

R Boot Camp Problem Set

Digvijay Yadav

August 11, 2022

Establishing reliable biomarkers for assessing and validating clinical diagnosis at early stages of Parkinson's disease is crucial for developing therapies to slow or halt disease progression. This data set uses whole blood gene simulatedession profiling from over 500 individuals where we will attempt to find a gene signature. This repository contains the gene simulatedession profiles collected in the GENEPARK consortium. The main study sought a classifier for IPD. These data contain 233 healthy controls, 205 IPD patients, and 48 patients with other neurodegenerative diseases (NDD). Other samples are available in the data and can be used for additional analyses. The largest class of these additional samples are 22 samples from genetic unaffected controls and 41 genetic PD patients.

Note: the original study which uploaded this data to NIH Geo is not yet published.

Data Wrangling

Let's start by loading in our data sets. Download these from the canvas site, and make a new folder for R bootcamp. We'll switch to this directory here.

The tinyTex package will allow you to actually knit .pdf documents from RMarkdown:

Note that we have both a phenotype file, as well as a file which includes the normalized and log transformed simulatedession values. We can use the read.csv function to load in these files.

```
# load data in here
pheno = read.csv("parkPheno.csv")
simulated = read.csv('simulatedData.csv')
```

We should start by summarizing both these files. Try the following functions: head(), and View(). Note that while the dimensions on our phenotype file are reasonable, we have 552 columns in our simulatedession file. Just summarize the first 10 columns of this file.

```
# your code here
#View(pheno)
#View(simulated)
```

Try summarizing the phenotype data:

```
# your code here
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.1
##
## 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
```

```
summarize(pheno)
```

```
## data frame with 0 columns and 1 row
```

```
summarize(simulated)
```

```
## data frame with 0 columns and 1 row
```

We make the following observations.

1. We have some unnecessary data in this file. We aren't interested in the submission and last update date. We can reduce the dimensions of this file so it handles nicer from now on.
2. We have a LOT of missing data. You'll learn how to handle this in some of your biostats classes! For now, we'll run what analyses we can given the data we have.
3. Some of our scores have been read in as character values (and they should be numbers). If you investigate this further, you'll find that some values have been recorded as "ND", which we'll assume means "no data". We will need to record these as NA values in R.

Our next step is to address item one. We will reduce the dimensions of our pheno data frame to include only that information that we're interested in modelling. We can exclude the dates, type (as it's all RNA), tissue (all whole blood), organism (all homo sapiens), and subject ID (we will be using geo_accession as our unique indicator). As well, we will exclude mutated_pd_genes, as we intend to define our own gene signature later this week.

Subset your pheno data frame to include columns 1,8,9,11:20.

```
# your code here
pheno = pheno[,c(1,8,9,11:20)]
head(pheno)
```

```
##   geo_accession disease_label    sex age_at_exam age_at_symptoms updrs
## 1   GSM2631171  ATYPICAL_PD   Male         NA           53         1
## 2   GSM2631309  ATYPICAL_PD   Male         NA           64         0
## 3   GSM2631219  ATYPICAL_PD   Male         NA           NA         0
## 4   GSM2630775      CBD   Female         NA           60         0
## 5   GSM2631147      CBD   Female         NA           66         0
## 6   GSM2630853   CONTROL   Male         NA           41         0
##   updrs_ii updrs_iii_score_on updrs_iii_score_off updrs_iv hoehn_yahr_on
## 1         4                19                   0         0             2
## 2         0                 0                   0         0             9
## 3         0                 0                   0         0             0
## 4         0                 0                   0         0             9
## 5         0                30                   0         0             9
## 6         0                 1                   0         0             8
##   hoehn_yahr_off moca_score
## 1              0         21
## 2              0          0
## 3              0          0
## 4              0          0
## 5              0          0
## 6              0         30
```

Next we need to correct the columns which contain "ND". You can use the "which" function to find the index of the matrices which are "ND", and then set these to NA. Set columns 8,9,11,12,13 to numeric values

using the “as.numeric” function inside a “sapply” loop. Run a summary of the data frame again.

#your code here

```
ix = which(pheno$disease_label == "ND", arr.ind = T)
pheno[ix] = NA
```

```
j = c(8,9,11,12,13)
pheno[,j] = sapply(unlist(pheno[,j]), as.numeric)
summary(pheno)
```

```
##  geo_accession      disease_label      sex      age_at_exam
##  Length:550      Length:550      Length:550      Min.   :30.00
##  Class :character  Class :character  Class :character  1st Qu.:54.75
##  Mode  :character  Mode  :character  Mode  :character  Median :61.00
##                                     Mean  :60.56
##                                     3rd Qu.:68.25
##                                     Max.   :82.00
##                                     NA's   :266
##  age_at_symptoms    updrs      updrs_ii    updrs_iii_score_on
##  Min.   :10.00      Min.   : 0.000      Min.   : 0.000      Min.   : 0.0
##  1st Qu.:45.00      1st Qu.: 0.000      1st Qu.: 0.000      1st Qu.: 0.0
##  Median :55.00      Median : 0.000      Median : 0.000      Median : 0.5
##  Mean   :53.61      Mean   : 1.171      Mean   : 4.593      Mean   : 9.0
##  3rd Qu.:64.00      3rd Qu.: 2.000      3rd Qu.: 7.000      3rd Qu.:16.0
##  Max.   :78.00      Max.   :36.000      Max.   :35.000      Max.   :75.0
##  NA's   :325      NA's   :122      NA's   :123      NA's   :154
##  updrs_iii_score_off updrs_iv    hoehn_yahr_on  hoehn_yahr_off
##  Min.   : 0.000      Min.   : 0.000      Min.   :0.000      Min.   :0.0000
##  1st Qu.: 0.000      1st Qu.: 0.000      1st Qu.:0.000      1st Qu.:0.0000
##  Median : 0.000      Median : 0.000      Median :1.000      Median :0.0000
##  Mean   : 2.523      Mean   : 1.236      Mean   :2.487      Mean   :0.1961
##  3rd Qu.: 0.000      3rd Qu.: 1.000      3rd Qu.:3.000      3rd Qu.:0.0000
##  Max.   :64.000      Max.   :14.000      Max.   :9.000      Max.   :5.0000
##  NA's   :110      NA's   :118      NA's   :158      NA's   :109
##  moca_score
##  Min.   : 0.00
##  1st Qu.: 0.00
##  Median :26.00
##  Mean   :17.62
##  3rd Qu.:29.00
##  Max.   :30.00
##  NA's   :16
```

We have a LOT of missing values present in the data! As mentioned before, imputation of missing values is an entire field unto itself. While we won't be imputing data today, we are going to wrangle the above data to attempt to ameliorate some of these missing values.

To do this we will:

1. Combine our Age variables to be age_at_exam where known, but age_at_symptoms where that is observed without age at exam
2. Combine our updrs scores to be the average updrs
3. Combine our hoehn scores to be the average hoehn
4. Keep our moca score as is
5. Remove the old variables from our pheno dataset.

```

# your code here
#initializing AgeMaster with age_at_exam columns, and age_at_symptoms where age_at_exam values were NA.
pheno$AgeMaster = pheno$age_at_exam
pheno$AgeMaster[is.na(pheno$AgeMaster)] = pheno$age_at_symptoms[is.na(pheno$AgeMaster)]

updir_cols = c(6:10)
pheno$overallUPDRS = rowMeans(pheno[,updir_cols])

hoehn_cols = c(11,12)
pheno$overallHoehn = rowMeans(pheno[,hoehn_cols])

res = c(1,2,3,13,14,15,16)
pheno = pheno[,res]

#View(pheno)

```

As you can see, we have far fewer missing values to contend with!

Let's look at a summary of the first 10 columns of simulatedession data set.

```
summary(simulated[,1:10])
```

```

##           X           GeneName      GSM2631171      GSM2631309
## Min.      :    1  Length:20668  Min.      :-5.223788  Min.      :-6.09018
## 1st Qu.: 5168  Class :character 1st Qu.: -0.960423 1st Qu.: -0.92906
## Median :10334  Mode  :character Median :-0.004842 Median : 0.01385
## Mean      :10334                      Mean      :-0.009648 Mean      : 0.01249
## 3rd Qu.:15501                      3rd Qu.: 0.953228 3rd Qu.: 0.95912
## Max.      :20668                      Max.      : 5.766301 Max.      : 5.66627
## GSM2631219      GSM2630775      GSM2631147      GSM2630853
## Min.      :-6.39097  Min.      :-5.206869  Min.      :-5.27578  Min.      :-6.115736
## 1st Qu.: -0.97337  1st Qu.: -0.981831 1st Qu.: -0.96379 1st Qu.: -0.944666
## Median :-0.01097  Median : 0.001772 Median : 0.01906 Median :-0.007942
## Mean      :-0.00354  Mean      :-0.000010 Mean      : 0.00298 Mean      :-0.009892
## 3rd Qu.: 0.95324  3rd Qu.: 0.971013 3rd Qu.: 0.98545 3rd Qu.: 0.945826
## Max.      : 6.56118  Max.      : 5.275719 Max.      : 5.18612 Max.      : 5.570111
## GSM2630769      GSM2631196
## Min.      :-5.608142  Min.      :-6.303044
## 1st Qu.: -0.968002  1st Qu.: -0.970730
## Median :-0.001583  Median :-0.004689
## Mean      : 0.014813  Mean      :-0.006484
## 3rd Qu.: 0.987677  3rd Qu.: 0.977216
## Max.      : 5.591597  Max.      : 5.434250

```

We don't need the X1 variable - this is just remaining row labels in the csv file. Let's remove this variable.

```

#your code here
simulated = simulated[,-1]
#head(simulated)

```

We don't see any evidence of missing values in our summary, but we should check all of the columns (excluding the GeneName). You can check this with the "anyNA" function.

```

# your code here
anyNA(simulated)

```

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

Let's identify how big this problem is, and where it occurs.

```
# your code here
```

```
na_ix = which(is.na(simulated), arr.ind=T)  
head(simulated)
```

```
##   GeneName GSM2631171 GSM2631309 GSM2631219 GSM2630775 GSM2631147 GSM2630853  
## 1   A1BG -1.0366136 -0.27662929 -2.6292442  1.52266682  0.3358301  0.97332552  
## 2 A1BG-AS1 -0.8753413  0.54284119 -1.4575540  0.69927167  2.4918360 -1.30756851  
## 3   A1CF  1.2650380  0.07009968  0.7666068  0.02032404  2.5133798  1.82584774  
## 4   A2M -1.1735898 -1.61353987  0.3632498 -1.22731028  0.3997275  0.86599329  
## 5 A2M-AS1  1.2247663 -0.93452615  0.4948203  0.75509020  1.7236542 -0.22818120  
## 6   A2ML1 -0.4382954  0.51954673  0.2610932 -1.54324974  1.4411747  0.01398596  
##   GSM2630769 GSM2631196 GSM2631194 GSM2631197 GSM2631195 GSM2631198  
## 1  3.4533159  1.81379834 -1.9211147  0.7257764  0.6575387 -0.91583986  
## 2  1.5772826  4.00922478  0.4967702 -0.3360566 -1.5340809  1.05406203  
## 3  1.4629778  1.01145195  0.4692595 -0.3195250  1.2800251  0.05179562  
## 4  0.2364756 -0.87955380  0.4852329 -1.8378022  1.2444026  0.99429936  
## 5 -2.0413473  0.59855387  1.1834725  1.5956586 -1.8124371 -1.54911402  
## 6 -0.7479307 -0.07684364 -0.4987176 -0.1837047 -1.9271727  0.87524246  
##   GSM2631306 GSM2631162 GSM2631172 GSM2631241 GSM2631252 GSM2630927  
## 1 -0.84655288  1.1791279 -3.9196878  0.4744313 -2.0468679  1.64638684  
## 2  1.26922525  1.4960082 -0.1600686 -0.5493613 -0.6329077  0.76567160  
## 3 -0.61375284 -0.6789608 -0.6724768  1.0463665 -1.9405701  0.97085682  
## 4  0.03649272 -0.9076411  1.0203311 -0.7038708 -0.5943931 -0.01985207  
## 5  0.24188310 -0.9571613  0.7420052 -1.2601389  0.7727551 -1.60393326  
## 6 -0.43301581  1.9297295 -0.7885334  0.6856696 -1.6556987 -0.21708071  
##   GSM2630928 GSM2631227 GSM2631231 GSM2631235 GSM2631236 GSM2631238  
## 1  2.3494621 -1.28747722 -0.1375166 -2.64930848 -0.5211835  2.3866101  
## 2  0.8918579  2.20784667  0.2789515 -0.15288028 -1.0366904 -0.4633186  
## 3 -0.7185732  0.02302333 -2.0125364 -1.64575408  0.8203788  2.4663765  
## 4  2.3943697 -0.27391520  3.0961026 -0.37190510 -0.5522156  0.9172300  
## 5 -1.0772197 -0.61419330 -1.1996913  2.39657548 -1.6125899  0.1666942  
## 6  1.2236939  2.52062795  0.7489049 -0.06606603 -2.3478171 -0.6764611  
##   GSM2631239 GSM2631243 GSM2630771 GSM2630783 GSM2630830 GSM2630857  
## 1  1.4916053 -0.1782536 -1.659963 -0.7384035  0.9146473  1.38723172  
## 2  0.6665841  0.5192995 -1.277140  0.4070137 -1.4913926  0.31651129  
## 3 -2.3290718  0.8769891  0.296691  0.3015948  0.3285657 -0.32255513  
## 4  0.0466384 -1.4333307  1.122082 -1.4973222 -1.1639669  0.31919656  
## 5  0.1494168 -1.6651591 -1.222706 -1.7475125 -1.2604841 -1.59258362  
## 6 -1.4187163 -0.1474803  1.193786  2.7291049 -0.8678376 -0.07839134  
##   GSM2630868 GSM2630818 GSM2630907 GSM2630909 GSM2630916 GSM2630923  
## 1  2.0282589115  0.2867639  0.8765503  2.08729263  1.3751344 -1.0568484  
## 2 -0.1954580107  2.6901454  0.1853255 -1.37467254 -1.5710871 -2.8557396  
## 3 -2.2136408014  0.2940256  1.9085677 -2.50835849  0.7822219 -2.2953350  
## 4 -1.5002781106 -0.2019019  1.3212764 -1.74592908  0.0169678 -2.5227859  
## 5  1.3021516710 -3.1433825  0.3361751 -0.85923943 -1.2375982  0.1887502  
## 6 -0.0004901291  0.2447780  0.5887452 -0.03101765  0.3052142 -0.6726640  
##   GSM2630925 GSM2630929 GSM2630930 GSM2630932 GSM2631221 GSM2631230  
## 1 -0.2958283 -2.10300595 -0.04158138  1.2354445734  0.945052270 -1.9297252  
## 2  1.3469523 -1.82135528 -0.38894496  0.0071051808 -1.090033105  0.4064769  
## 3 -3.8422052  0.73112033  2.15930231  2.1235227231 -1.135663786 -0.2278403  
## 4  0.9195979 -0.33096684  0.62955475  1.2427952651  0.004803398 -0.4513778
```

```

## 5 1.2995907 -0.01940292 0.73045524 -1.4692743866 -1.795355641 1.0710448
## 6 1.1671110 0.48598974 2.22099957 0.0002136866 -1.077369034 0.1596775
## GSM2631232 GSM2631234 GSM2631237 GSM2631240 GSM2631242 GSM2631248
## 1 0.1310315 0.3029325 0.93292405 0.2450774 -2.2010466 0.07872818
## 2 -0.6413753 -1.8839945 0.97608066 0.4029095 -2.0829699 -1.47481732
## 3 1.0109173 0.1761590 -0.09362843 -0.2220153 0.6564502 0.58223808
## 4 -0.2036200 -1.4464613 -2.10500710 1.4502341 0.9879688 -0.05621746
## 5 0.1302830 -2.6410427 -0.15299668 0.0286268 -0.6600126 1.15699441
## 6 -1.0913582 1.0965872 0.01632357 -1.4416563 2.6812462 -1.04320546
## GSM2630899 GSM2630905 GSM2630906 GSM2630917 GSM2630922 GSM2630924
## 1 -0.5619655 0.8303993 -0.24152334 -0.04881529 2.4694273 -0.2634438
## 2 -0.9276068 1.3494852 -0.52145494 0.65227447 2.6459861 0.5821107
## 3 1.1728327 1.4226776 -1.05983269 0.52152354 2.5326991 -0.2391859
## 4 1.0841522 1.2791545 0.09251289 0.35721429 -0.6689961 -0.5378807
## 5 -1.0304385 0.1649847 2.18569789 -0.52206253 -4.1137056 -1.2505132
## 6 -0.5379009 0.4572370 1.08616722 -0.92926893 -0.9148150 1.5600256
## GSM2631298 GSM2631300 GSM2631301 GSM2631304 GSM2631305 GSM2631310 GSM2631312
## 1 -2.4041004 0.4142408 2.3051650 1.2032466 0.3102905 1.7362371 -1.4367006
## 2 1.2034354 0.6377265 1.4346190 2.6897943 -0.1815019 0.6370929 -0.4128298
## 3 0.1616847 0.9017189 2.4009772 -2.0181330 1.0952450 1.6810226 1.5093923
## 4 -0.6081584 -2.3985431 -0.1928422 2.1144050 0.4915416 -2.0634988 1.1022631
## 5 0.9835483 1.4162336 -3.5037736 1.2611128 -0.7937164 -0.7809311 2.0441711
## 6 1.2264029 0.7266947 3.9301177 0.5370959 1.5592591 1.1318930 -1.4997471
## GSM2631152 GSM2631153 GSM2631158 GSM2631159 GSM2631161 GSM2631163
## 1 -0.72592990 1.6656161 -1.2741775 0.5933994 2.69854110 -1.2529726
## 2 -2.29867241 -1.2727443 0.3678687 -1.2929641 1.80110090 -0.7875862
## 3 2.07877411 -1.0814380 -0.7687855 -0.1583923 -0.21089494 4.0526096
## 4 0.47757174 0.9726368 1.4798518 1.8658899 -2.30680656 -1.6684783
## 5 2.39637960 1.0536109 1.4657355 -1.0820489 -0.07070203 -1.0199878
## 6 -0.09733715 0.4041567 1.0818082 0.7551520 -0.35814807 -1.0583512
## GSM2631165 GSM2631167 GSM2631175 GSM2631176 GSM2631177 GSM2630777 GSM2630788
## 1 -0.1222497 -1.0789216 1.566529 0.9987111 0.1889327 -2.57214026 2.3263281
## 2 -0.5373867 0.3892825 -2.082986 -2.1671795 -1.2980895 -1.20746737 1.8176666
## 3 -1.1355584 -0.9998817 1.055984 0.8628208 0.6923748 0.05547596 1.9799421
## 4 0.3718242 -2.7818915 2.234714 1.0116093 -0.4044052 -0.25345266 0.1262320
## 5 0.2939756 0.5576378 0.880692 0.6894485 -1.7259526 -1.51679548 -0.8335691
## 6 -0.3863178 -1.5446292 -0.709811 2.0376934 -1.5553287 0.80176416 2.2422168
## GSM2631173 GSM2630908 GSM2630787 GSM2630790 GSM2630810 GSM2631138
## 1 1.1433088 0.33480168 1.8895206 1.3762785 -0.74705084 2.5770894
## 2 -1.0086858 -2.13654681 -0.3511912 -0.4478314 -1.29001456 -1.7157469
## 3 -1.4209240 2.58002782 -2.1834945 -2.3061270 -0.30136830 -0.8003415
## 4 0.4600274 1.13479164 -0.9740478 0.4491862 0.09954752 -1.4483073
## 5 0.3648058 -0.07249906 -0.2107581 0.2913374 -1.03057215 1.9680884
## 6 -1.3754587 -1.66898638 3.2593801 1.1504076 -1.75377378 1.5024215
## GSM2631028 GSM2631287 GSM2631100 GSM2631056 GSM2631118 GSM2631025
## 1 0.52698585 -0.4075548 2.315731 0.008797516 0.9740736 -0.14549094
## 2 2.20410107 -0.5065407 -2.364385 1.637490214 -1.3569627 0.84664460
## 3 0.78370739 0.6512651 -1.470522 -0.107786530 -0.8652369 0.01782888
## 4 0.01856317 0.8949406 1.818547 -0.857038646 1.0411629 0.23380696
## 5 -0.62248236 -0.8862523 1.413855 0.712979748 -0.5784515 0.79508014
## 6 1.47218901 1.9137140 1.573577 1.899656047 0.3787707 -1.34104537
## GSM2631281 GSM2631039 GSM2631057 GSM2631110 GSM2631010 GSM2631102 GSM2631103
## 1 -1.7351305 0.3808066 -1.5458882 -3.0564501 0.6290531 -0.59629003 -0.5823345
## 2 0.5947805 1.2028405 -0.6623765 1.5138483 3.3469296 -0.65688493 -0.5830482

