Practicum #2 - SOLUTIONS

Due: Tuesday, November 9 @ 6pm

Adapted from Dr. Rosemary Braun

1. Read the help pages for `dim`, `nrow`, and `ncol`. Using one or more of these functions, find out how many rows and columns the `phenotype_df` dataframe has.

```
dim(phenotype_df)
```

```
## [1] 128  24
```

```
nrow(phenotype_df)
```

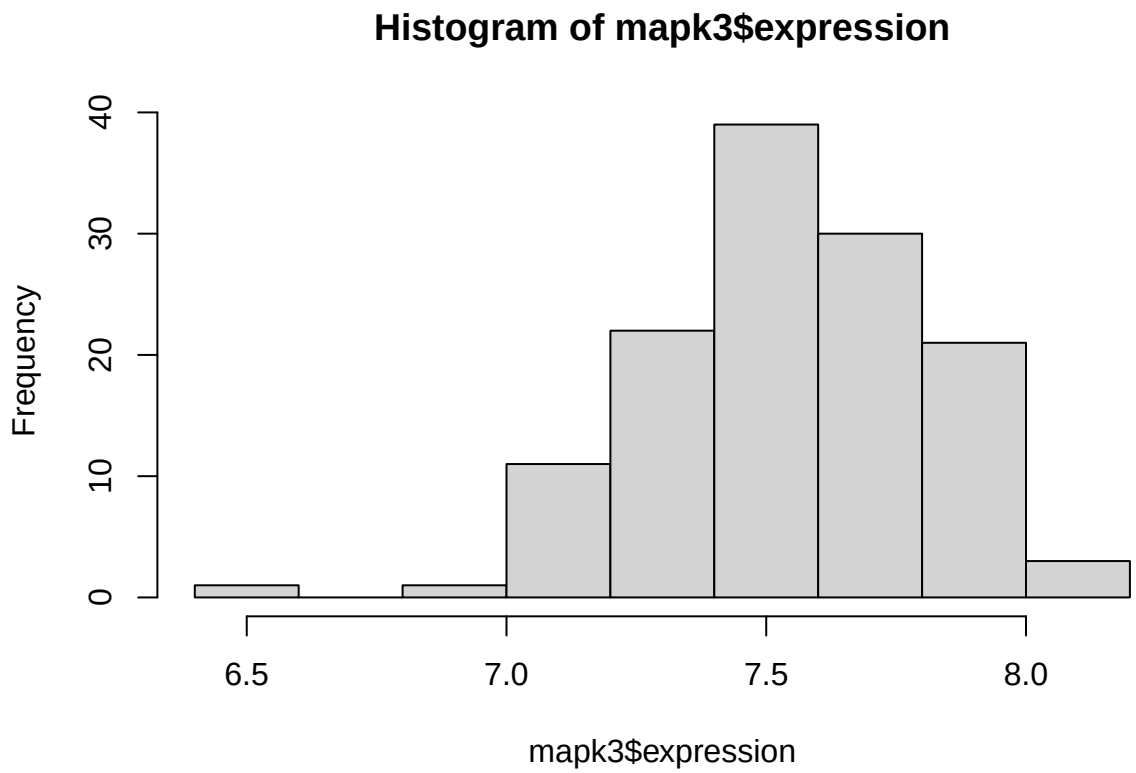
```
## [1] 128
```

```
ncol(phenotype_df)
```

```
## [1] 24
```

