

# College Scorecard: Earnings Premium & Value Proposition

*Michael L. Thompson*

*January 2, 2017*

## Contents

Introduction . . . . .	1
Earnings Premium . . . . .	1
Value Proposition . . . . .	2
Data Preparation . . . . .	2
Feature Engineering: Bayes Factors . . . . .	3
Estimating the Probabilities . . . . .	3
Earnings Premium and Value Proposition Calculation . . . . .	4
Model to Predict Expected Earnings . . . . .	4
Plots of the Earnings Premium . . . . .	8
Value Proposition . . . . .	10
Case: High-SAT, High-Income (i.e., Low-Financial-Need) . . . . .	12
Case: Low-SAT, High-Income (i.e., Low-Financial-Need) . . . . .	13
Case: High-SAT, Low-Income (i.e., High-Financial-Need) . . . . .	14
Value Proposition Given Student's SAT Level and Financial Need Levels . . . . .	15
Interpretation . . . . .	20
Caveats . . . . .	21
Copyright Notice . . . . .	21

## 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 a quick-and-dirty stab at an analysis that serves as an exploratory data analysis. It's 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, and this analysis is too crude to capture the earnings component with the the rigor and reliability demanded of academic scholarship.**

Think of this analysis as a means of raising questions as the basis for proposing hypotheses to be tested by serious academic research.

## Earnings Premium

**The earnings premium for a college** is the *earnings in excess of the average annual earnings at 6 years after matriculation* after controlling for the following factors:

- (1) the academic abilities of the students at the college,
- (2) the academic discipline distribution of Bachelor's degrees of the students graduating from the college, and
- (3) the geographic location and campus locale of the university.

Academic ability is measured by SAT. Academic discipline, by percentages of students receiving each type of degree. And, geographic location, 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).

## Value Proposition

**The value proposition of a college** is defined as *the ratio of the adjusted annual earnings divided by the expected annual costs of attendance*.

The adjusted annual earnings is computed by (1) adding the earnings premium to the overall mean (across all colleges) of annual earnings 6 years after matriculation, and (2) multiplying this sum by the completion rate for the college to adjust for the probability of actually achieving a degree from the college.

The annual costs are computed 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).

## 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.

```
##      name
## Scorecard

## Warning: package 'bindrcpp' was built under R version 3.4.1

## Warning in rsq_lite_fetch(res@ptr, n = n): Column `Treasury_pell_ever`:
## mixed type, first seen values of type real, coercing other values of type
## string

## Warning in rsq_lite_fetch(res@ptr, n = n): Column `Treasury_md_earn_wne_p6`:
## mixed type, first seen values of type integer, coercing other values of
## type string

## Warning in rsq_lite_fetch(res@ptr, n = n): Column `Treasury_pell_ever`:
## mixed type, first seen values of type real, coercing other values of type
## string

## Warning in rsq_lite_fetch(res@ptr, n = n): Column `Treasury_md_earn_wne_p6`:
## mixed type, first seen values of type integer, coercing other values of
## type string

## Warning in rsq_lite_fetch(res@ptr, n = n): Column `Treasury_md_earn_wne_p8`:
## mixed type, first seen values of type real, coercing other values of type
## string

## Warning in rsq_lite_fetch(res@ptr, n = n): Column `C150_4_POOLED_SUPP`:
## mixed type, first seen values of type real, coercing other values of type
## string

student %<>%
  filter( unitID %in% unitIDs ) %>%
  select( -contains('SATWR') ) %>%
  group_by(Year) %>%
  arrange(unitID) %>%
  do( (.) %>% mutate(College = colleges[paste0('u_',unitID)]) ) %>%
```

```
ungroup()
```

```
# # add an indicator variable for presence of at least one standardized test statistic:  
# a<- student %>% dplyr::select(matches('^(SAT|ACT)')) %>% apply(1,function(x) !all(is.na(x)))  
# student %<>% mutate(stdzdtest=a)
```

## Feature Engineering: Bayes Factors

Approximate Bayes factors serve as the features of the model. The Bayes factor 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 in this case is the attribute of a student. For example, “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.)

Compute the following for the feature corresponding to attribute `Attribute_Y` for each college of interest `College_X` (we use log-base 10 on the Bayes factor.):

```
BF_log10( hypothesis = College_X | evidence = Attribute_Y )      = log10( Odds( hypothesis  
= College_X | evidence = Attribute_Y )/Odds( hypothesis = College_X ) )      = log10( P(  
evidence = Attribute_Y | hypothesis = College_X )/P( evidence = Attribute_Y | hypothesis  
= NOT(College_X) ))      ~ log10( P( evidence = Attribute_Y | hypothesis = College_X )/P(  
evidence = Attribute_Y )
```

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).

Above, the flip in the propositions on each side of the conditional (“|”) when we go from `Odds(H|E)` to probabilities, `P(E|H)`, occurs by applying Bayes’ Rule.

## Estimating the Probabilities

For some of the probabilities used in approximating the Bayes factors, the probabilities `P( evidence = Attribute_Y | hypothesis = 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 `P( evidence = Attribute_Y | hypothesis = College_X )` by simply computing the upper tail probability of SAT with lower limit 1400. Finally, to compute `P( evidence = 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 1100+ college universe. In the database, field `UGDS` is the number of undergraduate students attending the college. So the proportion of all students attending `College_X` is `P( hypothesis = College_X ) = UGDS[College_X]/sum( UGDS[College_X] )`.

