

# Worksheet2\_group01

Antonio Alfaro de Prado, Hyunchang Oh, Tanya Toluay

2024-04-28

## 1. Posthoc power analysis

### 1.1.

Read the lung\_data.csv from Worksheet 1 and compute the (post-hoc) power of a t-test at a significance level of  $\alpha = 0.05$ . For sigma assume the empirical standard deviation from the control group. What did you use as delta and what's the result?

Answer: We chose delta as Cohen's d. calculated by dividing the difference in sample means by the standard deviation. The standard deviation here was chosen to be the control group's standard deviation as specified in the assignment (and as in many practical experimental settings where it is assumed to represent the population's standard deviation.)

The power we got was approx. 0.99, indicating that the test was very much capable of rejecting the null when it should.

```
# Read in the trial data
lung_data <- read.csv("C:/Users/tanya/Downloads/lung_data.csv")

control <- subset(lung_data, Trial.arm=="Control")
treatment <- subset(lung_data, Trial.arm=="Treatment")

n1 <- nrow(control)
n2 <- nrow(treatment)

mu1 <- mean(control$Lung.function)
mu2 <- mean(treatment$Lung.function)

sd <- sd(control$Lung.function)

Cohens_d <- abs(mu1-mu2)/sd

alpha <- 0.05

t <- "two.sample"

power <- power.t.test(n=n1, delta=Cohens_d, sd=sd, sig.level=alpha, power=NULL, type=t, strict=TRUE)$power
power

## [1] 0.9911819
```

## 1.2.

To explore post-hoc power further, sample two normally distributed samples of size  $n = 20$  with sd 1, one as “control” with mean 0 and one as “treatment” with mean 0.5. Compute the p-value of the t-test of the two 2 groups, and the post-hoc power using the observed difference in means between the two groups and the sd computed from their pooled (averaged) variance. Repeat the simulation a large number of times and plot p-values against the post-hoc power. Discuss whether post-hoc power analysis is generally a reliable measure.

Answer: Post-hoc power analysis is indeed a reliable measure, as an increase in post-hoc power leads to decreasing the p-value (they are strongly inversely related to each other.)

```
simulations <- 1000

p_values <- numeric(simulations)
powers <- numeric(simulations)

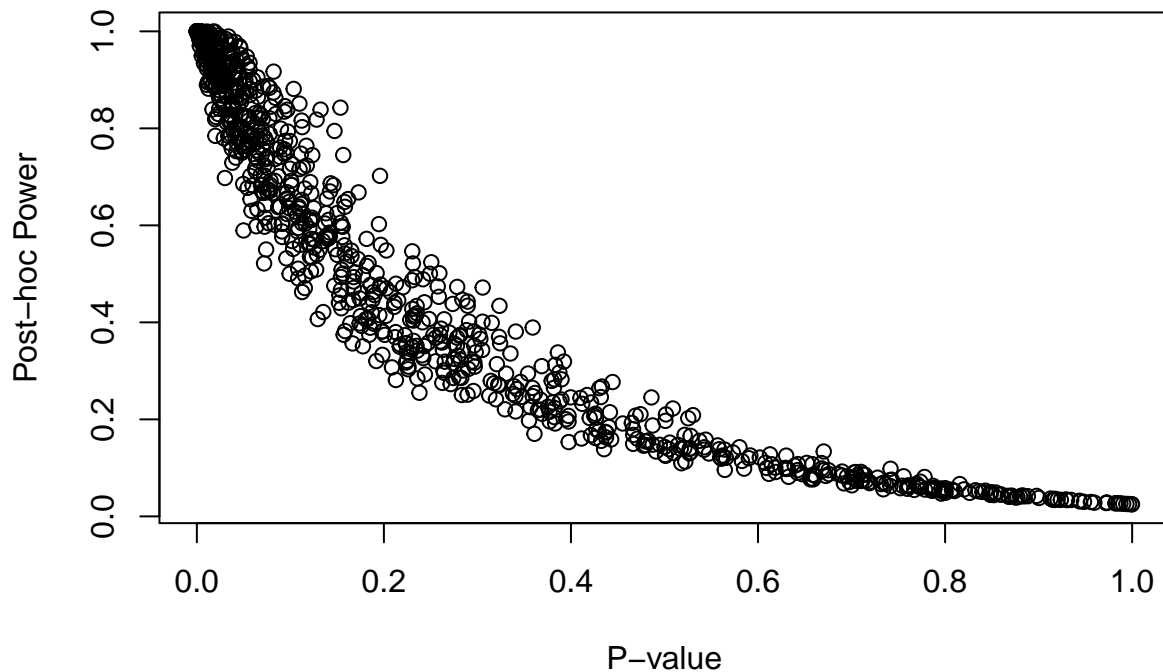
get_power <- function(x, y) {
  mean_diff <- mean(x) - mean(y)
  averaged_sd <- (sd(x)+sd(y))/2
  d <- mean_diff / averaged_sd
  power <- power.t.test(n = length(x) + length(y), delta = d, sd = averaged_sd, sig.level = 0.05, type = "two.sided")
  return(power)
}

for (i in 1:simulations) {
  control <- rnorm(20, mean = 0, sd = 1)
  treatment <- rnorm(20, mean = 0.5, sd = 1)
  p_values[i] <- t.test(control, treatment)$p.value

  powers[i] <- get_power(control, treatment)
}

plot(p_values, powers, xlab = "P-value", ylab = "Post-hoc Power", main = "P-values vs. Post-hoc Power")
```

