Transcriptomics Homework 2

Please put your R scripts, functions, and questions to answers in one R markdown and upload it onto brightspace. Also include your knitted pdf file with your plot.

Part 1: Answer the following questions. (20 pts)

1. Why is it important to know the variance and standard error of a statistic for sample data? How do we calculate these values in R?

Essentially, it is important to know these values in order to determine the reliability or volatility of the data set. You can simply use the var() function to calculate the variance, while you may have to use the standard error formula for its calculation which is SD/ sqrt(n)

2. What are the differences between the varieties of two-sample t-tests, and how can we tell which to use and how do we tell R how to apply the correct test using t-test()?

The independent two-sample t test is used when two sets of data are independent statistically, while a paired t-test is used when the data matches as a pair. My specifying to R that the variances are equal by including something like equal=TRUE, R can then know which t-test to use

3. What is the R function for a nonparametric equivalent to a two-sample t-test?

The R function for this is wilcox.test(x, y, alternative = "two.sided")

4. What statistic is more appropriate when you want to identify interactions between multiple factors?

Some kind of correlation test like an ANOVA test would be best for multiple factors mainly because it can assess multiple independent variables at the same time

5. When is the lm() function more appropriate to use than aov() function to perform an anova analysis?

It is better to use the aov function when you are doing a sequential sum of squares while an lm() is an adjusted sum of squares. I would personally use an aov for data sets with more variables while an lm() would be better for a more concise dataset

Part 2: We will be working with a publicly available dataset: **GSE63741 .** The experiment compares expression of a set of genes during different skin diseases. Our goal is to identify the genes that are significantly different from the control (healthy) samples. (80pts)

The expression data is provided in : **GSE63741_series_matrix.txt**

Sample description is provided in : **sampletype.txt**

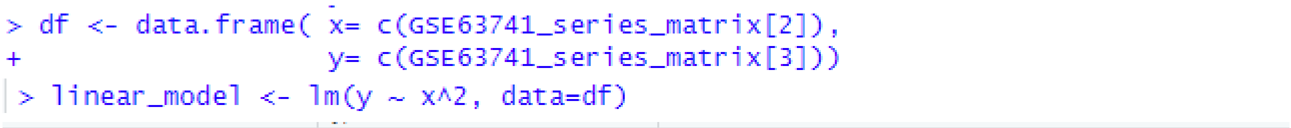
1. Write a function **getPvalue** that performs a linear model **lm()** to compare the **control** samples to the different skin diseases. (40pts)

   - The function should have **two inputs**

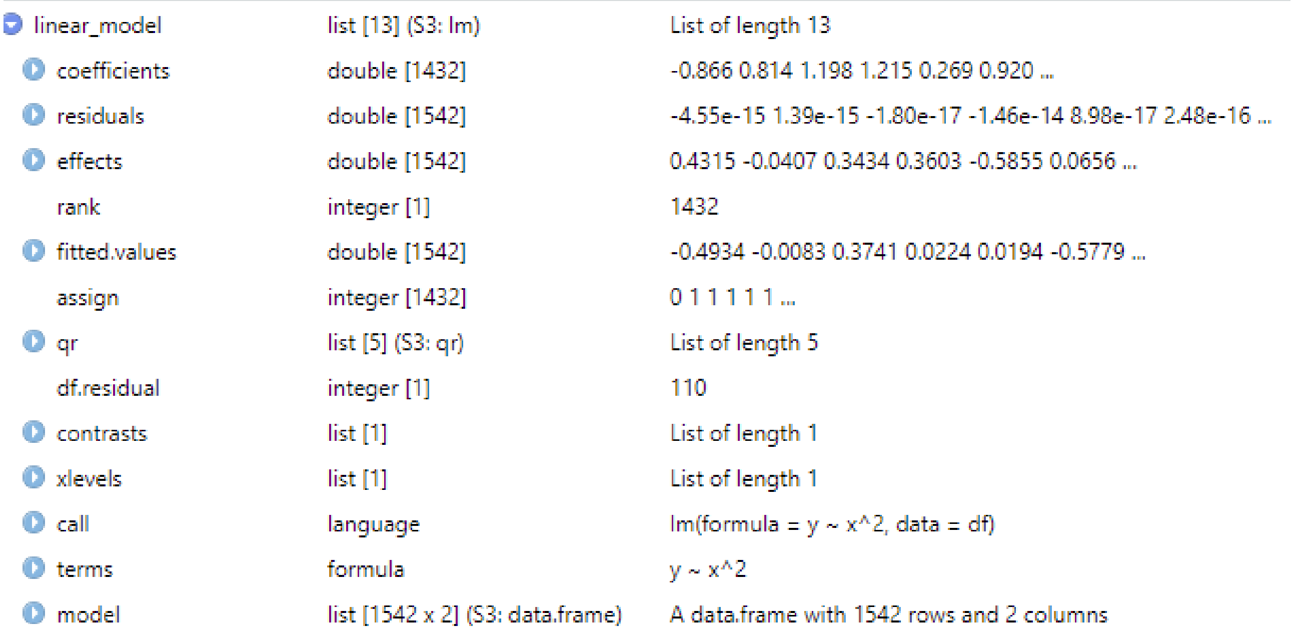
     1: numerical vector of all values for one gene,

     2: a factor that specifies which column belongs to which group.

