# College Scorecard: Cluster Analysis

*Michael L. Thompson*

*September 4, 2017*

## Contents

## Introduction

This is an exploratory analysis of the U.S. Dept. of Education College Scorecard database. My intent is to investigate patterns amongst the colleges as visualized using t-distributed Stochastic Neighbor Embedding (t-SNE)[1] using the R package **Rtsne**[2]. This method projects the high-dimensional data into two dimensions. From there, I can apply hierarchical clustering to identify clusters in the new 2-D space.

## Prepare Data

We read in the College Scorecard dataset and convert columns into Bayes factors, which accentuate differences amongst the colleges. Colleges having a disproportionately high number of students with a certain attribute – say, an SAT in excess of 1400 – will have highly positive Bayes factors for that attribute.

I strip out a lot of the variables that define the student body demographics. The idea is that I'd like to identify structure in the "outcome" variables – things like academic disciplines, completion rates, future earnings, credit default rates, etc. – and then later check if this structure is correlated to demographics – things like geographic location, campus setting, student ethnicity, etc.

```
glmdata_all <- DataSpec$studentBF %>%
  dplyr::select(
    c(-1, -(3:8)), -matches('_(WHITE|BLACK|ASIAN|OTHER|HISP|NRA|AIAN|UNKN)|2MOR|UNKN|NHPI|AIAN|BF_male|
```

[1] L.J.P. van der Maaten. **Accelerating t-SNE using Tree-Based Algorithms.** *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014. PDF [Supplemental material]

[2] Jesse H. Krijthe (2015). **Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation**, URL: https://github.com/jkrijthe/Rtsne

```
    -matches('Challenge|_DEP_STAT_|notvet|le24y|OUTOFSTATE|prior|(^BF_[gl][et].+[0-9]+K$)|locale|FarWes
  ) %>%
  select_if( .predicate = function(x) any(x != x[[1]]) ) %>%
  filter( complete.cases(.) )
tsne_mat_all <- glmdata_all %>% select(-College) %>% as.matrix() %>% scale()
```

## Perform t-Distributed Stochastic Neighbor Embedding (t-SNE)

Now, I'll map the data into a 2-D space using t-SNE. Hopefully, it will be easy to see clusters of colleges.

It takes a bit of trial and error (short of doing a formal hyperparameter optimization) to arrive at hyperparameters capable of generating discernible structure in a 2-D scatterplot.

```
set.seed( 173 )
tsne_all <- Rtsne( tsne_mat_all, perplexity = 10, initial_dims = 50, theta = 0.5, max_iter = 2000 )
```

### Rotate Coordinates

Now, I'll rotate the coordinates so that high-prestige colleges appear at high `Y2` coordinates. This will put most of the Ivy League colleges in the top-center of the plot.

```
# Rotate coordinates so that high-prestige colleges appear at high Y2 coordinates,
# i.e. in the top center of the plot.
i_harvard <- grep( 'Harvard', glmdata_all$College )
harvard_coord <- tsne_all$Y[i_harvard,]
harvard_angle <- atan(harvard_coord[2]/harvard_coord[1])
rotate_angle  <- pi/2 - harvard_angle
rotation_matrix <- matrix(
  c(cos(rotate_angle),sin(rotate_angle),-sin(rotate_angle),cos(rotate_angle)),
  2,2, byrow = TRUE
)
tsne_all$Y %<>% { (.) %*% rotation_matrix }
if( abs(tsne_all$Y[i_harvard,2]) < abs(tsne_all$Y[i_harvard,1]) ){
  tmp <- tsne_all$Y[,1]
  tsne_all$Y[,1] <- tsne_all$Y[,2]
  tsne_all$Y[,2] <- tmp
}
if( tsne_all$Y[i_harvard,2] < 0 ){
  tsne_all$Y[,2] <- -tsne_all$Y[,2]
}
```

I'll highlight colleges at the minimum and maximum of each of the coordinate axes and diagonals. This is done by projecting each college's coordinates onto vectors pointing into those 4 direction vectors – up, right, top-right, top-left – and finding the colleges that are at the maximum positive and negative points along those vectors.

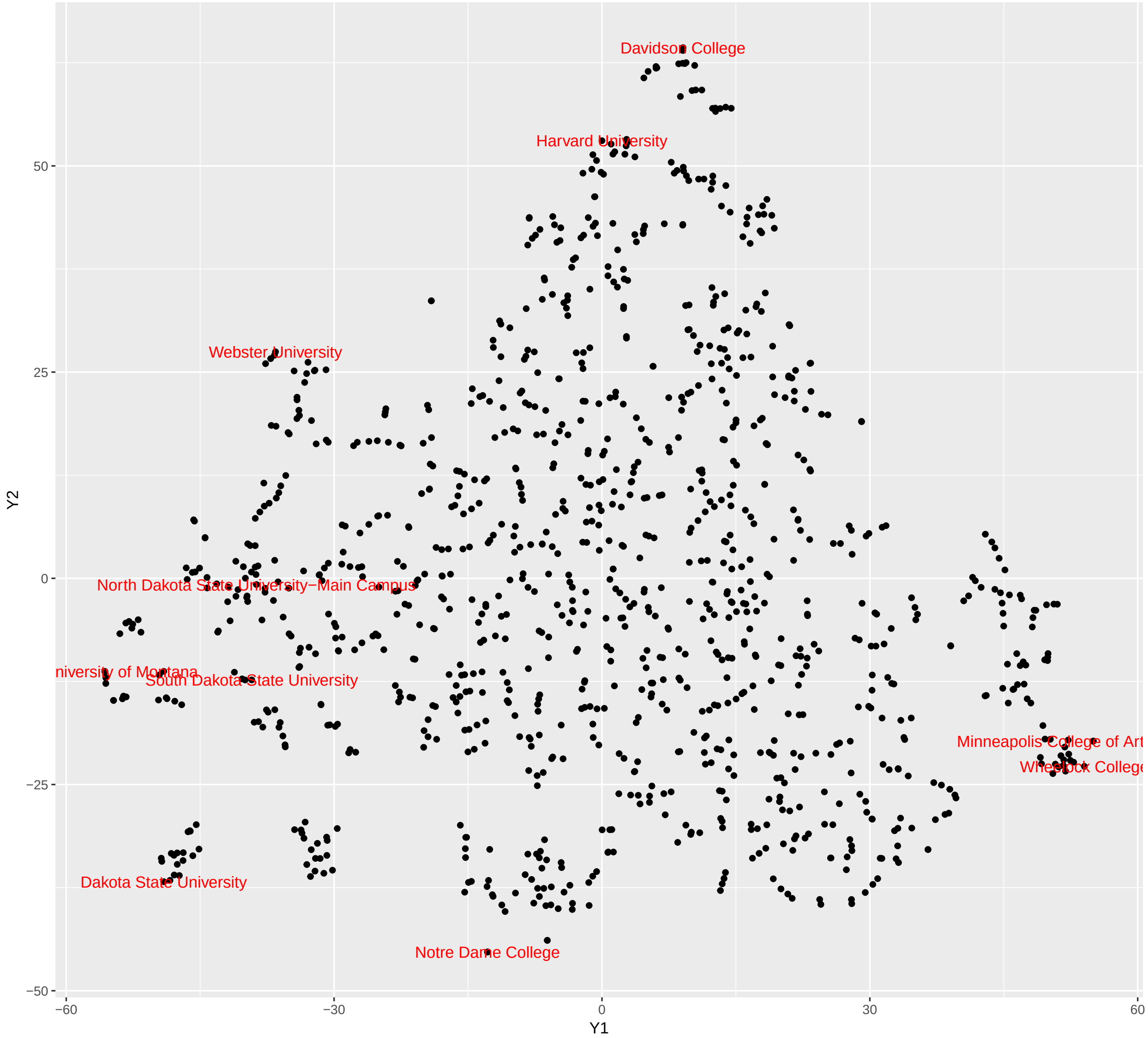
The names of those colleges at the extremes are added along with "Harvard" as names to be highlighted in the 2-D scatterplot.

```
# Project each college's coordinates along the 4 direction vectors.
prj <- tsne_all$Y %*% matrix(c(1,0,0,1,1,1,-1,1),nrow=2,ncol=4)

# Identify the colleges to be highlighted as Harvard and those at the min and max of the direction vect
highlights <- union(
```

```r
  'Harvard',
  as.character(glmdata_all$College)[c(apply(prj,2,function(x) c(which.min(x),which.max(x))))]
) %>%
  paste(collapse="|")

# Plot the 2D scatterplot with highlighted colleges labeled by the college name.
tsne_all$Y %>%
  as_tibble() %>%
  setNames(c('Y1','Y2')) %>%
  mutate( College = glmdata_all$College) %>%
  {
    ggplot(.,aes(x=Y1,y=Y2)) +
      geom_point() +
      geom_text(
        data    = (.) %>% filter( grepl(highlights,College) ),
        mapping = aes( label = College ),
        color   = 'red',
        size    = 4
      )
  } %>%
  print()
```



