Week 5 Discussion: Multiple ANOVA

Ken Hydock

2022-08-01

# Load the Required Libraries

From our demos, I’m going to use data.table as it’s necessary and ggpubr for the more robust plots.

# Import, Convert, and Inspect Data

Loading from my working directory - remember to change this to your own if you run this code yourself. We’re also converting the data to a data.table and we’ll check out its structure.

setwd("C:\\Users\\jerem\\OneDrive\\Documents\\School\\\_REGIS\\2022-05\_Summer\\MSDS660\\Week5")  
dt <- read.csv("Interactions\_Categorical.csv")  
dt <- as.data.table(dt)  
str(dt)

## Classes 'data.table' and 'data.frame': 80 obs. of 3 variables:  
## $ Enjoyment: num 81.9 84.9 90.3 89.6 97.7 ...  
## $ Food : chr "Hot Dog" "Hot Dog" "Hot Dog" "Hot Dog" ...  
## $ Condiment: chr "Mustard" "Mustard" "Mustard" "Mustard" ...  
## - attr(\*, ".internal.selfref")=<externalptr>

Looks like we have variables for Enjoyment, Food, and Condiment. If I had to guess, I’d say we’re seeing how much subjects enjoy various foods with various condiments…

# Convert Food and Condiment to factors

We know by now that categorical variables aren’t going to play nice unless we factorize them. I’ve been enjoying exploring different ways to do things, so here is a way we can convert a list of columns to factors.

library('magrittr')  
library('dplyr')  
  
cols <- c("Food", "Condiment")  
  
dt %<>% mutate\_each\_(funs(factor(.)),cols)

## Warning: `mutate\_each\_()` was deprecated in dplyr 0.7.0.  
## Please use `across()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was generated.

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was generated.

str(dt)

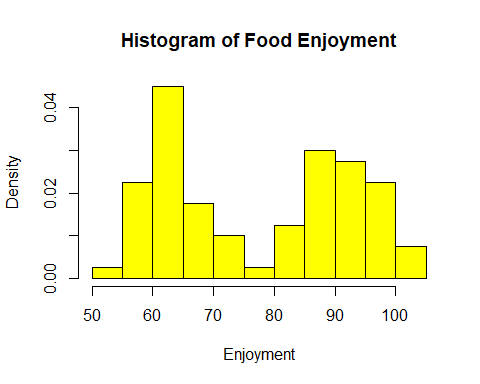
## Classes 'data.table' and 'data.frame': 80 obs. of 3 variables:  
## $ Enjoyment: num 81.9 84.9 90.3 89.6 97.7 ...  
## $ Food : Factor w/ 2 levels "Hot Dog","Ice Cream": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Condiment: Factor w/ 2 levels "Chocolate Sauce",..: 2 2 2 2 2 2 2 2 2 2 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

That makes more sense now - we have two foods, Hot Dog and Ice Cream and putting both str outputs together, Chocolate Sauce and Mustard as our condiments. Yum…?

# Plot a Histogram of Enjoyment

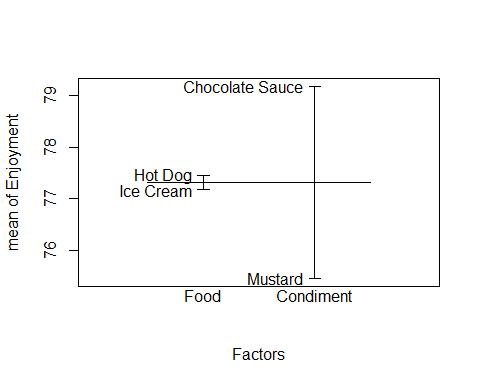
We can check out what we’re working with by a quick histogram to see the density of our dependent variable, Enjoyment. In honor of Mustard, let’s use yellow!

hist(dt$Enjoyment,  
 prob=TRUE,  
 main = "Histogram of Food Enjoyment",  
 xlab = "Enjoyment",  
 ylab = "Density",  
 col="yellow")

 That’s a pretty interesting distribution; not normal in any way. Enjoyment is either love it or hate it, it seems.

# Plot Enjoyment vs the 2 other factors

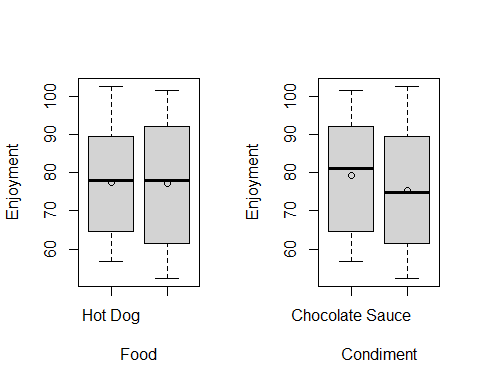
plot.design(Enjoyment ~ ., data = dt)

 I think what we can derive from this plot.design is that Chocolate Sauce has a more significant effect on Enjoyment than Mustard and that Hot Dogs are enjoyed more than Ice Cream. Let’s see if we can prove these conclusions with other plots.