```

```

## 3 -2.5323911 -0.4115594 0.4554766 -0.5420236 1.8493693 -1.18571877 1.6540798
## 4 -1.8155050 0.1217753 1.3740148 -0.1332096 -0.1180980 -0.02744244 1.2625902
## 5 -1.7003155 0.6344448 0.9252425 -1.0453155 3.8906284 -2.30940390 -0.1582876
## 6 1.7512150 0.8523116 -0.1901621 -0.1564972 0.5843132 0.76503528 0.7188814
## GSM2631017 GSM2631143 GSM2631024 GSM2631091 GSM2631275 GSM2631061
## 1 -0.87853228 -1.7813371 -1.9767790 -1.4422440 2.5379420 0.07145703
## 2 0.07439874 -0.4595939 -0.4162442 0.3061578 2.0518596 -1.06527536
## 3 -0.48347565 1.2879970 0.3310463 0.2365571 -0.1488802 1.13474823
## 4 -2.33379235 -0.1357224 0.3258514 1.2145831 -2.3105343 -1.51628874
## 5 0.28988425 -1.0916108 -0.8019365 -1.2614718 -0.7845362 -0.68313960
## 6 0.80314671 -1.4096304 1.8122266 1.1655381 -0.2559388 -0.34840860
## GSM2631273 GSM2631027 GSM2631101 GSM2631043 GSM2631125 GSM2630980 GSM2631031
## 1 1.6253900 -0.5252060 0.38358051 -1.779937 -1.7261420 0.7246159 0.6261896
## 2 1.0732556 -0.3886405 0.01148672 -1.152604 -0.8224051 -2.9806320 0.5489321
## 3 0.9797699 -2.5120573 0.76669521 -1.421151 -0.3183001 1.3330879 -2.1587893
## 4 -1.1977223 -0.6401546 -1.09335145 0.359461 1.3719563 -1.7138857 -1.6569751
## 5 1.2681438 0.4962858 1.62720997 1.076525 -1.0231948 0.5132847 -0.3278649
## 6 -0.5439811 -1.1232809 0.24317858 -1.515547 0.7247312 -0.5224958 1.3821682
## GSM2631130 GSM2631065 GSM2631105 GSM2631032 GSM2631082 GSM2631029
## 1 -0.29874524 -0.7003298 -0.0130223 -2.0146450 -0.005375429 -0.00657707
## 2 1.21215898 1.0153620 -0.1671471 0.0148141 0.617685172 0.59296794
## 3 3.82549650 0.2818832 2.8488830 1.4095834 0.876435907 0.62688123
## 4 0.91585804 2.8344216 1.8624700 0.2577295 2.735238068 1.11879834
## 5 0.45318630 0.7242130 -1.9804688 1.8094937 0.733913196 0.78404326
## 6 -0.07073392 -0.3480175 -0.4300965 0.7692640 2.007983514 -0.71612239
## GSM2631109 GSM2631005 GSM2630961 GSM2631280 GSM2631289 GSM2631131
## 1 -0.176132999 -1.2703050 -1.7383726 0.6542143 -0.4537444 1.9597551
## 2 -0.488867920 1.7564457 -1.0078234 -1.1817298 -0.3454855 -0.8111280
## 3 -0.008372826 0.1633467 -1.5899370 -2.2073443 0.4280094 -2.2969203
## 4 -0.487787580 1.3388416 0.3850796 -2.3433585 -1.1495464 0.4145710
## 5 -0.335284994 2.5598942 -1.7941523 0.4791979 -2.2468236 0.7133927
## 6 -1.017798100 -0.1120748 -0.1882884 0.6157505 -0.8401064 -1.7997261
## GSM2630990 GSM2631126 GSM2631047 GSM2631180 GSM2631049 GSM2631276
## 1 -0.006288985 -0.1486120 -1.6657266 0.5744428 -0.0414362 2.6849470
## 2 -0.812004639 -0.6701084 1.3709726 -2.6676896 1.2857053 0.6755781
## 3 0.287382440 0.2404943 0.4195041 0.8186180 1.8774089 -1.2489386
## 4 1.604406773 -0.9202510 -0.7035831 -1.1622770 -1.4313841 -1.6907005
## 5 -0.320995441 -0.2702325 -1.1184597 2.2104271 1.6951576 -1.6442383
## 6 0.478528813 -0.6732020 -0.6898093 0.6354484 2.2645528 -0.8226126
## GSM2631038 GSM2631041 GSM2631111 GSM2631059 GSM2631079 GSM2631012
## 1 0.5238011 1.42444199 -0.5133415 0.6018857 -0.59678180 -1.3119656
## 2 -0.3911746 -0.31383525 1.4802045 1.2191924 1.10767081 0.7133019
## 3 1.4987714 -0.50102608 1.0171982 3.2773973 0.80947622 1.2298903
## 4 0.9009478 -0.31815564 -0.2945238 -2.4237729 0.03434935 -0.5911419
## 5 -2.5905611 0.34238483 -0.7013252 1.8971469 0.52884767 -1.6105446
## 6 -0.8236771 -0.09696512 0.2438323 2.1391033 0.77902942 0.9199876
## GSM2631085 GSM2631077 GSM2630761 GSM2631022 GSM2631011 GSM2631136 GSM2630971
## 1 0.1445498 -0.7782432 -0.6621799 -1.4397310 -0.01079765 -1.3580874 0.6767844
## 2 0.9195574 -2.8538133 2.4034807 -0.2855208 -1.01541055 -0.5620236 -0.2247734
## 3 0.0142644 0.8202620 2.9316805 -0.9566524 2.79936551 1.0335741 -0.3574654
## 4 0.5817370 -1.2714134 1.4517983 0.2734970 0.63779439 0.8226159 -0.3449582
## 5 -2.7723251 1.5593833 -0.7030889 1.4760425 -0.69784088 -0.1664608 1.8601628
## 6 -0.5829344 0.8412565 -0.8365451 -0.4202488 -0.60715530 1.8582394 -0.9929007
## GSM2631279 GSM2631137 GSM2631185 GSM2631277 GSM2631072 GSM2631075

```

```

## 1 0.24822362 0.5958569 1.9747501 -1.3945018 1.211791840 -1.85793278
## 2 2.13204451 1.1550869 -0.8583587 0.4001038 1.647163140 0.40005541
## 3 0.18512102 0.1622205 1.9511865 -1.6268437 0.379795128 0.03401872
## 4 -0.07084703 2.0315725 0.8087325 -0.5744978 -0.004533444 0.23779417
## 5 -3.15956761 1.4847417 -1.4685318 1.1553210 -1.472630009 -1.46899056
## 6 -1.88047059 1.2446217 0.5825393 -0.7047452 0.128044702 -0.12248576
## GSM2631026 GSM2631064 GSM2631191 GSM2631084 GSM2631284 GSM2631042
## 1 -0.8569993 1.6575005 -3.69837788 -1.4998651 -1.7335238 -0.2508002
## 2 -2.6325459 0.9753336 -2.06325543 0.2677603 0.8322555 -0.1837639
## 3 -1.9535139 1.7832400 -2.52525359 -2.1364791 1.7778852 -0.1212560
## 4 1.8222826 2.5258067 -0.05292615 -2.7553230 1.0006457 -0.8630717
## 5 -2.6350817 -2.1917706 3.27001876 -1.3372618 1.9882784 1.6348581
## 6 1.6260681 -1.0279670 -0.55488699 -0.1030791 0.7962318 1.0903450
## GSM2631274 GSM2630774 GSM2630888 GSM2630910 GSM2631001 GSM2631015
## 1 -0.19297860 -2.04216993 -1.4737406 -0.49455264 0.4744007 -1.08237658
## 2 -0.09181115 -0.05562984 -1.0135349 -0.61010286 -0.8384263 -0.01633685
## 3 0.11774514 0.95298645 0.1331115 -0.08088916 -0.8476263 -2.05198372
## 4 -2.30145875 -1.35434587 1.2115518 0.61438292 1.3005096 -1.26011808
## 5 -0.22869657 0.08293144 -1.3751410 1.16766042 -1.3055796 -1.95834331
## 6 -1.23370193 -1.56724976 1.3902641 -0.60315833 1.8454994 -1.37849863
## GSM2630993 GSM2631046 GSM2631112 GSM2631142 GSM2630984 GSM2631020 GSM2631033
## 1 0.5954746 -0.2809995 -0.5750774 0.8847492 0.8753416 -2.4835415 2.15835028
## 2 0.2859204 -0.1074177 0.3376264 1.0563840 -1.3219969 -0.5013726 -0.03706037
## 3 0.5947390 -2.1211387 -0.4001051 -1.4520147 0.5813996 0.1745257 -2.02068438
## 4 -0.8725275 1.5127311 1.8170247 0.6928014 0.3711322 0.3469125 1.05804235
## 5 1.4441232 0.3009225 1.8997046 -3.2638935 -0.9294172 1.9492260 -0.41902617
## 6 2.1599959 -0.5815594 -0.9829718 -3.7769798 -1.6175987 -1.3215400 -2.53014029
## GSM2631083 GSM2631076 GSM2631068 GSM2631037 GSM2631272 GSM2631278 GSM2631123
## 1 0.8132632 -0.1665356 -0.4533609 0.7764306 0.2887474 -0.8968962 1.6098799
## 2 -1.5583816 -1.3671551 -0.1183107 -0.9382491 2.8599579 -1.7492974 2.8032541
## 3 -2.8634790 0.5530108 0.3409070 -2.0879133 -0.1957375 -0.4510495 -0.5826275
## 4 -2.0311809 -1.6229216 -0.6354456 -1.7475981 -0.9689664 3.0148110 -0.7770252
## 5 1.0803221 1.1208230 -1.4605397 0.2795217 -1.1435162 0.2920177 -0.9885841
## 6 -0.9779857 1.8359800 -1.4752632 -0.9769487 -0.5094334 -0.4377578 1.0975914
## GSM2631073 GSM2631135 GSM2630855 GSM2630758 GSM2630800 GSM2630864
## 1 2.30457342 -0.5139108 0.2882738 0.12803488 -2.78279194 -0.9955399
## 2 -0.03736602 0.5825417 -1.8838247 1.27382314 -2.15053713 -2.2294953
## 3 1.30731740 -0.4037840 0.5073351 0.68600903 -0.07151086 -1.1448278
## 4 -0.49201201 -1.4054259 -0.2945177 -0.13684929 -3.41616641 1.2438209
## 5 0.05869560 -1.4606700 -0.7883273 0.07768596 2.81996646 -1.6889511
## 6 -0.40957281 -1.9623860 -1.1034986 -0.46828826 -1.44753150 1.1372986
## GSM2630863 GSM2630865 GSM2630866 GSM2630897 GSM2630869 GSM2630789
## 1 3.57605310 1.060177243 0.9551789 -1.62706826 0.4851587 2.1630916
## 2 -1.31525382 -1.246625089 -2.0880019 -0.09900947 0.8682380 0.1142679
## 3 -1.64598292 -0.700052242 -2.0084536 0.22750715 0.8281284 -0.5814209
## 4 -1.19050468 0.004009321 1.0525064 -0.76345712 -1.4298399 1.5932022
## 5 0.19828946 1.226606156 -0.6484885 0.23352453 -1.8446979 -1.3383445
## 6 -0.03249254 -0.420138931 -0.9206416 1.00934982 1.4644204 -1.6721853
## GSM2630872 GSM2630877 GSM2630880 GSM2630840 GSM2630799 GSM2630778
## 1 -0.3540056 -1.0366153 0.2655742 0.52625706 0.5313232 -0.405133105
## 2 0.7582431 0.8604537 2.6260678 0.11438487 -1.4539887 1.807236561
## 3 0.8365683 2.3922543 -0.7327170 1.35788396 0.7477949 -0.296241812
## 4 0.3687222 2.3048555 2.4958387 0.31364276 -0.9460197 -0.008897561
## 5 -0.2289534 0.4649464 0.1510526 0.04284032 0.8720284 -4.312471140

