Lab 17 - Post-Hoc Tests

Nick Sumpter (Edited by Eddie-Williams Owiredu & Guy Twa)

2023-10-18

Today's Lab

ANOVAs are referred to as omnibus tests, and as such finding a significant ANOVA main effect (or interaction) only tells you that there is a difference in means across all groups (or that the effect of one variable depends on the other). To find out which specific combinations of groups differ (or to interpret an interaction), it is necessary to perform post-hoc comparisons. These tests determine whether there is a significant difference between specific pairs of levels of different grouping variables.

The basic form of post-hoc analysis is performing pairwise t-tests for each combination of groups and adjusting the critical p-value based on the number of tests performed. The adjustment methods we will be using are Bonferroni and Holm's. This will be done semi-manually. We will also cover Tukey's Honestly Significant Difference, an omnibus multiple comparison procedure that performs all multiple comparisons at once in R (only for between-subjects ANOVA).

Loading Packages and Data

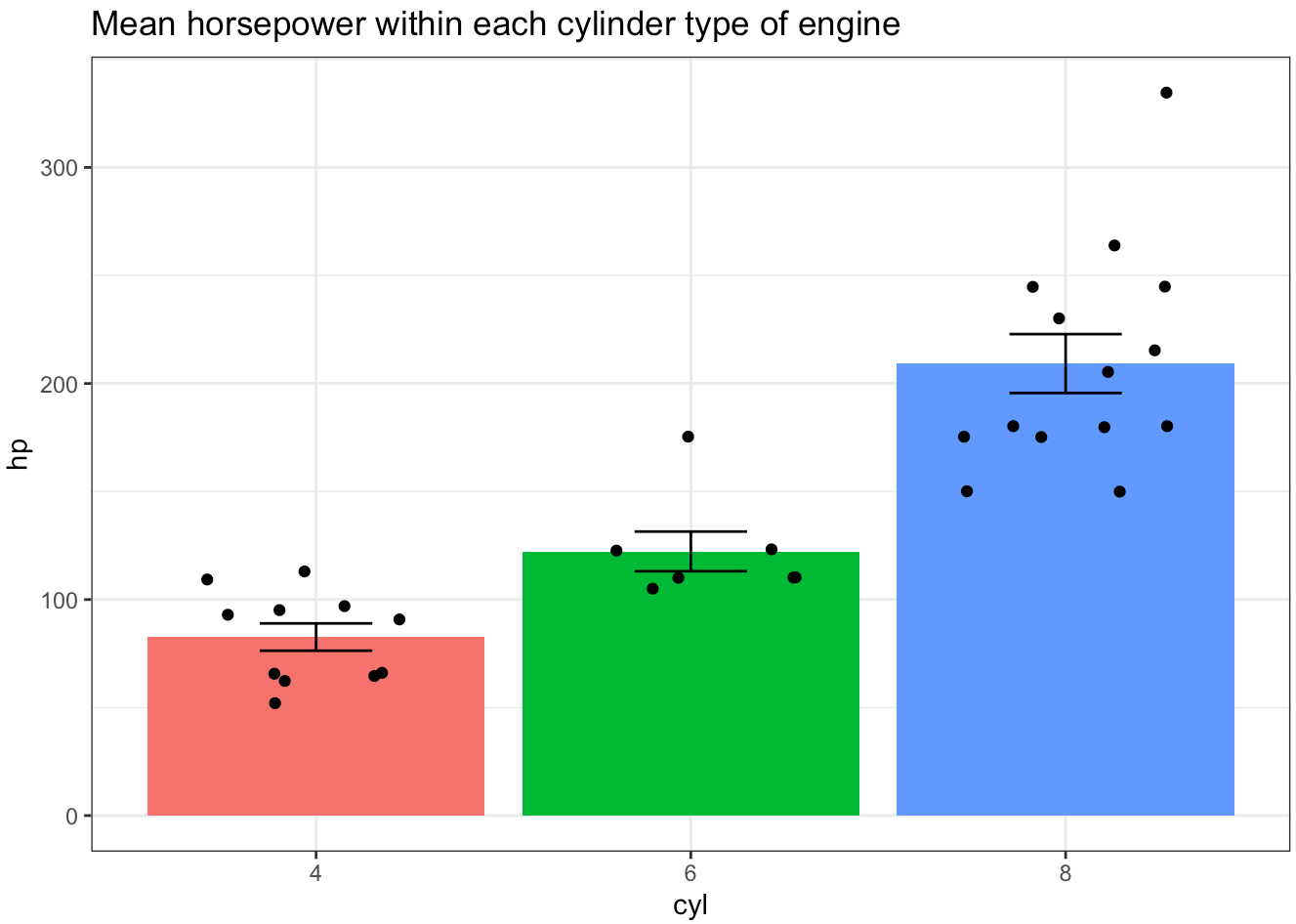
We will be using the `mtcars` dataset that is built into R, then you will use the dataset stored in `Post-Hoc-Data.RData` for your independent practice.

```
library(ez)
library(dae)
library(car)
library(pastecs)
library(sciplot)
library(tidyverse)

theme_set(theme_bw())

setwd("/Users/eddie-williamsowiredu/Desktop/grd770_23/Lab17")

mtcars <- mtcars

load("Post-Hoc-Data.RData")

source("functions.R")
```

