

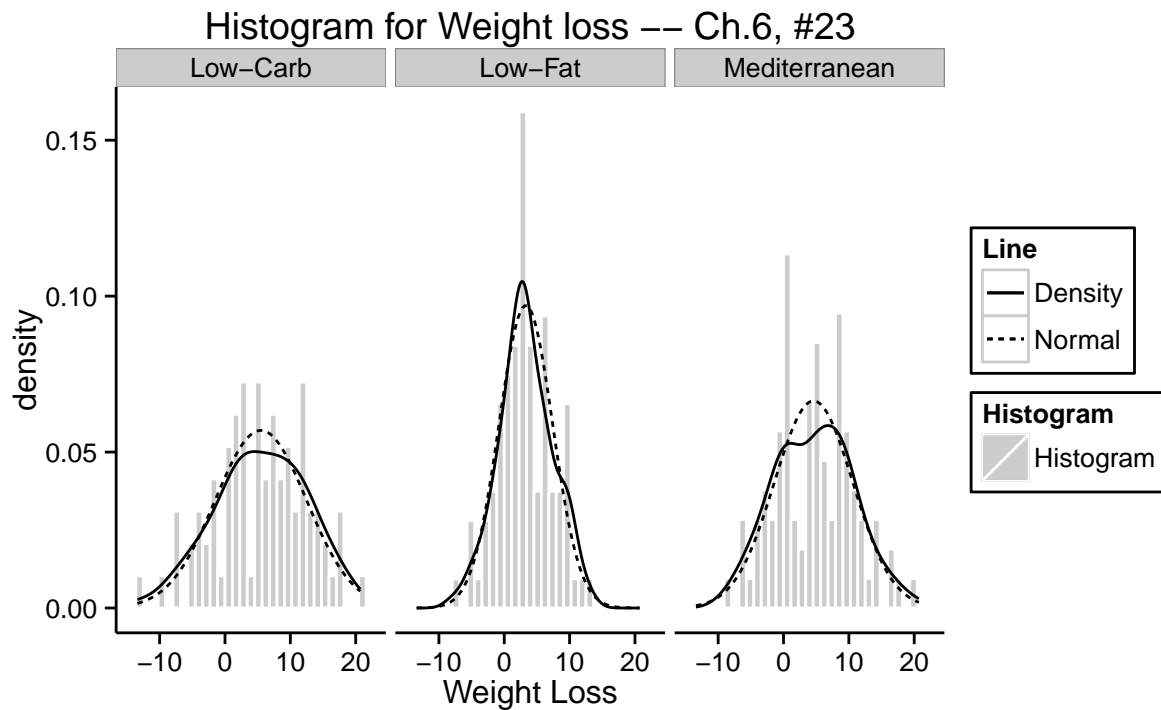
hw6 Stat139

Callin Switzer

October 19, 2014

#3

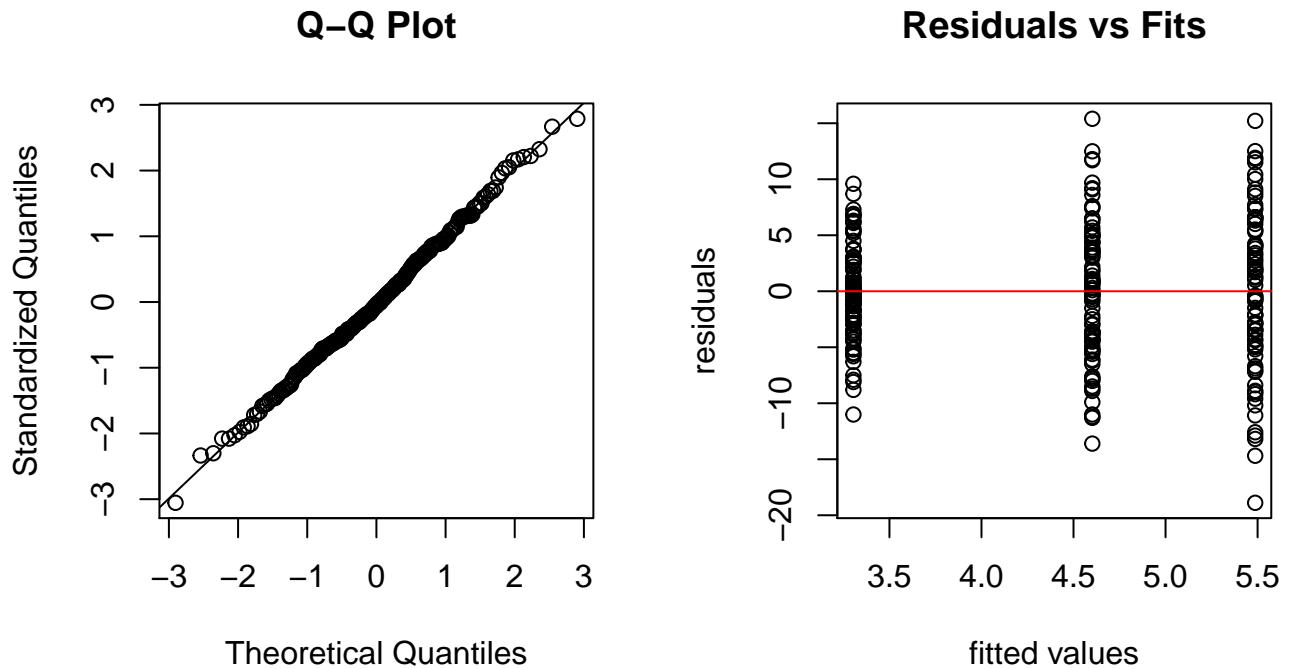
Ch.6, #23 (Diet Wars)



```
## [1] "This comparison of histograms already shows that the groups' variances may not be equal"
```

