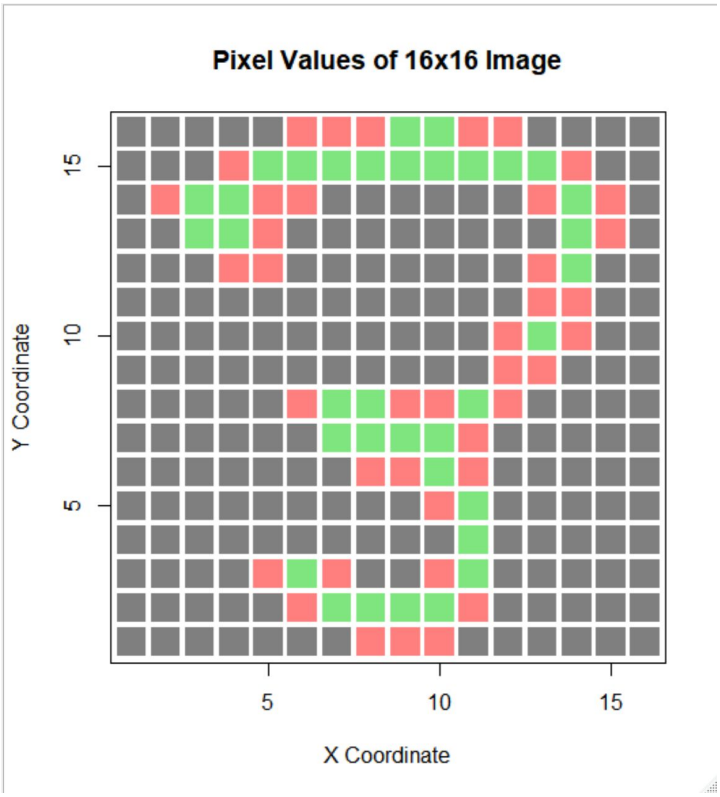
Jisu Kim (jk5nc), Timothy Kim (tsk8va), Olivia Ryu (hr2ad)
STAT 3280-001
Professor Tianxi Li
12 February 2020