Post-Hoc Analysis of a Significant Main Effect

We will perform a post-hoc analysis on the results of the one-way independent ANOVA from lab 12. As a reminder, we found a significant difference in mean horsepower between engines with different numbers of cylinders (`hp` vs `cyl` in the `mtcars` dataset). We can visualize this main effect as follows:

```
mtcars2 <- mtcars %>%
  mutate(cyl = factor(cyl))

ggplot(data = mtcars2, mapping = aes(x = cyl, y = hp)) +
  geom_bar(mapping = aes(fill = cyl), stat = "summary", fun = "mean", show.legend = FALS
E) +
  geom_errorbar(stat = "summary", fun.data = "mean_se", width = 0.3) +
  geom_jitter(width = 0.3) +
  labs(title = "Mean horsepower within each cylinder type of engine") +
  theme_bw()
```



So, as you can see, the horsepower seems to be higher in engines with higher numbers of cylinders, but we need to compare specific group combinations using post-hoc tests to confirm this.

Running the Post-Hoc Analysis

We need to determine which cylinder groups show a significant difference between each other. As there are more than 2 groups in our grouping variable, we will need to run a post-hoc analysis. If there were only 2 groups, we would already know which groups were different (as there is only one pairwise comparison). We will use the `pairwise.t.test` function to test for pairwise differences in `hp` between levels of `cyl`. Look this function up in the Help panel and you will see that it takes on several arguments. The arguments we are interested in are:

```
`x` = the outcome variable

`g` = the grouping variable

`paired` = set to FALSE if the levels are independent

`p.adjust.method` = either "none", "holm", or "bonferroni" indicating which method you w
ish to use for adjusting p-values
```

Let's go ahead and run the `pairwise.t.test` function on our data three times: with no adjustment, Holm correction, and Bonferroni correction respectively.

```
pairwise.t.test(x = mtcars$hp, g = mtcars$cyl, p.adjust.method = "none", paired = FALSE)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mtcars$hp and mtcars$cyl
##
##   4       6
## 6 0.039   -
## 8 3.9e-09 2.9e-05
##
## P value adjustment method: none
```

```
pairwise.t.test(x = mtcars$hp, g = mtcars$cyl, p.adjust.method = "holm", paired = FALSE)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mtcars$hp and mtcars$cyl
##
##   4       6
## 6 0.039   -
## 8 1.2e-08 5.8e-05
##
## P value adjustment method: holm
```

```
pairwise.t.test(x = mtcars$hp, g = mtcars$cyl, p.adjust.method = "bonferroni", paired =
FALSE)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mtcars$hp and mtcars$cyl
##
##   4       6
## 6 0.12    -
## 8 1.2e-08 8.7e-05
##
## P value adjustment method: bonferroni
```

As you can see, the output of each of these functions shows a table with the levels of the grouping variable as the row and column names. The values in each of these cells is the **adjusted** p-value. In the first model, we didn't adjust our p-values and thus these are the raw p-values from running an independent t-test on each pair of `cyl` levels. In the following two functions, we have run a form of multiple testing correction on our p-values (either Holm or Bonferroni). These two forms of p-value adjustment are explained below:

Bonferroni vs Holm Correction

Multiple testing corrections are important for controlling family-wise error rate. With each subsequent t-test being run, there is a higher chance of one of them being a false positive (or type-1 error). We therefore need to adjust our alpha-value (significance threshold) to make it so we still only have a 0.05 probability of a false positive. Holm and Bonferroni both account for this in a slightly different way, with Bonferroni being the more conservative method of the two.

Bonferroni correction divides the alpha-value by the number of tests being run, so you need a p-value several times lower to reach significance.

Holm correction is similar, however it does this sequentially, with only the test with the lowest p-value requiring this stringent of a p-value, and the next highest p-value only has to meet an alpha-value of 0.05 / number of tests - 1. This process continues until the highest p-value which is simply compared to 0.05.