For attributes like region and locale, to estimate  $P(\text{evidence} = \text{Attribute\_Y} \mid \text{hypothesis} = \text{College\_X})$ , 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.

```
## Loading required namespace: DT
```

## 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 matrix...
      ihi <- which(abs(fcor[seq_len(which(nms==nm)), nm]) > 0.8)
    }
  )

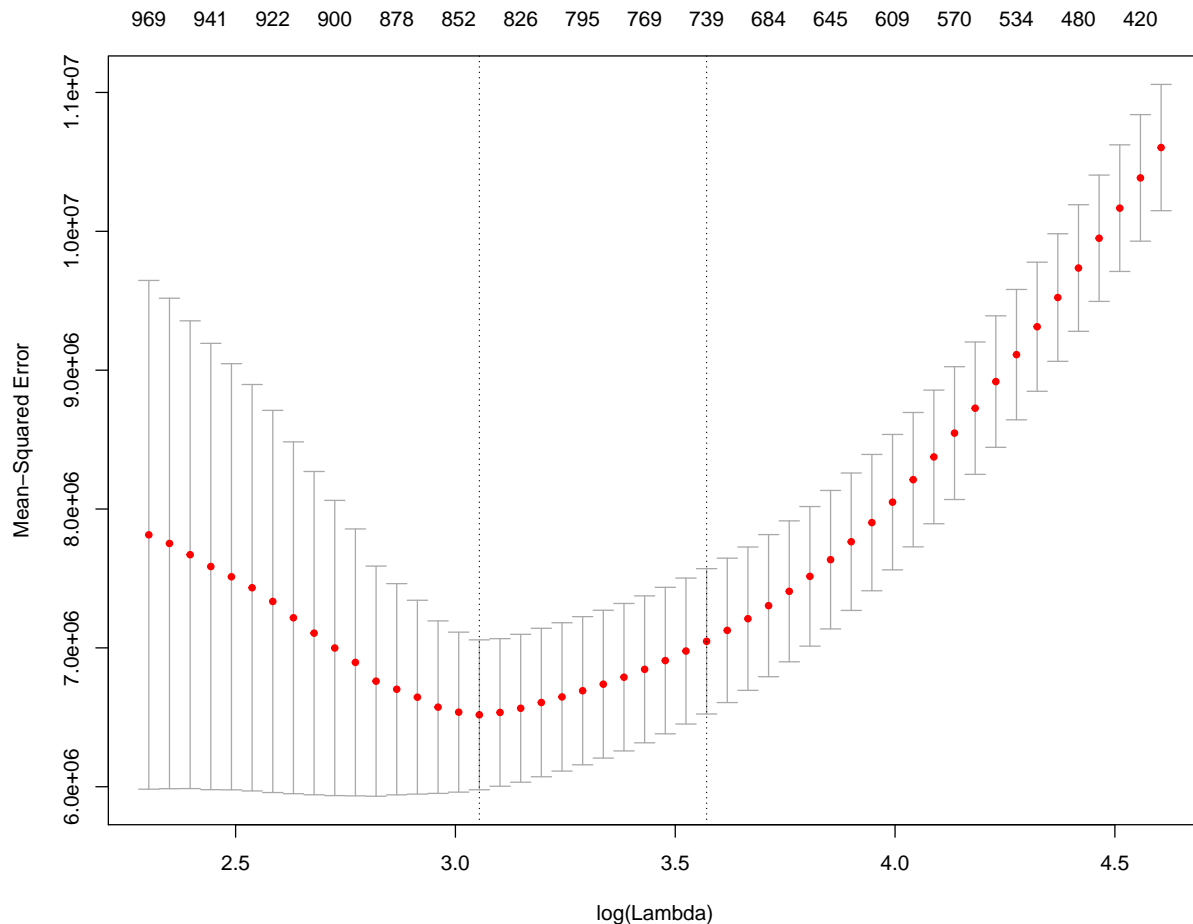
```

```

    setNames(fcor[ihi,nm],nms[ihi])
  }
)
hicor      <- hicor[sapply(hicor,length)>0]

# 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(
  model_mat[,setdiff(colnames(model_mat),names(hicor))], #glmdata %>% select(-outcome) %>% as.matrix(),
  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 t
# 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
```

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)))))
)

```

```

# 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())
  ) %>%
  inner_join( dominant_discipline %>% select(-College), by = c('Year','unitID') ) %>%
  select(Year,unitID,College,outcome,predicted_earnings,dominant_discipline)