# Plot Individual Boxplots with means

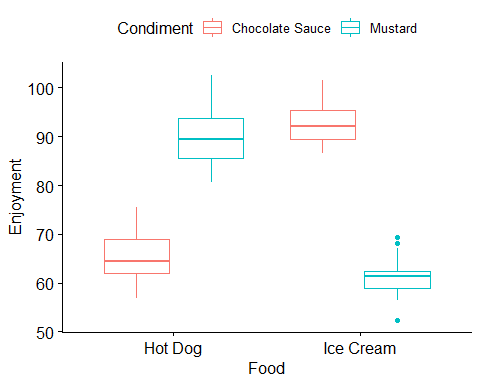
Next up we can look at boxplot to get more visible information on our data. We can see who is the winner by mean from our hypothesis above.

par(mfrow = c(1,2))  
boxplot(Enjoyment ~ Food, data = dt)  
points(dt[, mean(Enjoyment), by=Food])  
boxplot(Enjoyment ~ Condiment, data = dt)  
points(dt[, mean(Enjoyment), by=Condiment])

 I would say the mean Enjoyment for Food is equal, but Chcoloate Sauce is the clear favorite Condiment. Even enjoying both, I would have to say that Chocolate Sauce does have and extra kick of happiness where as Mustard is just “yummy”.

We can try to ggboxplot to make a more powerful (read colorful) graph while including all of our data.

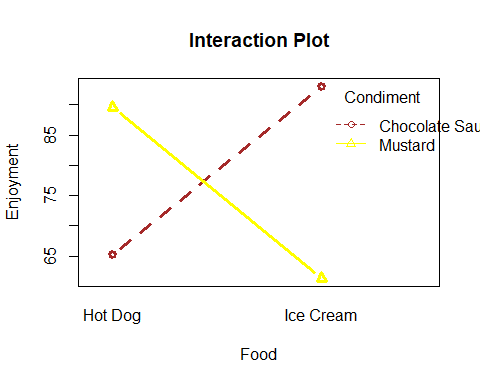
ggboxplot(dt, x = "Food", y = "Enjoyment", color = "Condiment")

 We can see the natural pairing of Food and Condiment does result in higher Enjoyment. It is also quite distinct that Ice Cream + Chocolate Sauce is slightly more enjoyable. However, can can also observe that people derive more Enjoyment from chocolate covered hot dogs than mustard on ice cream - though there are a couple of weirdos that seem to like the latter combo.

# Create interaction plot looking at Condiment and Food

Thankfully, since we only have two groups of two independent variables we can squeeze everything into a single interaction.plot and cleverly color it according to our Condiments in this case. The colors didn’t show very well so I specified the line thickness with lwd = 3

interaction.plot(x.factor = dt$Food,  
 trace.factor = dt$Condiment,   
 response = dt$Enjoyment,  
 fun = mean,   
 type = "b", # shows each point  
 main = "Interaction Plot",  
 legend = TRUE,  
 trace.label = "Condiment",  
 xlab = "Food",  
 ylab="Enjoyment",  
 pch=c(1, 2),  
 col = c("Brown", "Yellow"),  
 lwd = 3)

 No new insights gleaned from this chart. We see the same relationships we observed on the ggboxplot.

# Build ANOVA Model

From our demo, we know that the \* and + operator perform as interactions and specific groupings respectively. Let’s see what we get when we use everything we have with the interaction operator.

fit <- aov(Enjoyment ~ Food \* Condiment, data = dt)  
summary(fit)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Food 1 2 2 0.064 0.80136   
## Condiment 1 278 278 11.071 0.00135 \*\*   
## Food:Condiment 1 15696 15696 626.153 < 2e-16 \*\*\*  
## Residuals 76 1905 25   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We can quickly see that the Food group is statistically insignificant. Recalling our purpose of ANOVA, that means our means between the Hot Dog and Ice Cream are not different or plainly said, people like both statistically, equally. Both Condiment and Food:Condiment are significant, as we should expect from our graphs.

With the tools we’ve learned from our demos, we can narrow our model down to just the significant variables.

fit2 <- aov(Enjoyment ~ Condiment + Food:Condiment, data = dt)  
summary(fit2)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Condiment 1 278 278 11.07 0.00135 \*\*   
## Condiment:Food 2 15697 7849 313.11 < 2e-16 \*\*\*  
## Residuals 76 1905 25   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Everything is significant! We can takeaway from this that the Condiment types and the relationship between Condiment:Food results in a different mean. In absolute layman terms, more people enjoy Chocolate Sauce than Mustard and the effect of these as a condiment on Hot Dog and Ice Cream is not identical.

# Perform TukeyHSD

Now that we have our ANOVA model fitted, we can perform some post hoc analysis. Admittedly, post hoc of multiple ANOVA can get messy fast, which is why it was important for us to eliminate the insignificant relationships early so they won’t cause clutter here.

