Analyzing Golf Ball Backspin

Tyler Wiederich

Fall 2022

# Introduction

In the previous lab, we analyzed the carry distance of various golf ball types via the difference of two means. In this lab, we will look at the backspin (measured in RPM) for these same types. The data were collected as a part of the experiment by Mark Crossfield. We will analyze this data using ANOVA methodology, and we will gain experience with the emmeans R package.

* Backspin is important for a golfer because it allows the golfer to control the ball as it lands. This is especially important when taking shots with clubs called irons. The data collected are shots hit using a 7-iron club.
* For this example, we will assume that more backspin is more desirable.

The corresponding data can be found in the 7IronSpin.csv file. Below is a preview of the data set.

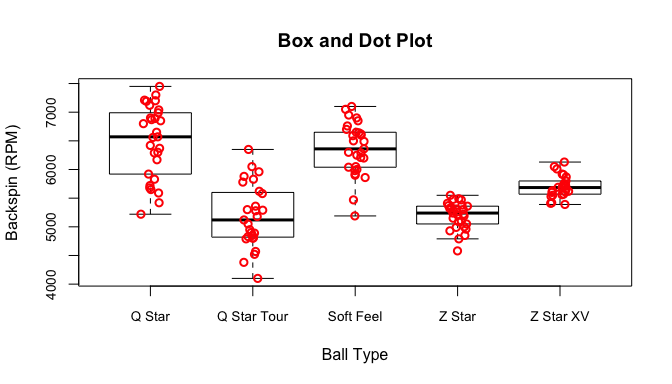
> # make sure you have this file in the directory of your  
> # .rmd file  
> spin <- read.csv("7IronSpin.csv")  
> head(spin)

Ball Backspin  
1 Q Star 7450  
2 Q Star 7300  
3 Q Star 7210  
4 Q Star 7200  
5 Q Star 7190  
6 Q Star 7120

**Questions**

* What is the factor? What are the factor levels?
* Identify the response variable.
* What is the experimental unit?
* Describe the design of this experiment. What type of design is this? Are there any drawbacks of this design? What do you think could be done to improve the design of the experiment?

Let’s take a look at a plot of the spin rates obtained by Mark Crossfield.



**Question** - Do you think there are differences among the means? Remember, a plot like this is meant to give an initial impression of the data relative to the research hypotheses. Focus on the shifts of the points, the variability of the points, and the location of the medians.

# Analysis of Variance

Recall that our goal is to determine whether there is a difference in mean backspin levels among different ball types.

ANOVA methods allow us to test the following set of hypotheses

where is the number of treatments. For our problem, we can express the full set of hypotheses as

We will conduct this hypothesis test using .

In the class notes, we discussed the test statistic method and the p-value method for the evaluation of the aforementioned hypothesis test.

* For the test statistic method, we use where and represent the mean square for the treatments and the mean square for the error, respectively. Recall that we reject when where denotes the total number of observations and denotes the number of treatments.
* For the p-value method, we calculate the where is a random variable from an F distribution with and degrees of freedom and is calculated as above. We reject when the p-value is less than .

Let’s proceed with these calculations piece by piece first, and then we will verify our calculations using built in R functions. We will need to calculate and . Recall, the formulas are

> # We will calculate mean square of treatments first. We  
> # need our treatment means, the grand/overall mean, and the  
> # sample sizes  
> save.mean <- aggregate(Backspin ~ Ball, data = spin, FUN = mean)[,  
 2]  
> save.n <- aggregate(Backspin ~ Ball, data = spin, FUN = length)[,  
 2]  
> tot.mean <- mean(spin$Backspin)  
> N <- length(spin$Backspin)  
> t <- length(unique(spin$Ball))  
> # now we compute mst  
> mst <- sum(save.n \* (save.mean - tot.mean)^2)/(t - 1)  
> # now we compute mse. We evaluate the inner summation for  
> # each treatment. sumSq is a custom function. Please see  
> # the markdown document for details  
> inner <- aggregate(Backspin ~ Ball, data = spin, FUN = sumSq)[,  
 2]  
> # now we add those together and divide by N-t  
> mse <- sum(inner)/(N - t)  
> # to calculate our observed F statistic we take MST/MSE  
> F.obs <- mst/mse  
> # now we need the critical value for our F-statistic  
> nu1 <- t - 1  
> nu2 <- N - t  
> alpha <- 0.05  
> F.crit <- qf(alpha, df1 = nu1, df2 = nu2, lower.tail = F)  
> F.crit