In the `pairwise_t_test` function, the reported p-value has already been adjusted such that a reported p-value of < 0.05 is significant under that version of multiple testing correction.

In the above examples, you will notice that, for the Holm corrected p-values, the highest p-value (0.039) doesn't change compared to the uncorrected p-value, the second highest p-value is double that of the uncorrected p-value, and the lowest p-value is triple that of the uncorrected p-value. When looking at the Bonferroni corrected p-values, you will notice that all three p-values are exactly 3 times larger than the uncorrected p-values, and thus a corrected p-value of 0.05 will be equivalent to a raw p-value of 0.05 / 3 (the number of tests).

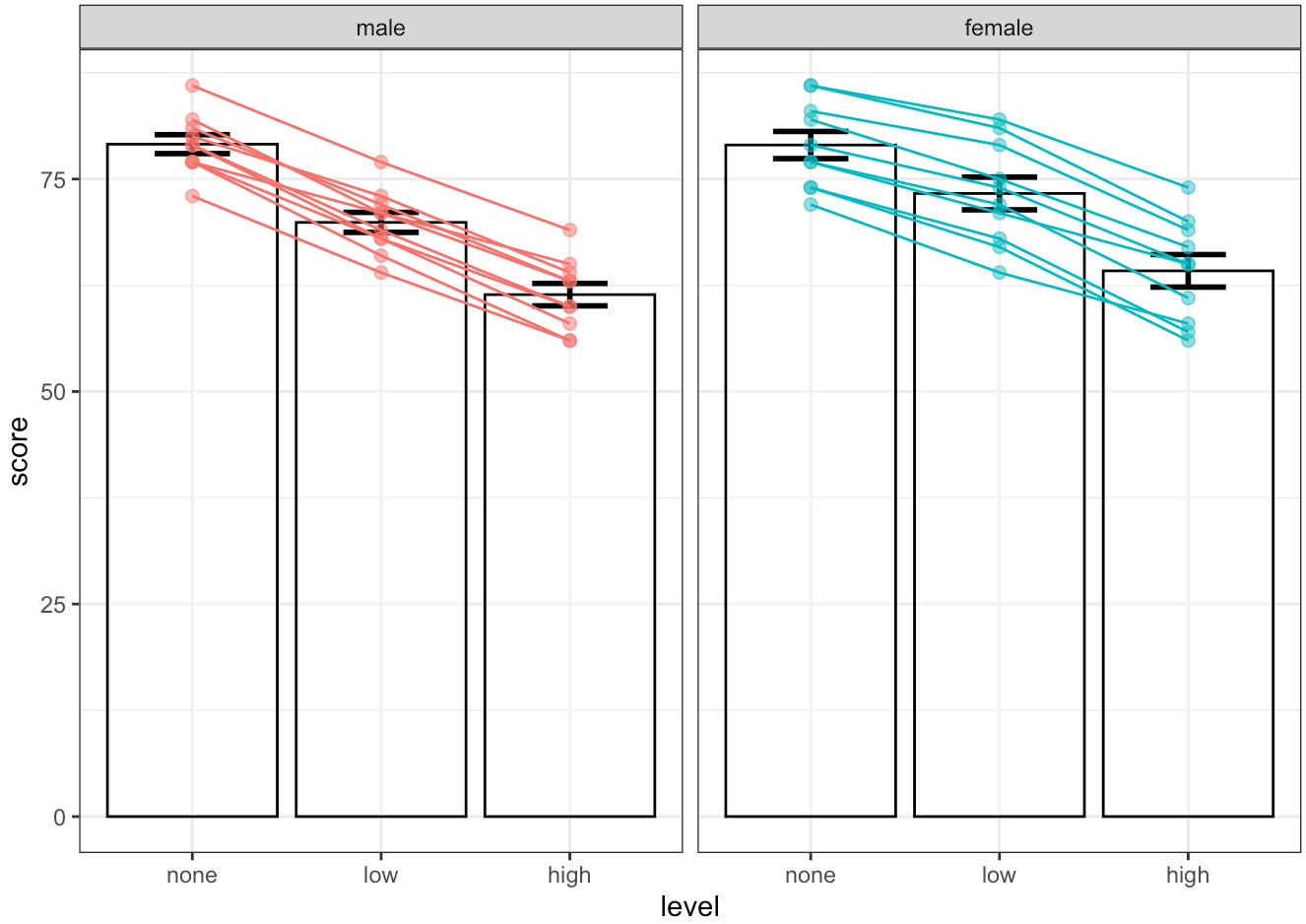
We will go over reporting these results at the end of the lab.