```



```

## 6 1.2489190 0.7312674 -2.2775310 0.82643768 1.9918721 1.058603217
## GSM2630878 GSM2630876 GSM2630965 GSM2630958 GSM2631054 GSM2631121
## 1 2.0320443 0.63831337 -0.3234034 0.8241467 1.68996869 -0.2450795
## 2 1.6351966 -0.03472817 -0.1337309 2.1512599 0.07565683 1.1627951
## 3 0.1006738 -2.44585574 -2.1784096 -2.5838142 1.47548783 -1.0686879
## 4 -0.2912621 -1.09422536 1.8463706 0.9794458 1.53688690 1.2470483
## 5 -1.3173954 1.16179577 1.0485654 -2.0757138 1.21866061 0.8438397
## 6 2.8655005 0.08460314 -3.1627962 1.3825692 -0.71070139 1.3018340
## GSM2631044 GSM2631074 GSM2631080 GSM2631179 GSM2631181 GSM2631132
## 1 -0.451056152 -1.30504998 0.5574428 0.7280974 -0.4422642 -0.1472550
## 2 2.018871597 -0.55401732 -2.1754949 -0.9168467 0.6970140 1.5271219
## 3 -0.006553977 -0.07509473 -0.3930274 1.0986744 1.3671002 1.2633472
## 4 0.536701739 -0.04266681 0.9711842 0.4399270 -0.2012428 2.9859738
## 5 -1.623538792 0.39021641 -1.9522432 2.1377246 1.6730906 0.3253378
## 6 0.588906510 -0.62497032 -2.4683918 -0.8707716 0.1429637 0.6042389
## GSM2631122 GSM2631295 GSM2631040 GSM2631116 GSM2631283 GSM2630999
## 1 2.2862897 -0.2695502 0.08859814 1.1004308 4.47381607 2.0070914
## 2 0.1417431 -0.1646073 2.32379637 -0.9772755 0.06003625 0.8989509
## 3 -1.2295959 -0.5327321 0.35948290 2.6015253 0.55695504 0.7020283
## 4 3.0566889 0.5495787 -0.06080490 0.5116226 -0.13864699 -0.8161250
## 5 -0.5408809 2.0199070 -0.35020465 -0.3434844 0.01797183 0.8990312
## 6 -2.3906790 -2.2778674 0.40121134 0.9474281 -0.77325586 0.2240085
## GSM2631023 GSM2631058 GSM2631081 GSM2631140 GSM2631045 GSM2631016
## 1 -1.3544320 0.8696196 -0.09816091 2.1263195 0.98299969 1.8729932
## 2 -2.1147392 -0.1401676 0.43607676 1.7540595 0.48506653 -0.2512882
## 3 0.1890884 0.7795670 1.97126660 -0.8039588 0.73441052 -2.0449685
## 4 1.9350101 0.4864215 1.23995016 -0.9083851 -0.08056894 0.1355115
## 5 0.8669565 1.4149110 -1.01651194 -0.3111239 3.77600749 0.6820395
## 6 -1.6033564 -0.3402092 0.20084008 -0.9980821 1.04015824 -0.5283904
## GSM2631070 GSM2631133 GSM2631055 GSM2630974 GSM2630812 GSM2630780
## 1 -0.08199385 -0.3959699 -0.01781219 1.726817 -0.1113749 1.81127698
## 2 -0.07732189 -0.9702949 1.18010654 2.182153 1.8608078 4.17293934
## 3 -1.05673436 0.6822056 0.45094070 1.361244 -2.8560560 1.10902799
## 4 -0.75907847 0.5982004 1.76425529 2.015668 -2.9716736 -0.35828302
## 5 1.18643542 -0.4006234 0.53460195 -1.606576 3.7462551 -0.05346301
## 6 -1.44993459 -1.6008807 -0.54258838 0.849591 -2.9097115 1.02921213
## GSM2630816 GSM2630871 GSM2630873 GSM2630874 GSM2630881 GSM2630896
## 1 0.71098552 1.6521639 1.3091015 0.4122760 0.8336254 -0.21484673
## 2 1.16817749 -3.4048560 2.2369101 -0.5453645 -1.4738963 -0.07202177
## 3 0.81561138 -1.3480853 2.9911421 2.6297330 1.5677235 -0.81658164
## 4 -1.34676226 -0.3457633 0.7505215 3.0615665 0.6154556 2.16652548
## 5 1.60992368 -1.0069409 0.2929605 1.7252094 0.6434004 1.35576818
## 6 -0.09619389 -1.4236535 0.3429904 0.1482518 -1.9566537 1.73713650
## GSM2630913 GSM2630784 GSM2630843 GSM2630884 GSM2631149 GSM2630882
## 1 0.08060693 0.180867 -2.26753707 -0.34923227 -0.4374839 -0.4566960
## 2 0.94214563 -1.292060 -1.15741320 -0.95510201 -2.3555882 1.3368513
## 3 0.78668032 -1.900018 0.79750461 2.04656810 -0.8976752 -0.0147355
## 4 0.79246096 -1.458918 0.34546316 0.02100691 -0.3113275 -0.5359418
## 5 0.94357695 -4.203185 0.06455922 -0.27422236 0.8035920 2.9860535
## 6 0.84446387 0.757127 -1.96283598 1.38895346 0.5050591 2.3269982
## GSM2630883 GSM2630885 GSM2630912 GSM2630861 GSM2630879 GSM2630785 GSM2630803
## 1 -0.07651831 -2.1652342 2.4036434 1.8453570 0.9829156 0.6972424 0.7615698
## 2 -0.16429595 -0.8379603 1.5215375 -1.5221091 -1.6454411 0.8961529 0.2047758
## 3 -1.90264602 -2.5563495 -1.2647831 -0.3876221 0.5339187 1.0152980 -0.5032352

```

```

## 4 -1.10750709 0.2068966 -2.9413087 -1.2214525 0.3450632 1.0765632 1.0693646
## 5 -0.14927114 0.9853385 0.4318646 0.4930612 -0.8671823 -1.5619163 -1.5424990
## 6 0.41396481 -0.9686804 -1.6607134 1.4533682 0.2952022 -1.3979587 -1.9343446
## GSM2630779 GSM2630772 GSM2630875 GSM2630767 GSM2631145 GSM2631303 GSM2631308
## 1 1.5594929 2.4013352 -2.5514166 -0.8397405 -1.49990442 1.2407076 0.1824498
## 2 0.1312732 -1.8095569 -0.4964831 -1.2996112 0.79665186 2.4733899 3.1694323
## 3 0.6654876 -0.7076195 -1.8275705 0.5020707 -0.14092144 -0.6263343 -0.2438267
## 4 0.1203579 3.9675519 -1.4765361 -0.8154002 1.10521974 0.3367187 -2.8420954
## 5 0.1164749 -0.8403366 2.9739504 1.3214948 -0.07615102 1.6347751 -0.2398015
## 6 0.7127972 -1.0539378 0.8064198 1.1658008 -0.55747072 -0.1625275 1.8033302
## GSM2631311 GSM2630832 GSM2631302 GSM2631307 GSM2630919 GSM2631222 GSM2630933
## 1 1.0414307 0.40214818 3.6658353 -0.1802106 1.3483058 0.2852920 -0.7267964
## 2 1.4829696 1.02828090 -1.8275640 -4.0199292 -1.1262076 0.8343550 -0.9410521
## 3 1.6152926 2.14601498 -1.2782202 -0.1271007 1.2577649 -1.0899355 -1.0008434
## 4 3.2142075 -1.34488765 -0.6029543 2.4441042 -0.1307031 1.2334014 -0.4741005
## 5 -0.7246055 0.04782523 -0.9618192 1.5020218 0.6545610 0.8891888 -0.5590053
## 6 -1.1839583 1.06881224 0.2667522 -2.3050172 2.8131676 -0.3885623 -0.6485836
## GSM2630934 GSM2630935 GSM2631250 GSM2630936 GSM2630937 GSM2630926
## 1 -3.5444279 0.7237088 -0.8615052 -1.67618729 -2.5704700 -0.9075943
## 2 -1.6429975 1.4570222 1.0222164 1.17271894 -0.6766507 -2.1637730
## 3 -1.0892822 -0.6486775 0.7668590 -0.82879079 1.1143742 -2.3096814
## 4 -1.3183899 1.0976942 1.9611691 0.15315115 -0.6401510 0.4041687
## 5 0.8874933 -2.7470495 0.4819399 -0.05950031 -1.1835529 0.1875655
## 6 -1.6534896 0.2976375 1.9675371 -0.31441317 1.2102524 -1.1843326
## GSM2630833 GSM2630920 GSM2630844 GSM2630834 GSM2630847 GSM2631217
## 1 -0.92213376 -0.41709417 -2.01974778 -1.1497743 0.3593286 3.4303709
## 2 -1.95378993 1.77239602 -0.05392453 -1.3315041 3.7884726 0.8790657
## 3 0.05453025 0.03080373 -0.88912580 -0.7683926 0.4330502 1.3050043
## 4 1.30107069 -0.73035170 1.44103776 -0.9487000 1.2092206 0.7558618
## 5 3.48231285 -0.58962376 1.07427237 2.0443349 -0.6571164 0.9862983
## 6 1.13091121 0.91935818 -3.38999739 -1.1597187 2.2140089 -0.6259500
## GSM2630848 GSM2630798 GSM2630849 GSM2631229 GSM2631226 GSM2631202 GSM2631062
## 1 -0.6377118 1.7676495 -0.5015730 -1.191422 -0.76236318 -1.2111075 -1.1480953
## 2 0.2527629 -2.2220643 -1.0391876 -1.230290 0.10256914 0.9219089 1.3269312
## 3 0.8412057 -2.3800666 -0.6557411 -3.123735 -0.48895418 -1.0816982 -0.9552616
## 4 1.3117698 0.9058562 -1.5988363 0.811381 -0.53590471 1.2965380 2.2673514
## 5 0.2385470 0.1314863 -0.7849668 1.605401 0.55243817 -0.5087077 -0.0512388
## 6 0.6816004 -0.7288785 1.0454908 -2.142498 -0.03667886 -0.7448849 0.2889521
## GSM2631206 GSM2631205 GSM2631201 GSM2631207 GSM2631193 GSM2630955 GSM2631203
## 1 -1.1957774 0.7416789 -1.3602342 -1.2033843 -1.1591597 -1.6338993 -1.2900632
## 2 -2.5780624 1.6053224 0.4777948 0.1404821 -3.2700809 -0.6436568 -1.3778382
## 3 -0.3753229 -0.3674893 0.4606674 -0.3981390 -0.4572879 -0.3462053 -0.5100098
## 4 0.8976925 -0.6679614 -2.1155933 1.7919720 0.3948714 -1.2635245 -2.7358849
## 5 -1.6226672 2.2060799 -1.9641189 -0.7453575 -0.3718707 -0.8374466 -0.4228606
## 6 1.0211205 -2.0289390 0.4097881 2.2560777 -3.4880238 1.8811388 1.4605595
## GSM2630811 GSM2631282 GSM2631063 GSM2631141 GSM2630946 GSM2631294
## 1 -0.8571240 -1.232013640 -0.6049448 -0.3876590 -0.4828608 0.3324155
## 2 2.2085579 0.725556529 0.1797668 0.5050713 -0.5459387 -0.5667721
## 3 -0.1226983 0.316187786 -1.2629183 -0.8387031 -3.2483731 -1.0092337
## 4 3.7153521 -0.170180663 -1.3881362 -2.3104037 0.5271802 0.1267025
## 5 -0.6905953 0.005194436 0.8770569 1.8385740 -0.9385243 -3.1400500
## 6 0.9187864 0.514823762 0.6171483 -0.3351933 -2.6466666 0.6153448
## GSM2630845 GSM2631255 GSM2630809 GSM2631018 GSM2630987 GSM2630851 GSM2630996
## 1 -0.4391544 -1.0106745 -1.1083826 -2.6875508 -0.9173226 -0.76764428 -0.8589442

