College Scorecard: Earnings Premium & Value Proposition

Michael L. Thompson

September 2, 2017

Contents

Introduction

The U.S. Department of Education College Scorecard database is a rich source of information intended to help students and parents understand the true costs of attending college in America. This script leverages the database to compute an *earnings premium* and a *value proposition* for each college.

IMPORTANT: This is an exploratory data analysis. It is not a commentary on the actual value represented by a college. Future earnings aren't the sum total of the value of a college education.

College Exploration Application

See the Best Colleges for You web-based app to explore this same College Scorecard database.

Earnings Premium

We begin with this premise:

- *The earnings permium for a college is the earnings in excess of the average (across all colleges) of the median annual earnings, measured at* 6 *years after matriculation, plus key factors other than college reputation that might predict earnings.*

The earnings premium for college i, π_i, is given by the following formula:

$$\pi_i \equiv e_i - (\bar{e} + f(A_i, \mathbf{D}_i, L_i)),$$

where e_i is the reported median annual earnings for students from college i at 6 years after matriculation; $\bar{e} \equiv \frac{1}{N} \times \sum_{i=1}^{N} e_i$; and $\sum_{i=1}^{N} \pi_i \equiv 0$.

The function $f(A_i, \mathbf{D}_i, L_i)$ serves to control for the following factors:

(1) Academic abilities of the students at the college, A_i.
(2) Academic discipline distribution of Bachelor's degrees d_{ij} of the students graduating from the college, $\mathbf{D}_i = \{d_{ij}\}$.
(3) Geographic location and campus locale of the university, L_i.
(4) Any significant interactions amongst A_i, $\mathbf{D}_i$, and L_i.

Academic ability A_i is measured by SAT. Academic discipline $\mathbf{D}_i$, by percentages of students receiving each type of degree d_{ij}.
And, geographic location L_i, by the region (New England, Mid-East, Southeast, Southwest, Plains, Great Lakes, Rocky Mountains, and Far West) and the locale (City, Suburb, Town, and Rural) in which college i is located.

Therefore, we intend that the earnings premium π_i captures the impact of the reputation of college i on the future earnings of its students.

Value Proposition

The value proposition of a college i, V_i, is defined as follows:

- *The ratio of the adjusted annual earnings $\tilde{e}_i$ to the expected annual costs of attendance c_i.*

Thus, V_i is represented by the following formula:

$$V_i \equiv \frac{\tilde{e}_i - c_i}{c_i} = \frac{\tilde{e}_i}{c_i} - 1$$

The adjusted annual earnings for college i, $\tilde{e}_i$, is computed as follows:

$$\tilde{e}_i = e_i^* \times p_i = (\pi_i + \bar{e}) \times p_i = (e_i - f(A_i, \mathbf{D}_i, L_i)) \times p_i$$

.

The component terms are defined as follows:

(1) The expected annual earnings for college i, e_i^* is gotten by adding the earnings premium π_i to the overall mean (across all colleges) of annual earnings 6 years after matriculation $\bar{e}$, thus controlling the reported earnings e_i for A_i, $\mathbf{D}_i$, and L_i.
(2) The *adjustment* is performed by multiplying e_i^* by the completion rate p_i, i.e., we adjust for the probability of actually achieving a degree from the college.

The annual costs c_i are provided by the *net price* (i.e., tuition, fees and living expenses less financial aid) across a range of financial need levels:

- without financial aid and
- with financial aid corresponding to the household income level (i.e., the net price for the household income level).

Therefore, we intend that the value proposition V_i captures the return on your costs for attending the college.

Values greater than 0 indicate the college is a good value (at that level of financial aid); and values less than 0 indicate a poor value.

Data Preparation

Load the kaggle.com version of the U.S. Dept. of Education College Scorecard Dataset and generate features for modeling using similar code as used in the previously submitted "Best Colleges for You" script. It uses package `RSQLite` to load the database, and we pare down the database to approximately 900 4-year Bachelor's degree granting colleges.

Feature Engineering: Bayes Factors

Approximate Bayes factors serve as the features of the model.

The Bayes factor BF is the ratio of the posterior-odds of the hypothesis (i.e., after receiving the evidence) to the prior-odds of the hypothesis (i.e., before seeing the evidence).

The evidence is the attribute of a student – like "*SAT score greater than 1400*" – and the hypothesis is attendance at the college.

(*There may not be a compelling reason to do this, but informal testing in other applications indicates a greater ability to distinguish amongst the colleges in the Bayes factors feature space than in the raw variable space.*)

We compute the Bayes factor, $\text{BF}_{\log 10}$ by applying $\log_{10}$ to the posterior-to-prior odds ratio. The equations below show the calculation for $\text{BF}_{\log 10}$ defining the feature corresponding to attribute Attribute Y for each college of interest College X.

$$
\begin{aligned}
&\text{BF}_{\log 10}(H = \text{College } X | E = \text{Attribute } Y) \\
&= \log_{10}\left[\frac{\text{Odds}(H = \text{College } X | E = \text{Attribute } Y)}{\text{Odds}(H = \text{College } X)}\right] \\
&= \log_{10}\left[\frac{\text{P}(E = \text{Attribute } Y | H = \text{College } X)}{\text{P}(E = \text{Attribute } Y | H = \neg(\text{College } X))}\right] \\
&\approx \log_{10}\left[\frac{\text{P}(E = \text{Attribute } Y | H = \text{College } X)}{\text{P}(E = \text{Attribute } Y)}\right]
\end{aligned}
$$

E represents the *evidence* – in this case, observing a college student with Attribute Y. H represents the *hypothesis* – in this case, the proposition that the student is enrolled at College X.

Above, the flip in the propositions on each side of the conditional ("|") when we go from $\text{Odds}(H|E)$ to probabilities, $\text{P}(E|H)$, occurs by applying Bayes' Rule.

The final approximation is a good one because there are 900 colleges in our working set; so the students attending College X are a very small proportion of the entire American student population. Therefore, the probability of finding a student with Attribute Y amongst students *not* at College X is basically the same as finding such a student amongst the entire student population (College X included).

Estimating the Probabilities

For some of the probabilities used in approximating the Bayes factors, the probabilities $\text{P}(E = \text{Attribute } Y | H = \text{College } X)$ are simply the reported proportions in the database.

But, the main challenge in using Bayes factors as features is computing the final probabilities for attributes for which only moments and quantiles are reported in the database. For such attributes, we first approximate a full probability distribution of students at the college of interest and then apply the proposition.

Take SAT for example. We are given the mean, median and quartiles for SAT scores of students at each college. For each college, we fit a model to these values to approximate the continuous distribution of SAT scores of students at the college. Then, for an attribute such as Attribute $Y =$ "*SAT score greater than 1400*", we compute $\mathrm{P}(E = \text{Attribute } Y | H = \text{College } X)$ by simply computing the upper tail probability of SAT with lower limit 1400. Finally, to compute $\mathrm{P}(E = \text{Attribute } Y)$, we simply take the weighted average, across all colleges, of the previously calculated conditional probability, where the weighting is simply the proportion of students attending College X amongst all students in our 900-college universe. In the database, field `UGDS[i]` is the number of undergraduate students attending a college i, mathematically denoted as $N_{\mathrm{UGDS}}(\text{College } i)$. So the proportion of all students attending College X is $\mathrm{P}(H = \text{College } X) = \frac{N_{\mathrm{UGDS}}(\text{College } X)}{\sum_i N_{\mathrm{UGDS}}(\text{College } i)}$.

For attributes like region and locale, to estimate $P(E = \text{Attribute } Y | H = \text{College } X)$, we define Attribute Y to reflect preference. So, for, say `Region == 'Great Lakes'`, Attribute Y is "prefers attending college in Great Lakes region". And, we use our own subjective knowledge to just assume that of the students attending a college in a specific region (or locale), a non-zero proportion of them preferring attending colleges an alternative region (or locale) and that the alternative regions (locales) that are more similar to the one in which the college exists would have a higher proportion of such students than alternatives that are less similar. For regions, similarity is determined by distance – e.g., Mid-East is more similar to Great Lakes than is Far West. For locales, similarity is determined by character – i.e., campuses in suburbs of large cities are more similar to those in large cities than are remote rural campuses.

Earnings Premium and Value Proposition Calculation

A true "earnings premium"" cannot be reliably computed from this dataset because we only have earnings data for years 2003 and 2005 years, and all manner of factors are obscured by the way the data are aggregated and the noisiness of values.

Package `glmnet` is used to estimate a model, including the discrete factor `College_ID` as a fixed effect. For those colleges with a corresponding coefficient included in the model, we use the estimated coefficient as the earnings premium. Many of the colleges drop out of the model. So, as you'll see below, the histogram of earnings premiums looks like a mixture of a spike at zero and a broad normal distribution with tails stretching away from zero in either direction.

Model to Predict Expected Earnings

First, we use package `glmnet` to fit the model to the median earnings at 6-years after matriculation.

After defining `covariates`, a character vector of the covariate names, we computed the correlation matrix of the columns in `studentBF`. Subsets of the columns were defined by a single multi-valued variable, e.g., `region`, so there's lots of collinearity and some pure confounding of interactions with main effects and other interactions. So, we pruned the covariate vector based on a heuristic screening of highly correlated terms in the `model.matrix`.