Post-Hoc Analysis of a Significant Interaction

When you have a significant interaction term it becomes a fair amount more complicated to both test and report post-hoc comparisons. As a reminder, as soon as you have a significant interaction, the main effects essentially become far less important, as the effect of one variable now depends on the level of the other variable. This will be especially apparent when observing the results of post-hoc analyses.

For an example of an ANOVA with a significant interaction, we will be using the same mixed ANOVA analysis as last lab. There was a significant main effect of noise level but not gender on a perceptual task. However, the significant interaction told us that the effect of noise depends on the gender of the individual. We can visualize this as follows:

```
noise_long <- noise %>%
  pivot_longer(cols = none:high, names_to = "level", values_to = "score") %>%
  mutate(level = factor(level, levels = c("none", "low", "high"), ordered = TRUE))

ggplot(data = noise_long, mapping = aes(x = level, y = score, color = gender)) +
  geom_bar(stat = "summary", fun = mean, fill = "white", color = 'black', alpha = 0.4) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.4, color = 'black', size
= 1) +
  geom_point(size = 2, alpha = 0.5, show.legend = FALSE) +
  geom_line(mapping = aes(group = ID), show.legend = FALSE) +
  facet_wrap(~ gender)
```



So it looks like the effect of increasing noise level has a slightly greater effect in males vs females. We can use post-hoc tests to confirm this.

Running the Post-Hoc Analysis - pairwise.t.test method

As mentioned previously, it is not as simple to run a post-hoc analysis on a significant interaction. Basically, we are going to have to do the following:

1. Subset the dataset based on all levels of either grouping variable
2. Perform pairwise t-tests **on the non-subsetted grouping variable** in each of these subsets

   This will enable us to compare the effect of one variable on the outcome when the other variable is kept constant
3. Manually calculate the Bonferroni/Holm corrected p-values in order to assess significance

So in our case, our two grouping variables are `level` and `gender`, with 3 and 2 levels respectively. We therefore need to create 5 different subsets (3 + 2) of the `noise_long` dataset in order to test our pairwise comparisons for the significant interaction effect:

```
noise_long_none <- noise_long %>%
  filter(level == "none")

noise_long_low <- noise_long %>%
  filter(level == "low")

noise_long_high <- noise_long %>%
  filter(level == "high")

noise_long_male <- noise_long %>%
  filter(gender == "male")

noise_long_female <- noise_long %>%
  filter(gender == "female")
```

Within each of our newly created datasets, lets run the `pairwise.t.test` function and save the output, remembering to not adjust the p-values as we will do this manually:

Note: **If you have a repeated-measures variable, make sure to set paired = TRUE for the functions where the g argument =**

```
none_test <- pairwise.t.test(x = noise_long_none$score, g = noise_long_none$gender, p.ad
just.method = "none", paired = TRUE)

low_test <- pairwise.t.test(x = noise_long_low$score, g = noise_long_low$gender, p.adjus
t.method = "none", paired = TRUE)

high_test <- pairwise.t.test(x = noise_long_high$score, g = noise_long_high$gender, p.ad
just.method = "none", paired = TRUE)

male_test <- pairwise.t.test(x = noise_long_male$score, g = noise_long_male$level, p.adj
ust.method = "none", paired = FALSE)

female_test <- pairwise.t.test(x = noise_long_female$score, g = noise_long_female$level,
p.adjust.method = "none", paired = FALSE)
```

Now we need to extract all of the p-values from each of these and combine them together using the `c` function. To help you understand the next steps, let's quickly look at how we extract p-values from the results of a pairwise t-test:

```
# First for any comparison of just two groups, such as for males vs females in a single
noise level:
none_test # This is the complete output from the test
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  noise_long_none$score and noise_long_none$gender
##
##        male
## female 0.96
##
## P value adjustment method: none
```

```
none_test$p.value # This is the p.value itself with the column and row names
```

```
##             male
## female 0.9640421
```

```
as.numeric(none_test$p.value) # This is the p.value itself without the column and row na
mes
```

```
## [1] 0.9640421
```

```
# Now for a scenario comparing three groups, let's look at the results of the comparison
s between noise levels within males only:
male_test # This is the complete output from the test
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  noise_long_male$score and noise_long_male$level
##
##      none    low
## low  1.0e-05 -
## high 6.3e-11 3.2e-05
##
## P value adjustment method: none
```

```
male_test$p.value # This is the p.values in a 2x2 table with the column and row names, n
ote that one comparison is NA as it is comparing the low noise group with itself
```

```
##              none          low
## low  1.048393e-05           NA
## high 6.280810e-11 3.154557e-05
```

```
as.numeric(male_test$p.value) # This is the list of p.values from the above table withou
t the column and row names - note that it outputs the first column and then the second c
olumn, so in this case it is ordered: none-low, none-high, low-low, and low-high
```

```
## [1] 1.048393e-05 6.280810e-11           NA 3.154557e-05
```