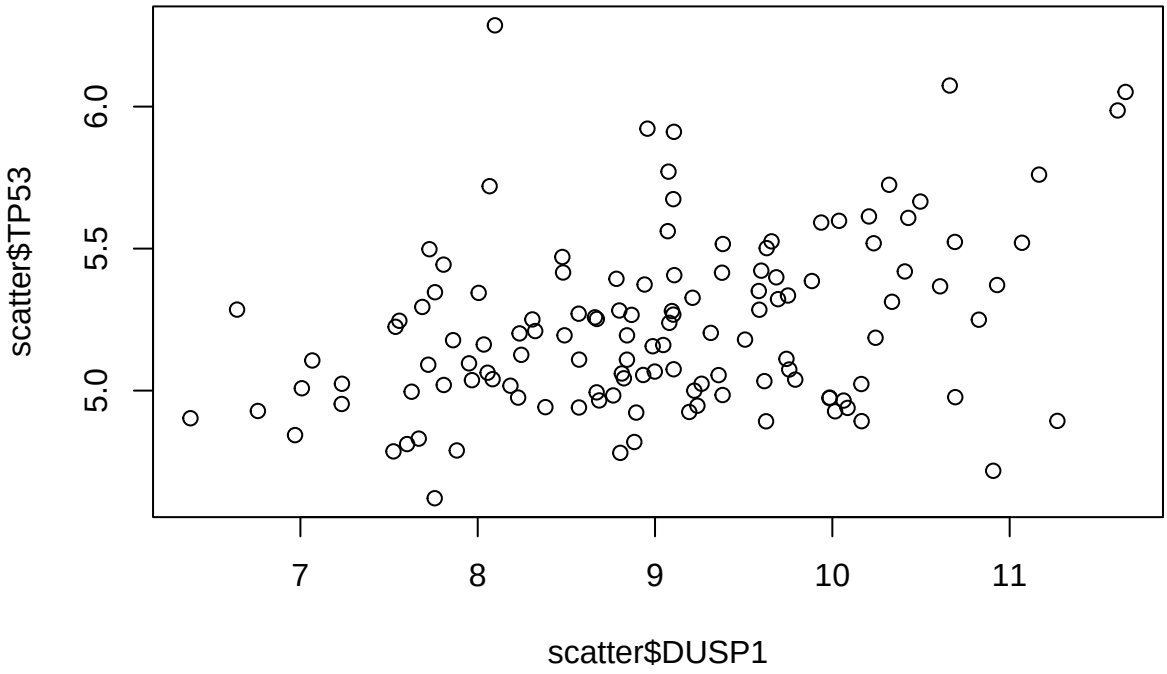
2. Let's practice filtering and plotting the data:
    a. Plot a histogram of the expression of MAPK3 in all samples.

```
# several ways to make the same plot
mapk3 <- long_expr %>%
    dplyr::filter(gene == "MAPK3")
hist(mapk3$expression)
```



b. Using `plot()`, make a scatterplot of the expression of DUSP1 vs TP53.

```
# several ways to make the same plot - either gene can be on the x axis
scatter <- long_expr %>%
    dplyr::filter(gene %in% c("DUSP1", "TP53")) %>%
    tidyr::pivot_wider(names_from = gene, values_from = expression)
plot(scatter$DUSP1, scatter$TP53)
```



3. Look at the help page for t.test:
    a. Write an expression to extract the degrees of freedom from `null.t.out`

```
null.t.out$parameter
```

```
##       df
## 20.84065
```

b. Write an expression to extract the upper bound of the confidence interval from ‘null.t.out‘

```
null.t.out$conf.int[2]
```

```
## [1] 0.5663935
```

4. Using `kegg_pathways`:
    a. How many pathways are in kegg_pathways? (*Hint: use* ***`length()`***)

```
length(kegg_pathways)
```

```
## [1] 229
```

b. How many genes are in the pathway called "p53 signaling pathway"?

