EPID674 Epidemiologic Data Analysis using R

Graphing in R

Kelly Bakulski

Last compiled on November 10, 2020

# Set up options

# Install new packages

# Install packages. Do this only once.  
options(repos="https://cran.rstudio.com" )  
install.packages("epiDisplay")  
install.packages("Hmisc")  
install.packages("ggplot2")  
install.packages("GGally")  
install.packages("vioplot")  
install.packages("forestplot")  
  
# To avoid installing every time: change set up in curly brackets to eval=FALSE

# Load packages

# Specify file directories

#directory <- "/cloud/project/" #Class option when coding on RStudio Cloud  
  
# Three alternate options to customize and specify source data directory if programming on a personal computer (only need to do the one that works for you)  
# directory<-"M:/EPID674/Data/" # Alternate option 1: location of the parent dataset in afs space of school computer  
 directory <- "/Volumes/GoogleDrive/My Drive/Teaching/EPID674/2020\_fall/EPID674\_Week5\_Class/" # Alternate option 2: location of data on my Macbook  
# directory<-"C:/Users/bakulski/Documents/EPID674/Data/" #Alternate option 3: location of data on a personal PC  
  
date <- format(Sys.Date(), "%Y%m%d")

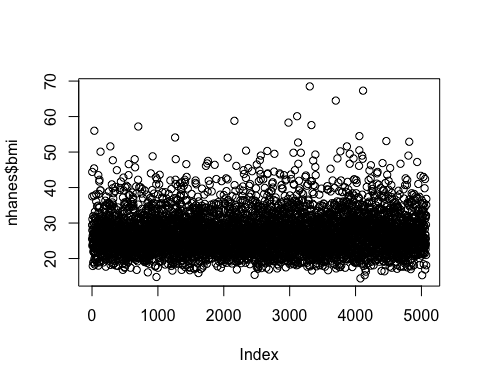
# Load data, remake useful variables

# Check the file path  
paste0(directory, "nhanes3.rda")

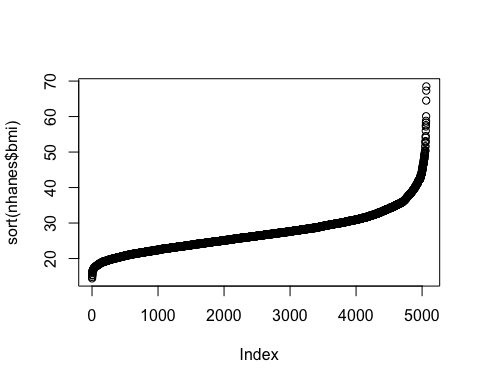
## [1] "/Volumes/GoogleDrive/My Drive/Teaching/EPID674/2020\_fall/EPID674\_Week5\_Class/nhanes3.rda"

# Load the saved R data  
load(paste0(directory, "nhanes3.rda"))  
### Or load RData if you saved the workspace image  
# load(".RData")  
  
  
# Remake a few variables from last class if they are no longer in your environment  
sex1 <- factor(nhanes$sex, levels = c(1, 2), labels = c("male", "female"))  
AGE5b <- cut(nhanes$age, quantile(nhanes$age, c(0, .2, .4, .6, .8, 1)), include.lowest = T) # quintiles  
AGE5c <- cut(nhanes$age, breaks = c(19, 40, 50, 60, 70, 90))  
age5c <- unclass(AGE5c)  
nhanes <- cbind(nhanes, sex1, AGE5b, AGE5c, age5c)

### Basic plotting functions  
  
plot(nhanes$bmi)



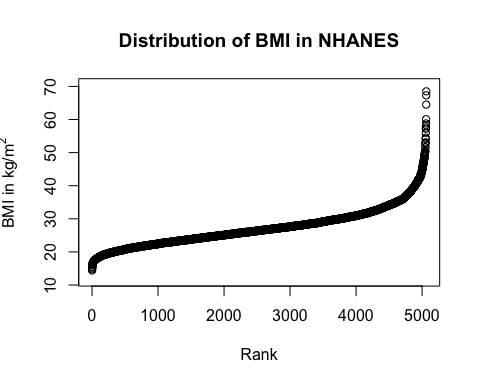
plot(sort(nhanes$bmi))



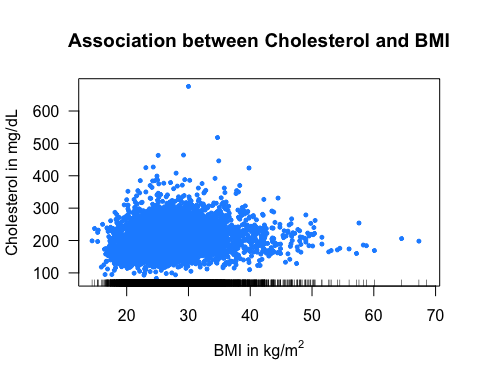
summary(nhanes$bmi)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 14.40 23.20 26.40 27.24 30.12 68.50 10

plot(sort(nhanes$bmi), ylim = c(12, 70), xlab = "Rank", ylab = expression(paste("BMI in kg/m"^"2")), main = "Distribution of BMI in NHANES")



plot(nhanes$bmi, nhanes$chol, pch = 20, cex = 0.8, ylab = "Cholesterol in mg/dL", xlab = expression(paste("BMI in kg/m"^"2")), main = "Association between Cholesterol and BMI", las = 1, col = "dodgerblue")  
rug(nhanes$bmi, side = 1)



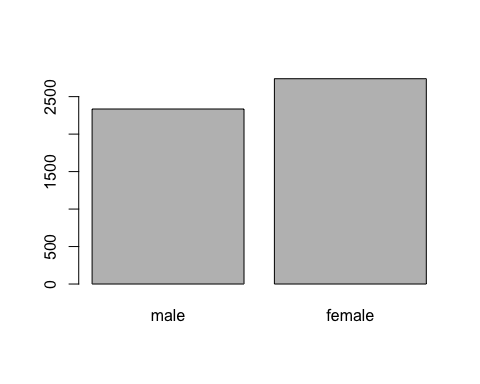
# If you want your plot to appear in a separate window  
# quartz() #mac only  
# x11() #pc or mac (kind of slow)  
  
# If you want to save your plot directly to file, use pdf(), png(), jpg(), etc. Close the plotting with dev.off()  
pdf(file = paste0(directory, "BMI\_plot.pdf"))  
plot(nhanes$bmi, nhanes$chol, pch = 20, cex = 0.8, ylab = "Cholesterol in mg/dL", xlab = expression(paste("BMI in kg/m"^"2")), main = "Association between Cholesterol and BMI", las = 1, col = "dodgerblue")  
rug(nhanes$bmi, side = 1)  
dev.off()

## quartz\_off\_screen   
## 2

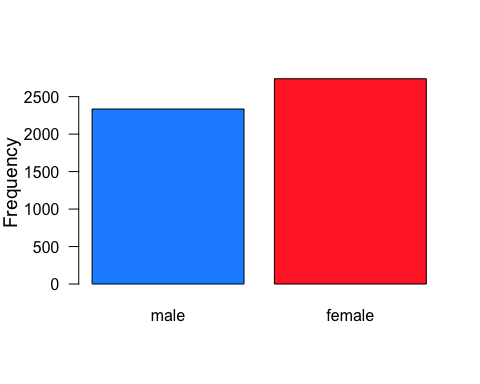
# barplot, base graphics  
table(nhanes$sex1)

##   
## male female   
## 2335 2739

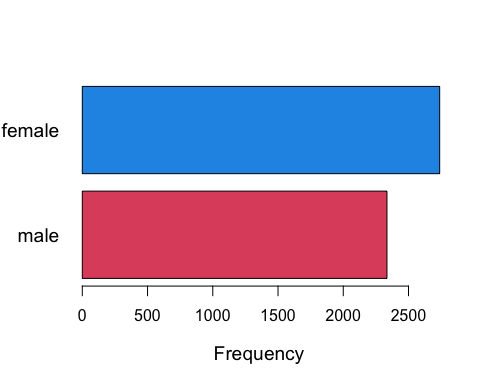
barplot(table(nhanes$sex1))



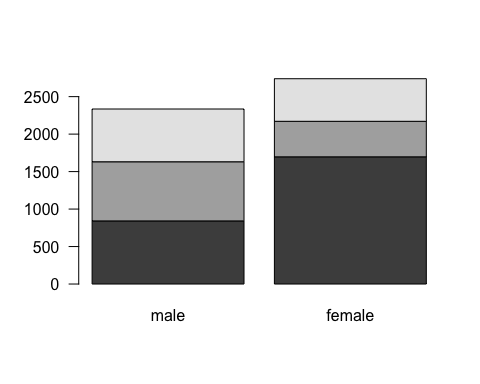
barplot(table(nhanes$sex1), col = c("dodgerblue", "firebrick1"), las = 1, ylab = "Frequency", cex.lab = 1.2)



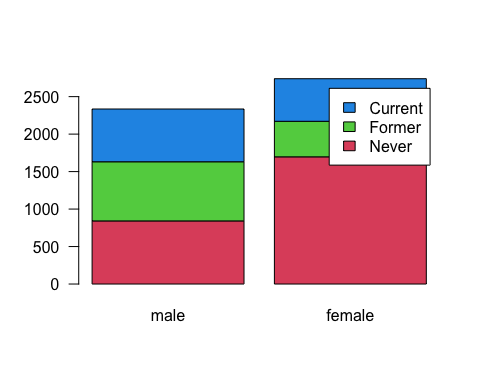
# horizontal  
barplot(table(nhanes$sex1), col = c(2, 4), horiz = T, las = 1, cex.names = 1.2, xlab = "Frequency", cex.lab = 1.2)



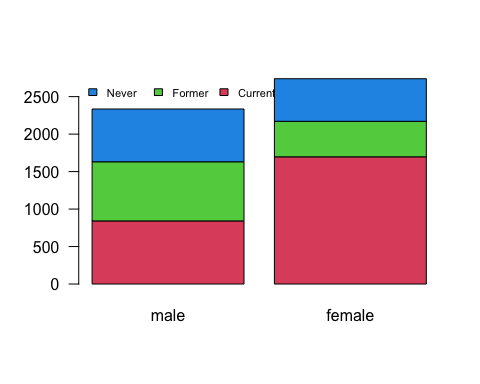
# barplots for subgroups  
barplot(table(nhanes$smk, nhanes$sex1), las = 1)



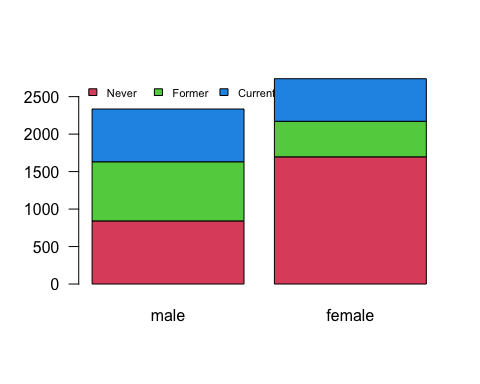
# add legends  
barplot(table(nhanes$smk, nhanes$sex1), col = c(2:4), legend = c("Never", "Former", "Current"), las = 1)



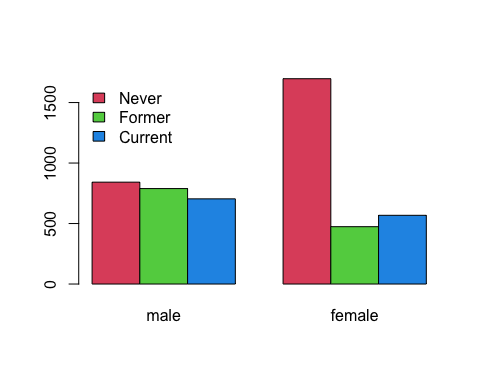
barplot(table(nhanes$smk, nhanes$sex1), col = c(2:4), legend.text = TRUE, args.legend = list(x = "topleft", legend = c("Never", "Former", "Current"), bty = "n", ncol = 3, cex = 0.7), las = 1)