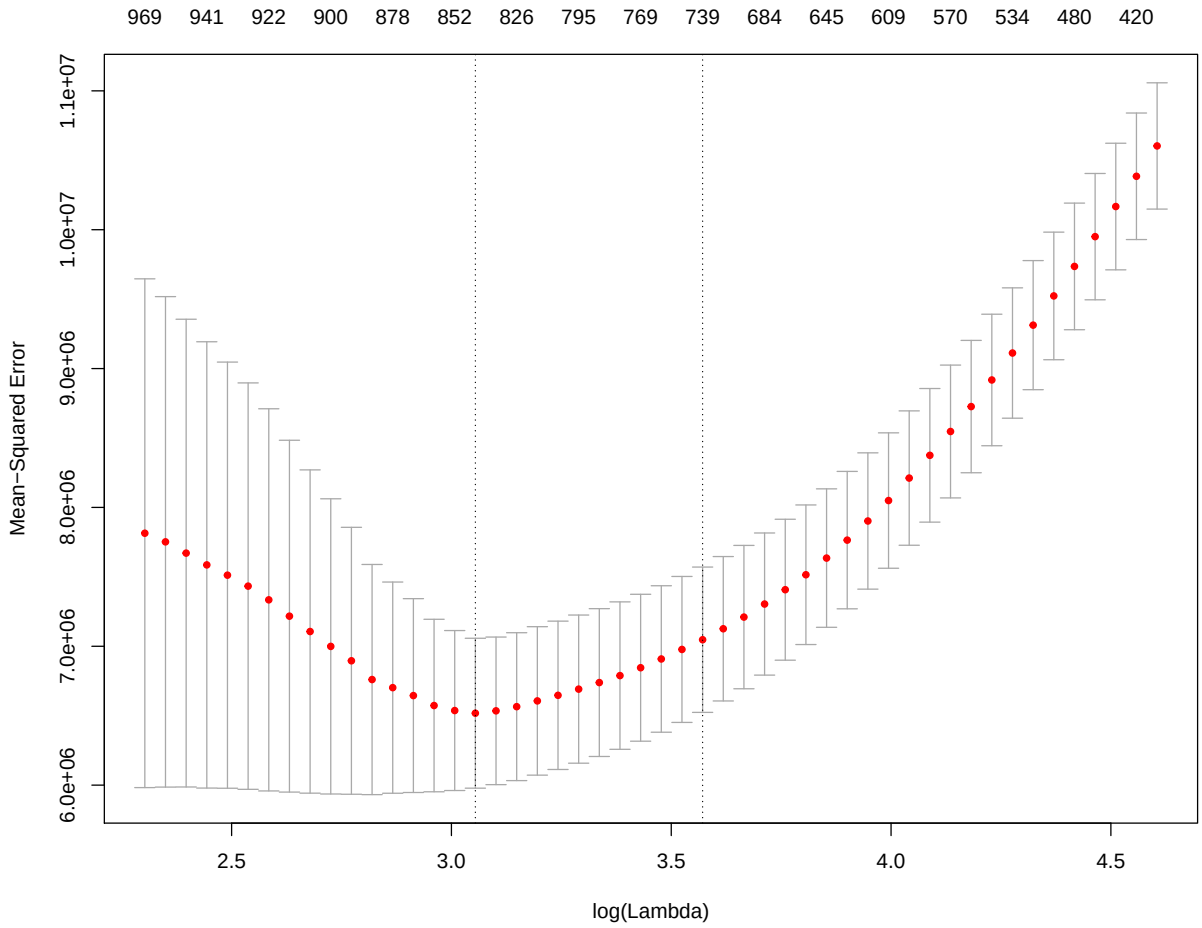
```
covariates <- c(
  "Year2003",
  "BF_localeAggRural",
  #"BF_localeAggTownRemote",
  #"BF_localeAggTownDistant",
  #"BF_localeAggSuburbSmall/Midsize & Town:Fringe","BF_localeAggSuburbLarge",
  "BF_localeAggCitySmall",
  #"BF_localeAggCityMidsize",
```

```
  "BF_localeAggCityLarge",
  "BF_FarWest(AK,CA,HI,NV,OR,WA)",
  "BF_GreatLakes(IL,IN,MI,OH,WI)",
  "BF_MidEast(DE,DC,MD,NJ,NY,PA)",
  #"BF_NewEngland(CT,ME,MA,NH,RI,VT)",
  "BF_Plains(IA,KS,MN,MO,NE,ND,SD)",
  #"BF_RockyMountains(CO,ID,MT,UT,WY)",
  "BF_Southeast(AL,AR,FL,GA,KY,LA,MS,NC,SC,TN,VA,WV)",
  "BF_Southwest(AZ,NM,OK,TX)",
  "BF_SAT_le800",
  #"BF_SAT_gt800le1000",
  "BF_SAT_gt1000le1200","BF_SAT_gt1200le1400",
  "BF_SAT_gt1400",
  "BF_AgricultureAgriculture",
  "BF_NaturalResources","BF_ArchitectureRelated",
  "BF_AreaEthnic","BF_CommunicationJournalism",
  "BF_CommunicationsTechnologies","BF_ComputerInformation",
  "BF_PersonalCulinary","BF_Education",
  "BF_Engineering","BF_EngineeringTechnologies",
  "BF_ForeignLanguages","BF_FamilyConsumer",
  "BF_LegalProfessions","BF_EnglishLanguage",
  "BF_LiberalArts","BF_LibraryScience",
  "BF_BiologicalBiomedical","BF_MathematicsStatistics",
  #"BF_MilitaryTechnologies",
  "BF_MultiInterdisciplinary",
  "BF_ParksRecreation","BF_PhilosophyReligious",
  "BF_TheologyReligious","BF_PhysicalSciences",
  "BF_ScienceTechnologies","BF_Psychology",
  "BF_HomelandSecurity","BF_PublicAdministration",
  "BF_SocialSciences","BF_ConstructionTrades",
  "BF_MechanicRepair","BF_PrecisionProduction",
  "BF_TransportationMaterials","BF_VisualPerforming",
  "BF_HealthProfessions","BF_BusinessManagement",
  "BF_History",
  "College_ID"
  #"BF_discBreadth",
  #"Living_Expenses",
  #"topSAT2","bottomSAT2"#,
  # "breadth_health","breadth_arts",
  # "breadth_engg","breadth_busmgmt",
  # "breadth_libarts","breadth_bio",
  # "breadth_socsci","breadth_edu",
  # "breadth_eng","breadth_psych",
  # "breadth_prks","breadth_hmsec"
)

# formula_string <- paste0(
#   'outcome ~ (.)^2 + ',
#   paste(sprintf('poly(BF_%s,2)',setdiff(discNames,"MilitaryTechnologies")), collapse="+")
# )
#formula_string <- 'outcome ~ (.-College_ID)^2 + College_ID'
glmdata   <- studentBF %>%
  mutate(College_ID = paste(unitID,College,sep="__")) %>%
```

```
  select(outcome, one_of(covariates) ) %>%
  filter(complete.cases(.))

# Try a model with `College` as a fixed effect; and prune out a bunch of correlated stuff.
model_mat <- model.matrix( outcome ~ (.-College_ID)^2 + College_ID, data=glmdata )
fcor      <- cor(model_mat[,-1]) - diag(ncol(model_mat)-1)
nms       <- colnames(fcor)
hicor     <- nms %>%
  sapply(
    function(nm) {
      # scan the upper triangle of the correlation matrxi...
      ihi<-which(abs(fcor[seq_len(which(nms==nm)),nm])>0.8)
      setNames(fcor[ihi,nm],nms[ihi])
    }
  )
hicor     <- hicor[sapply(hicor,length)>0]
```

Now, we have the model's design matrix, `model_mat`. We use `cv.glmnet` to perform variable selection to arrive at the final model for each college's earnings `outcome`.

```
# Since we know that some of the interactions are confounded with each other and
# with some main effects, which causes the cross-validation error to blow up,
# we'll find the models that have fewer terms than those that blow up and plot
# the cross-validation mean-squared error for them. Range of `lambda` determined
# by trial and error.
set.seed( 2393 )
glmnet_cv <- cv.glmnet(
  x      = model_mat[,setdiff(colnames(model_mat),names(hicor))],
  y      = glmdata %>% select(outcome) %>% as.matrix(),
  family = 'gaussian',
  lambda = exp(seq(log(10),log(100),length.out = 50))
)
plot( glmnet_cv )
```




```
# # These are the non-zero coefficients for the model using `lambda.1se`.
# acv %>% coef() %>% {(.)[abs((.)[,1])>0,1]} %>% print()
# # These are the non-zero coefficients for the model using `lambda.min`.
# acv %>% coef(s='lambda.min') %>% {(.)[abs((.)[,1])>0,1]} %>% print()
# These are the non-zero coefficients for the model using the geometric
# mean of `lambda.min` and `lambda.1se`.
df_coef <- glmnet_cv %$%
  coef(.,s=exp(mean(log(c(lambda.1se,lambda.min))))) %>%
  { (.)[abs((.)[,1])>0,1] } %>%
  { data_frame(Coefficient = names(.), `$/util` = round(.,0)) }
df_coef %>%
  formattable() %>%
  as.datatable()
```

```
# We use that latter model.
# It's intercept serves as the grand-mean, i.e., expected earnings
# of a college after controlling for the covariates
# and before adding the college's earnings premium.
Expected_Earnings <- glmnet_cv %$%
  coef(.,s=exp(mean(log(c(lambda.1se,lambda.min)))))['(Intercept)',]
print( Expected_Earnings )
```

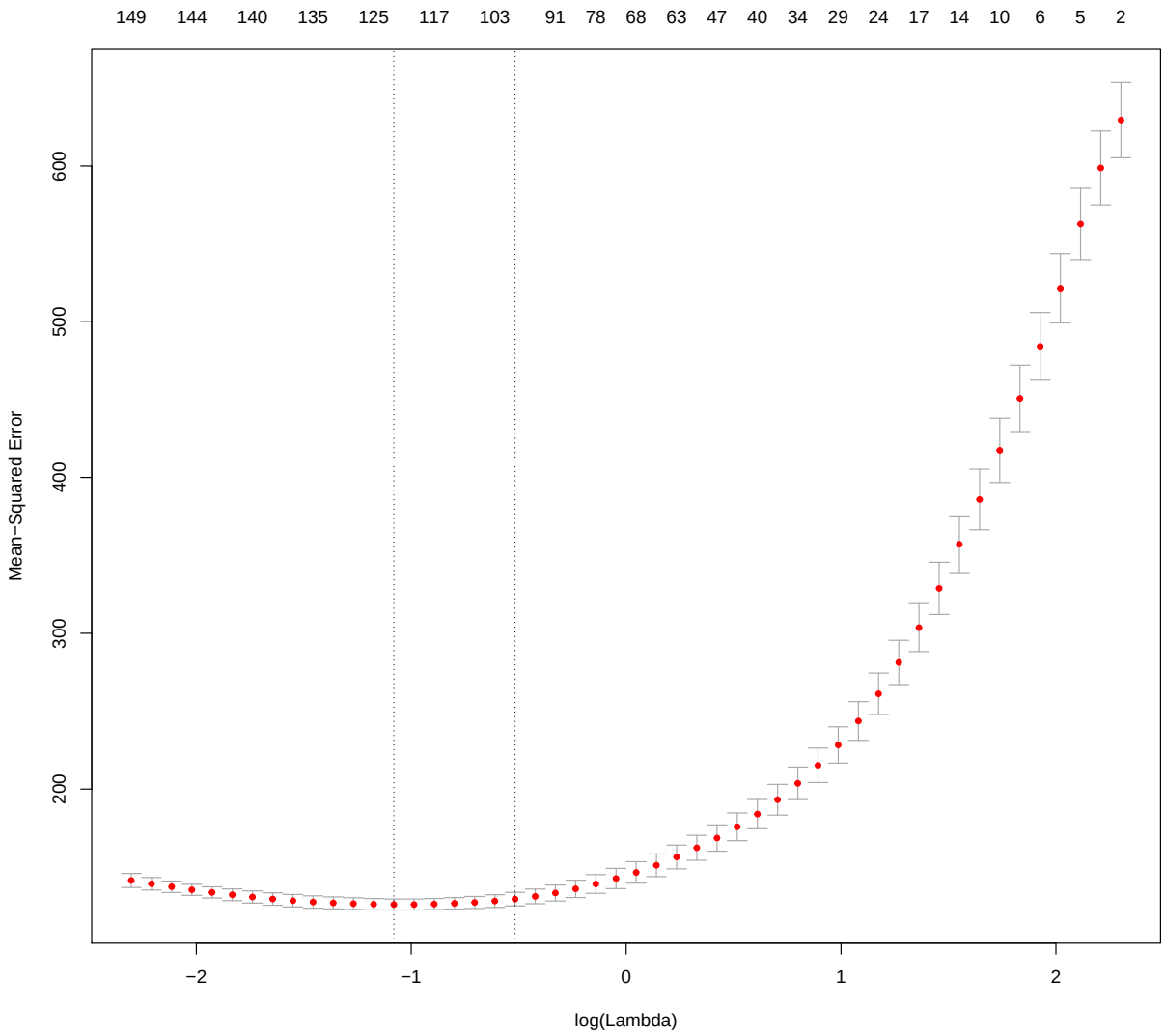
```
## [1] 36370.82
```

Visualization of College (t-SNE Biplot)