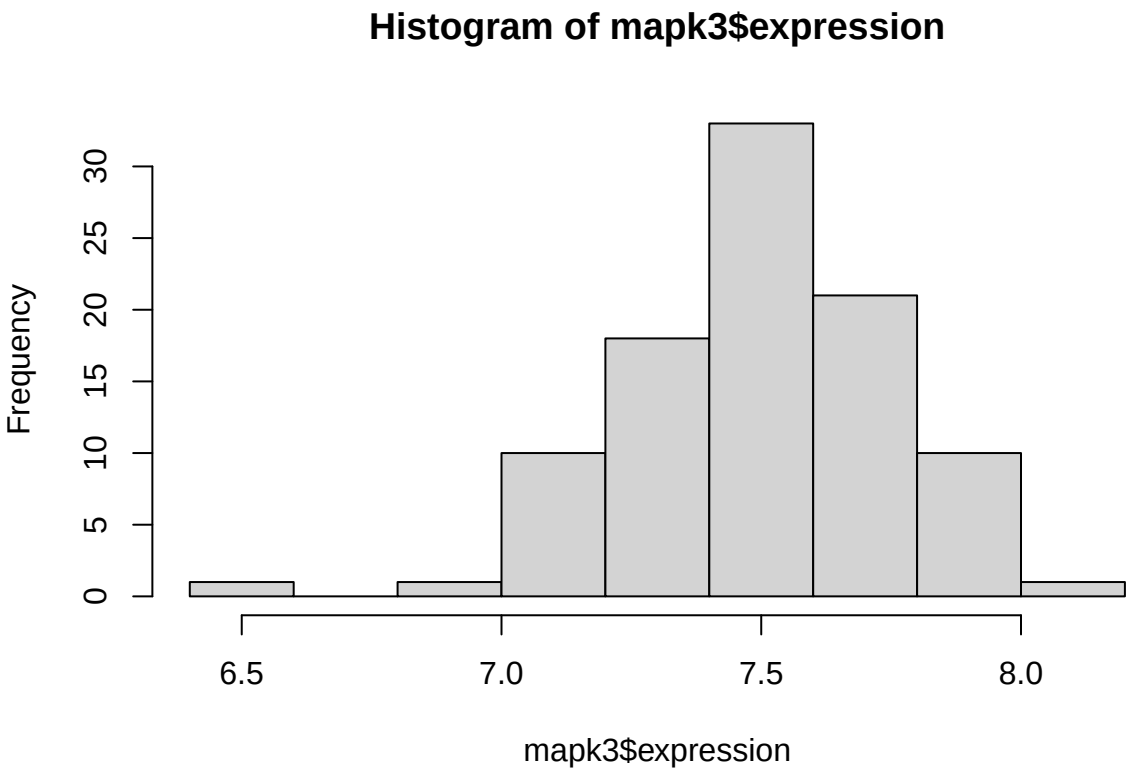
```
length(kegg_pathways$`p53 signaling pathway`)
```

```
## [1] 52
```

5. Using what you have just learned, plot:

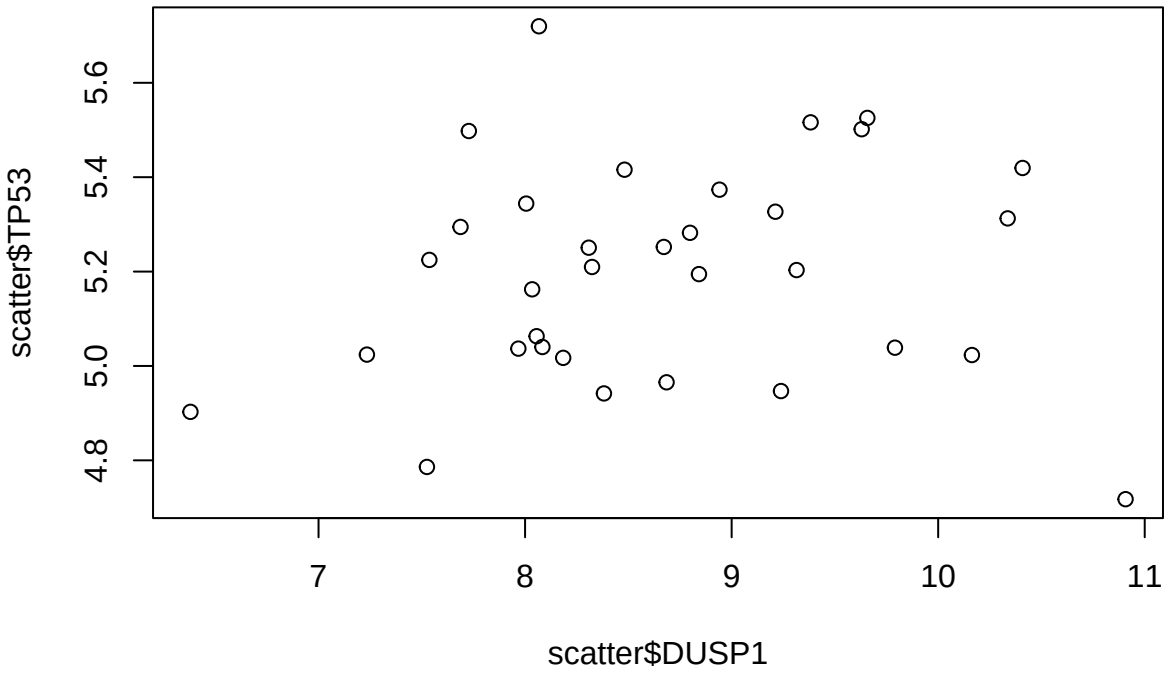
a. A histogram of MAPK3 only in B-cell ALL patients, where `phenotype_df$BT=="B"`.

```
# again many ways to solve this problem, I prefer using the merged data frame
mapk3 <- merged %>%
    dplyr::filter(gene == "MAPK3", BT == "B")
hist(mapk3$expression)
```



