Gene Expression Data

Theo Dimitrasopoulos

2019-12-29

#General Information

The central dogma of molecular biology is DNA → RNA → Protein. This means that for a given gene, DNA is transcribed into RNA and then RNA is translated into proteins. One way to characterize the level to which genes are active – or "expressed" – is to quantify their RNA abundances in a sample of cells. This provides a snapshot of which biological processes are active. We can do this simultaneously for nearly every gene in the genome, which is called *genome-wide gene expression profiling*.

In this project, I am working with this type of data measured on tumor biopsies from many individuals diagnosed with different types of cancer. The specific genome-wide gene expression profiling technique considered in this project is called RNA-Seq.

The data in this project was obtained from this paper: http://www.nature.com/nbt/journal/v33/n3/full/nbt.3080.html

The raw gene expression measurements were transformed into a measure called "RPKM": http://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/

#Part 1: Data Wrangling

```
library(dplyr)
library(stringr)
library(ggplot2)
library(RColorBrewer)
library(broom)
library(magrittr)
library(reshape2)
library(lattice)
library(caret)
```

```
glioma_melanoma <- read.table("/Users/Theo/Desktop/SML_201/Projects/project_4/glioma_melanoma.txt", sep
head(glioma_melanoma)
```

```
##     gene_id     sample      rpkm
## 1         1 Sample 403  5.954799
## 2       100 Sample 403 10.268384
## 3      1000 Sample 403 19.955654
## 4     10000 Sample 403 18.672745
## 5 100009676 Sample 403  4.467588
## 6     10001 Sample 403 14.879444
```

```
tail(glioma_melanoma)
```

```
##         gene_id     sample        rpkm
## 1075891     999 Sample 243  0.06394188
## 1075892    9990 Sample 243  2.22167900
## 1075893    9991 Sample 243 13.20715000
## 1075894    9993 Sample 243 18.93163000
## 1075895    9994 Sample 243 10.61039000
## 1075896    9997 Sample 243 12.45104000
```

```
with_covariates <- read.table("/Users/Theo/Desktop/SML_201/Projects/project_4/with_covariates.txt", sep=
                              header=TRUE, quote="")
head(with_covariates)
```

```
##   gene_id      sample        rpkm   organ                 disease age    sex
## 1       1 Sample 101 0.07088731   colon   colon adenocarcinoma  56   male
## 2       1 Sample 101 0.07088731   colon   colon adenocarcinoma  56   male
## 3       1 Sample 104 0.22686950   colon   colon adenocarcinoma  65   male
## 4       1 Sample 104 0.22686950   colon   colon adenocarcinoma  65   male
## 5       1 Sample 109 0.13967584 stomach gastric adenocarcinoma  62 female
## 6       1 Sample 109 0.13967584 stomach gastric adenocarcinoma  62 female
##   gene_name
## 1      A1BG
## 2      A1BG
## 3      A1BG
## 4      A1BG
## 5      A1BG
## 6      A1BG
```

```
tail(with_covariates)
```

```
##           gene_id    sample     rpkm organ              disease age    sex
## 3065875 100775107 Sample 92 1.331926 colon colon adenocarcinoma  70   male
## 3065876 100775107 Sample 92 1.331926 colon colon adenocarcinoma  70   male
## 3065877 100775107 Sample 93 3.969307 colon      colon carcinoma  55 female
## 3065878 100775107 Sample 93 3.969307 colon      colon carcinoma  55 female
## 3065879 100775107 Sample 95 2.546954 colon      colon carcinoma  69   male
## 3065880 100775107 Sample 95 2.546954 colon      colon carcinoma  69   male
##         gene_name
## 3065875      LUST
## 3065876      LUST
## 3065877      LUST
## 3065878      LUST
## 3065879      LUST
## 3065880      LUST
```

```
gene_ids <- read.table("gene_ids.txt", sep="\t",
                              header=TRUE, quote="")
head(gene_ids)
```

```
##      gene_name   gene_id
## 1 LOC100288966 100288966
## 2 LOC100134409 100134409
## 3 LOC100507395 100507395
## 4 LOC100507412 100507412
## 5       RN18S1 100008588
## 6      RN5-8S1 100008587
```

```
tail(gene_ids)
```

```
##       gene_name gene_id
## 26005       ND5    4540
## 26006       ND6    4541
## 26007      TRNE    4556
## 26008      CYTB    4519
## 26009      TRNT    4576
## 26010      TRNP    4571
```

```
design <- read.table("design.txt", sep="\t",
                              header=TRUE, quote="")
head(design)
```

```
##   Source.Name Characteristics.cell.line. Characteristics.organism.part.
## 1    Sample 1                      A2780                          ovary
## 2    Sample 2                   COLO 679                           skin
## 3    Sample 3                   COLO 800                           skin
## 4    Sample 4                   COLO 849                           skin
## 5    Sample 5                   Hs 852.T                           skin
## 6    Sample 6                    IPC-298                           skin
##   Characteristics.disease. Characteristics.age. Characteristics.sex.
## 1        ovarian carcinoma        not available               female
## 2                 melanoma                   47               female
## 3                 melanoma                   14                 male
## 4      metastatic melanoma                   43                 male
## 5                 melanoma        not available        not available
## 6                 melanoma                   64               female
##   Characteristics.ethnicity.
## 1              not available
## 2              not available
## 3              not available
## 4                  Caucasian
## 5              not available
## 6              not available
```

```
tail(design)
```

```
##     Source.Name Characteristics.cell.line. Characteristics.organism.part.
## 670  Sample 670                      RMG-I                          ovary
## 671  Sample 671                     RMUG-S                          ovary
## 672  Sample 672                TYK-nu.CP-r                          ovary
## 673  Sample 673                     TYK-nu                          ovary
## 674  Sample 674                     OVMANA                          ovary
## 675  Sample 675                     Calu-1                           lung
##              Characteristics.disease. Characteristics.age. Characteristics.sex.
## 670                 ovarian carcinoma        not available        not available
## 671            ovarian adenocarcinoma                   62               female
## 672                 ovarian carcinoma        not available        not available
## 673                 ovarian carcinoma        not available        not available
## 674 ovarian clear cell adenocarcinoma        not available        not available
## 675                    lung carcinoma                   47               female
##     Characteristics.ethnicity.
## 670              not available
## 671                   Japanese
## 672              not available
## 673              not available
## 674              not available
## 675                  Caucasian
```

```
names(design) <- c("sample","cell_line","organ","disease","age","sex","ethnicity")
```

`glioma_melanoma.txt` contains just gene expression in these two cancers (glioma and melanoma), while `with_covariates.txt` contains additional cancer types which have been filtered to contain only observations where both age and sex recorded. These are two subsets of the above cited very large study. The entire dataset is difficult to fit into memory on ordinary computers, so these subsets were sectioned out of the original dataset.

```
glioma_melanoma_gene_ids <- inner_join(glioma_melanoma,gene_ids, by="gene_id")
glioma_melanoma_tidy <- inner_join(glioma_melanoma_gene_ids,design,by="sample")
```

There are a lot of missing values in `glioma_melanoma.txt`. A small test illustrates the details.

```
not_available <- glioma_melanoma_tidy == "not available"
glioma_melanoma_tidy[not_available] <- NA
```

The way R deals with missing values is with NA. Using “not available” does not really help with the various built-in functions that R has, as those recognize NA. Also, “not available” would be a string or factor, while NA is not a string or a numeric value, but a flag that indicates a missing value. This is helpful in creating vectors and manipulating data accurately. Finally, NA cannot be used in comparisons, unlike other statistical languages that assign a crazy numeric value to a missing value (looking at you SAS!), which might lead to potential errors in our data manipulation.

```
glioma_melanoma_tidy <- select(glioma_melanoma_tidy,-sample)
glioma_melanoma_tidy <- glioma_melanoma_tidy %>% group_by(organ) %>% mutate(sample = paste(organ,1:n(),s
glioma_melanoma_tidy <- glioma_melanoma_tidy %>% select(gene_id,sample,rpkm,gene_name,cell_line,organ,di
```

#Part 2: Becoming familiar with the dataset.

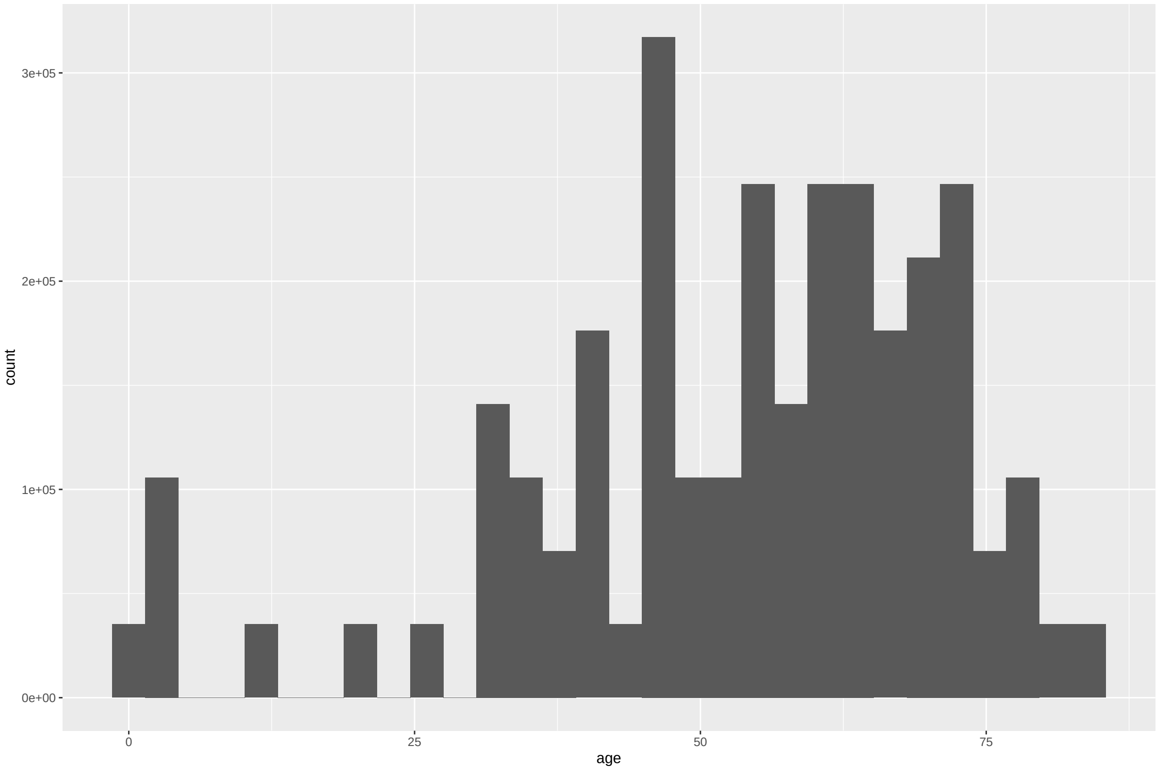
Genes that have gene expression measurements available in `glioma_melanoma.txt` are shown here:

```
length(unique(glioma_melanoma_tidy$gene_id))
```

```
## [1] 14943
```

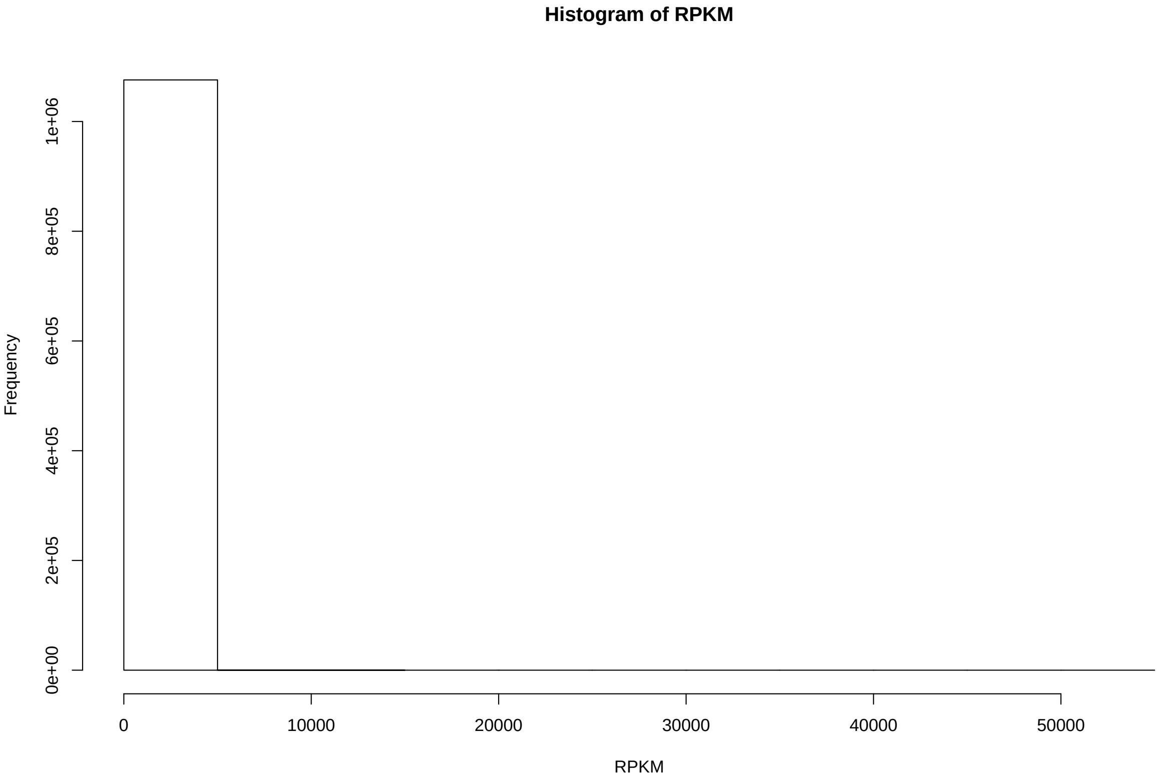
This is a histogram of the recorded ages of the individuals in `with_covariates.txt`.

```
ggplot()+geom_histogram(data= with_covariates, mapping = aes(age))
```