So now that we know how to extract the p-values and the order that they are in, we can combine them all together into a complete list using the `c` function. Pay attention to the order you input the p-values, as this will matter for the next step:

```
all_pval <- c(as.numeric(none_test$p.value),
              as.numeric(low_test$p.value),
              as.numeric(high_test$p.value),
              as.numeric(male_test$p.value),
              as.numeric(female_test$p.value))
```

Finally, we will create a tibble with the first three columns describing the pairwise comparison of interest and the fourth containing the associated p-value. To make this easier, just use the first letter of each variable (i.e. none = n, low = l, high = h, male = m, female = f), though make sure you can distinguish between each variable this way. You should be able to see each combination of tests that was run by reading down the three letters, for example, the first combination is “n”, “m”, “f” which denotes the difference between males and females within the none noise level. The str_split function is just a way to make it easier to input the data, all you need to do is type all of the letters with a space between them.

```
output_table <- tibble(Subset = str_split("n l h m m m m f f f f", pattern = " ")[[1]],
                       Group1 = str_split("m m m n n l l n n l l", pattern = " ")[[1]],
                       Group2 = str_split("f f f l h l h l h l h", pattern = " ")[[1]],
                       Pval = all_pval)
```

We will then do the following (note that you shouldn’t have to change any of this code as long as you followed the above steps correctly):

1. Remove rows with NA for their p-value (these were just leftover from the method for extracting p-values)
2. Arrange the rows by ascending p-value
3. Create a column for the Bonferroni critical p-value (0.05 divided by the number of rows)
4. Create a column that displays TRUE for unadjusted p-values that are less than the Bonferroni critical p-value
5. Create a column with the rank of each p-value (smallest p-value = 1)
6. Create a column with the appropriate Holm critical p-value (Bonferroni critical p-value multiplied by rank)
7. Create a column that displays TRUE for p-values that are less than the Holm critical p-value

```
output_table2 <- output_table %>%
  filter(!is.na(Pval)) %>%
  arrange(Pval) %>%
  mutate(Bonf_P = 0.05 / nrow(.),
         Bonf_sig = Pval < Bonf_P,
         Rank = 1:nrow(.),
         Holm_P = Bonf_P * Rank,
         Holm_sig = Pval < Holm_P)

output_table2
```

```
## # A tibble: 9 × 9
##   Subset Group1 Group2     Pval  Bonf_P Bonf_sig  Rank  Holm_P Holm_sig
##   <chr>  <chr>  <chr>     <dbl>   <dbl> <lgl>    <int>   <dbl> <lgl>
## 1 m      n      h      6.28e-11 0.00556 TRUE         1 0.00556 TRUE
## 2 f      n      h      4.01e- 6 0.00556 TRUE         2 0.0111  TRUE
## 3 m      n      l      1.05e- 5 0.00556 TRUE         3 0.0167  TRUE
## 4 m      l      h      3.15e- 5 0.00556 TRUE         4 0.0222  TRUE
## 5 f      l      h      1.47e- 3 0.00556 TRUE         5 0.0278  TRUE
## 6 f      n      l      3.52e- 2 0.00556 FALSE        6 0.0333  FALSE
## 7 l      m      f      2.07e- 1 0.00556 FALSE        7 0.0389  FALSE
## 8 h      m      f      2.95e- 1 0.00556 FALSE        8 0.0444  FALSE
## 9 n      m      f      9.64e- 1 0.00556 FALSE        9 0.05    FALSE
```

