

Daily Check 8

2025-04-23

R Markdown

Question: Daily Check 8

Answer this question: After running an ANOVA and obtaining a statistically significant p-value, we might want to know exactly WHICH means are different from each other. As discussed in class and in our textbook, why would it be a bad idea to simply do t-tests on each pair of means?

Also complete the two Your Turns, put all of this into an R Markdown file and submit both your pdf output and Rmd files.

Answer this question: After running an ANOVA and obtaining a statistically significant p-value, we might want to know exactly WHICH means are different from each other. As discussed in class and in our textbook, why would it be a bad idea to simply do t-tests on each pair of means?

It is a bad idea to simply do t-tests on each pair of means after a significant ANOVA because doing so does not control the experiment wise error rate. When you perform multiple t-tests, the probability of making at least one Type I error increases with the number of comparisons you make. This means you are more likely to incorrectly conclude that a difference exists when it actually does not. Instead, you should use special multiple comparison procedures that are designed to control the overall Type I error rate when making several pairwise comparisons.

```
google <- c(46, 49, 51, 42)
waze <- c(44, 47, 47, 43)
gut <- c(50, 51, 45, 43)
nav_df <- data.frame(
  time = c(google, waze, gut),
  method = c(rep("google", length(google)),
    rep("waze", length(waze)),
    rep("gut", length(gut)))
)
```

```
pan <- c(12.70, 12.30, 12.40, 12.80, 12.60,
12.50, 12.20, 12.70, 12.40, 12.60)
grilled <- c(11.80, 11.55, 11.75, 11.65, 11.50,
11.75, 11.30, 12.10, 11.50, 11.45)
broiled <- c(11.75, 11.85, 11.60, 11.80, 11.75,
11.65, 11.75, 11.70, 11.85, 11.75)
burgers_df <- data.frame(
  sat_fat = c(pan, grilled, broiled),
  method = c(rep("pan", length(pan)),
    rep("grilled", length(grilled)),
    rep("broiled", length(broiled)))
)
```

```
sample(nav_df$time)
```

```
## [1] 43 45 50 44 51 46 47 51 42 43 49 47
```

```
set.seed(10242024)
nav_perm <- nav_df # temporarily copy the whole df
nav_perm$time <- sample(nav_perm$time)
perm_anova <- aov(time ~ method, data=nav_perm)
summary(perm_anova)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## method         2   45.5    22.75   3.033 0.0984 .
## Residuals      9   67.5     7.50
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
names(perm_anova)
```

```
## [1] "coefficients" "residuals"      "effects"      "rank"
## [5] "fitted.values" "assign"        "qr"           "df.residual"
## [9] "contrasts"     "xlevels"      "call"         "terms"
## [13] "model"
```

```
summary(perm_anova)[[1]][1,4]
```

```
## [1] 3.033333
```

```
# Create a vector to store the F values
perm_f <- NA
# set the number of replications
reps <- 10000
# do it lots of times in a loop
for(i in 1:reps){
  nav_perm <- nav_df
  nav_perm$time <- sample(nav_perm$time)
  perm_f[i] <- summary(aov(time ~ method, data=nav_perm))[[1]][1,4]
}
```

```
summary(aov(time ~ method, data=nav_df))[[1]][1,4]
```

```
## [1] 0.4130435
```

```
F <- summary(aov(time ~ method, data=nav_df))[[1]][1,4]
```

```
sum(perm_f >= F)/reps
```

```
## [1] 0.6527
```

Your Turn #1

```

set.seed(4232025)
data(PlantGrowth)

#Create a vector to store the F values
perm_f <- NA
# set reps
reps <- 10000
#do it lots of times
for(i in 1: reps)
{
  plant_perm <- PlantGrowth
  plant_perm$weight <- sample(plant_perm$weight)
  perm_f[i] <- summary(aov(weight ~ group, data=plant_perm))[[1]][1,4]
}
F <- summary(aov(weight ~ group, data=PlantGrowth))[[1]][1,4]
sum(perm_f >= F)/reps

```

```
## [1] 0.0163
```

Here is the ANOVA analysis:

```

model1 <- aov(weight ~ group, data = PlantGrowth)
summary(model1)

```

```

##              Df Sum Sq Mean Sq F value Pr(>F)
## group         2   3.766   1.8832    4.846 0.0159 *
## Residuals    27  10.492   0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
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

