

Lab03 - wrapping up t tests and confidence intervals for ANOVA

Your Name Here

Goals

The goal in this lab is to understand:

1. how the output from the `fit.contrast` function relates to all the formulas we discussed on Friday
2. how those results are used to find the t statistic for a hypothesis test
3. how those results are used to find a confidence interval

Loading packages

Here are some packages with functionality you may need for this lab. Run this code chunk now.

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##   combine
```

```
library(mosaic)
```

```
## Loading required package: lattice

## Loading required package: ggformula

##
## New to ggformula? Try the tutorials:
##   learnr::run_tutorial("introduction", package = "ggformula")
##   learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features. The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum

library(gmodels)

options("pillar.sigfig" = 10) # print 10 significant digits in summarize output
```

Example Study (Case Study 6.1.1 in Sleuth 3, description excerpted from the book)

A 1990 study conducted a randomized experiment to explore how physical handicaps affect people's perception of employment qualifications. The researchers prepared five videotaped job interviews using the same two male actors for each. A set script was designed to reflect an interview with an applicant of average qualifications. The videos differed only in that the applicant appeared with a different handicap:

1. in one, he appeared to have no handicap;
2. in a second, he appeared to have one leg amputated;
3. in a third, he appeared on crutches;
4. in a fourth, he appeared to have impaired hearing;
5. and in a fifth, he appeared in a wheelchair.

Seventy undergraduate students from a US university were randomly assigned to view the videos, fourteen to each video. After viewing their video, each subject rated the qualifications of the applicant on a 0 to 10 point applicant qualification scale.

The following R code reads in a data set with the results. By setting the levels of the Handicap variable in this data set, I have told R to use the ordering of the groups outlined above instead of alphabetic order.

```
handicaps <- read_csv("http://www.evanlray.com/data/sleuth3/ex0601_handicaps.csv") %>%
  mutate(
```

```

    Handicap = factor(Handicap, levels = c("None", "Amputee", "Crutches", "Hearing", "Wheelchair"))
  )

## Parsed with column specification:
## cols(
##   Score = col_double(),
##   Handicap = col_character()
## )

dim(handicaps)

## [1] 70  2

head(handicaps)

## # A tibble: 6 x 2
##       Score Handicap
##       <dbl> <fct>
## 1 1.9      None
## 2 2.5      None
## 3 3        None
## 4 3.6      None
## 5 4.100000000 None
## 6 4.2      None

```

1. Exploration of sample data

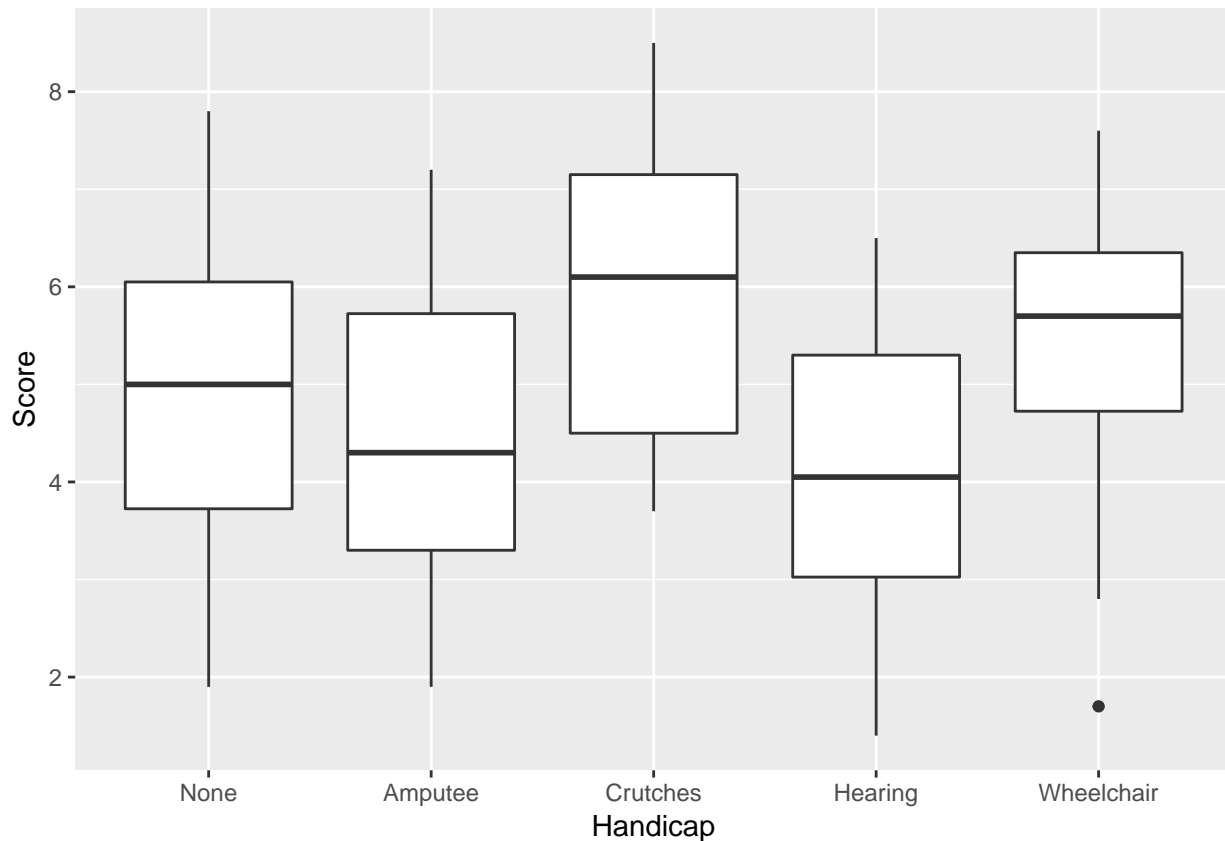
(a) Plot

Use the space below to make a plot comparing applicant qualification scores for the different handicaps. (No need to make every possible plot, one will do.)

