EDA

Kyoko Yamaguchi

2/28/2022

Loading in libraries, data.

```
library(dplyr)
library(tidyr)
library(Rtsne)
library(umap)
library(ggplot2)
```

```
# Loading in partially cleaned tables
clincsv<-read.csv(file="clinicaltablesmall.csv", row.names = 1)
cpgcsv<-read.csv(file="hnsc_cpg1000.csv", row.names = 1)
```

After finishing up the cleaning, the clinical table that we originally read in, "clinicaltablesmall.csv", looks like this:

```
sortedclin[1:5,5:10]
```

```
##              patient.anatomic_organ_subdivision patient.laterality
## TCGA-4P-AA8J                       Oral Tongue              Right
## TCGA-BA-4074                       Oral Tongue               Left
## TCGA-BA-4075                       Oral Tongue              Right
## TCGA-BA-4076                            Larynx              Right
## TCGA-BA-4077                     Base of tongue              Right
##              patient.prospective_collection patient.retrospective_collection
## TCGA-4P-AA8J                             NO                                YES
## TCGA-BA-4074                             NO                                YES
## TCGA-BA-4075                             NO                                YES
## TCGA-BA-4076                             NO                                YES
## TCGA-BA-4077                             NO                                YES
##              patient.gender patient.birth_days_to
## TCGA-4P-AA8J           MALE                -24222
## TCGA-BA-4074           MALE                -25282
## TCGA-BA-4075           MALE                -17951
## TCGA-BA-4076           MALE                -14405
## TCGA-BA-4077         FEMALE                -16536
```

And the CPG table, "hnsc_cpg1000.csv", looks like this:

```
sortedcpg[1:5,1:5]
```

```
##              cg00106345 cg00123762 cg00178984 cg00282249 cg00347563
## TCGA-4P-AA8J  0.5807570 0.06970429  0.6541720 0.03396252 0.27660392
## TCGA-BA-4074  0.6859209 0.82456800  0.7617553 0.54832958 0.60928171
## TCGA-BA-4075  0.7277643 0.38678205  0.5804289 0.78487857 0.18836765
## TCGA-BA-4076  0.9054642 0.46605054  0.8747722 0.07592710 0.84043269
```

```
## TCGA-BA-4077  0.9104171 0.27383676  0.4287539 0.53396373 0.02828893
```

Missing data

The current matrix of patients vs. CPGs does not contain any missing values. There was missing data in the original matrix of patients vs. CPGs, but the missing data was 1) kicked out if the CPG had greater than 30% missing values, and 2) the missing values for any remaining CPGs were imputed via mean imputation. The code for the mean imputation and filtering is here for reference. The Rmarkdown could not be knitted from the original table because the file size of the original table of patients vs unfiltered 400,000+ CPGs was > 1GB and not possible to share over cloud to the rest of the group. This matrix was further filtered to include only the top 1000 most variably expressed CPGs (using standard deviation).

```
# Drop CPGs that have greater than X% NAs in the row
# (as in, X% or more of patients have "NA" values for these CGs)
mycutoff<-0.3
cutoffrowna<-mycutoff*ncol(RAW)
numberofna<-which(rowSums(is.na(RAW))/cutoffrowna > mycutoff)
RAW_cutoffna<-RAW[-numberofna,]
dim(RAW_cutoffna)

# Impute the rows that have X% or less missing
meanimputation <- function(x) {x[is.na(x)] <- mean(x, na.rm=TRUE); return(x)}
RAW_cutoffna_impute  <- as.data.frame(apply(RAW_cutoffna, 1, meanimputation ))
RAW_cutoffna_impute<-t(RAW_cutoffna_impute)

# Keep top 1000 most variably expressed CPGs
cutoff_top1000<-min(head(sort(rowSds(RAW_cutoffna_impute), decreasing=TRUE),1000))
keepers_top1000<-which(rowSds(RAW_cutoffna_impute) >= cutoff_top1000)
RAW_cutoffna_impute_top1000<-RAW_cutoffna_impute[keepers_top1000,]
dim(RAW_cutoffna_impute_top1000)
write.csv(RAW_cutoffna_impute_top1000, "hnsc_cpg1000.csv")
```

Additionally, I realized this matrix had 52 duplicate patient IDs, which have been removed because it did not make sense for these to be in the dataset, since they correspond to code "06" and "11", which stand for "metastatic" and "solid tissue normal", respectively. We are only keeping the samples labeled "01" which stands for "primary solid tumor". (Reference: https://gdc.cancer.gov/resources-tcga-users/tcga-code-tables/sample-type-codes)

```
table(v)
```

```
## v
## 01A 01B 06A 11A 11B
## 523   5   2  45   5
```