I wanted to check the assumptions of the ANOVA F-test before using it. It's assumptions are the following: 1. Normality of populations; 2. Equal population variances for all groups; 3. Independence within each group; 4. Independence between groups; 5. Homogeneity within each group; 6. Random sampling

I plotted the residuals and the Q-Q plot below to help check these assumptions.



The only assumptions that might not be met are the following: Random sampling wasn't used, and the populations might not have equal variances. Because the variances look different, I'll use Kruskal-Wallis test to see if I get the same answer.

Here is the summary of the anova F-test

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Group	2	216.86	108.43	3.24	0.0409
Residuals	269	9013.89	33.51		

Here is the summary of the Kruskal-Wallis test

```
##
##  Kruskal-Wallis rank sum test
##
## data:  list(LC, LF, ME)
## Kruskal-Wallis chi-squared = 6.6058, df = 2, p-value = 0.03678
```

Both the Kruskal-Wallis test and the ANOVA F-test give significant results. From the ANOVA F-test, the hypotheses are these:

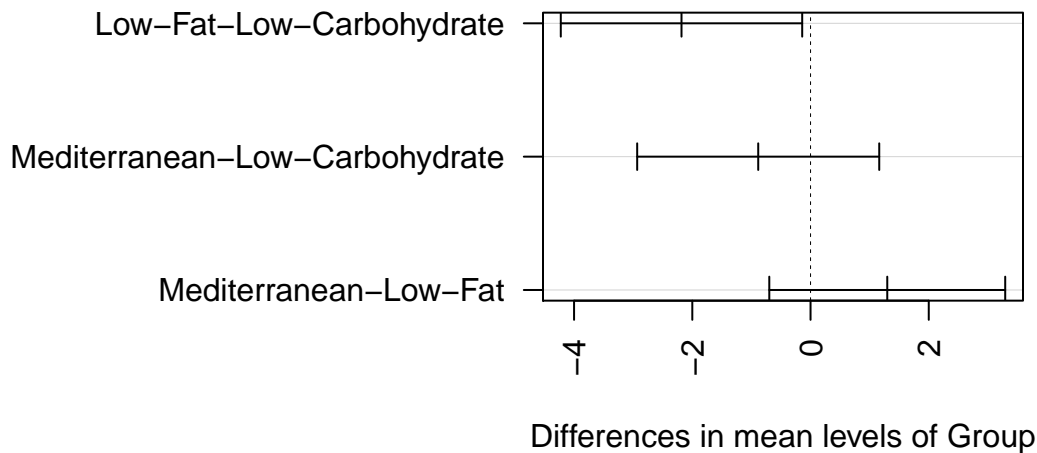
H0: All groups have the same mean HA: At least one group has a different mean from the others

I found evidence that at least one group mean is different from the others. To find which group is different, I will use the Tukey-Kramer method (since sample sizes are different). The Tukey-Kramer method ensures that the familywise error rate stays at a constant rate, which I will use 0.05. Here is the R output for the pairwise comparison, using the Tukey-Kramer method for multiple comparisons (with a plot included):

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = WtLoss24 ~ Group, data = lunch)
##
## $Group
##
```

	diff	lwr	upr	p adj
Low-Fat-Low-Carbohydrate	-2.1828035	-4.224771	-0.1408361	0.0329364
Mediterranean-Low-Carbohydrate	-0.8849083	-2.932082	1.1622656	0.5656813
Mediterranean-Low-Fat	1.2978952	-0.697418	3.2932084	0.2771180

95% family-wise confidence level



When I don't correct for multiple comparisons, I get this:

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: lunch$WtLoss24 and lunch$Group
##
##           Low-Carbohydrate Low-Fat
## Low-Fat      0.012          -
## Mediterranean 0.309          0.126
##
## P value adjustment method: none
```