Homework 1

Problem 1

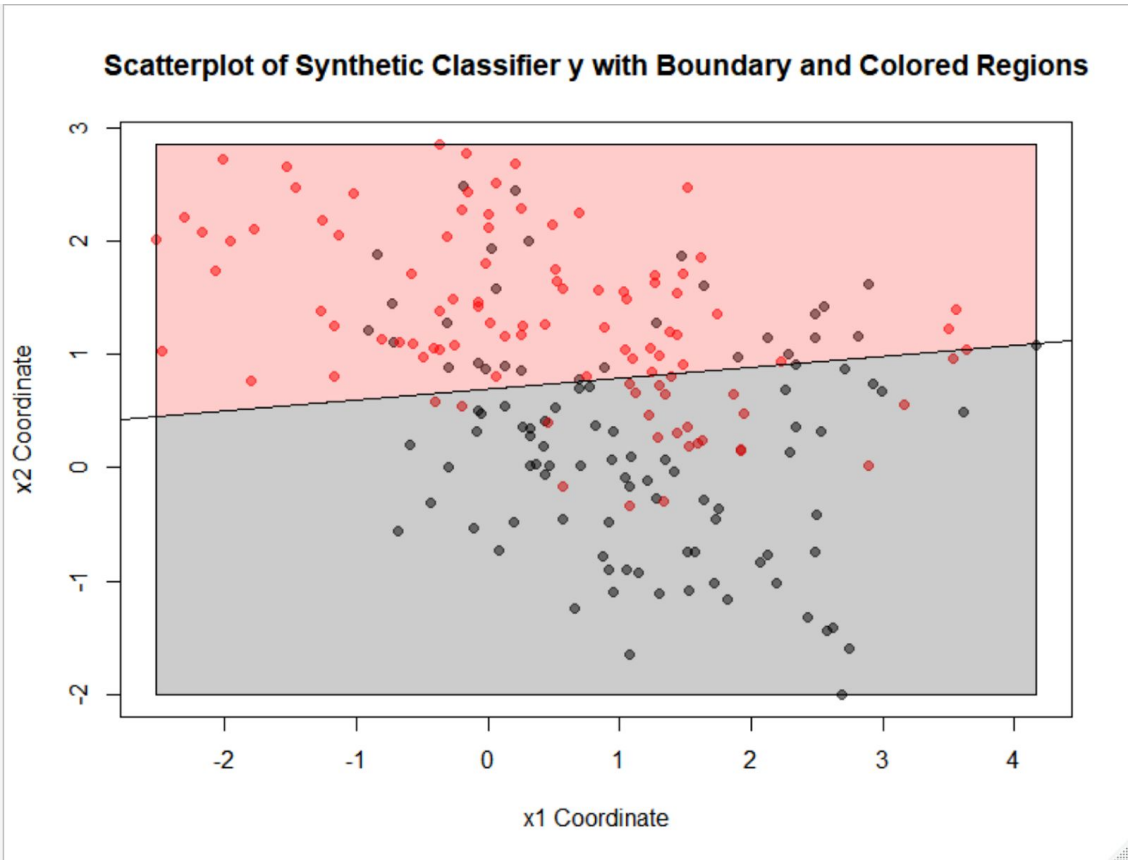


The number appears to be 3. Please see Appendix A for the solution code. This problem was solved by first importing the file "HandWritten.txt" using the scan() function, and splitting the string using the strsplit() function over whitespaces, and then transforming all of the resulting strings into numeric types before storing into a list "pixel.val," which now contains numeric type values of all 256 pixel density values. In order to visualize this data, our group decided to attempt to plot the points and then assign color values to the pixel densities.

Two numeric vectors were created using the rep() function: one which counts from 1 to 16, 16 times, and the other replicating each of 1 to 16, 16 times. By combining these two vectors we would get coordinate points, e.g. (1,1), (1,2), …, (16,16). We additionally added the list pixel.val to this merge to create a three-column data.frame.

Finally, we plotted our points using the plot() function, with the first numeric vector as the x-coord argument, the second as the y-coord, and pixel.val as colors. We used the alpha() command to fulfill the "col" argument in plot() and selected plotting style 15 for a filled square, and cex = 3 to increase the point size.

Problem 2

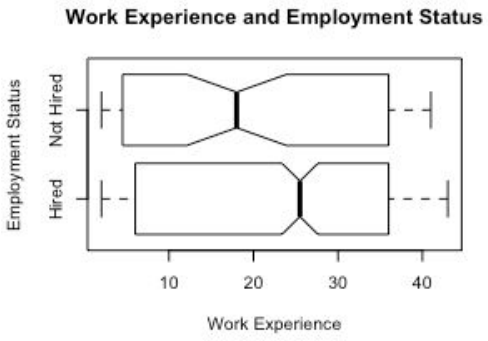


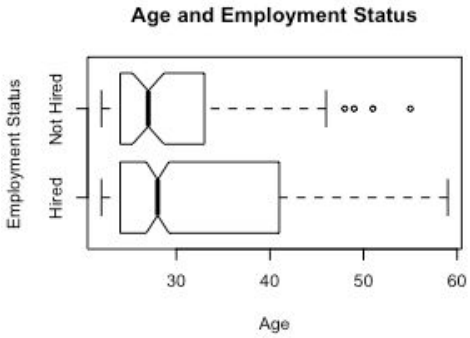
This problem was solved by first importing the file "mixture.csv" using the read.csv() function. We first computed each point's actual classifier value according to the provided classifier, by plugging each entry's x1 and x2 values into the expression. We then plotted the points using the plot() function, with x1 on the x-axis, x2 on the y-axis, and the provided y as the point's color. We again selected style 19 for a filled dot. We additionally used the abline() function to draw the classifier boundary, and hand-computed the intercept and slope values given the expression's coefficients.