[1] 2.438739

> # now let's calculate the p-value  
> p.val <- pf(F.obs, df1 = nu1, df2 = nu2, lower.tail = F)

Now, let’s take a look at the ANOVA table we just computed (the code is hidden).

Df Sum.Sq Mean.Sq F.value p.value  
Treatment 4 43866042 10966510 53.79 4.87e-27  
Error 135 27522572 203871

Let’s take a look at what R produces and check to see if it matches our output.

> spin$Ball <- factor(spin$Ball)  
> mod.fit <- aov(Backspin ~ Ball, data = spin)  
> summary(mod.fit)

Df Sum Sq Mean Sq F value Pr(>F)   
Ball 4 43866042 10966510 53.79 <2e-16 \*\*\*  
Residuals 135 27522572 203871   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

As we can see, the aov() function in R outputs the same values that we obtained using our own calculations. Now, let’s use the results to make a conclusion for the original hypothesis test.

* For the test statistic, we see that 53.79 2.44, therefore, we reject and conclude there is at least one pair of ball type means that are different.
* For the p-value method 4.87e-27, therefore, we reject and conclude there is at least one pair of ball type means that are different.

**Questions**

* Think about what we have concluded thus far. Do you think our analysis is finished?
* What important questions have been left unanswered? How may we go about answering these questions?

# Multiple Comparisons

After we reject in an ANOVA setting, we must now locate which population means are different. With most research questions, it is not useful to know only that there exists some difference among the means. More often than not we wish to know *which pairs of means are different*. This is the question that multiple comparisons help us to answer.

The definition of multiple comparisons is quite literal. In statistics, multiple comparisons refers to conducting a set of inferences simultaneously. In our example, we are interested in comparing all pairs of means. We now wish to extend the analysis a step further to reach conclusions about which means are higher and which means are lower.

The easiest way to do this in R is to use the pairwise.t.test() function. We see the results below.

> pairwise.t.test(x = spin$Backspin, g = spin$Ball, p.adjust.method = "none",  
 alternative = "two.sided")

Pairwise comparisons using t tests with pooled SD   
  
data: spin$Backspin and spin$Ball   
  
 Q Star Q Star Tour Soft Feel Z Star   
Q Star Tour < 2e-16 - - -   
Soft Feel 0.28 < 2e-16 - -   
Z Star < 2e-16 0.94 < 2e-16 -   
Z Star XV 1.8e-09 5.8e-05 3.0e-07 5.9e-05  
  
P value adjustment method: none

Unsurprisingly, pairwise.t.test() computes p-values for hypothesis tests involving two independent samples. The tests are for all possible combinations of our factor levels. We can use the entries in the output to see which comparisons yield significant differences. However, there is a problem with the way we have performed this test. Notice, we have not adjusted our p-values for the fact that we are performing multiple tests. Therefore, our experimentwise error rate will be inflated.

Let’s take a look at the p-values when using the Bonferroni adjustment.

> pairwise.t.test(x = spin$Backspin, g = spin$Ball, p.adjust.method = "bonferroni",  
 alternative = "two.sided")

Pairwise comparisons using t tests with pooled SD   
  
data: spin$Backspin and spin$Ball   
  
 Q Star Q Star Tour Soft Feel Z Star   
Q Star Tour < 2e-16 - - -   
Soft Feel 1.00000 3.0e-16 - -   
Z Star < 2e-16 1.00000 < 2e-16 -   
Z Star XV 1.8e-08 0.00058 3.0e-06 0.00059  
  
P value adjustment method: bonferroni

Look at what happened to the p-values. The Bonferroni adjustment ensures that our experimentwise error rate will be no larger than our stated level.