```

ggplot(data = handicaps, mapping = aes(x = Handicap, y = Score)) +
  geom_boxplot()

```



(b) Sample means in each group

Use the space below to find the sample mean score for each handicap.

```
handicaps %>%
  group_by(Handicap) %>%
  summarize(
    mean_score = mean(Score)
  )
```

```
## # A tibble: 5 x 2
##   Handicap    mean_score
##   <fct>      <dbl>
## 1 None      4.900000000
## 2 Amputee   4.428571429
## 3 Crutches  5.921428571
## 4 Hearing    4.05
## 5 Wheelchair 5.342857143
```

2. Hypothesis test set up

Suppose we want to conduct a test of whether or not the average qualification score when the applicant appears to have a hearing impairment is the same as or different than the average qualification score when the applicant is using an assistive device for mobility (crutches or a wheelchair). Express the null and alternative hypotheses for the test as a linear combination of the means μ_1 through μ_5 . What are the coefficients for each mean?

Note: our book uses the symbol γ (gamma) to represent this combination of population means; we can think of this as a parameter to estimate.

Our null hypothesis will be that the mean for the hearing impairment group is equal to the average of the means for the crutches group and the wheelchair group. In symbols,

$$\mu_4 = \frac{1}{2}(\mu_3 + \mu_5)$$

We need to get this in a form where a linear combination of the five means is equal to 0. We can do this in either of two equally valid ways: by subtracting μ_4 from both sides of the equation above, or by subtracting $\frac{1}{2}(\mu_3 + \mu_5)$ from both sides of the equation above. Either way, there is a hidden $0\mu_1$ and a hidden $0\mu_2$ that I will add in, and the alternative hypothesis will be that the resulting equation does not hold. So your answer to this part could be either of the following two options:

Option 1:

$$H_0 : 0\mu_1 + 0\mu_2 - \frac{1}{2}\mu_3 + 1\mu_4 - \frac{1}{2}\mu_5 = 0$$

$$H_A : 0\mu_1 + 0\mu_2 - \frac{1}{2}\mu_3 + 1\mu_4 - \frac{1}{2}\mu_5 \neq 0$$

Option 2:

$$H_0 : 0\mu_1 + 0\mu_2 + \frac{1}{2}\mu_3 + (-1)\mu_4 + \frac{1}{2}\mu_5 = 0$$

$$H_A : 0\mu_1 + 0\mu_2 + \frac{1}{2}\mu_3 + (-1)\mu_4 + \frac{1}{2}\mu_5 \neq 0$$

Again, either of these is fine. I will proceed using option 2 for the rest of this document.

3. Conduct the hypothesis test and confidence interval calculations using `lm` and `fit.contrast`

Use the `fit.contrast` function in R to estimate γ , obtain a p-value for the test you set up in part (a), and find a 95% confidence interval for γ . (Here, your argument should be the actual level of the confidence interval that you want to calculate, 0.95 in this case.)