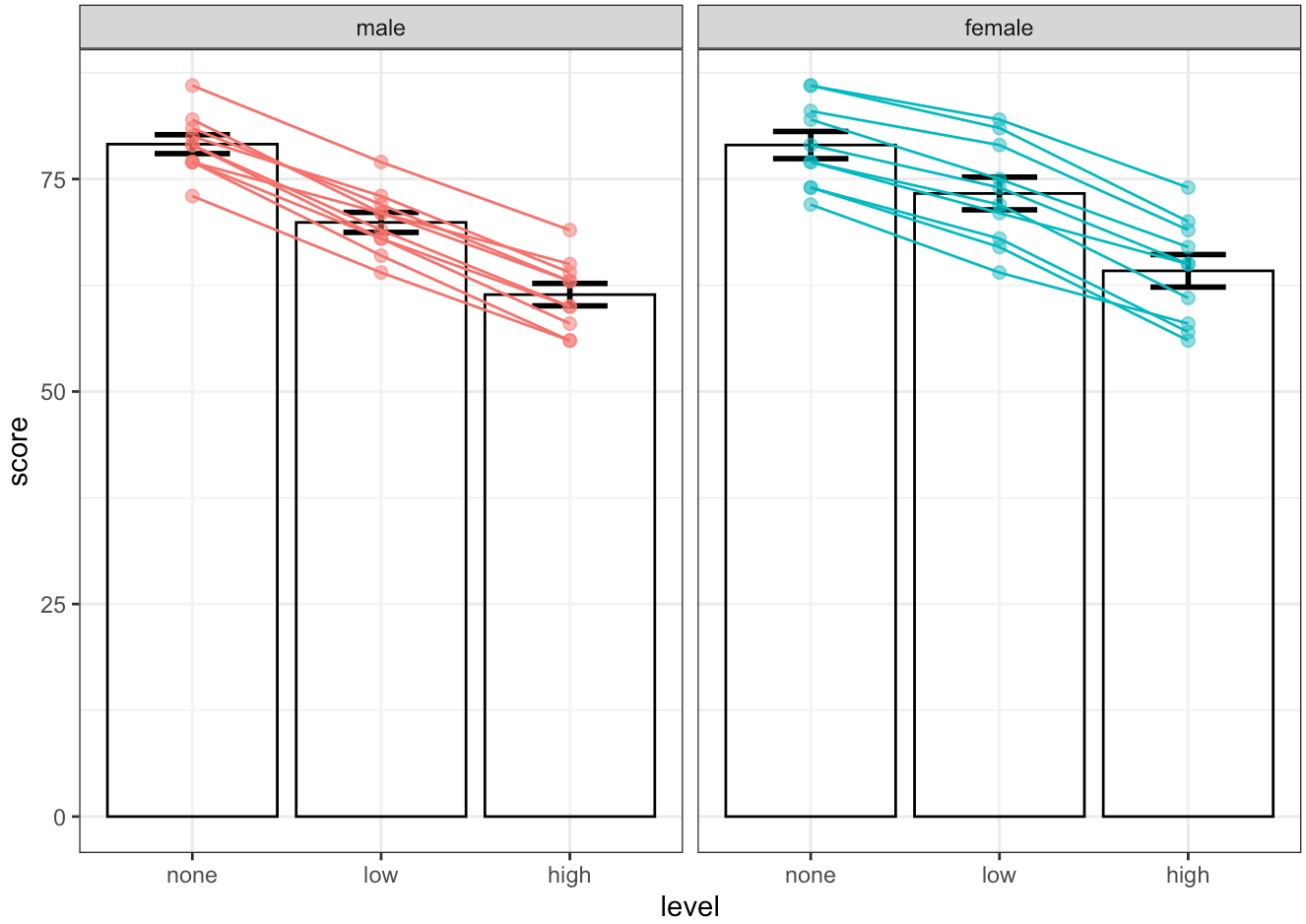
Using either method, 5 of the 9 comparisons of interest meet significance after multiple testing correction. For this table:

1. The three least significant results were for male vs female comparisons within each noise level (reflecting the non-significant main effect of gender)
2. All combinations of noise levels within males were significantly different from each other
3. Only two of the three combinations of noise levels within females were significantly different from each other (none and low were not significantly different)

Points 2 and 3 enable us to interpret the significant interaction effect. Essentially, the mean score at each noise level is consistently different in males but not consistently different in females. Therefore, the effect of noise level on score **depends on** whether the individual is male or female.