### Find Underlying Factors Driving 2-D Structure

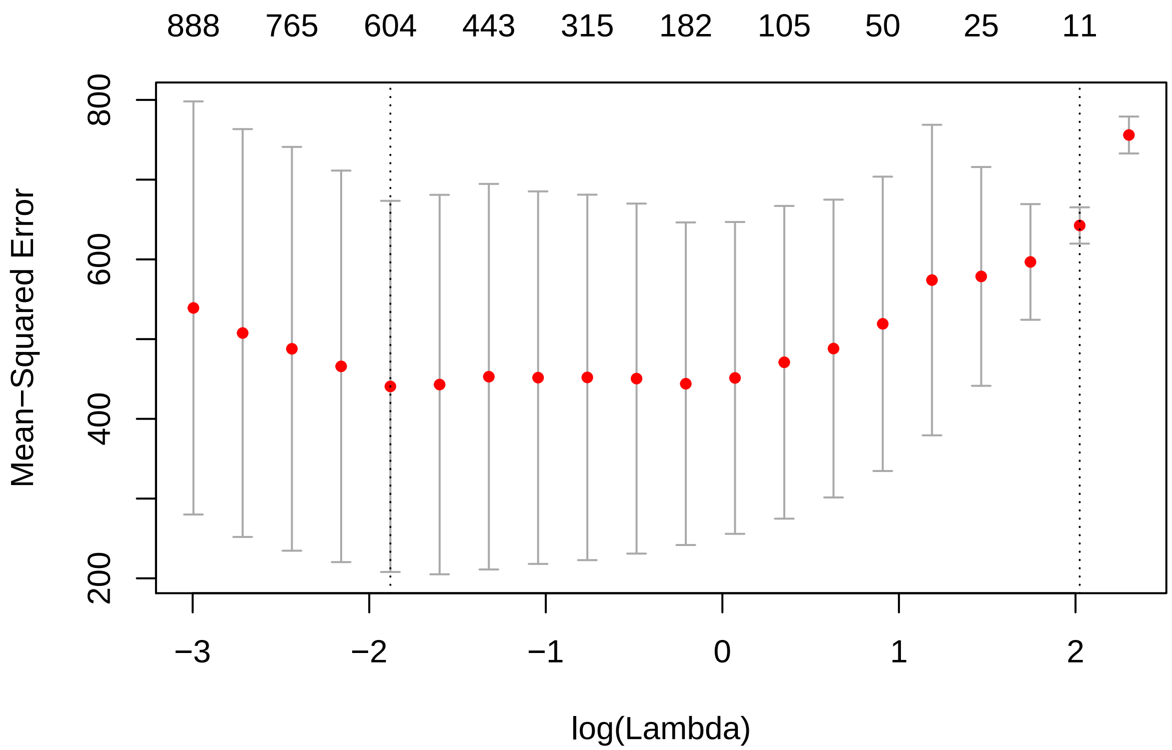
Using R package **glmnet**[3], I perform regularization (variable selection) in modeling of the 2-D t-SNE coordinates as responses vs. the original college Bayes factor features from which the t-SNE coordinates were found. This way we'll have a linear model showing which features contributed to which coordinate. As such, we'll have the basis for plotting a biplot of colleges overlayed on feature dimensions in 2-D, analogous to a PCA biplot.

```
mmat <- model.matrix( ~ .:. - 1, as.data.frame(tsne_mat_all))
# b <- eigen(cor(mmat))
# mmat <- mmat[,apply(b$vectors[,1:200],2,function(x) which.max(abs(x))) %>% unique() %>% sort()]
```

```
set.seed( 2393 )
tsne_glmnet_all <- cv.glmnet(
  x       = mmat,
```

[3] Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). **Regularization Paths for Generalized Linear Models via Coordinate Descent.** *Journal of Statistical Software*, 33(1), 1-22. URL http://www.jstatsoft.org/v33/i01/.

```
  y      = tsne_all$Y,
  family = 'mgaussian',
  lambda = exp(seq(log(0.05),log(10),length.out = 20))
)
plot( tsne_glmnet_all )
```



### Check the Predictions

It can be tricky to find a subset of features and their interactions that both describe the t-SNE coordinates well *and* do not suffer from extreme collinearity, which can make the validation error at low `lambda` explode when applying function `cv.glmnet()`.

Judging from the cross-validation curve above and the observed vs. predicted plots below, it looks like we've got a decent model.