```

```

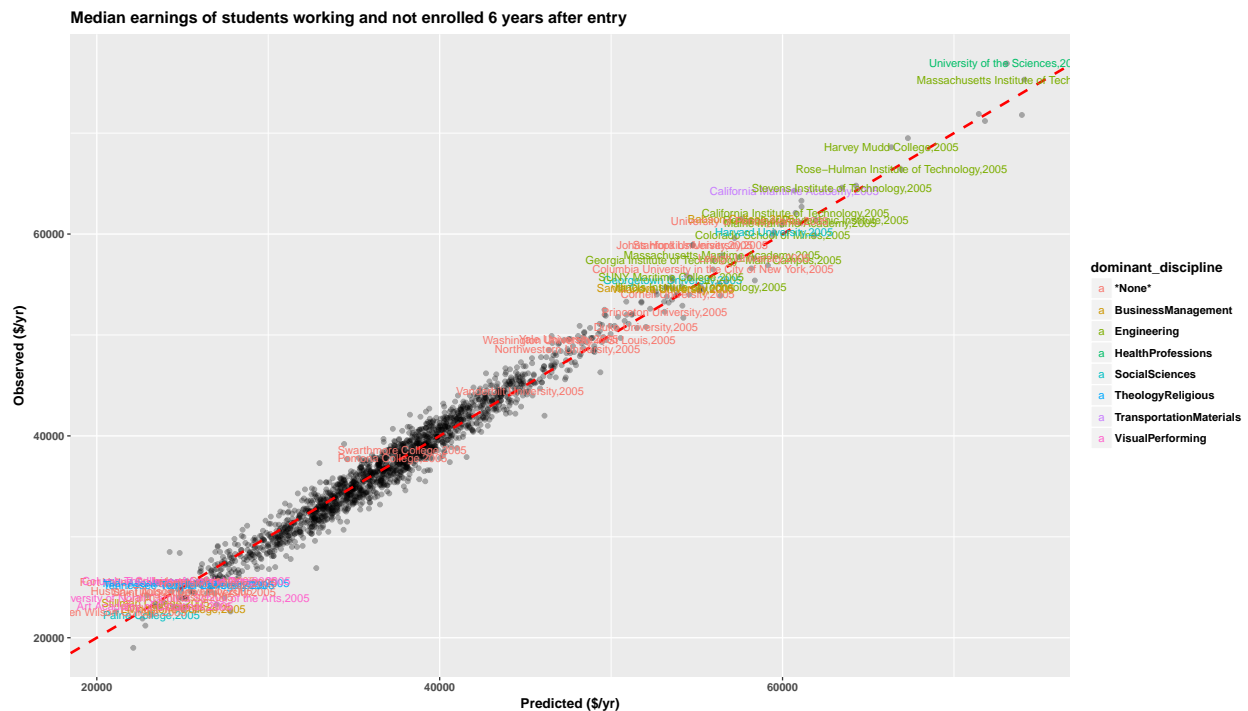
select_colleges <- c('Harvard','Princeton','Yale','Stanford','(Cornell U)','(^Univ.+Pennsylv[a-v]+$)',
  '(Northwestern U)','Duke','Vanderbilt','Swarthmore','(Pomona C)','Johns Hopk','Wash
predictions %>%
  mutate( College = sprintf("%s,%d",College,Year) ) %>%
  {
    ggplot(., aes(x=predicted_earnings,y=outcome)) +
      geom_point(alpha=0.3) +
      geom_abline(intercept = 0, slope = 1, color = 'red', linetype = 2, size = 1) +
      geom_text(
        data = . %>%
          filter(
            Year==max(Year),
            outcome>quantile(outcome,0.975)|
            outcome<quantile(outcome,0.025)|
            grepl(paste(select_colleges,collapse='|'),College)),