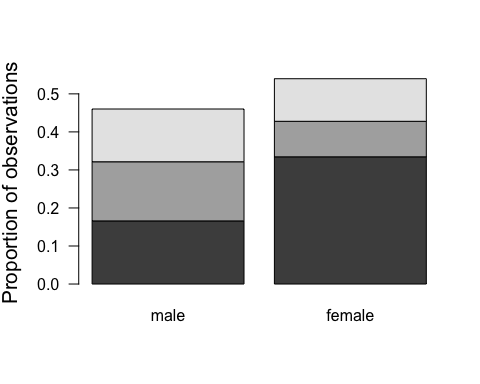
barplot(table(nhanes$smk, nhanes$sex1), col = c(2:4), las = 1)  
legend(x = "topleft", legend = c("Never", "Former", "Current"), bty = "n", ncol = 3, cex = 0.7, fill = c(2:4))



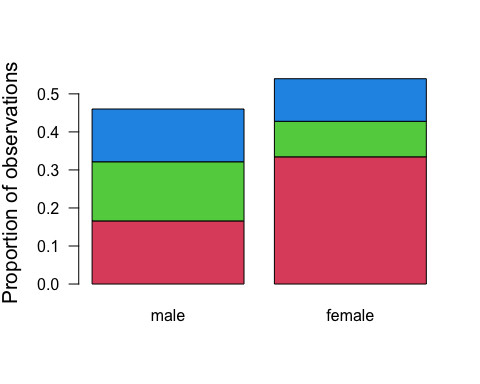
barplot(table(nhanes$smk, nhanes$sex1), col = c(2:4), legend = levels(nhanes$smk), beside = T)  
legend(x = "topleft", legend = c("Never", "Former", "Current"), bty = "n", fill = c(2:4))



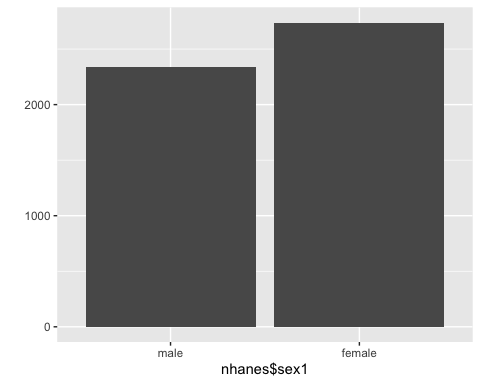
# y-axis as proportion  
barplot(prop.table(table(nhanes$smk, nhanes$sex1)), las = 1, ylab = "Proportion of observations", cex.lab = 1.3)



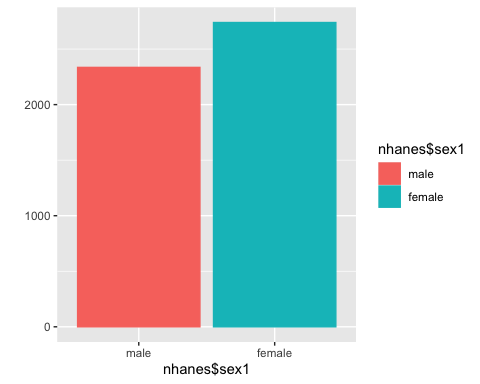
barplot(prop.table(table(nhanes$smk, nhanes$sex1)), col = c(2:4), las = 1, ylab = "Proportion of observations", cex.lab = 1.3)



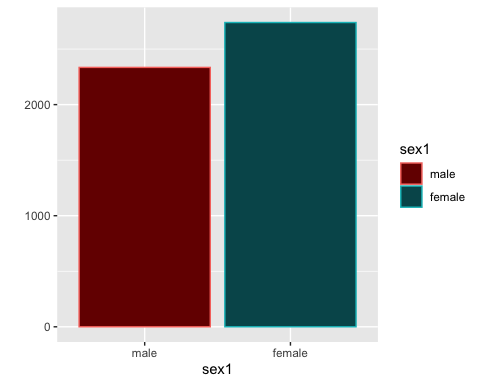
# barplot, ggplot2 package  
  
# using qplot()  
qplot(nhanes$sex1, geom = "bar")



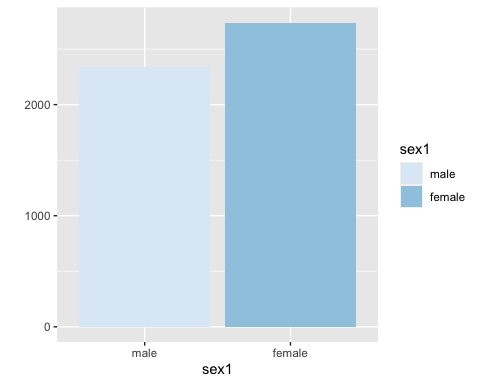
qplot(nhanes$sex1, fill = nhanes$sex1, color = nhanes$sex1, geom = "bar")



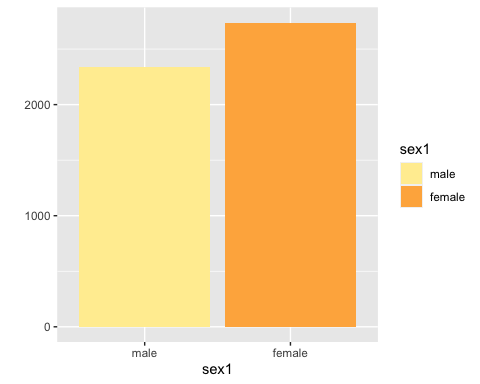
### Lets try to change the colors  
## 1. By changing luminance (default=65)  
qplot(sex1, fill = sex1, color = sex1, geom = "bar") + scale\_fill\_hue(l = 20)



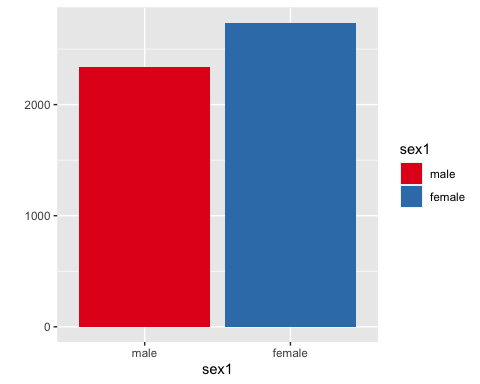
## 2. By changing RColorBrewer  
qplot(sex1, fill = sex1, geom = "bar") + scale\_fill\_brewer(palette = 1)



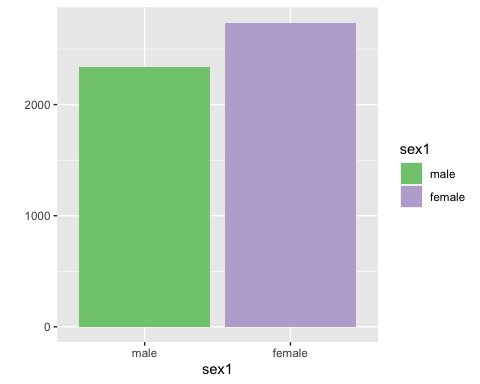
qplot(sex1, fill = sex1, geom = "bar") + scale\_fill\_brewer(palette = 18)



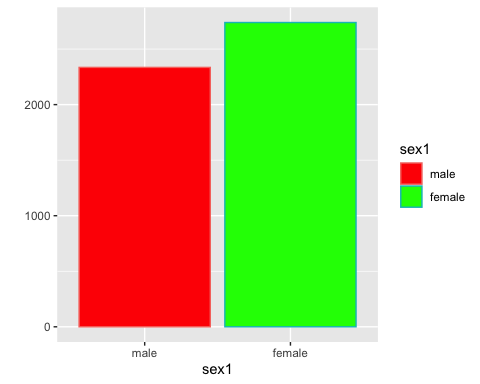
qplot(sex1, fill = sex1, geom = "bar") + scale\_fill\_brewer(palette = "Set1")



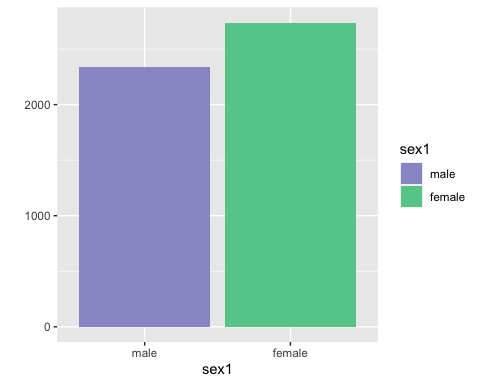
qplot(sex1, fill = sex1, geom = "bar") + scale\_fill\_brewer(palette = "Accent")



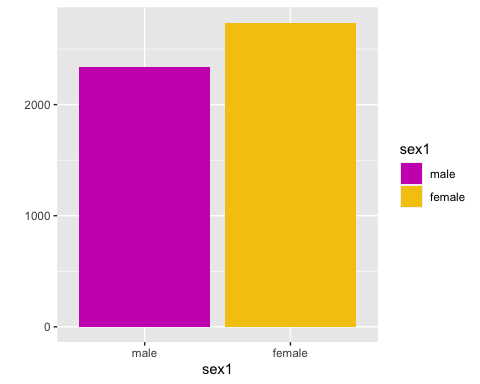
## 3. By manually defining colors  
qplot(sex1, fill = sex1, color = sex1, geom = "bar") + scale\_fill\_manual(values = c("red", "green"))



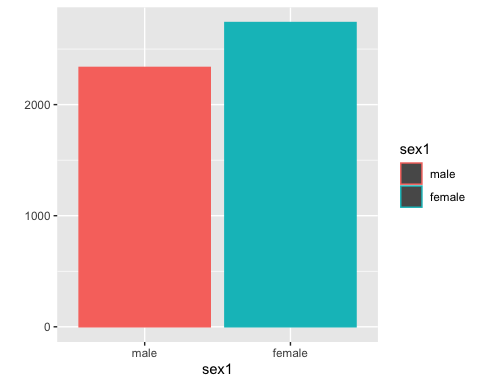
qplot(sex1, fill = sex1, geom = "bar") + scale\_fill\_manual(values = c("#9999CC", "#66CC99"))



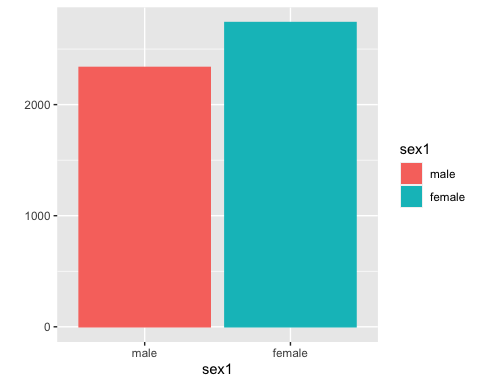
qplot(sex1, fill = sex1, geom = "bar") + scale\_fill\_manual(values = c(6, 7)) + scale\_color\_manual(values = c(1, 2))



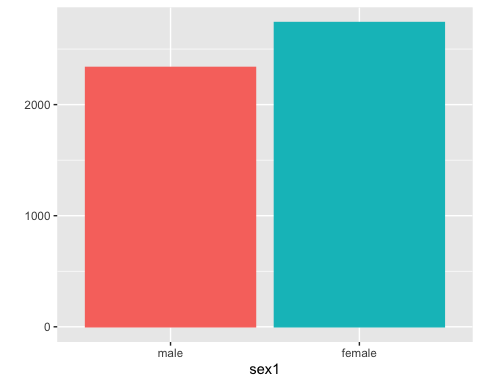
## suppress legend  
qplot(sex1, fill = sex1, color = sex1, geom = "bar") + guides(fill = F)



