R Notebook

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Elle Reagan, epr427

#install.packages("mlbench")  
#install.packages("lmtest")  
library(readxl)  
library(dplyr)  
library(tidyverse)  
library(ggplot2)  
library(lmtest)  
#install.packages("sandwich")  
library(sandwich)  
#install.packages("vegan")  
library(vegan)  
#install.packages("readxl")  
  
 Book4 <- read\_excel("cereal.xlsx", skip = 1)  
cer<- Book4  
  
cer1<- cer%>%select(-name)  
cer1$mfr<-ifelse(cer$mfr== "K", "K","N")  
head(cer1)

## # A tibble: 6 x 5  
## mfr calories sodium shelf rating  
## <chr> <dbl> <dbl> <chr> <dbl>  
## 1 N 70 130 Top 68.4  
## 2 N 120 15 Top 34.0  
## 3 K 70 260 Top 59.4  
## 4 K 50 140 Top 93.7  
## 5 N 110 200 Top 34.4  
## 6 N 110 180 Bottom 29.5

cer$mfr<- ifelse(cer$mfr== "K", 1,0)  
  
  
cer$shelf <- as.factor(cer$shelf)  
cer$mfr<- as.factor(cer$mfr)  
  
  
class\_diag <- function(probs,truth){  
#CONFUSION MATRIX: CALCULATE ACCURACY, TPR, TNR, PPV  
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  
#CALCULATE EXACT AUC  
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,sens,spec,ppv,auc)  
}

#Question 0.  
  
#My datset is a collection of cereals with their name, whether or not they were manufactured by Kellog's, calories per serving, sodium in mg, what shelf they are placed on, and overall rating. I got this dataset from Kaggle. In total, there are 77 observations.

#Question 1.  
  
#MANOVA  
man1<-manova(cbind(rating,sodium, calories )~shelf, data=cer)  
  
summary(man1)

## Df Pillai approx F num Df den Df Pr(>F)   
## shelf 2 0.24965 3.4706 6 146 0.003094 \*\*  
## Residuals 74   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#we see that the pvalue is less than .05 and so we reject the null hypothesis and continue on to univariate ANOVA  
  
#univariate ANOVA  
summary.aov(man1)

## Response rating :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## shelf 2 1719.8 859.92 4.7928 0.01103 \*  
## Residuals 74 13277.0 179.42   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response sodium :  
## Df Sum Sq Mean Sq F value Pr(>F)  
## shelf 2 9628 4814.1 0.6792 0.5101  
## Residuals 74 524489 7087.7   
##   
## Response calories :  
## Df Sum Sq Mean Sq F value Pr(>F)  
## shelf 2 559.5 279.74 0.7317 0.4845  
## Residuals 74 28292.5 382.33

#we see that for rating and sodium the shelf position differs whereas calories do not  
  
diffmeans<- cer%>%group\_by(shelf)%>%summarize(mean(rating),mean(sodium))  
diffmeans

## # A tibble: 3 x 3  
## shelf `mean(rating)` `mean(sodium)`  
## <fct> <dbl> <dbl>  
## 1 Bottom 46.1 176.  
## 2 Middle 35.0 146.  
## 3 Top 45.2 159.

#we see the difference in means in both groups  
  
pairwise.t.test(cer$sodium,cer$shelf,  
 p.adj="none")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: cer$sodium and cer$shelf   
##   
## Bottom Middle  
## Middle 0.25 -   
## Top 0.45 0.58   
##   
## P value adjustment method: none

#we see that the middle and bottom and top and bottom are significant   
pairwise.t.test(cer$rating,cer$shelf,  
 p.adj="none")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: cer$rating and cer$shelf   
##   
## Bottom Middle  
## Middle 0.0093 -   
## Top 0.8050 0.0068  
##   
## P value adjustment method: none

#we see that the top and middle are significant  
  
#number of tests: one manova, 3 anova and 6 t tests  
.05/10

## [1] 0.005

#bonferonni is .005  
#after bonferonni the bottom and middle and top and middle are significant for sugars but rating is no longer significant  
  
#type one error  
1-(.95^10)

## [1] 0.4012631

#type 1 error is .4012631

I think that this data has passed assumptions for random samples and independent observations. The data does not pass for multivariate normality of DV’s since each group does not have greater than 25 counts. Based on this, it would be difficult for the data to pass other assumptions because the dataset is very small.

