INFO 511 Homework 3

Scott Gillespie

April 3, 2016

Establish working directory, call required libraries, and read in raw data via convereted CSV files from SAS datasets using the "fread" function from the "data.table" package.

setwd("C:/Users/sgille3/Projects/Course Work/INFO 511 - Analytics/Homework 3/")  
library(dplyr)  
library(data.table)  
library(ggplot2)  
library(gtable)  
library(grid)  
grid.newpage()  
  
head(pear<-fread("pearSG.csv"))  
head(pearDemo<-fread("peardemoSG.csv"))  
str(pear)  
str(pearDemo)

Merge dataframes and calculate averages and standard errors for SBP and DBP by hour and race (AA vs. CC) using functions via the "dplyr" package. Two separate dplyr dataframes are created for the purposes of specifying race by SBP and DBP variables. This facilitates legend creation while merging two indepent ggplots.

full1 <- inner\_join(pear,pearDemo,"pear\_id") %>%  
 group\_by(hour,aa) %>%  
 summarise(meanSBP = mean(sbp), meanDBP = mean(dbp),  
 sdSBP = sd(sbp), sdDBP = sd(dbp), n=n()) %>%  
 mutate(seSBP = sdSBP/sqrt(n), seDBP = sdDBP/sqrt(n),  
 race=ifelse(aa==1,"SBP AA","SBP CC")) %>%  
 filter(race != "<NA>") %>%  
 select(hour,meanSBP,seSBP, n, hour,race) %>%  
 arrange(hour)  
head(full1)  
  
full2 <- inner\_join(pear,pearDemo,"pear\_id") %>%  
 group\_by(hour,aa) %>%  
 summarise(meanSBP = mean(sbp), meanDBP = mean(dbp),  
 sdSBP = sd(sbp), sdDBP = sd(dbp), n=n()) %>%  
 mutate(seSBP = sdSBP/sqrt(n), seDBP = sdDBP/sqrt(n),  
 race=ifelse(aa==1,"DBP AA","DBP CC")) %>%  
 filter(race != "<NA>") %>%  
 select(hour,meanDBP,seDBP, n, hour,race) %>%  
 arrange(hour)  
head(full2)

ggplot creation by outcome (SBP and DBP) followed by some fancy code (tip of the hat to Hadley W.) that merges plots for the final figure.

###appears to be 2 StdErrs in the plot  
pd=position\_dodge(0.2)  
plot1 <- ggplot(full1, aes(x=hour, y=meanSBP, group=race, colour=race)) +  
 geom\_errorbar(position=pd, width=0.8, size=0.6,   
 aes(ymin=meanSBP-2\*seSBP,ymax=meanSBP+2\*seSBP)) +  
 scale\_color\_manual(values=c("blue","red")) +  
 geom\_line(position=pd, aes(linetype=race)) +  
 scale\_linetype\_manual(values=c(2,1)) +  
 geom\_point(position=pd) +   
 coord\_cartesian(ylim=c(60,150), xlim=c(0:23)) +  
 scale\_y\_continuous(breaks = seq(60,150,by=10)) +  
 scale\_x\_continuous(breaks = 0:23\*1) +  
 theme\_bw() +  
 labs(x="Hour", y="Blood Pressure (mmHg)") +  
 annotate(geom="segment", y=seq(60,150,2), yend = seq(60,150,2), x=-1.15, xend= -0.95) +  
 theme(axis.title = element\_text(color="#000000", face="bold", size=18),  
 axis.text.x = element\_text(face="bold", color="#000000", size=16),  
 axis.text.y = element\_text(face="bold", color="#000000", size=16),  
 axis.ticks = element\_line(size = 1.5),  
 legend.text = element\_text(face="bold", color="#000000", size=16),  
 legend.title= element\_blank(), legend.position="bottom",  
 panel.border = element\_rect(linetype = "solid", colour = "black", size = 1.25),  
 panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),  
 panel.background = element\_blank(), axis.line = element\_line(colour = "black"))  
  
plot2 <- ggplot(full2, aes(x=hour, y=meanDBP, group=race, colour=race)) +  
 geom\_errorbar(position=pd, width=0.8, size=0.6,   
 aes(ymin=meanDBP-2\*seDBP,ymax=meanDBP+2\*seDBP)) +  
 scale\_color\_manual(values=c("blue","red")) +  
 geom\_line(position=pd, aes(linetype=race)) +  
 scale\_linetype\_manual(values=c(2,1)) +  
 geom\_point(position=pd) +   
 coord\_cartesian(ylim=c(60,150), xlim=c(0:23)) +  
 scale\_y\_continuous(breaks = seq(60,150,by=10)) +  
 scale\_x\_continuous(breaks = 0:23\*1) +  
 theme\_bw() +  
 labs(x="Hour", y="Blood Pressure (mmHg)") +  
 annotate(geom="segment", y=seq(60,150,2), yend = seq(60,150,2), x=-1.15, xend= -0.95) +  
 theme(axis.title = element\_text(color="#000000", face="bold", size=18),  
 axis.text.x = element\_text(face="bold", color="#000000", size=16),  
 axis.text.y = element\_text(face="bold", color="#000000", size=16),  
 axis.ticks = element\_line(size = 1.5),  
 legend.text = element\_text(face="bold", color="#000000", size=16),  
 legend.title= element\_blank(),legend.position="bottom",  
 panel.border = element\_rect(linetype = "solid", colour = "black", size = 1.25),  
 panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),  
 panel.background = element\_blank(), axis.line = element\_line(colour = "black"))  
  
# extract gtable  
gTable1 <- ggplot\_gtable(ggplot\_build(plot1))  
gTable2 <- ggplot\_gtable(ggplot\_build(plot2))  
  
# overlap the panel of 2nd plot on that of 1st plot  
pp <- c(subset(gTable1$layout, name == "panel", se = t:r))  
g <- gtable\_add\_grob(gTable1, gTable2$grobs[[which(gTable2$layout$name == "panel")]], pp$t,   
 pp$l, pp$b, pp$l)  
  
# axis tweaks  
ia <- which(gTable2$layout$name == "axis-l")  
ga <- gTable2$grobs[[ia]]  
ax <- ga$children[[2]]  
ax$widths <- rev(ax$widths)  
ax$grobs <- rev(ax$grobs)  
ax$grobs[[1]]$x <- ax$grobs[[1]]$x - unit(1, "npc") + unit(0.15, "cm")  
g <- gtable\_add\_cols(g, gTable2$widths[gTable2$layout[ia, ]$l], length(g$widths) - 1)  
  
# extract legends  
leg1 <- gTable1$grobs[[which(gTable1$layout$name == "guide-box")]]  
leg2 <- gTable2$grobs[[which(gTable2$layout$name == "guide-box")]]  
  
g$grobs[[which(g$layout$name == "guide-box")]] <-   
 gtable:::cbind\_gtable(leg1, leg2, "first")  
  
# draw it  
grid.draw(g)