Let's plot the colleges in 2-D space to see if we can find patterns. We use *t-Distributed Stochastic Neighborhood Embeddings* (t-SNE) on the relevant predictors of the outcome variable to find a good 2-D representation in which we hope to be able to identify clusters of similar colleges having similar earnings outcomes.

```
set.seed( 191 )
# system.time(
#   {
tsne_mat <- model_mat %>%
  as_tibble() %>%
  filter(Year2003==0) %>%
  dplyr::select(one_of(df_coef$Coefficient)) %>%
  dplyr::select(-matches('Intercept|Year|College_ID') ) %>%
  select_if( .predicate = function(x) any(x != x[[1]]) ) %>%
  as.matrix() %>%
  scale()
tsne_out <- Rtsne( tsne_mat, perplexity = 10, initial_dims = 20 )
#   }
# )
```

```
set.seed( 2393 )
tsne_glmnet_cv <- cv.glmnet(
  x      = tsne_mat,
  y      = tsne_out$Y,
  family = 'mgaussian',
  lambda = exp(seq(log(0.1),log(10),length.out = 50))
)
plot( tsne_glmnet_cv )
```




```
tsne_glmnet_coef <- tsne_glmnet_cv %>% coef()
tsne_glmnet_coef$y1 %>%
{ (.)[abs((.)[,1])>0,1] } %>%
{ data_frame(Coefficient = names(.), `$/util` = round(.,2)) } %>%
  print()
```

```
## # A tibble: 98 x 2
##                                        Coefficient `$/util`
##                                              <chr>    <dbl>
##  1                                     (Intercept)     0.00
##  2 `BF_Southeast(AL,AR,FL,GA,KY,LA,MS,NC,SC,TN,VA,WV)`    -0.57
##  3                           BF_SAT_gt1200le1400    -0.68
##  4                                BF_SAT_gt1400    -0.86
##  5                           BF_PersonalCulinary     0.19
##  6                                BF_Engineering    -0.12
##  7                             BF_FamilyConsumer     0.06
##  8                        BF_PhilosophyReligious    -0.04
##  9                          BF_TheologyReligious    -4.22
```

```
## 10                                     BF_SocialSciences    -0.56
## # ... with 88 more rows
```

```
tsne_glmnet_coef$y2 %>%
{ (.)[abs((.)[,1])>0,1] } %>%
{ data_frame(Coefficient = names(.), `$/util` = round(.,2)) } %>%
  print()
```

```
## # A tibble: 98 x 2
##                                           Coefficient `$/util`
##                                                 <chr>    <dbl>
##  1                                          (Intercept)     0.00
##  2 `BF_Southeast(AL,AR,FL,GA,KY,LA,MS,NC,SC,TN,VA,WV)`     5.59
##  3                                  BF_SAT_gt1200le1400    -0.40
##  4                                       BF_SAT_gt1400    -1.74
##  5                                  BF_PersonalCulinary    -0.01
##  6                                      BF_Engineering    -0.94
##  7                                    BF_FamilyConsumer     0.93
##  8                                BF_PhilosophyReligious    -0.02
##  9                                  BF_TheologyReligious     0.43
## 10                                     BF_SocialSciences    -0.25
## # ... with 88 more rows
```

```
tsne_coef_df <-
  tsne_glmnet_coef$y1 %>%
  as.matrix() %>%
  as.data.frame() %>%
  as_tibble() %>%
  rownames_to_column() %>%
  setNames(c("Coefficient","Y1")) %>%
  full_join(
    tsne_glmnet_coef$y2 %>%
      as.matrix() %>%
      as.data.frame() %>%
      as_tibble() %>%
      rownames_to_column() %>%
      setNames(c("Coefficient","Y2")) %>%
      mutate(Y2 = -Y2 ),
    by = "Coefficient"
  ) %>%
  filter( abs(Y1) > 1.0E-9 | abs(Y2) > 1.0E-9 )

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

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

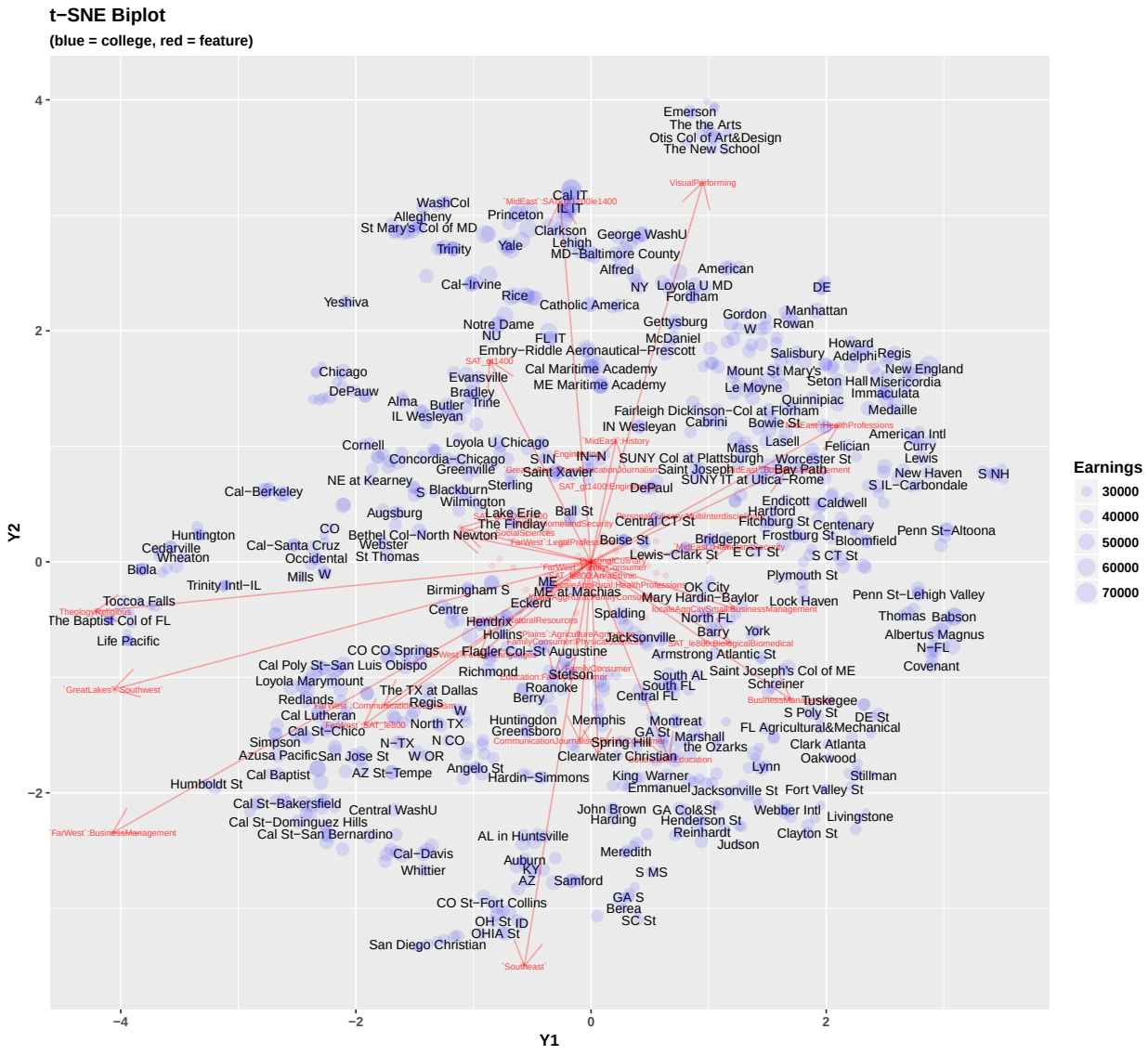
```
college_names <- glmdata %>%
  filter(Year2003 == 0 ) %$%
  College_ID %>%
  { gsub('^[0-9_]+','',. ) } %>%
  { gsub('Northwestern University','NU',.) } %>%
  { gsub('California','Cal',. ) } %>%
  { gsub('Mass.+Inst.+Tech','MIT',. ) } %>%
  { gsub('(Mass|Penn|Wash)[^ ]+ *','\\1',.) } %>%
  { gsub('Polytechnic','Poly',. ) } %>%
  { gsub('Institute of Tech[^ ]+','IT',. ) } %>%
  { gsub('Tech.+Inst.+','Tech',. ) } %>%
  { gsub('State','St',. ) } %>%
  { gsub('University','U',. ) } %>%
  { gsub('(U of )|( U$)','',. ) } %>%
  { 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],.) }
}

categories <- {
  tsne_mat[,key_terms] %*%
    (tsne_coef_df %>% 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 <- tsne_out$Y %>%
  as_tibble() %>%
  setNames(c("Y1","Y2")) %>%
  mutate(
    College = college_names,
    category = categories,
    Y2 = -Y2,
    Earnings = glmdata %>% filter(Year2003==0) %$% outcome
  ) %>%
```

```
  dplyr::select( College, category, Earnings, everything() ) %>%
  mutate_at(funs(1.7*scale(.)),.vars=vars(Y1,Y2))

tsne_df %>%
  {
    ggplot(.,aes(x=Y1,y=Y2,color=category)) +
      geom_point() +
      geom_text(aes(label=College),size=3,check_overlap = TRUE) +
      theme( text = element_text( face = 'bold' ) )
  } %>%
  print()
```

A t-SNE biplot can be created by overlaying the colleges in the 2-D t-SNE space on top of the dimensions defined by the terms in a linear model between the outcome model's predictors and the t-SNE coordinates.

In this way, we can readily interpret which features (Bayes factors) define the similarity amongst clustered colleges.