The p-values are slightly lower, but there is not a difference in which groups are significantly different from one another (i.e. the only two groups that are estimated to have significantly different means are low-fat and low-carbohydrate)

In summary, I used the Tukey-Kramer procedure to compare all possible pairwise means. From this test, I rejected the hypothesis that low-fat and low-carbohydrate diets have the same mean weight loss. I found evidence that low carbohydrate diets are estimated to have higher mean weight loss than low-fat diets.

#4

The difference between planned and unplanned comparisons is as follows: planned comparisons were planned by the researcher before looking at the data. Unplanned comparisons are those which the researcher has decided to investigate after collecting and looking at the data.

Summary of “No Adjustments Are Needed for Multiple Comparisons”: Rothman claims that the problem of multiple comparisons is based on two presumptions: 1. Chance can and does cause unusual findings and 2. No one would want to investigate something caused by chance. He goes on to say that both of these presumptions are false. He claims that when we correct for multiple comparisons, we’re assuming that a “universal” null hypothesis is true. This hypothesis is that all associations we observe reflect random variation. He goes on to say that we should not be penalized for peeking at new data.

How does Dallal respond to the Rothman’s argument? Dallal responds by essentially condensing that experiments are random – He says when you do multiple comparisons, you are increasing the familywise type I error. He responds to Rothman’s argument by saying that it is shortsighted. He claims that experiments with multiple comparisons are fundamentally different from multiple experiments with two groups. Multiple experiments can test targeted hypotheses, while all pairwise comparisons in a single experiment produce more uncertainty. He says that multiple comparisons are tricky, and we can sometimes use a combination of Fisher’s LSD and Tukey’s HSD to guide our analyses.

CODE

```
#####  
##### SETUP  
#####  
  
## read in csv  
lunch <- read.csv("data/ex0623.csv")  
  
## plot histograms to check assumptions:  
AnovaAssumpChecker <- function(group1,  
                                group2,  
                                group3,  
                                group1title = "group1",  
                                group2title = "group2",  
                                group3title = "group3",  
                                graphTitle = "Histograms for Groups") {  
  
  #group1 <- LC  
  #group2 <- LF  
  #group3 <- ME  
  
  mdat <- data.frame(dat = c(group1, group2, group3),  
                     trt = c(rep(group1title, length(group1)),  
                             rep(group2title, length(group2)),  
                             rep(group3title, length(group3))))  
  sp <- data.frame(row.names = 1:nrow(mdat))  
  sp$PctChange <- mdat$dat  
  sp$SpeedLimit <- mdat$trt
```

```

grid <- with(sp, seq(min(PctChange), max(PctChange), length = 100))
normaldens <- ddpby(sp, "SpeedLimit",
  function(df) {
    data.frame(
      predicted = grid,
      density = dnorm(grid, mean(df$PctChange),
        sd(df$PctChange)))
  })

# look at distributions of data
ggplot(sp, aes(x = PctChange)) +
  # histogram
  geom_histogram(aes(y = ..density.., fill = "Histogram"), color = "white",
    alpha = 0.2) +
  # kernel density line
  geom_line(aes(y = ..density.., lty = "Density"), stat = 'density')+
  # normal line
  geom_line(aes(y = density, x = predicted, lty = "Normal"), data = normaldens) +
  # facet
  facet_grid(~SpeedLimit) +

  # labels and theme
  #xlim(c(-3.5, 3.5))+
  labs(x = "Value", title = graphTitle) +
  theme_bw() +
  theme(legend.background = element_rect(colour = "black"),
    plot.background = element_blank()
    ,panel.grid.major = element_blank()
    ,panel.grid.minor = element_blank()
    ,panel.border = element_blank()
    ,axis.line = element_line(color = 'black')) +
  # Names for the legend
  scale_linetype(name = "Line")+
  scale_fill_manual(name = "Histogram", values = c("black"))
## the distribution looks like we need to log-transform the data
}

library(ggplot2)
library(plyr)
library(car)

#####
##### Question 3
#####

options(xtable.comment = FALSE)
LC <- subset(lunch, subset = Group == "Low-Carbohydrate")$WtLoss24
LF <- subset(lunch, subset = Group == "Low-Fat")$WtLoss24
ME <- subset(lunch, subset = Group == "Mediterranean")$WtLoss24

AnovaAssumpChecker(LC, LF, ME,
  "Low-Carb", "Low-Fat", "Mediterranean") + labs(x = "Weight Loss",
  title = "Histogram for Weight loss -- Ch.6, #23")