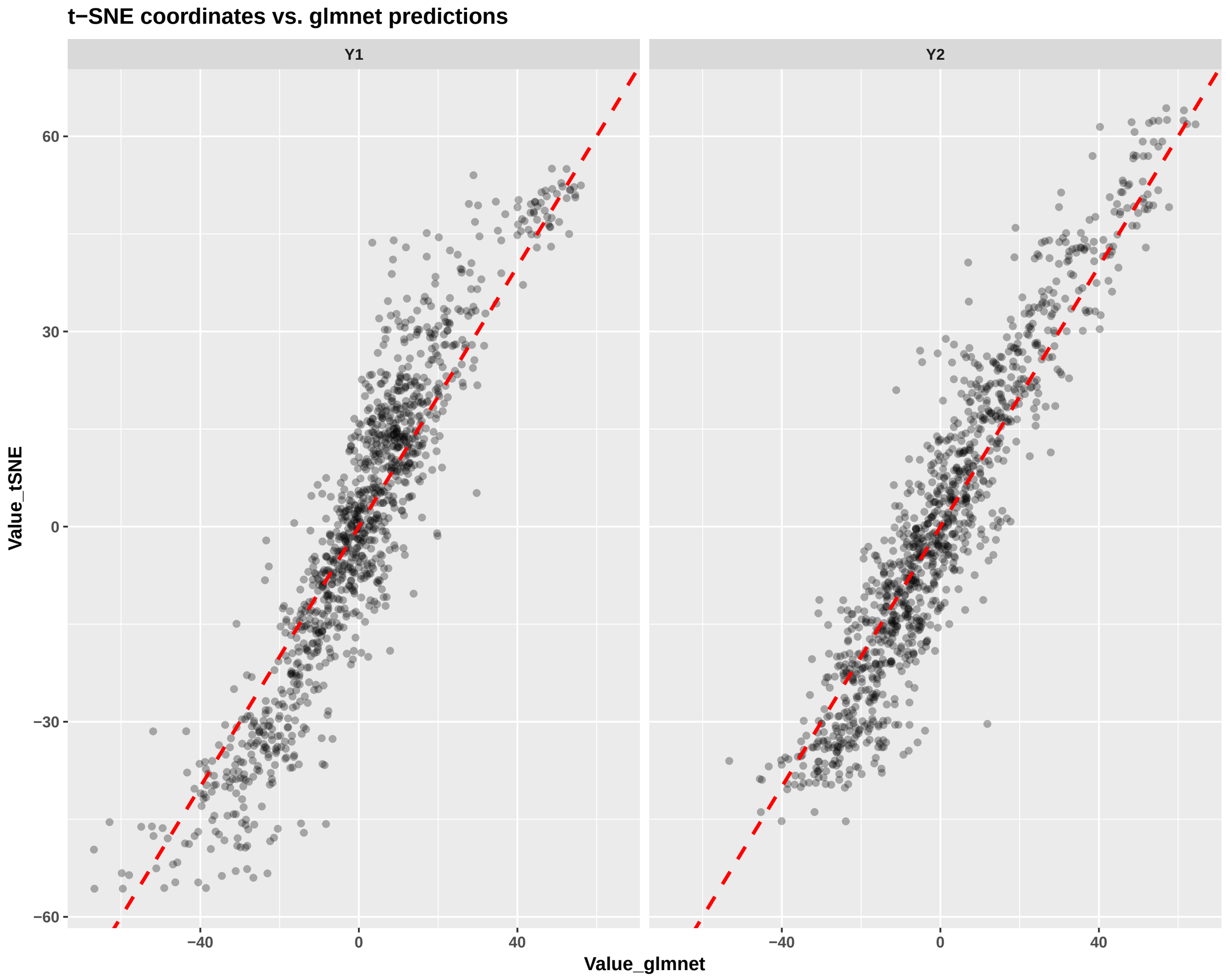
```
# Get glmnet predictions of the t-SNE coordinates, combine them with the original t-SNE coordinates,
# and plot the originals vs. predictions.
lambda <- tsne_glmnet_all %$% { exp( mean(log(c(lambda.min,lambda.1se))) ) } # mid lambda

pred_df <- tsne_glmnet_all %>%
  predict( newx = mmat, s = lambda ) %>%
  drop() %>%
  as_data_frame() %>%
  setNames( c( "Y1", "Y2" ) ) %>%
  mutate( rowid = 1:nrow(.) )
```

```r
tsne_df <- tsne_all$Y %>%
  as_data_frame() %>%
  setNames( c( "Y1", "Y2" ) ) %>%
  mutate( rowid = 1:nrow(.) ) %>%
  gather( key = Coordinate, value = Value , -rowid )

combo_df <- pred_df %>%
  gather( key = Coordinate, value = Value , -rowid ) %>%
  left_join( tsne_df, by = c('Coordinate','rowid'), suffix = c( "_glmnet","_tSNE" ) )

combo_df %>%
{
  ggplot(., aes(x = Value_glmnet, y = Value_tSNE ) ) +
    geom_point( alpha = 0.3 ) +
    geom_abline( intercept = 0, slope = 1, color = 'red', linetype = 2, size = 1 ) +
    facet_wrap( ~ Coordinate ) +
    ggtitle( "t-SNE coordinates vs. glmnet predictions" ) +
    theme( text = element_text( face = 'bold' ) )
}
```



```r
rm( pred_df, tsne_df, combo_df )
```