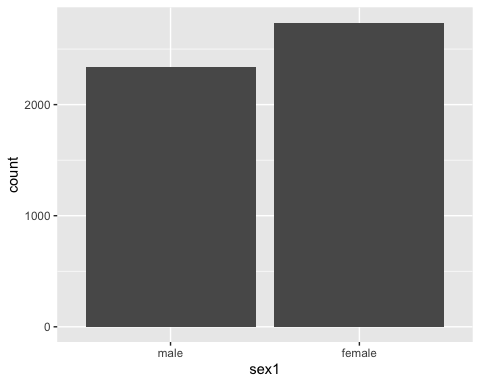
qplot(sex1, fill = sex1, color = sex1, geom = "bar") + guides(color = F)



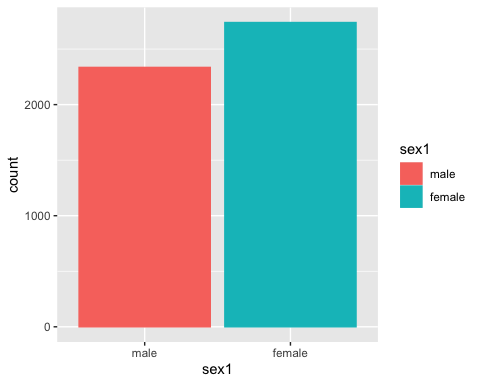
qplot(sex1, fill = sex1, color = sex1, geom = "bar") + guides(fill = F) + guides(color = F)



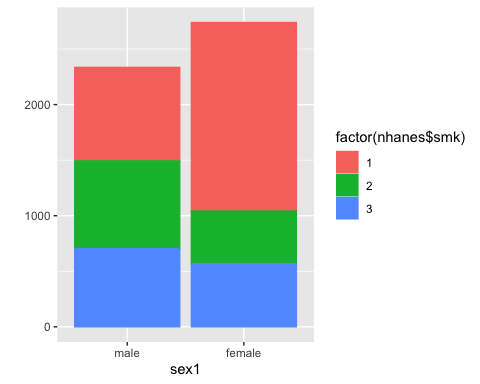
# the same graphs can be made using ggplot()  
ggplot(nhanes, aes(sex1)) + geom\_bar()



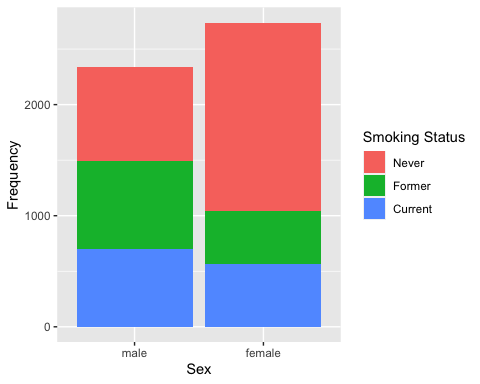
ggplot(nhanes, aes(sex1, fill = sex1, color = sex1)) + geom\_bar()



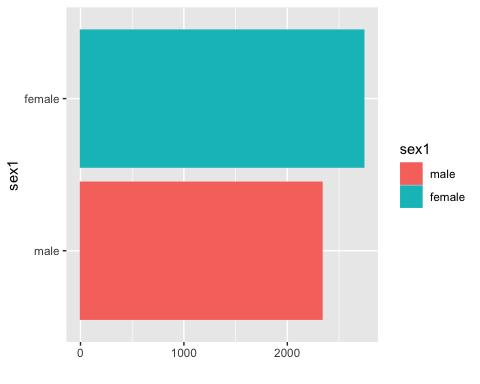
# barplots for subgroups  
qplot(sex1, fill = factor(nhanes$smk), color = factor(nhanes$smk), geom = "bar")



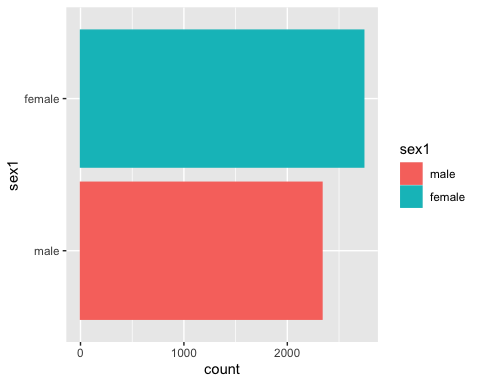
qplot(sex1, fill = factor(nhanes$smk), geom = "bar", xlab = "Sex", ylab = "Frequency") + scale\_fill\_discrete(name = "Smoking Status", breaks = c(1, 2, 3), labels = c("Never", "Former", "Current"))



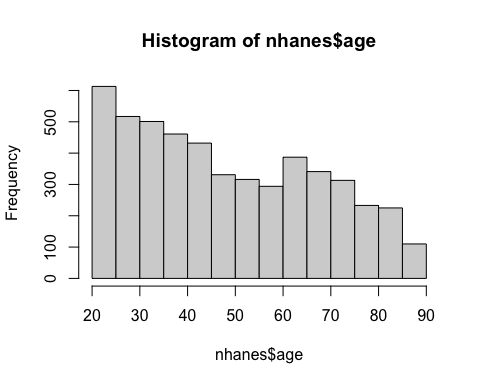
# horizontal  
qplot(sex1, fill = sex1, color = sex1, geom = "bar") + coord\_flip()



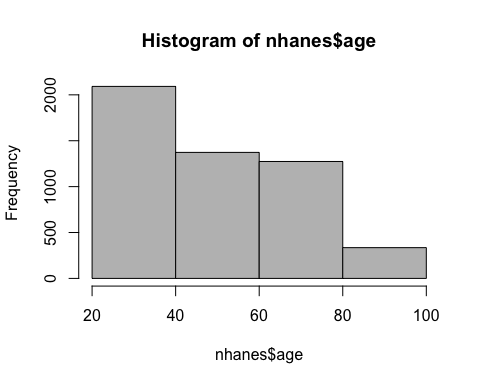
ggplot(nhanes, aes(sex1, fill = sex1, color = sex1)) + geom\_bar() + coord\_flip()



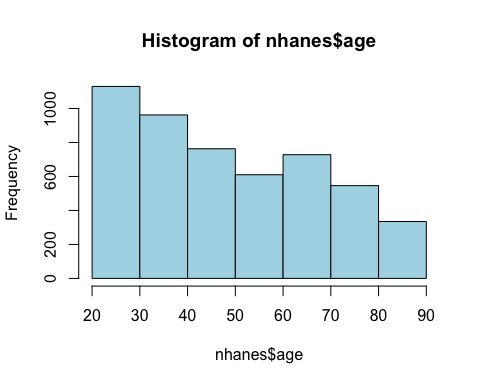
# Histogram, base graphics  
hist(nhanes$age)



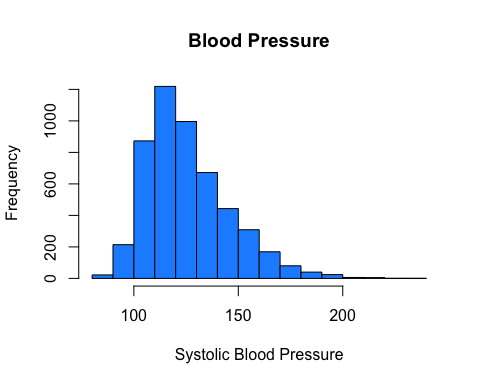
hist(nhanes$age, breaks = 4, col = "gray")



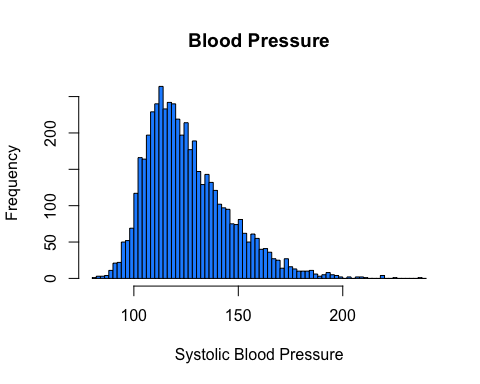
hist(nhanes$age, breaks = c(20, 30, 40, 50, 60, 70, 80, 90), col = "lightblue")



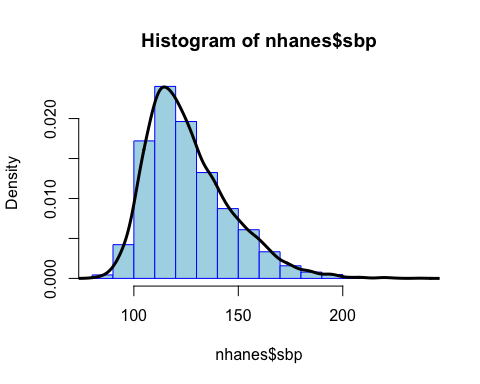
hist(nhanes$sbp, col = "dodgerblue", xlab = "Systolic Blood Pressure", main = "Blood Pressure")



hist(nhanes$sbp, col = "dodgerblue", xlab = "Systolic Blood Pressure", main = "Blood Pressure", breaks = seq(80, 240, by = 2))



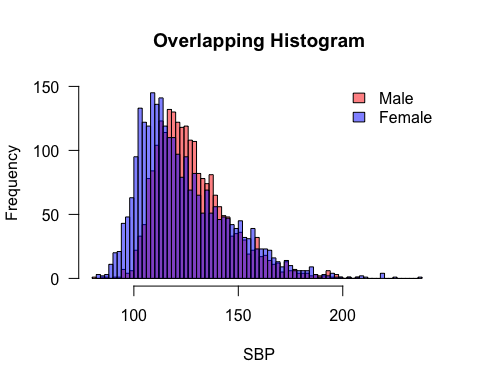
# histogram with density  
hist(nhanes$sbp, breaks = 20, col = "lightblue", border = "blue", freq = F)  
lines(density(nhanes$sbp), lwd = 3)



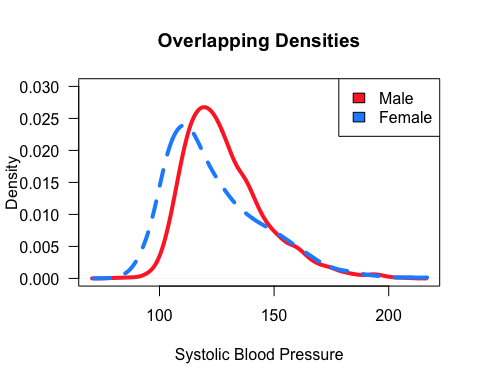
# Overlapping histograms  
tapply(nhanes$sbp, nhanes$sex1, summary)

## $male  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 80.0 116.0 125.0 128.6 138.0 207.0   
##   
## $female  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 83.0 109.0 119.0 124.3 137.0 237.0

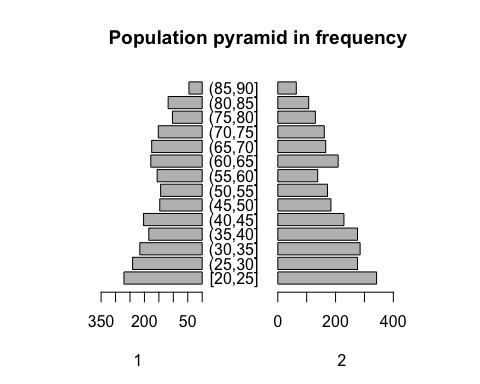
hist(nhanes$sbp[nhanes$sex == 1], col = rgb(1, 0, 0, 0.5), xlim = c(80, 240), ylim = c(0, 150), main = "Overlapping Histogram", xlab = "SBP", breaks = 50, las = 1)  
hist(nhanes$sbp[nhanes$sex == 2], col = rgb(0, 0, 1, 0.5), breaks = 100, add = T)  
legend("topright", legend = c("Male", "Female"), fill = c(rgb(1, 0, 0, 0.5), rgb(0, 0, 1, 0.5)), bty = "n")