b. A scatter plot of DUSP1 vs TP53 only in T-cell ALL patients, which is indicated by "T" in the BT columns of phenotype_df

```
# again many ways to solve this problem, I prefer using the merged data frame
scatter <- merged %>%
    dplyr::filter(gene %in% c("DUSP1", "TP53"), BT == "T") %>%
    tidyr::pivot_wider(names_from = gene, values_from = expression)
plot(scatter$DUSP1, scatter$TP53)
```



6. Modify the above to test whether MAPK3 expression differs in people younger than 20.

```
# create a dataframe of expression for B-cell ALL
bcell <- merged %>%
    dplyr::filter(gene == "MAPK3",
                  BT == "B",
                  age < 20)

# create a dataframe of expression for T-cell ALL
tcell <- merged %>%
    dplyr::filter(gene == "MAPK3",
                  BT == "T",
                  age < 20)

# perform a t.test for expression
t.test(bcell$expression, tcell$expression)
```

```
##
##  Welch Two Sample t-test
##
## data:  bcell$expression and tcell$expression
## t = -1.2975, df = 11.331, p-value = 0.2203
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4413671  0.1132538
## sample estimates:
## mean of x mean of y
##  7.481277  7.645334
```

7. Modify the above to spit out TRUE when the expression is higher in B-cell ALL and FALSE when it's higher in T-cell ALL.

```r
# one way
estimate <- t.test(bcell$expression, tcell$expression)$estimate
estimate[[1]] > estimate[[2]]
```

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

```r
# another option - this is less elegant because it requires doing the same t test twice!
t.test(bcell$expression, tcell$expression)$estimate[[1]] > t.test(bcell$expression, tcell$expression)$esti
```

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

8. In the last block of code given above,

    a. What is the null hypothesis?

Null hypothesis: MAPK3 expr are the same in B and T cell

    b. What is the alternative hypothesis?

Alt hypothesis: MAPK3 expr are not the same in B and T cell

    c. If we were to do this test for all genes, how many tests would we do?

```r
nrow(expression_df)
```

```
## [1] 8594
```

```r
length(unique(merged$gene))
```

```
## [1] 8594
```

There are 8594 genes in the dataset, so we would do 8594 tests.

    d. Of those, how many would we expect to have $p < 0.05$ by pure chance?

8594*0.05 = 429.7!