```r
y2_min <- -3.5
tsne_coef_df %>%
  mutate(
    mag = sqrt(Y1^2 + Y2^2),
    Y2 = pmax(y2_min,Y2),
    Coefficient = gsub('\\([^)]+\\)','',gsub('BF_','',Coefficient))
  ) %>%
  {
    ggplot(., aes( x = Y1, y = Y2 ) ) +
      geom_point( color = 'red', alpha = 0.1 ) +
      geom_text(
        aes( label = Coefficient),
        color = 'red',
        alpha = 0.7,
        size = 2,
        check_overlap = TRUE
      ) +
      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_df,
        aes( x=Y1, y=Y2, label=College ),
        mapping=,
        color = 'black',
        size=3,
        check_overlap = TRUE
      )  +
      geom_point( data=tsne_df, aes(x=Y1,y=Y2, size = Earnings ), 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))
  } %>%
  print()
```



Voronoi Tessalation and Bayesian Blocks in 2-D

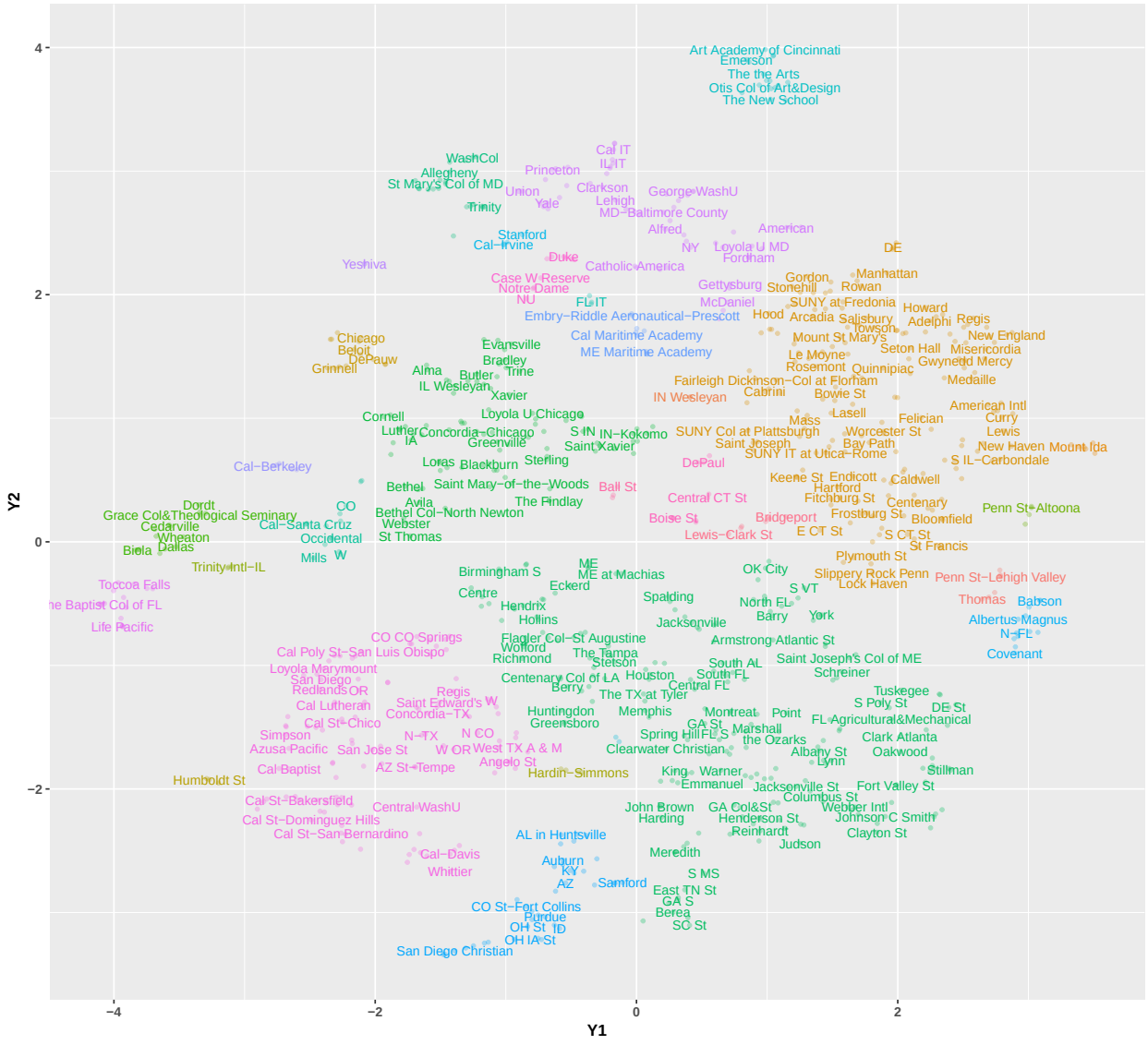
See "Bayesian Blocks in Two or More Dimensions: Image Segmentation and Cluster Analysis" by J.D. Scargle, Nov. 11, 2001, *Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MAXENT 2001)*, held at Johns Hopkins University, Baltimore, MD USA on August 4-9, 2001.

See "Voronoi Diagram and Delaunay Triangulation in R" by Nathan Yau, April 12, 2016.

Hierarchical Clustering

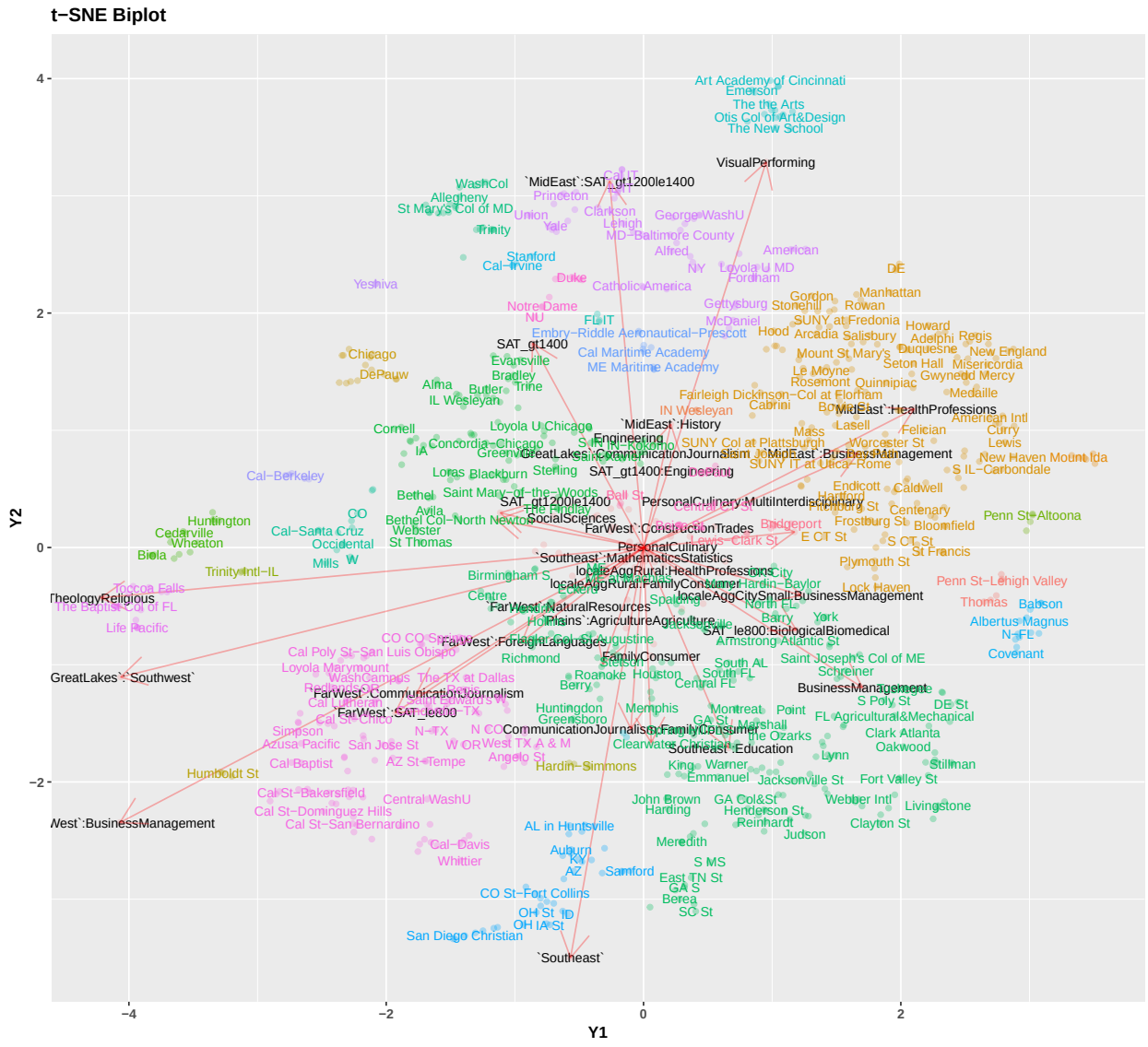
```
tsne_mat_hc <- tsne_df %>% select(Y1,Y2) %>% as.matrix() %>% set_rownames(tsne_df$College)
hc <- hclust( d = dist( tsne_mat_hc ), method = 'single' )
n_cluster <- 30
cluster_id <- cutree( hc, k = n_cluster )

# 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 <- setNames( sample.int(n_cluster)[cluster_id], names(cluster_id) )

tsne_mat_hc %>%
  as_tibble() %>%
  mutate( College = names(cluster_id), cluster = factor( cluster_id ) ) %>%
  {
    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()
```




```
y2_min <- -3.5
tsne_coef_df %>%
  mutate(
    mag = sqrt(Y1^2 + Y2^2) ,
    Y2 = pmax(y2_min,Y2),
    Coefficient = gsub('\\([^)]+\\)','',gsub('BF_','',Coefficient))
  ) %>%
  {
    ggplot(., aes( x = Y1, y = Y2 ) ) +
      geom_point( color = 'red', alpha = 0.1 ) +
      geom_text(
        aes( label = Coefficient ),
        color = 'black',
        size = 3,
        check_overlap = TRUE
      ) +
      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 %>%
        as_tibble() %>%
        mutate(
          College = names(cluster_id),
          cluster = factor( cluster_id )
        ),
      aes( x=Y1, y=Y2, label=College, color = cluster ),
      mapping=,
      show.legend = FALSE,
      size=3,
      check_overlap = TRUE
    )  +
    geom_point(
      data = tsne_mat_hc %>%
        as_tibble() %>%
        mutate(
          College = names(cluster_id),
          cluster = factor( cluster_id )
        ),
      aes(x=Y1,y=Y2, color = cluster ),
      show.legend = FALSE,
      alpha=0.3
    ) +
    ggtitle( "t-SNE Biplot" ) +
    theme( text = element_text( face = 'bold' ) ) +
    scale_y_continuous(limits = c(y2_min,4))
} %>%
print()
```

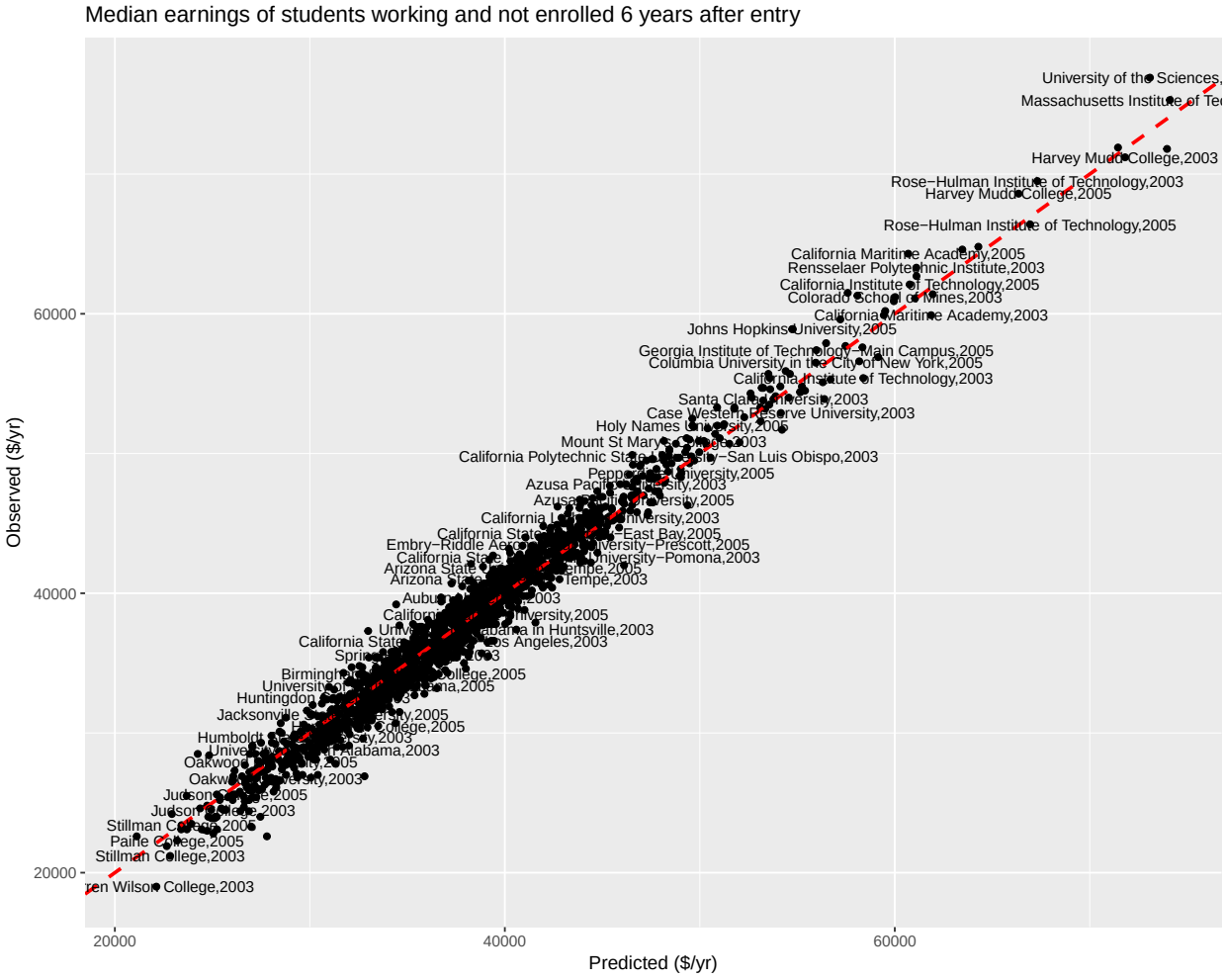


Visualize Predictions

We now collect the predictions and the earnings premia as determined by the model coefficient for the corresponding college. Any college that does not have a coefficient in the model receives an earnings premium of zero.