The RPKM gene expression data in `glioma_melanoma.txt` are gathered in a single vector and plotted on a histogram.

```
RPKM <- glioma_melanoma$rpkm
hist(RPKM, breaks=10)
```

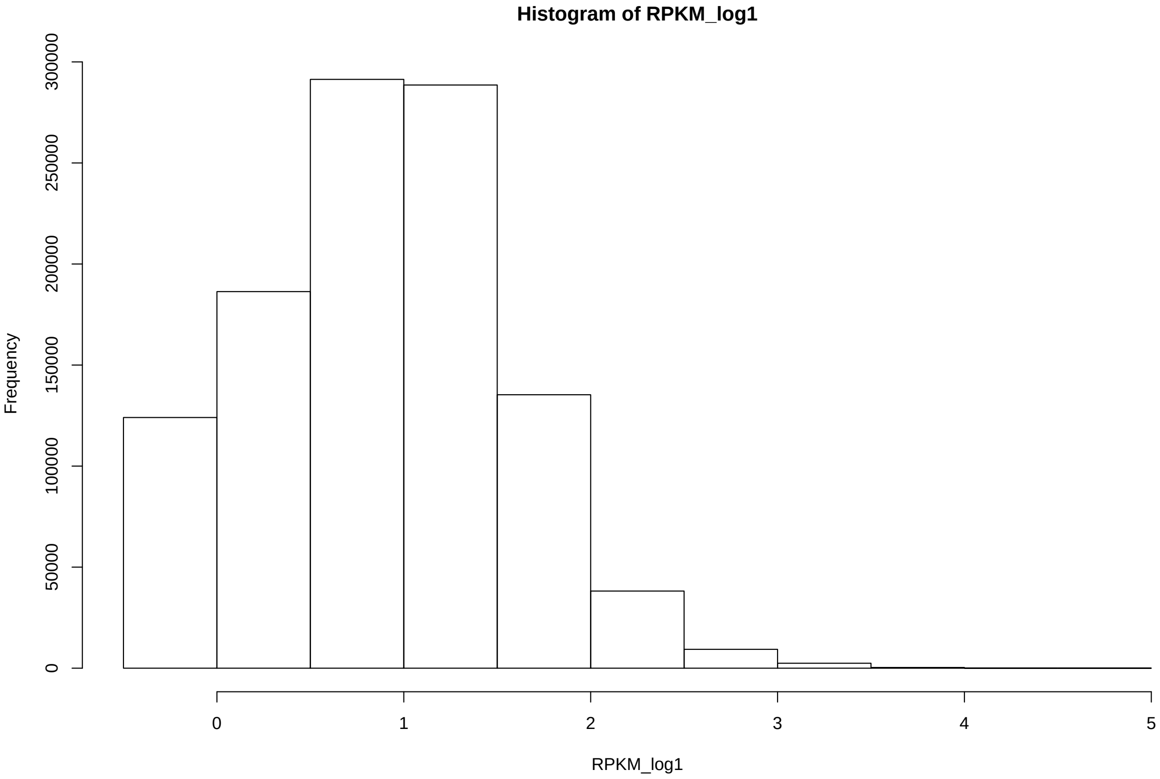


There is

one tall bar on the left because the data is right-skewed.

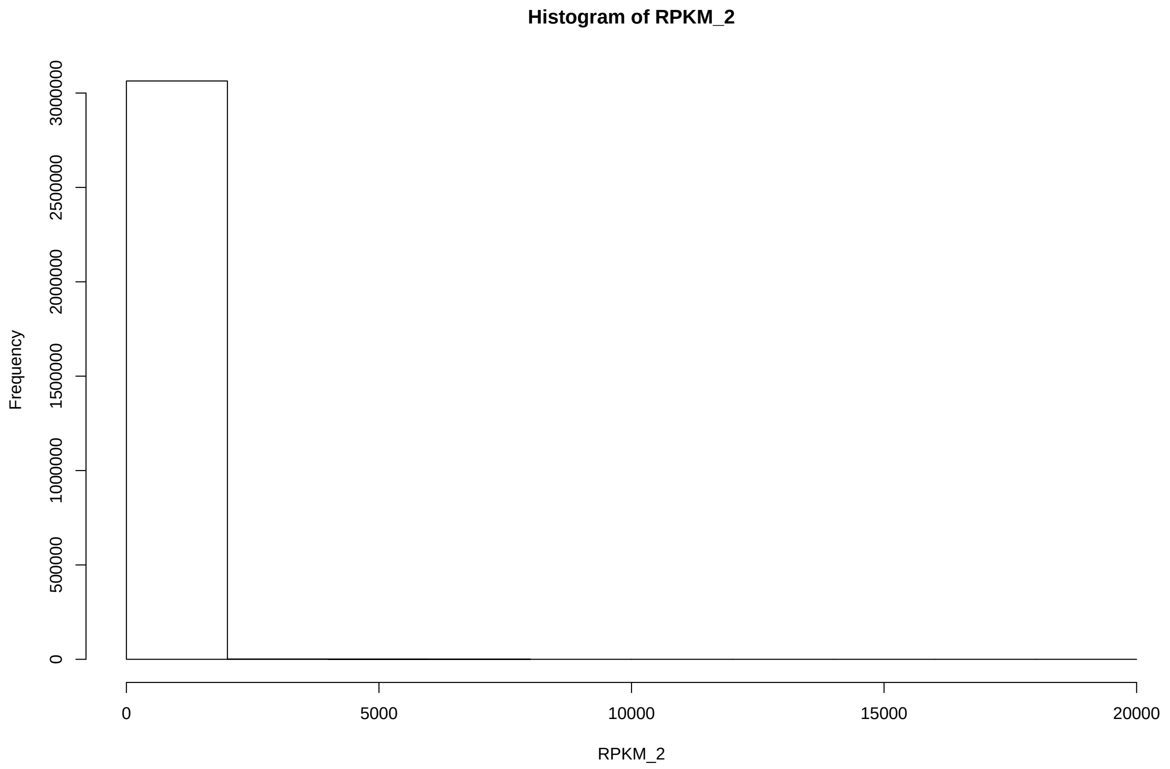
Locations where the data does not look Normal:

```
RPKM_log1 <- log10(RPKM+0.5)
hist(RPKM_log1, breaks=10)
```

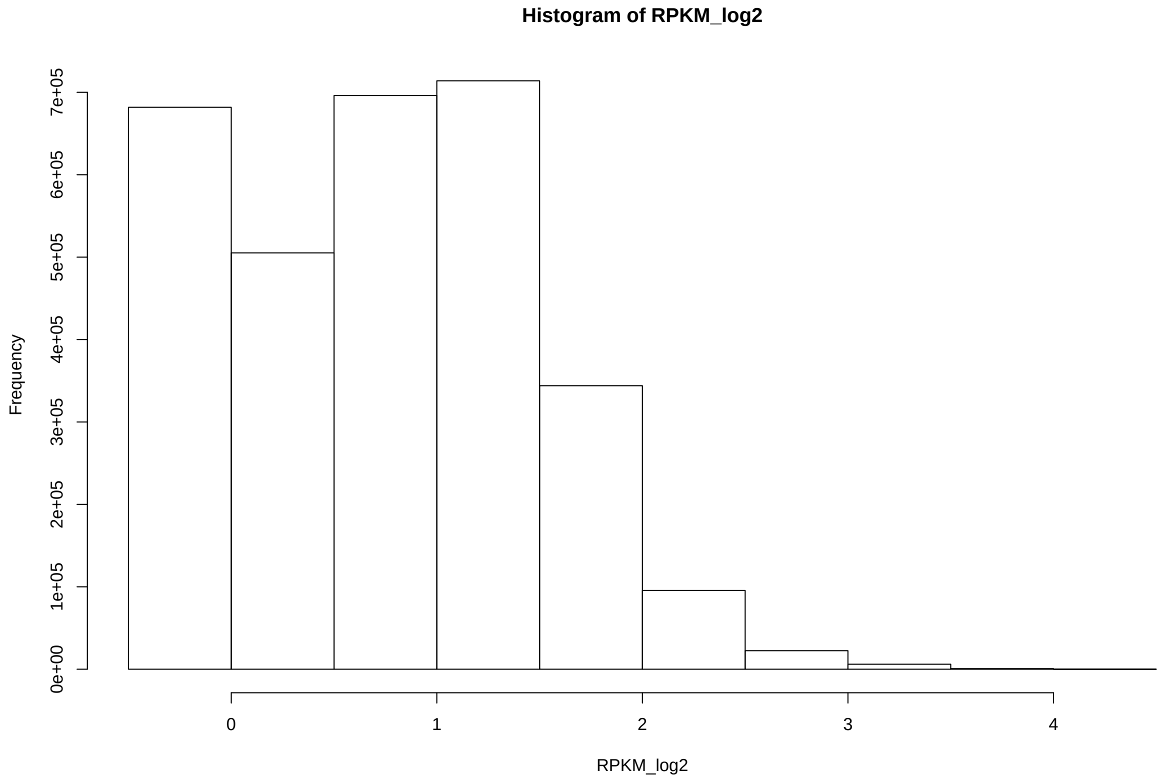


The data does not look normal for the negative numbers. There are a lot of data between 0 and 5 and a lot especially between 0 and 1, giving the data a right skew.

```
RPKM_2 <- with_covariates$rpkm
hist(RPKM_2, breaks=10)
```



```
RPKM_log2 <- log10(RPKM_2+0.5)
hist(RPKM_log2, breaks=10)
```



```r
with_covariates$transformed_RPKM <- log10(with_covariates$rpkm+0.5)
```

The with_covariates data has even more data between 0 and 1 and has a lot of extremely small numbers close to 0, which makes the data right skewed, and thus harder to tranform.

Here, I perform a hypothesis test of whether there a mean difference in gene expression between males and females in `glioma_melanoma.txt`.

```r
male <- subset(glioma_melanoma_tidy, subset = glioma_melanoma_tidy$sex =="male")
female <- subset(glioma_melanoma_tidy, subset = glioma_melanoma_tidy$sex =="female")

femlae_RPKM <- female$transformed_rpkm
male_RPKM <- male$transformed_rpkm

t.test(male_RPKM, femlae_RPKM)
```

```
##
##  Welch Two Sample t-test
##
## data:  male_RPKM and femlae_RPKM
## t = 12.017, df = 306060, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.02141145 0.02975670
## sample estimates:
## mean of x mean of y
## 0.8936354 0.8680513
```

```r
var(femlae_RPKM)
```

```
## [1] 0.4411348
```

```r
var(male_RPKM)
```

```
## [1] 0.448642
```

Assumptions: * The variances of the two populations are equal. * The data are normally distributed. * The data are independent and continuous.

(The assumptions were minimized by looking at the variance of the two samples to see if they are equal. The transformation applied to RPKM above also made the data normally distributed.)

#Part 3: Differences Between Diseases

Glioma is a cancer of the brain, while melanoma is a cancer of the skin. It is interesting to examine differences in gene expression between these two very different diseases to formulate an understanding of what is biologically different between them.