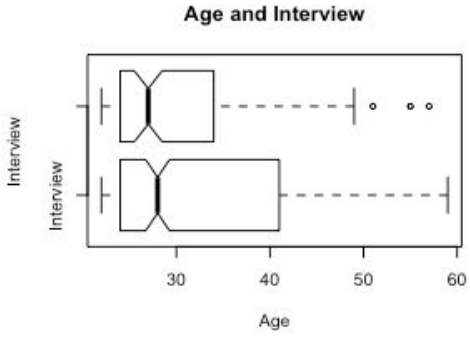
To create the shading for each binary classifier, we decided to use the polygon() function, which requires all endpoints of the desired shape as argument inputs. To achieve this we needed to compute the upper, lower, left, and right bounds for both classifier regions. Therefore, we used the x coordinate of the leftmost point, the y coordinate of the highest point, etc. to find the "four corners" of the scatterplot. We then plugged in the minimum x1 and maximum x1 into the binary classifier expression to find the points along the boundary line where the shading should end.

With these six points stored, the polygon() function was run twice, using the six points as required, and corresponding colors for both boundaries. The adjustcolor() function was used to achieve the actual shading, using the same colors as the points with an alpha.f argument value of only 0.2 to allow sufficient transparency.

Problem 3



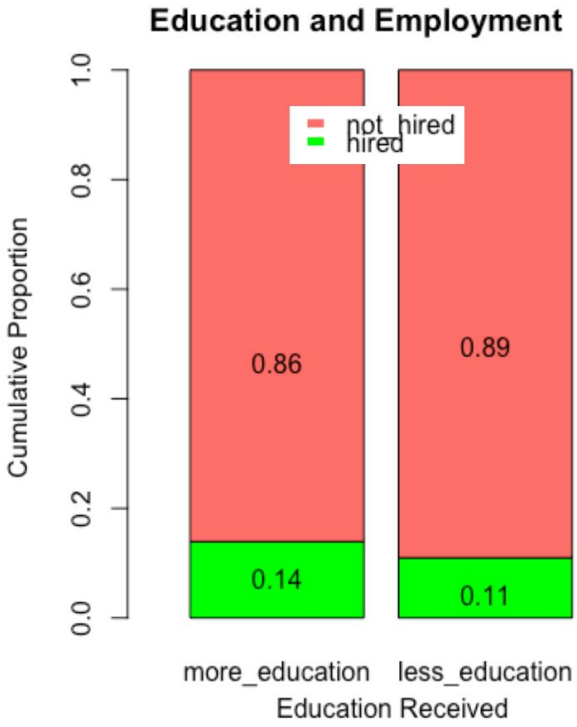




The data set "TeacherHires.csv" was imported with modified column names. We then started the data cleaning process, in which we first repaired seemingly duplicate entries. For example, within the sex variable, we grouped "M" entries with "Male", and "F" entries with "Female." In addition, two new columns, attended.u and attended.g, were created to indicate under-graduate and graduate school attendance; We assumed that an N/A in either the GPA.u or GPA.g column meant that the person did not attend under-graduate or graduate school, respectively. The work variable was transformed into a continuous numeric type as well.