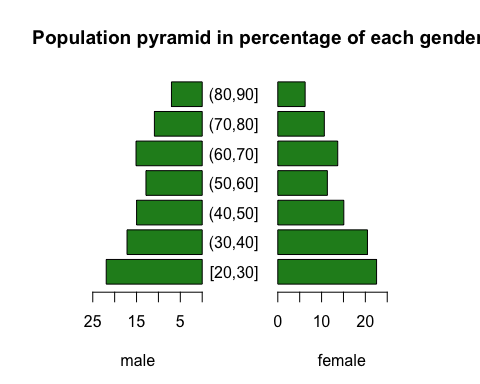
# overlapping densities  
plot(density(nhanes$sbp[nhanes$sex == 1]),  
 lwd = 4, col = "firebrick1", xlab = "Systolic Blood Pressure", las = 1,  
 ylab = "Density", main = "Overlapping Densities", ylim = c(0, 0.03)  
)  
lines(density(nhanes$sbp[nhanes$sex == 2]), lwd = 4, col = "dodgerblue", lty = 2)  
legend("topright", fill = c("firebrick1", "dodgerblue"), legend = c("Male", "Female"))



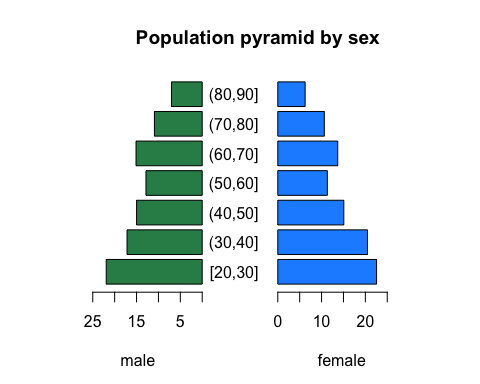
# pyramid plot  
pyramid(nhanes$age, nhanes$sex)



pyramid(nhanes$age, nhanes$sex1, binwidth = 10, col = "forestgreen", percent = "each")

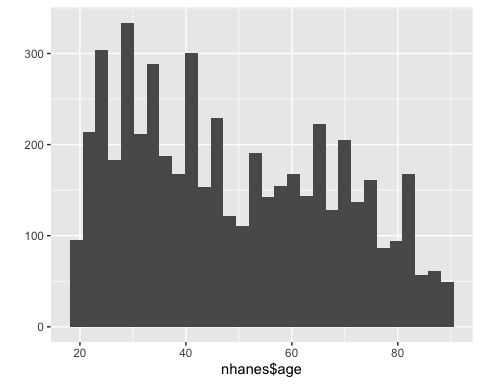


pyramid(nhanes$age, nhanes$sex1, binwidth = 10, col.gender = c("seagreen", "dodgerblue"), percent = "each", main = "Population pyramid by sex")

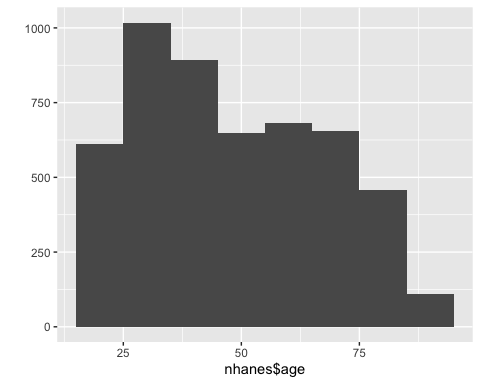


# Histogram, ggplot2 package  
qplot(nhanes$age, geom = "histogram")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

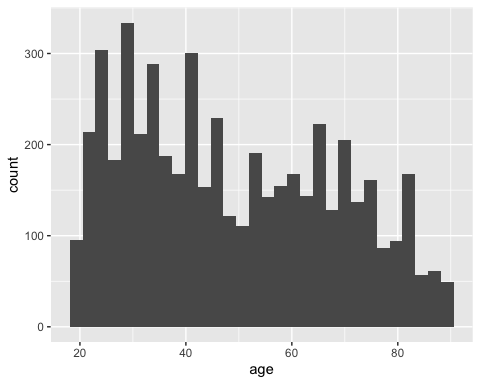


qplot(nhanes$age, geom = "histogram", binwidth = 10)

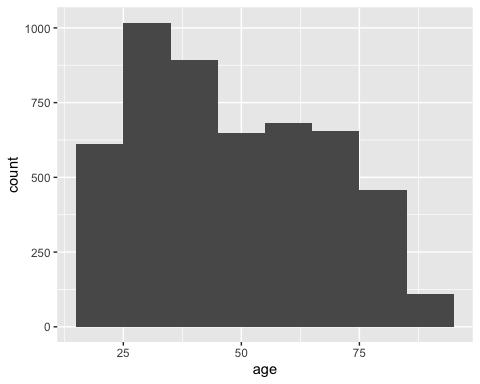


ggplot(nhanes, aes(age)) + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

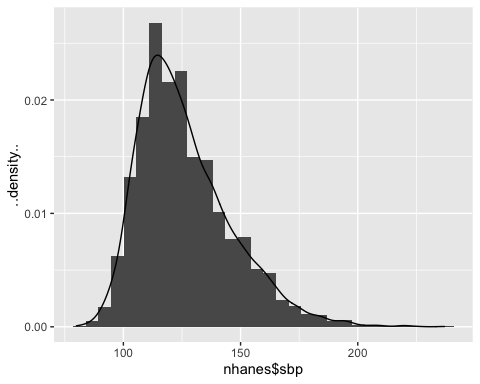


ggplot(nhanes, aes(age)) + geom\_histogram(binwidth = 10)

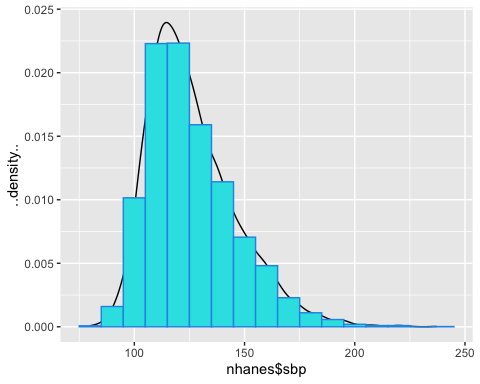


# add density plot  
qplot(nhanes$sbp, y = ..density.., geom = c("histogram", "density"))

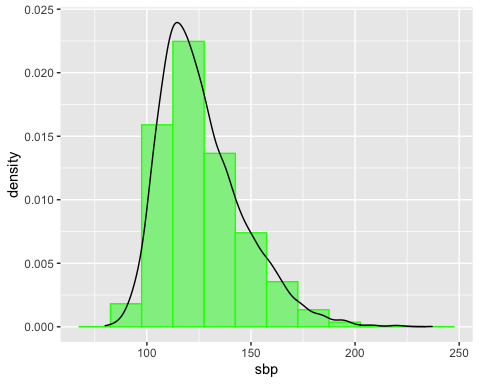
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



qplot(nhanes$sbp, y = ..density.., geom = "density") + geom\_histogram(fill = 5, color = 4, binwidth = 10)

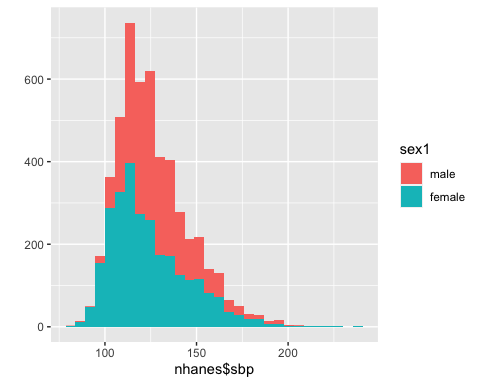


ggplot(nhanes, aes(sbp, ..density..)) + geom\_histogram(fill = "lightgreen", color = "green", binwidth = 15) + geom\_density()



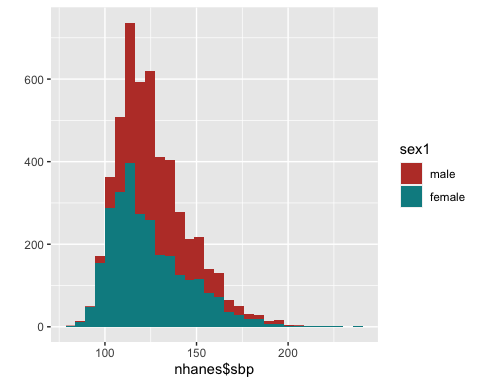
# overlapping histograms  
qplot(nhanes$sbp, geom = "histogram", fill = sex1)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



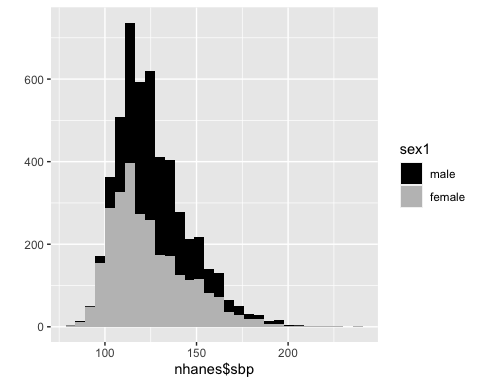
qplot(nhanes$sbp, geom = "histogram", fill = sex1) + scale\_fill\_hue(l = 45)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



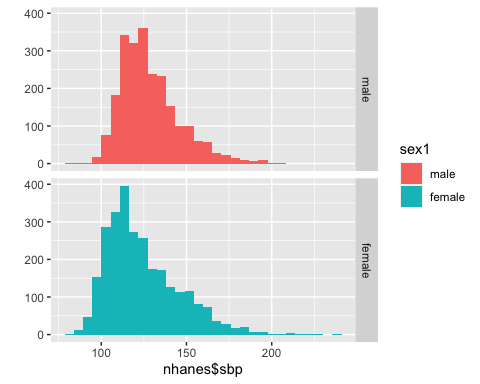
qplot(nhanes$sbp, geom = "histogram", fill = sex1) + scale\_fill\_grey(start = 0, end = 0.75)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



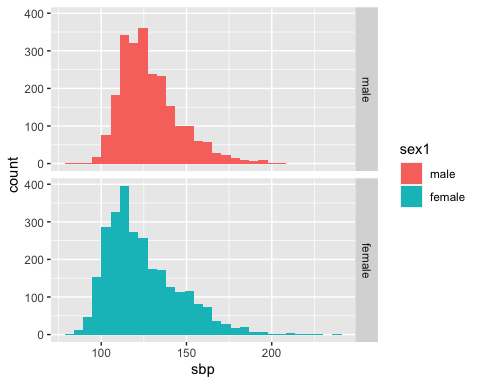
qplot(nhanes$sbp, geom = "histogram", fill = sex1, facets = sex1 ~ .)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



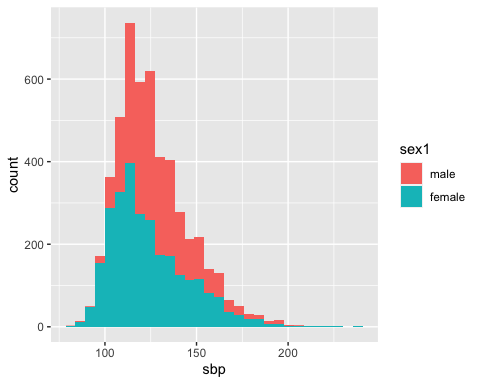
ggplot(nhanes, aes(sbp, fill = sex1)) + geom\_histogram() + facet\_grid(sex1 ~ .)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