```

```

## 2 -3.0193836 -1.6861420 0.7802439 -4.4755934 0.9281720 0.31830953 2.4942995
## 3 0.5970295 -0.7813371 3.0992954 -0.9053522 -0.8739992 -0.90370136 1.8898746
## 4 -0.4870853 -1.3489689 -3.7320674 1.9349311 -1.2551727 -0.25057233 -0.8521667
## 5 -2.2835696 -0.2769043 0.8979115 -1.9681822 1.7660997 1.49878465 -3.2712217
## 6 1.5374995 1.0581987 0.4574137 0.9486238 -0.3122088 0.06904076 -0.3422267
## GSM2630796 GSM2631292 GSM2630887 GSM2630819 GSM2630850 GSM2631069
## 1 -0.4700442 -2.52265188 -1.8783165 0.93646900 -1.32825436 -0.9589495
## 2 1.1152344 0.02320227 -1.6913829 -0.06709222 -0.23105871 -0.4504241
## 3 -1.2218764 -0.93966881 -1.6289648 -0.78351058 0.03551342 -1.2130022
## 4 0.7468373 -1.30931469 -0.2513103 0.40717358 -1.55006601 0.6484371
## 5 -0.7110857 0.16428379 0.6202517 3.25109367 -0.26906879 0.9651497
## 6 0.6492021 0.03724693 0.5131735 -1.31520492 -0.33255513 -2.0283444
## GSM2630820 GSM2631251 GSM2630782 GSM2630858 GSM2630786 GSM2630846 GSM2631218
## 1 -1.8959157 0.2539446 -1.7701014 -1.4818171 -1.0376929 -2.5619360 0.08055711
## 2 1.7066288 -0.4483734 1.9770960 0.6629819 -0.3294843 -0.1178808 -1.09146696
## 3 -0.4437509 -0.9278382 0.3664970 -2.4278685 0.6942003 -0.4026135 -0.16974664
## 4 0.9282020 -1.3645759 -0.2582078 -0.6430071 0.2017015 1.7539388 0.73544571
## 5 -1.1668979 1.2727199 1.8558053 -1.2931958 0.5405148 1.0531320 1.06966646
## 6 -0.9555321 -0.1518477 -0.4512699 -1.9903513 -0.7549119 1.5572617 2.67699011
## GSM2630797 GSM2631228 GSM2630860 GSM2630859 GSM2630852 GSM2631088 GSM2631087
## 1 -1.2181568 -1.2855767 0.92995150 -1.2365196 -0.9722010 -2.0256017 -1.0897619
## 2 -0.7596254 -0.9498746 0.01636692 1.9682346 -0.5961916 -1.0113355 -1.8086644
## 3 1.5056399 -0.8748691 2.66524010 0.9106952 -0.9686264 0.8700383 1.5396297
## 4 -0.2335612 0.5153979 -0.50887611 2.6749423 0.7417117 -1.3553926 1.5669716
## 5 -2.5806811 1.7096362 0.19639006 -1.7216788 -2.8700579 -0.3126360 0.2430791
## 6 0.9556707 -0.2197358 0.75427414 2.0605258 -1.0293989 1.2667391 1.4940747
## GSM2631139 GSM2630952 GSM2631285 GSM2630953 GSM2631086 GSM2631270 GSM2630962
## 1 2.9288311 -0.7742563 1.2388876 -1.2170413 1.3385194 0.6549759 1.0767303
## 2 0.9589744 2.2363733 -2.3101121 1.5750327 -0.4707110 -0.6389563 1.8540541
## 3 -0.2020964 -0.8730254 0.2259918 1.7408934 -2.0846519 1.5631676 -0.1804550
## 4 -1.1731534 0.5171663 -1.9527740 -0.5002877 -1.0827927 0.1556494 1.3622466
## 5 0.4166722 1.0571114 0.2207533 2.6860543 3.0179445 -0.8344649 0.3120049
## 6 -0.7964007 0.2629671 -1.1877441 0.5021782 0.6324775 1.5852673 0.7140946
## GSM2630963 GSM2630954 GSM2631000 GSM2631034 GSM2631288 GSM2631092
## 1 0.9866417 -1.5649693 0.7974478 2.02374607 -0.1909183 -0.7562428
## 2 -0.2372467 0.4551308 -2.4641422 -0.50471745 0.4733015 -0.1420464
## 3 0.2700906 -0.9205626 -2.1934623 0.08449291 -1.1618634 -0.6817765
## 4 0.6638912 -1.0448715 -2.0489023 -1.34335866 -0.2717609 -2.0272399
## 5 0.8982164 2.8042585 2.0540807 -4.11715053 0.6947429 -0.4162787
## 6 -1.0398024 -1.4181667 0.5853475 -0.25083227 -3.2123897 -2.3554965
## GSM2631286 GSM2631113 GSM2631098 GSM2631093 GSM2630768 GSM2630989
## 1 -1.08568312 0.2724468 3.7114051 0.4984759 -0.8386057 -0.74064328
## 2 -0.02988436 1.5154031 -1.3837900 0.7246344 1.0325210 -0.07185176
## 3 0.12082333 -1.6965196 4.0709246 1.2591368 -2.2517170 0.83825625
## 4 0.58136244 -1.7869708 1.7508827 -1.7100305 1.3080645 0.74248755
## 5 -1.69977363 4.2856709 1.6404151 -5.1151786 -0.1930789 -1.18802939
## 6 1.02770076 1.7117783 0.6760802 -1.6974102 2.3210766 -0.39852916
## GSM2631127 GSM2631258 GSM2630802 GSM2630770 GSM2630854 GSM2630801 GSM2630983
## 1 -1.5495944 -1.692934 -2.84909581 -2.3715391 0.3339154 -0.7573349 -1.4593277
## 2 -0.3880660 0.816672 0.03749752 -1.5025354 -1.0418032 -0.6470000 1.9304994
## 3 0.7654619 -0.530256 2.63938842 -0.7979930 1.1283544 -1.3002454 1.8970389
## 4 -1.7384800 -1.413566 -0.42840880 -2.0496676 2.3717067 -1.2088167 -0.8437321
## 5 3.1044464 -1.332465 -0.19948102 0.2150763 0.3693599 0.4762589 -2.7803432
## 6 -0.4084369 -1.163991 -0.50482302 0.1794318 -0.8588559 -0.8153944 -0.1315860

```

```

##      GSM2630914 GSM2630997 GSM2631188 GSM2630964 GSM2630945 GSM2631119 GSM2631124
## 1  0.25660485  0.9952940 -1.3469058 -2.6288359 -0.6004389 -3.6448690  0.4604806
## 2  1.58544511  0.5375188  0.4849139  0.1851821 -0.8126650  1.5784185 -0.4324193
## 3 -0.81259552 -0.5304101  1.4112754 -2.7503869  0.2768437 -0.3669341 -0.7770545
## 4  1.21889363  0.6393521  0.5247390 -1.0666782  0.4229489  1.6791770 -1.9217731
## 5  1.42478006 -0.2924676 -0.0393988  0.1494384  0.7503645 -1.2916385 -2.6804973
## 6 -0.07425011  0.4636789 -1.3464223 -2.3465660 -0.4503549  1.6227011  0.6490980
##      GSM2630806 GSM2630921 GSM2630972 GSM2630807  GSM2631108 GSM2630988 GSM2630992
## 1 -1.7610496  -1.992002 -1.5775681 -0.2813225 -2.03602760 -1.2907785 -0.4253443
## 2  1.3646026  -1.099208 -1.0487502 -0.3115299 -2.32265723  2.6255155 -0.3318395
## 3 -1.8486124  -3.528570  2.1013188  1.3446661  0.03279319 -0.2083101 -0.1840665
## 4  0.1547986  -1.376280 -0.3023897  2.5924390  0.44802348 -1.4146712 -0.1612782
## 5  0.2857544  2.531673  0.3394547 -0.3960770 -1.85661345 -1.1728334 -0.7997469
## 6 -2.1824287  -1.309204 -0.8410377  1.0544945 -0.20773138  0.2764553 -2.3564241
##      GSM2631263 GSM2630823  GSM2631002 GSM2630949  GSM2631254  GSM2631050
## 1  -0.510393 -1.2246214 -1.40324525 -2.4158496 -2.2235014 -0.76460953
## 2  -2.085572  1.6518916  0.08143373  0.8454936  0.3612837  2.36792865
## 3  -1.464422  0.6981699 -2.73991585 -0.7195946 -0.8606767  1.59327432
## 4  2.168009 -0.7452748 -1.07063151 -1.4730156 -1.2451833 -1.95005075
## 5  -0.491203 -1.0116534  1.56540344  0.3879424 -0.6131017  0.09477680
## 6  2.074802  2.1189323  0.80792017  0.1068090  0.8428611  0.09846694
##      GSM2631052 GSM2630835  GSM2631290 GSM2630994  GSM2630975  GSM2631021  GSM2630825
## 1 -2.3685322 -0.8036398  0.1382104 -1.0755093 -1.15882104 -0.7863459 -4.2561487
## 2  3.0024018  0.6702478 -1.7558121 -1.2562552  1.52141854  1.0846899  1.9321547
## 3  1.1855898  0.2164077 -1.9180887 -1.2159128 -0.69377644  1.6081912 -2.7058179
## 4  0.5437075 -1.3103535 -3.1066575 -0.8986997  0.01229438 -0.2684751 -2.2043384
## 5 -0.3110372  1.2135375  0.9965829 -1.5967986  1.97630073  1.8143218 -0.8557193
## 6  0.8761223 -1.9657801 -1.8767866 -0.8367942  0.45847003  0.8578002 -0.1522699
##      GSM2630821 GSM2631297  GSM2630978 GSM2631114  GSM2631117  GSM2631260  GSM2631200
## 1 -0.1622274 -1.1443590 -0.7532958 -1.5370875  1.82655665  0.7218192 -0.6612711
## 2 -0.7747135 -0.1513008  0.7612781 -0.3381853  0.42434275  0.2783784 -0.1561070
## 3  3.5773099 -0.8792346 -3.5246298  0.5523385  2.10679075  0.4791960  1.1697372
## 4  0.8759624  0.1899872  0.6405843  1.4726698  0.68538181  0.2121191  1.2254250
## 5 -0.2486916  1.8689987  0.4489273 -4.0054246  0.01183867  0.5595167  0.8606279
## 6  2.0310790 -0.2015794 -0.8389120  1.3811014 -1.51268614 -0.1939641 -0.2580617
##      GSM2631096  GSM2631257  GSM2631003 GSM2631190  GSM2630985  GSM2631120
## 1 -2.4212539 -1.06302197 -1.2130250 -0.4617777 -1.0117914  0.007785086
## 2  1.5147531 -0.33091720  0.3136001  0.7050810  0.1423768  2.376800984
## 3  1.9645479  1.29755515 -1.6743548 -0.5661822  0.4317530  2.010469875
## 4  0.3114422  2.19449918 -0.8528134  0.3399771 -0.4818381  0.014261062
## 5  0.4907612 -1.27699379  1.7048295 -0.3928432  0.8290455 -2.269813753
## 6  1.5377082 -0.06833721  1.8263866  2.3763421 -4.4379861 -0.096749201
##      GSM2631030 GSM2631095  GSM2631256  GSM2630837  GSM2630940  GSM2630805
## 1 -1.5352028 -2.1795588 -0.6929879 -3.21701314 -1.8192779 -1.30369789
## 2 -2.3326461  1.3236388 -0.4937494  0.68376772  0.9010151  2.20391920
## 3  1.6185871 -0.8923174  1.5692905 -0.65985416 -2.3261075  0.06090074
## 4 -1.5010816  1.0186319 -1.1515244  0.35782589 -2.2142404 -1.21951463
## 5 -0.6337451 -0.9952099  2.0115955 -0.03992841  1.6019707 -1.82031947
## 6  0.2654832  1.4005273 -2.3159228 -1.57183188 -1.5297614  0.99725330
##      GSM2631253  GSM2630986  GSM2631009 GSM2630944  GSM2631225  GSM2631014  GSM2631071
## 1 -0.2890620 -0.19193068 -1.2311383 -2.392967  -1.768182 -0.4820649 -1.3289695
## 2 -0.4013985  0.76874921  0.7224752 -2.235160  -1.691606 -1.3247027  1.9265923
## 3  3.1137601  0.03180602 -0.3864629  1.480967  -1.086509 -0.2184726  0.5128313
## 4  1.2743420  1.17901815 -0.2717955 -1.087226  -3.424833 -1.9517511 -1.5062462

```

```

## 5 -1.8709296 -1.11761266 -0.7103786 2.908685 1.119041 3.6344712 -0.1916060
## 6 2.4145872 -0.33861885 1.9648486 -2.138953 -1.629056 -1.9914970 -1.7496990
## GSM2630842 GSM2630795 GSM2630773 GSM2630759 GSM2630947 GSM2630892
## 1 0.1308747 -1.5520803 -1.3866572 -0.06445793 -0.8860218 -0.5668457
## 2 -1.1271187 1.8546198 -1.4207905 1.71342996 -0.8398652 -0.3581936
## 3 0.4113337 3.2900138 0.3106350 -1.64283867 0.3420142 -0.5601045
## 4 -1.8495338 -0.3747986 2.6810301 1.10387333 -0.3624510 0.4016261
## 5 0.1597748 -2.9652716 -1.4968257 1.39734056 2.3799516 -1.2756506
## 6 0.2731774 1.0885919 0.2276535 0.84098376 -1.5762497 0.7889591
## GSM2630973 GSM2630828 GSM2631192 GSM2631144 GSM2630822 GSM2630998
## 1 -2.45748013 0.2550459 -1.7929279 0.1347951 -1.72714698 -0.4968247
## 2 -0.06077072 -1.1702743 0.3868574 -0.2615195 -1.12221062 0.4073223
## 3 -1.59082985 0.7738476 0.2946796 1.6032987 0.47635309 1.7139224
## 4 -1.34028180 -1.2094223 1.2500812 -3.3455574 0.05965252 0.9890002
## 5 3.65585758 0.6028805 -1.0393715 0.2622450 0.24821305 -1.7079292
## 6 1.30910763 0.7224150 1.5673340 1.0004102 0.07108647 0.1223179
## GSM2631134 GSM2631183 GSM2630969 GSM2631296 GSM2631053 GSM2630941 GSM2630982
## 1 -0.3068574 -1.6197081 -1.8650792 -0.8094853 -0.2752381 -0.4932378 -1.31670144
## 2 1.3451030 -2.2579868 0.7256644 1.3620749 -3.2233097 -0.1106730 0.92723336
## 3 0.7512489 0.9713404 0.1347048 1.6651827 -0.0859006 -0.6275822 -0.04111648
## 4 0.5787729 0.5735459 -0.5158575 -0.7671034 2.5859527 -1.8001465 -0.92814344
## 5 0.5394397 2.8804621 0.2687693 1.8066045 2.4093135 1.6035381 1.68328145
## 6 -0.3179972 0.3105416 1.3394577 1.5870281 1.9586991 -0.5295211 -1.14438473
## GSM2631271 GSM2630956 GSM2631008 GSM2630951 GSM2630957 GSM2631006
## 1 -1.76084037 -0.69439097 -0.7809703 -1.5660034 -0.8136909 -1.3471725
## 2 -0.94077657 -1.48994087 -0.7464725 0.5604905 0.6845381 -1.1838408
## 3 -0.28826751 -0.11495293 -1.1190199 1.8939693 -1.1075196 -0.3041053
## 4 -1.33804070 0.02946498 -0.3750879 2.3603219 -0.8275910 -0.1254125
## 5 0.09384828 -0.26772644 -0.5678378 1.5751923 1.0989572 0.2713802
## 6 -1.01532791 -0.21394266 0.6173899 1.8608349 0.2065982 0.6213892
## GSM2630959 GSM2630948 GSM2631148 GSM2631007 GSM2630939 GSM2630970
## 1 -1.5177243 0.2052843 -1.4534225 -0.2114351 -2.065763436 -1.8510976
## 2 1.4696764 0.6488713 1.5320964 -2.0374561 -0.022157147 0.9694143
## 3 -2.5127760 0.7456361 -0.1806734 0.8862420 -1.265041451 -2.0604068
## 4 -1.1569297 -2.5560028 -1.2356528 1.9492697 -0.003278473 1.0086813
## 5 -1.4449753 0.2492546 0.3432307 -0.5286115 -1.035860784 -0.5988114
## 6 -0.8437152 0.6846856 -0.9624983 1.5528926 0.872332576 -1.3708169
## GSM2631066 GSM2630967 GSM2631291 GSM2630981 GSM2631115 GSM2630824 GSM2631067
## 1 -1.2968154 0.22153774 -1.7380806 -0.5934177 -0.9629834 -0.1138853 -1.5898242
## 2 1.0604712 0.31653120 -0.3238792 -0.2477040 0.4033683 -2.1581080 0.0717733
## 3 0.9954290 -0.44931668 0.8205933 0.3421513 -0.8268833 1.1674422 -1.3245795
## 4 0.5340759 -0.08455345 2.4248516 -1.3343908 -0.4308926 -1.9239171 1.5104257
## 5 0.9956399 0.72605717 1.0006117 -2.7020389 0.4208248 1.6265413 -1.2216904
## 6 -1.6604372 0.08442453 -0.2111012 0.1744080 -2.1694413 -1.5639067 1.6116437
## GSM2631107 GSM2631097 GSM2631036 GSM2631186 GSM2630979 GSM2631262
## 1 -0.9739718 -0.7179147 -0.925369635 -0.9818926 -1.745825 -0.69442132
## 2 -0.7224086 -3.0883940 -2.106625909 -1.4326404 -3.084894 -0.25848504
## 3 -1.6356987 -1.5128747 -0.692826832 0.9829072 -2.033306 0.08873056
## 4 -0.1552184 0.8863757 0.005652508 -0.2714030 1.735795 0.98611479
## 5 1.7156022 -0.2537189 -0.385480363 -0.6298005 -2.048085 -1.15704137
## 6 -0.6954149 1.8186869 -0.960849676 -0.5225415 -1.519835 0.55662102
## GSM2630960 GSM2630942 GSM2630867 GSM2631094 GSM2631266 GSM2631293
## 1 -1.0501747 -2.5952102 -0.05407524 -1.51197572 -0.8711843 -0.8591307
## 2 -0.2877108 -0.1810444 0.34667709 0.06280491 -1.7319095 -1.1627578