```
anova_fit <- lm(Score ~ Handicap, data = handicaps)
fit.contrast(anova_fit, "Handicap", c(0, 0, 0.5, -1, 0.5), conf.int = 0.95)
```

```
##                                Estimate Std. Error t value    Pr(>|t|)
## Handicap c=( 0 0 0.5 -1 0.5 ) 1.582143   0.5345041  2.96002 0.004289282
##                                lower CI upper CI
## Handicap c=( 0 0 0.5 -1 0.5 ) 0.5146644 2.649621
```

4. Verify estimate

We estimate γ (a linear combination of “population” means) with the corresponding linear combination of sample means. This estimate is referred to as g in our textbook. Using your results from 1 (b), verify that the estimate calculated by `fit.contrast` in part 3 makes sense.

We estimate the linear combination of population means specified in the null hypothesis with the corresponding linear combination of sample means:

```
0.5 * 5.921429 - 1 * 4.050000 + 0.5 * 5.342857
```

```
## [1] 1.582143
```

Note that this agrees with the “Estimate” in the output from `fit.contrast` in part 3.

5. Verify calculation of t statistic

The general form of a t statistic is

$$\frac{\text{Estimate} - \text{Parameter Value from Null Hypothesis}}{SE(\text{Estimate})}$$

In this case, that works out to

$$\frac{g - \gamma^{null}}{SE(g)}$$

$SE(g)$ is the standard error of g : an estimate of the variability of the statistic g across all possible samples. This standard error is calculated using the formula we discussed last time, $SE(g) = s_{pooled} \sqrt{\frac{C_1^2}{n_1} + \dots + \frac{C_I^2}{n_I}}$. We don't need to reproduce that calculation; we will just let the computer do that and make sure we understand how it is used for confidence intervals and hypothesis tests.

Taking $SE(g)$ as given in the output from `fit.contrast`, verify that the t statistic shown in that output is correct. What are the degrees of freedom for the t statistic?

In our hypothesis test, the null hypothesis was that the specified linear combination of means was equal to 0. Therefore $\gamma^{null} = 0$. g is the estimate, which came out to be 1.582 in part 4. From the output of `fit.contrast` in part 3, the standard error of g is about 0.535. Plugging these into the formula for the t statistic, we obtain a t statistic of approximately

```
1.582/0.535
```

```
## [1] 2.957009
```

This is pretty close to the t statistic of 2.96002 listed in the output from `fit.contrast` in part 3.

Our total sample size in this case was 70, and there were 5 groups in the data set. The degrees of freedom for the t statistic is therefore $70 - 5 = 65$.

6. Verify calculation of p-value.

The p-value for the test is calculated as the probability of obtaining a t statistic at least as extreme as the value calculated in part 5. We can use the `pt` function in R to calculate this probability as in the handout.

Since we specified a two-sided test, the p-value is the probability of obtaining a t statistic less than or equal to -2.96, or greater than or equal to 2.96. This is calculated using the `pt` function as follows:

```
pt(-2.96, df = 65) + pt(2.96, df = 65, lower.tail = FALSE)
```

```
## [1] 0.004289525
```

This agrees quite closely with the p-value of 0.004289282 given in the output from `fit.contrast` in part 3.

7. Verify calculation of confidence interval.

The generic formula for a confidence interval is

$$[\text{Estimate} - t^*SE(\text{Estimate}), \text{Estimate} + t^*SE(\text{Estimate})]$$

Using appropriate output from `qt` as well as the estimate and the standard error of the estimate as given in the output from `fit.contrast`, verify the confidence interval bounds given in the output from `fit.contrast`.

```
qt(0.975, df = 65)
```

```
## [1] 1.997138
```

```
1.582143 - 1.997 * 0.535
```

```
## [1] 0.513748
```

```
1.582143 + 1.997 * 0.535
```

```
## [1] 2.650538
```

The confidence interval of [0.514, 2.651] calculated here agrees fairly closely with the confidence interval of [0.5146644, 2.649621] given in the output from `fit.contrast` in question 3.