While it does not make a difference in this example, it is easy to see that we could very well run into a case where our results are significant (reject means are equal) *without* a multiple comparison adjustment, but they are *not significant* (don’t reject means are equal) once we control for type I error appropriately.

Another popular adjustment that controls experimentwise error appropriately is *Tukey’s Honestly Significant Difference (HSD)* adjustment. There is a built-in function in R which accepts the object from an ANOVA model and performs the adjustment.

> TukeyHSD(x = mod.fit, conf.level = 0.95)

Tukey multiple comparisons of means  
 95% family-wise confidence level  
  
Fit: aov(formula = Backspin ~ Ball, data = spin)  
  
$Ball  
 diff lwr upr p adj  
Q Star Tour-Q Star -1302.694764 -1636.5706 -968.8190 0.0000000  
Soft Feel-Q Star -129.310345 -457.1699 198.5492 0.8111596  
Z Star-Q Star -1293.448276 -1621.3078 -965.5887 0.0000000  
Z Star XV-Q Star -787.466844 -1124.6513 -450.2824 0.0000000  
Soft Feel-Q Star Tour 1173.384419 839.5086 1507.2602 0.0000000  
Z Star-Q Star Tour 9.246488 -324.6293 343.1223 0.9999921  
Z Star XV-Q Star Tour 515.227920 172.1907 858.2651 0.0005476  
Z Star-Soft Feel -1164.137931 -1491.9975 -836.2784 0.0000000  
Z Star XV-Soft Feel -658.156499 -995.3409 -320.9721 0.0000029  
Z Star XV-Z Star 505.981432 168.7970 843.1659 0.0005555

The nice thing about this function is it also provides the confidence intervals. You can see the adjusted p values in the last column.

**Question** - Rank the p-values obtained for the Soft Feel vs. Q Star brands using the three pairwise tests above from most conservative to least conservative (conservative means less likely to reject ). What do you find?

# The emmeans Package

The [emmeans](https://cran.r-project.org/web/packages/emmeans/vignettes/) (Estimated Marginal Means) package is the most comprehensive R package so far for performing multiple comparisons. It has support for many different types of statistical models (beyond ANOVA), dynamic plotting options, and complicated experimental designs (important for STAT 802). Chris believes it will be the standard in the future.

Let’s take a look at some of the functionality of the package. The first thing we have to do is create an emmeans object using the ANOVA model we fit.

> library(emmeans)  
> # the specs specifies the predictors or factors over which  
> # we want means computed  
> calc.est <- emmeans(mod.fit, specs = ~Ball)  
> # by default emmeans displays an optimal amount of digits  
> # based on a criteria which you can read about on the  
> # vignette; we override this with the command below  
> emm\_options(opt.digits = F)  
> # We can obtain confidence intervals for pairwise  
> # comparisons  
> confint(contrast(calc.est, method = "pairwise"), level = 0.95)

contrast estimate SE df lower.CL upper.CL  
 Q Star - Q Star Tour 1302.6948 120.7510 135 968.8190 1636.5706  
 Q Star - Soft Feel 129.3103 118.5751 135 -198.5492 457.1699  
 Q Star - Z Star 1293.4483 118.5751 135 965.5887 1621.3078  
 Q Star - Z Star XV 787.4668 121.9476 135 450.2824 1124.6513  
 Q Star Tour - Soft Feel -1173.3844 120.7510 135 -1507.2602 -839.5086  
 Q Star Tour - Z Star -9.2465 120.7510 135 -343.1223 324.6293  
 Q Star Tour - Z Star XV -515.2279 124.0644 135 -858.2651 -172.1907  
 Soft Feel - Z Star 1164.1379 118.5751 135 836.2784 1491.9975  
 Soft Feel - Z Star XV 658.1565 121.9476 135 320.9721 995.3409  
 Z Star - Z Star XV -505.9814 121.9476 135 -843.1659 -168.7970  
  