#Question 2.  
  
dists<-cer%>%select(sodium, calories)%>%dist()  
adonis(dists~shelf,data=cer)

##   
## Call:  
## adonis(formula = dists ~ shelf, data = cer)   
##   
## Permutation: free  
## Number of permutations: 999  
##   
## Terms added sequentially (first to last)  
##   
## Df SumsOfSqs MeanSqs F.Model R2 Pr(>F)  
## shelf 2 10188 5093.9 0.68191 0.0181 0.537  
## Residuals 74 552781 7470.0 0.9819   
## Total 76 562969 1.0000

SST<- sum(dists^2)/77  
SSW<-cer%>%group\_by(shelf)%>%select(sodium,calories)%>%  
do(d=dist(.[1:2],"euclidean"))%>%ungroup()%>%  
summarize(sum(d[[1]]^2)/20 + sum(d[[2]]^2)/21+ sum(d[[3]]^2)/36)%>%pull  
  
F\_obs<-((SST-SSW)/2)/(SSW/74)   
  
Fs<-replicate(1000,{  
new<-cer%>%mutate(shelf=sample(shelf)) #permute the species vector  
SSW<-new%>%group\_by(shelf)%>%select(sodium,calories)%>%  
do(d=dist(.[1:2],"euclidean"))%>%ungroup()%>%  
summarize(sum(d[[1]]^2)/20 + sum(d[[2]]^2)/21+ sum(d[[3]]^2)/36)%>%pull  
((SST-SSW)/2)/(SSW/74) #calculate new F on randomized data  
})  
{hist(Fs,prob = T); abline(v=F\_obs, col="red", add=T)}

mean(Fs>F\_obs)

## [1] 0.501

Null hypothesis is there is no difference in the mean distance or spread between groups. The alternative hypothesis is that there is a difference in mean distance or spread between groups.The pvalue is very large and so we would fail to reject the null hypothesis.

#Question 3.  
rating\_c <- cer$rating - mean(cer$rating)  
sugars\_c <- cer$sugars - mean(cer$sugars)  
calories\_c<- cer$calories - mean(cer$calories)  
sodium\_c<- cer$sodium - mean(cer$sodium)  
  
  
#linear regresssion  
fit<- lm(rating~calories\_c\*shelf, data = cer)  
summary(fit)

##   
## Call:  
## lm(formula = rating ~ calories\_c \* shelf, data = cer)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.730 -6.198 -1.145 4.365 25.476   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 41.1711 2.0449 20.134 < 2e-16 \*\*\*  
## calories\_c -1.1349 0.2066 -5.493 5.81e-07 \*\*\*  
## shelfMiddle -2.5714 2.7830 -0.924 0.358626   
## shelfTop 4.4053 2.4599 1.791 0.077588 .   
## calories\_c:shelfMiddle -0.2385 0.3076 -0.776 0.440574   
## calories\_c:shelfTop 0.7367 0.2129 3.460 0.000918 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.199 on 71 degrees of freedom  
## Multiple R-squared: 0.6817, Adjusted R-squared: 0.6593   
## F-statistic: 30.41 on 5 and 71 DF, p-value: < 2.2e-16

#linear regression plot  
qplot(x = calories, y = rating, color = shelf, data = cer) +  
stat\_smooth(method = "lm", se = FALSE, fullrange = TRUE)

#checking assumptions  
resids<-fit$residuals; fitvals<-fit$fitted.values  
ggplot()+geom\_point(aes(fitvals,resids))+geom\_hline(yintercept=0, col="red") #checks for linearity and homoskedacity

ggplot()+geom\_histogram(aes(resids),bins=20) #normality

ggplot()+geom\_qq(aes(sample=resids))+geom\_qq\_line(aes(sample=resids), color='red') #normality

#regression results  
coeftest(fit, vcov. = vcovHC(fit))

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 41.17107 2.20473 18.6740 < 2.2e-16 \*\*\*  
## calories\_c -1.13489 0.25952 -4.3731 4.115e-05 \*\*\*  
## shelfMiddle -2.57143 3.03737 -0.8466 0.400063   
## shelfTop 4.40525 2.59364 1.6985 0.093795 .   
## calories\_c:shelfMiddle -0.23854 0.37296 -0.6396 0.524507   
## calories\_c:shelfTop 0.73666 0.27155 2.7128 0.008365 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