        mapping = aes(label = College, color=dominant_discipline ),
        size = 3
      ) +
  }

```

```

ggtitle('Median earnings of students working and not enrolled 6 years after entry') +
labs(
  x = 'Predicted ($/yr)',
  y = 'Observed ($/yr)'
) +
theme( text = element_text( face = 'bold' ) )
} %>%
print()

```



```

epremium <- df_coef %>%
  filter(grepl('^College',Coefficient)) %>%
  mutate(
    unitID = gsub('.+ID([0-9]+)__', '\\1',Coefficient) %>% as.integer(),
    College = gsub('^College_ID.+__', '',Coefficient)
  ) %>%
  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

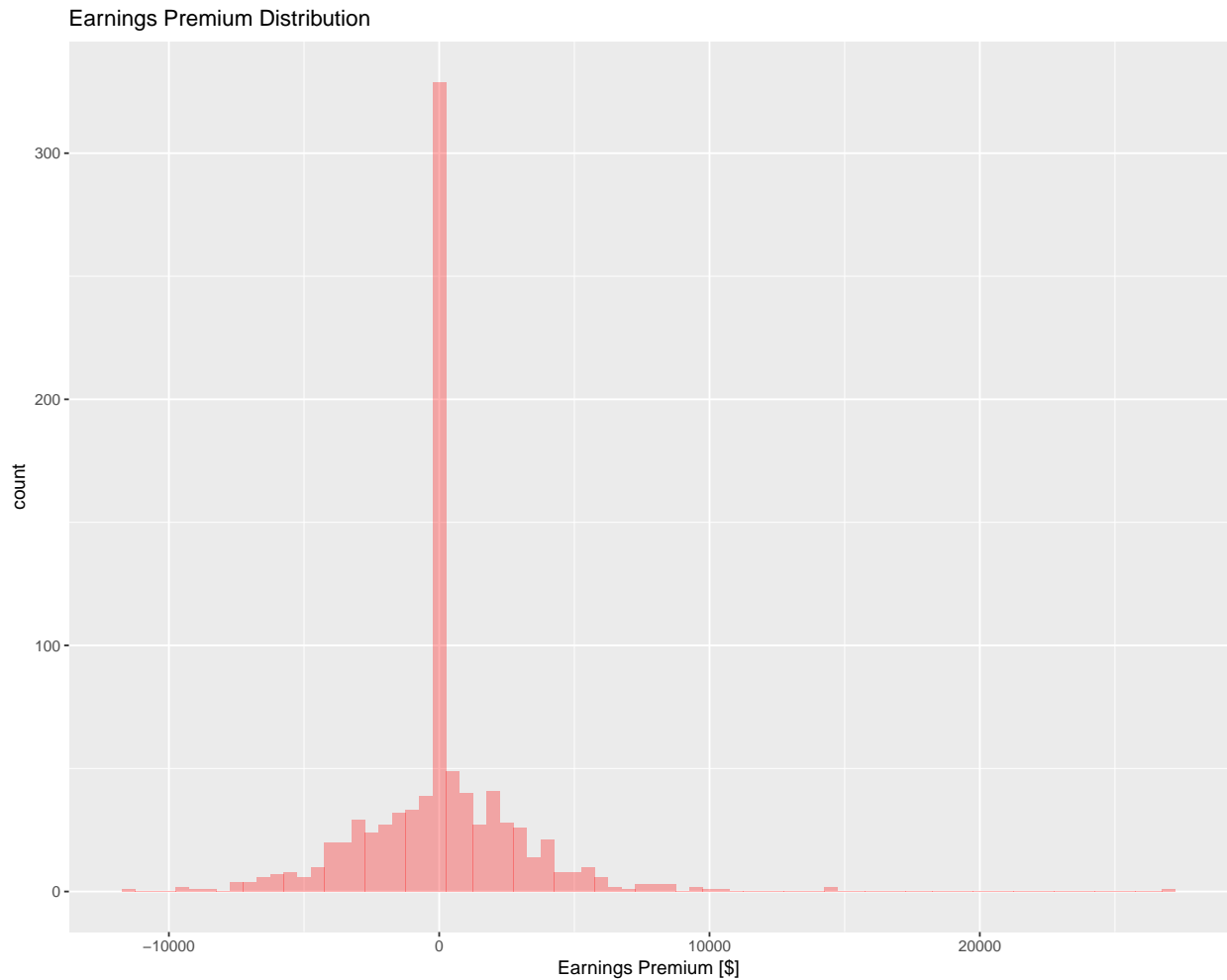
First, plot the histogram of earnings premia, demonstrating, as mentioned above mixture of zero and non-zero premia.