## Visualize the Colleges in 2-D

```r
tsne_glmnet_coef_all <- tsne_glmnet_all %>% coef( s = lambda )
# tsne_glmnet_coef_all$y1[-1] %>%
# { (.)[abs((.)[,1])>0,1] } %>%
# { data_frame(Coefficient = names(.), value = round(.,2)) } %>%
#   print()
# tsne_glmnet_coef_all$y2[-1] %>%
# { (.)[abs((.)[,1])>0,1] } %>%
# { data_frame(Coefficient = names(.), value = round(.,2)) } %>%
#   print()

tsne_coef_df_all <-
  tsne_glmnet_coef_all$y1 %>%
  as.matrix() %>%
  as.data.frame() %>%
  as_tibble() %>%
  rownames_to_column() %>%
  setNames(c("Coefficient","Y1")) %>%
  full_join(
    tsne_glmnet_coef_all$y2 %>%
      as.matrix() %>%
      as.data.frame() %>%
      as_tibble() %>%
      rownames_to_column() %>%
      setNames(c("Coefficient","Y2")),
    by = "Coefficient"
  ) %>%
  filter( abs(Y1) > 1.0E-9 | abs(Y2) > 1.0E-9 ) %>% slice(-1) %>%
  mutate(
    # Flip direction of interactions
    Y1 = ifelse( grepl(':',Coefficient), -Y1 , Y1 ),
    Y2 = ifelse( grepl(':',Coefficient), -Y2 , Y2 )
  )

tsne_coef_df_all %>% mutate(mag = sqrt(Y1^2+Y2^2)) %>% arrange(desc(mag)) %>% print(n = 30)
```

```
## # A tibble: 138 x 4
##                                           Coefficient         Y1
##                                                 <chr>      <dbl>
##  1                             BF_ScienceTechnologies -3.9812050
##  2                                BF_ForeignLanguages -2.2302846
##  3            BF_fsend_1_2005:BF_MilitaryTechnologies -2.0887989
##  4                                     BF_discBreadth -3.5830984
##  5                                      BF_SAT_gt1400  0.5614495
##  6                          BF_AgricultureAgriculture -2.8681726
##  7                                BF_PhysicalSciences -1.8053768
##  8                                BF_PersonalCulinary -2.3148902
##  9                                    BF_fsend_5_2005  1.2762902
## 10               BF_SAT_gt800le1000:BF_MechanicRepair  2.1766514
## 11                      BF_CommunicationsTechnologies -2.4546770
## 12                                         BF_veteran -1.9400272
## 13                                  BF_pell_ever_2005 -1.0141942
## 14                                BF_VisualPerforming -2.3261339
## 15                                         BF_CDR3est -0.3818700
## 16                                      BF_AreaEthnic -1.2574297
## 17                           BF_MathematicsStatistics -1.5306761
## 18        BF_RPY_7YR_RT:BF_CommunicationsTechnologies  0.8409629
## 19                             BF_ArchitectureRelated -1.6625438
## 20                                  BF_MechanicRepair -1.6748699
## 21                             BF_PhilosophyReligious -0.5828815
## 22                         BF_EngineeringTechnologies -1.3986440
## 23                             BF_ComputerInformation -1.4255263
## 24                                 BF_EnglishLanguage -1.3712745
## 25                                BF_HomelandSecurity -1.0268479
## 26         BF_CommunicationsTechnologies:BF_Education  1.1930009
## 27                               BF_HealthProfessions -1.0598328
## 28 BF_CommunicationsTechnologies:BF_TheologyReligious  0.9991621
## 29        BF_RPY_5YR_RT:BF_CommunicationsTechnologies  0.8204905
## 30    BF_C150_4_POOLED_SUPP:BF_AgricultureAgriculture  0.7301438
## # ... with 108 more rows, and 2 more variables: Y2 <dbl>, mag <dbl>
```

```
# tsne_coef_df %>%
# {
#   ggplot(., aes(x=Y1,y=Y2,label=Coefficient)) +
#     geom_point() +
#     geom_text( check_overlap = TRUE )
# } %>%
#   print()

key_terms <- tsne_coef_df_all %>%
  mutate(mag= sqrt(Y1^2+Y2^2)) %>%
  filter(abs(mag)>quantile(abs(mag),0.9)) %>%
  arrange(desc(mag)) %$% Coefficient %>% setdiff("(Intercept)")

# Abbreviate names so they don't clutter the plots so much.
shorten_names <- function( df ){
  college_names <- df %$%
    College %>%
    { gsub('^[0-9_]+','',. ) } %>%
    { gsub('Northwestern University','NU',.) } %>%
    { gsub('University of Notre Dame','Notre Dame U.',.) } %>%
    { gsub('Cornell College','Cornell C',.) } %>%
    { gsub('Cornell University','Cornell U',.) } %>%
    { gsub('California','Cal',. ) } %>%
    { gsub('Mass.+Inst.+Tech.+','MIT',. ) } %>%
    { gsub('(Mass|Penn|Wash)[^ ]+ *','\\1',.) } %>%
    { gsub('Polytechnic','Poly',. ) } %>%
    { gsub('Institute of Tech[^ ]+','IT',. ) } %>%
    { gsub('Tech.+Inst.+','Tech',. ) } %>%
    { gsub('State','St',. ) } %>%
    { gsub('University','U',. ) } %>%
    { gsub('(U of )|( U$)','',. ) } %>%
    { gsub('College','Col',. ) } %>%
    { gsub('New York','NY',.)} %>%
    { gsub('International','Intl',.) } %>%
    { gsub('North[^ ]+','N',.)} %>%
    { gsub('South[^ ]+','S',.)} %>%
    { gsub('West[^ ]+','W',.)} %>%
    { gsub('East[^ ]+','E',.)} %>%
    { gsub(' U-','-',.)} %>%
    { gsub('-Penn St ','',.)} %>%
    { gsub(' Col *$','',.)} %>%
    { gsub('-(Main)* Campus','',.)} %>%
    { gsub('^PennSt([^-]+)$','Penn St-\\1',.)} %>%
    { gsub(' and ','&',.)} %>%
    { gsub('Agricultural & Mechanical','A&M',.)}

  st_abb <- state.abb %>% setNames( state.name )
  for( st_nm in names(st_abb) ){
    college_names %<>% { gsub(st_nm,st_abb[st_nm],.) }
  }
  return( college_names )
}

college_names <- shorten_names( glmdata_all )
college_names_student <- shorten_names( DataSpec$student)

categories <- {
  mmat[,key_terms] %*%
    (tsne_coef_df_all %>% filter(Coefficient %in% key_terms) %$% Y2)
} %>%
  sapply(
    function(x,q){ length(q) - sum(x>q) + 1 },
    q=quantile(.,c(0.1,0.25,0.75,0.9))
  ) %>%
  factor()

tsne_df_all <- tsne_all$Y %>%
  as_tibble() %>%
  setNames(c("Y1","Y2")) %>%
  mutate(
    College = college_names,
    category = categories,
    BF_Income_gt110K = glmdata_all %$% {10.0^BF_p_gt110K}
  ) %>%
  dplyr::select( College, category, BF_Income_gt110K, everything() ) %>%
  mutate_at(funs(scale(.)),.vars=vars(Y1,Y2))
```