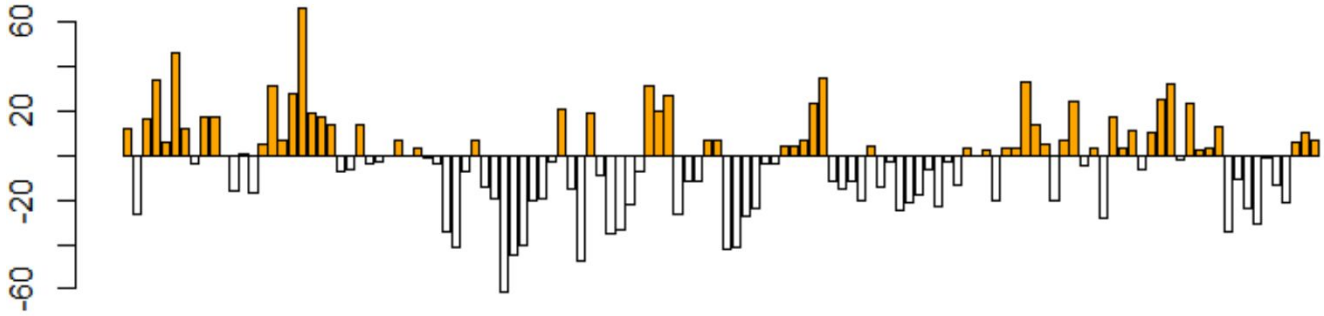
In order to see whether age discrimination was valid, we created four graphs:

- From the Work Experience and Employment Status plot, we can see that there is a clear advantage for those who have more work experience; those who were hired have a higher median work experience, compared to those who were not hired.
- From the Age and Employment Status plot, we can see that those who were hired have a slightly higher median age. This goes against the teacher's claim that she was age-discriminated against.
- From the Age and Interview plot, we can see those who received an interview have a slightly higher median age. This also goes against the claim of age-discrimination.
- Finally, to observe any other potential relationships between hiring decisions and education, we created a stacked bar plot that portrays the proportion of candidates who were hired or not hired on the basis of level of education acquired. It seemed that there was no significant relationship between the two variables.



In conclusion, there is no evidence to claim that the school practices age discrimination in the interviewing and hiring of teachers.

Problem 4

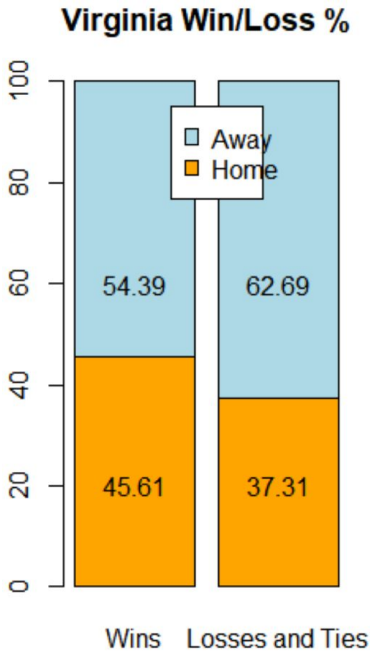


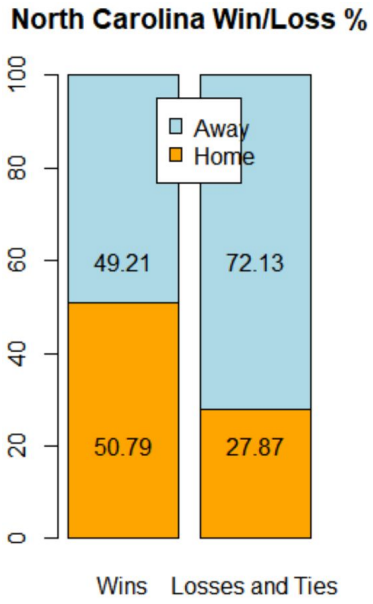
In order to summarize the entire head-to-head history of UVA vs. UNC, we decided to create a bar graph illustrating the point differentials for all 124 games. The point differential was calculated by subtracting UNC's score from UVA's score. Thus, if UVA won the game, there is a positive point differential (as indicated by the orange bars above the x-axis), and if UVA lost the game, there is a negative point differential (as indicated by the white bars below the x-axis).

The rivalry's record of 57-63-4 in favor of UNC was computed by creating a column in the data.frame that lists the winner of the match, with options including "VA," "NC," and "Tie," and then simply tallying the amount of occurrences of each possibility.