```

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.

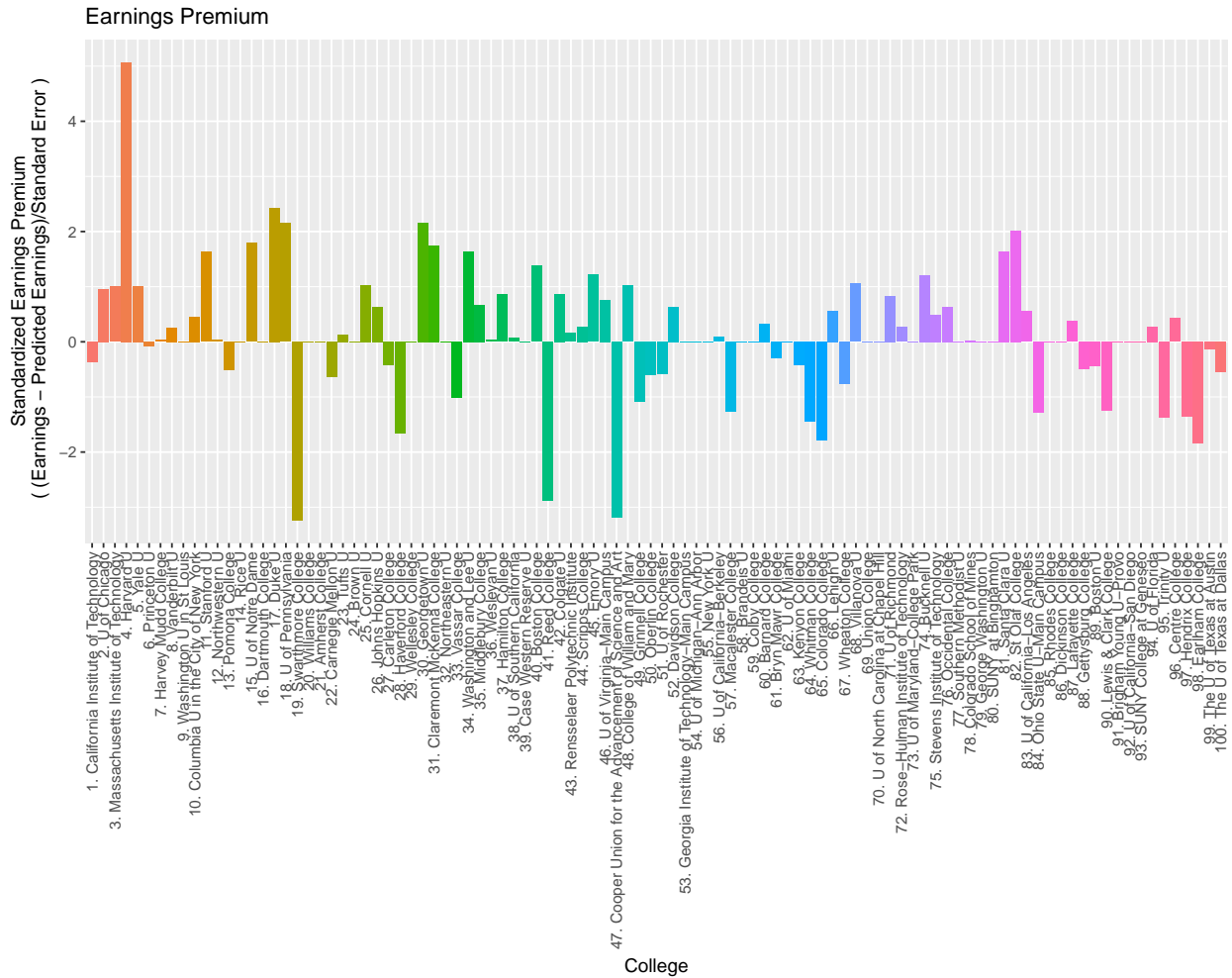
(Interestingly, it still looks like some of the more tech-oriented colleges have positive premia while more liberal-arts-oriented colleges have negative premia, even though we explicitly attempted to model the impact of degree distribution at each college. Probably an illustration of the limitations 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='g
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) / (Earnings_Adjusted[[1]]/Living_Expenses[[1]]), # Ea
    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 & unorm < 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),Earnings
  ) %>%
  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',
        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
)

```

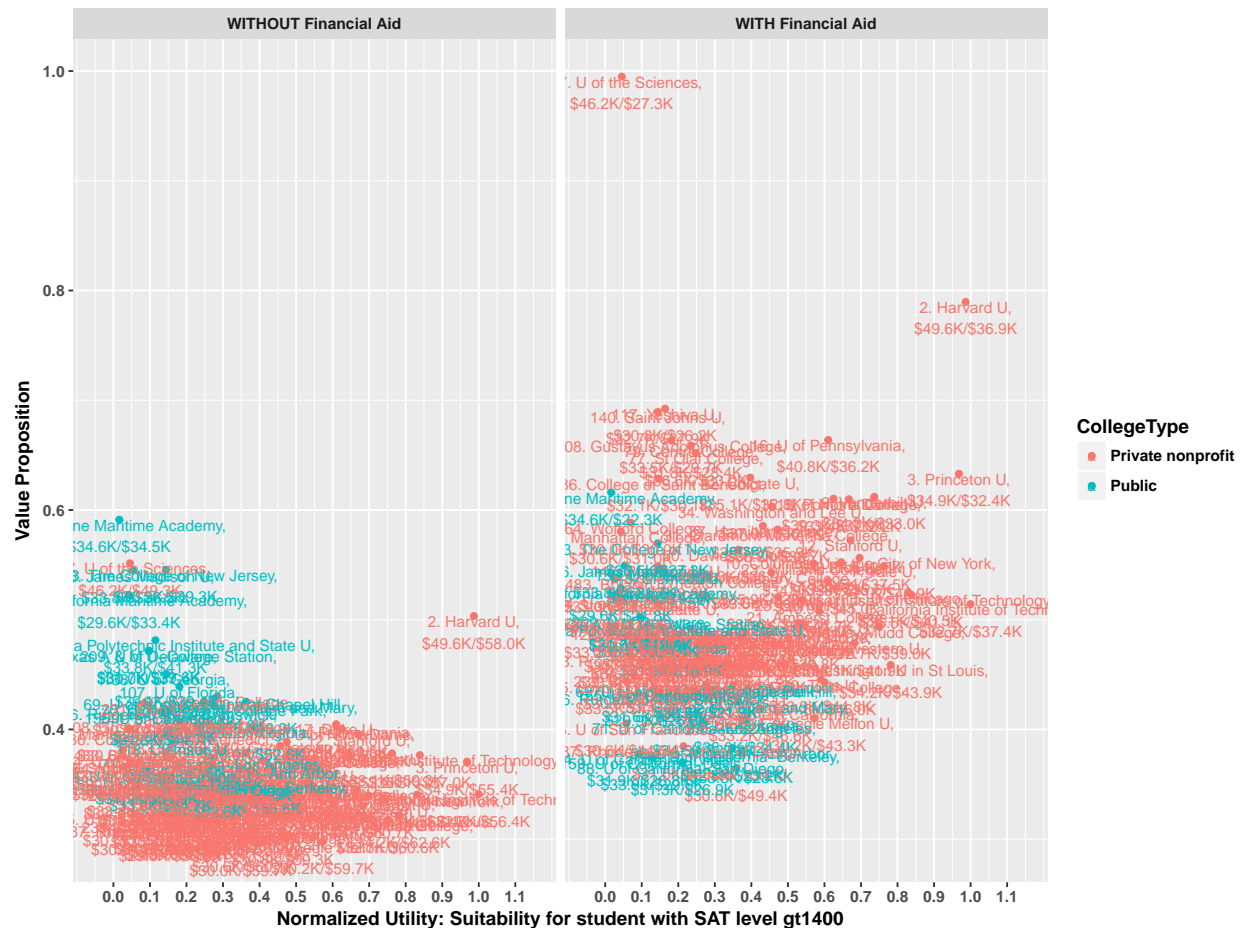
```

