35 1 0 1 3.22 3.24 11 12 8 10 41 11
## 36 1 36 0 1 1 3.87 3.86 10 11 8 11 40 8
## 37 1 37 1 0 1 3.49 3.54 11 9 10 6 36 18
## 38 1 38 1 0 0 3.08 3.40 8 6 4 9 27 7
## 39 1 39 0 1 0 3.82 3.87 10 10 5 9 34 12
## 40 1 40 0 1 1 3.10 3.14 10 9 8 11 38 12
## 41 1 41 1 0 0 3.56 3.37 6 8 9 8 31 4
## 42 1 42 1 0 0 3.19 3.38 9 8 8 9 34 6
## 43 1 43 0 1 0 3.53 3.62 11 10 8 11 40 5
## 44 1 44 0 1 0 3.98 3.94 13 8 8 10 39 12
## 45 1 45 1 0 0 3.19 3.37 10 9 6 9 34 5
## 46 1 46 1 0 0 3.25 3.36 11 8 9 12 40 12
## 47 1 47 0 1 0 3.98 3.97 11 13 10 14 48 6
## 48 1 48 1 0 1 2.75 3.04 10 9 8 9 36 7
## 49 1 49 1 0 0 3.12 3.29 11 10 8 8 37 12
## 50 1 50 1 0 0 3.53 3.67 12 10 9 8 39 6
## 51 1 51 1 0 1 2.41 2.72 8 8 8 8 32 7
## 52 1 52 1 0 1 3.51 3.56 11 8 6 9 34 6
## 53 1 53 0 1 0 3.43 3.48 7 10 7 10 34 14
## 55 1 55 1 0 1 3.36 3.44 11 11 8 9 39 1
## attr(,"assign")
## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12
## attr(,"contrasts")
## attr(,"contrasts")$Accept
## [1] "contr.treatment"
##
## attr(,"contrasts")$Sex
## [1] "contr.treatment"
```

```
x<-x[,-1]
#cv<-cv.glmnet(x,y,family="binomial")
#lasso<-glmnet(x,y,family="binomial",lambda=cv$lamba.1se)
#coef(lasso)
```

Where do I find data again?

You can choose ANY datasets you want that meet the above criteria for variables and observations. You can make it as serious as you want, or not, but keep in mind that you will be incorporating this project into a portfolio webpage for your final in this course, so choose something that really reflects who you are, or something that you feel will advance you in the direction you hope to move career-wise, or something that you think is really neat, or whatever. On the flip side, regardless of what you pick, you will be performing all the same tasks, so it doesn’t end up being that big of a deal.

If you are totally clueless and have no direction at all, log into the server and type

```
data(package = .packages(all.available = TRUE))
```

This will print out a list of **ALL datasets in ALL packages** installed on the server (a ton)! Scroll until your eyes bleed! Actually, do not scroll that much… To start with something more manageable, just run the command on your own computer, or just run `data()` to bring up the datasets in your current environment. To read more about a dataset, do `?packagename::datasetname` .

If it is easier for you, and in case you don’t have many packages installed, a list of R datasets from a few common packages (also downloadable in CSV format) is given at the following website: https://vincentarelbundock.github.io/Rdatasets/datasets.html (https://vincentarelbundock.github.io/Rdatasets/datasets.html).

- A good package to download for fun/relevant data is `fivethiryeight` . Run `install.packages("fivethirtyeight"),` load the packages with `library(fivethirtyeight)` , run `data()` , and then scroll down to view the datasets. Here is an online list of all 127 datasets (with links to the 538 articles). Lots of sports, politics, current events, etc.

- If you have already started to specialize (e.g., ecology, epidemiology) you might look at discipline-specific R packages (vegan, epi, respectively). We will be using some tools from these packages later in the course, but they come with lots of data too, which you can explore according to the directions above

- However, you *emphatically DO NOT* have to use datasets available via R packages! In fact, I would much prefer it if you found the data from completely separate sources and brought them together (a much more realistic experience in the real world)! You can even reuse data from your SDS328M project, provided it shares a variable in common with other data which allows you to merge the two together (e.g., if you still had the timestamp, you could look up the weather that day: https://www.wunderground.com/history/ (https://www.wunderground.com/history/)). If you work in a research lab or have access to old data, you could potentially merge it with new data from your lab!
- Here is a curated list of interesting datasets (read-only spreadsheet format): https://docs.google.com/spreadsheets/d/1wZhPLMCHKJvwOkP4juclhjFgqIY8fQFMemwKL2c64 (https://docs.google.com/spreadsheets/d/1wZhPLMCHKJvwOkP4juclhjFgqIY8fQFMemwKL2c64
- Here is another great compilation of datasets: https://github.com/rfordatascience/tidytuesday (https://github.com/rfordatascience/tidytuesday)
- Here is the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php (https://archive.ics.uci.edu/ml/index.php)
    - See also https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research#Biological_data (https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research#Biological_data)
- Here is another good general place to look: https://www.kaggle.com/datasets (https://www.kaggle.com/datasets)
- To help narrow your search down or to see interesting variable ideas, check out https://www.tylervigen.com/spurious-correlations (https://www.tylervigen.com/spurious-correlations). This is the spurious correlations website, and it is fun, but if you look at the bottom of each plot you will see sources for the data. This is a good place to find very general data (or at least get a sense of where you can scrape data together from)!
- If you are interested in medical data, check out www.countyhealthrankings.org
- If you are interested in scraping UT data, they make *loads* of data public (e.g., beyond just professor CVs and syllabi). Check out all the data that is available in the statistical handbooks: https://reports.utexas.edu/statistical-handbook (https://reports.utexas.edu/statistical-handbook)

Broader data sources:

Data.gov (www.data.gov) 186,000+ datasets!

Social Explorer (Social%20Explorer) is a nice interface to Census and American Community Survey data (more user-friendly than the government sites). May need to sign up for a free trial.

U.S. Bureau of Labor Statistics (www.bls.gov)

U.S. Census Bureau (www.census.gov)

Gapminder (www.gapminder.org/data), data about the world.

…