One piece of information we were interested in exploring was the alleged sports phenomenon of "home field advantage," which is the idea that a team has a higher chance of winning if they play on their own home field. For example, would the UVA Cavaliers have a higher chance of winning if they played against UNC at Scott Stadium? To quantify this we first created a confusion matrix for each school, with wins and losses on the y-axis and home field or away field on the x-axis. These matrices were plotted as percentage bar graphs as shown below.

From these graphs we can observe that both UVA and UNC have roughly equal percentages of wins and losses on their home field, suggesting that playing on home turf offers no statistically significant advantage. However, both teams suffer lower percentages of wins on foreign fields, defined as either a neutral field (e.g. a field in Richmond) or the opponent's field. This suggests that playing on an unfamiliar field is correlated with higher rates of losing, and may be worth further statistical analysis.





Appendix 1

```
# Problem 1

library(scales)

pixel.val <- as.numeric(strsplit(scan("HandWritten.txt", what="", sep=""),
"[[:space:]]+")) + 2
x.coord <- rep(1:16, 16)
y.coord <- rep(1:16, each=16)
handwritten <- data.frame(x.coord, y.coord, pixel.val)

attach(handwritten)
plot(x.coord, y.coord, col=alpha(pixel.val, 0.5), pch=15, cex=3.5,
main="Pixel Values of 16x16 Image", xlab="X Coordinate", ylab="Y
Coordinate")
```

Appendix 2

```
# Problem 2

library(scales)

mixture <- read.csv("mixture.csv", header=TRUE)
mixture$class = -0.978 - 0.134*mixture$x1 + 1.398*mixture$x2
attach(mixture)
plot(x1, x2, col=alpha(y+1, 0.5), pch=19, main="Scatterplot of Synthetic
Classifier y with Boundary and Colored Regions", xlab="x1 Coordinate",
ylab="x2 Coordinate")
abline(a=0.978/1.398, b=0.134/1.398)

max.x1 <- max(mixture$x1)
min.x1 <- min(mixture$x1)
max.x2 <- max(mixture$x2)
min.x2 <- min(mixture$x2)
line.min <- (0.134 * min.x1 + 0.978)/1.398
line.max <- (0.134 * max.x1 + 0.978)/1.398

polygon(c(min.x1, max.x1, max.x1, min.x1), c(max.x2, max.x2, line.max,
line.min), col=adjustcolor("red",alpha.f=0.2))
polygon(c(min.x1, max.x1, max.x1, min.x1), c(line.min, line.max, min.x2,
min.x2), col=adjustcolor("black",alpha.f=0.2))
```