### Show Biplot for Structure Interpretation

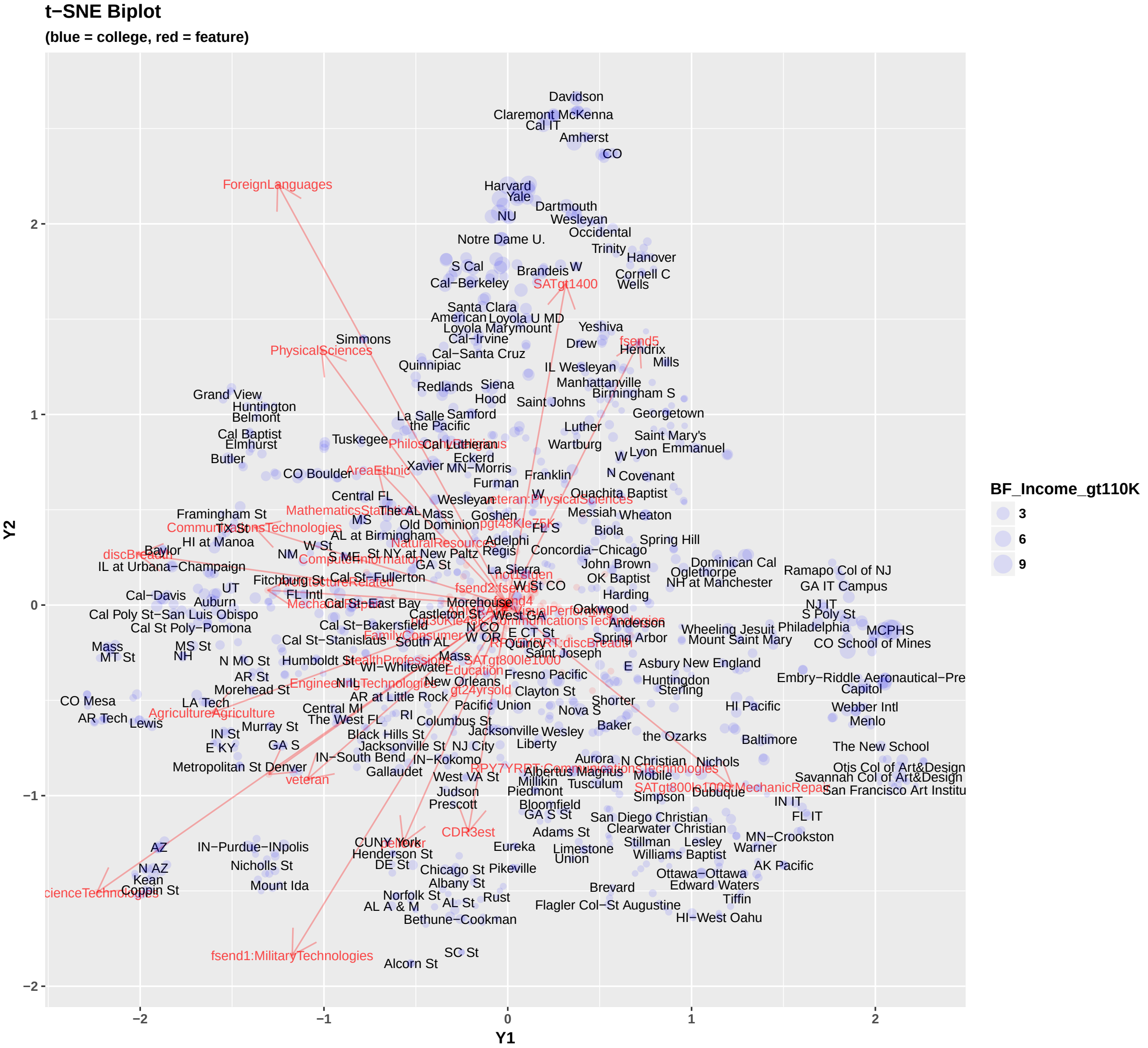
We can overlay the feature dimensions on the college scatterplot in the 2-D t-SNE coordinate space. This allows us to more easily interpret the structure we're seeing.

However, some of the interaction terms, in particular, are tricky to interpret because they have a positive value for a college if both of the features in the product making up the interaction have the same sign. So it could be that the college has a disproportionately higher *or* lower number of students having the attributes of *both* of the corresponding features.

By plotting the College points sized by their Bayes factor on incomes greater than $110,000, we can see where the colleges lie that have disproportionately high/low proportions of high-income students.

```r
# scale factor for coefficients:
f_mult <-
  max(sqrt(tsne_df_all$Y1^2 + tsne_df_all$Y2^2))/
  max(sqrt(tsne_coef_df_all$Y1[-1]^2 + tsne_coef_df_all$Y2[-1]^2))

y2_min <- -3.5
tsne_coef_df_all %>%
  mutate(
    Y2 = pmax(y2_min,Y2*f_mult),
    Y1 = Y1*f_mult,
    mag = sqrt(Y1^2 + Y2^2),
    Coefficient = gsub('\\([^)]+\\)|(_*2005)|_','',gsub('BF_','',Coefficient))
  ) %>%
  {
    ggplot(., aes( x = Y1, y = Y2 ) ) +
      geom_point( color = 'red', alpha = 0.1 ) +
      # Labels for the coefficients
      geom_text(
        aes( label = Coefficient),
        color = 'red',
        alpha = 0.7,
        size = 3,
        check_overlap = TRUE
      ) +
      # Rays on the coefficients
      geom_segment(
        inherit.aes = FALSE,
        data = (.) %>% filter(mag>1),
        aes( x=0, y=0, xend=Y1, yend=Y2 ),
        color = 'red',
        alpha = 0.3,
        arrow = arrow(length = unit(0.03, "npc"))
      ) +
      # Labels for the Colleges
      geom_text(
        inherit.aes = FALSE,
        data = tsne_df_all,
        aes( x=Y1, y=Y2, label=College ),
        mapping=,
        color = 'black',
        size=3,
        check_overlap = TRUE
      )  +
      # Points for the colleges
      geom_point( data=tsne_df_all, aes(x=Y1,y=Y2, size = BF_Income_gt110K ), color='blue',alpha=0.1) +
      ggtitle( "t-SNE Biplot" , subtitle = "(blue = college, red = feature)") +
      theme( text = element_text( face = 'bold' ) ) #+
      #scale_y_continuous(limits = c(y2_min,4))
    #scale_y_continuous(limits = c(y2_min,4))
  } %>%
print()
```



