

# Big Data Course Project

## Registrar Data Analysis

Tianzhixi Yin, Damian Stansbury, Dongyang Kuang

May, 3, 2014

# In the beginning...

## Our Goals

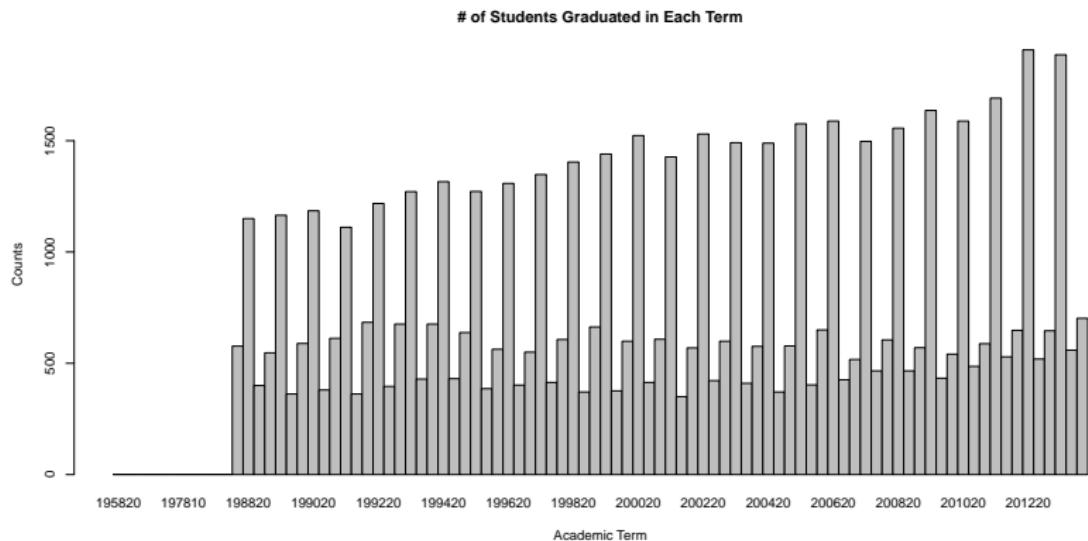
- Dig out meaningful information from the dataset.
- Make it easy for others to utilize our methods.

# What is the most important issue in Data Analysis?

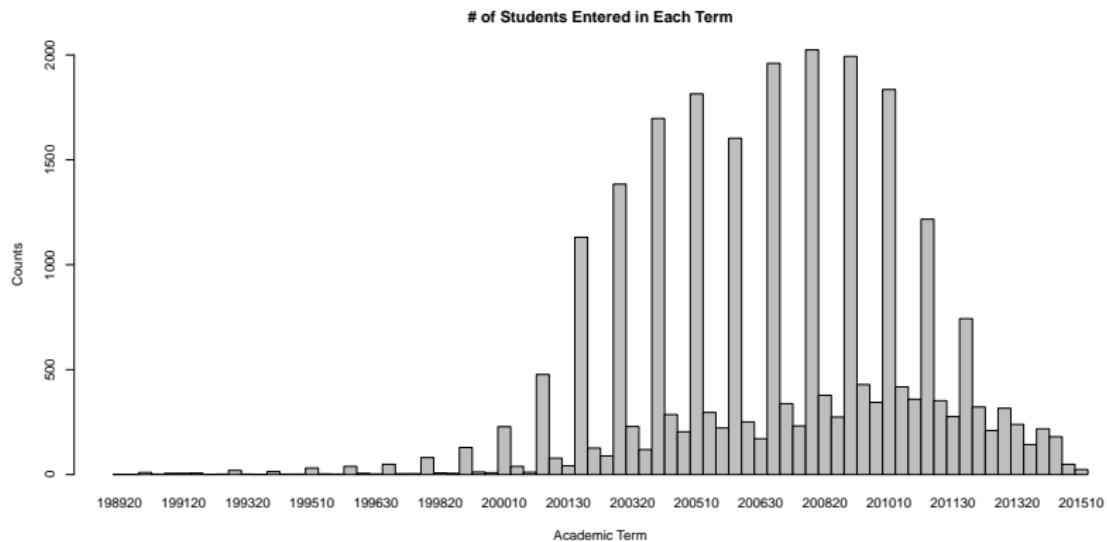
## Our Major Questions

- What are the factors influencing students' performance?
- What geographic areas and high schools are sending good students to UW?
- Is Financial Aid really helping the students?

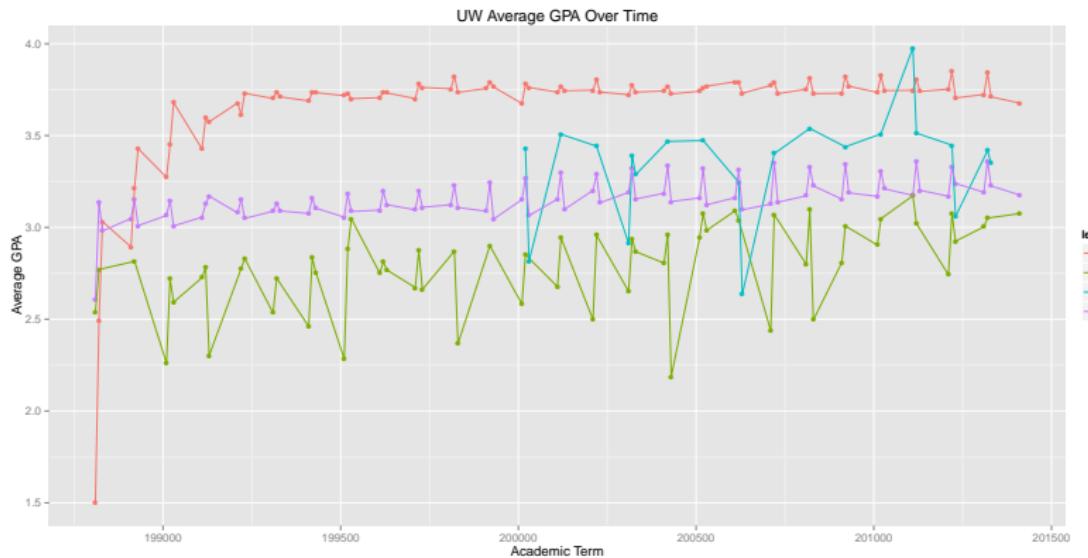
# How many students have graduated from UW?