For each gene individually, I perform a hypothesis test of whether there is a population mean difference in expression between the two cancer types.

```r
gene_p_values <- glioma_melanoma_tidy %>% group_by(gene_name) %>% do(t = t.test(.$transformed_rpkm~.$di
names(gene_p_values) <- c("gene_name","p_values")
gene_p_values <- transform(gene_p_values, p_values = as.numeric(p_values))
sapply(gene_p_values, mode)
```

```
## gene_name  p_values
## "numeric" "numeric"
```

```
sapply(gene_p_values, class)
```

```
## gene_name  p_values
##  "factor" "numeric"
```

```
gene_t_values <- glioma_melanoma_tidy %>% group_by(gene_name) %>% do(t = t.test(.$transformed_rpkm~.$di
names(gene_t_values) <- c("gene_name","t_statistics")
gene_t_values <- transform(gene_t_values, t_statistics = as.numeric(t_statistics))
sapply(gene_t_values, mode)
```

```
##    gene_name t_statistics
##    "numeric"    "numeric"
```

```
sapply(gene_t_values, class)
```

```
##    gene_name t_statistics
##     "factor"    "numeric"
```

```
gene_estimates <- glioma_melanoma_tidy %>% group_by(gene_name) %>% do(t = t.test(.$transformed_rpkm~.$d
names(gene_estimates) <- c("gene_name","estimates")
gene_estimates[,"lower_bound_est"] <- NA
gene_estimates[,"upper_bound_est"] <- NA

#extract bounds:
for (i in 1:nrow(gene_estimates)) {
  gene_estimates$lower_bound_est[i] <- gene_estimates[[2]][[i]][[1]]
  gene_estimates$upper_bound_est[i] <- gene_estimates[[2]][[i]][[2]]
}

#get rid of estimate intervals column:
gene_estimates <- select(gene_estimates,-estimates)
#add effect size column:
gene_estimates <- gene_estimates %>% mutate(effect_size = lower_bound_est-upper_bound_est)
sapply(gene_estimates, mode)
```

```
##       gene_name lower_bound_est upper_bound_est     effect_size
##       "numeric"       "numeric"       "numeric"       "numeric"
```

```
sapply(gene_estimates, class)
```

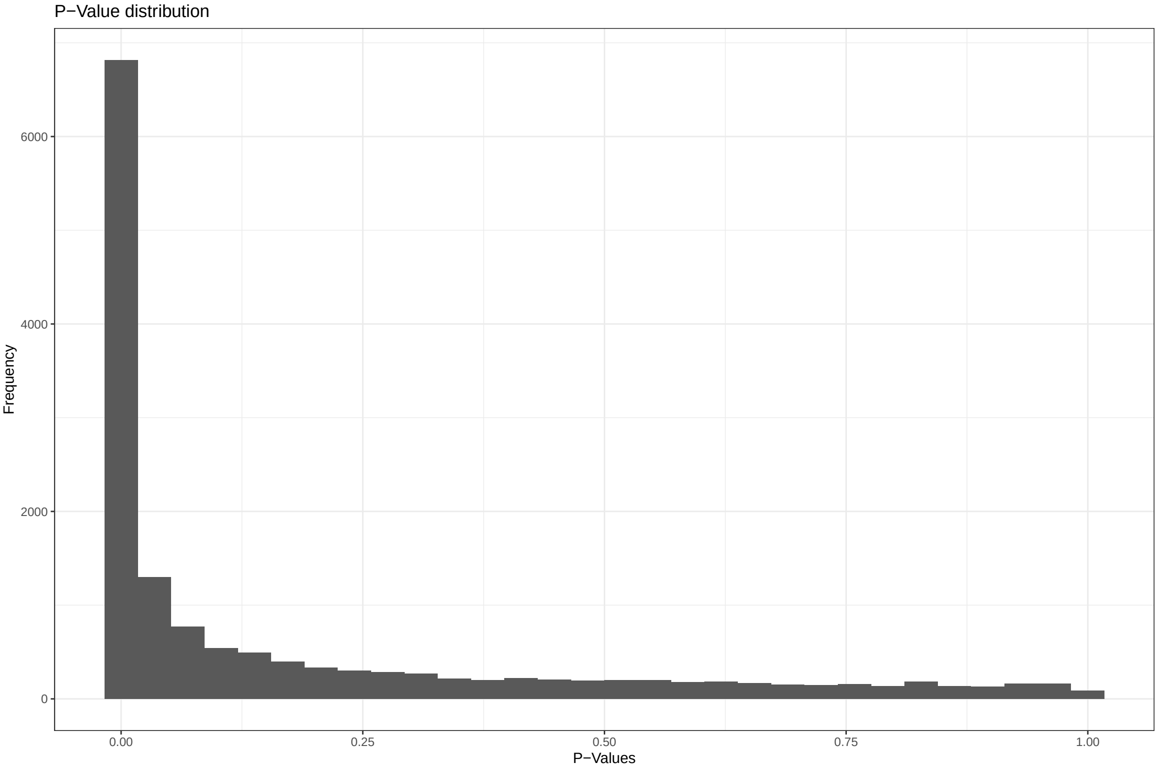
```
##       gene_name lower_bound_est upper_bound_est     effect_size
##        "factor"       "numeric"       "numeric"       "numeric"
```

```
gene_ci <- glioma_melanoma_tidy %>% group_by(gene_name) %>% do(t = t.test(.$transformed_rpkm~.$disease)
names(gene_ci) <- c("gene_name","confidence_intervals")

gene_statistics <- inner_join(gene_p_values,gene_t_values,by="gene_name")
gene_statistics <- inner_join(gene_statistics,gene_ci,by="gene_name")
gene_statistics <- inner_join(gene_statistics,gene_estimates,by="gene_name")

#split confidence intervals into two columns:
gene_statistics[,"lower_bound_ci"] <- NA
gene_statistics[,"upper_bound_ci"] <- NA

#extract bounds:
for (i in 1:nrow(gene_statistics)) {
  gene_statistics$lower_bound_ci[i] <- gene_statistics[[4]][[i]][[1]]
  gene_statistics$upper_bound_ci[i] <- gene_statistics[[4]][[i]][[2]]
}

#get rid of confidence_intervals column:
gene_statistics <- select(gene_statistics,-confidence_intervals)
```

Some thoughts on scale: For this part I used the original scale of the data. I did this to avoid writing more code that would discern the two types of cancer for each gene and executing the t.test function. Since each data point has the same meaning as the original, there is no difference in the interpretation of the t.test above.

This is a histogram of the resulting p-values:

```
#The default one:
ggplot(data=gene_p_values, aes(gene_p_values$p_values)) +
  geom_histogram() +
  xlab("P-Values") +
  ylab("Frequency") +
  ggtitle("P-Value distribution") +
  theme_bw()
```



The distribution of the p-values seems to be greatly skewed to the right, meaning that they are for the most part extremely small, reinforcing the idea that there is a mean difference in the expression of the various genes i.e.null hypothesis is false.

The two genes below are vastly different in how they are expressed among the two cancer types. This boxplot illustrates their value distributions:

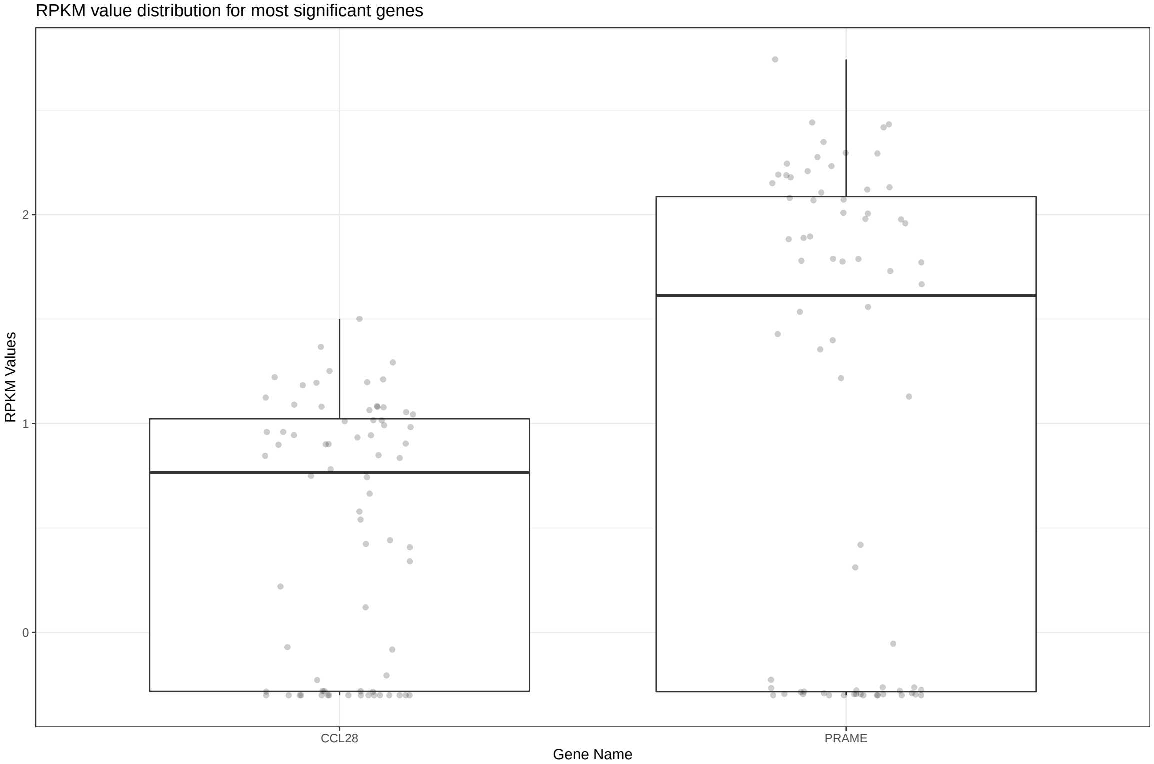
```
#which ones are the most significant?
most_significant <- head(gene_statistics[order(gene_p_values[,2]),])
most_significant
```

```
##          gene_name     p_values t_statistics lower_bound_est upper_bound_est
## 2002         CCL28 2.193933e-27    -17.72099      -0.2605266     0.861153484
## 10109        PRAME 2.119555e-22    -15.54181      -0.2214200     1.694347818
## 4567          GJA1 5.231955e-22     14.02155       1.4917116    -0.009581146
## 10867       RNF125 3.118459e-21    -14.34274      -0.2104248     0.615915719
## 9882        PNPLA4 7.299503e-21    -13.34159      -0.2251384     0.593869723
## 6321  LOC100287493 3.669985e-20     14.68833       0.8257513    -0.113384399
##       effect_size lower_bound_ci upper_bound_ci
## 2002   -1.1216801     -1.2479536     -0.9954067
## 10109  -1.9157678     -2.1624814     -1.6690541
## 4567    1.5012927      1.2877461      1.7148393
## 10867  -0.8263405     -0.9415401     -0.7111409
## 9882   -0.8190081     -0.9414528     -0.6965635
## 6321    0.9391357      0.8108412      1.0674303
```

```
#which ones are the least significant?
least_significant <- head(gene_statistics[order(-gene_p_values[,2]),])
least_significant
```

```
##       gene_name  p_values  t_statistics lower_bound_est upper_bound_est
## 14863    ZNF805 0.9998685  0.0001657914       0.1909715      0.19096706
## 2820       CUL2 0.9996934 -0.0003868319       1.0177825      1.01779900
## 1630    C5orf38 0.9994621 -0.0006780836       0.0124572      0.01253225
## 12534    STAT5B 0.9994375  0.0007083869       1.0456512      1.04561812
## 8829      NINJ1 0.9993887  0.0007703239       1.3244660      1.32440626
## 4365     FUNDC2 0.9993472 -0.0008211344       1.0469953      1.04701881
##         effect_size lower_bound_ci upper_bound_ci
## 14863  4.427082e-06    -0.05383283     0.05384168
## 2820  -1.650817e-05    -0.08637851     0.08634549
## 1630  -7.505265e-05    -0.22333438     0.22318428
## 12534  3.311395e-05    -0.09379222     0.09385845
## 8829   5.973639e-05    -0.15608997     0.15620944
## 4365  -2.351955e-05    -0.05715461     0.05710757
```

```
#extract data for genes of interest
important_genes <- glioma_melanoma_tidy %>% filter(gene_name == "CCL28" | gene_name == "PRAME")
important_genes <- important_genes %>% group_by(gene_name)

#assemble plot
ggplot(important_genes, aes(gene_name,transformed_rpkm)) +
  geom_boxplot() +
  xlab("Gene Name") +
  ylab("RPKM Values") +
  ggtitle("RPKM value distribution for most significant genes") +
  geom_jitter(width = 0.15, alpha = 0.2) +
  theme_bw()
```



Testing for problematic values:

```
which(is.infinite(gene_p_values$p_values))
```

```
## integer(0)
```

```
which(is.infinite(gene_t_values$t_statistics))
```

```
## integer(0)
```

```
which(is.nan(gene_p_values$p_values))
```

```
## integer(0)
```

```
which(is.nan(gene_t_values$t_statistics))
```

```
## integer(0)
```

The following plot shows the 95% confidence intervals for the mean difference in the five most significant genes and the five least significant genes.