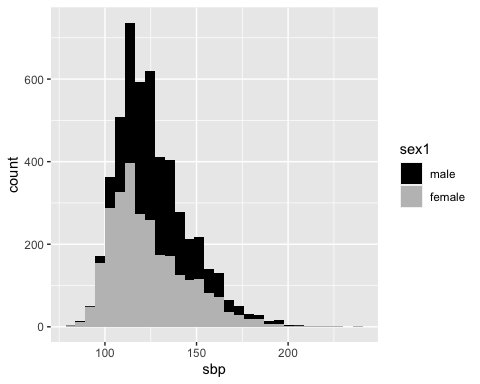
ggplot(nhanes, aes(sbp, fill = sex1)) + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot(nhanes, aes(sbp, fill = sex1)) + geom\_histogram() + scale\_fill\_grey(start = 0, end = 0.75)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Boxplot, base graphics  
median(nhanes$bpb)

## [1] 3.2

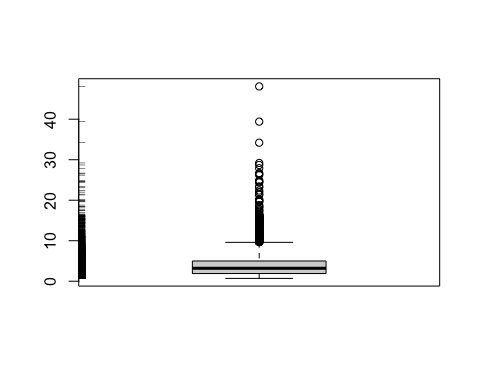
quantile(nhanes$bpb)

## 0% 25% 50% 75% 100%   
## 0.7 1.9 3.2 5.0 48.1

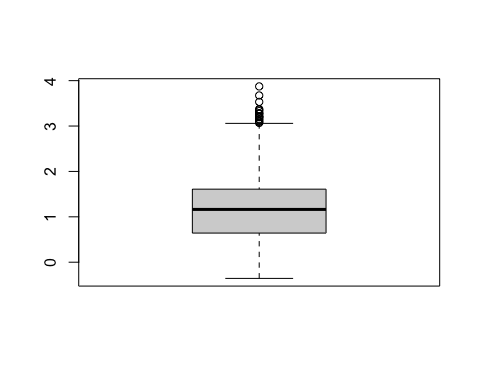
quantile(nhanes$bpb, c(0.1, 0.9))

## 10% 90%   
## 1.1 7.5

boxplot(nhanes$bpb)  
rug(nhanes$bpb, side = 2)



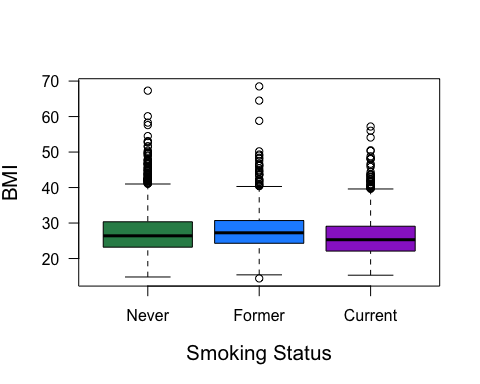
boxplot(log(nhanes$bpb), col = "lightgray")



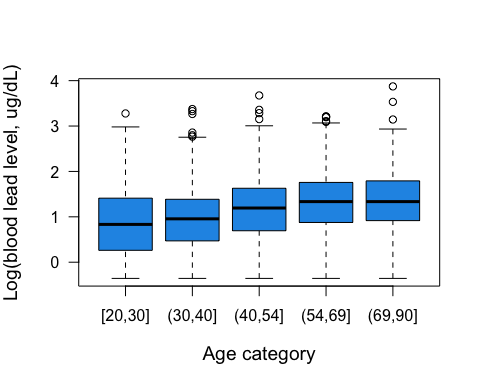
# split boxes by another variable  
table(nhanes$smk)

##   
## 1 2 3   
## 2539 1263 1272

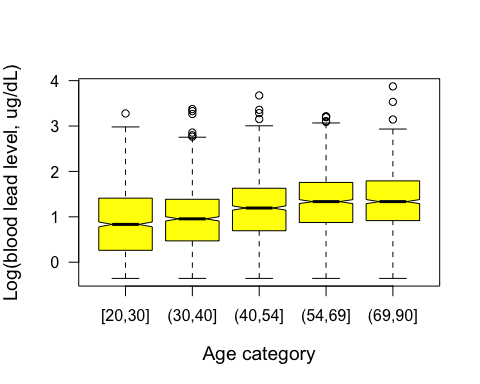
boxplot(nhanes$bmi ~ nhanes$smk, col = c("seagreen", "dodgerblue", "darkorchid"), names = c("Never", "Former", "Current"), las = 1, ylab = "BMI", cex.lab = 1.3, xlab = "Smoking Status")



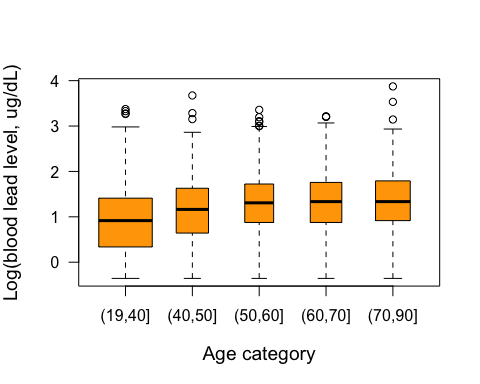
# options for boxes  
boxplot(log(nhanes$bpb) ~ nhanes$AGE5b, col = 4, las = 1, ylab = "Log(blood lead level, ug/dL)", xlab = "Age category", cex.lab = 1.2)



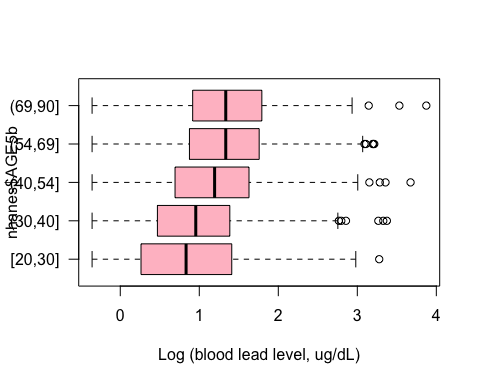
boxplot(log(nhanes$bpb) ~ nhanes$AGE5b, notch = T, col = "yellow", las = 1, ylab = "Log(blood lead level, ug/dL)", xlab = "Age category", cex.lab = 1.2)



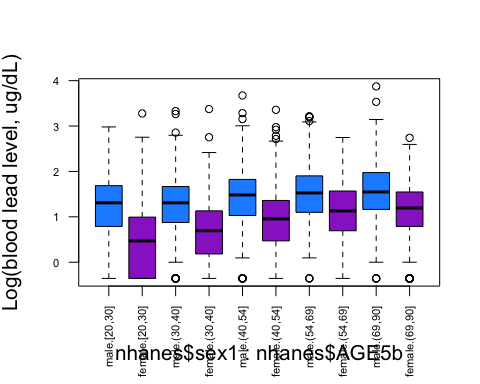
boxplot(log(nhanes$bpb) ~ nhanes$AGE5c, varwidth = T, col = "orange", las = 1, ylab = "Log(blood lead level, ug/dL)", xlab = "Age category", cex.lab = 1.2)



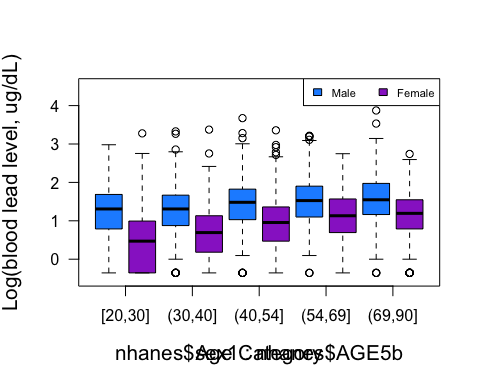
## draw widths proportional to sqrt(No of obs) in the groups  
boxplot(log(nhanes$bpb) ~ nhanes$AGE5b, horizontal = T, col = "pink", las = 1, xlab = "Log (blood lead level, ug/dL)")



## group boxes by two variables  
boxplot(log(nhanes$bpb) ~ nhanes$sex1 \* nhanes$AGE5b, las = 2, ylab = "Log(blood lead level, ug/dL)", col = c("dodgerblue", "darkorchid"), cex.lab = 1.3, cex.axis = 0.7)

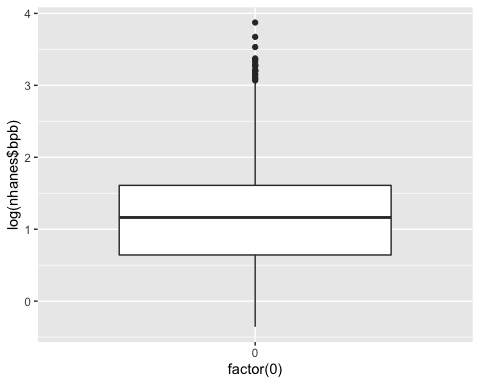


boxplot(log(nhanes$bpb) ~ nhanes$sex1 \* nhanes$AGE5b, las = 2, ylab = "Log(blood lead level, ug/dL)", col = c("dodgerblue", "darkorchid"), cex.lab = 1.3, xaxt = "n", ylim = c(-0.5, 4.5))  
axis(1, at = seq(1.5, 10, 2), labels = levels(AGE5b))  
title(xlab = "Age Category", cex.lab = 1.3)  
legend("topright", c("Male", "Female"), fill = c("dodgerblue", "darkorchid"), cex = 0.7, ncol = 2)

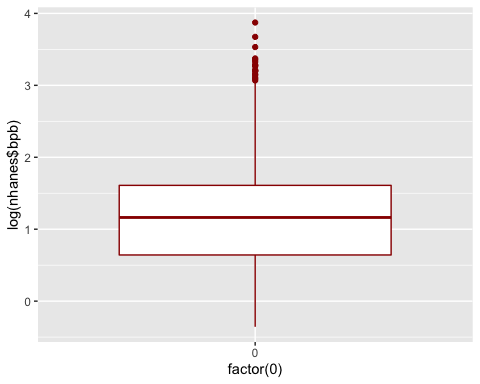


# violin plot  
vioplot(log(nhanes$bpb))  
vioplot(log(nhanes$bpb[nhanes$smk == 1]), log(nhanes$bpb[nhanes$smk == 2]), log(nhanes$bpb[nhanes$smk == 3]), col = "dodgerblue", names = c("Never", "Former", "Current"))  
title("Log(Blood Lead Level) by Smoking Status")

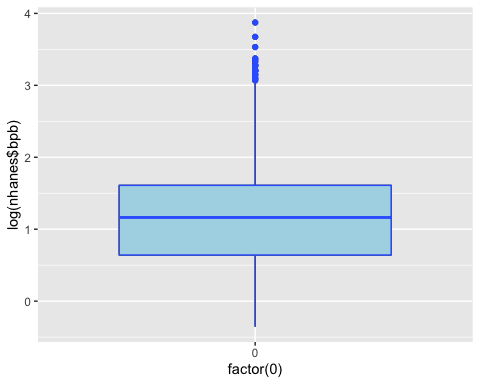
# Boxplot, ggplot2 package  
qplot(factor(0), log(nhanes$bpb), geom = "boxplot")



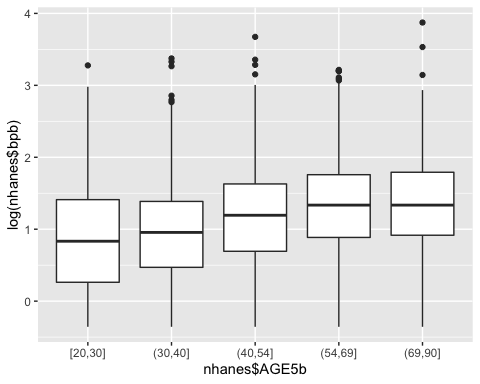
qplot(factor(0), log(nhanes$bpb), geom = "boxplot", color = I("#990000"))