## P-values vs. Post-hoc Power



## 2. Multiple testing correction

Familiarize yourself with the data set. Then

- correct the p-values with Bonferroni/Holm controlling the FWER,
- correct the p-values with Benjamini-Hochberg controlling the FDR

Compare the number of significant genes per correction with appropriate visualisations or tables. Do your numbers match column q in the data frame `lung_cancer_df`? Do you think they controlled the FWER or FDR?

Answer: The number of significant genes selected through 'q' roughly match the number of those selected through Bonferroni or Holm correction method. It appears that the authors were trying to control FWER through their correction through q. For the plot and the number of significant genes per correction method, see below.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2     3.5.1      v tibble     3.2.1
## v lubridate   1.9.3      v tidyr      1.3.1
## v purrr       1.0.2
```

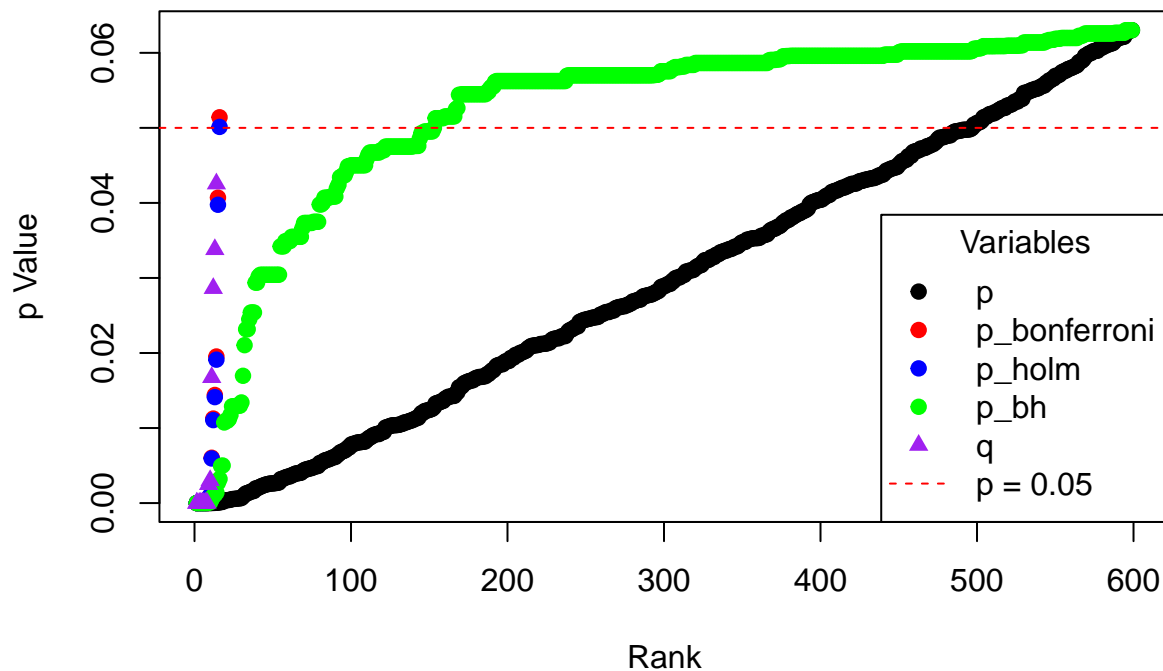
```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