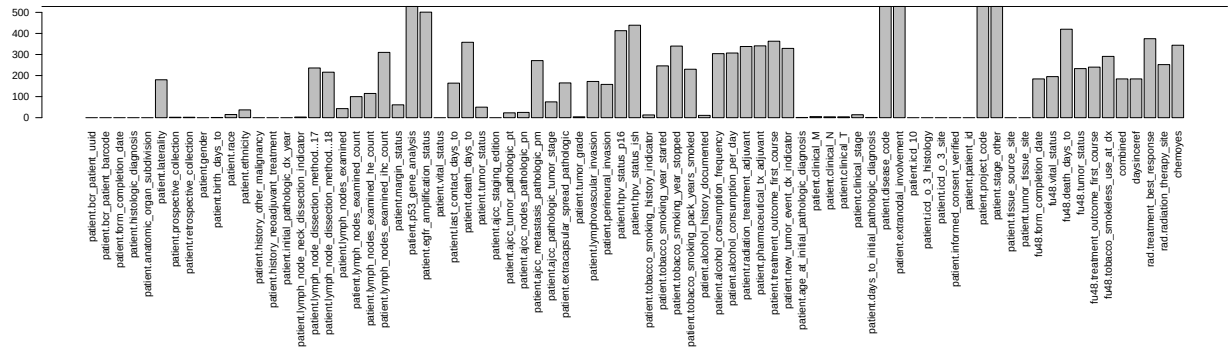
There are still missing data in the matrix of patients vs. clinical information, which we can use as a table of response variables. For your reference, the code below shows how certain interesting values from the original clinical tables from TCGA were hand-selected and wrangled to generate the table "clinicaltablesmall.csv". Since some tables contained multiple entries for the same patient (ex. if they came back for more than one follow up appointment), I prioritized keeping one account over another to make the final clinical table have only one entry per "bcr_patient_barcode". For example, for the follow up table, in the case there were two or more follow up entries, I have prioritized the FINAL follow up date.

```
patient<-clinical.BCRtab.all2$clinical_patient_hnsc
rad<-clinical.BCRtab.all2$clinical_radiation_hnsc
drug<-clinical.BCRtab.all2$clinical_drug_hnsc
fu48<-clinical.BCRtab.all2$clinical_follow_up_v4.8_hnsc
```

```
# Wrangling the radiation therapy table
rad2<-rad[,c("rad-bcr_patient_barcode","rad-treatment_best_response","rad-radiation_therapy_site")]
rad3<-distinct(rad2)
rad3$`rad-radiation_therapy_site` %>% table()
rad4<-rad3[which(rad3$`rad-radiation_therapy_site`=="Primary Tumor Field"),]
unique(rad4$`rad-bcr_patient_barcode`) %>% length()
radfinal<-rad4

# Wrangling of drug therapy table
drug2<-drug[,c("drug-bcr_patient_barcode","drug-pharmaceutical_therapy_type")] %>%distinct()
drug2$chemoyes<-ifelse(drug2$`drug-pharmaceutical_therapy_type` =="Chemotherapy", "Y", NA)
drug3<-drug2[,c(1,3)] %>% drop_na() %>% distinct()
drug3$`drug-bcr_patient_barcode` %>% length()
drugfinal<-drug3

# Wrangling the follow up table
interestedincols<-c("fu48-bcr_patient_barcode",
"fu48-form_completion_date",
"fu48-vital_status",
"fu48-death_days_to",
"fu48-tumor_status",
"fu48-treatment_outcome_first_course",
"fu48-tobacco_smokeless_use_at_dx")
fu48_2<-fu48[,interestedincols] %>% distinct()

# Calculate days elapsed since reference date of January 1, 2012,
# which is a date that is earlier than any of the dates that the followup form was completed, yielding
fu48_2$`fu48-form_completion_date` - as.Date("2012-01-01")
fu48_2$daysinceref<-as.numeric(as.Date(fu48_2$`fu48-form_completion_date`)- as.Date("2012-01-01"), units
fu48_max<-fu48_2 %>% group_by(`fu48-bcr_patient_barcode`) %>% dplyr::filter(daysinceref == max(daysince

# left join all tables to to clinical.BCRtab.all2$clinical_patient_hnsc dataset
leftjoin1<-patient %>% left_join(fu48_max,
       by=c("patient-bcr_patient_barcode"="fu48-bcr_patient_barcode"))
leftjoin2<-leftjoin1 %>% left_join(radfinal,
       by=c("patient-bcr_patient_barcode"="rad-bcr_patient_barcode"))
leftjoin3<-leftjoin2 %>% left_join(drugfinal,
       by=c("patient-bcr_patient_barcode"="drug-bcr_patient_barcode"))

write.csv(leftjoin3, file="clinicaltablesmall.csv")
```