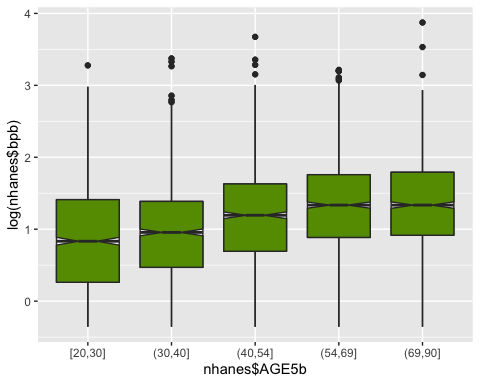
qplot(factor(0), log(nhanes$bpb), geom = "boxplot") + geom\_boxplot(fill = "lightblue", color = "#3366FF", outlier.colour = "#3366FF")



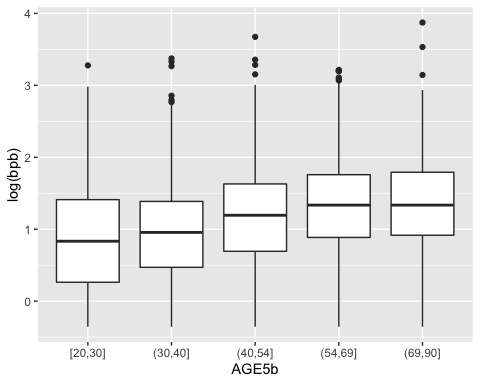
# boxplot split by another variable  
qplot(nhanes$AGE5b, log(nhanes$bpb), geom = "boxplot")



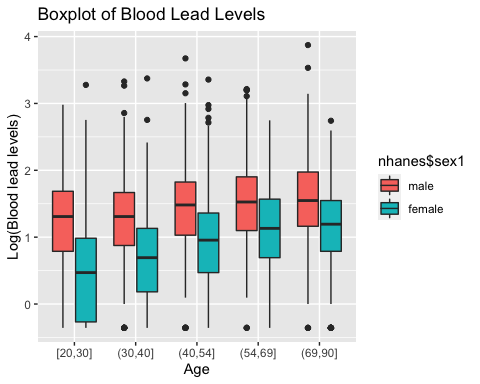
qplot(nhanes$AGE5b, log(nhanes$bpb), geom = "boxplot") + geom\_boxplot(fill = "#669900", notch = T, notchwidth = .3)



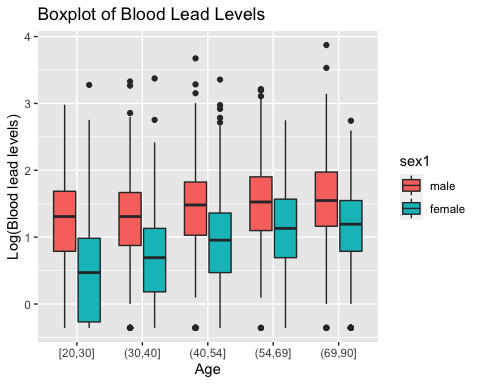
ggplot(nhanes, aes(AGE5b, log(bpb))) + geom\_boxplot()



# boxplot split by two variables  
qplot(nhanes$AGE5b, log(nhanes$bpb), geom = "boxplot", fill = nhanes$sex1, main = "Boxplot of Blood Lead Levels", xlab = "Age", ylab = "Log(Blood lead levels)")



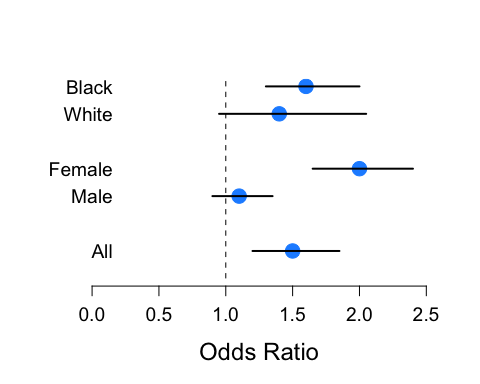
ggplot(nhanes, aes(AGE5b, log(bpb))) + geom\_boxplot(aes(fill = sex1)) + labs(title = "Boxplot of Blood Lead Levels") + scale\_x\_discrete("Age") + scale\_y\_continuous("Log(Blood lead levels)")



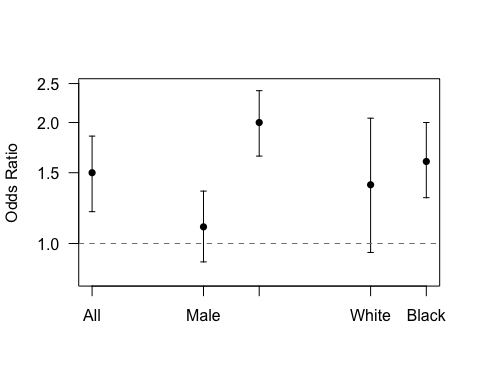
Exercise 5A

## Assuming that you obtained ORs for all and by gender and race.  
## Let's create a data frame for these ORs and 95% CIs.  
  
x.num <- c(1, 3, 4, 6, 7)  
x1 <- c("All", "Male", "Female", "White", "Black")  
or <- c(1.5, 1.1, 2.0, 1.4, 1.6)  
or.ll <- c(1.2, 0.9, 1.65, 0.95, 1.3)  
or.ul <- c(1.85, 1.35, 2.4, 2.05, 2.0)

# Forest plot, base graphics  
plot(or, x.num, xlim = c(0, 2.5), ylim = c(0, 7), pch = 20, bty = "n", ylab = "", xaxt = "n", cex.lab = 1.5, cex = 3, col = "dodgerblue", yaxt = "n", xlab = "Odds Ratio")  
for (i in 1:5) {  
 lines(x = c(or.ll[i], or.ul[i]), y = c(x.num[i], x.num[i]), lwd = 2)  
}  
axis(1, cex.axis = 1.2)  
axis(2, at = x.num, labels = x1, las = 1, lwd = 0, cex.axis = 1.2, pos = .3) # pos is x-axis location of labels  
segments(1, 0, 1, 7.5, lty = 2)



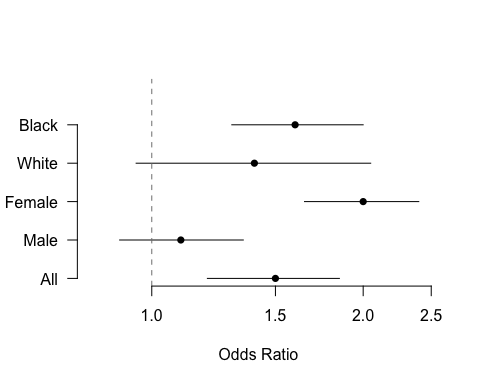
# With errbar() in Hmisc  
## x is a numeric, therefore, this will generate vertical error bars  
errbar(x.num, or, or.ul, or.ll, log = "y", xlab = "", ylab = "Odds Ratio", ylim = c(exp(-0.2), exp(0.9)), xaxt = "n", las = 1)  
abline(h = 1, lty = 2, col = "gray50")  
axis(side = 1, at = c(1, 3, 4, 6, 7), labels = c("All", "Male", "Female", "White", "Black"))



## x1 is a factor (character), therefore, this will generate horizontal error bars  
errbar(x1, or, or.ul, or.ll, log = "x", xlab = "", ylab = "Odds Ratio", ylim = c(exp(-0.2), exp(0.9)), yaxt = "n")

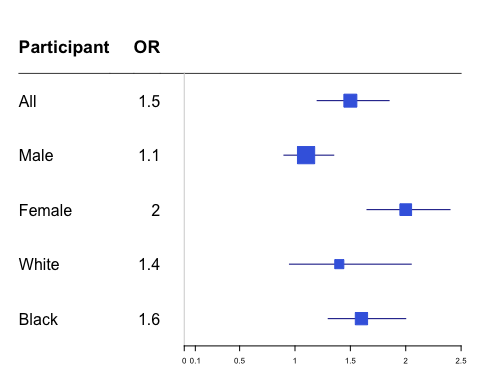
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "log" is not a graphical  
## parameter

abline(v = 1, lty = 2, col = "gray50")



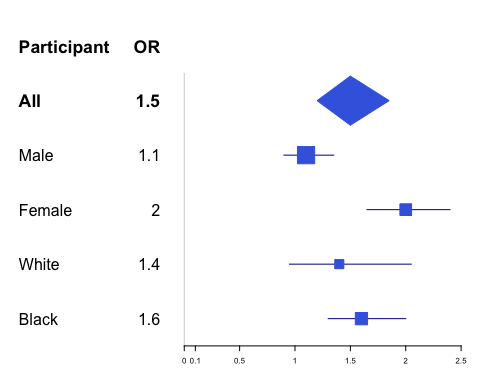
# With forestplot package  
results <- structure(list(  
 mean = c(NA, or),  
 lower = c(NA, or.ll),  
 upper = c(NA, or.ul)  
),  
# .Names = c("mean", "lower", "upper")),  
row.names = x1,  
class = "data.frame"  
)  
tabletext <- cbind(c("Participant", x1), c("OR", or))  
forestplot(tabletext,  
 hrzl\_lines = gpar(col = "#444444"),  
 results, new\_page = TRUE,  
 is.summary = c(TRUE, rep(FALSE, 5)),  
 clip = c(0.1, 3),  
 xlog = FALSE,  
 col = fpColors(box = "royalblue", line = "darkblue", summary = "royalblue")  
)

## Warning in omit | x: longer object length is not a multiple of shorter object  
## length

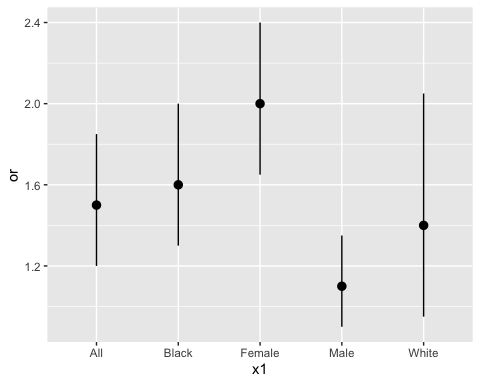


# add summary OR  
forestplot(tabletext,  
 results,  
 new\_page = TRUE,  
 is.summary = c(TRUE, TRUE, rep(FALSE, 4)),  
 clip = c(0.1, 3),  
 xlog = FALSE,  
 col = fpColors(box = "royalblue", line = "darkblue", summary = "royalblue")  
)

## Warning in omit | x: longer object length is not a multiple of shorter object  
## length



# Forest plot, ggplot2 package  
qplot(x1, or, ymin = or.ll, ymax = or.ul) + geom\_pointrange()



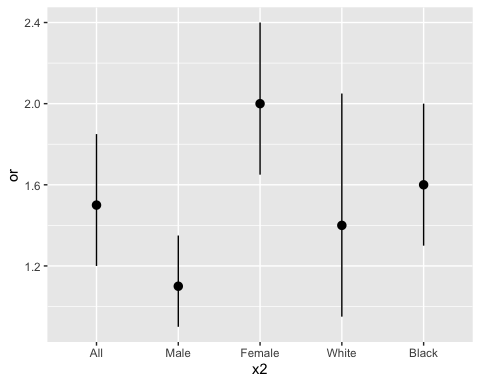
# X-axis is in alphabetical order. To keep the order of the vector, specify 'levels=unique(x1)'.  
  
class(x1)

## [1] "character"

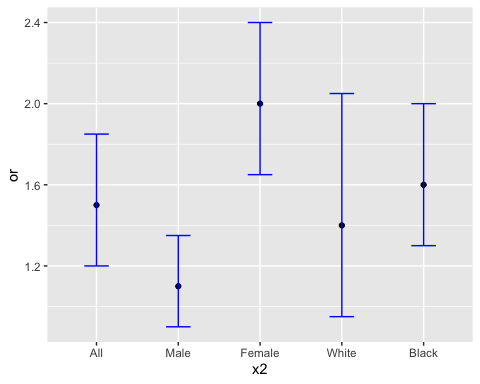
x2 <- factor(x1, levels = unique(x1))  
class(x2)