url <- paste0("https://gdac.broadinstitute.org",
"/runs/analyses_latest/reports/cancer/",
"LUSC-TP/MutSigNozzleReport2CV/sig_genes.txt")
lung_cancer_df <- read_tsv(url,show_col_types=FALSE)

lung_cancer_df$rank <- rank(lung_cancer_df$p)
lung_cancer_df <- lung_cancer_df[lung_cancer_df$rank<600,]
lung_cancer_df$p_bonferroni <- p.adjust(lung_cancer_df$p,method="bonferroni")
lung_cancer_df$p_holm <- p.adjust(lung_cancer_df$p,method="holm")
lung_cancer_df$p_bh <- p.adjust(lung_cancer_df$p,method="BH")

plot(lung_cancer_df$rank, lung_cancer_df$p, xlab = "Rank", ylab = "p Value", main = "p-value and its corrections")
points(lung_cancer_df$rank, lung_cancer_df$p_bonferroni, col = "red",pch=19)
points(lung_cancer_df$rank, lung_cancer_df$p_holm, col = "blue",pch=19)
points(lung_cancer_df$rank, lung_cancer_df$p_bh, col = "green",pch=19)
points(lung_cancer_df$rank, lung_cancer_df$q, col = "purple",pch=17)
abline(h = 0.05, col = "red", lty = 2)
legend("bottomright", legend = c("p", "p_bonferroni","p_holm","p_bh", "q","p = 0.05"), pch = c(19, 19,19,19,17,2))
```

## p-value and its corrections



```

p_sig <- sum(lung_cancer_df$p<0.05)
p_bonferroni_sig <- sum(lung_cancer_df$p_bonferroni<0.05)
p_holm_sig <- sum(lung_cancer_df$p_holm<0.05)
p_bh_sig <- sum(lung_cancer_df$p_bh<0.05)
q_sig <- sum(lung_cancer_df$q<0.05)

data <- data.frame(Criterion = c("p", "Bonferroni", "Holm", "Benjammini-Hochberg", "q"),
                   Significant_Samples = c(p_sig, p_bonferroni_sig, p_holm_sig, p_bh_sig, q_sig))
print(data)

```

```

##           Criterion Significant_Samples
## 1              p              496
## 2      Bonferroni              15
## 3              Holm              15
## 4 Benjammini-Hochberg          153
## 5              q              14

```

## 2.2.

Repeat your analysis for lung adenocarcinoma and compare the number of significantly mutated genes between both lung cancer types. To obtain the lung adenocarcinoma data replace LUSC with LUAD in the above url.

Answer: In all correction methods, there appeared to be more significant genes in lung adenocarcinoma, which indicate that there are more mutated genes correlated to lung adenocarcinoma than in lung squamous cell carcinoma.

```

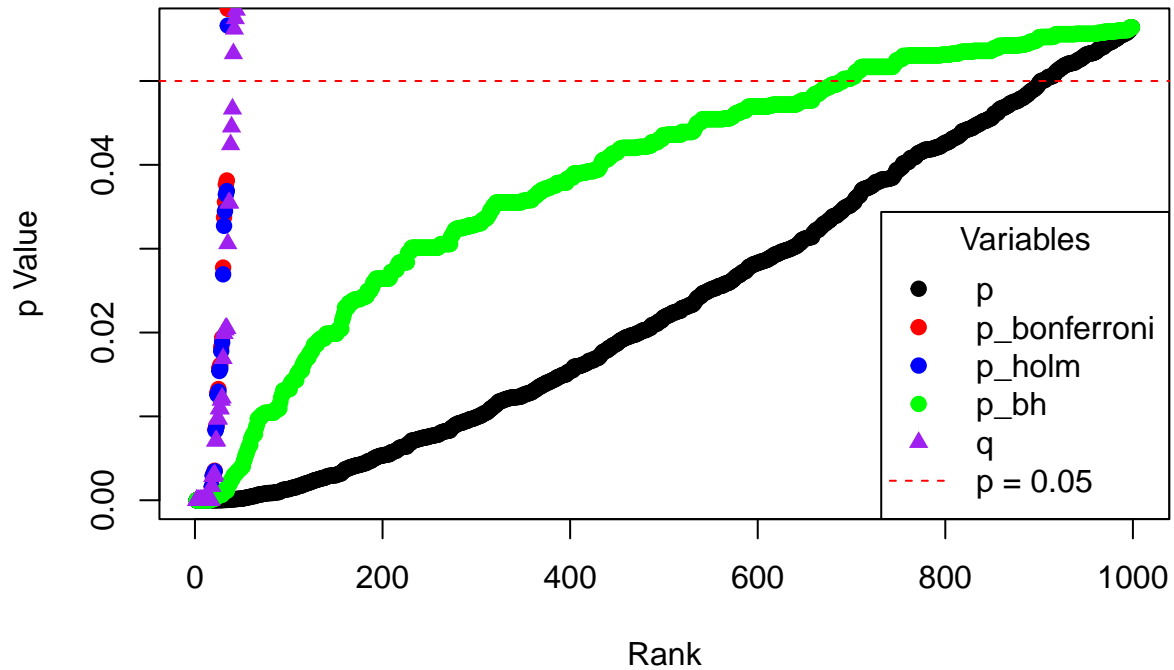
url <- paste0("https://gdac.broadinstitute.org",
              "/runs/analyses_latest/reports/cancer/",
              "LUAD-TP/MutSigNozzleReport2CV/sig_genes.txt")
luad_df <- read_tsv(url, show_col_types=FALSE)