* Intercept: Predicted rating for a cereal with calories kept constant controlling for shelf placement is 41.17 points.
* ShelfMiddle: Controlling for calories, rating for the middle shelf group is 2.57 points lower than the bottom shelf on average.
* ShelfTop: Controlling for calories, rating for the top shelf group is 4.4 points higher than the bottom shelf on average.
* calories\_c: There is a decrease of -1.13 rating points for every one unit increase in sugar on average.
* shelfMiddle:calories\_c: The slope for calories on rating is .239 times lower for the middle shelf compared to the bottom shelf.
* shelfTop:calories\_c: The slope for calories on rating is .737 times higher for the top shelf compared to the bottom shelf.
* The original standard errors were 2.78 and 2.46 for the middle and top shelves while the interaction and rating were all around .2. The significant p values were for calories and top shelf. The robust standard errors increased for the middle shelf and slightly decreased for the top shelf but the interaction between top shelf and calories was now significant.
* Based on the adjusted r-squared value of the original model, we can estimate that rating and shelf level can explain about 65% of the variance in sugar per serving of cereal.

#Question 4  
#bootstrap SE  
  
samp\_distn<-replicate(5000, {  
boot\_dat <- sample\_frac(cer, replace=T) #bootstrap your data  
fit <- lm(rating~calories\_c\*shelf, data=boot\_dat) #fit model  
coef(fit) #save coefs  
})  
## Estimated SEs  
samp\_distn %>% t %>% as.data.frame %>% summarize\_all(sd)

## (Intercept) calories\_c shelfMiddle shelfTop calories\_c:shelfMiddle calories\_c:shelfTop  
## 1 3.12321 0.1821155 4.390677 3.81474 0.2539674 0.2162599

After running bootstrap errors, the middle and top shelf increased in value compared to the original as well as the robust errors. The calories st. error got smaller. The interaction between calories and middle shelf and top shelf decreased but only slightly. Since the SEs for shelves got bigger, we can assume that the p values also got bigger for those variables.

#Question 5  
head(cer)

## # A tibble: 6 x 6  
## name mfr calories sodium shelf rating  
## <chr> <fct> <dbl> <dbl> <fct> <dbl>  
## 1 100% Bran 0 70 130 Top 68.4  
## 2 100% Natural Bran 0 120 15 Top 34.0  
## 3 All-Bran 1 70 260 Top 59.4  
## 4 All-Bran with Extra Fiber 1 50 140 Top 93.7  
## 5 Almond Delight 0 110 200 Top 34.4  
## 6 Apple Cinnamon Cheerios 0 110 180 Bottom 29.5

fit3<- glm(mfr~sodium+rating, data = cer, family=binomial(link="logit"))  
coeftest(fit3)

##   
## z test of coefficients:  
##   
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4829901 1.2114207 -2.0497 0.0404 \*  
## sodium 0.0045534 0.0033596 1.3553 0.1753   
## rating 0.0205273 0.0195455 1.0502 0.2936   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

exp(coef(fit3))

## (Intercept) sodium rating   
## 0.0834932 1.0045637 1.0207395

prob<- predict(fit3, type = "response")  
  
#confusion matrix  
table(predict=as.numeric(prob>.5),truth=cer$mfr)%>%addmargins

## truth  
## predict 0 1 Sum  
## 0 54 22 76  
## 1 0 1 1  
## Sum 54 23 77

#accuracy  
(54+1)/77

## [1] 0.7142857

#sensitivity  
(1/23)

## [1] 0.04347826

#specificity  
54/54

## [1] 1

#ppv  
1/1

## [1] 1

#plot densit of log odds  
odds<-function(p)p/(1-p)  
p<-seq(0,1,by=.1)  
cbind(p, odds=odds(p))%>%round(4)

## p odds  
## [1,] 0.0 0.0000  
## [2,] 0.1 0.1111  
## [3,] 0.2 0.2500  
## [4,] 0.3 0.4286  
## [5,] 0.4 0.6667  
## [6,] 0.5 1.0000  
## [7,] 0.6 1.5000  
## [8,] 0.7 2.3333  
## [9,] 0.8 4.0000  
## [10,] 0.9 9.0000  
## [11,] 1.0 Inf

logit<-function(p)log(odds(p))  
cbind(p, odds=odds(p),logit=logit(p))%>%round(4)