```
# And, we make predictions for that model using the geometric mean of `lambda.min` and `lambda.1se`..
glmnet_pred <- glmnet_cv %$%
  predict(.,newx = model_mat[,setdiff(colnames(model_mat),names(hicor))], #glmdata %>% select(-outcome)
  s = exp(mean(log(c(lambda.1se,lambda.min))))
)

# abbreviate_college <- function( df, college_col ){
#   college_col <- enquo( college_col )
#   cname       <- quo_name( college_col )
#   nms <- df %>%
#     dplyr::select(!!college_col) %>%
```

```
#     { gsub("Mass.+Inst.+Tech.+","MIT", (.)[[1]] ) }
#   df %>% mutate( !!cname := nms )
# }

# Collect the predictions in a data table...
predictions <- data_frame(predicted_earnings = glmnet_pred[,1]) %>%
  bind_cols(
    studentBF %>%
      select(Year,unitID,College,outcome, one_of(setdiff(covariates,'College_ID')) ) %>%
      filter(complete.cases(.))
  ) %>%
  select(Year,unitID,College,outcome,predicted_earnings)

predictions %>%
  mutate( College = sprintf("%s,%d",College,Year) ) %>%
  {
    ggplot(., aes(x=predicted_earnings,y=outcome)) +
      geom_point() +
      geom_abline(intercept = 0, slope = 1, color = 'red', linetype = 2, size = 1) +
      geom_text(aes(label = College ), size=3, check_overlap = TRUE ) +
      ggtitle('Median earnings of students working and not enrolled 6 years after entry') +
      labs(
        x = 'Predicted ($/yr)',
        y = 'Observed ($/yr)'
      )
  } %>%
  print()
```

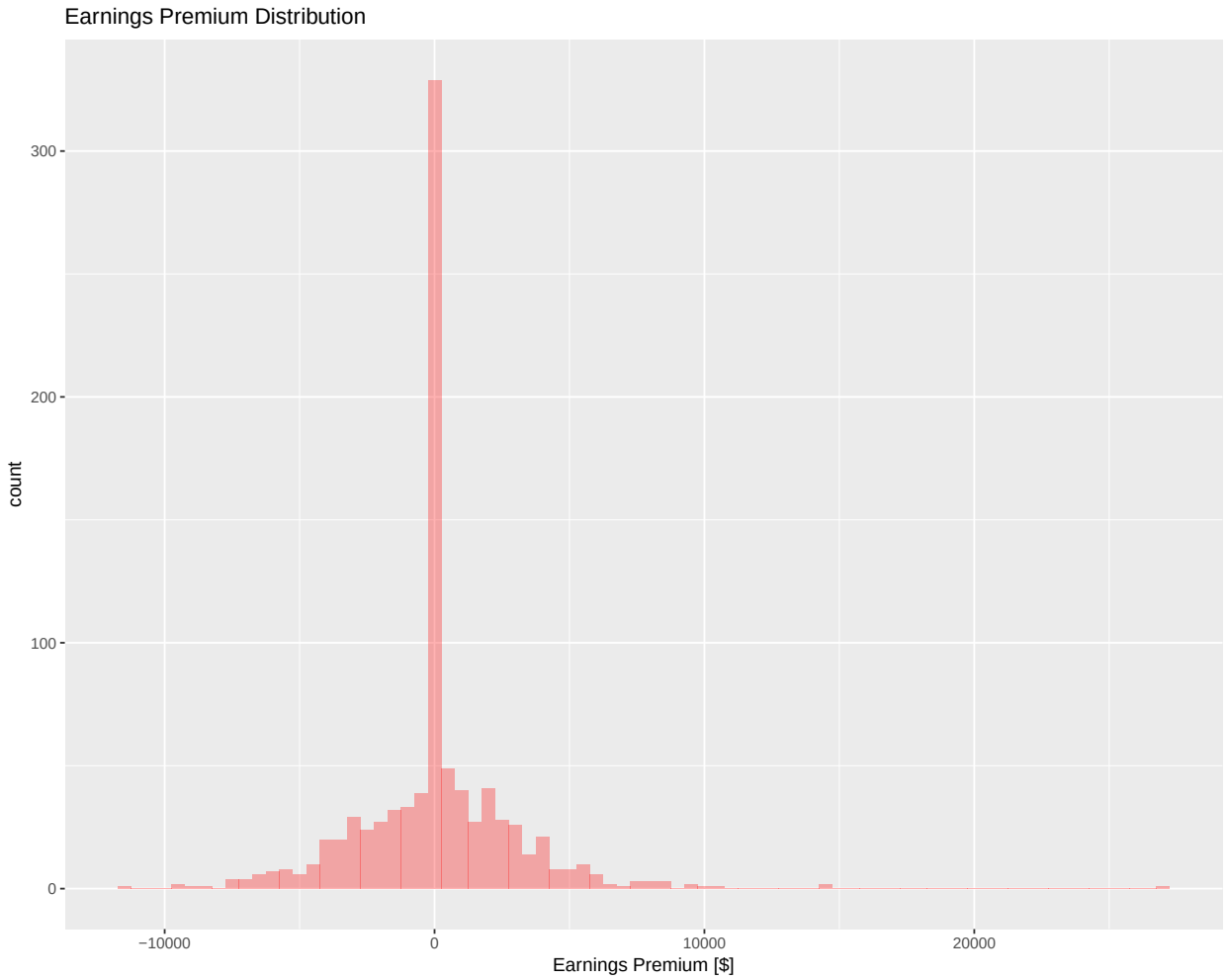



```
epremium <- df_coef %>%
  filter(grepl('^College',Coefficient)) %>%
  mutate(
    unitID = gsub('.+ID([0-9]+)__.+','\\1',Coefficient) %>% as.integer(),
    College = gsub('^College_ID.+__','',Coefficient)
  ) %>%
  #abbreviate_college() %>%
  rename(earnings_premium = `$/util`) %>%
  right_join(student %>% filter(Year==2013) %>% select(unitID,College,SAT_AVG)) %>%
  left_join( student %>% filter(Year==2005) %>% select(unitID,Treasury_md_earn_wne_p6)) %>%
  rename(outcome = Treasury_md_earn_wne_p6) %>%
  mutate( earnings_premium=ifelse(is.na(earnings_premium),0,earnings_premium) ) %>%
  left_join( predictions %>% filter(Year==2005)) %>%
  select( -Coefficient, -Year ) %>%
  select(unitID,College,everything())
```

Plots of the Earnings Premium

First, plot the histogram of earnings premia, demonstrating, as mentioned above mixture of zero and non-zero premia.

```
epremium %>%
{
  ggplot(.,aes(x=earnings_premium)) +
    geom_histogram(fill='red',alpha=0.3,binwidth = 500) +
    ggtitle('Earnings Premium Distribution') +
    labs( x = 'Earnings Premium [$]' )
}
```

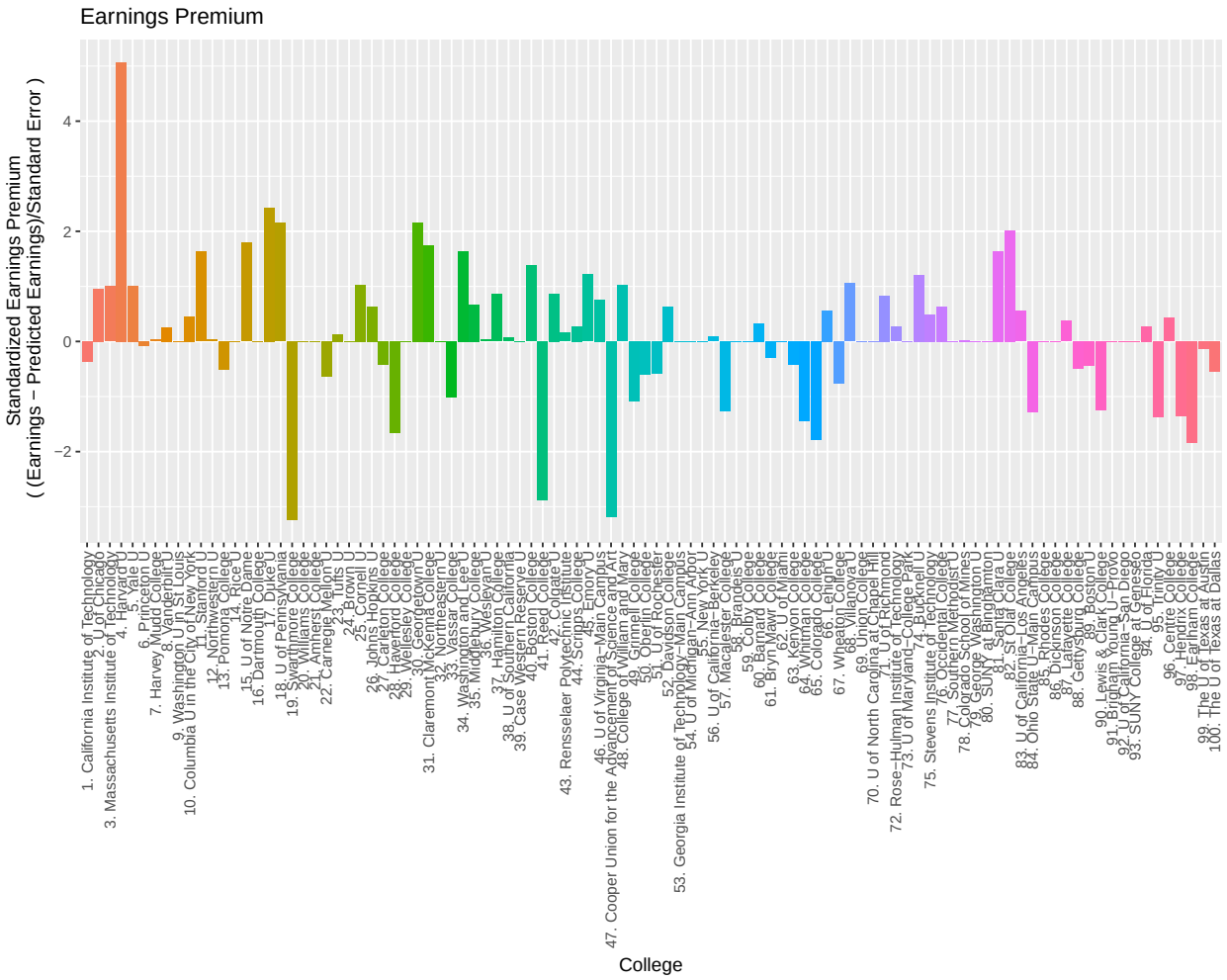


Plot a bar chart, with the colleges ordered by average SAT score, so we can see how the colleges sort out.

(Interestingly, it still looks like some of the more tech-oriented colleges have positive premia while more liberal-arts-oriented colleges have negative premia, even though we explictly attempted to model the impact of degree distribution at each college. Probably an illustration of the limitations of this approach. Also, a reminder of the fact that value from a college education can't be measured in earnings alone.)

```
epremium %>%
  arrange(desc(SAT_AVG)) %>%
  mutate(College = sprintf('%d. %s',seq_len(nrow(.)),gsub('University','U',College))) %>%
  mutate(College=factor(College,levels=College)) %>%
  mutate( tval = earnings_premium/sd(earnings_premium) ) %>%
  slice( seq_len(100) ) %>%
  #filter(tval<5) %>%
  {
```

```
    ggplot(., aes(x=College,y=tval,fill=College)) +
      geom_bar(position='dodge',stat='identity') +
      #scale_y_continuous(limits=c(-4,4)) +
      ggtitle( "Earnings Premium" ) +
      labs( y = 'Standardized Earnings Premium\n ( (Earnings - Predicted Earnings)/Standard Error )') +
      theme(axis.text.x=element_text(angle = 90, vjust = 0.5,hjust = 1),legend.position = 'none')
  }
```



Value Proposition

Value proposition is defined as the adjusted earnings divided by the expected costs for a household of the student's specific income level, thus level of financial need. (Of course, other factors dictate financial need. Here, decreasing annual household income of the students in a cohort is the best we have as an indicator of the financial need of those students.)

Adjusted earnings is defined as the (grand mean) expected earnings plus the earnings premium for the college multiplied by the completion rate of the college. I.e., we assume completion rate is the probability of a student graduating from the college. Then the expected earnings would be this probability multiplied by the earnings of a graduated plus the quantity one minus this probability multiplied by the earnings of non-graduates, which for this analysis, we assume to be zero. So these values will, obviously, tend to under-estimate the actual estimated earnings (even after our controlling for the covariates in the model).

Again, this is more an exploration into roughly how the colleges sort out rather than anywhere near a determination of the actual value represented by a college degree from any of the colleges.