9. Explain what each line in the `for` loop does.

```r
# First, we'll initialize a vector to hold our results.
gene.p.vals <- vector()

# Iterate over each unique gene in long_expr, assign to variable gene_name
for(gene_name in unique(long_expr$gene)) {
    # Assign the expression data for the given gene from the B cell group to bcell
    bcell <- merged %>%
        dplyr::filter(gene == gene_name,
                      BT == "B")

    # Assign the expression data for the given gene from the T cell group to tcell
    tcell <- merged %>%
```

```
        dplyr::filter(gene == gene_name,
                      BT == "T")

    # Calculate t-test p-value comparing gene expression between T cells and B cells
    # in given gene.  Add p-value for given gene to gene.p.vals vector
    gene.p.vals[gene_name] <- t.test(bcell$expression, tcell$expression)$p.value
}
```

10. Using `gene.p.vals`,

    a. How many p values are less than 0.05? How many are less than 0.01? How do those compare to what you expected?

```
sum(gene.p.vals < 0.05)
```

```
## [1] 3246
```

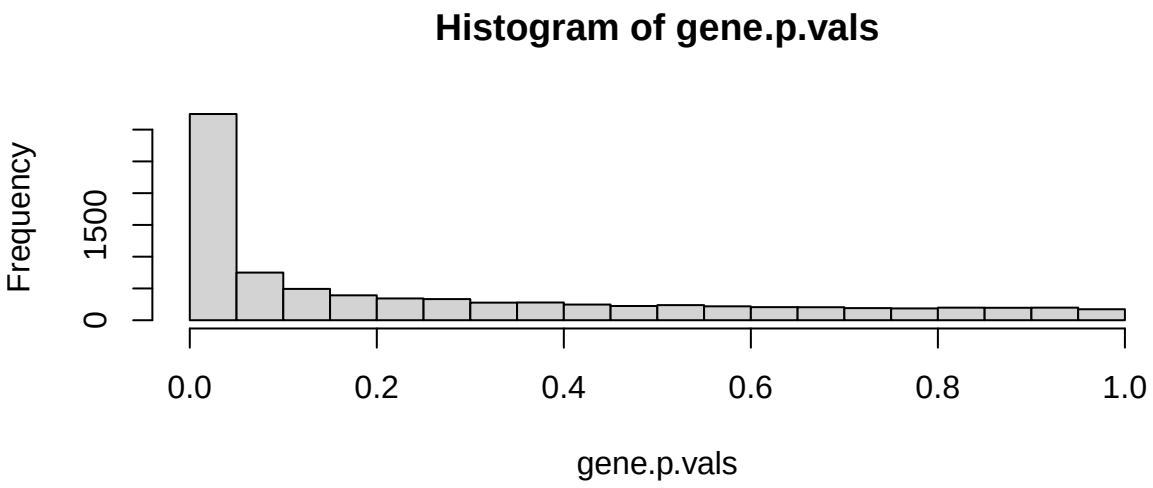
```
sum(gene.p.vals < 0.01)
```

```
## [1] 2216
```

There appear to be many more genes with significant differences in expression than you would expect by sampling chance alone.

    b. Plot a histogram of the p-values. Does it look the way you'd expect? Why or why not?

```
hist(gene.p.vals)
```



Any justified answer here. Note: We would expect a uniform distribution of p-values if there were no differences in gene expression at the population level.

    c. Use the `min()` function to find the smallest p-value.

```
min(gene.p.vals)
```

```
## [1] 3.552271e-44
```

    d. Use the `sort()` function to find the ten smallest p-values.

```
sort(gene.p.vals)[1:10]
```

```
##        CD79B          CD9         BLNK      HLA-DMB         CD3D       CDKN1A
## 3.552271e-44 1.424983e-43 5.499754e-42 7.098067e-41 1.484983e-38 1.745092e-38
##        FOXO1      THEMIS2      HLA-DMA     HLA-DQB1
## 1.001887e-34 4.363434e-33 9.121866e-33 1.305611e-31
```