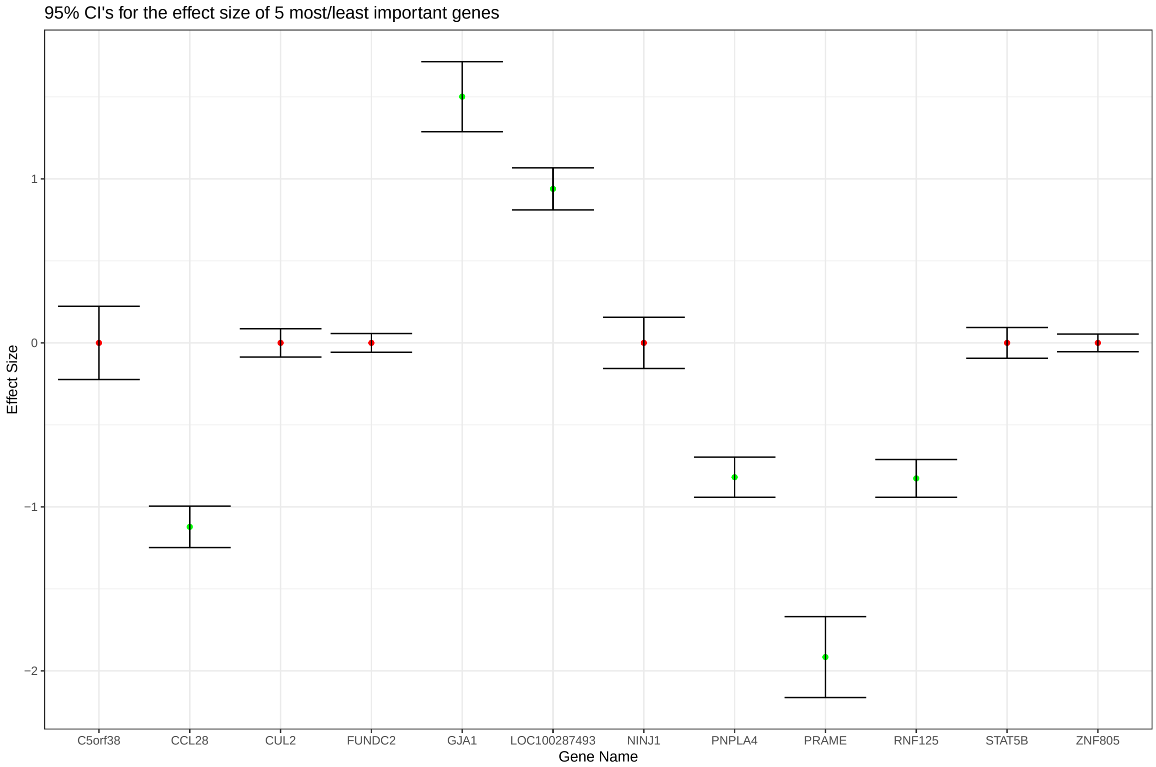
```
#which ones are the most significant?
most_significant
```

```
##          gene_name     p_values t_statistics lower_bound_est upper_bound_est
## 2002         CCL28 2.193933e-27    -17.72099      -0.2605266     0.861153484
## 10109        PRAME 2.119555e-22    -15.54181      -0.2214200     1.694347818
## 4567          GJA1 5.231955e-22     14.02155       1.4917116    -0.009581146
## 10867       RNF125 3.118459e-21    -14.34274      -0.2104248     0.615915719
## 9882        PNPLA4 7.299503e-21    -13.34159      -0.2251384     0.593869723
## 6321  LOC100287493 3.669985e-20     14.68833       0.8257513    -0.113384399
##       effect_size lower_bound_ci upper_bound_ci
## 2002   -1.1216801     -1.2479536     -0.9954067
## 10109  -1.9157678     -2.1624814     -1.6690541
## 4567    1.5012927      1.2877461      1.7148393
## 10867  -0.8263405     -0.9415401     -0.7111409
## 9882   -0.8190081     -0.9414528     -0.6965635
## 6321    0.9391357      0.8108412      1.0674303
```

```
#which ones are the least significant?
least_significant
```

```
##       gene_name  p_values  t_statistics lower_bound_est upper_bound_est
## 14863    ZNF805 0.9998685  0.0001657914       0.1909715      0.19096706
## 2820       CUL2 0.9996934 -0.0003868319       1.0177825      1.01779900
## 1630    C5orf38 0.9994621 -0.0006780836       0.0124572      0.01253225
## 12534    STAT5B 0.9994375  0.0007083869       1.0456512      1.04561812
## 8829      NINJ1 0.9993887  0.0007703239       1.3244660      1.32440626
## 4365     FUNDC2 0.9993472 -0.0008211344       1.0469953      1.04701881
##         effect_size lower_bound_ci upper_bound_ci
## 14863  4.427082e-06    -0.05383283     0.05384168
## 2820  -1.650817e-05    -0.08637851     0.08634549
## 1630  -7.505265e-05    -0.22333438     0.22318428
## 12534  3.311395e-05    -0.09379222     0.09385845
## 8829   5.973639e-05    -0.15608997     0.15620944
## 4365  -2.351955e-05    -0.05715461     0.05710757
```

```
library(ggplot2)
#plot of the t_statistics and the corresponding error bars:
ggplot(data = NULL, mapping = aes(x = gene_name, y = effect_size)) +
  geom_point(data = most_significant, color = "green") +
  geom_errorbar(data=most_significant, aes(ymin = lower_bound_ci, ymax = upper_bound_ci)) +
  geom_point(data = least_significant, color = "red") +
  geom_errorbar(data=least_significant, aes(ymin = lower_bound_ci, ymax = upper_bound_ci)) +
  ggtitle("95% CI's for the effect size of 5 most/least important genes") +
  xlab("Gene Name") +
  ylab("Effect Size") +
  theme_bw()
```



It is worthwhile noting that all of the estimates fall within their correspodnign confidence intervals. Also, all of the least significant genes have a very low effect size (makes sense) and the most significant ones are scattered on both sides.

Securing the super-senior risk of pure causality between the selected genes:

```
theoretical_cutoff <- nrow(gene_p_values)*0.05
theoretical_cutoff
```

```
## [1] 747.05
```

In the case where all null hypotheses are true, then everything would be purely chance and 5% of the p-values observed will be less than 0.05.

I confirm that there is mean difference in the selected expressions by comparing the theoretical cutoff to the number of p-values less than 0.05 observed in the data:

```
actual_cutoff <- gene_p_values %>% filter(p_values < 0.05)
actual_cutoff <- nrow(actual_cutoff)
actual_cutoff
```

```
## [1] 8072
```

The above quantity is the total p-values < 0.05 calculated originally. If all of the p-values observed were all larger than 0.05 then the null hypothesis could not be rejected. More values reject the null hypothesis than confirm it (8072 p-values < 0.05), leading us to the potential conclusion that there might actually be mean difference in the expression of genes among different cancer types.

This is a volcano plot, with observed mean difference (the effect size) on the x-axis and the $-\log_1 0$ transformed p-values on the y-axis. Link](http://www.r-bloggers.com/using-volcano-plots-in-r-to-visualize-microarray-and-rna-seq-results/).

```
#transform t_statistics
var_glioma <- var(filter(glioma_melanoma_tidy, disease == "glioma")$rpkm)
```

```
var_glioma
```

```
## [1] 14284.23
```

```
var_melanoma <- var(filter(glioma_melanoma_tidy, disease == "melanoma")$rpkm)
var_melanoma
```

```
## [1] 41801.44
```

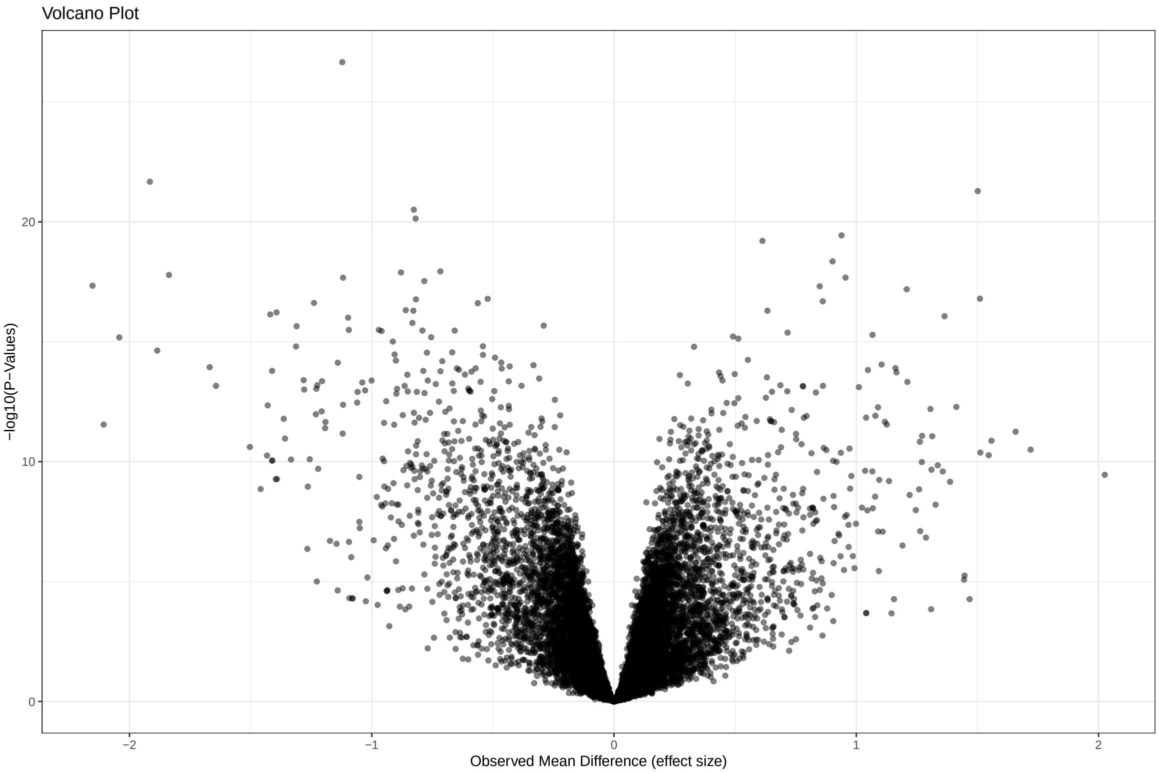
```
size_glioma <- nrow(filter(glioma_melanoma_tidy, disease == "glioma"))
size_glioma
```

```
## [1] 343689
```

```
size_melanoma <- nrow(filter(glioma_melanoma_tidy, disease == "melanoma"))
size_melanoma
```

```
## [1] 732207
```

```
gene_statistics <- gene_statistics %>% mutate(mean_difference = t_statistics*sqrt((var_glioma^2/size_gl



#a simple plot that accounts for the overplotting that occurs:
ggplot(data = gene_statistics, aes(x = effect_size, y = -log10(p_values))) +
  geom_point(alpha = 0.5) +
  theme_bw() +
  ggtitle("Volcano Plot") +
  xlab("Observed Mean Difference (effect size)") +
  ylab("-log10(P-Values)")
```



The regions that are statistically and practically significant lie on the edges of the two “tails” that were generated and, ideally, as far away from the main shape as possible. The points on either the left- or right-hand side exhibit a large magnitude fold change (putting them to the left- or right- of center) as well as high statistical significance (putting them closer to the top of the graph).

#Part 4: Modelling Gene Expression

I begin by fitting a model regressing RPKM on organ affected and age (long execution times).

```
library(broom)

fit_model <- function(t) {
  m = lm(lm(transformed_RPKM~organ+age, t))
  return(tidy(m))
}

m <- with_covariates %>%
  group_by(gene_name) %>%
  do(fit_model(.))

head(m)
```

```
## # A tibble: 6 x 6
## # Groups:   gene_name [1]
##   gene_name   term           estimate std.error statistic  p.value
##   <fct>       <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 1/2-SBSRNA4 (Intercept)    0.0420    0.0276       1.52  0.130
## 2 1/2-SBSRNA4 organcolon    -0.0588    0.0385      -1.53  0.128
## 3 1/2-SBSRNA4 organovary    -0.0803    0.0357      -2.25  0.0258
## 4 1/2-SBSRNA4 organpancreas -0.137     0.0385      -3.57  0.000464
## 5 1/2-SBSRNA4 organstomach  -0.0702    0.0361      -1.95  0.0533
## 6 1/2-SBSRNA4 age            0.000282  0.000566     0.499 0.618
```

```
tail(m)
```

```
## # A tibble: 6 x 6
## # Groups:   gene_name [1]
##   gene_name term           estimate std.error statistic  p.value
##   <fct>     <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 ZZZ3      (Intercept)    1.14      0.0402      28.4   4.04e-66
## 2 ZZZ3      organcolon    -0.165     0.0561      -2.94  3.78e- 3
## 3 ZZZ3      organovary    -0.0363    0.0520      -0.697 4.87e- 1
## 4 ZZZ3      organpancreas -0.115     0.0561      -2.04  4.24e- 2
## 5 ZZZ3      organstomach  -0.0941    0.0526      -1.79  7.54e- 2
## 6 ZZZ3      age            0.000791  0.000824     0.960 3.38e- 1
```

The intercept is the expected value of Y when Organ is at the baseline category, brain, and age is held to 0. The coefficient on organcolon, organovary, organpancreas, and organstomach all represent the change in Y with respect to the baseline category, brain, holding all else constant. The coefficient on the age term represents the expected change in Y for a 1 unit increase in age, holding all else constant.