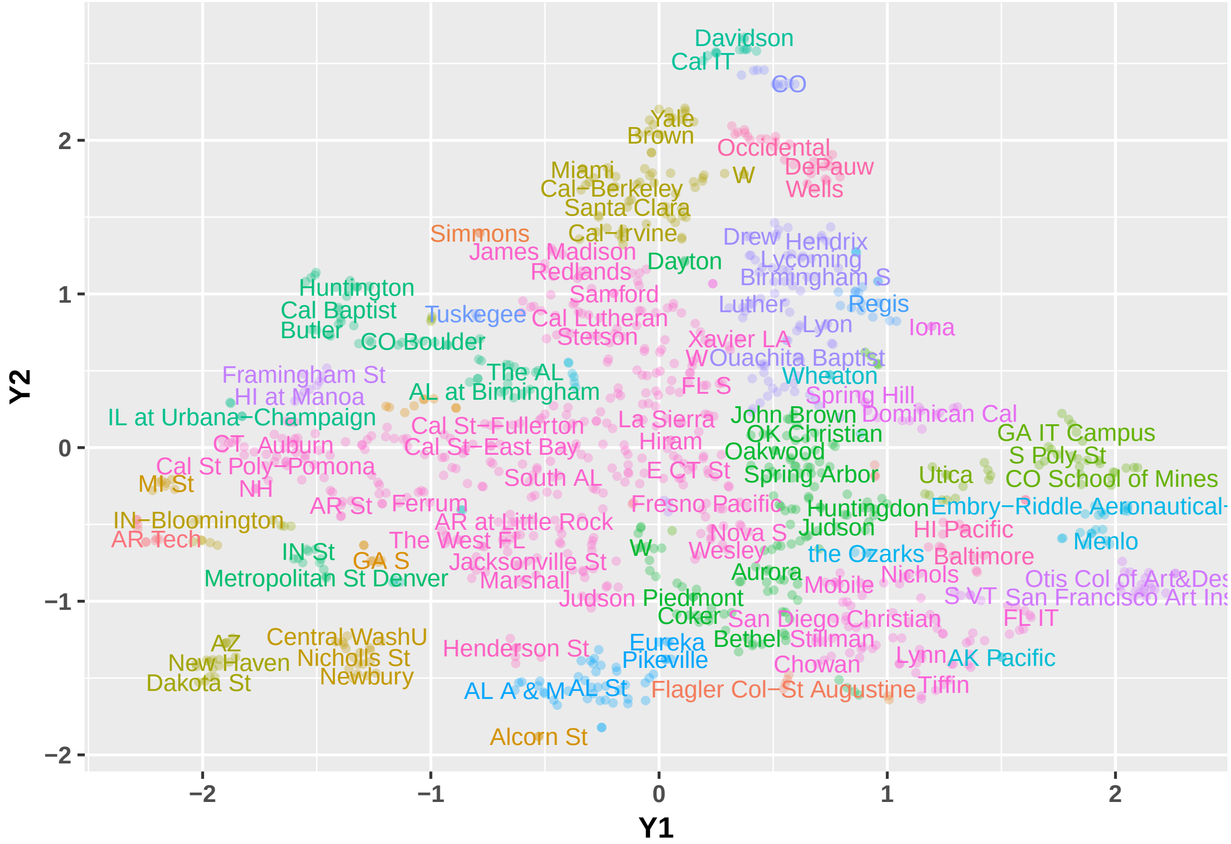
### Perform Hierarchical Clustering

Now, I perform cluster analysis. Hierarchical clustering is a quick way to identify clusters in the 2-D t-SNE space. We can then color the clusterings in a scatterplot to more easily visualize the structure.

```
tsne_mat_hc_all <- tsne_df_all %>% select(Y1,Y2) %>% as.matrix() %>% set_rownames(tsne_df_all$College)
hc_all <- hclust( d = dist( tsne_mat_hc_all ), method = 'single' )
n_cluster <- 55
cluster_id_all <- cutree( hc_all, k = n_cluster )

# plot( tsne_mat_hc, pch=20, cex=0.5 )
# for(j in seq_along(cl)){
#   points( tsne_mat_hc[ cl[[j]], ], pch=20, col=j, cex=1)
# }
# randomize so adjacent clusters are more likely to have very different colors.
set.seed(137)
cluster_id_all <- setNames( sample.int(n_cluster)[cluster_id_all], names(cluster_id_all) )
```

```
tsne_mat_hc_all %>%
  as_tibble() %>%
  mutate( College = names(cluster_id_all), cluster = factor( cluster_id_all ) ) %>%
  {
    ggplot(.,aes( x = Y1, y = Y2, color = cluster ) ) +
      geom_point( size = 1, alpha = 0.3 ) +
      geom_text( aes(label = College ), size = 3, check_overlap = TRUE ) +
      theme(
        text = element_text( face = 'bold' ),
        legend.position = 'none'
      )
  } %>%
  print()
```