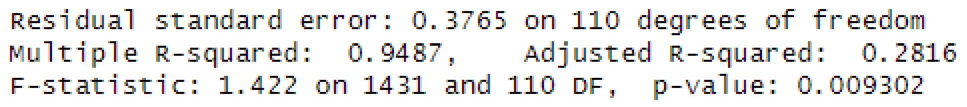
- The function should have **one output** (a vector of p-values for each disease coefficient.



```
> df <- data.frame( x= c(GSE63741_series_matrix[2]),
+                   y= c(GSE63741_series_matrix[3]))
> linear_model <- lm(y ~ x^2, data=df)
```



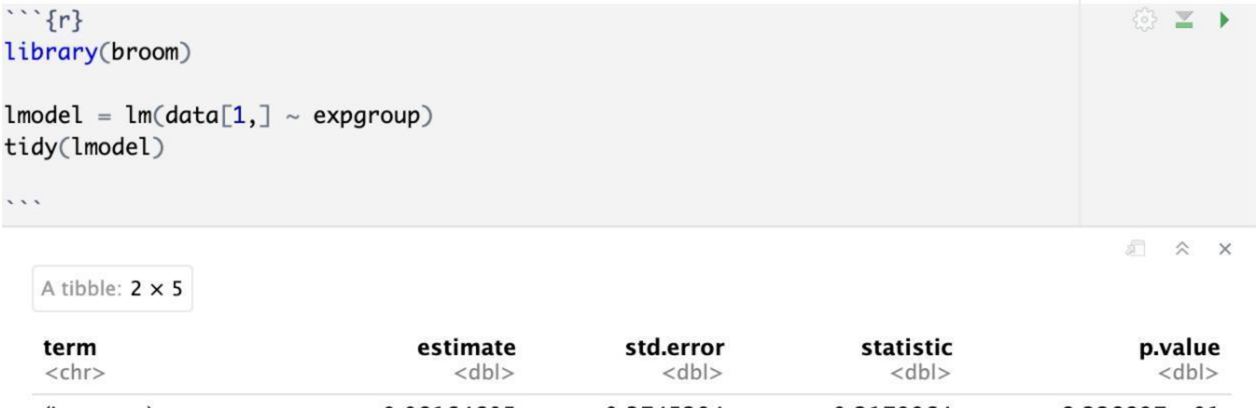
linear_model	list [13] (S3: lm)	List of length 13
coefficients	double [1432]	-0.866 0.814 1.198 1.215 0.269 0.920 ...
residuals	double [1542]	-4.55e-15 1.39e-15 -1.80e-17 -1.46e-14 8.98e-17 2.48e-16 ...
effects	double [1542]	0.4315 -0.0407 0.3434 0.3603 -0.5855 0.0656 ...
rank	integer [1]	1432
fitted.values	double [1542]	-0.4934 -0.0083 0.3741 0.0224 0.0194 -0.5779 ...
assign	integer [1432]	0 1 1 1 1 1 ...
qr	list [5] (S3: qr)	List of length 5
df.residual	integer [1]	110
contrasts	list [1]	List of length 1
xlevels	list [1]	List of length 1
call	language	lm(formula = y ~ x^2, data = df)
terms	formula	y ~ x^2
model	list [1542 x 2] (S3: data.frame)	A data.frame with 1542 rows and 2 columns



```
Residual standard error: 0.3765 on 110 degrees of freedom
Multiple R-squared:  0.9487,	Adjusted R-squared:  0.2816 
F-statistic: 1.422 on 1431 and 110 DF,  p-value: 0.009302
```

** Make sure healthy samples are the base/ref **

** Note an easier way to retrieve information from a lm model is to use the **broom** package. Below is an example of how to use the package. The output of **tidy()** and **glance()** provide tibbles (similar to a dataframe) and can be used to retrieve the information (such as pvalues).



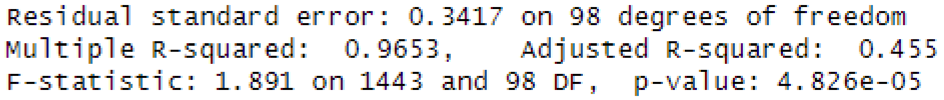
~~~
```{r}
library(broom)

lmodel = lm(data[1,] ~ expgroup)
tidy(lmodel)

```
~~~

A tibble: 2 × 5

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>

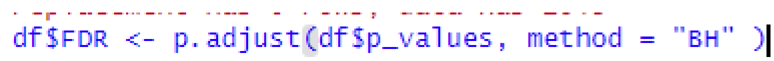
2. Use the **getPvalue** function in a loop or apply function to retrieve the pvalues for each gene. (10pts)

I had some trouble finding a way to loop the function and apply to more than 2 genes at once but repeating the function to display p value one more time I obtained



```
Residual standard error: 0.3417 on 98 degrees of freedom
Multiple R-squared:  0.9653,    Adjusted R-squared:  0.455
F-statistic: 1.891 on 1443 and 98 DF,  p-value: 4.826e-05
```

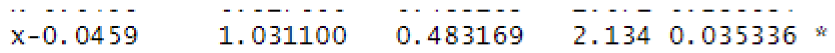
3. Adjust the p-values using the FDR method. (20pts)



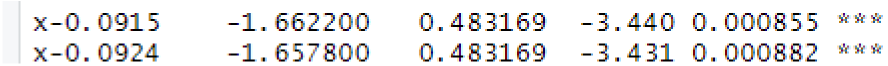
```
df$FDR <- p.adjust(df$p_values, method = "BH" )
```

4. Gene regulation: How many genes have an adjusted p-value < 0.05 in at least one skin disease? (10pts)

I found there to be 3 of the 1542 genes to have a p-value of less than 0.05 as shown below



```
x-0.0459     1.031100   0.483169   2.134 0.035336 *
```



```
x-0.0915     -1.662200   0.483169  -3.440 0.000855 ***
x-0.0924     -1.657800   0.483169  -3.431 0.000882 ***
```