```
makeStudentValue <- function(
  studentBF,
  epremium,
  residence_state,
  income_bracket = 'gt48Kle75K',
  sat_lvl = 'gt1000le1200',
  unthresh = 0
){
  student_value <- studentBF %>% select(-College,-outcome,-matches('Treasury|pell|Year2003')) %>%
    filter( Year == 2013 ) %>%
    left_join( student %>% select(Year,unitID, starts_with('NPT')), by = c('Year','unitID' ))  %>%
    left_join( epremium, by = 'unitID' ) %>%
    mutate(
      Utility = switch(
        sat_lvl,
        gt1400       = BF_SAT_gt1400,
        gt1200le1400 = BF_SAT_gt1200le1400,
        gt1000le1200 = BF_SAT_gt1000le1200,
        gt800le1000  = BF_SAT_gt800le1000,
        le800        = BF_SAT_le800
      )
    ) %>%
    arrange( desc(Utility) ) %>%
    mutate(
      ntp1 = ifelse(grepl('Public',CollegeType),NPT41_PUB,NPT41_PRIV),
      ntp2 = ifelse(grepl('Public',CollegeType),NPT42_PUB,NPT42_PRIV),
      ntp3 = ifelse(grepl('Public',CollegeType),NPT43_PUB,NPT43_PRIV),
      ntp4 = ifelse(grepl('Public',CollegeType),NPT44_PUB,NPT44_PRIV),
      ntp5 = ifelse(grepl('Public',CollegeType),NPT45_PUB,NPT45_PRIV)
    ) %>%
    select( -starts_with('NPT4',ignore.case=FALSE)) %>%
    #filter( complete.cases(.) ) %>%
    mutate(
      INSTATE = state == residence_state,
      Living_Expenses = COSTT4_A - TUITIONFEE_IN,
      maxcost = Living_Expenses + ifelse(INSTATE,TUITIONFEE_IN,TUITIONFEE_OUT),
      ntp  = switch(
        income_bracket,
        le30K       = ntp1,
        gt30Kle48K  = ntp2,
        gt48Kle75K  = ntp3,
        gt75Kle110K = ntp4,
        gt110K      = ntp5
      ),
      Earnings_Adjusted = (Expected_Earnings + earnings_premium)*C150_4_POOLED_SUPP, # Assumes all rema
      inccost = ifelse( INSTATE, ntp, ntp + TUITIONFEE_OUT - TUITIONFEE_IN),
      unorm   = Utility/max(Utility), #10^(Utility-max(Utility)),
      cnorm   = maxcost/Living_Expenses,
      c2norm  = inccost/Living_Expenses,
      vnorm   = (Earnings_Adjusted/Living_Expenses - 1.0) / (Earnings_Adjusted[[1]]/Living_Expenses[[1]]
```

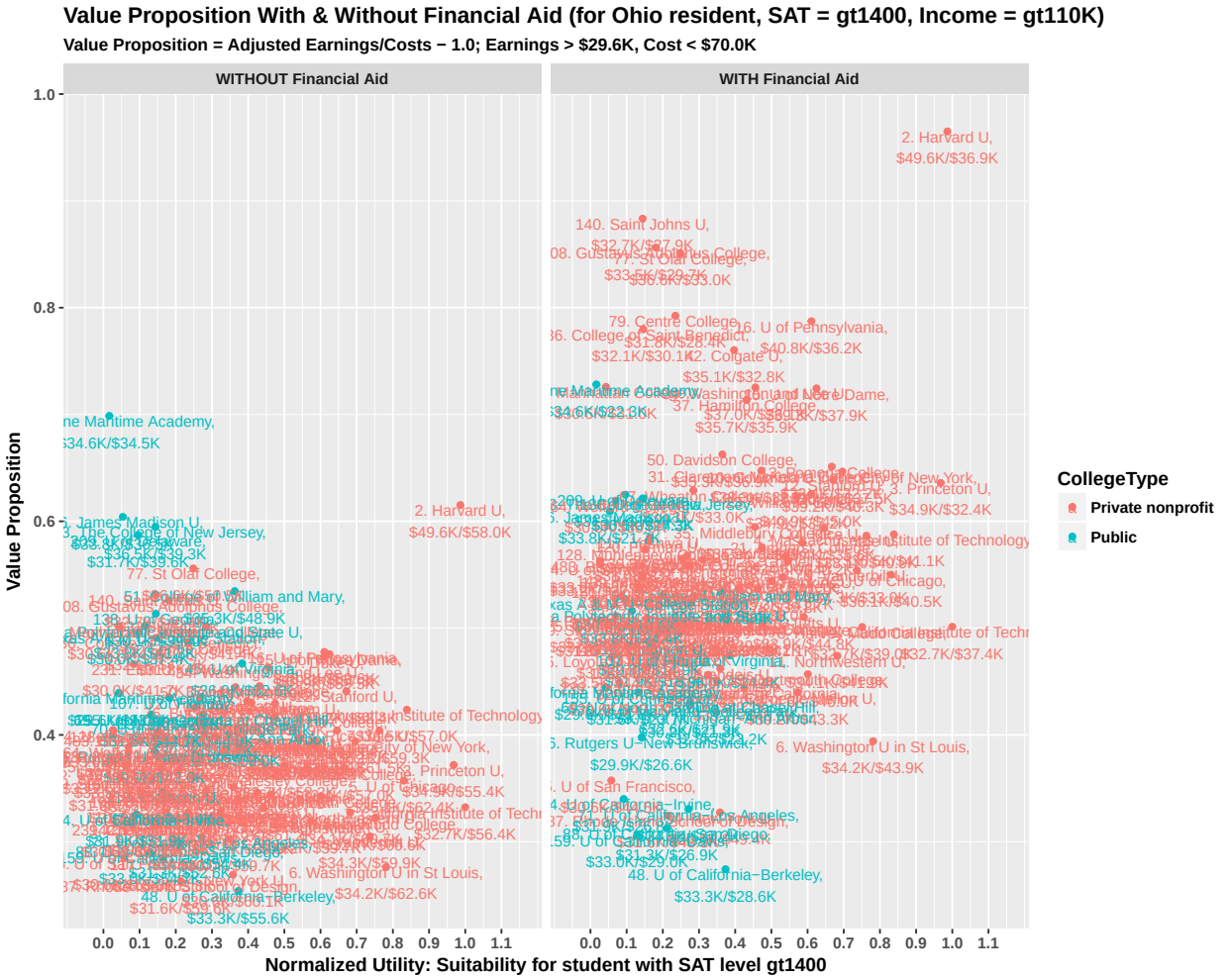
```
      value_prop1 = vnorm/cnorm, #unorm*vnorm/cnorm,
      value_prop2 = vnorm/c2norm, #unorm*vnorm/c2norm,
      College = sprintf("%d. %s", order(Utility,decreasing=TRUE), gsub('University','U',College) )
    ) %>%
    filter( unorm>unthresh, value_prop2 < 2*value_prop2[[1]]) %>%
    mutate(
      maxv2 = max(value_prop2[unorm>unthresh]),
      value_prop1 = value_prop1/maxv2,  #ifelse(scale_independently,max(value_prop1),max(value_prop2)),
      value_prop2 = value_prop2/maxv2
    ) %>%
    #filter(value_prop2 < 4 & vnorm < 2 ) %>%
    gather( key = aid, value = value_prop, value_prop1, value_prop2 ) %>%
    mutate(
      cost = ifelse( aid == 'value_prop1', maxcost, ntp ),
      coll_label = sprintf("%s,\n$%.1fK/$%.1fK",gsub("^(.+)[^a-zA-Z]+Main Campus","\\1",College),Earning
    ) %>%
    mutate( aid = ifelse( aid == 'value_prop1', "WITHOUT Financial Aid", "WITH Financial Aid" ) %>% fac
}


plotValue <- function( stdt_val, residence_state, sat_lvl, income_bracket, scale_independently = FALSE,
  xlims <- c(pmax(unthresh,floor(min(stdt_val$unorm)/0.1)*0.1)-0.05,1.15)
  stdt_val %>%
    filter( unorm > unthresh, cost < cost_thresh, Earnings_Adjusted > earnings_thresh ) %>%
    {
      ggplot(., aes( x = unorm, y = value_prop, color = CollegeType ) ) +
        geom_point() +
        geom_text( aes( label = coll_label), vjust = 1.0, size = 3 ) +
        scale_x_continuous( limits = xlims, breaks = seq(xlims[1]+0.05,xlims[2]-0.05,by=0.10) ) +
        ggtitle(
          label    = sprintf('Value Proposition With & Without Financial Aid (for %s resident, SAT = %s
          subtitle = paste(
            'Value Proposition = Adjusted Earnings/Costs - 1.0',
            sprintf('Earnings > $%2.1fK, Cost < $%2.1fK', earnings_thresh/1000, cost_thresh/1000),
            sep = "; "
          )
        ) +
        labs(
          x = sprintf('Normalized Utility: Suitability for student with SAT level %s',sat_lvl),
          y = ifelse(scale_independently,'Value Proposition (scaled independently)','Value Proposition'
        ) +
        facet_wrap( ~ aid, scales = ifelse(scale_independently,'free','fixed') ) +
        theme( text = element_text( face = 'bold' ) )
    }
}
```

Case: High-SAT, High-Income (i.e., Low-Financial-Need)

Here's the calculation for an Ohio resident from a high-income household (greater than $110,000/yr) and whose SAT score is at the highest level, greater than 1400.