This is a visualization of the "missingness" of each column in the clinical table:

```
isna<-apply(sortedclin, 2, function(x) sum(is.na(x)))
par(mar=c(20, 4, 1, 1))
barplot(isna, las=2)
abline(h=528)
```

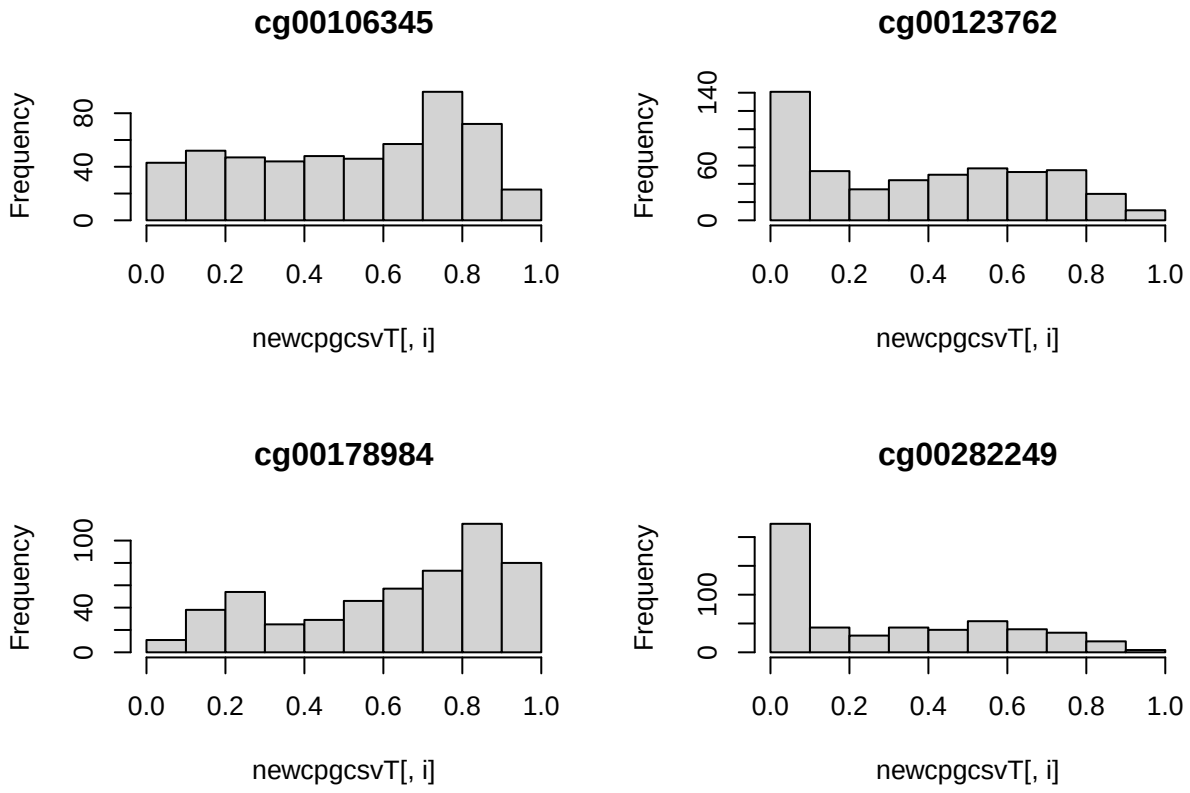


Some variables are entirely missing, and will not be used in the analysis. Since race and gender are variables commonly associated with disease severity and could be prognostic, it will be interesting to look at these and see if they coincide with any of the clusters found when we perform unsupervised clustering on the CPG table.