Confidence level used: 0.95   
Conf-level adjustment: tukey method for comparing a family of 5 estimates

> # if we want to specify an adjustment other than Tukey we  
> # add the 'adjust =' argument with the name of the  
> # adjustment included  
> # we can also obtain p-values  
> test(contrast(calc.est, method = "pairwise"))

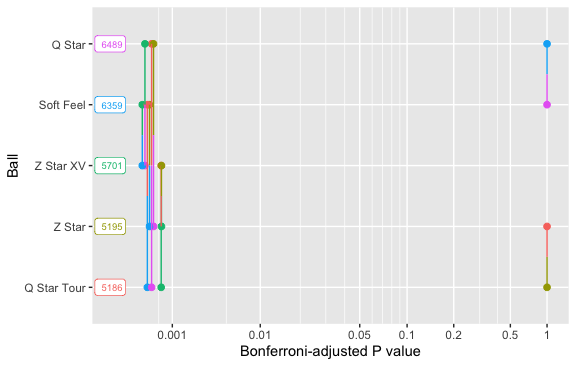
contrast estimate SE df t.ratio p.value  
 Q Star - Q Star Tour 1302.6948 120.7510 135 10.788 <.0001  
 Q Star - Soft Feel 129.3103 118.5751 135 1.091 0.8112  
 Q Star - Z Star 1293.4483 118.5751 135 10.908 <.0001  
 Q Star - Z Star XV 787.4668 121.9476 135 6.457 <.0001  
 Q Star Tour - Soft Feel -1173.3844 120.7510 135 -9.717 <.0001  
 Q Star Tour - Z Star -9.2465 120.7510 135 -0.077 1.0000  
 Q Star Tour - Z Star XV -515.2279 124.0644 135 -4.153 0.0005  
 Soft Feel - Z Star 1164.1379 118.5751 135 9.818 <.0001  
 Soft Feel - Z Star XV 658.1565 121.9476 135 5.397 <.0001  
 Z Star - Z Star XV -505.9814 121.9476 135 -4.149 0.0006  
  
P value adjustment: tukey method for comparing a family of 5 estimates

**Question** - For which pairs of means are there significant differences? How do you know?

Notice that the default adjustment is Tukey’s HSD. In fact, six different multiple comparison methods are supported and they can be found in the documentation. Now, let’s take a look at some of the graphical options.

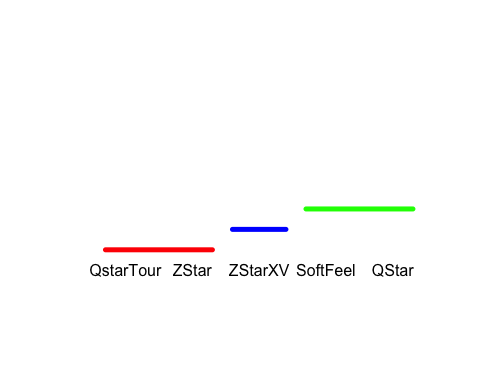
We can use the package to obtain visuals of the results. The plotting mechanism utilizes the R package ggplot2, so you must install this package if you wish to create the plots shown. The first plot is the “pairwise P-value plot” shown below

> pwpp(calc.est, adjust = "bonferroni")



Here, we see the means for each factor level sorted on the y axis. The p-value for each pairwise comparison is denoted by the horizontal position of the vertical line segment along the x axis. The colors and positions of the dots on the vertical line segments indicate the groups being compared. You can view the other plotting options in the documentation.

We can also create a plot similar to what we see in the notes, but this is not part of the package.



**Exploration**

* Perform ANOVA on the carry distance data from the week 10 lab (you may use the aov() function and you can skip the computations of the mst and mse).
* Regardless of your findings for the ANOVA, use the emmeans package to find confidence intervals for the pairwise comparisons. Use the Bonferroni adjustment.
* Produce the pairwise p-value plot from the emmeans package. Explain and interpret the visualization.