## [1] "factor"

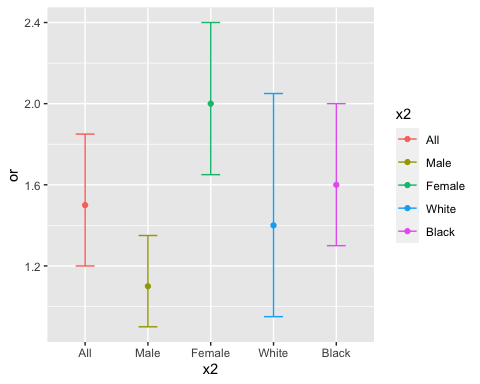
qplot(x2, or, ymin = or.ll, ymax = or.ul) + geom\_pointrange()



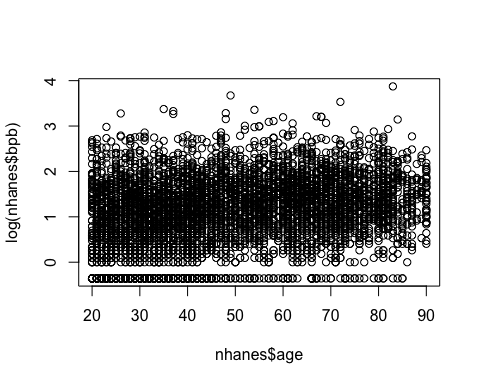
qplot(x2, or, ymin = or.ll, ymax = or.ul) + geom\_errorbar(color = "blue", width = 0.3)



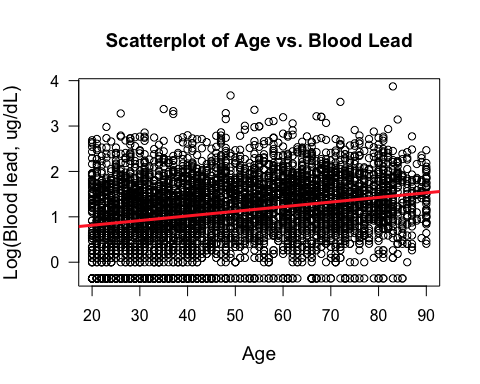
qplot(x2, or, ymin = or.ll, ymax = or.ul, color = x2) + geom\_errorbar(width = 0.3)



# Scatterplot, base graphics  
plot(nhanes$age, log(nhanes$bpb))

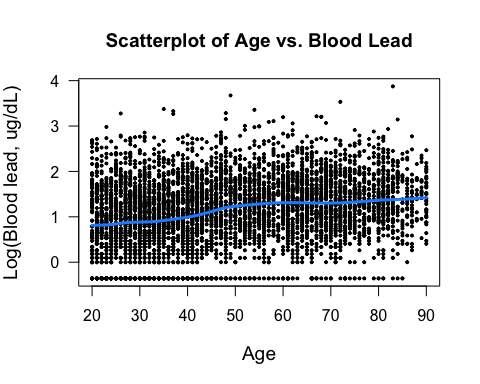


plot(nhanes$age, log(nhanes$bpb), xlab = "Age", ylab = "Log(Blood lead, ug/dL)", las = 1, cex.lab = 1.2)  
title("Scatterplot of Age vs. Blood Lead")  
abline(lsfit(nhanes$age, log(nhanes$bpb)), col = "firebrick1", lwd = 3)



# identify(age,log(bpb))  
# Tool to location spots on the plot  
# locator()

## add a smoothing line  
plot(nhanes$age, log(nhanes$bpb), xlab = "Age", ylab = "Log(Blood lead, ug/dL)", las = 1, cex.lab = 1.2, pch = 20, cex = 0.6)  
title("Scatterplot of Age vs. Blood Lead")  
lines(smooth.spline(nhanes$age, log(nhanes$bpb), df = 10), col = "dodgerblue", lwd = 3)



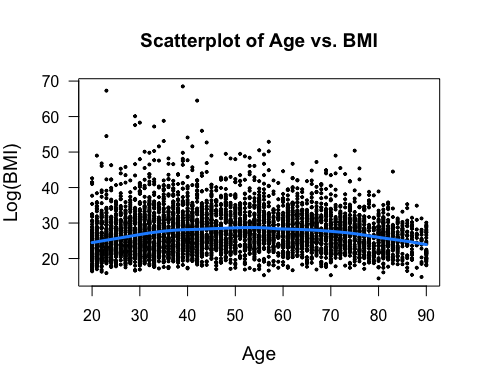
# plot(nhanes$age, nhanes$bmi, xlab = "Age", ylab = "Log(BMI)", las = 1, cex.lab = 1.2, pch = 20, cex = 0.6)  
# title("Scatterplot of Age vs. BMI")  
# lines(smooth.spline(nhanes$age, nhanes$bmi, df = 10), col = "dodgerblue", lwd = 3)  
# smooth.spline() errors out with missing data  
  
nomiss <- na.omit(data.frame(nhanes$age, nhanes$bmi)) # make an object without missing  
dim(nomiss)

## [1] 5064 2

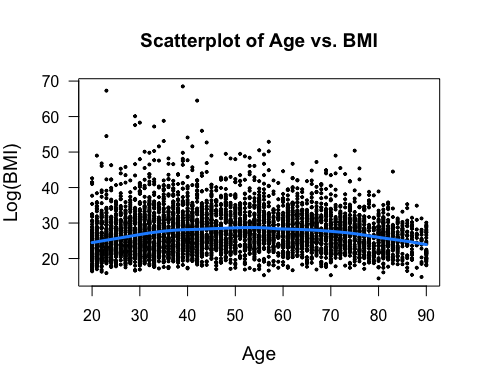
head(nomiss)

## nhanes.age nhanes.bmi  
## 1 21 25.5  
## 2 35 29.4  
## 3 44 44.4  
## 4 48 37.5  
## 5 66 23.6  
## 6 63 31.1

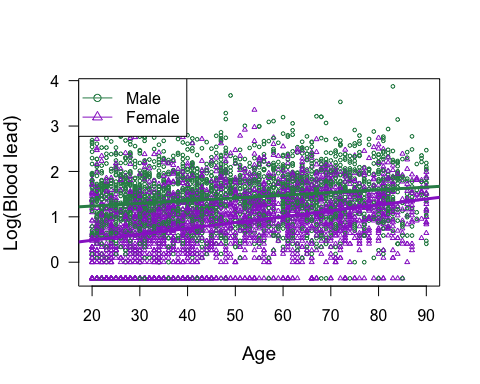
plot(nomiss$nhanes.age, nomiss$nhanes.bmi, xlab = "Age", ylab = "Log(BMI)", las = 1, cex.lab = 1.2, pch = 20, cex = 0.6)  
title("Scatterplot of Age vs. BMI")  
lines(smooth.spline(nomiss$nhanes.age, nomiss$nhanes.bmi, df = 10), col = "dodgerblue", lwd = 3)



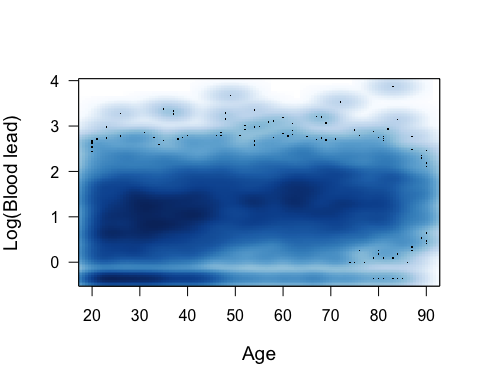
# or restrict to nonmissing within the smooth.spline function  
plot(nhanes$age, nhanes$bmi, xlab = "Age", ylab = "Log(BMI)", las = 1, cex.lab = 1.2, pch = 20, cex = 0.6)  
title("Scatterplot of Age vs. BMI")  
lines(smooth.spline(nhanes$age[!is.na(nhanes$bmi)], na.omit(nhanes$bmi), df = 10), col = "dodgerblue", lwd = 3)



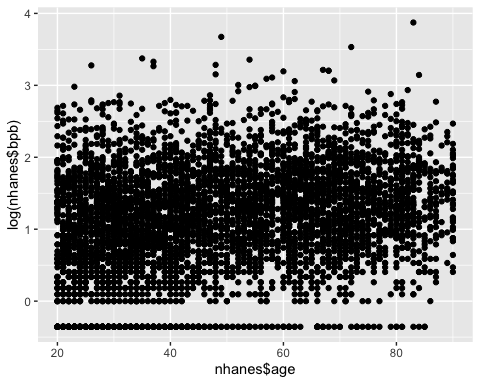
## Scatterplots by group  
plot(nhanes$age, log(nhanes$bpb), xlab = "Age", ylab = "Log(Blood lead)", pch = nhanes$sex, col = c("seagreen", "darkorchid")[nhanes$sex], cex = 0.5, las = 1, cex.lab = 1.2)  
abline(lsfit(nhanes$age[nhanes$sex == 1], log(nhanes$bpb)[nhanes$sex == 1]), lty = 1, lwd = 3, col = "seagreen")  
abline(lsfit(nhanes$age[nhanes$sex == 2], log(nhanes$bpb)[nhanes$sex == 2]), lty = 1, lwd = 3, col = "darkorchid")  
legend("topleft", c("Male", "Female"), lty = c(1, 1), pch = c(1, 2), col = c("seagreen", "darkorchid"))



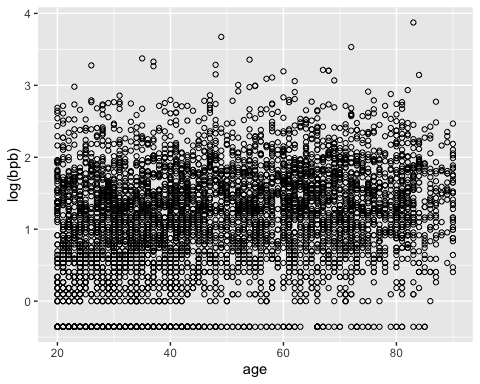
# dealing with dense data overplotting  
smoothScatter(nhanes$age, log(nhanes$bpb), ylab = "Log(Blood lead)", xlab = "Age", las = 1, cex.lab = 1.2)



# Scatterplot, ggplot2 package  
qplot(nhanes$age, log(nhanes$bpb))



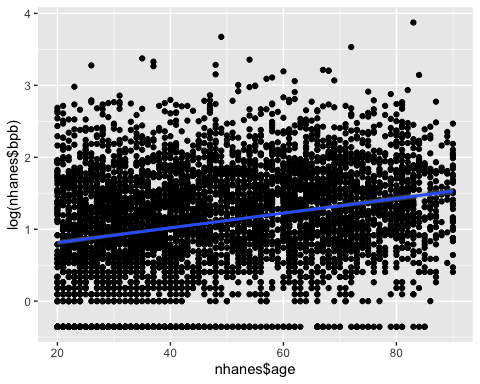
ggplot(nhanes, aes(age, log(bpb))) + geom\_point(shape = 1) # open circles



## add a linear fit  
qplot(nhanes$age, log(nhanes$bpb), geom = c("point", "smooth"), method = lm)

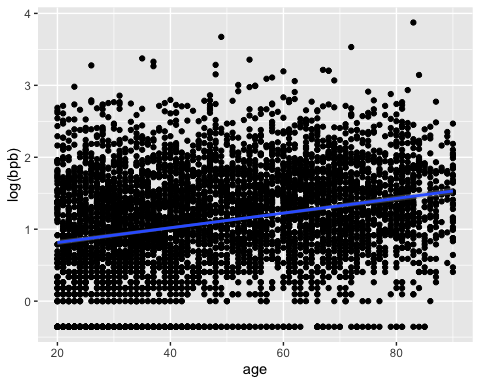
## Warning: Ignoring unknown parameters: method