luad_df$rank <- rank(luad_df$p)
luad_df <- luad_df[luad_df$rank < 1000, ]
luad_df$p_bonferroni <- p.adjust(luad_df$p, method = "bonferroni")
luad_df$p_holm <- p.adjust(luad_df$p, method = "holm")
luad_df$p_bh <- p.adjust(luad_df$p, method = "BH")

plot(luad_df$rank, luad_df$p, xlab = "Rank", ylab = "p Value", main = "p-value and its corrections", pch = 1)
points(luad_df$rank, luad_df$p_bonferroni, col = "red", pch = 19)
points(luad_df$rank, luad_df$p_holm, col = "blue", pch = 19)
points(luad_df$rank, luad_df$p_bh, col = "green", pch = 19)
points(luad_df$rank, luad_df$q, col = "purple", pch = 17)
abline(h = 0.05, col = "red", lty = 2)
legend("bottomright", legend = c("p", "p_bonferroni", "p_holm", "p_bh", "q", "p = 0.05"), pch = c(19, 19, 19, 19, 17, 1))

```

## p-value and its corrections



```
p_sig <- sum(luad_df$p < 0.05)
p_bonferroni_sig <- sum(luad_df$p_bonferroni < 0.05)
p_holm_sig <- sum(luad_df$p_holm < 0.05)
p_bh_sig <- sum(luad_df$p_bh < 0.05)
q_sig <- sum(luad_df$q < 0.05)

data <- data.frame(Criterion = c("p", "Bonferroni", "Holm", "Benjammini-Hochberg", "q"),
                   Significant_Samples = c(p_sig, p_bonferroni_sig, p_holm_sig, p_bh_sig, q_sig))
print(data)
```

```
##           Criterion Significant_Samples
## 1              p              901
## 2      Bonferroni              34
## 3              Holm              34
## 4 Benjammini-Hochberg              690
## 5                q              40
```

### 3. Multiple testing correction calibration

Simulate 90 times 10 samples from a standard normal distribution and perform a one sample t-test each time. Then simulate 10 times 10 samples with mean  $\mu = 3$  and standard deviation  $\sigma = 1$  and again perform a one sample t-test against zero mean. Collate the 100 p-values into a single vector or data.frame.

Correct the p-values with both Bonferroni and BH. Were there any type I errors when controlling either FWER or FDR? Compute the FWER and FDR for both corrections.

Repeat the simulations a large number of times and estimate the average FWER and average FDR. Compare the averages to the expected values of the FWER and FDR.

```
# Stimulating 90 times 10 samples with standard normal distribution and t-test for each
num_simulations_1 <- 90
sample_size_1 <- 10
p_values_1 <- numeric(num_simulations_1)
set.seed(123)

for (i in 1:num_simulations_1) {
  samples_1 <- rnorm(sample_size_1)
  t_test_result_1 <- t.test(samples_1, mu = 0)
  p_values_1[i] <- t_test_result_1$p.value
}

#print(p_values_1)