```

```

## 3 2.8949967 3.1345713 1.19697026 1.80316659 0.2212331 -0.4381697
## 4 0.1336727 -0.2269786 0.68194648 -0.16581476 0.3591209 1.2146119
## 5 -4.3992315 -0.6628863 0.73238797 -1.42973513 -0.4548546 -1.7927766
## 6 2.1828713 0.4955701 0.97260758 2.37936592 1.5323763 -0.2714716
## GSM2631089 GSM2631184 GSM2630815 GSM2630763 GSM2631220 GSM2631265
## 1 -1.1469050 -2.1068226 -1.094622476 -1.2105456 -1.3327316 0.4511375
## 2 2.0404509 1.4341754 -0.792890069 0.8265847 -2.3320596 -1.9727023
## 3 0.9519919 1.9416845 -0.008425992 -0.1088397 -2.0135913 1.1194101
## 4 -1.9585188 0.1666770 -0.442004812 -0.5209865 -2.5347666 0.2784159
## 5 1.0614899 -0.3866254 0.395022879 -1.5930349 0.8535663 -1.3680189
## 6 -0.3290498 1.0939036 0.099346939 -0.6914789 -1.1954855 4.4215435
## GSM2631267 GSM2631268 GSM2630793 GSM2630766 GSM2630839 GSM2631004
## 1 -1.3910108 -0.51684365 -1.0915782 -0.5271289 -1.85737643 -0.5077607
## 2 -1.8576513 -1.28001515 -1.0291323 2.2450508 1.20759048 0.1086124
## 3 -1.6007191 2.39137770 0.2842064 -0.9194415 0.40088796 0.1044426
## 4 -1.9591825 -0.31676634 1.3168862 -1.1104305 0.70578663 0.1745603
## 5 0.3552181 0.45836683 -0.7534361 0.2013076 -0.08190932 -0.1203836
## 6 1.3691075 -0.05379665 1.9647524 0.1428687 1.06916993 -0.8284710
## GSM2630991 GSM2631048 GSM2631090 GSM2631104 GSM2631259 GSM2630760
## 1 -1.1200137 -0.15374120 -1.7485344 0.83951225 -0.15204624 -2.278649
## 2 -0.2740660 0.31820532 -1.2277700 0.62982078 -2.26425188 -1.930328
## 3 -0.1889586 -0.11367331 0.1941571 2.46967092 -0.06262525 3.328803
## 4 -0.6192581 1.67550368 1.6309044 0.07444913 1.23983880 1.534670
## 5 -0.1510299 0.01744889 -0.9272385 -3.19901355 -1.32234578 -1.839099
## 6 -2.0476375 -0.86173055 1.2566474 1.90342347 0.28276929 1.665934
## GSM2631051 GSM2630977 GSM2630829 GSM2631264 GSM2630938 GSM2630966
## 1 -0.5666831 -1.4585332 -0.7953674 -0.5058689 -1.34167262 -0.66248760
## 2 -0.9595825 0.9747705 0.5077913 -0.4134887 -1.49280621 -0.22238330
## 3 -0.9975200 1.0420441 -1.1815935 0.8847078 0.78873469 1.02312460
## 4 -1.5579128 0.5954689 -1.5211193 1.1682399 -3.25535716 0.09272712
## 5 0.1818649 1.0241005 -1.4561993 -3.1967779 0.03454615 -0.39298937
## 6 -4.0821619 1.9401031 0.1563138 -1.0634005 -0.30759402 0.81010773
## GSM2631269 GSM2631129 GSM2631013 GSM2630968 GSM2630893 GSM2630804
## 1 -1.4377106 -0.1690443 -2.0468439 0.5080472 -1.551709319 -2.0847908
## 2 -0.7957836 -1.6075090 -0.7125311 -0.4860767 -0.105184265 -0.5118204
## 3 1.2394915 0.6369946 0.2236099 1.8891413 -2.273596535 -1.6314025
## 4 0.3978321 -0.6032546 -1.0590484 3.8411773 0.005706778 -1.7508346
## 5 -1.4730465 -1.4724794 -1.9039240 -1.1086202 -0.838587812 -1.5086458
## 6 0.4062617 -0.1880743 -0.5164401 1.2926072 -2.325125187 1.9326107
## GSM2630995 GSM2631099 GSM2631106 GSM2631146 GSM2631187 GSM2631261
## 1 -1.2917122 -0.011241915 -1.6440218 0.1035240 -0.9993951 -2.3166343
## 2 -2.2323878 0.002007725 1.3628626 -0.6213251 1.8612672 -0.2294808
## 3 -0.7385566 -0.816077852 -0.1680688 0.2415535 -0.1269104 -0.1622849
## 4 1.5285242 -0.450266476 1.5679011 1.5169937 -0.8112110 2.6269330
## 5 0.3996913 0.550356134 1.0799902 -1.0975514 2.1821346 0.4853786
## 6 -0.1289923 -1.867645759 -0.4295978 1.6945018 0.2530582 -2.7495508
## GSM2630950 GSM2631128 GSM2631182 GSM2631078 GSM2631060 GSM2631189 GSM2630891
## 1 0.8778112 0.4071387 -2.5472541 -0.8806653 0.2554293 -0.9725052 -0.2296447
## 2 0.9873322 0.3170618 0.6250888 0.9246205 3.9100273 0.1962435 -1.3776335
## 3 0.5140307 1.1473247 0.9344183 -2.3892165 1.1715350 1.1034283 -2.3470833
## 4 1.4708719 3.4444170 -0.8411303 2.4462454 -1.2068418 0.8252213 -0.5154171
## 5 1.0298641 1.0138297 -0.9226196 -0.7691148 -2.5888016 0.9691800 -0.8953786
## 6 -0.3260145 0.0531847 0.2496307 -0.9531151 -0.9048991 -2.0074446 -2.7725265
## GSM2630976 GSM2630911 GSM2630856 GSM2630813 GSM2630827 GSM2631035 GSM2630841

```

```

## 1 -0.9915351 -1.8750664 -0.4646803 -2.5757948 0.5438591 -1.5434953 -1.3155691
## 2 -0.1011587 2.7102561 1.2333097 -0.8929845 1.8778427 -0.4226869 -1.7352509
## 3 2.0353925 -0.8087983 3.1124356 0.5526647 -1.2078296 -0.8029114 -2.5366676
## 4 2.4922071 -0.5692714 -1.0639654 1.2902401 0.9006928 -1.7780069 -0.7949673
## 5 2.7523347 1.0825978 -0.6203469 -0.3998847 -0.3467247 -0.3430218 3.5007693
## 6 2.1888079 -0.7576955 1.5597356 0.7027909 -2.1996613 0.3258566 0.1920471
## GSM2630918 GSM2631249 GSM2630762 GSM2630817 GSM2630826 GSM2630895
## 1 -1.2986092 -2.42941895 -0.9350261 -0.37996662 -0.4584036 -0.69238779
## 2 0.3465669 0.05149691 1.1108494 0.26030700 -2.2505916 0.89422510
## 3 0.8493634 -0.94044765 -1.2715152 0.81079661 0.1534913 -0.02163383
## 4 -1.5202480 -0.75500556 -0.7416564 0.05808131 -2.0597607 0.66738612
## 5 -1.7931431 -0.81336074 3.4432288 -0.36783057 -0.5720272 -2.78615588
## 6 -1.2726958 1.31302528 -1.0914065 0.06458691 2.0165737 0.41071473
## GSM2630915 GSM2630838 GSM2630900 GSM2631224 GSM2631244 GSM2631245 GSM2631246
## 1 -2.1617964 -2.9460590 -3.2483643 -1.0006847 -1.6309566 -2.0338106 -1.39393569
## 2 -1.1630619 -0.4383554 1.4244318 -2.1456871 -0.7838586 0.5408137 0.07039058
## 3 -0.1535229 -0.6301082 -2.8449701 -1.8251298 -0.8007756 0.2636484 -0.69858486
## 4 1.7257607 0.4795585 0.3067929 -3.1554234 0.3294863 -1.4724555 -0.10892016
## 5 0.5151349 -0.5017270 0.7477007 -0.8315515 1.1038200 0.1027122 2.11318780
## 6 0.3990196 -0.1669634 2.2484424 0.7162219 -0.2258110 0.8898958 0.25544186
## GSM2631247 GSM2630794 GSM2630870 GSM2630890 GSM2630901 GSM2630902
## 1 -2.4753535 0.7991488 -1.40960390 -0.6337080 -1.50662866 -1.8395544
## 2 0.9822148 -0.2551122 0.83835754 -2.3165937 0.25015145 0.2488519
## 3 -1.4221877 0.6031899 -0.09603965 0.8903494 -1.04196712 1.4884015
## 4 -0.8548419 1.3079159 0.12845980 -1.6229459 -0.39361727 -0.9591289
## 5 1.5416138 3.3182213 1.06532594 -1.4372410 -0.08971241 0.3287321
## 6 1.9241219 0.5483882 1.98021674 1.5186465 -2.50160101 -0.1706318
## GSM2630904 GSM2630764 GSM2630781 GSM2630791 GSM2630862 GSM2630898
## 1 0.1253636 0.6046803 -1.95391688 -0.6687297 -0.4248194 -0.08884345
## 2 0.2945730 -1.2632990 -2.16164638 1.0633079 -1.4184357 2.10651825
## 3 1.1114717 -0.9397692 0.94325478 -1.2234248 0.2227079 -0.06559287
## 4 0.0718093 -2.0337492 -0.86594119 -0.8073664 -0.5313776 -0.10746751
## 5 -1.9445166 -0.3978018 0.07081287 -0.3135973 0.3379879 -1.43685363
## 6 -2.8855137 0.6338114 -0.34048153 0.8991488 -2.3761936 -0.41727529
## GSM2630903 GSM2630931 GSM2631150 GSM2631151 GSM2631155 GSM2631156 GSM2631157
## 1 -0.9729093 -0.7205041 -1.8279255 -1.6272927 -1.02188472 -0.3030778 -0.5422523
## 2 2.1354871 3.2053897 0.3156530 0.9477126 -2.23292421 -0.2826414 -2.4505954
## 3 1.1211419 0.8662322 -1.7624382 1.1713244 0.09656530 -1.9224780 -0.1501010
## 4 2.1393422 1.3141604 -0.8578469 -1.7569774 1.25159885 1.3405230 -0.1525208
## 5 -0.9372988 -0.4733000 -0.5279991 0.4059818 -0.34102972 3.3732013 2.2906663
## 6 -0.2351801 0.8366783 -1.1768148 -2.0200991 -0.02430283 -1.1990947 -0.2696430
## GSM2631160 GSM2631169 GSM2631154 GSM2631164 GSM2631166 GSM2631168
## 1 0.04725979 -1.1099001 -1.0273597 -1.4456580 -1.7858547 -1.86787636
## 2 -2.05168976 0.5904935 -0.5084441 1.4034084 0.1951422 0.24927716
## 3 -0.33821937 0.1357660 -2.8707103 0.4180560 -1.2119563 -1.44836641
## 4 1.96718555 1.3873547 -2.6130987 -0.6005612 0.2702867 -1.14206299
## 5 0.81164647 0.5248717 1.6765126 0.6401359 -0.8962291 2.23808444
## 6 -1.22897043 -0.7205411 -0.4615426 2.5951893 1.2824862 -0.06497648
## GSM2631170 GSM2631178 GSM2631019 GSM2630886 GSM2630765 GSM2630831
## 1 -1.74967542 -1.1327505734 -3.008161 -0.9991245 -0.4969681 2.2484317
## 2 -0.82887470 -0.6814695654 -1.645220 1.8169026 2.3886774 1.4722638
## 3 -0.18454581 -1.4938808451 1.659018 -1.6057020 2.3485171 -0.9689060
## 4 -0.04992567 -1.4915257484 -3.531577 1.2048392 -0.9250487 1.3091459
## 5 0.15756681 0.9589022805 -1.191587 -1.1013387 -1.5079540 -0.8022214

```

```
## 6 2.41532699 0.0004061308 3.065940 -3.4045271 1.8612625 -1.5007237
## GSM2631315 GSM2630814 GSM2630776 GSM2631204 GSM2630943 GSM2631199
## 1 2.20927469 0.6651546 2.2135823 1.76544755 1.0995474 0.7946736
## 2 -1.47914201 -1.3190896 -0.5982497 -0.03760468 0.4629934 0.6515067
## 3 -1.11353700 -0.6498948 1.9491410 -2.13039273 0.8892568 -1.0151381
## 4 -0.37045225 -1.1137832 -1.6589510 -1.33587653 -0.7164474 0.8255911
## 5 -0.09040963 1.6516621 0.6565135 1.89144254 -0.9794713 -0.6025549
## 6 -0.19918438 0.9448221 1.3535971 1.49580444 -0.6143860 2.1925398
## GSM2631299 GSM2631223 GSM2631233 GSM2631313 GSM2630808 GSM2631314 GSM2630836
## 1 2.3605210 -2.2012231 -0.02146046 -0.6326650 2.5049257 -2.3359606 -0.1942268
## 2 -1.7673217 -1.7258935 -0.25278472 -1.6125410 -2.3967102 0.5067168 0.2582003
## 3 0.8265741 -1.5204924 -0.51980557 -2.2917605 1.0289422 -0.4145393 -0.0167197
## 4 -0.8028693 -0.5615761 -2.08426765 0.3029939 -0.3865245 -1.3629297 -1.0357384
## 5 0.5505165 0.3204596 2.41206947 -0.7231661 -0.1578205 1.6636426 -1.0738555
## 6 0.3544965 -0.9325789 -1.15736210 -1.7341359 3.7518119 -2.1687875 -0.2819072
## GSM2631216 GSM2630889 GSM2631174 GSM2630894 GSM2630792
## 1 2.53162113 -1.1165815 1.4991129 0.8804499 1.5075483
## 2 -0.08970246 0.9657226 -1.2643005 0.1925006 -0.8114261
## 3 1.75623085 0.3486122 0.2483123 0.3231254 1.1785312
## 4 -0.73100763 -1.2232381 1.0063012 0.2391760 1.8140335
## 5 0.42965378 -1.0862381 0.3670247 0.5919914 -0.2519061
## 6 -0.72533283 -0.2918044 -0.4020399 -0.2462007 -0.4054978
```

So one of our gene names is NA! This isn't useful, so let's remove this row.

```
# your code here
simulated_clean = simulated[-na_ix,]
#simulated_clean
anyNA(simulated_clean)
```

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

We should see if the unique identifiers in our two data sets match. Check for a perfect match using the "identical" function.

```
# your code here
identical(colnames(simulated_clean[, -1]), as.character(pheno[, 1]))
```

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

So that we don't lose any work, let's clean up our workspace to include only our cleaned simulatedession and pheno data sets, which we can reload later.

Exploratory Data Analysis

In this section we are going to explore some of the data we have, and maybe develop a diagnostic signature for Parkinson's disease.

First, load in your data from yesterday.

Let's re-examine our pheno data set with the summary function again.

```
summary(pheno)
```

```
## geo_accession      disease_label      sex      moca_score
## Length:550      Length:550      Length:550      Min.   : 0.00
## Class :character Class :character Class :character 1st Qu.: 0.00
## Mode  :character Mode  :character Mode  :character Median :26.00
```



```
##                                     Mean   :17.62
##                                     3rd Qu.:29.00
##                                     Max.   :30.00
##                                     NA's   :16
##      AgeMaster      overallUPDRS      overallHoehn
## Min.   :10.00      Min.   : 0.000      Min.   :0.000
## 1st Qu.:53.00      1st Qu.: 0.000      1st Qu.:0.000
## Median :60.00      Median : 0.000      Median :0.000
## Mean   :59.37      Mean   : 1.525      Mean   :1.364
## 3rd Qu.:67.00      3rd Qu.: 0.600      3rd Qu.:4.000
## Max.   :82.00      Max.   :18.400      Max.   :4.500
## NA's   :199       NA's   :264       NA's   :258
```

We need to further delve into our disease label in order to simplify some of this analysis. Attach your pheno data frame using the attach function, and then summarize the disease label vector.

```
attach(pheno)
summary(disease_label)
```

```
##      Length      Class      Mode
##      550 character character
```

Here we have the counts of all the diseases in our data set. If you look at the actual excel file (not the csv), I've put in a dictionary for these acronyms if you're curious. Here, our controls and our genetic unaffected are both considered to be healthy controls. Any label which contains PD is some subset of Parkinson's Disease, and the other labels represent other neurological disorders. We need to make a variable which records a 1 for our cases, and a 0 for our controls. Here, since we are interested in a signature that distinguishes PD from our other disease, the other diseases are technically part of the control set.