11. Compute the FWER (Dunn/Bonferroni) adjusted p-values, and store it as `gene.bonf.vals`. Then repeat parts (a),(c),(d) of the previous exercise using the adjusted p-vals.

```
# bonferonni correction: dividing alpha by n OR multiplying pval by n
gene.bonf.vals <- (gene.p.vals * (length(gene.p.vals)))

# number of sig genes - WAY LESS!
sum(gene.bonf.vals < 0.05)
```

```
## [1] 668
```

```
sum(gene.bonf.vals < 0.01)
```

```
## [1] 561
```

```
# min values
min(gene.bonf.vals)
```

```
## [1] 3.052821e-40
```

```
sort(gene.bonf.vals)[1:10]
```

```
##        CD79B          CD9         BLNK      HLA-DMB         CD3D       CDKN1A
## 3.052821e-40 1.224631e-39 4.726489e-38 6.100079e-37 1.276194e-34 1.499732e-34
##        FOXO1      THEMIS2      HLA-DMA     HLA-DQB1
## 8.610217e-31 3.749936e-29 7.839332e-29 1.122042e-27
```

12. Compute the FDR adjusted p-values, and store it as `gene.FDR.vals`. (Hint: use the `p.adjust` function.) Then repeat parts (a),(c),(d) of the previous exercise using the FDR-vals.

```
gene.FDR.vals <- p.adjust(gene.p.vals, "fdr")

# number of sig genes - compare to BF!
sum(gene.FDR.vals < 0.05)
```

```
## [1] 2413
```

```
sum(gene.FDR.vals < 0.01)
```

```
## [1] 1550
```

```
# min values
min(gene.FDR.vals)
```

```
## [1] 3.052821e-40
```

```
sort(gene.FDR.vals)[1:10]
```

```
##        CD79B          CD9         BLNK      HLA-DMB       CDKN1A         CD3D
## 3.052821e-40 6.123153e-40 1.575496e-38 1.525020e-37 2.499554e-35 2.499554e-35
##        FOXO1      THEMIS2      HLA-DMA     HLA-DQB1
## 1.230031e-31 4.687419e-30 8.710369e-30 1.122042e-28
```

13. What do the expressions `t.test(bcell$expression, tcell$expression)$estimate` and `test_est[1]-test_est[2]` do? Hint: look at `test_est`

```
# The t.test() performs a t test comparing the MAPK3 expression from the B cell group
# to the MAPK3 expression from the T cell group,
# with the sample estimate means assigned to test_est.

# The second line determine the single fold change by taking the estimate [2] (mean of y)
# from the estimate [1] (mean of x), and assigning it to fold_change}
```

14. Refer back to how we extended the p-value computation to loop over all genes.

    a. Create some code to compute the fold changes for all the genes and store them in a vector called `gene_fcs`. You may use either `for` or `apply` to do this.

```
# First, we'll initialize a vector to hold our results.
gene_fcs <- vector()

# note how similar this code is to the code above for MAPK3
for(gene_name in unique(long_expr$gene)) {
    bcell <- merged %>%
        dplyr::filter(gene == gene_name,
                      BT == "B")
    tcell <- merged %>%
        dplyr::filter(gene == gene_name,
                      BT == "T")

    # grab the estimates from the t.test
    gene_est <- t.test(bcell$expression, tcell$expression)$estimate

    # put the fold change into our final vector
    gene_fcs[gene_name] <- gene_est[1] - gene_est[2]
}
```

    b. Now that you have `gene_fcs`, you can pick out the ones that meet the doubling/halving criteria with `which(abs(gene_fcs)>1)`. How would you modify this to get only the genes that are up-regulated in T-cell ALL?

```
length(which(gene_fcs < -1))
```

```
## [1] 66
```

The sign depends on how fold change was calculated (geneB - geneT) vs. (geneT - geneB). In the above function, fold change was calculated using (geneB - geneT).

c. How many genes have an absolute fold change greater than 1.5?

```
length(which(abs(gene_fcs) > 1.5))
```

```
## [1] 69
```

15. How many genes are differentially expressed? (Use R, don't just count them off the page!)

```
DEgenes <- ((abs(gene_fcs) > 1.5) & (gene.FDR.vals < 0.01))
sum(DEgenes)
```

```
## [1] 69
```

16. Read the help page for `intersect`. What does the second to last line do?

The second to last line returns the number of DE genes that are in the Complement and coagulation cascades pathway, and assigns that number to my.x

17. What does the line inside the for loop do? Explain all parts of it.

The line finds the p.value from the overrepresentation test (probability we would observe at least the number of differential genes corresponding to the given pathway by chance) and assigns it to the first5pathway.pvals vector under the given pathway name