Sampling units of variables and distributions

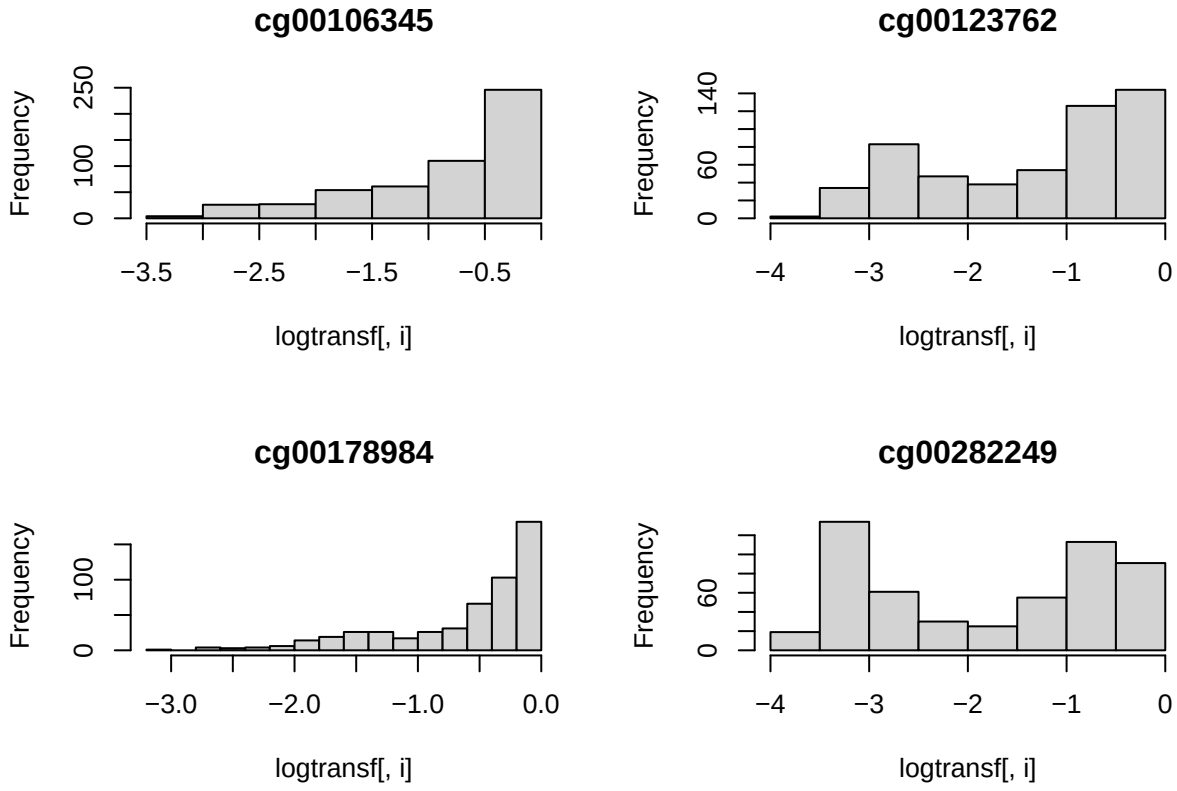
The sampling units for the CPG table are methylation beta values, which range from 0 to 1.

```
set.seed(123)
randompick<-sample(ncol(newcpgcsvT),4)

par(mfrow=c(2,2))
for (i in 1:length(randompick)){
  hist(newcpgcsvT[,i], main=colnames(newcpgcsvT)[i])
}
```



The distribution of the beta values are quite skewed, and although log transformation is generally recommended for skewed data, it does not seem to make much of a difference, and in fact, may make thinks look even more skewed:

```
logtransf<-log(newcpgcsvT)

par(mfrow=c(2,2))
for (i in 1:length(randompick)){
  hist(logtransf[,i], main=colnames(logtransf)[i])
}
```