```

```

summary <- burgers_df %>% group_by(method) %>%
summarize(means=mean(sat_fat))
print(summary)

```

```

## # A tibble: 3 x 2
##   method means

```

```
##    <chr>    <dbl>
## 1 broiled  11.7
## 2 grilled  11.6
## 3 pan      12.5
```

```
summary(aov(sat_fat ~ method, data=burgers_df))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## method         2  4.653   2.3266    73.6 1.17e-11 ***
## Residuals      27  0.853   0.0316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
q <- abs(summary$means[1] - summary$means[2]) / sqrt(summary(aov(sat_fat~method, data = burgers_df))[[1]])
print(q)
```

```
## [1] 1.956468
```

```
ptukey(q, nmeans=3, df=27, lower.tail=FALSE)
```

```
## [1] 0.3634447
```

```
##Your Turn 2
```

```
library(dplyr)
summary <- PlantGrowth %>% group_by(group) %>% summarize(means = mean(weight))

print(summary)
```

```
## # A tibble: 3 x 2
##   group means
##   <fct> <dbl>
## 1 ctrl   5.03
## 2 trt1   4.66
## 3 trt2   5.53
```

```
q <- abs(summary$means[2] - summary$means[3]) / sqrt(summary(aov(weight ~ group, data = PlantGrowth))[[1]])
ptukey(q, nmeans = 3, df = 27, lower.tail = FALSE)
```

```
## [1] 0.01200642
```

All pairwise p-values:

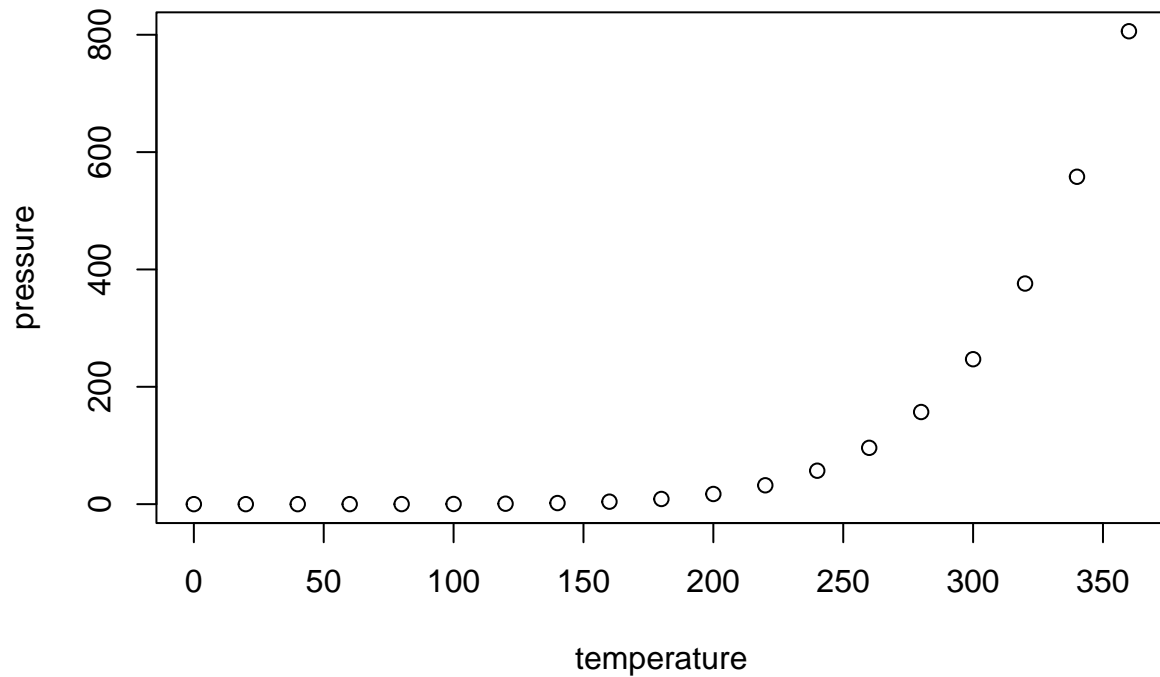
```
TukeyHSD(model1)
```

```
##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
```

```
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
##
## $group
##          diff          lwr          upr          p adj
## trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711
## trt2-ctrl  0.494 -0.1972161 1.1852161 0.1979960
## trt2-trt1  0.865  0.1737839 1.5562161 0.0120064
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.