#simulate 10 times 10 samples with mean mu = 3 and standard deviation sigma = 1 and perform a one sample t-test
num_simulations_2 <- 10
sample_size_2 <- 10
true_mean <- 3
true_sd <- 1
p_values_2 <- numeric(num_simulations_2)
set.seed(123)

for (i in 1:num_simulations_2) {
  samples_2 <- rnorm(sample_size_2, mean = true_mean, sd = true_sd)
  t_test_result_2 <- t.test(samples_2, mu = 0)
  p_values_2[i] <- t_test_result_2$p.value
}

#print(p_values_2)

# Collate the 100 p-values into a single vector or data.frame
# A single vector combination
all_p_values <- c(p_values_1, p_values_2)
# Dataframe combination
p_values_df <- data.frame(p_values = all_p_values)

# Bonferroni correction
bonferroni_corrected_p_values <- p_values_df$p_values * 100
# Benjamini-Hochberg (BH) correction
bh_corrected_p_values <- p.adjust(p_values_df$p_values, method = "BH")
# Type I errors for Bonferroni correction (FWER)
bonferroni_type_I_errors <- sum(bonferroni_corrected_p_values < 0.05)
# Type I errors for BH correction (FDR)
bh_type_I_errors <- sum(bh_corrected_p_values < 0.05)
# Family-Wise Error Rate (FWER) for Bonferroni correction
bonferroni_fwer <- bonferroni_type_I_errors / 100
# False Discovery Rate (FDR) for BH correction
bh_fdr <- bh_type_I_errors / sum(bh_corrected_p_values < 0.05)
```

```
print("Bonferroni Type I Errors (FWER):")
```

```
## [1] "Bonferroni Type I Errors (FWER):"
```

```
print(bonferroni_type_I_errors)
```

```
## [1] 10
```

```
print("Bonferroni FWER:")
```

```
## [1] "Bonferroni FWER:"
```

```
print(bonferroni_fwer)
```

```
## [1] 0.1
```

```
print("BH Type I Errors (FDR):")
```

```
## [1] "BH Type I Errors (FDR):"
```

```
print(bh_type_I_errors)
```

```
## [1] 10
```

```
print("BH FDR:")
```

```
## [1] "BH FDR:"
```

```
print(bh_fdr)
```

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

Now below: we conduct 1000 simulations, and for each simulation, we generate samples, perform a one-sample t-test, and correct the resulting p-value using both Bonferroni and BH methods. We then determine if the null hypothesis is rejected based on each correction method and update the FWER and FDR vectors accordingly. Finally, we calculate the average FWER and FDR over all simulations and compare them to their expected values.

```
num_simulations <- 1000
sample_size <- 10

true_mean <- 3
true_sd <- 1

fwer_values <- numeric(num_simulations)
fdr_values <- numeric(num_simulations)
```



```

set.seed(123)

for (i in 1:num_simulations) {
  samples <- rnorm(sample_size, mean = true_mean, sd = true_sd)
  t_test_result <- t.test(samples, mu = 0)
  p_value <- t_test_result$p.value

  # Bonferroni correction
  bonferroni_corrected_p_value <- p_value * num_simulations

  # Benjamini-Hochberg (BH) correction
  bh_corrected_p_value <- p.adjust(p_value, method = "BH")

  # Determine if null hypothesis is rejected based on Bonferroni correction
  bonferroni_rejected <- bonferroni_corrected_p_value < 0.05

  # Determine if null hypothesis is rejected based on BH correction
  bh_rejected <- bh_corrected_p_value < 0.05

  # Update FWER and FDR values
  fwer_values[i] <- ifelse(bonferroni_rejected, 1, 0)
  fdr_values[i] <- ifelse(bh_rejected, 1, 0)
}

average_fwer <- mean(fwer_values)
average_fdr <- mean(fdr_values)
print("Average FWER:")

## [1] "Average FWER:"

print(average_fwer)

## [1] 0.89

print("Average FDR:")

## [1] "Average FDR:"

print(average_fdr)

## [1] 1

```

The expected error rates for both the Bonferroni and Benjamini-Hochberg (BH) corrections are 0.05. However, in our simulation, we found that the average error rates were much higher.

For the Bonferroni correction, the observed average error rate (FWER) was 0.89. This suggests that the Bonferroni correction method was too strict, leading to many false negatives. In simpler terms, it might have missed detecting real differences due to its overly conservative approach.

On the other hand, for the BH correction, the observed average error rate (FDR) was 1. This means that all rejected hypotheses were likely false discoveries. In other words, the BH correction method might have been too lenient, leading to a high rate of false positives.