We can confirm this by referring back to the figure, which will also enable us to determine the direction of these differences.

```
ggplot(data = noise_long, mapping = aes(x = level, y = score, color = gender)) +
  geom_bar(stat = "summary", fun = mean, fill = "white", color = 'black', alpha = 0.4) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.4, color = 'black', size
= 1) +
  geom_point(size = 2, alpha = 0.5, show.legend = FALSE) +
  geom_line(mapping = aes(group = ID), show.legend = FALSE) +
  facet_wrap(~ gender)
```

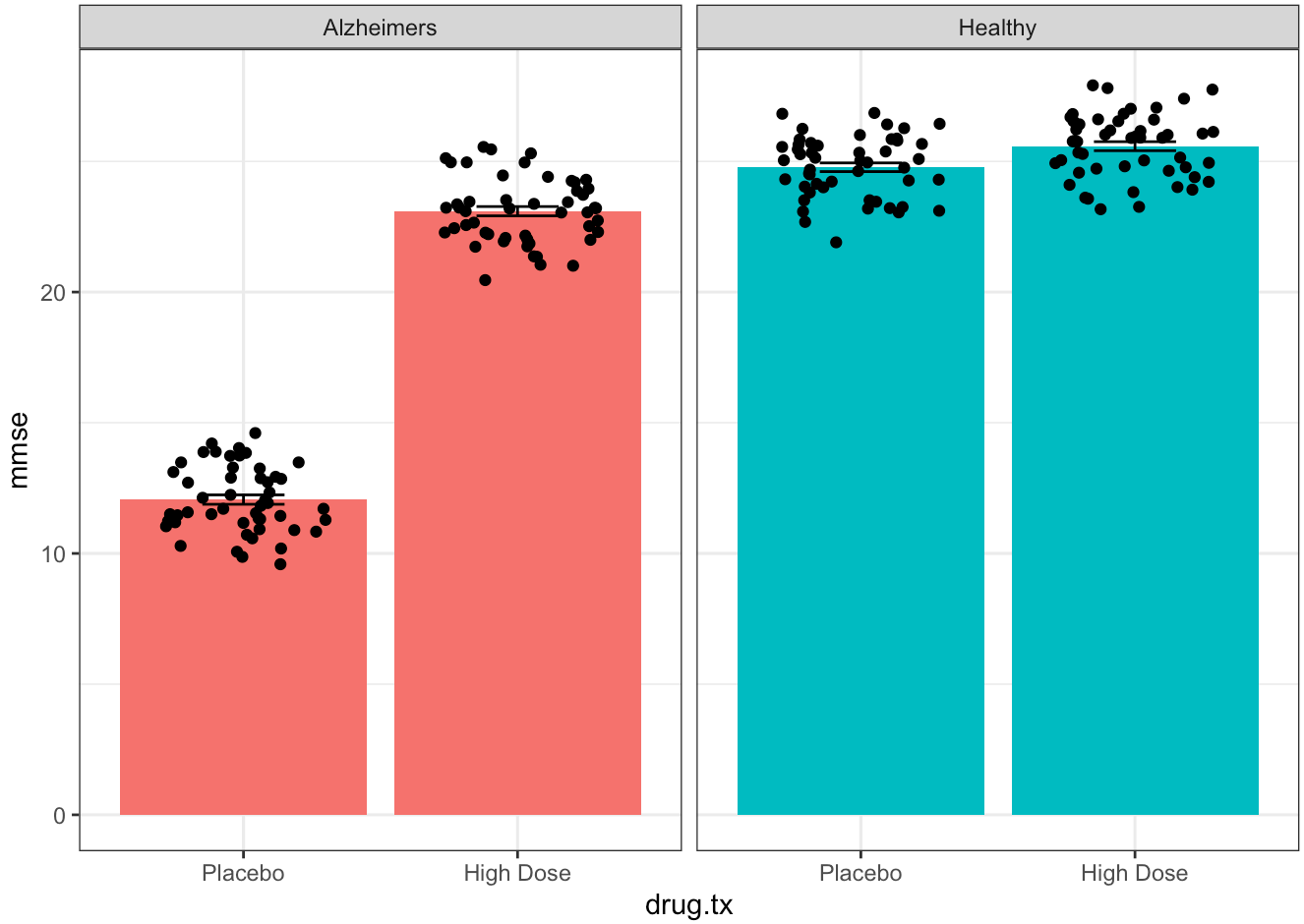


So we can see that the mean score at each noise level is clearly different in males. In females however, the none and low noise levels seem to exhibit a less obvious difference, which was reflected by the non-significant pairwise post-hoc t-test.