```

```

print("This comparison of histograms already shows that the groups' variances may not be equal")

#var(LC)
#var(LF)
#var(ME)

library(xtable)

## variances are not equal
mod23 <- aov(WtLoss24 ~ Group, data = lunch)
#plot(mod23)

# this way produced too many plots
# plot(mod23, pch = ".")
## q-q plot
stdLunch <- (lunch$WtLoss24 - mean(lunch$WtLoss24))/sd(lunch$WtLoss24)
qqnorm(stdLunch, main='Q-Q Plot', ylab = "Standardized Quantiles"); qqline(stdLunch)

## residuals plot
{plot(residuals(mod23)~fitted(mod23),
      xlab="fitted values",ylab="residuals",main="Residuals vs Fits")
  abline(h = 0, col = "red")}

#summary(mod23)
fm1.table <- xtable(mod23)
print(fm1.table,floating=FALSE)
# kruskal-test
kt <- kruskal.test(list(LC, LF, ME))
kt

# multiple comparisons
TukeyHSD(mod23)

par(mai = c(1,3,1,1))
plot(TukeyHSD(mod23), las = 2)

# pairwise t-test with pooled variances -- i.e. pairwise t-test for I samples.
pairwise.t.test(lunch$WtLoss24, lunch$Group, p.adj = "none", pool.sd = T)

library(multcomp)
# another method for multiple comparisons
### specify all pair-wise comparisons among levels of variable "group"
## all pairwise comparisons is known as "Tukey Contrasts"
tuk <- glht(mod23, linfct = mcp(Group = "Tukey"))
summary(tuk) # this gives same output as TukeyHSD

# this gives same output as unadjusted pairwise t-tests, with pooled variance
contrasts <- summary(tuk, test = adjusted(type = "none"))

# look at all pairwise combinations
contrasts
plot(cld(contrasts))

```

```

### use sufficiently large upper margin
old.par <- par(mai=c(1,1,1.25,1), no.readonly = TRUE)
### plot
dev.off()
plot(contrasts)
par(old.par)
detach(lunch)

## here's a workaround
library(multcomp)

# make up some data
set.seed(123)
df <- data.frame(response = c(rnorm(46, 10), rnorm(53, 9.5), rnorm(44, 7)),
                        trt = c(rep("one", 46), rep("two", 53), rep("three", 44)))

# ANOVA
modell1 <- aov(response~trt, data = df)
summary(modell1)

## here are two ways to do unadjusted pairwise t-tests with pooled variance:
###
# 1
###
pairComp <- glht(modell1, linfct = mcp(trt = "Tukey")) # make all pairwise comparisons
summary(pairComp, test = adjusted(type = "none")) # unadjusted p-values, using t-tests with pooled vari

###
# 2
###
pairwise.t.test(df$response, df$trt, p.adj = "none", pool.sd = T))

## here are two ways to do adjusted pairwise comparisons:
###
# 1
###
pairComp <- glht(modell1, linfct = mcp(trt = "Tukey")) # make all pairwise comparisons
summary(pairComp) # adjusted p-values, using "Tukey" method

###
# 2
###
TukeyHSD(modell1) # model 1 is the ANOVA from above

#####
##### Checking Q1
#####
# check #1

```

```

pop <- rnorm(100000, mean = 10, sd = 10)

means <- replicate(10000, mean(sample(pop, 100)))

var(pop)
var(means)

sampleM <- rnorm(100000, 10, sd = sqrt(var(means)))

var(sampleM)

mean(pop)
mean(sampleM)

mean(pop - sampleM)
var(pop - sampleM)

100*(101/100)

nes <- (pop - sampleM) / sqrt(100*(101/100))
mean(nes)
var(nes)

#####
##### Check Q2
#####

##### check #2 (#16, ch.6) #####

# Tukey-Kramer
qtukey(p = 0.95, nmeans = 6, df = 35, nranges = 1, lower.tail = TRUE, log.p = FALSE)/sqrt(2)
qtukey(p = 0.95, nmeans = 5, df = 65, nranges = 1, lower.tail = TRUE, log.p = FALSE)
?qtukey

1- 0.05/(2*15)
# Bonf
qt(p = 1- 0.05/(2*15), df = 30)

qt(p = 1- 0.05/(2*10), df = 65)

# scheffe
sqrt(4 * qf(0.95, df1 = 4, 65))

sqrt(5* qf(0.95, df1=5, df2 = 30))

```