)
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
  )

```

### Value Proposition With & Without Financial Aid (for Ohio resident, SAT = gt1400, Income = gt110K)

Value Proposition = Adjusted Earnings/Costs; Earnings > \$29.6K, Cost < \$70.0K

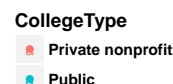


### Case: Low-SAT, High-Income (i.e., Low-Financial-Need)

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.

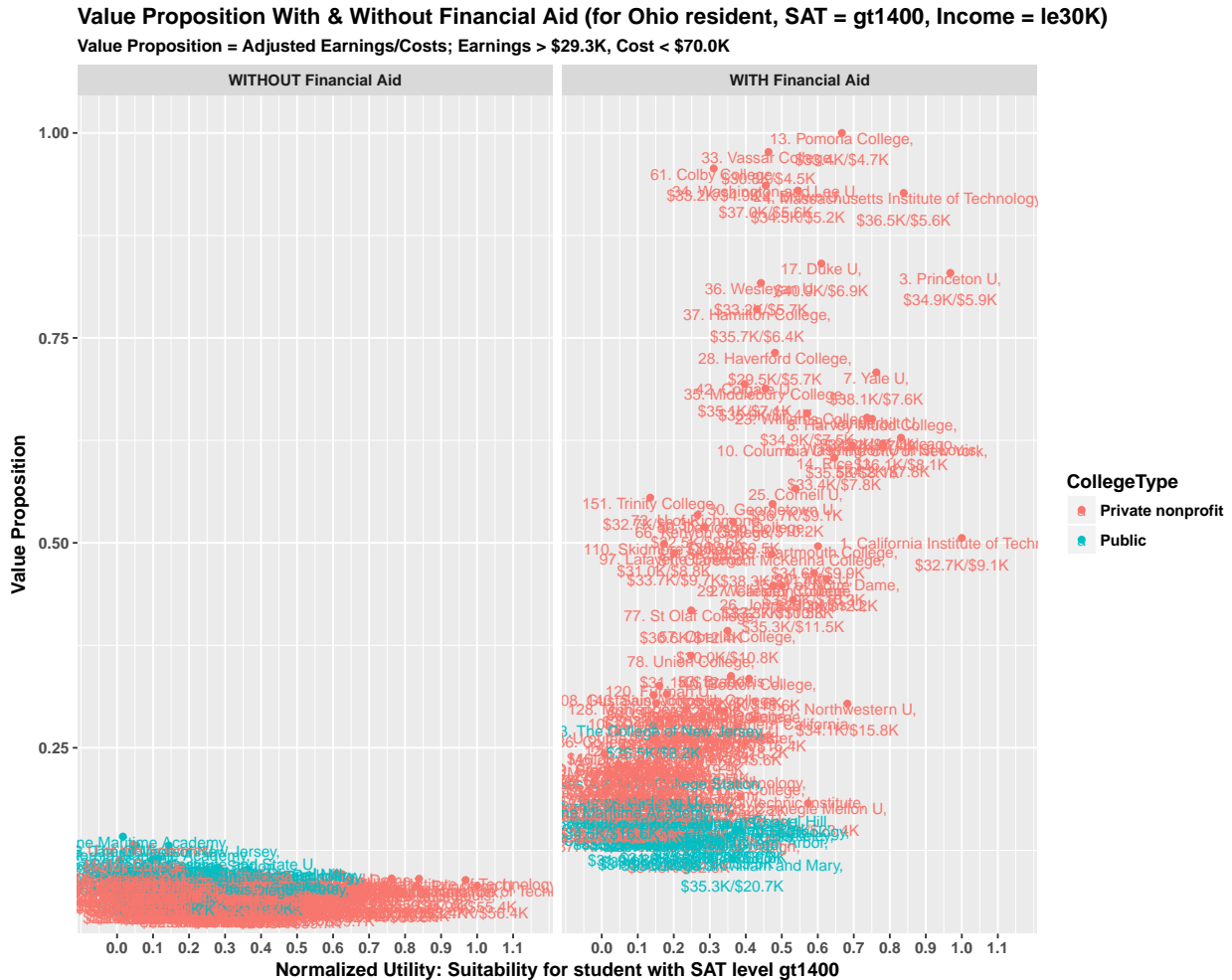
Notice how the adjusted earnings shown in the data point labels are significantly lower than those of the previous plot: In aggregate, the academic ability of a college's students is a strong predictor of the future earnings of the students. (Says nothing about any individual's case.) Also, see how financial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

Value Proposition = Adjusted Earnings/Costs: Earnings > \$7.3K, Cost < \$70.0K