## `geom\_smooth()` using formula 'y ~ x'



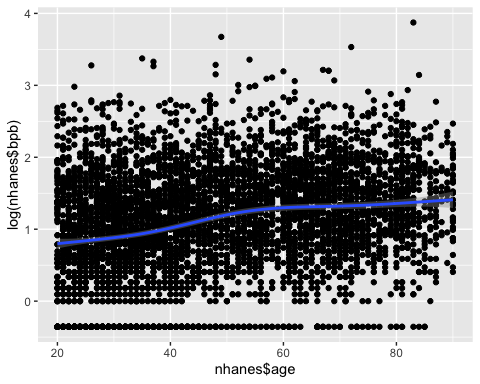
ggplot(nhanes, aes(age, log(bpb))) + geom\_point() + geom\_smooth(method = lm)

## `geom\_smooth()` using formula 'y ~ x'



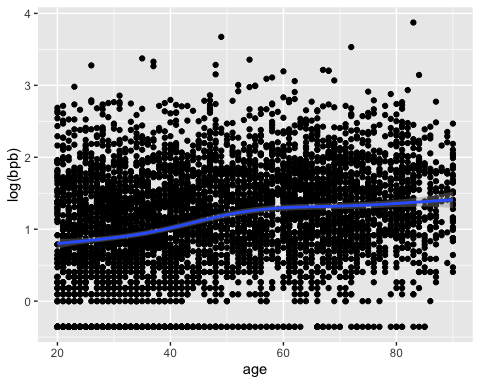
## add a smoothing line  
qplot(nhanes$age, log(nhanes$bpb), geom = c("point", "smooth"))

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



ggplot(nhanes, aes(age, log(bpb))) + geom\_point() + geom\_smooth()

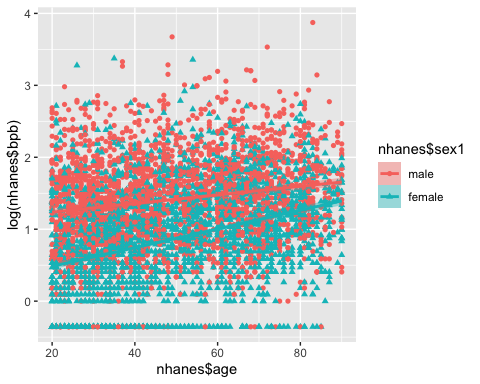
## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



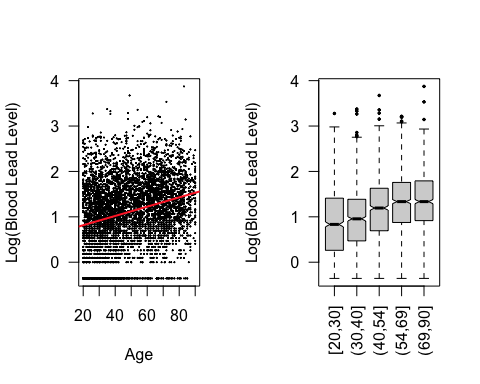
# split by another variable  
qplot(nhanes$age, log(nhanes$bpb), geom = c("point", "smooth"), method = lm, shape = nhanes$sex1, col = nhanes$sex1, fill = nhanes$sex1)

## Warning: Ignoring unknown parameters: method

## `geom\_smooth()` using formula 'y ~ x'



# multiple graphs on one page  
par(mfrow = c(1, 2)) # Here, make 1 row of graphs with 2 columns  
plot(nhanes$age, log(nhanes$bpb), pch = 20, cex = 0.2, ylab = "Log(Blood Lead Level)", las = 1, xlab = "Age")  
abline(lsfit(nhanes$age, log(nhanes$bpb)), col = "firebrick1", lwd = 2)  
boxplot(split(log(nhanes$bpb), nhanes$AGE5b), notch = T, las = 2, ylab = "Log(Blood Lead Level)", pch = 20, cex = 0.5)



par(mfrow = c(1, 1)) # Go back to 1 row of graphs with 1 column of graphs  
## In traditional graphics, just do the graphis in order and they find their place

## In grid, we must instead save the plot to an object  
plot1 <- qplot(nhanes$sex1, fill = nhanes$sex1, geom = "bar") + scale\_fill\_brewer(palette = "Accent") + guides(fill = F) + guides(color = F)  
plot2 <- qplot(nhanes$sbp, geom = "histogram", fill = sex1) + scale\_fill\_hue(l = 45)  
plot3 <- qplot(nhanes$AGE5b, log(nhanes$bpb), geom = "boxplot", fill = sex1, main = "Boxplot of Blood Lead Levels", xlab = "Age", ylab = "Log(Blood lead levels)")  
plot4 <- qplot(nhanes$age, log(nhanes$bpb), geom = c("point", "smooth"), method = lm, shape = nhanes$sex1, col = nhanes$sex1, fill = nhanes$sex1)

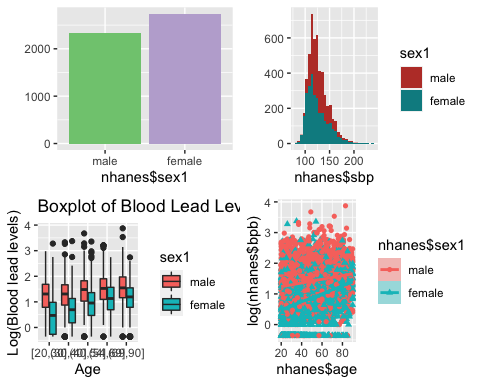
## Warning: Ignoring unknown parameters: method

## first clear the page witht the grid.newpage() function.  
## This is an important step as Otherwise plots printed using  
## the following methods will appear on top of any previous plots.  
  
grid.newpage()  
  
## Next, use the pushViewport() function to define the various frames (viewports)  
## in the grid graphic system  
  
pushViewport(viewport(layout = grid.layout(2, 2)))  
  
## and then use the print function to print the objects into the viewport.  
  
print(plot1, vp = viewport(layout.pos.row = 1, layout.pos.col = 1))  
print(plot2, vp = viewport(layout.pos.row = 1, layout.pos.col = 2))

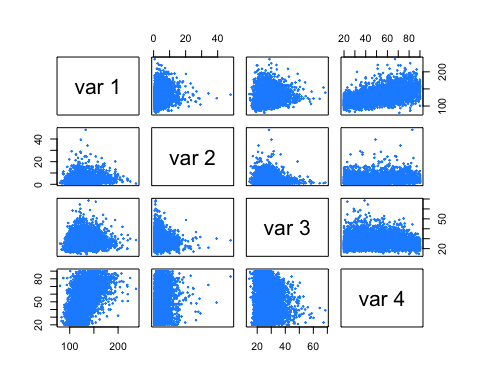
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

print(plot3, vp = viewport(layout.pos.row = 2, layout.pos.col = 1))  
print(plot4, vp = viewport(layout.pos.row = 2, layout.pos.col = 2))

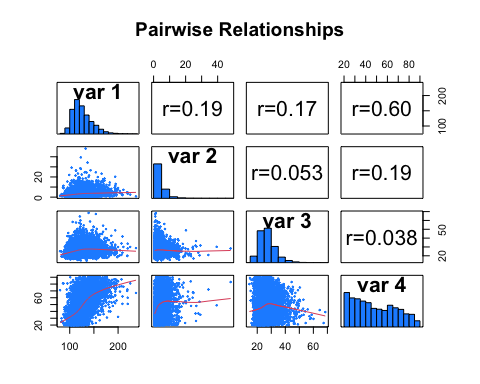
## `geom\_smooth()` using formula 'y ~ x'



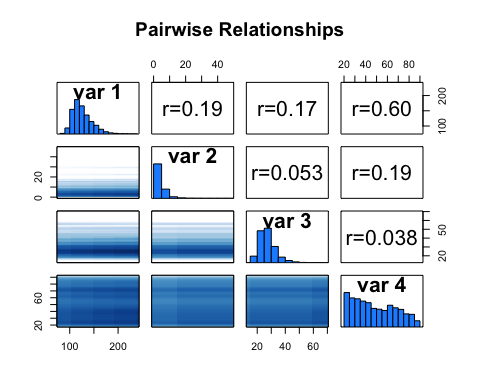
# Scatterplot matrix, base graphics  
  
# pairs function, basic  
pairs(cbind(nhanes$sbp, nhanes$bpb, nhanes$bmi, nhanes$age), pch = 20, cex = 0.4, col = "dodgerblue")



# Set up the function for the correlation coefficient text for the upper cells  
panel.cor <- function(x, y, digits = 2, prefix = "r=", cex.cor, ...) {  
 usr <- par("usr")  
 on.exit(par(usr))  
 par(usr = c(0, 1, 0, 1))  
 r <- abs(cor(x, y, use = "pairwise.complete.obs"))  
 txt <- format(c(r, 0.123456789), digits = digits)[1]  
 txt <- paste0(prefix, txt)  
 if (missing(cex.cor)) cex.cor <- 0.8 / strwidth(txt)  
 text(0.5, 0.5, txt, cex = 2)  
}  
  
# Set up the function for the histograms for the middle cells  
panel.hist <- function(x, ...) {  
 usr <- par("usr")  
 on.exit(par(usr))  
 par(usr = c(usr[1:2], 0, 1.5))  
 h <- hist(x, plot = FALSE)  
 breaks <- h$breaks  
 nB <- length(breaks)  
 y <- h$counts  
 y <- y / max(y)  
 rect(breaks[-nB], 0, breaks[-1], y, ...)  
}  
  
# Put it all together  
pairs(cbind(nhanes$sbp, nhanes$bpb, nhanes$bmi, nhanes$age), main = "Pairwise Relationships", pch = 20, cex = 0.4, font.labels = 2, cex.labels = 2, col = "dodgerblue", upper.panel = panel.cor, diag.panel = panel.hist, lower.panel = panel.smooth)



# One thing to keep in mind: Your NHANES dataset is so large, you'll likely have issues with overplotting. You may need to make it a cairo png instead of a pdf (which would save each dot). You may also need to change the lower panel from the default of plotting each point to a scatterSmooth() function from the graphics package:  
panel.scatterSmooth <- function(x, ...) {  
 smoothScatter(..., nrpoints = 0, add = T)  
}  
pairs(cbind(nhanes$sbp, nhanes$bpb, nhanes$bmi, nhanes$age), main = "Pairwise Relationships", pch = 21, , font.labels = 2, cex.labels = 2, col = "dodgerblue", upper.panel = panel.cor, diag.panel = panel.hist, lower.panel = panel.scatterSmooth)

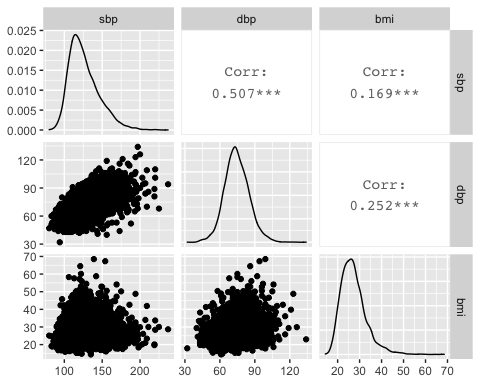


# Scatterplot matrix, ggplot2 package  
ggpairs(nhanes[, c("sbp", "dbp", "bmi")])

## Warning in ggally\_statistic(data = data, mapping = mapping, na.rm = na.rm, :  
## Removed 10 rows containing missing values  
  
## Warning in ggally\_statistic(data = data, mapping = mapping, na.rm = na.rm, :  
## Removed 10 rows containing missing values

## Warning: Removed 10 rows containing missing values (geom\_point).  
  
## Warning: Removed 10 rows containing missing values (geom\_point).

## Warning: Removed 10 rows containing non-finite values (stat\_density).



Exercise 5B