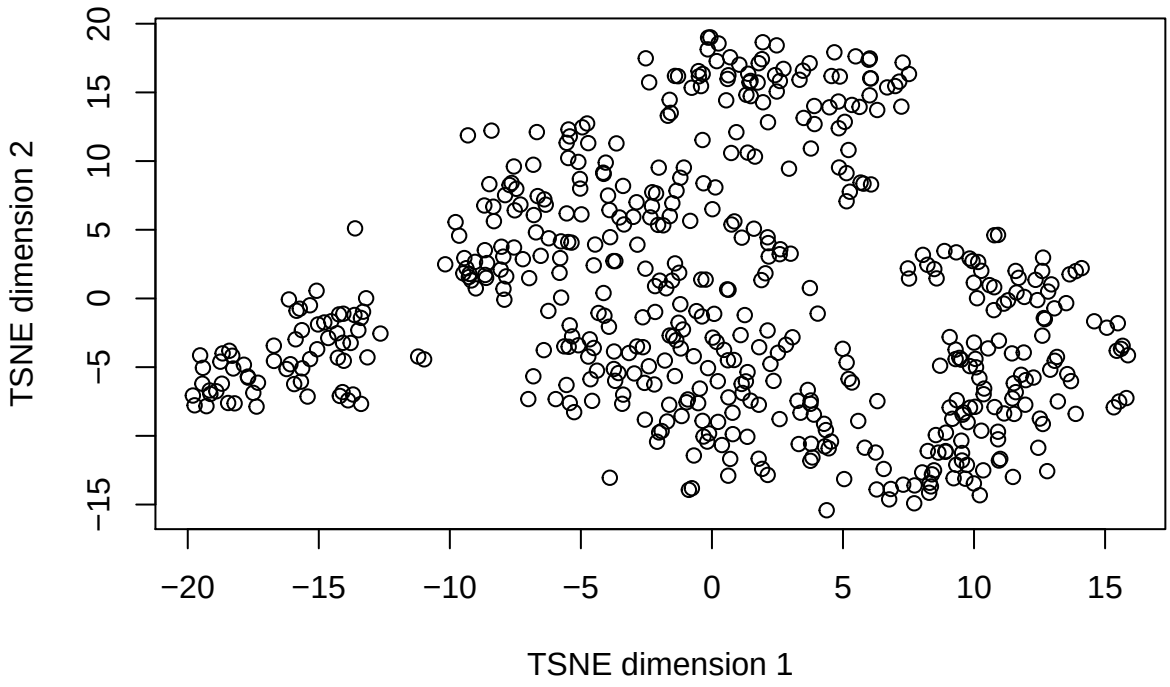


Certain clustering algorithms do perform well even with non-Gaussian data, so we could consider many clustering algorithms s.t. we are considering groups of patients with the outcome instead of individuals.