### Show Biplot with Cluster Coloring

Finally, we can overlay the feature dimensions on the 2-D

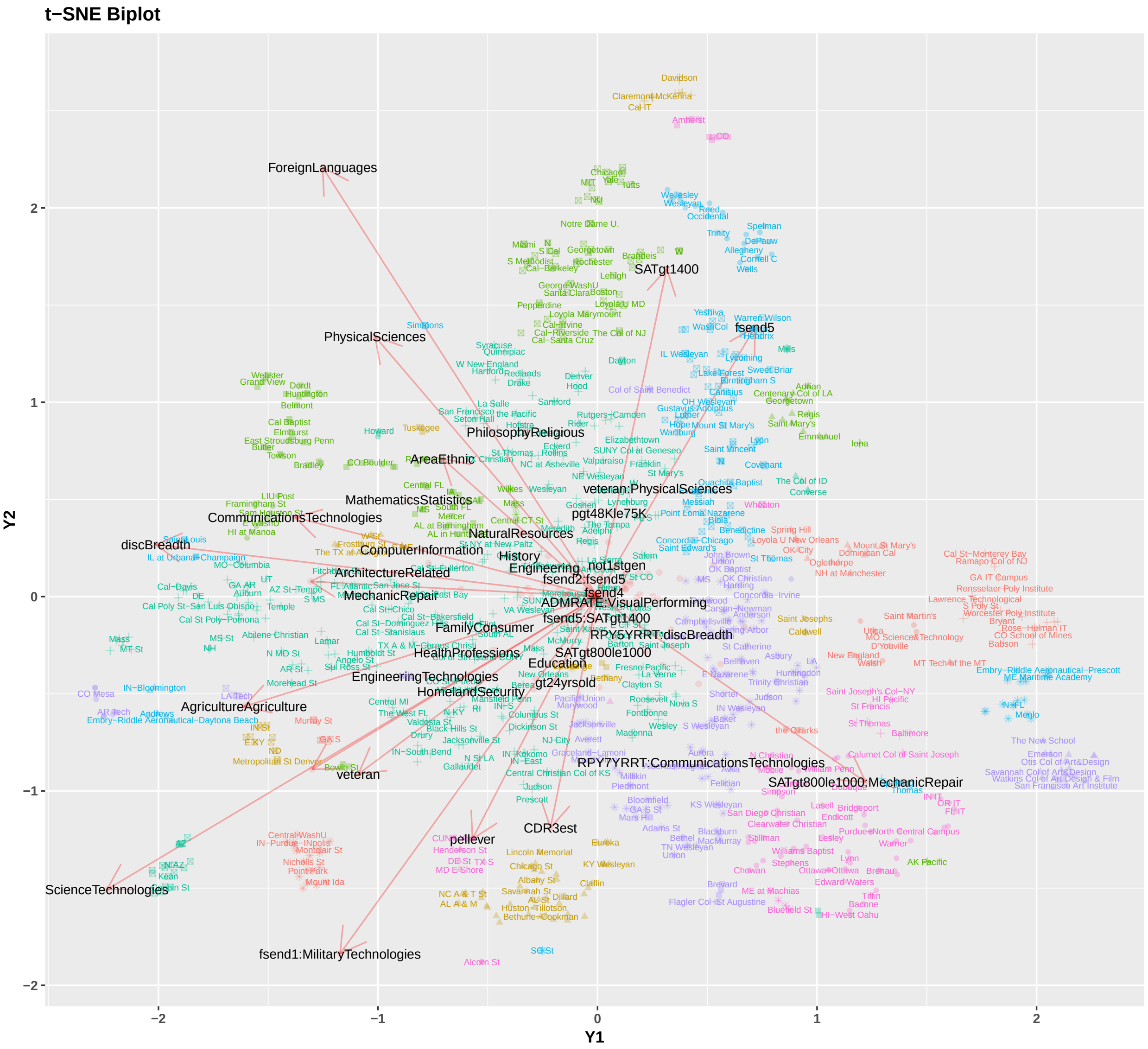
```
cluster_id_all <- cutree( hc_all, k = n_cluster )
y2_min <- -4
y2_max <- 3.49
y1 <- range(tsne_mat_hc_all[,1])
```

```r
y1[1] <- 0.5*floor(y1[1]/0.5)
y1[2] <- 0.5*ceiling(y1[2]/0.5)
y2 <- range(tsne_mat_hc_all[,2])
y2[1] <- 0.5*floor(y2[1]/0.5)
y2[2] <- 0.5*ceiling(y2[2]/0.5)

is_out_of_bounds <- function(x,bounds){ x<bounds[1] | x>bounds[2] }
# Assumes that value violating bounds is of same sign as bound violated AND that bounds are of opposite
bound_factor <- function(x,bounds){
  f1 <- ifelse(x<bounds[1],x/bounds[1],0)
  f2 <- ifelse(x>bounds[2],x/bounds[2],0)
  mapply(function(b1,b2) if(b1>b2) c(1,b1) else c(2,b2),f1,f2)
}
tsne_modified <- tsne_coef_df_all %>%
  mutate(
    Coefficient = gsub('\\([^)]+\\)|(_*2005)|_','',gsub('BF_','',Coefficient)),
    Y1  = f_mult*Y1,
    Y2  = f_mult*Y2 ,
    mag = sqrt(Y1^2 + Y2^2)
  )

# check bounds to find if any violated
bchk1 <- bound_factor(tsne_modified$Y1,y1)
bchk2 <- bound_factor(tsne_modified$Y2,y2)
# bound on Y1 violated
w1 <- which(bchk1[2,] != 0)
# bound on Y2 violated
w2 <- which(bchk2[2,] != 0)
# Keep only coord Y1 or Y2 violated the most by each violating pt.
for( i in intersect(w1,w2)) { if(bchk1[2,i]>bchk2[2,i]) w2<-setdiff(w2,i) else w1<- setdiff(w1,i) }
# bound on Y1 violated: fix it
for( i in w1 ){
  tsne_modified$Y2[i] <- tsne_modified$Y2[i]*y1[bchk1[1,i]]/tsne_modified$Y1[i]
  tsne_modified$Y1[i] <- y1[bchk1[1,i]]
}
# bound on Y2 violated: fix it
for( i in w2 ){
  tsne_modified$Y1[i] <- tsne_modified$Y1[i]*y2[bchk2[1,i]]/tsne_modified$Y2[i]
  tsne_modified$Y2[i] <- y2[bchk2[1,i]]
}

tsne_modified %>%
  {
    ggplot(., aes( x = Y1, y = Y2 ) ) +
      geom_point( color = 'red', alpha = 0.1 ) +
      geom_segment(
        inherit.aes = FALSE,
        data = (.) %>% filter(mag>1),
        aes( x=0, y=0, xend=Y1, yend=Y2 ),
        color = 'red',
        alpha = 0.3,
        arrow = arrow(length = unit(0.03, "npc"))
      ) +
      geom_text(
        inherit.aes = FALSE,
        data = tsne_mat_hc_all %>%
          as_tibble() %>%
          mutate(
            College = names(cluster_id_all),
            cluster = factor( (cluster_id_all %% 7) + 1 )
          ),
        aes( x=Y1, y=Y2, label=College, color = cluster ),
        mapping=,
        show.legend = FALSE,
        size=2,
        check_overlap = TRUE
      ) +
      geom_text(
        aes( label = Coefficient ),
        color = 'black',
        size = 3,
        check_overlap = TRUE
      )  +
      geom_point(
        data = tsne_mat_hc_all %>%
          as_tibble() %>%
          mutate(
            College = names(cluster_id_all),
            cluster = factor( (cluster_id_all %% 7) + 1 ),
            cluster_shape = factor( (cluster_id_all %% 6) + 1 )
          ),
        aes(x=Y1,y=Y2, color = cluster, shape = cluster_shape ),
        show.legend = FALSE,
        alpha=0.3
      ) +
      ggtitle( "t-SNE Biplot" ) +
      theme( text = element_text( face = 'bold' ) ) #+
      #scale_y_continuous(limits = c(y2_min,5))
  } %>%
  print()
```



## Conclusions

We do find some structure in the plot. And, the rotation of the axis to put Harvard University at the top-center helps us to interpret the axes and give meaning to that structure.

### Notable Colleges

Clusters are colored with repeating colors and marked with repeating symbols, reflecting a limit of **ggplot2**. But each cluster should have an unique color-symbol combination.