Sample model fit:

```
ACTB <- subset(with_covariates, subset = with_covariates$gene_name=="ACTB")

model2 <- lm(transformed_RPKM~organ+age, data=ACTB)
```

- All values reported for `(Intercept)`, `age`, and `organpancreas`. This includes `Estimate`, `Std. Error`, `t value`, and `Pr(>|t|)`
- `Multiple R-squared`

```r
summary(model2)$coefficients[1,]
```

```
##      Estimate    Std. Error       t value      Pr(>|t|) 
##  3.331520e+00  5.039781e-02  6.610446e+01 3.529407e-122
```

```r
summary(model2)$coefficients[6,]
```

```
##     Estimate   Std. Error      t value     Pr(>|t|) 
## 0.0002983308 0.0010323295 0.2889879275 0.7729469058
```

```r
summary(model2)$coefficients[4,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|) 
## 0.07272315 0.07023605 1.03541066 0.30196526
```

```r
summary(model2)$r.squared
```

```
## [1] 0.08310237
```

The intercept represents the value of transformed RPKM for organBrain, which is 3.331520. This estimate of the transformed RPKM has a standard error of 7.258113e-02. The t value tests the hypotheses that the corresponding population parameters are 0. A large t-value, 4.590063e+01, shows that the parameters are not 0. The p-value is the probability of obtaining a t-value greater than the absolute value of the t-value we obtained. The probability of that for organBrain is 9.487170e-60, or insignificant.

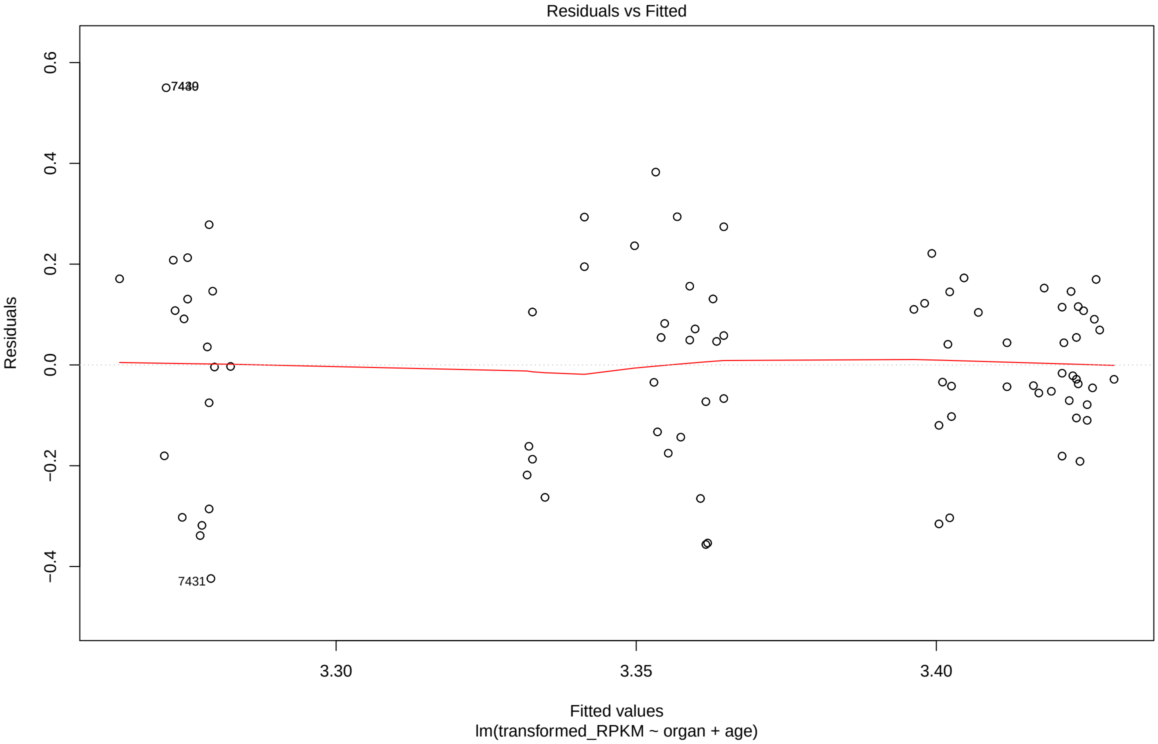
The coefficient on the age variable represents the value of transformed RPKM for a 1 unit increase in age, which is 0.0002983308 added to the intercept, holding all else constant. This estimate of the transformed RPKM has a standard error of 0.0014867244. The t value tests the hypotheses that the corresponding population parameters are 0. A small t-value, 0.2006631239, shows that the parameters could be 0. The p-value is the probability of obtaining a t-value greater than the absolute value of the t-value we obtained. The probability of that for age is 0.8414649901, very big. Meaning that the coefficient on age is not statistically discernable from 0.

The coefficient on the variable organPancreas represents the value of trasnformed RPKM compared to the baseline category organBrain which has a value of 3.331520. The organPacreas variable represents an increase of 0.07272315 in transformed RPKM compared to the intercept. This estimate of the RPKM has a standard error of 0.10115147. The t value tests the hypotheses that the corresponding population parameters are 0. A small t-value, 0.71895300, shows that the parameter could be 0. The p-value is the probability of obtaining a t-value greater than the absolute value of the t-value we obtained. The probability of that for organPancreas is 0.47423890, large. This means that the coefficient on organPancreas is not statistically discernable from 0.

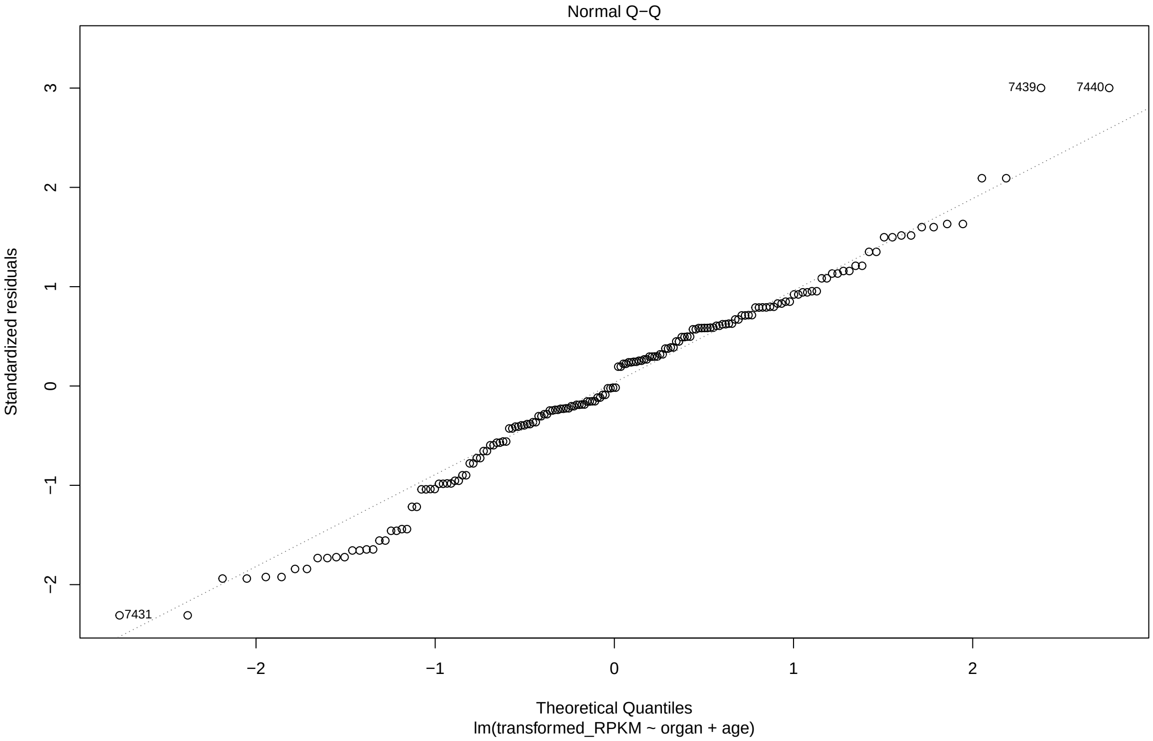
The r-squared value can be thought of as the percent of the variance in the data explained by our model. The model above has a r-squared value of 0.08310237, meaning that the model only explains 8.3% of the variance in the data.

Key assumptions on inference:

```r
plot(model2, which=1)
```



```
plot(model2, which=2)
```

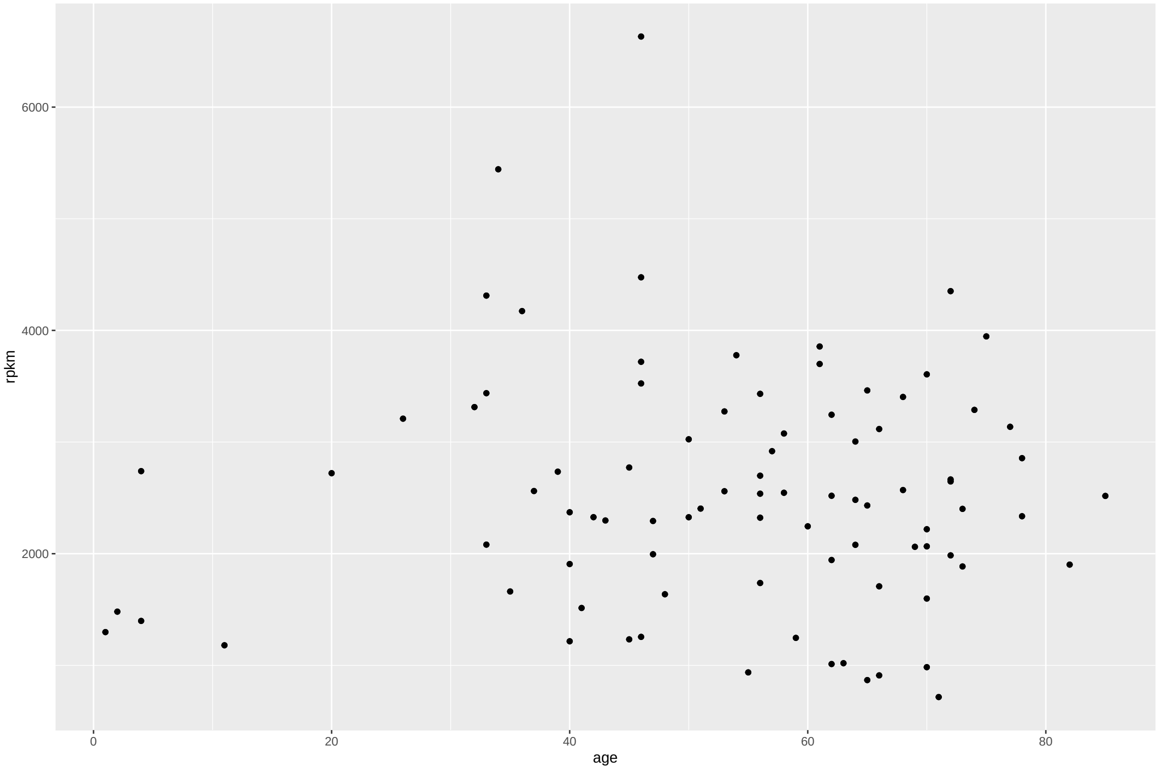


The residuals and fitted values do appear to have some trends with respect to each other. There are 3 clusters apparent.