TukeyHSD(fit2)

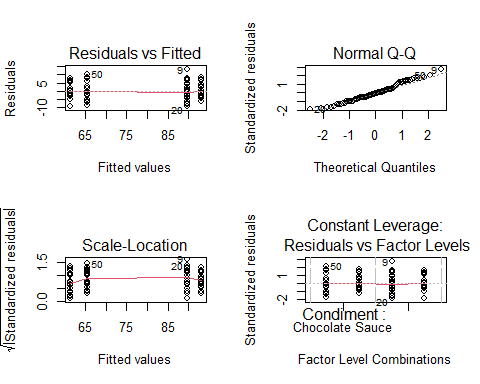
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = Enjoyment ~ Condiment + Food:Condiment, data = dt)  
##   
## $Condiment  
## diff lwr upr p adj  
## Mustard-Chocolate Sauce -3.725054 -5.954796 -1.495311 0.0013533  
##   
## $`Condiment:Food`  
## diff lwr  
## Mustard:Hot Dog-Chocolate Sauce:Hot Dog 24.289075 20.1301777  
## Chocolate Sauce:Ice Cream-Chocolate Sauce:Hot Dog 27.731484 23.5725866  
## Mustard:Ice Cream-Chocolate Sauce:Hot Dog -4.007699 -8.1665961  
## Chocolate Sauce:Ice Cream-Mustard:Hot Dog 3.442409 -0.7164887  
## Mustard:Ice Cream-Mustard:Hot Dog -28.296774 -32.4556714  
## Mustard:Ice Cream-Chocolate Sauce:Ice Cream -31.739183 -35.8980803  
## upr p adj  
## Mustard:Hot Dog-Chocolate Sauce:Hot Dog 28.447973 0.0000000  
## Chocolate Sauce:Ice Cream-Chocolate Sauce:Hot Dog 31.890382 0.0000000  
## Mustard:Ice Cream-Chocolate Sauce:Hot Dog 0.151199 0.0630825  
## Chocolate Sauce:Ice Cream-Mustard:Hot Dog 7.601306 0.1397654  
## Mustard:Ice Cream-Mustard:Hot Dog -24.137876 0.0000000  
## Mustard:Ice Cream-Chocolate Sauce:Ice Cream -27.580285 0.0000000

First, we see what we already know - our Condiment groups are significant of each other. Below that, we see our Condiment:Food relationships broken down and a lot of good information! The proper combination (Hot Dog:Mustard and Ice Cream:Chocolate Sauce) as well as their inverse are not significant. Meaning the proper combination is mutally enjoyable and the improper mutually revolting. The rest of the comparisons are saying that a good combo:bad combo is statistically significant.

# Plot the residuals of the fit

Now we can take a look at our residuals, a plot set that we’re becoming quite familiar with…

par(mfrow = c(2,2))  
plot(fit2)

 No surprises here - I would actually say that’s the best Residuals vs Fitted plot I’ve seen this class; maybe skewed ever so slightly to the negatives, but the homoskedasticity looks great (I think). The Normal Q-Q plot shows a notable departure from normality, but if you recall our histogram of enjoyment density there wasn’t much in the middle; this is a love it or hate it data set. I think the love it-hate it relationship is responsible for the plateau in Scale-Location. We know we have no outliers (other than a few weirdos who thought mustard on ice cream wasn’t so bad), so the leverage plot is relatively perfect.

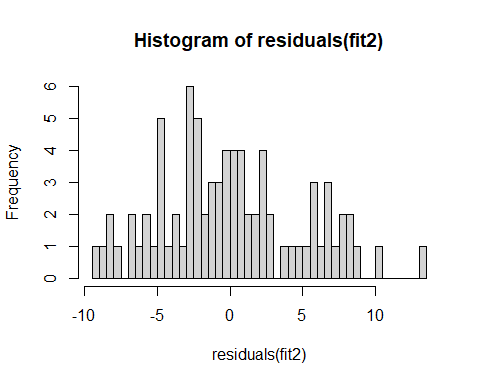
# Perform Shapiro test to see if residuals are normally distributed.

I didn’t quite understand what I was looking at with the text results from this test, so I looked it up on Statology for a better explanation.

shapiro.test(residuals(fit2))

##   
## Shapiro-Wilk normality test  
##   
## data: residuals(fit2)  
## W = 0.97937, p-value = 0.2229

hist(residuals(fit2), breaks=40)

 It’s important to remember that we’re testing the **residuals** here, not the Enjoyment, which we know was anything but normal. We have a p-value of 0.2229 and a p-value **above** significance (0.05) is good, or normally distributed. I can sympathize that this is normal, but it definitely looks skewed to the left and isn’t that far from the confidence level. We can call it my favorite term: “mostly” normal.

# References

Zach. (2021, September 29). How to perform a Shapiro-Wilk test in R (with examples). Statology. Retrieved August 1, 2022, from <https://www.statology.org/shapiro-wilk-test-r/>