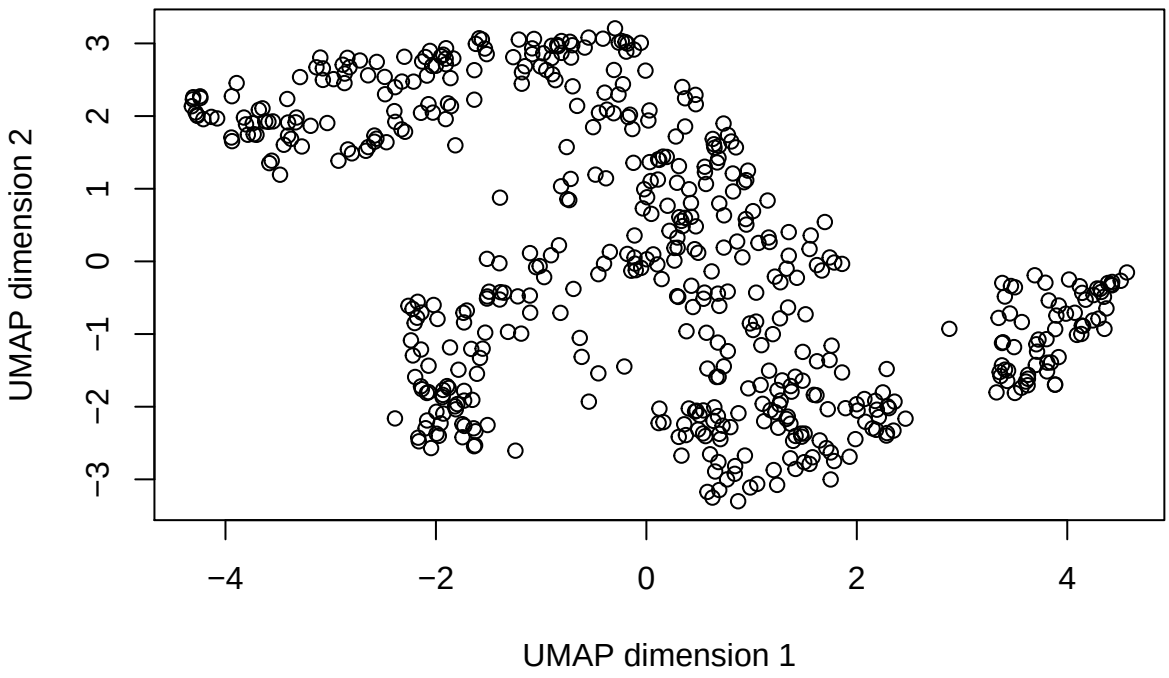
Key figures for later use in analysis

Key figures would be the plot of dimensions reduced by methods such as principal component analysis (PCA), T-distributed Stochastic Neighbor Embedding (TSNE), and Uniform Manifold APproximation (UMAP). Each dot represents a patient, and the dimensions are TSNE dimension 1 and 2. Note that traditional PCA using princomp() was attempted, but failed to do spectral decomposition on our matrix (n>p).

```
# TSNE
set.seed(123)
cpgtsne<-Rtsne(sortedcpg)
plot(cpgtsne$Y, xlab="TSNE dimension 1", ylab="TSNE dimension 2")
```




```
# UMAP
set.seed(123)
cpgumap<-umap(sortedcpg)
plot(cpgumap[["layout"]], xlab="UMAP dimension 1", ylab="UMAP dimension 2")
```



Although TSNE and UMAP are not clustering algorithm, we can visually see that some patients do cluster together. I have seen that common clustering algorithms like K-means and partioning around medioids (PAM) tends to give clusters that do not agree with what the human brain sees from plots of reduced dimension 1 vs reduced dimension 2. Some alternative clustering methods that could be successful with data like ours, would be spectral clustering and model-based clustering (mclust) to name a few.

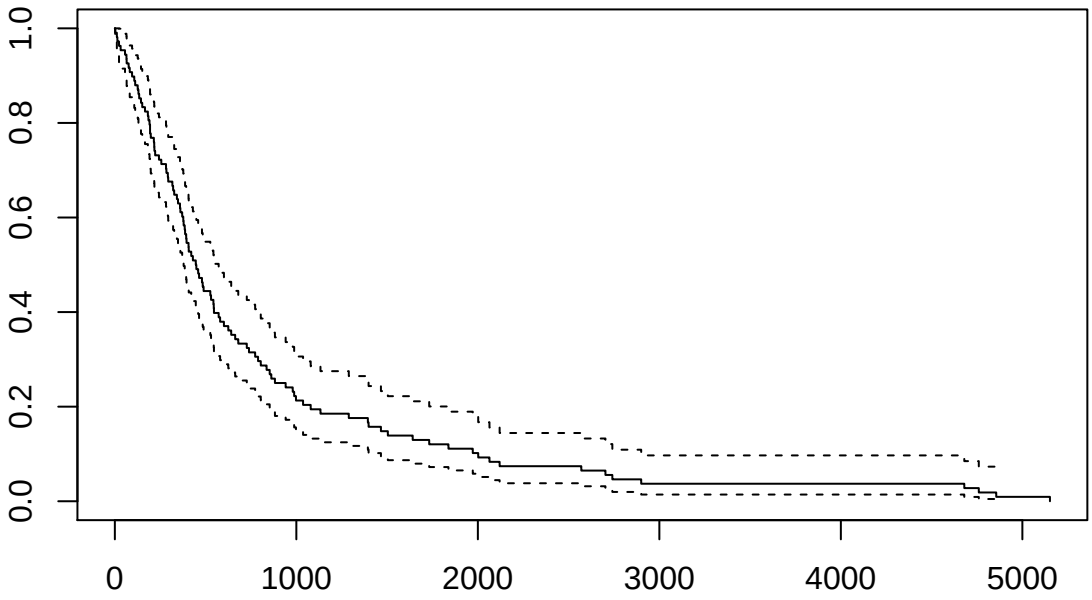
An interesting question would be: would any of these clusters be related to patient survival outcome? Here is a Kaplan-Meier curve showing the outcome data for these set of patients. Depending on what clusters we define, we can create multiple Kaplan-Meier curves corresponding to different clusters of patients and determine if the cluster is a prognostic variable. We can also dig into existing literature to compare how well our clusters defined solely by methylation data, fares as a predictive variable compared to already known prognostic factors like patient age and status of the tumor at time of diagnosis.

```
library(survival)
```

```
mysurvdf<-data.frame(event=sortedclin$fu48.vital_status %>% as.factor() %>% as.numeric() -1, #Alive=0,D
           time=sortedclin$fu48.death_days_to)

# proportion of missing data in the columns used for survival analysis
apply(mysurvdf, 2, function(x) sum(is.na(x))/nrow(mysurvdf))
```

```
##     event      time
## 0.3693182 0.7954545
```

```
plot(Surv(time=mysurvdf$time , event=mysurvdf$event))
```


0.0
0.2
0.4
0.6
0.8
1.0
0
1000
2000
3000
4000
5000