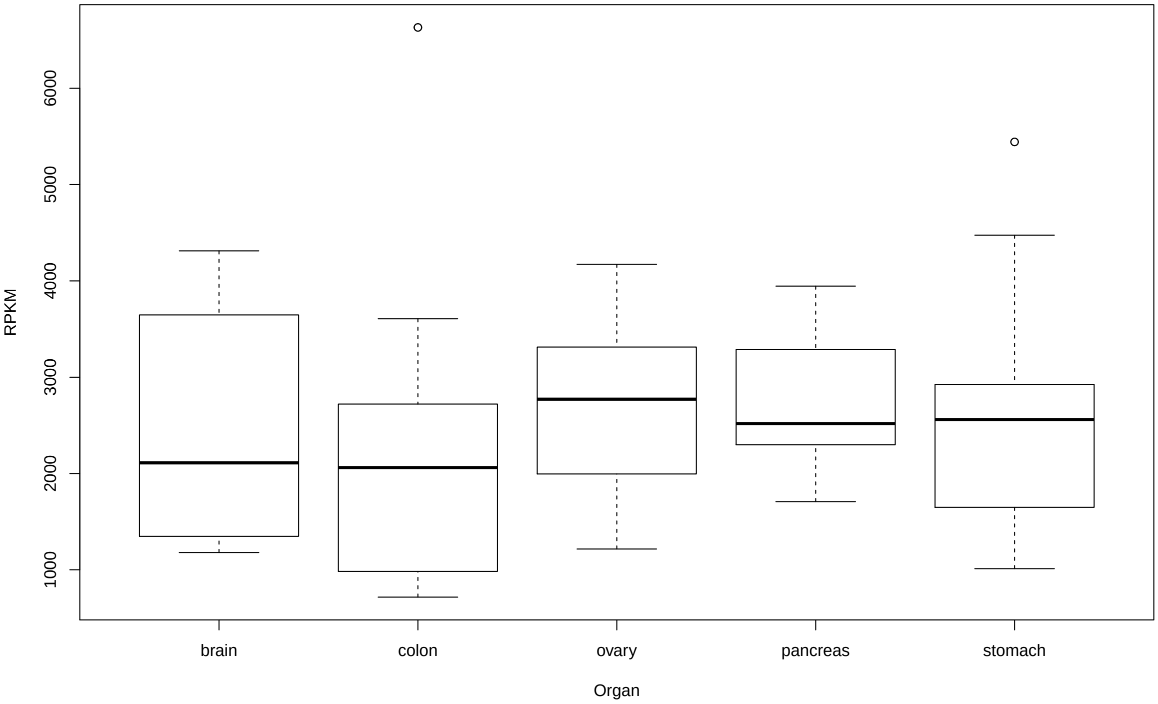
The residuals do appear to be distributed normally.

Plotting against age:

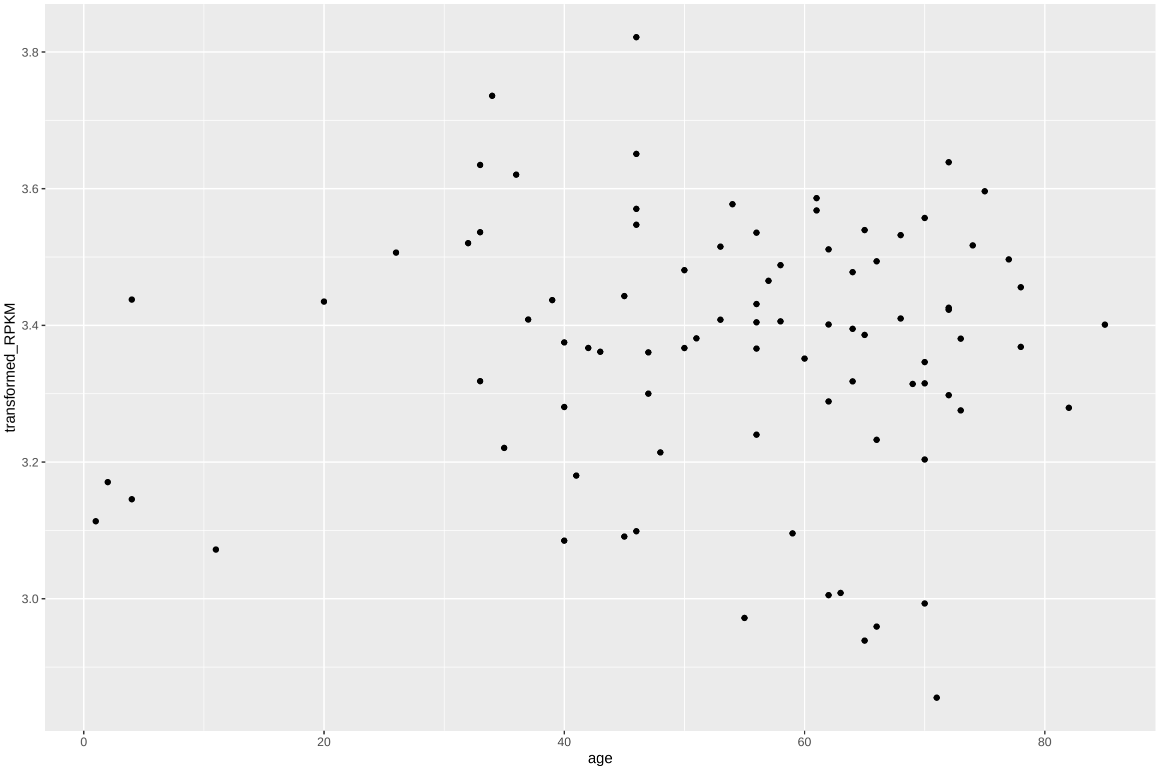
```
#HEAD
ggplot(ACTB)+geom_point(mapping=aes(age, rpkm))
```



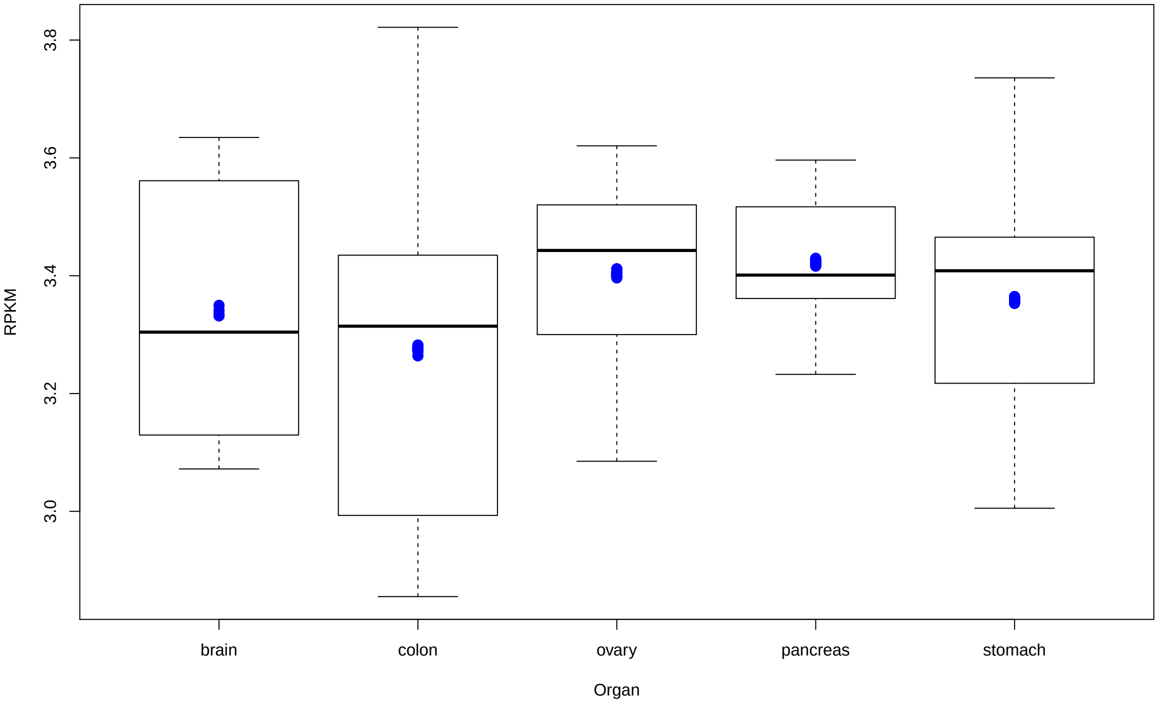
```
plot(ACTB$organ, ACTB$rpkm, xlab="Organ", ylab="RPKM")
```



```
####################
ggplot(ACTB)+geom_point(mapping=aes(age, transformed_RPKM))
```

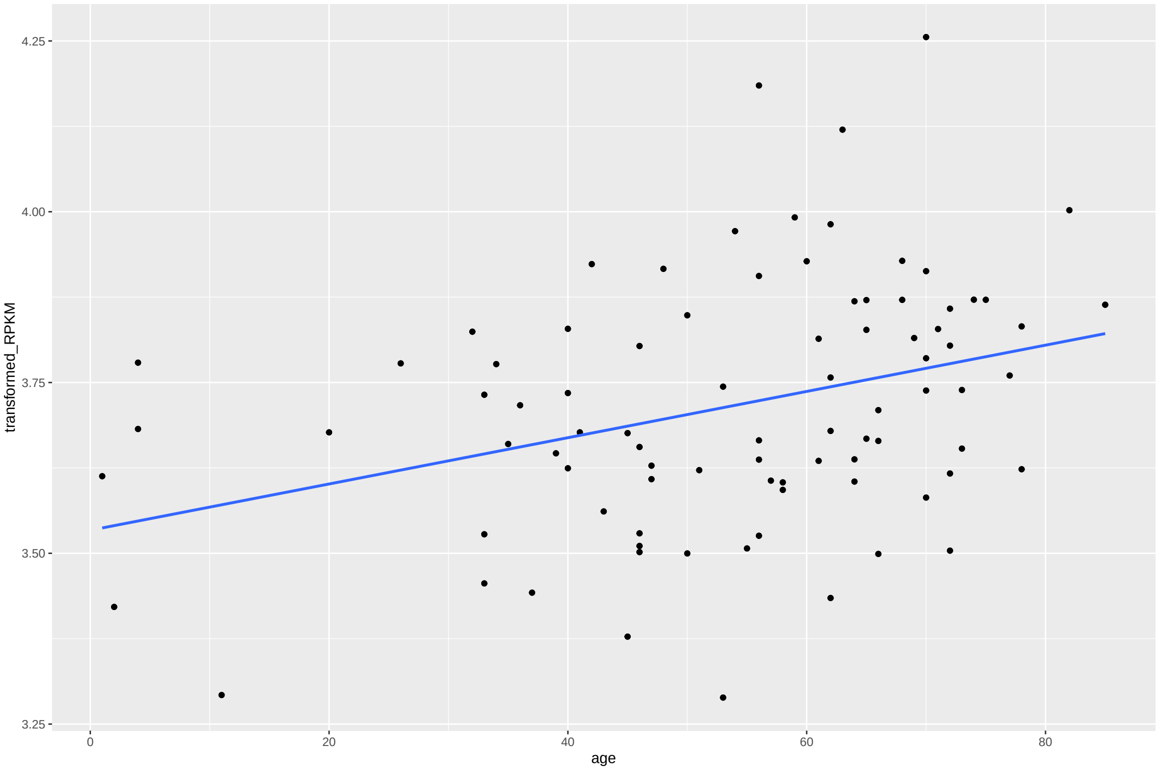


```
plot(ACTB$organ, ACTB$transformed_RPKM, xlab="Organ", ylab="RPKM")
#c77c2f809e696e695db121ab1f418396ae39d043
points(ACTB$organ, model2$fitted.values, col="blue", pch=20, cex=2)
```



Sample gene with large effect size:

```
ATP <- subset(with_covariates, subset= with_covariates$gene_name=="ATP8")
ggplot(ATP, mapping=aes(age, transformed_RPKM))+geom_point(mapping=aes(age, transformed_RPKM))+geom_smo
```



In the data for ATP8, RPKM increases with age. There are some irregularities, e.g. a point at 70 years of age that looks to be an outlier and there are multiple other points after the age of 50 that heavily influence the fit of the line.