Try to set your case control vector using the grep function to find the indices which contain "PD". At the end, sum your case vector to check that it worked. Make another variable of the words "case" and "control"

```
pdI = grep("PD", disease_label, value=F)
case = rep(0, length(disease_label))
case[pdI] = 1
sum(pdI)
```

```
## [1] 100799
```

```
##alternative
```

```
case = grepl("PD", disease_label)*1 #returns list of True and False
sum(case)
```

```
## [1] 251
```

```
caseName = ifelse(case ==1, "case", "control")
```

We need to find differentially simulated genes. You'll learn more about this later. For now, feel free to use some of my code. Start by downloading the limma package

```
head(simulated_clean[,1])
```

```
## [1] "A1BG-AS1" "A1CF"      "A2M"      "A2M-AS1"  "A2ML1"    "A2MP1"
```

```
## If using Windows, first go to https://cran.rstudio.com/bin/windows/Rtools/ and install the appropriate
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
#BiocManager::install("limma")
library(limma)
```

We will use the following code. **Comment this code with your thoughts below.**

```
#subset our data for a training and test set
set.seed(2) #seeding the dataset
prob = runif(ncol(simulated_clean)-2) #generating probability based on random uniformity of values in d
k = which(prob>=0.3333333) # selecting values with occurrences more than 0.33
eset = simulated_clean[,2:ncol(simulated_clean)] #taking columns from GSM2631309.
eset = eset[,k] #updating the column vals with occurrences of more than 33%.
rownames(eset) = simulated_clean[,1]
design = model.matrix(~0+as.factor(case[k]))
fit = eBayes(lmFit(eset,design)) #applying empirical bayesian statistical algorithm for differentiating
topTable(fit, coef=2) #table of top genes from linear model fit from second column.
```

##	logFC	AveExpr	t	P.Value	adj.P.Val	B
## EXOC3L4	3.142306	1.471867	29.50981	3.960809e-149	4.472996e-145	329.8665
## FAM132A	-3.159535	-1.461491	-29.50492	4.328845e-149	4.472996e-145	329.7780
## MDM2	3.127701	1.426391	29.45016	1.171911e-148	8.072902e-145	328.7868
## CCR3	-3.145072	-1.547047	-29.40990	2.436485e-148	1.258810e-144	328.0583
## MYO9A	-3.172079	-1.571136	-29.27467	2.842523e-147	1.174872e-143	325.6132
## GADD45GIP1	-3.086818	-1.391784	-29.08998	8.109246e-146	2.793095e-142	322.2781
## ANXA2	-3.106348	-1.517455	-29.04306	1.898445e-145	5.604751e-142	321.4315
## CCNJ	-3.123721	-1.454289	-28.99762	4.324144e-145	1.117034e-141	320.6122
## EMC6	3.070413	1.590585	28.97713	6.268035e-145	1.439280e-141	320.2427
## GEMIN4	3.100428	1.427101	28.89015	3.027401e-144	6.256426e-141	318.6752

```
results = topTable(fit, coef=2, number=Inf) #number indicates maximum number of genes
```

Here, we have our gene names, our log fold change for simulated session, average simulated session, t statistic, pvalue, adjusted pvalue (for multiple testing!!), and the log odds of differential simulated session.

Next, we select those genes that have adjusted p-values below 0.001. **Comment the code with your thoughts about what its doing below.**

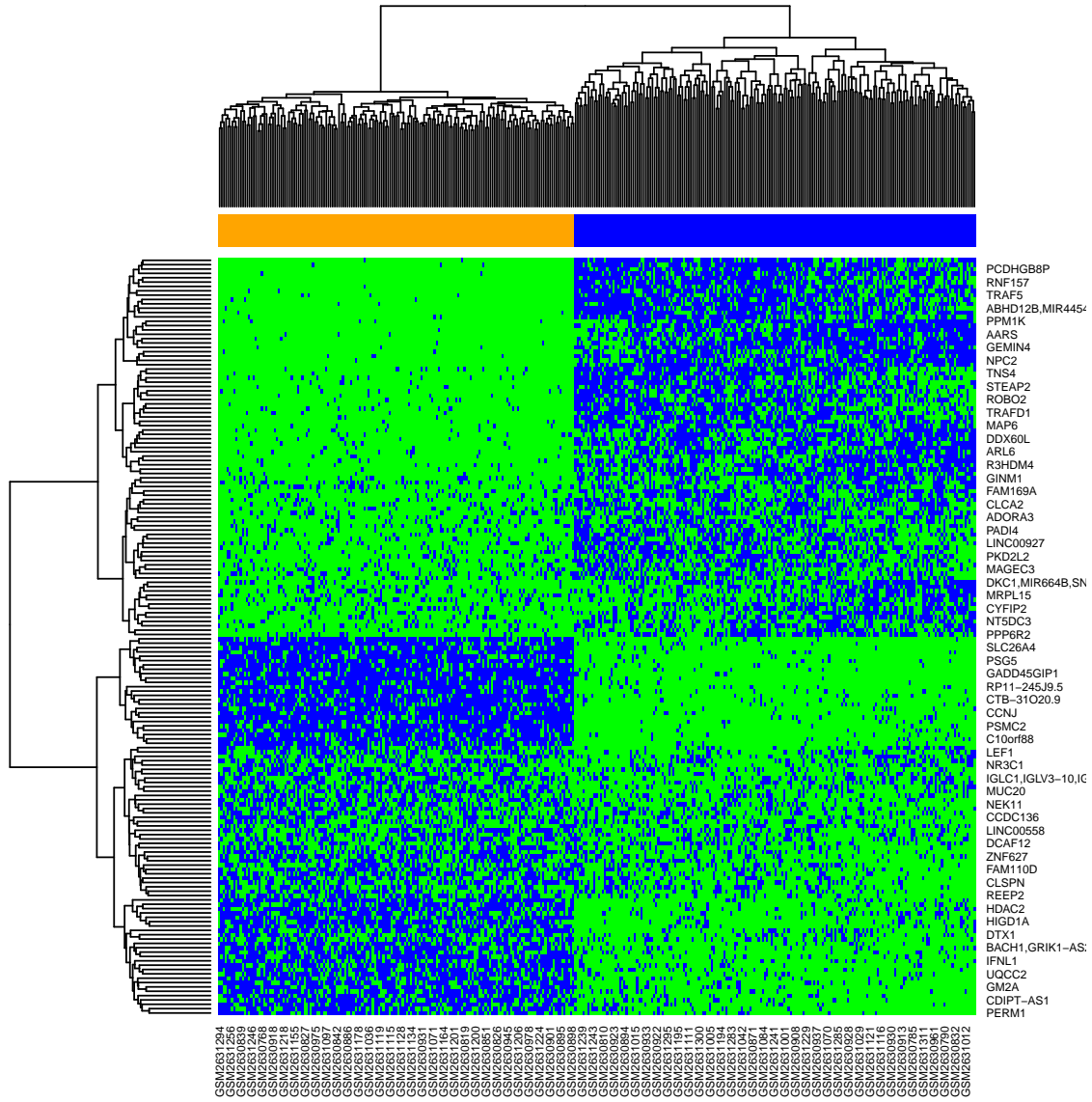
```
selected = row.names(results)[p.adjust(results$P.Value, method="fdr")<0.001] #selecting genes based on p
direction = sign(results$logFC) #using sign() function to get the sign of logFC scores for real numbers
esetSel = eset[selected, ] #esetSel consists of rows of genes that were selected based on p values less
nrow(esetSel)
```

```
## [1] 174
```

Okay! So we're now looking at just 175 genes!

We are going to make a heat map here. I've provided the code, but **try changing colours, labels, etc. to make it your own.**

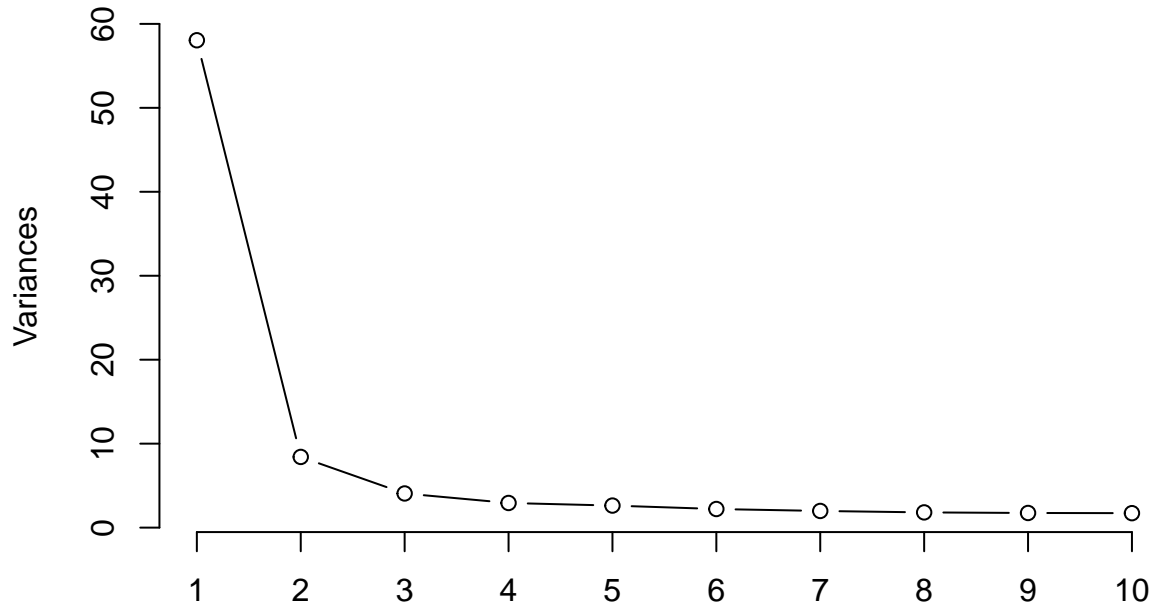
```
patientcolors = ifelse(case[k]==1,"orange","blue")
heatmap(as.matrix(esetSel), col=c("blue", "green"), ColSideColors=patientcolors, distfun = function(x) c
```



Notice the annotation bar along the top. This indicates PD vs not PD samples. This heat map is an example of a ‘non-supervised method’ - where we didn’t feed the labelled data to the algorithm. Instead, it is just clustering similar samples together. Because all of our PD samples cluster away from the non-PD samples, we are relatively certain we’ve picked good biomarkers! We should also check a PCA plot.

```
pc = prcomp(t(esetSel),center=T,scale=T) #reduce dimensionality to eigenvectors
plot(pc,type="l",main="Checking the number of Principle Components")
```

Checking the number of Principle Components



Again, I’ve provided code for you here. **Change it to something you like better!**

```
#install.packages("devtools")
#install.packages("ggfortify")

library(devtools)
library(ggpubr)
#library(ggfortify)
#install_github("vqv/ggbiplot")

library(ggbiplot)

g = ggbiplot(pc, obs.scale = 1.5, var.scale = 1.5,
             groups = as.factor(caseName[k]), ellipse = F,
             circle = T, labels=disease_label[k],var.axes = F, col = c("blue", "orange"))
g = g + scale_color_discrete()
g = g + theme(legend.direction = 'horizontal',
             legend.position = 'top')
g = g + theme_classic() + theme_dark()
```

```
print(g)
```


We have separation! Notice the obvious differences between cases and controls.

Make a variable which only contains the differential gene names and call it diffGenes AND print out all of these gene names using one line of code.

```
#your code here
diffGenes = selected
```

To use these genes as a classifier, we will need to define a score function. Our score will be the sum of the average simulatedession for the upregulated (positive) genes and the average for the down regulated (negative) genes. Here, I've written you a function which will do this. Please enter it and **make comments to show you understand what its doing.**

```
PDscore = function(x,g,v,s){
  #x simulatedession values for a sample
  #g all the genes
  #v the diffGenes
  #s is the sign of the logFC

  i = which(g%in%v)
  x = x[i] #element in genes getting updated in a sample
  s = s[i] #logFC values of a element in genes, updated to s variable
  p = c() #initializing a vector
  n = c()
  for(i in 1:length(x)){ #looping in a sample and checking logFC scores > 0
    if(s[i]>0){
      p = append(p,(x[i])) # adding these values to a vector
    }
    else if(s[i]<0){
      n = append(n,(x[i])) #negative logFC scores are updated here.
    }
  }

  if(is.null(p)){p[1]=0} #replacing null values with 0
  if(is.null(n)){n[1]=0} #replacing null values with 0

  score = mean(p)-mean(n) #mean difference between vectors that have logFC positive and negative scores
  return(score)
}
```

Now we can apply our function to our simulatedession set to define a score for each patient. **Comment what this is doing and why each step is necessary!**

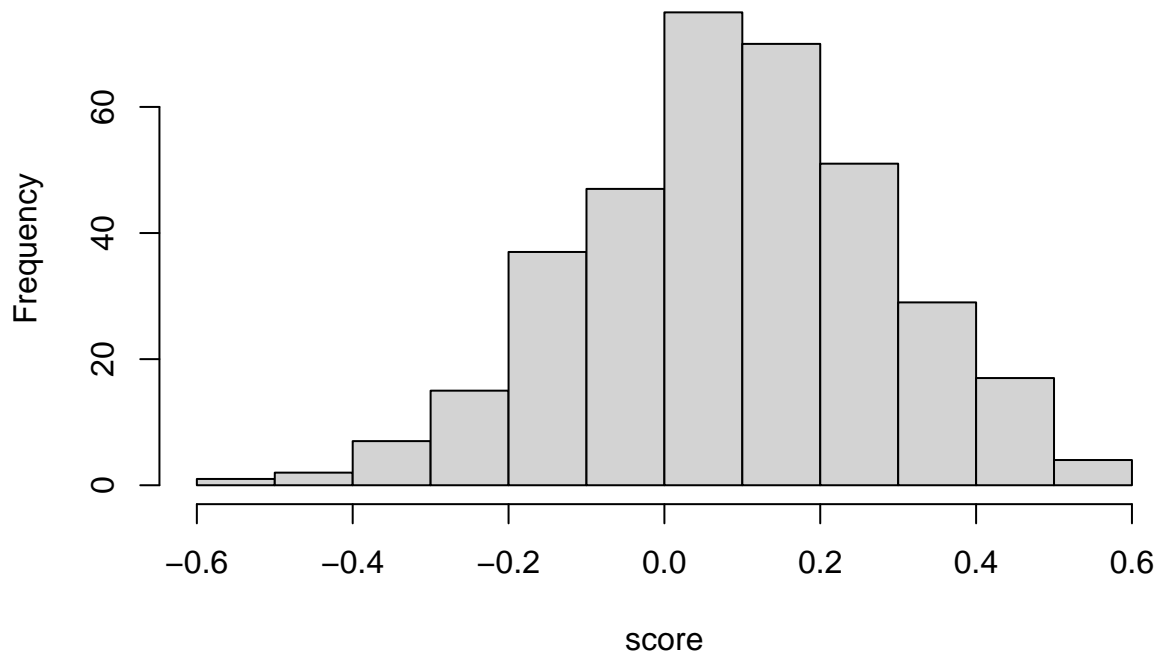
```
score = c() #initializing empty vector

#char vals of results in simulated dataset.
allGenes = as.character(simulated[as.character(simulated$GeneName)%in%rownames(results),1])

#lopping in the subset and updating the score.
for(i in 1:ncol(eset)){
  score[i] = PDscore(eset[,i],allGenes,diffGenes,direction)
}

hist(score,main="Distribution of our PD Scores")
```