```
res_state <- 'Ohio'
sat_level <- 'gt1400'
```

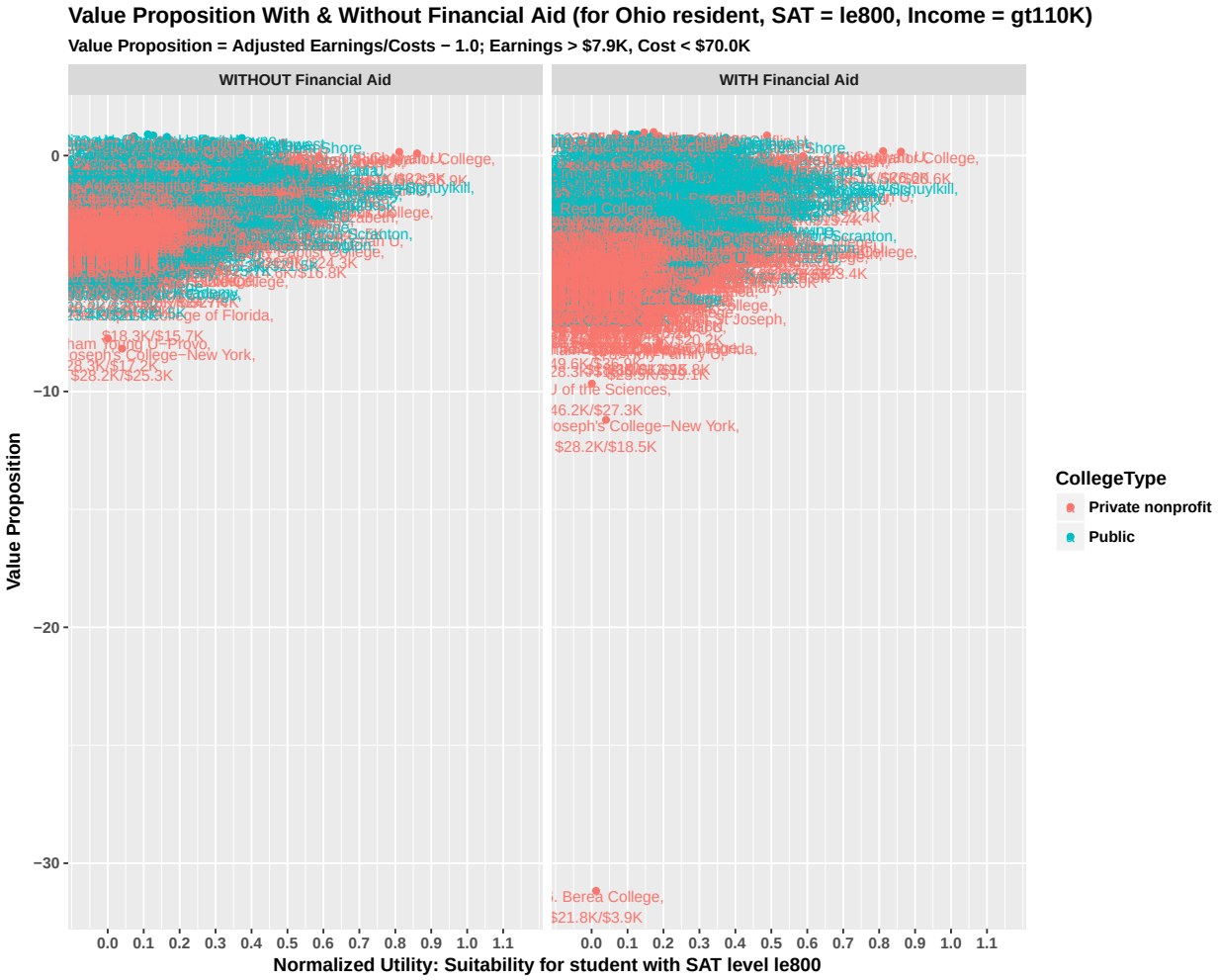
```
inc_level <- 'gt110K'
student_value <- makeStudentValue(
  studentBF,
  epremium,
  residence_state = res_state,
  sat_lvl         = sat_level,
  income_bracket  = inc_level
)
ethresh <- quantile( student_value$Earnings_Adjusted[1:30], 0.04 )
student_value %>%
  plotValue(
    residence_state = res_state,
    sat_lvl         = sat_level,
    income_bracket  = inc_level,
    earnings_thresh = ethresh
  )
```



Case: Low-SAT, High-Income (i.e., Low-Financial-Need)

And here's the calculation for an Ohio resident from a high-income household (greater than $110,000/yr) and whose SAT score is at the lowest level, less than 800.

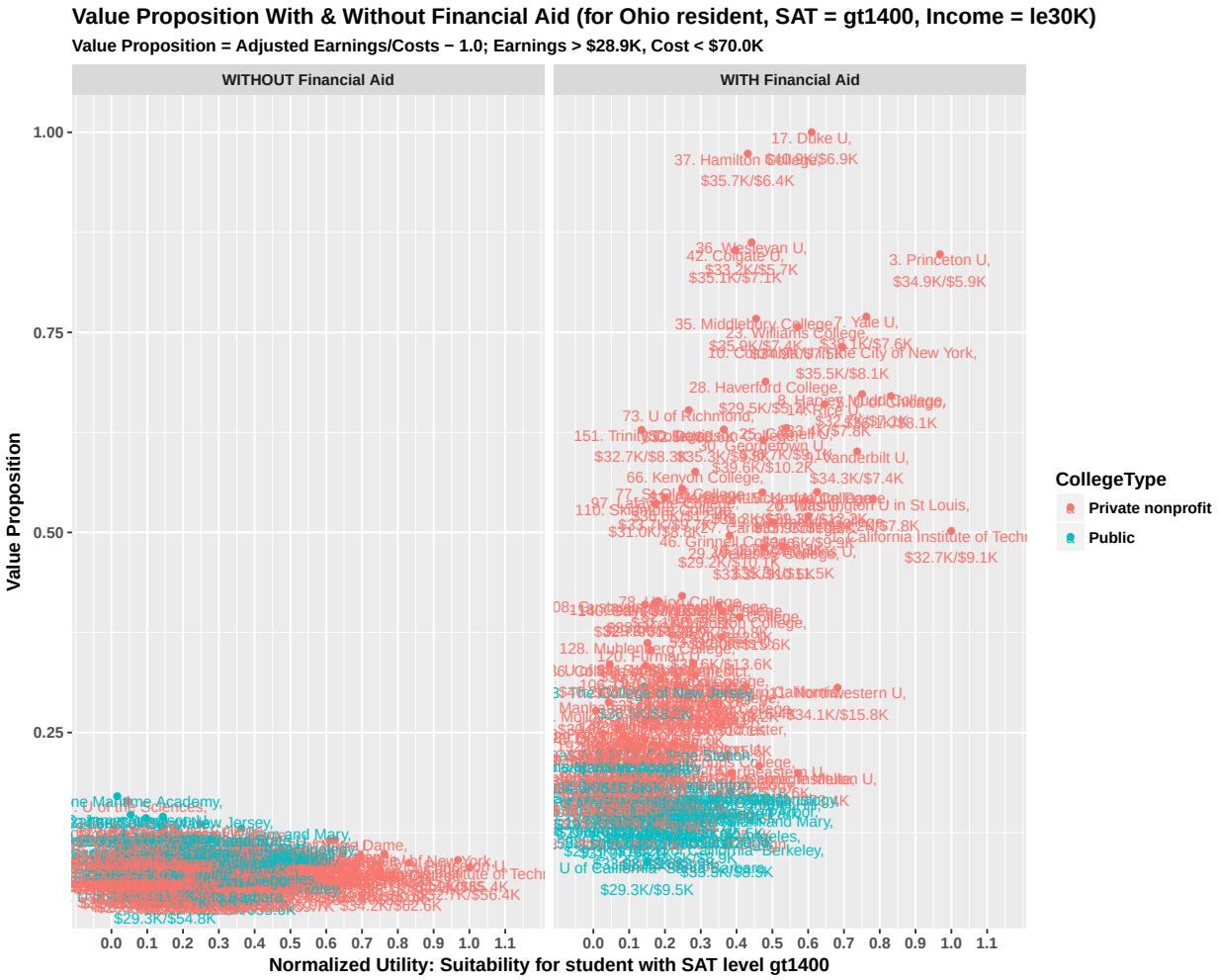
Notice how the adjusted earnings shown in the data point labels are significantly lower than those of the previous plot: In aggregate, the academic ability of a college's students is a strong predictor of the future earnings of the students. (Says nothing about any individual's case.) Also, see how financial aid makes the private colleges competitive in value to the public colleges by reducing the net price in the denominator of the formula.



Case: High-SAT, Low-Income (i.e., High-Financial-Need)

Here's the calculation for an Ohio resident from a low-income household (less than or equal to $30,000/yr) and whose SAT score is at the highest level, greater than 1400.

Notice how financial aid makes attending an elite private college a fantastic value for such a student.



Value Proposition Given Student's SAT Level and Financial Need Levels

Rather than continue plotting individual cases, sweep over SAT levels and financial need levels (i.e., houshold income levels) to create a lattice of plots.

```
sat_levels      <- c('le800','gt800le1000','gt1000le1200','gt1200le1400','gt1400')
income_levels   <- c('le30K','gt30Kle48K','gt48Kle75K','gt75Kle110K','gt110K','no_aid') %>% rev()
residence_state <- 'Ohio'
unthresh <- 0.4
student_value_all <-
  income_levels %>% lapply(
    function(income_bracket){
      sat_levels %>% lapply(
        function(sat_lvl){
          studentBF %>% select(-College,-outcome,-matches('Treasury|pell|Year2003')) %>%
            filter( Year == 2013 ) %>%
            left_join( student %>% select(Year,unitID, starts_with('NPT')), by = c('Year','unitID' ))
            left_join( epremium, by = 'unitID' ) %>%
            mutate(
              Utility = switch(
                sat_lvl,
```

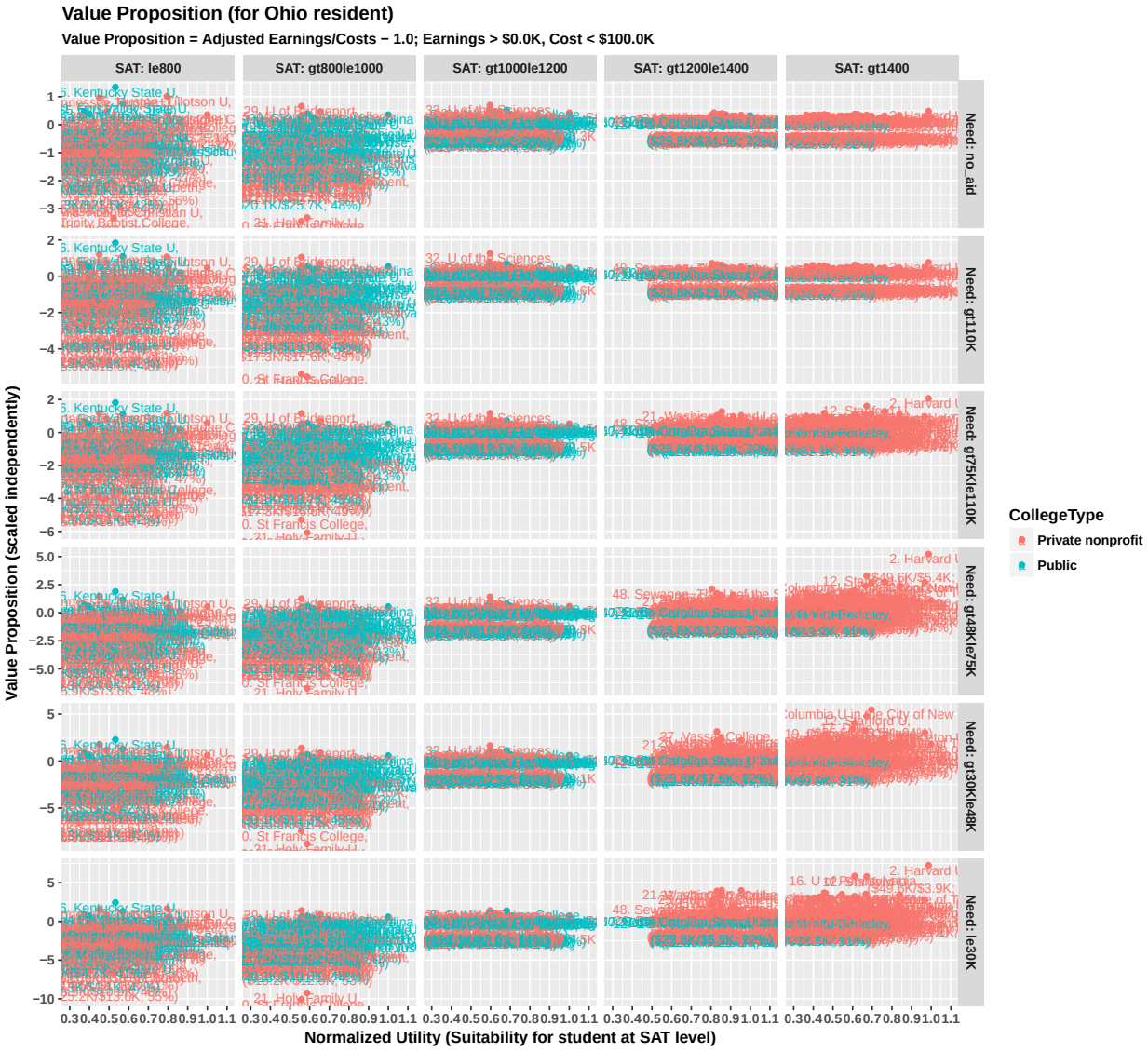
```
            gt1400       = BF_SAT_gt1400,
            gt1200le1400 = BF_SAT_gt1200le1400,
            gt1000le1200 = BF_SAT_gt1000le1200,
            gt800le1000  = BF_SAT_gt800le1000,
            le800        = BF_SAT_le800
          )
        ) %>%
        arrange( desc(Utility) ) %>%
        mutate(
          ntp1 = ifelse(grepl('Public',CollegeType),NPT41_PUB,NPT41_PRIV),
          ntp2 = ifelse(grepl('Public',CollegeType),NPT42_PUB,NPT42_PRIV),
          ntp3 = ifelse(grepl('Public',CollegeType),NPT43_PUB,NPT43_PRIV),
          ntp4 = ifelse(grepl('Public',CollegeType),NPT44_PUB,NPT44_PRIV),
          ntp5 = ifelse(grepl('Public',CollegeType),NPT45_PUB,NPT45_PRIV)
        ) %>%
        select( -starts_with('NPT4',ignore.case=FALSE)) %>%
        filter( !is.na(ntp1),!is.na(ntp2),!is.na(ntp3),!is.na(ntp4),!is.na(ntp5) ) %>%
        mutate(
          INSTATE = state == residence_state,
          Living_Expenses = COSTT4_A - TUITIONFEE_IN,
          maxcost = Living_Expenses + ifelse(INSTATE,TUITIONFEE_IN,TUITIONFEE_OUT),
          ntp  = switch(
            income_bracket,
            le30K       = ntp1,
            gt30Kle48K  = ntp2,
            gt48Kle75K  = ntp3,
            gt75Kle110K = ntp4,
            gt110K      = ntp5,
            no_aid      = maxcost
          ),
          Earnings_Adjusted = (Expected_Earnings + earnings_premium)*C150_4_POOLED_SUPP, # Assumes
          inccost = ifelse( INSTATE, ntp, ntp + TUITIONFEE_OUT - TUITIONFEE_IN),
          unorm = Utility/max(Utility), #10^(Utility-max(Utility)),
          cnorm  = inccost/Living_Expenses,
          vnorm = (Earnings_Adjusted/Living_Expenses - 1.0) / (Earnings_Adjusted[[1]]/Living_Expens
          value_prop = vnorm/cnorm, #unorm*vnorm/cnorm,
          coll_rank = order( Utility, decreasing = TRUE ),
          College = sprintf("%d. %s", coll_rank, gsub('University','U',College) )
        )  %>%
        #filter(value_prop2 < 4 & vnorm < 2 ) %>%
        mutate(
          cost = ntp,
          coll_label = sprintf("%s,\n($%.1fK/$%.1fK; %.0f%%)",gsub("^(.+)[^a-zA-Z]+Main Campus","\\
          SAT  = factor( sat_lvl, levels = sat_levels ),
          Need = factor( income_bracket, levels = income_levels )
        )
      }
    )
  }
) %>%
unlist( recursive = FALSE ) %>%
{ do.call( bind_rows, . ) } %>%
#filter( unorm>unthresh, value_prop2 < 2*value_prop2[[1]]) %>%
```