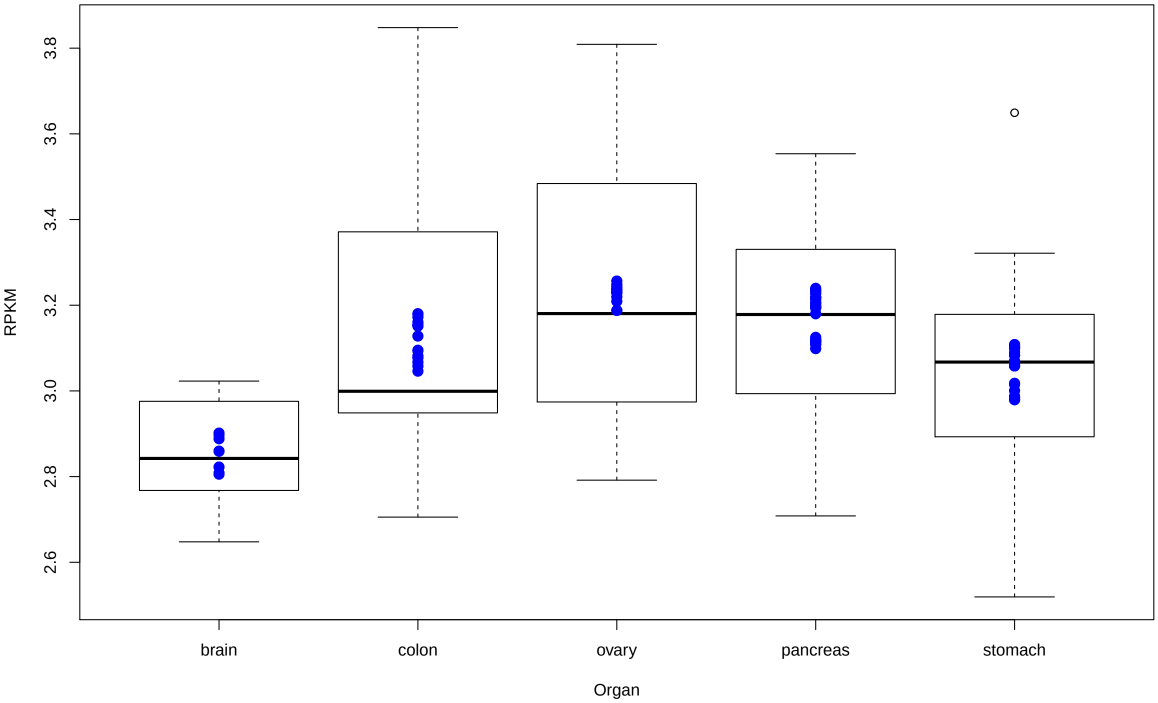
Potential variables for the linear model:

```
fit_model <- function(t) {
  m = lm(lm(transformed_RPKM~organ+age+sex, t))
  return(tidy(m))
}
model.1 <- with_covariates %>%
  group_by(gene_name) %>%
  do(fit_model(.))
fit_model <- function(t) {
  m = lm(lm(transformed_RPKM~organ+age+sex+age:sex, t))
  return(tidy(m))
}
model.2<-with_covariates %>%
  group_by(gene_name) %>%
  do(fit_model(.))
```

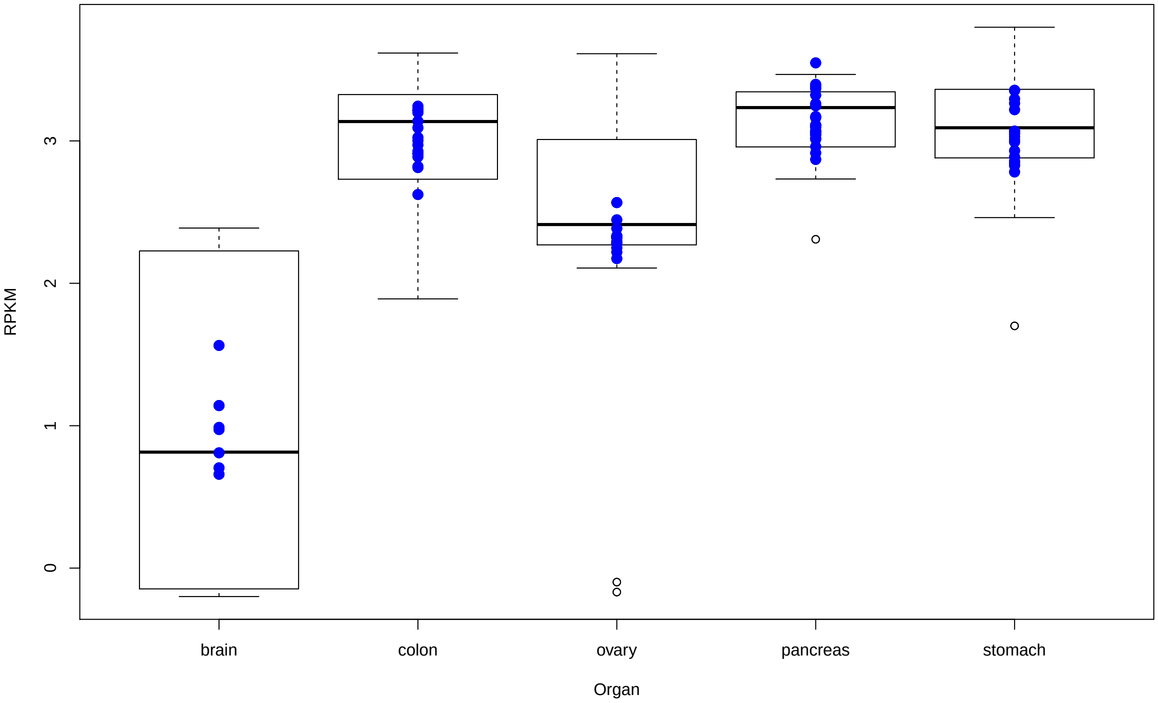
Model results:

```
TRNE <- subset(with_covariates, subset = gene_name == "TRNE")
KRT18 <- subset(with_covariates, subset = gene_name == "KRT18")
model.trne <- lm(transformed_RPKM~organ+age+sex, data=TRNE)
model.KRT18 <- lm(transformed_RPKM~organ+age+sex+age:sex, data = KRT18)
```

```
plot(TRNE$organ, TRNE$transformed_RPKM, xlab="Organ", ylab="RPKM")
points(TRNE$organ, model.trne$fitted.values, col="blue", pch=20, cex=2)
```



```
plot(KRT18$organ, KRT18$transformed_RPKM, xlab="Organ", ylab="RPKM")
points(KRT18$organ, model.KRT18$fitted.values, col="blue", pch=20, cex=2)
```



- TRNE is notable for its large coefficient for organOvary that was very significant and had the largest effect for a variable in model1.
- KRT18 also had a large coefficient for organStomach and organPancreas that was very statistically

significant and had the largest effect for a variable in model2.

To test the two models, I use ANOVA:

```
LOC <- subset(with_covariates, subset=with_covariates$gene_name=="KRT18")
model.loc <- lm(transformed_RPKM~organ+age+sex, data=LOC)
model.loc2 <- lm(transformed_RPKM~organ+age+sex+age:sex, data=LOC)

anova(model.loc, model.loc2)
```

```
## Analysis of Variance Table
##
## Model 1: transformed_RPKM ~ organ + age + sex
## Model 2: transformed_RPKM ~ organ + age + sex + age:sex
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    167 68.529
## 2    166 67.752  1   0.77656 1.9027 0.1696
```

With a p=value of 0.3442 the null hypothesis i.e. the additional terms have coefficients equal to zero is retained. The first model is not significantly different from the second model at a 0.05 significance level.

#Part 5: Prediction

p53 (TP53) is a well-known oncogene (gene associated with the development of cancer).

Testing prediction accuracy:

```
set.seed(201)
TP53_data <- with_covariates %>% filter(gene_name == "TP53")
TP53_data <- TP53_data %>% mutate(transformed_rpkm = log10(rpkm+0.5))
TP53_Train <- createDataPartition(TP53_data$transformed_rpkm, p=0.6, list=FALSE)
training <-TP53_data[TP53_Train,]
testing <-TP53_data[-TP53_Train,]
TP53_model_fit_lm <- train(transformed_rpkm~age, data=training, method="lm")
TP53_model_fit_lm
```

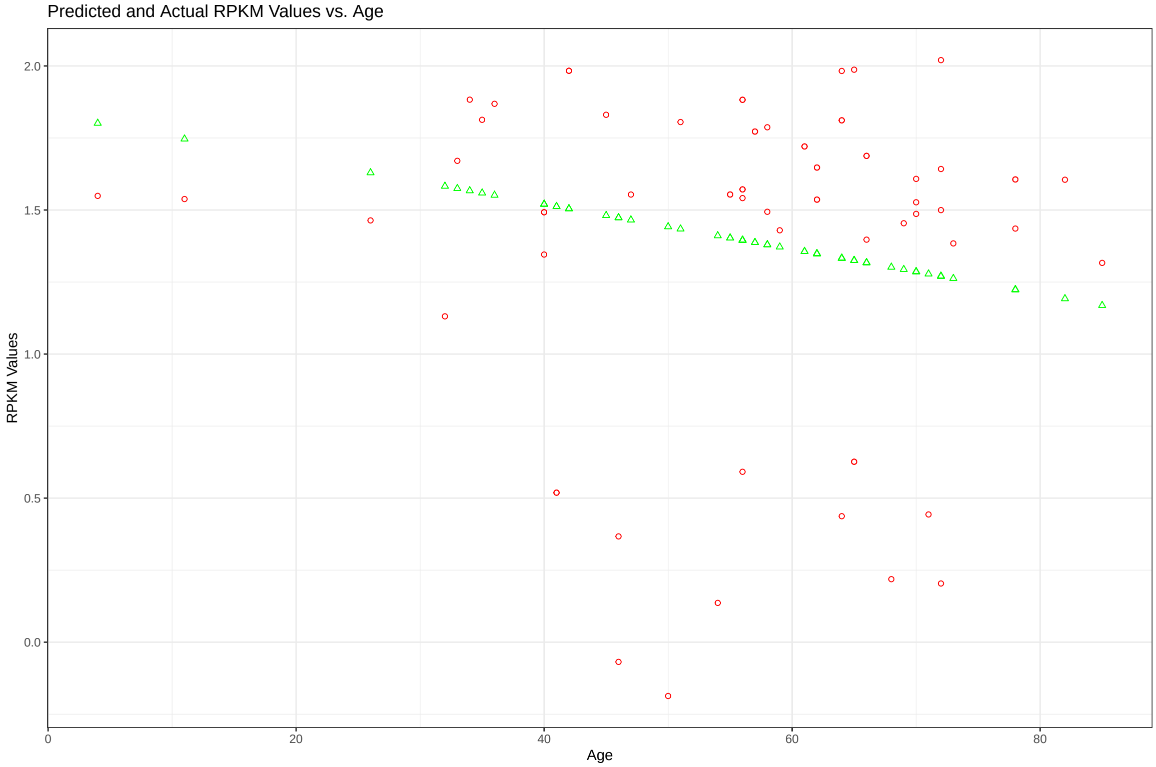
```
## Linear Regression
##
## 107 samples
##   1 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 107, 107, 107, 107, 107, 107, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.4959868  0.1002899  0.3640989
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
#now we calculate the predictions:
predictions <- predict(TP53_model_fit_lm, newdata = testing)
predictions <- as.data.frame(predictions)
names(predictions) <- c("predicted_rpkm")

#create matrix of actual and predicted values:
actual <- testing %>% select(transformed_rpkm,age)
```

```
actual_predicted <- cbind(predictions,actual)
```

Prediction model:

```
#main plot:
ggplot(data = actual_predicted) +
  geom_point(aes(x = age, y = transformed_rpkm, shape = "b"),color = "red") +
  geom_point(aes(x = age, y = predicted_rpkm, shape = "c"),color = "green") +
  scale_shape(solid = FALSE) +
  ggtitle("Predicted and Actual RPKM Values vs. Age") +
  xlab("Age") +
  ylab("RPKM Values") +
  theme_bw() +
  theme(legend.position="none")
```



Limitations: This model utilizes the lm method within train() does not use the testing set in any way, and is therefore a biased measure of error. lm() objects or a different package (such as MLR that has more functionalities than caret) could provide an unbiased measure of error.