Appendix 3

```
# Problem 3

library(YaleToolkit)
library(scales)

hires <- read.csv("TeacherHires.csv", header=TRUE)
hires <- hires[, -which(whatis(hires)$missing==nrow(hires))]
names(hires) <- c("interview", "hired", "app.date",
                  "age", "sex", "residence", "GPA.u",
                  "GPA.g", "MA", "sub",
                  "teaching", "work", "work.kids",
                  "volunteer")
whatis(hires, type.truncate = 4)

# Cleaning up sex variable
hires$sex <- as.character(hires$sex)
unique(hires$sex)
levels(hires$sex)
hires$sex[hires$sex=="  "] <- NA
hires$sex[hires$sex=="M"] <- "Male"
hires$sex[hires$sex=="F"] <- "Female"
hires$sex <- factor(hires$sex)

# Cleaning up age variable
hires$age <- as.numeric(as.character(hires$age))

hires$agegroup <- factor(hires$age<=39, levels=c(FALSE,TRUE),
                         labels=c("older", "younger"))

# Cleaning up damaged entries
hires$interview[hires$interview=="yes "] <- "yes"
hires$interview <- factor(hires$interview)

hires$MA[hires$MA=="yes "] <- "yes"
hires$MA <- factor(hires$MA)

hires$sub[hires$sub=="no "] <- "no"
hires$sub[hires$sub==""] <- NA
```

```
hires$sub <- factor(hires$sub)

hires$teaching[hires$teaching=="yes "] <- "yes"
hires$teaching[hires$teaching==""] <- NA
hires$teaching <- factor(hires$teaching)

#clean up hired column
levels(hires$hired)
hires$hired[hires$hired==""] <- NA
hires$hired[hires$hired=="yes "] <- "yes"
hires$hired[hires$hired=="yes*"] <- "yes"
hires$hired <- factor(hires$hired)

#clean up workkids column
levels(hires$work.kids)
hires$work.kids[hires$work.kids=="no "] <- "no"
hires$work.kids[hires$work.kids==""] <- NA
hires$work.kids = factor(hires$work.kids)

#clean up volunteer column
levels(hires$volunteer)
hires$volunteer[hires$volunteer==""] <- NA
hires$volunteer = factor(hires$volunteer)

# Creating a new column for undergrad attendance
hires$attended.u <- hires$GPA.u != "N/A"
hires$attended.g <- hires$GPA.g != "N/A"



#box-and-whisker plot to see relationships

par(mfrow=c(2,2))

# work & hired
boxplot( as.numeric(hires$work)~as.numeric(hires$hired), main="Work
Experience and Employment Status",
         xlab="Work Experience",
         ylab="Employment Status",
```

```
         at = c(1,2),
         names = c("Hired","Not Hired"),
         horizontal = TRUE,
         notch = TRUE)

# age & hired
boxplot( as.numeric(hires$age)~as.numeric(hires$hired), main="Age and
Employment Status",
         xlab="Age",
         ylab="Employment Status",
         at = c(1,2),
         names = c("Hired","Not Hired"),
         horizontal = TRUE,
         notch = TRUE)

# age & interview
boxplot( as.numeric(hires$age)~as.numeric(hires$interview), main="Age and
Interview",
         xlab="Age",
         ylab="Interview",
         at = c(1,2),
         names = c("Interview","No Interview"),
         horizontal = TRUE,
         notch = TRUE)


par(mfrow=c(1,1))

# stacked bar graph to analyze relationship between employment and education

# Creating a new data.frame for college attendance
hires$higherEd <- hires$MA == "yes" | hires$attended.g == TRUE
graduates <- hires[which(hires$higherEd == TRUE),]
hires$hired <- ifelse(hires$hired=="yes",1,0)

#hired & education
relevant <- data.frame(hires$hired, as.numeric(hires$higherEd))
colnames(relevant)[1] <- "hired"
colnames(relevant)[2] <- "higherEd"
```

```
education <- sum(relevant$higherEd, na.rm=TRUE)
less_education <- length(relevant$higherEd)-education

edu_nothired <- nrow(subset(relevant, higherEd==1 & hired == 0))
edu_hired <- nrow(subset(relevant, higherEd==1 & hired == 1))

edu_nothired_prop <- edu_nothired/education
edu_hired_prop <- edu_hired/education

ledu_nothired <- nrow(subset(relevant, higherEd==0 & hired==0))
ledu_hired <- nrow(subset(relevant, higherEd==0 & hired==1))

ledu_nothired_prop <- ledu_nothired/less_education
ledu_hired_prop <- ledu_hired/less_education

data.perc <- data.frame(
  row.names =c("more_education", "less_education"),
  hired =c(edu_hired_prop,ledu_hired_prop),
  not_hired =c(edu_nothired_prop, ledu_nothired_prop))

x <- barplot(t(data.perc),
             main = "Education and Employment",
             col=c("green", "salmon"),
             legend=TRUE, border="black", xlim=c(0,4), args.legend=
               list(bty="n", border=NA),
             ylab="Cumulative Proportion", xlab="Education Received")
text(x, data.perc$hired, labels=round(data.perc$hired,digits=2),
adj=c(0.5,2.5), col="black")
text(x, data.perc$not_hired, labels=round(data.perc$not_hired,digits=2),
adj=c(0.5,12), col="black")
```