## p odds logit  
## [1,] 0.0 0.0000 -Inf  
## [2,] 0.1 0.1111 -2.1972  
## [3,] 0.2 0.2500 -1.3863  
## [4,] 0.3 0.4286 -0.8473  
## [5,] 0.4 0.6667 -0.4055  
## [6,] 0.5 1.0000 0.0000  
## [7,] 0.6 1.5000 0.4055  
## [8,] 0.7 2.3333 0.8473  
## [9,] 0.8 4.0000 1.3863  
## [10,] 0.9 9.0000 2.1972  
## [11,] 1.0 Inf Inf

ggplot()+stat\_function(aes(p),fun=logit,geom="line")+ylab("g(p)=logit(p)")+xlab("p")

> - Intercept: odds of manufacterer being Kellogs when sodium and rating are held constant is .083. > - sodium: controlling for rating, for every one mg increase in sodium, odds of the manufacterer being Kellogs increase by a factor of 1.0046 > - rating: controlling for sodium, for every one unit increase in rating, odds of the manufacterer being Kellogs increase by a factor of 1.0207

# ROC curve  
#install.packages("plotROC")  
library(plotROC)  
  
cer1$mfr<- as.factor(cer1$mfr)  
  
ROCplot<-ggplot(cer1)+geom\_roc(aes(d=mfr,m=prob), n.cuts=0)+  
geom\_segment(aes(x=0,xend=1,y=0,yend=1),lty=2)  
ROCplot

#AUC  
calc\_auc(ROCplot)

## PANEL group AUC  
## 1 1 -1 0.4202899

#needed for 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  
   
 #CALCULATE EXACT AUC  
 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,sens,spec,ppv,auc)  
}

#Question 5 continued  
  
set.seed(1234)  
k=10 #choose number of folds  
data<-cer[sample(nrow(cer)),] #randomly order rows  
folds<-cut(seq(1:nrow(cer)),breaks=k,labels=F) #create folds  
diags<-NULL  
for(i in 1:k){  
## Create training and test sets  
train<-data[folds!=i,]  
test<-data[folds==i,]  
truth<-test$mfr ## Truth labels for fold i  
## Train model on training set (all but fold i)  
fit<-glm(mfr~rating+sodium,data=train,family="binomial")  
## Test model on test set (fold i)  
probs<-predict(fit,newdata = test,type="response")  
## Get diagnostics for fold i  
diags<-rbind(diags,class\_diag(probs,truth))  
}  
summarize\_all(diags,mean)

## acc sens spec ppv auc  
## 1 0.6821429 0 0.9833333 NaN 0.5257143

* The AUC is .45 which would fall below the Bad category.
* The out of sample performance for accuracy is .69, the specificity is .98, and the AUC is .45.

#Question 6  
  
#LASSO  
#install.packages("glmnet")  
library(glmnet)  
#install.packages("plyr")  
library(dplyr)  
  
y<-as.matrix(cer1$mfr) #grab response  
x<-model.matrix(mfr~.,data=cer1)[,-1] #grab predictors  
head(x)

## calories sodium shelfMiddle shelfTop rating  
## 1 70 130 0 1 68.40297  
## 2 120 15 0 1 33.98368  
## 3 70 260 0 1 59.42551  
## 4 50 140 0 1 93.70491  
## 5 110 200 0 1 34.38484  
## 6 110 180 0 0 29.50954

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

## 6 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 8.534898e-01  
## calories .   
## sodium -1.590880e-18  
## shelfMiddle .   
## shelfTop .   
## rating .

set.seed(1234)  
k=10 #choose number of folds  
  
data<-cer[sample(nrow(cer)),] #randomly order rows  
folds<-cut(seq(1:nrow(cer)),breaks=k,labels=F) #create folds  
  
diags<-NULL  
for(i in 1:k){  
 ## Create training and test sets  
 train<-data[folds!=i,]   
 test<-data[folds==i,]  
   
 truth<-test$mfr ## Truth labels for fold i  
   
 ## Train model on training set (all but fold i)  
 fit<-glm(mfr~sodium,data=train,family="binomial")  
   
 ## Test model on test set (fold i)   
 probs<-predict(fit,newdata = test,type="response")  
   
 ## Get diagnostics for fold i  
 diags<-rbind(diags,class\_diag(probs,truth))  
}  
  
  
summarize\_all(diags,mean)

## acc sens spec ppv auc  
## 1 0.6946429 0 1 NaN 0.5385714

The only variable that was retained was the sodium variable. The AUC for question 5 was .45 and the out of sample performance was better as the AUC was .48 and the accuracy went from from .69 to .701. Since these values are higher, we do not assume overfitting. Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.