```
  # mutate(
  #   maxu = quantile(value_prop[unorm>unthresh],0.99),
  #   value_prop = value_prop/maxu  #ifelse(scale_independently,max(value_prop1),max(value_prop2)),
  # ) %>%
  filter( value_prop < 9 ) # To avoid outliers
```

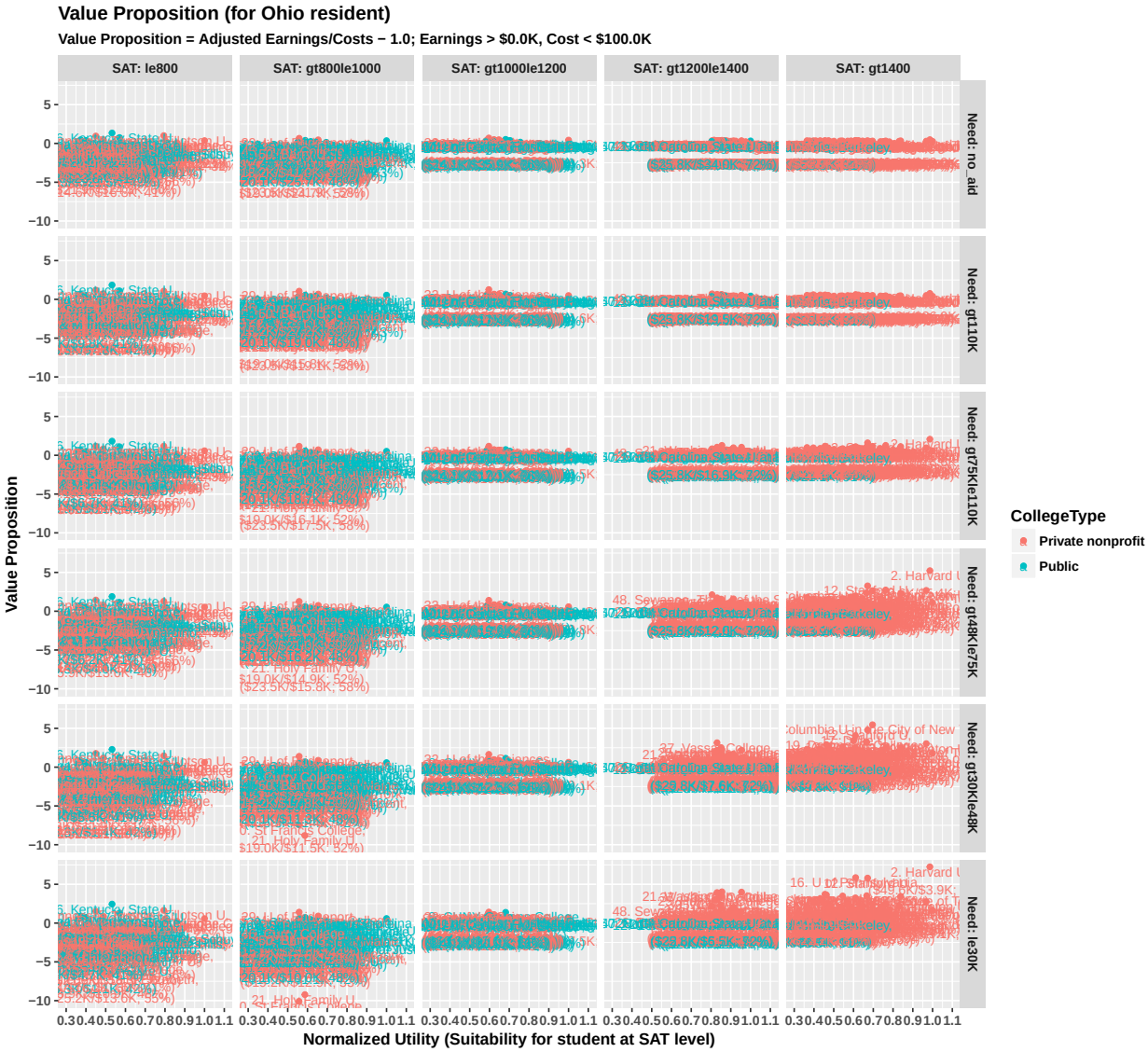
The following plots sweep across SAT level, as columns, and financial need level, as rows. Again, as above, the plots are for a resident of Ohio, so out-of-state tuition and fees apply for public schools outside of Ohio.

(You can fork the script and edit it to test with different states as that in which the student resides.)

```
scale_independently <- TRUE

rank_thresh <- 50
cost_thresh <- 100000
earnings_thresh <- 0.0

student_value_all %>%
  filter( coll_rank <= rank_thresh ) %>%
  group_by( SAT ) %>%
  filter( coll_rank<=10 | (coll_rank>10 & Earnings_Adjusted>earnings_thresh & cost < cost_thresh)  ) %>%
  ungroup() %>%
  {
    ggplot(., aes( x = unorm, y = value_prop, color = CollegeType ) ) +
      geom_point( na.rm = TRUE ) +
      scale_x_continuous(limits = c(unthresh-0.1,1.1), breaks = seq(unthresh-0.1,1.1,by=0.1)) +
      geom_text( aes( label = coll_label), vjust = 1.0, size = 3, na.rm = TRUE ) +
      ggtitle(
        label    = sprintf('Value Proposition (for %s resident)',residence_state),
        subtitle = paste(
          'Value Proposition = Adjusted Earnings/Costs - 1.0',
          sprintf('Earnings > $%2.1fK, Cost < $%2.1fK', earnings_thresh/1000, cost_thresh/1000),
          sep = "; "
        )
      ) +
      labs(
        x = 'Normalized Utility (Suitability for student at SAT level)',
        y = ifelse(scale_independently,'Value Proposition (scaled independently)','Value Proposition')
      ) +
      theme( text = element_text( face = 'bold' ) ) +
      facet_grid(
        Need ~ SAT,
        scales = ifelse(scale_independently,'free','fixed'),
        labeller = label_both
      )
  }
```



Now the same plot but with the y-axes fixed.



Finally, plot the lattice one more time, allowing the y-axis of every plot to move independently.

Interpretation

The labeling of the rows by "Need" includes the income bracket to which the student's household belongs. In other words, the panels in that row correspond to the value proposition for a student whose need level is typical for a household of that income bracket. The top row corresponds to a need level of "no_aid", meaning the household is wealthy enough to not qualify for any financial aid and would pay the full cost of tuition, fees and living expenses.

Note that within a row, the y-coordinate of the college, which is the value proposition estimated for any specific college given the financial need level specified for that row, does not change. That y-value for a given college stays the same as SAT level changes because the inputs to the value proposition equation – the adjusted earnings, the costs at that row's need level, and the completion rate – are only properties of the college and not the student and therefore they all stay the same as the student's SAT level changes. But a different mix of colleges appears in each panel within a row because only the top-100 colleges, in terms of suitability as determined by the Bayes factor at that row's SAT level, are included in each panel, and they change with change in SAT level.

Also note, that within a column, the x-coordinate of a college, which is the suitability of the school for a student scoring at the SAT level specified for that column, does not change. Of course the y-values do change because the estimated costs, which is the denominator in the value proposition equation, decreases with

increasing financial need.

You can see that the value proposition stays roughly the same as we increase SAT level, moving across the columns from left to right. Also, as student SAT level increases, the mix of suitable colleges transitions from many public colleges to primarily the elite private colleges, like the Ivies, MIT, CalTech, Stanford, etc.The fact that the mix of colleges also changes with SAT level indicates that for each level of student ability, there is a set of colleges that fit right into the economic niche to serve them. However, there is a bit of a dip in value proposition in the middle column, corresponding to students from households with incomes near the median American income. This issue might be a cause for concern in our nation. . . .

Also, the adjusted earnings (shown in the data point labels) for the colleges populating the panels in the leftmost columns are significantly lower than those for the colleges populating the panels in the rightmost columns. This indicates that the expected earnings for low-SAT students are dramatically lower than those of high-SAT students. (Because we adjust the earnings to account for the SAT distribution at each college, this drop in adjusted earnings for the low-SAT students is driven mainly by dramatically lower completion rates at the colleges most suitable for them that at the colleges for high-SAT students.)

The value proposition increases dramatically as we increase the financial need level, moving down the rows from top to bottom. As more and more financial aid kicks in, the costs of the colleges drop significantly (while adjusted earnings and completion rates for each school stay constant). In fact the value proposition for a high-achieving student, at SAT level greater than 1400, who comes from a household with great financial aid, at income level less than $30,000/yr, the Ivy League schools pose an unparallel value proposition at the highest suitability levels for that student.

Caveats

The underlying dataset and the methods applied here really **are not** suitable for drawing any strong conclusions about the value proposition of a specific college for a specific student. This analysis merely demonstrates some of the considerations one might need to address in investigating the value proposition of colleges. It would be interesting to see how the quick-and-dirty results shown here compare to a more rigorous analysis and to published sources of value comparisons and value rankings of colleges.

Hope you found this to be interesting and it prompts you to dig deeper and to do more. . . .

Copyright Notice