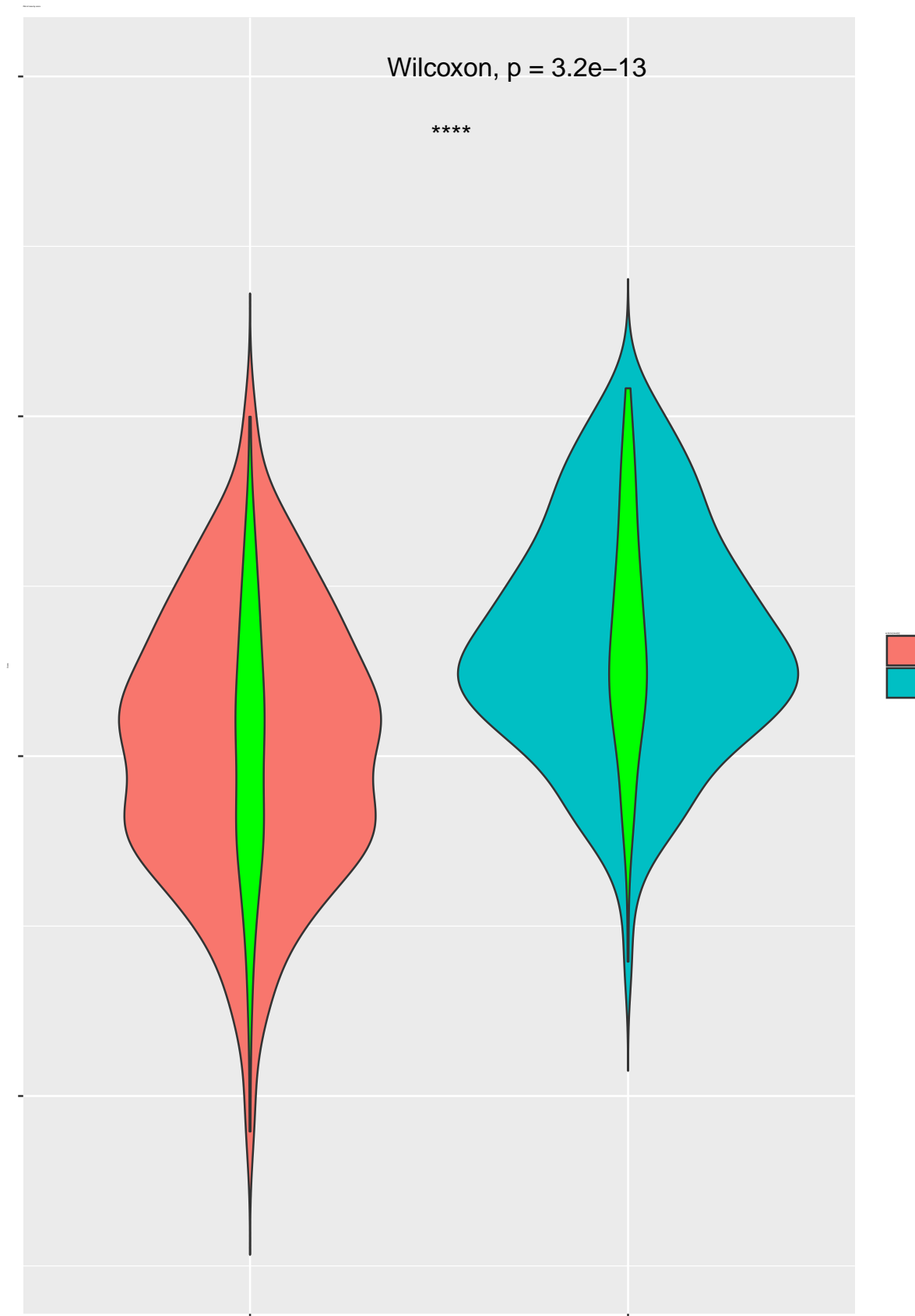
Distribution of our PD Scores



Now we'll use ggplot to make and interpret a violin plot of our score. I've provided some code to do this, but **try to change labels, colours, etc. to make it your own.**

```
#install.packages("ggpubr")
library(ggpubr)

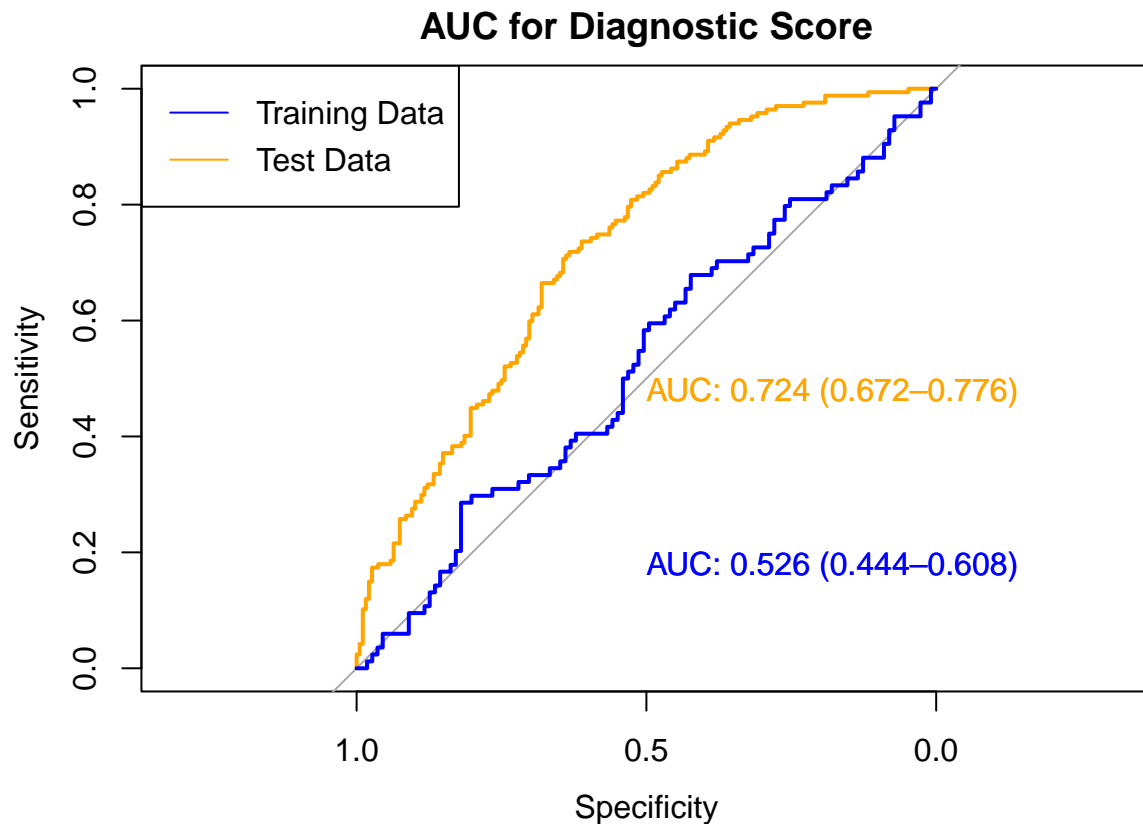
df = data.frame(cbind(case[k],score))
dp = ggplot(df, aes(x=as.factor(case[k]), y=score, fill=as.factor(case[k]))) +
  geom_violin(trim=FALSE)+
  geom_violin(width=0.1, fill="green")+
  labs(title="Plot of case by score",x="Case ", y = "Score")+
  stat_compare_means(label.x = 1.5, label.y = 1, size=5)+
  stat_compare_means(aes(label = ..p.signif..),
                     label.x = 1.5, label.y = 0.9, size = 5)
dp + theme(text = element_text(size = 0.5))
```

This shows not only the boxplot of our data, but also the distribution of our data points around the boxplot! As before, we can see that we DO have significant separation for our score, and we can see that the cases are trending to have a higher score. With more time and data cleaning we may be able to find something here!

Let's make a roc plot, first with our training data, and then with our test data. As before, **play with the plot options to make something you like!**

```
#install.packages("verification")
#install.packages("pROC")
library("pROC")
testEset = simulated[,2:ncol(simulated)]
testEset = testEset[,-k]
newScore = apply(testEset,2,FUN=PDscore,allGenes,diffGenes,direction)
plot.roc(case[k]~score, data=df,legacy.axes=F,print.auc=T, ci=T, main="AUC for Diagnostic Score",col="orange")
plot.roc(case[-k]~newScore,data=data.frame(cbind(case[-k],newScore)),add=T,print.auc=T, ci=T, col="blue")
legend("topleft",c("Training Data","Test Data"),lty=c(1,1),col=c("blue","orange"))
```



Notice that our score does better with our training data - this is expected! This is why we need to split our data, to avoid problems with over-fitting. These scores are better than random (the grey line), but we'd like to see an AUC as close to 1 as possible. Let's see if we can do better!

Statistics!

We can run a t-test to see if our score is significantly different between cases and controls. Try using the `t.test` function in R.

```

# your code here
allTscore = c(score, newScore)
mergeCase = c(case[k], case[-k])

t.test(allTscore[mergeCase == 0], allTscore[mergeCase == 1])

##
## Welch Two Sample t-test
##
## data: allTscore[mergeCase == 0] and allTscore[mergeCase == 1]
## t = -6.2832, df = 535.65, p-value = 6.875e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.14076017 -0.07370768
## sample estimates:
## mean of x mean of y
## 0.009489271 0.116723192

```

The mean scores for our cases and controls are close, but they are significantly different with an extremely small p-value of $2.787e-13$. This highlights a classical statistical fallacy - while small p-values are great, they are often meaningless without a large enough effect size. Here, we have achieved significance due to the large sample size of our study, hence our study is adequately powered.

We could also run a simple regression to examine the impact of the score on the log odds of being a case.

```

# your code here
logistic = glm(case[k]~score, family="binomial")
summary(logistic)

##
## Call:
## glm(formula = case[k] ~ score, family = "binomial")
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9915  -1.0119  -0.5391   1.0851   2.0686
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5627     0.1343  -4.191 2.78e-05 ***
## score         4.8012     0.6966   6.892 5.51e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 490.89  on 354  degrees of freedom
## Residual deviance: 430.83  on 353  degrees of freedom
## AIC: 434.83
##
## Number of Fisher Scoring iterations: 3

```

Summarize this output!

Again, we conclude that the score is a statistically significant indicator of the odds of having PD. Let's build a larger model which examines other phenotype variables.

First, build a data frame which includes all the model data we're interested in. Start with the age variables in your pheno set, and then use the `cbind()` function to add on our scores and the binary case vector. Print a summary of the model data.

```
# your code here
```

```
modelData = cbind(pheno[k,3:7],score)
summary(modelData)
```

```
##      sex          moca_score      AgeMaster      overallUPDRS
## Length:355      Min.   : 0.00      Min.   :26.00      Min.   : 0.000
## Class :character 1st Qu.: 0.00      1st Qu.:53.50      1st Qu.: 0.000
## Mode  :character Median :25.00      Median :62.00      Median : 0.000
##              Mean   :17.01      Mean   :60.15      Mean   : 1.489
##              3rd Qu.:29.00      3rd Qu.:67.00      3rd Qu.: 0.600
##              Max.   :30.00      Max.   :82.00      Max.   :15.800
##              NA's   :12         NA's   :132         NA's   :167
## overallHoehn      score
## Min.   :0.000      Min.   : -0.55225
## 1st Qu.:0.000      1st Qu.: -0.05932
## Median :0.000      Median : 0.08870
## Mean   :1.412      Mean   : 0.08669
## 3rd Qu.:4.000      3rd Qu.: 0.21846
## Max.   :4.500      Max.   : 0.54121
## NA's   :161
```

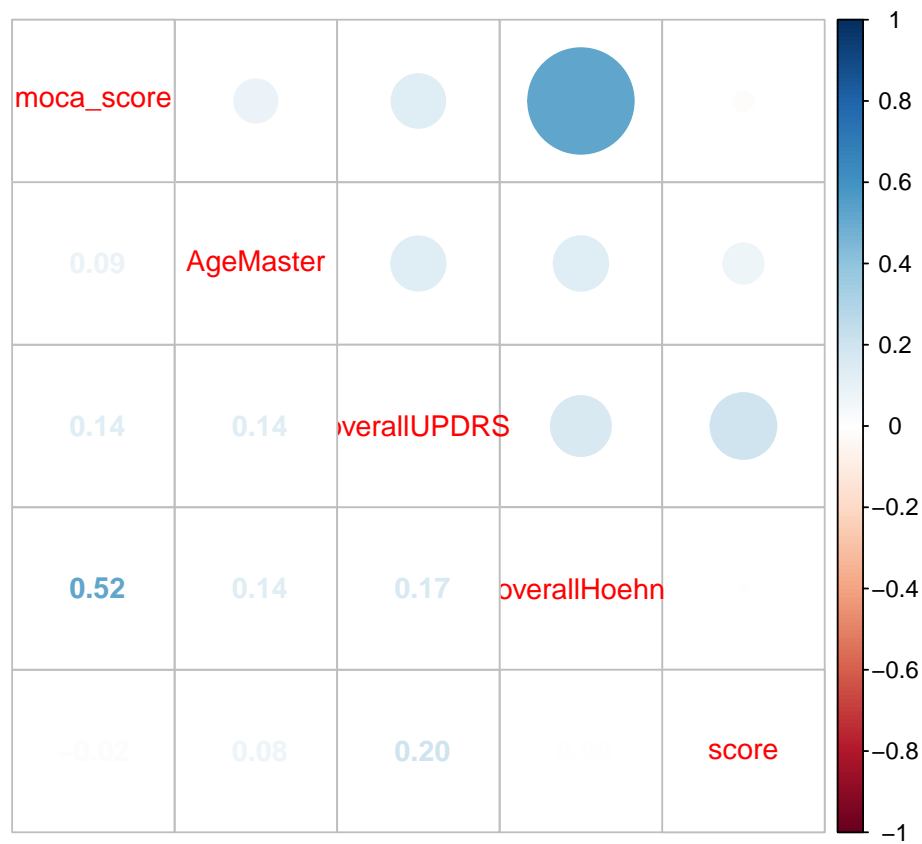
We should examine the correlations in our data set. You can do this quickly by building a correlation plot matrix.

```
#install.packages("corrplot")
```

```
library(corrplot)
```

```
M = cor(modelData[,-1],use="pairwise.complete.obs") #for missing data
```

```
corrplot.mixed(M)
```



How would you interpret this output? **Answer below!**

The blue color indicates the highest normalized correlation score between 0 and 1, from the plot it can be interpreted that not many variables are correlated with each other. The red color indicates no correlation exists between the variables.