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.

14



### 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., household 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) / (Earnings_Adjusted[[1]]/Living_Expenses[[1]]),
    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", "\\1"),
    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(
#   maxv = quantile(value_prop[unorm>unthresh],0.99),
#   value_prop = value_prop/maxv #ifelse(scale_independently,max(value_prop1),max(value_prop2)),
# ) %>%
filter( value_prop < 10 ) # 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

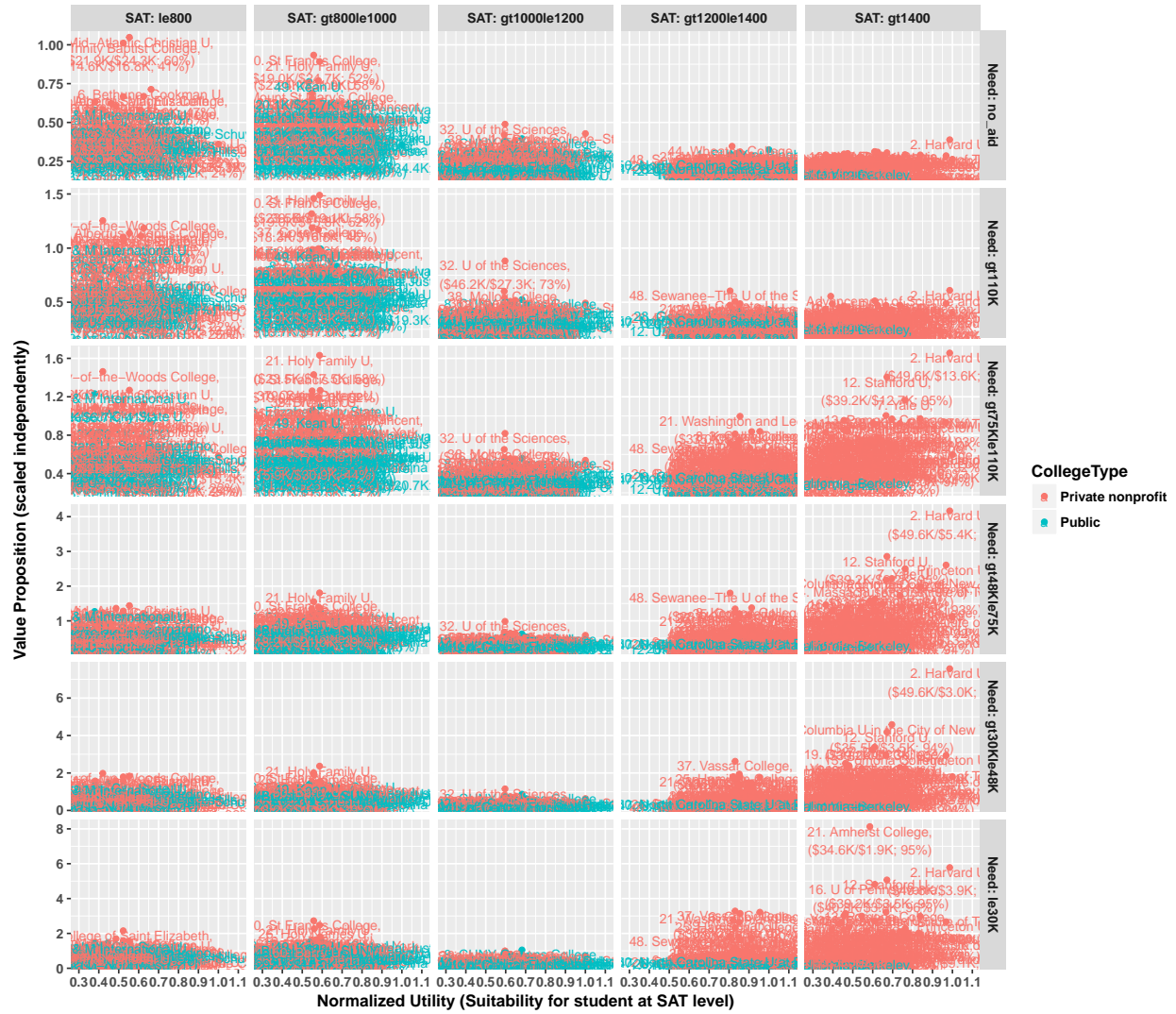
rank_thresh <- 50
cost_thresh <- 100000
earnings_thresh <- 0.0

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',
          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
      )
  }

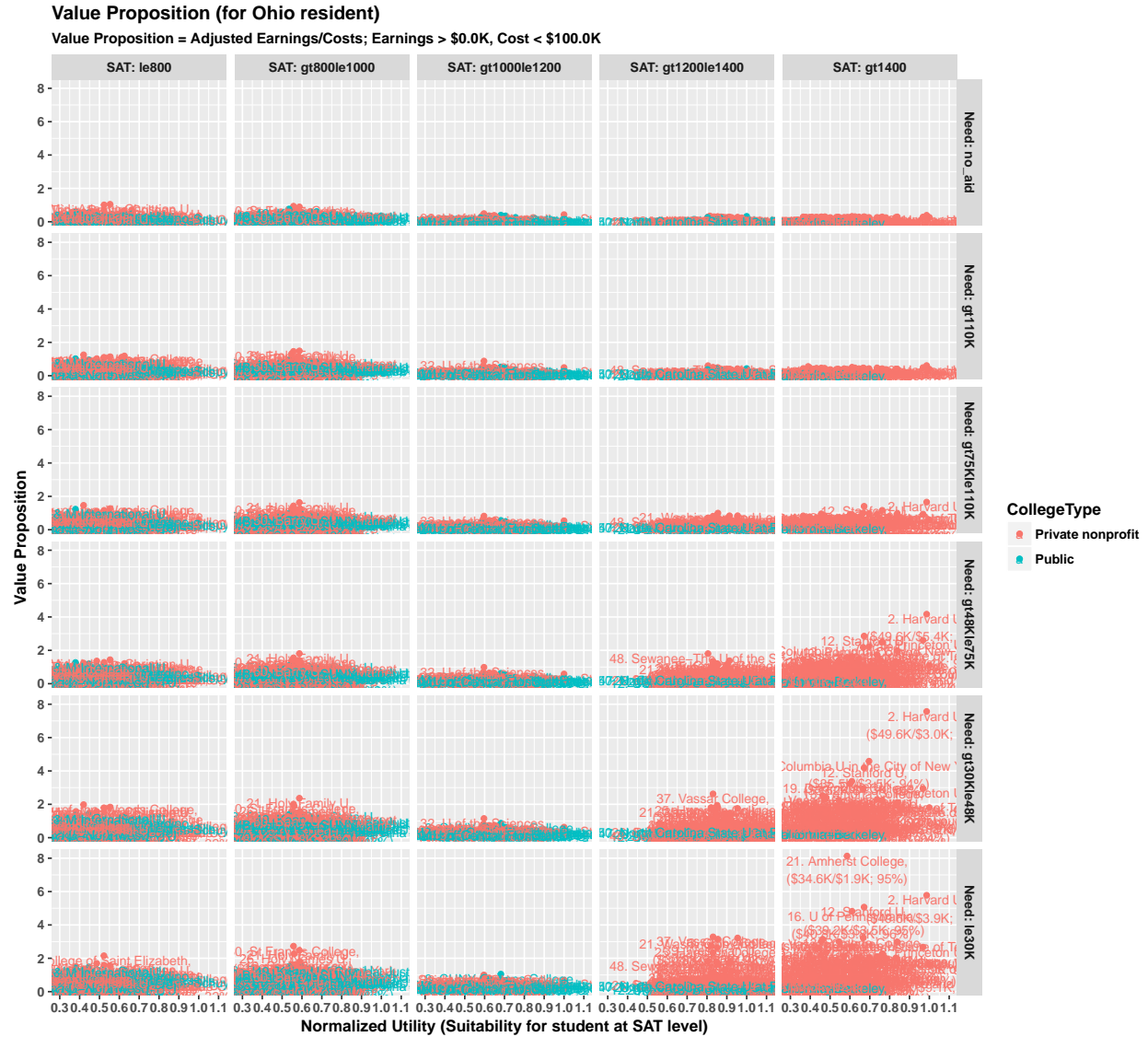
```