Running the Post-Hoc Analysis - Tukey's HSD method

For any **independent** ANOVA, an alternative way to perform a post-hoc analysis is using the Tukey's Honestly Significant Difference method. For example, let's look at the significant interaction effect from the factorial ANOVA lab. As a reminder, we found that the effect of a drug on mmse score depended on health status of the individual. We can see that in the following figure: **data from lab 15**

```
ggplot(data = ad, mapping = aes(x = drug.tx, y = mmse)) +
  geom_bar(mapping = aes(fill = health), stat = "summary", fun = "mean", show.legend = F
ALSE) +
  geom_errorbar(stat = "summary", fun.data = "mean_se", width = 0.3) +
  geom_jitter(width = 0.3) +
  facet_wrap(~ health)
```



Now, we can use the `TukeyHSD` function to run corrected pairwise comparisons on the ANOVA model for this interaction:

```
mod <- ezANOVA(data = ad,
               dv = mmse,
               between = .(health, drug.tx),
               wid = ID,
               type = 3,
               return_aov = TRUE)

mod_tukey <- TukeyHSD(mod$aov)

mod_tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = formula(aov_formula), data = data)
##
## $health
##                          diff      lwr      upr p adj
## Healthy-Alzheimers 7.601359 7.258331 7.944387     0
##
## $drug.tx
##                       diff      lwr      upr p adj
## High Dose-Placebo 5.919349 5.576321 6.262377     0
##
## $`health:drug.tx`
##                                                diff        lwr       upr
## Healthy:Placebo-Alzheimers:Placebo      12.7152020 12.0778069 13.352597
## Alzheimers:High Dose-Alzheimers:Placebo 11.0331919 10.3957968 11.670587
## Healthy:High Dose-Alzheimers:Placebo    13.5207072 12.8833121 14.158102
## Alzheimers:High Dose-Healthy:Placebo    -1.6820101 -2.3194052 -1.044615
## Healthy:High Dose-Healthy:Placebo        0.8055052  0.1681101  1.442900
## Healthy:High Dose-Alzheimers:High Dose   2.4875153  1.8501202  3.124910
##                                             p adj
## Healthy:Placebo-Alzheimers:Placebo      0.0000000
## Alzheimers:High Dose-Alzheimers:Placebo 0.0000000
## Healthy:High Dose-Alzheimers:Placebo    0.0000000
## Alzheimers:High Dose-Healthy:Placebo    0.0000000
## Healthy:High Dose-Healthy:Placebo       0.0068057
## Healthy:High Dose-Alzheimers:High Dose  0.0000000
```

As you can see, you get a large output, which includes three parts:

1. `$health` = pairwise comparisons for the main effect of `health`
2. `$drug.tx` = pairwise comparisons for the main effect of `drug.tx`
3. `$health:drug.tx` = pairwise comparisons for the interaction effect of `health` and `drug.tx`

As we had a significant interaction, we only really care about the third table of the output. As you can see, it lists every possible pairwise comparison for combinations of `health` and `drug.tx`. The comparisons with `p adj` less than 0.05 show a significant difference in means after multiple testing correction. Unfortunately, it is not really possible to remove non-informative comparisons and thus the adjusted p-values are probably more conservative than necessary. In this case, all 6 pairwise comparisons had significant adjusted p-values.

Homogeneous Subsets

Homogeneous subsets is a way of labeling your groups to indicate which groups are significantly different from each other. Unfortunately, R doesn't make this easy and so we will be skipping this for now.

Reporting the Results

First, lets get a summary of the mean and se for each group.

```
ad %>%
  group_by(health, drug.tx) %>%
  summarize(mean = mean(mmse),
            se = se(mmse))
```

```
## # A tibble: 4 × 4
## # Groups:   health [2]
##   health     drug.tx    mean    se
##   <fct>      <ord>     <dbl> <dbl>
## 1 Alzheimers Placebo    12.1 0.180
## 2 Alzheimers High Dose  23.1 0.178
## 3 Healthy    Placebo    24.8 0.166
## 4 Healthy    High Dose  25.6 0.172
```

When reporting results of an ANOVA, you should always report post-hoc analysis results (**including in any exams…**). In general, there are quite a few ways to report post-hoc results. One way would be like so:

“Post-hoc comparisons using Tukey’s Honestly Significant Difference revealed that all combinations of drug treatment and health status were significantly different from each other (mean ± SE for Alzheimers:Placebo = 12.06 ± 0.18, Alzheimers:High Dose = 23.09 ± 0.18, Healthy:Placebo = 24.77 ± 0.17, Healthy:High Dose = 25.6 ± 0.17).”

When reporting Bonferroni or Holm comparison corrections, you can list the comparisons that passed correction without mentioning the comparisons that did not. When reporting Bonferroni corrections, it would be helpful to report the critical p-value along with the raw p-values. When reporting Holm corrections, reporting raw p-values is fine. In both cases, mention which correction method was used. Provide the mean and standard error for the groups like in the above example.

Independent practice

For the exercises for this lab, you should go through each previous ANOVA lab and produce post-hoc analyses for the independent practice examples. Then go ahead and report the complete results section for each of these types of ANOVA.