Let's build our first model. Here, we consider the case as our dependent variable, and the others as our explanatory variables.

```
model1 = glm(case[k]~.,family=binomial,data=modelData)
summary(model1)

##
## Call:
## glm(formula = case[k] ~ ., family = binomial, data = modelData)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8009  -0.3493   0.2196   0.6385   1.6723
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.90400     2.57394  -1.905   0.0567 .
## sex Male       0.66544     0.86314   0.771   0.4407
## moca_score     0.11149     0.04786   2.330   0.0198 *
## AgeMaster      0.04711     0.04002   1.177   0.2392
## overallUPDRS   0.28691     0.12524   2.291   0.0220 *
## overallHoehn   0.17082     0.22774   0.750   0.4532
## score          4.98031     2.39979   2.075   0.0380 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 68.752  on 55  degrees of freedom
## Residual deviance: 41.104  on 49  degrees of freedom
## (299 observations deleted due to missingness)
## AIC: 55.104
##
## Number of Fisher Scoring iterations: 6
```

We will iteratively remove variables with the highest p-values, and then rerun the model until we have our optimal fit!. Try this on your own first.

```
modelData$case = case[k]
modelData2 = na.omit(modelData)
model2 = glm(case~sex+AgeMaster+moca_score+overallUPDRS+overallHoehn+score,family=binomial,data=modelData2)
summary(model2)

##
## Call:
## glm(formula = case ~ sex + AgeMaster + moca_score + overallUPDRS +
##      overallHoehn + score, family = binomial, data = modelData2)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8009  -0.3493   0.2196   0.6385   1.6723
```

```
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.90400    2.57394  -1.905  0.0567 .
## sex Male      0.66544    0.86314   0.771  0.4407
## AgeMaster     0.04711    0.04002   1.177  0.2392
## moca_score    0.11149    0.04786   2.330  0.0198 *
## overallUPDRS  0.28691    0.12524   2.291  0.0220 *
## overallHoehn  0.17082    0.22774   0.750  0.4532
## score         4.98031    2.39979   2.075  0.0380 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 68.752  on 55  degrees of freedom
## Residual deviance: 41.104  on 49  degrees of freedom
## AIC: 55.104
##
## Number of Fisher Scoring iterations: 6
modelData2$sex = NULL #removing sex column since it threw error.
#modelData2$AgeMaster = NULL

model3 = glm(case~AgeMaster+moca_score+overallUPDRS+overallHoehn+score,family=binomial,data=modelData2)
summary(model3)

##
## Call:
## glm(formula = case ~ AgeMaster + moca_score + overallUPDRS +
##      overallHoehn + score, family = binomial, data = modelData2)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6673  -0.3431   0.2415   0.6291   1.7149
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.86113    2.52803  -1.923  0.0545 .
## AgeMaster     0.05279    0.03858   1.368  0.1712
## moca_score    0.11430    0.04795   2.384  0.0171 *
## overallUPDRS  0.25616    0.11283   2.270  0.0232 *
## overallHoehn  0.21284    0.22239   0.957  0.3385
## score         4.83636    2.36102   2.048  0.0405 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 68.752  on 55  degrees of freedom
## Residual deviance: 41.715  on 50  degrees of freedom
## AIC: 53.715
##
## Number of Fisher Scoring iterations: 6
```

This is our final model! Notice that our largest effect size is controlled by our genetic score. At a first glance, we might assume this means that the score has the largest effect on the model. However, if we recall how to interpret our coefficients, the estimated effect size is the change in log odds of being a case for a 1 unit increase in our score. Think about the score distribution: the range of our scores is fairly small. In contrast, the range of the updrs scores varies from 0 to 36. Keep in mind the scale of our data when interpreting these models!

Compare this to your outcome if you use a step function to reduce the model:

```
#your code here
stepModel = step(model3)

## Start:  AIC=53.72
## case ~ AgeMaster + moca_score + overallUPDRS + overallHoehn +
##       score
##
##              Df Deviance    AIC
## - overallHoehn  1   42.680 52.680
## <none>              41.715 53.715
## - AgeMaster     1   43.863 53.863
## - score         1   46.543 56.543
## - overallUPDRS  1   47.771 57.771
## - moca_score    1   50.233 60.233
##
## Step:  AIC=52.68
## case ~ AgeMaster + moca_score + overallUPDRS + score
##
##              Df Deviance    AIC
## - AgeMaster     1   44.561 52.561
## <none>              42.680 52.680
## - score         1   47.094 55.094
## - overallUPDRS  1   49.515 57.515
## - moca_score    1   50.361 58.361
##
## Step:  AIC=52.56
## case ~ moca_score + overallUPDRS + score
##
##              Df Deviance    AIC
## <none>              44.561 52.561
## - score         1   49.481 55.481
## - moca_score    1   50.543 56.543
## - overallUPDRS  1   54.169 60.169

summary(stepModel)

##
## Call:
## glm(formula = case ~ moca_score + overallUPDRS + score, family = binomial,
##      data = modelData2)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3970  -0.5641   0.2373   0.5741   2.0603
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

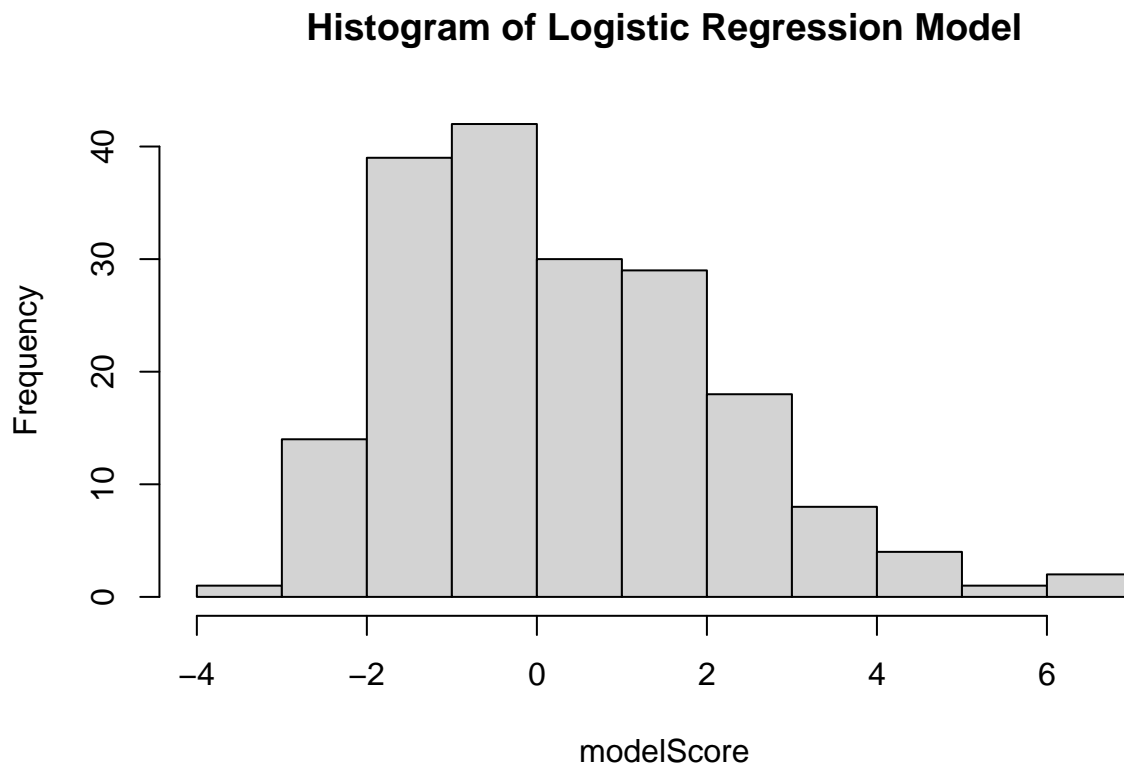


```
## (Intercept)  -1.40257    0.64691   -2.168    0.03015 *
## moca_score    0.08638    0.04217    2.048    0.04052 *
## overallUPDRS  0.30910    0.11082    2.789    0.00529 **
## score         4.47227    2.14542    2.085    0.03711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 68.752  on 55  degrees of freedom
## Residual deviance: 44.561  on 52  degrees of freedom
## AIC: 52.561
##
## Number of Fisher Scoring iterations: 6
```

Let's predict the probability of having a case given our manually reduced model. Make a histogram of the score from this model.

your code here

```
modelScore = predict(stepModel, newdata = modelData)
hist(modelScore, main="Histogram of Logistic Regression Model")
```



Like before, we'll build a violin plot to compare the output of our regression model. See if you can adapt the violin plot code from before to do this now.

```
df = data.frame(cbind(case[k], modelScore))
dp = ggplot(df, aes(x=as.factor(case[k]), y=modelScore, fill=as.factor(case[k]))) +
```

```

geom_violin(trim=FALSE)+
geom_violin(width=0.1, fill="white")+
labs(title="Plot of case by score",x="Case ", y = "Score") +
stat_compare_means(label.x = 1.5, label.y = 1, size=5)+
stat_compare_means(aes(label = ..p.signif..),
                    label.x = 1.5, label.y = 0.9, size=5)
dp + theme(text = element_text(size = 0.2))

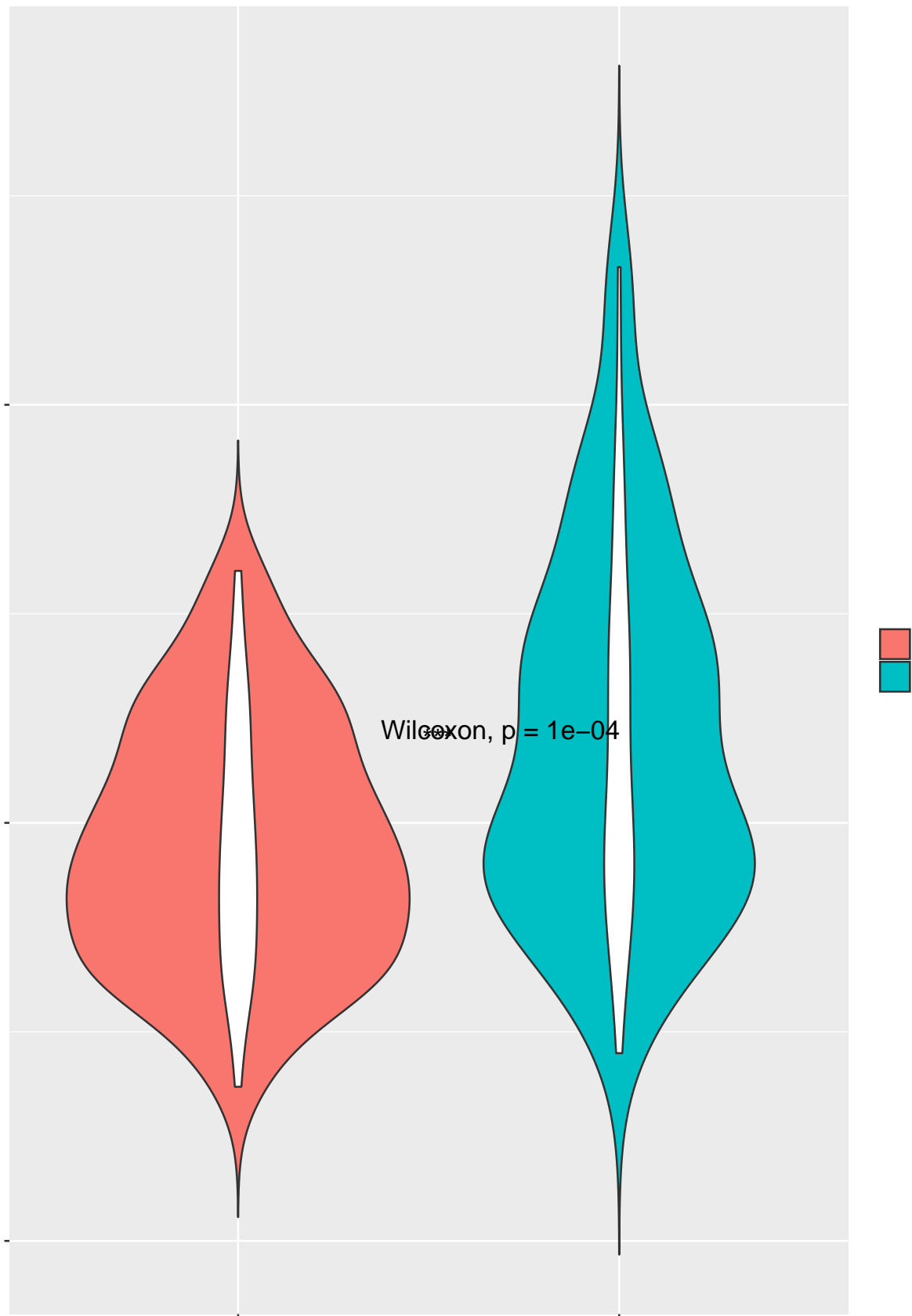
```

```

## Warning: Removed 167 rows containing non-finite values (stat_ydensity).
## Removed 167 rows containing non-finite values (stat_ydensity).

## Warning: Removed 167 rows containing non-finite values (stat_compare_means).
## Removed 167 rows containing non-finite values (stat_compare_means).

```



Now we're starting to see a clearer separation of scores! It's clear that by including the established tests to pre-screen patients for PD and other neurological diseases we have improved overall performance. While this may be an obvious conclusion, it is worth noting that the context with which our diagnostic signature would be used would be on patients already exhibiting potential PD symptoms. Clearly this needs a little more work, but for a first pass at assessing raw data, it's not bad!

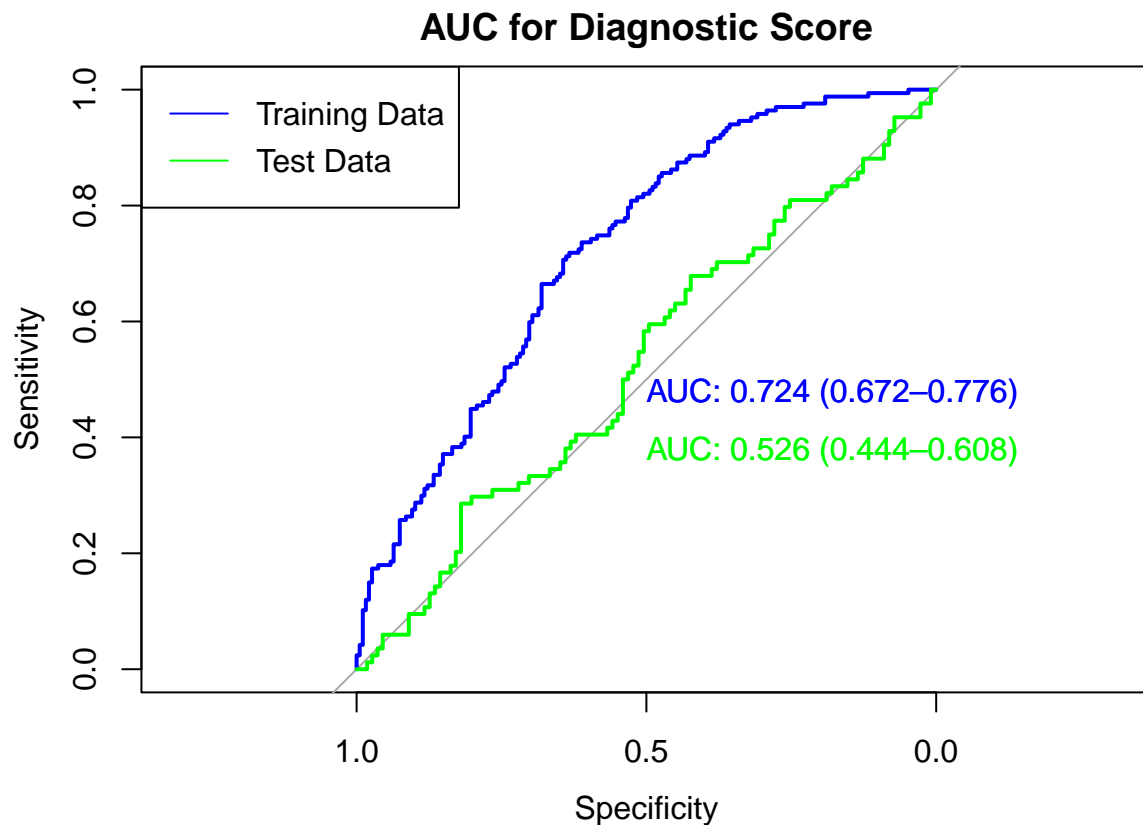
Again, we can examine ROC curves. I've done some of the set up to get the data in the right format. Use the ROC code above to then build your own plot!

```
library("pROC")
nd = cbind(pheno[-k,],newScore)
colnames(nd) = c(colnames(nd[1:ncol(nd)-1]),"score")
newMScore = predict(stepModel,newdata=nd)

plot.roc(case[k]~score, data=df,legacy.axes=F,print.auc=T, ci=T, main="AUC for Diagnostic Score", col="blue")

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
plot.roc(case[-k]~newScore,data=data.frame(cbind(case[-k],newScore)),add=T,print.auc=T, ci=T, col="green")

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
legend("topleft",c("Training Data","Test Data"),lty=c(1,1),col=c("blue","green"))
```



Here, we have a notable increase in AUC, particularly for our training data. Our test data shows an overall improvement as well, although with a large confidence interval. There are clearly some data points in here

which are abnormal - and perhaps worth investigating.

##Congratulations, you have finished the R Bootcamp Assignment!