Here are the t-SNE 2-D coordinates for some notable universities:

```
select_colleges <- c(
  '^OH St', '^MI-Ann Arbor', '^Purdue$', '^NU$','Harvard',
  'Yale', 'Princeton','^Penn$','^Cornell$','^Brown$',
  '^Howard$','Tuskegee','Hampton','Morehouse','Grambling',
  'Bethune-Cookman','Stanford','Johns Hopkins','Duke','Vanderbilt',
  'Rice','Wash.+St Louis','Notre Dame U\\.','^Pomona$','Harvey Mudd',
  'Swarthmore','MIT','Cal *IT','WI-Madison','IN-Bloomington',
  'Dartmouth',"Otis Col of Art&Design","San Francisco Art Institute",
  "Watkins Col of Art Design & Film","Rose-Hulman IT",
  "Worcester Poly Institute","GA IT Campus","Davidson"
)

names( select_colleges ) <-
  c(
    "Ohio State","Michigan","Purdue","Northwestern",
    "Harvard","Yale","Princeton","Penn","Cornell","Brown",
    "Howard","Tuskegee","Hampton Inst","Morehouse","Grambling","Bethune-Cookman",
    "Stanford","Johns Hopkins","Duke","Vanderbilt","Rice","Wash.U.-St.L.",
    "Notre Dame","Pomona","Harvey Mudd","Swarthmore",
    "MIT","CalTech","Wisconsin","Indiana","Dartmouth",
    "Otis Col of Art&Design","San Francisco Art Institute","Watkins Col of Art Design & Film",
    "Rose-Hulman IT","Worcester Poly Institute","Georgia Tech","Davidson"
  )

rowid_select <- sapply( select_colleges, function(nm_regex) grep(nm_regex,tsne_df_all$College) )

sat_ugds_select <- DataSpec$student %>%
  slice( sapply( select_colleges, function(nm_regex) grep(nm_regex,college_names_student) ) ) %>%
  dplyr::select(1:2,UGDS,SAT_AVG,pctDisc1,pctDisc2,C150_4_POOLED_SUPP,CDR3,median_hh_inc_2005,pell_ever_
  mutate(
    UGDS = prettyNum( UGDS, big.mark = "," ),
    SAT_AVG = round(SAT_AVG),
    median_hh_inc_2005 = prettyNum(100*round(median_hh_inc_2005/100),big.mark=","),
    pctDisc_top2 = round(pctDisc1+pctDisc2),
    cluster = cluster_id_all[rowid_select]
  ) %>%
  dplyr::select(1:2,pctDisc_top2,everything(),-pctDisc1,-pctDisc2) %>%
  left_join(
    DataSpec$studentBF %>%
      dplyr::select(unitID,BF_discBreadth,BF_SAT_gt1400,BF_not1stgen,BF_fsend_5_2005,BF_CDR3),
    by = "unitID"
  )

tsne_select <- tsne_df_all %>%
  slice( rowid_select ) %$%
  set_rownames(as.matrix(select(.,Y1,Y2)),College) %>%
  round(1)
```

| Group | College | Y1 | Y2 | SAT avg. | Cluster |
|---|---|---|---|---|---|
| **Ivy League** | Harvard | 0 | 2.2 | 1501 | 16 |
| | Yale | 0.1 | 2.1 | 1497 | 16 |
| | Penn | 0 | 2.1 | 1442 | 16 |
| | Princeton | 0.4 | 2.4 | 1495 | 20 |
| | Dartmouth | 0.3 | 2.1 | 1446 | 18 |
| | Brown | 0 | 2 | 1425 | 16 |
| | Cornell | -0.1 | 2 | 1422 | 16 |
| **Big 10** | Ohio State | -1.8 | 0.2 | 1289 | 32 |
| | Wisconsin | -1.7 | 0.1 | 1268 | 3 |

| Group | College | Y1 | Y2 | SAT avg. | Cluster |
|---:|---:|---:|:--|---:|---:|
| | Purdue | -2 | -0.6 | 1211 | 25 |
| | Indiana | -2 | -0.5 | 1198 | 25 |
| | Michigan | -0.3 | 1.8 | 1352 | 16 |
| | Northwestern | 0 | 2 | 1458 | 16 |
| **HBCUs** | Howard | -1 | 0.9 | 1081 | 24 |
| | Tuskegee | -0.8 | 0.9 | 937 | 8 |
| | Hampton Inst | 0 | -0.7 | 990 | 5 |
| | Morehouse | -0.2 | 0 | 990 | 3 |
| | Grambling | -0.4 | -1.6 | 863 | 1 |
| | Bethune-Cookman | -0.3 | -1.6 | 812 | 1 |
| **Arts Specialty** | SF Art Inst | 2.1 | -1 | 1061 | 19 |
| | Otis C Art&Des | 2.1 | -0.8 | 1002 | 19 |
| | Watkins Art,Des,Film | 2.1 | -0.9 | 971 | 19 |
| **Tech Specialty** | Rose-Hullman | 2 | -0.2 | 1310 | 21 |
| | Georgia Tech | 1.8 | 0.1 | 1352 | 21 |
| | WPI | 1.9 | -0.1 | 1256 | 21 |
| **Others** | Stanford | 0.1 | 2.2 | 1466 | 16 |
| | MIT | 0 | 2.1 | 1503 | 16 |
| | CalTech | 0.2 | 2.5 | 1534 | 15 |
| | Johns Hopkins | -0.1 | 1.8 | 1418 | 16 |
| | Duke | 0.1 | 2.2 | 1444 | 16 |
| | Vanderbilt | 0.1 | 2.2 | 1475 | 16 |
| | Rice | 0.1 | 2.1 | 1454 | 16 |
| | Wash.U.-St.L. | 0 | 2.1 | 1474 | 16 |
| | Notre Dame | 0 | 1.9 | 1450 | 16 |
| | Pomona | 0.3 | 2.6 | 1454 | 15 |
| | Harvey Mudd | 0.2 | 2.6 | 1483 | 15 |
| | Swarthmore | 0.2 | 2.6 | 1442 | 15 |
| | Davidson | 0.4 | 2.7 | 1353 | 15 |

### Interpretation of Quadrants

The combination of cluster locations and Bayes factors feature rays helps us assign meaning to each quadrant of the biplot.

### Elite private & top-academic public, wealthy & smart

The vertical `Y2` axis is now almost perfectly aligned with the ray `pgt110K`, which is the ($\log_{10}$) Bayes factor capturing the prevalance of students from families with annual incomes greater than $110,000. All the Ivy League, “Ivy wannabes”, and top-academic public universities (e.g., Cal-Berkeley, U. Michigan-Ann Arbor) are aligned along the positive vertical axis. That axis is almost perfectly countered by the downward-pointed ray `SATle800`, which is the Bayes factor capturing the prevalance of students with combined Verbal & Math SAT scores less than or equal to 800, i.e., the lowest tail of SAT scores.

### Breadth versus specialization

The horizontal `Y1` axis isn’t so readily interpretable. However, we see the ray `discBreadth`, whcih is the feature capturing the entropy (variety) in academic disciplines in which degrees are offered from the college, is pointing into the upper-left corner of the plot. So colleges aligned along this ray in the upper-right quadrant are the big public state universities that offer a broad range of degrees. On the other hand, the narrowly, highly specialized colleges appear in the lower-right quadrant of the plot.

### Pell grants & high 3-yr credit default rates

The colleges in the lower-left quadrant are the colleges most strongly aligned with rays `pellever`, which captures prevalence of students having ever received a federal Pell grant, and `CDR3est`, which captures prevalence of students defaulting on student loans within 3 years of leaving the college.

### More privates, but less elite

The upper-right quadrant is aligned with `SAT1400` (highest SAT students), `fsend5` (applied to many colleges), and `pgt48Kle75K` (mid-income families).

## Summary

This was an exploratory analysis investigating structure in the U.S. Dept. of Education College Scorecard dataset.