18. As a final exercise, we're going to test all the pathways!

a. Modify the code above to get a p-value for all the pathways, and store them in a vector called `pathway_pvals`.

```
pathway_pvals <- vector()

for(pathway_name in names(kegg_pathways)) {
    pathway_pvals[pathway_name] <- pathway.overrep.test(pathway_name)
}
```

b. How many pathways were significant with `p < 0.05`? Which pathways were they? Using R, spit out their p-values. Do the significant pathways surprise you (why or why not)?

```
sum(pathway_pvals < 0.05)
```

```
## [1] 23
```

```
# split out names and pvals
pathway_pvals[which(pathway_pvals < 0.05)]
```

```
##                            Viral myocarditis
##                                 3.031938e-07
##                          Rheumatoid arthritis
##                                 2.797912e-05
##                                    Phagosome
##                                 2.275858e-06
##                  Systemic lupus erythematosus
##                                 1.458265e-05
##                       Primary immunodeficiency
##                                 1.650746e-07
##    Chagas disease (American trypanosomiasis)
##                                 3.746703e-02
##                        Notch signaling pathway
##                                 2.698722e-02
##            T cell receptor signaling pathway
##                                 1.202881e-03
##            B cell receptor signaling pathway
##                                 2.105937e-03
##                                Toxoplasmosis
##                                 2.608347e-05
##                              Prostate cancer
##                                 2.538366e-02
##               Cell adhesion molecules (CAMs)
##                                 1.592373e-05
##                    Hematopoietic cell lineage
##                                 3.721042e-06
##    Natural killer cell mediated cytotoxicity
##                                 8.372017e-03
##                      Type I diabetes mellitus
##                                 1.093574e-08
##                    Autoimmune thyroid disease
##                                 2.381231e-08
##                           Allograft rejection
##                                 2.770337e-09
##                     Graft-versus-host disease
##                                 1.231637e-09
##          Antigen processing and presentation
##                                 7.574798e-09
## Intestinal immune network for IgA production
##                                 4.103537e-07
##              Staphylococcus aureus infection
##                                 7.739161e-07
##                                Leishmaniasis
##                                 7.645140e-06
##                                       Asthma
##                                 3.371637e-08
```

c. How many pathways did we test? Apply an appropriate multiple hypothesis correction.

```
length(pathway_pvals) # can also test length(kegg_pathways)
```

```
## [1] 229
```

```
# apply BF - can use either
pathway_bf <- pathway_pvals * length(pathway_pvals)
```

d. Repeat part (b) now that you've corrected. What would you conclude?

```
sum(pathway_bf < 0.05)
```

```
## [1] 17
```

```
# split out names and pvals
pathway_bf[which(pathway_bf < 0.05)]
```

```
##                              Viral myocarditis
##                                   6.943138e-05
##                            Rheumatoid arthritis
##                                   6.407218e-03
##                                      Phagosome
##                                   5.211716e-04
##                   Systemic lupus erythematosus
##                                   3.339428e-03
##                       Primary immunodeficiency
##                                   3.780209e-05
##                                   Toxoplasmosis
##                                   5.973116e-03
##                 Cell adhesion molecules (CAMs)
##                                   3.646533e-03
##                      Hematopoietic cell lineage
##                                   8.521185e-04
##                        Type I diabetes mellitus
##                                   2.504285e-06
##                      Autoimmune thyroid disease
##                                   5.453019e-06
##                             Allograft rejection
##                                   6.344071e-07
##                       Graft-versus-host disease
##                                   2.820448e-07
##            Antigen processing and presentation
##                                   1.734629e-06
## Intestinal immune network for IgA production
##                                   9.397101e-05
##                Staphylococcus aureus infection
##                                   1.772268e-04
##                                   Leishmaniasis
##                                   1.750737e-03
##                                         Asthma
##                                   7.721049e-06
```

Any justified response is fine. To me, looks like a lot of immune response pathways... which makes sense!