… #Part 6: PCA and Clustering

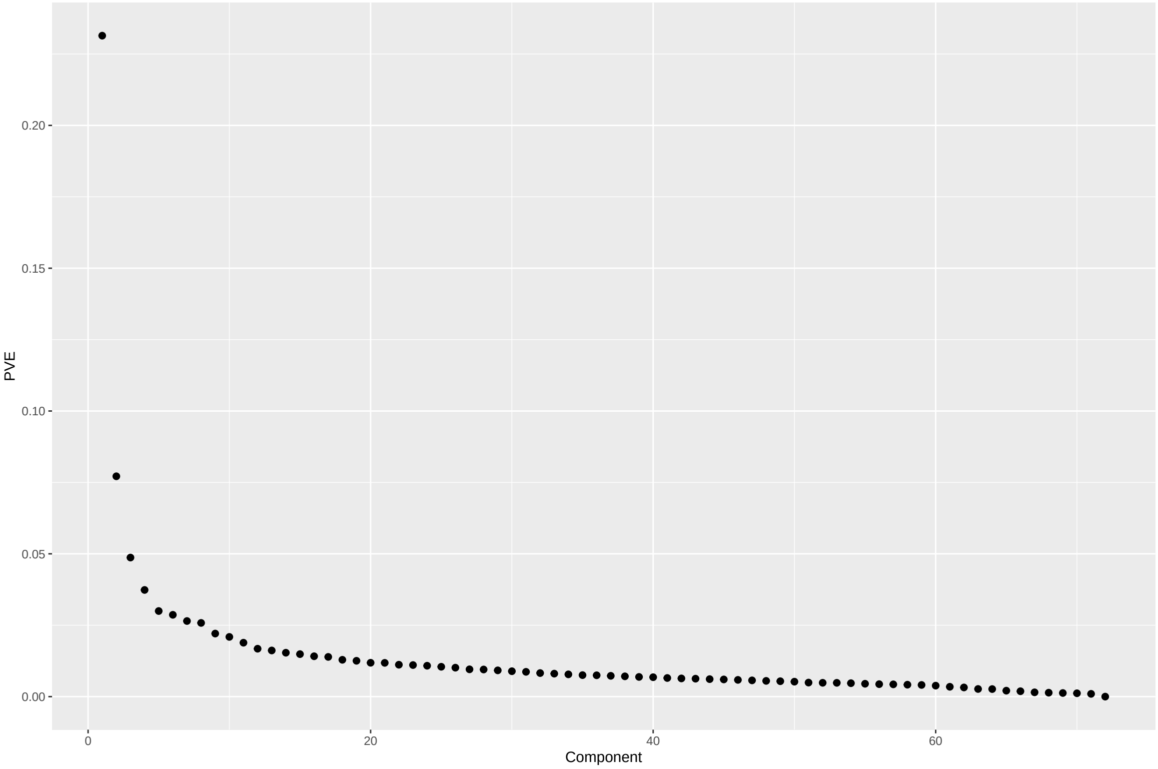
##PCA

```
glioma_melanoma$transformed_RPKM <- log10(glioma_melanoma$rpkm+0.5)

pca_dat_raw = glioma_melanoma[,c("sample","transformed_RPKM", "gene_id")]
pca_dat = acast(pca_dat_raw, gene_id ~ sample, value.var = "transformed_RPKM")

pca <- function(x, space=c("rows", "columns"),
 center=TRUE, scale=FALSE) {
 space <- match.arg(space)
 if(space=="columns") {x <- t(x)}
 x <- t(scale(t(x), center=center, scale=scale))
 s <- svd(x)
 loading <- s$u
 colnames(loading) <- paste0("Loading", 1:ncol(loading))
 rownames(loading) <- rownames(x)
 pc <- diag(s$d) %*% t(s$v)
 rownames(pc) <- paste0("PC", 1:nrow(pc))
 colnames(pc) <- colnames(x)
 pve <- s$d^2 / sum(s$d^2)
 if(space=="columns") {pc <- t(pc); loading <- t(loading)}
 return(list(pc=pc, loading=loading, pve=pve))
 }

mypca <- pca(pca_dat, space="rows")
```

PCA relies on the assumption of normally distributed data, and the transformed RPKM data is suitable for PCA.

Proportion of variance:

```
data.frame(Component=1:length(mypca$pve), PVE=mypca$pve) %>%
 ggplot() + geom_point(aes(x=Component, y=PVE), size=2)
```



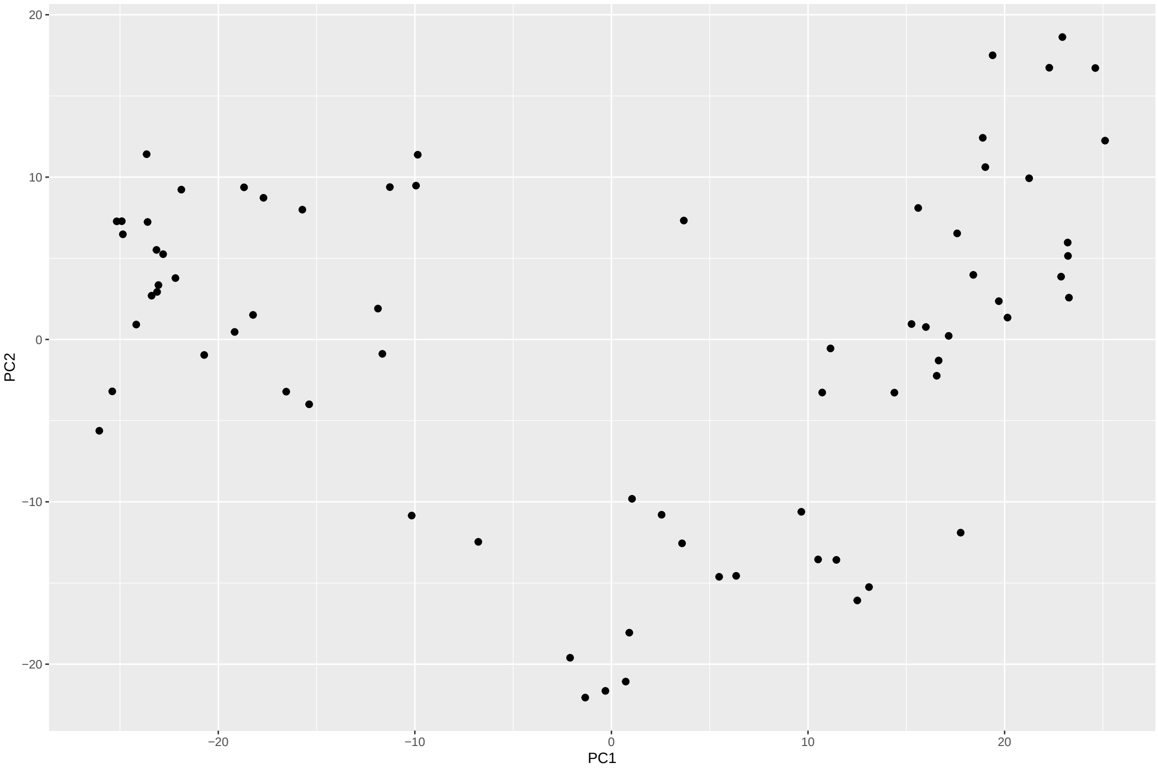
```
sum(mypca$pve[1:45])
```

```
## [1] 0.9055485
```

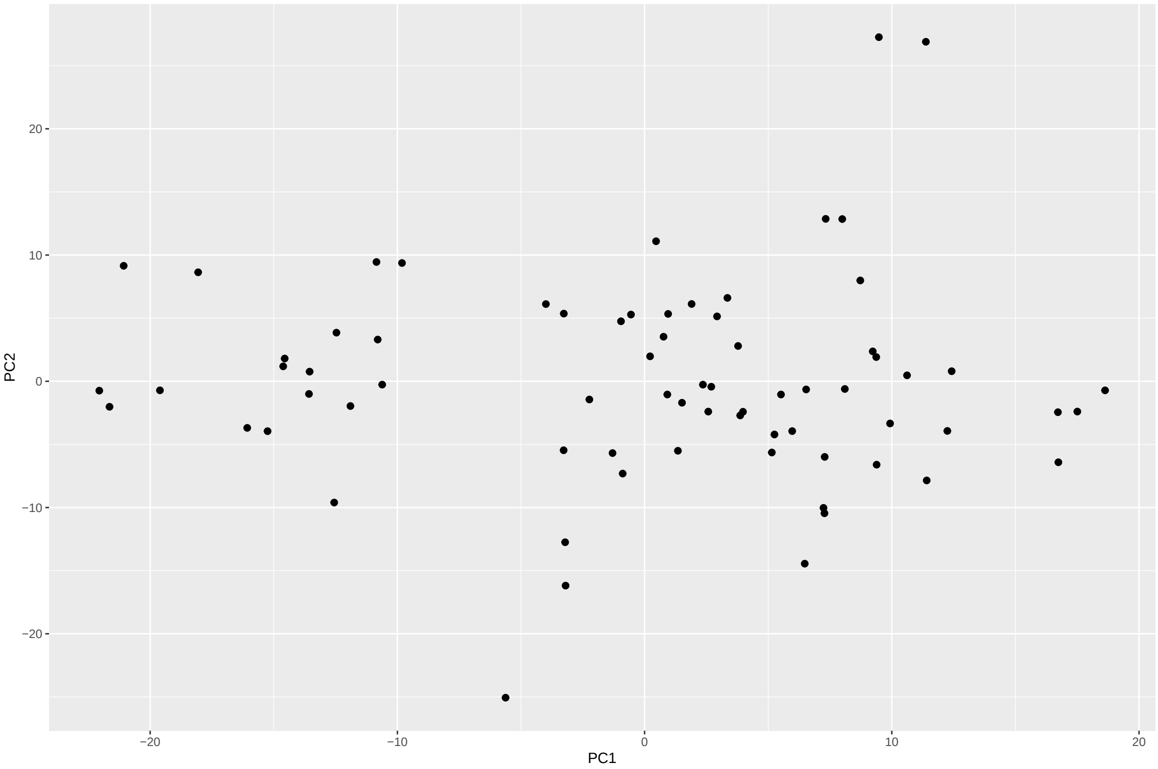
45 PC's are needed to explain 90% of the variance.

Sample PC-PC plotting:

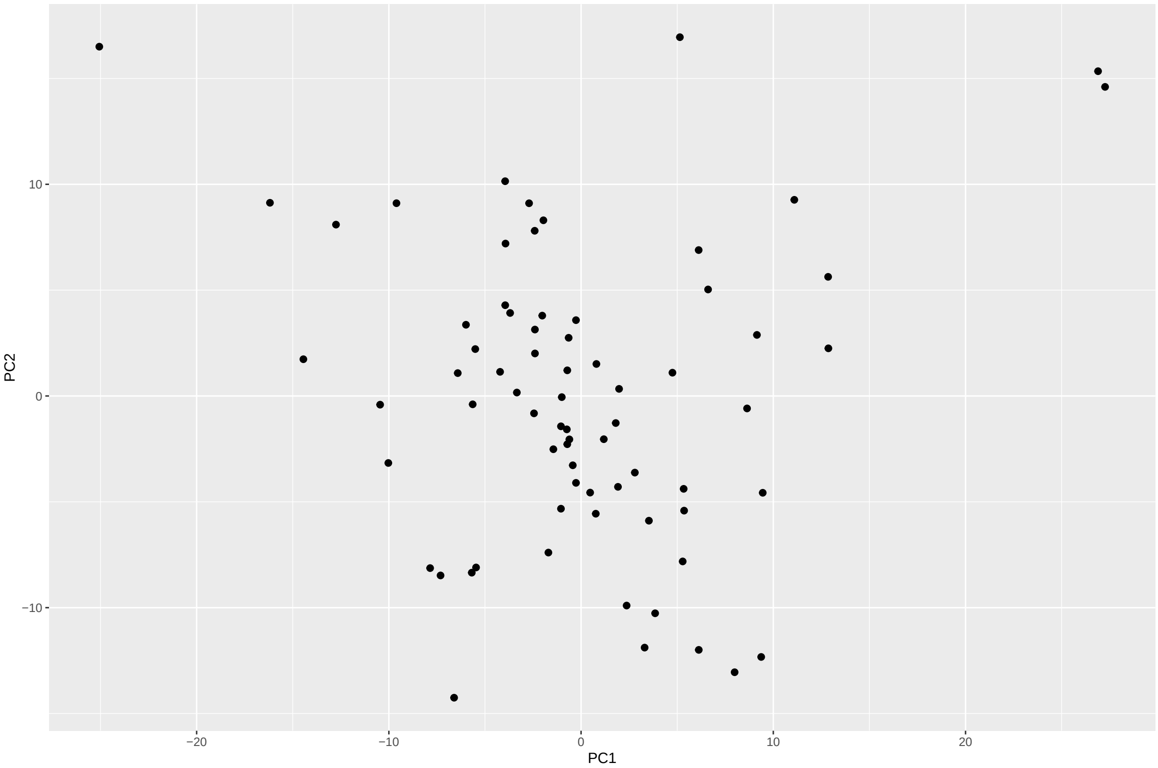
```
data.frame(PC1=mypca$pc[1,], PC2=mypca$pc[2,]) %>%
 ggplot() + geom_point(aes(x=PC1, y=PC2), size=2)
```



```
data.frame(PC1=mypca$pc[2,], PC2=mypca$pc[3,]) %>%
 ggplot() + geom_point(aes(x=PC1, y=PC2), size=2)
```



```
data.frame(PC1=mypca$pc[3,], PC2=mypca$pc[4,]) %>%
    ggplot() + geom_point(aes(x=PC1, y=PC2), size=2)
```



Plotting PC1 vs PC2 shows a parabolic relationship between the two PC's and two relatively undefined clusters.

Plotting PC2 vs PC3 shows a quasi linear relationship between the two PC's and two loose clusters.

Plotting PC3 vs PC4 we see that there is a single cluster around 0,0 with some outliers and no real defined relationship.

This means that the points that are close together correspond to observations that have similar scores on the components displayed in the plot, which is mainly determined by the RPKM values that are similar for genes that come from the same sample.

Samples that are close together are more closely related in the expression of RPKM for a gene than samples that are further away.

Session Information

Session information always included for reproducibility!

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.6.2  magrittr_1.5    tools_3.6.2     htmltools_0.4.0
##  [5] yaml_2.2.0      Rcpp_1.0.3      stringi_1.4.3   rmarkdown_2.0
##  [9] knitr_1.26      stringr_1.4.0   xfun_0.11       digest_0.6.23
## [13] rlang_0.4.2     evaluate_0.14
```