# Value Proposition (for Ohio resident)

Value Proposition = Adjusted Earnings/Costs; Earnings > \$0.0K, Cost < \$100.0K



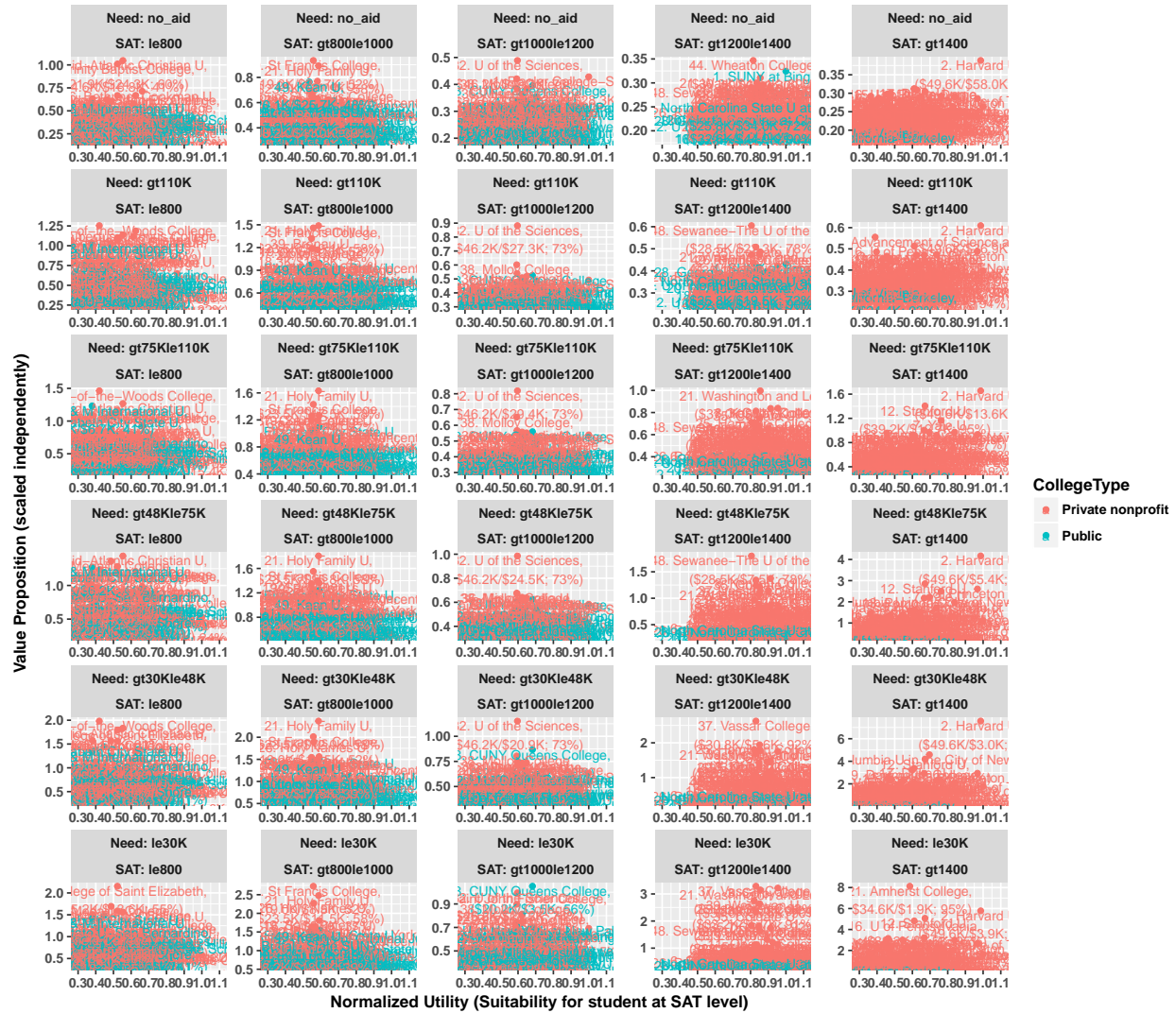
Now the same plot but with the y-axes fixed.



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

## Value Proposition (for Ohio resident)

Value Proposition = Adjusted Earnings/Costs; Rank <= 50, Earnings >=top-quartile, Cost < \$100.0K



## 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 than 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

---

Copyright 2017 Michael L. Thompson

---

“collegeValue.Rmd” reapplies portions of “BestCollegeforYou\_KaggleSubmission.Rmd” originally submitted to the kaggle.com competition “US Dept of Education: College Scorecard” under the Apache License, v. 2.0. at <https://www.kaggle.com/apollostar/d/kaggle/college-scorecard/which-college-is-best-for-you>

---

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---