Appendix 4

```
# Problem 4

results <- read.csv("OldestRivalry.csv", header=TRUE)
results$NC.score = as.numeric(results$NC.score)
results$VA.score = as.numeric(results$VA.score)

results$score.diff = results$VA.score - results$NC.score
results$outcome <- factor(sign(results$score.diff), levels=c(1, -1, 0),
labels=c("VA", "NC", "Tie"))

results <- transform(results, win.score = pmax(NC.score, VA.score))
results <- transform(results, lose.score = pmin(NC.score, VA.score))

head(results)

# Current rivalry record
VA.wins <- sum(results$outcome == "VA")
NC.wins <- sum(results$outcome == "NC")
ties <- sum(results$outcome == "Tie")
c(VA.wins, NC.wins, ties)

# Score differential bargraph
par(mfrow=c(1,1))
mycols = c("white", "black", "orange")
barplot(results$score.diff, col=mycols[sign(results$score.diff)+2])

# Home field advantage computation
par(mfrow=c(1,2))
# Counting ties as loss
results$venue <- factor(results$city, levels=c("Atlanta", "Chapel Hill",
"Charlottesville", "Norfolk", "Richmond"), labels=c("Neutral", "NC.Home",
"VA.Home", "Neutral", "Neutral"))
VA.loss <- 124 - VA.wins
NC.loss <- 124 - NC.wins

# For VA
VA.home.wins <- sum(results$outcome == "VA" & results$venue == "VA.Home")
VA.home.loss <- sum(results$outcome != "VA" & results$venue == "VA.Home")
```

```
VA.away.wins <- sum(results$outcome == "VA" & results$venue != "VA.Home")
VA.away.loss <- sum(results$outcome != "VA" & results$venue != "VA.Home")

VA.matrix <- matrix(c(VA.home.wins, VA.away.wins, VA.home.loss,
VA.away.loss), nrow=2, ncol=2)
colnames(VA.matrix) <- c("Wins", "Losses and Ties")
rownames(VA.matrix) <- c("Home", "Away")

VA.matrix.percent <- apply(VA.matrix, 2, function(x){x*100/sum(x,na.rm=T)})

VA.barplot <- barplot(VA.matrix.percent, col=c("Orange", "Light Blue"),
main="Virginia Win/Loss %")
legend(1, 95, legend=c("Away", "Home"), fill=c("Light Blue", "Orange"))
text(VA.barplot, 20, labels=round(VA.matrix.percent[1,], digits=2))
text(VA.barplot, 60, labels=round(VA.matrix.percent[2,], digits=2))

# For NC
NC.home.wins <- sum(results$outcome == "NC" & results$venue == "NC.Home")
NC.home.loss <- sum(results$outcome != "NC" & results$venue == "NC.Home")
NC.away.wins <- sum(results$outcome == "NC" & results$venue != "NC.Home")
NC.away.loss <- sum(results$outcome != "NC" & results$venue != "NC.Home")

NC.matrix <- matrix(c(NC.home.wins, NC.away.wins, NC.home.loss,
NC.away.loss), nrow=2, ncol=2)
colnames(NC.matrix) <- c("Wins", "Losses and Ties")
rownames(NC.matrix) <- c("Home", "Away")

NC.matrix.percent <- apply(NC.matrix, 2, function(x){x*100/sum(x,na.rm=T)})

NC.barplot <- barplot(NC.matrix.percent, col=c("Orange", "Light Blue"),
main="North Carolina Win/Loss %")
legend(1, 95, legend=c("Away", "Home"), fill=c("Light Blue", "Orange"))
text(NC.barplot, 20, labels=round(NC.matrix.percent[1,], digits=2))
text(NC.barplot, 60, labels=round(NC.matrix.percent[2,], digits=2))
```