# Are more and more students coming to UW?



# Has GPA changed over time?



# What truncated regression model thinks are most important

Predictors	Log-Likelihood
SAT Total	-2104

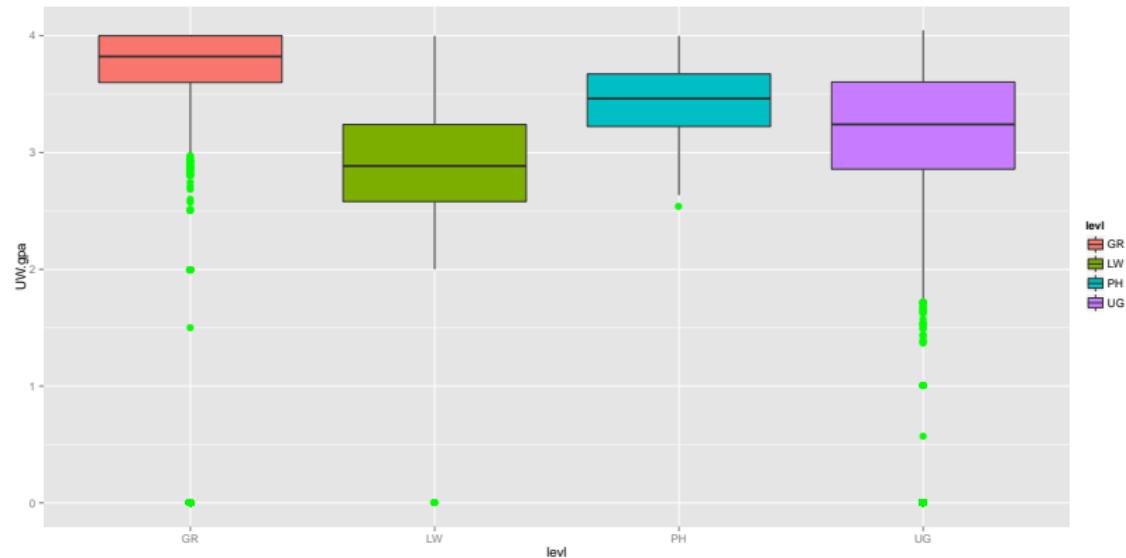
# What truncated regression model thinks are most important

Predictors	Log-Likelihood
SAT Total	-2104
Highschool GPA	-5891

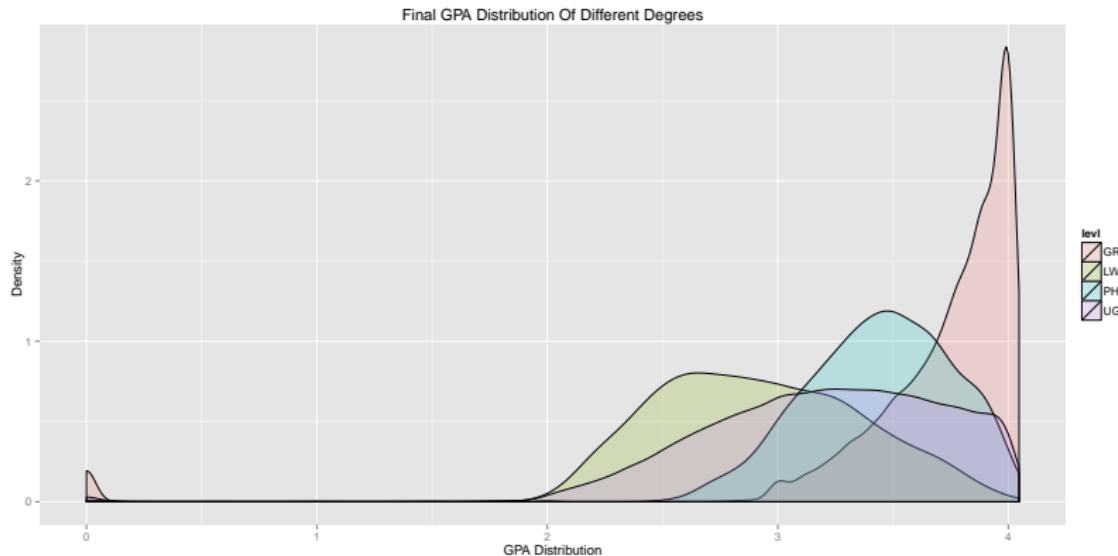
# What truncated regression model thinks are most important

Predictors	Log-Likelihood
SAT Total	-2104
Highschool GPA	-5891
ACT Composite	-13021
Major	-36215
Level	-37922
College	-41629
Marital Status	-42596
Sex	-42334
Highschool State	-43207
Race	-43110
Veteran	-43354

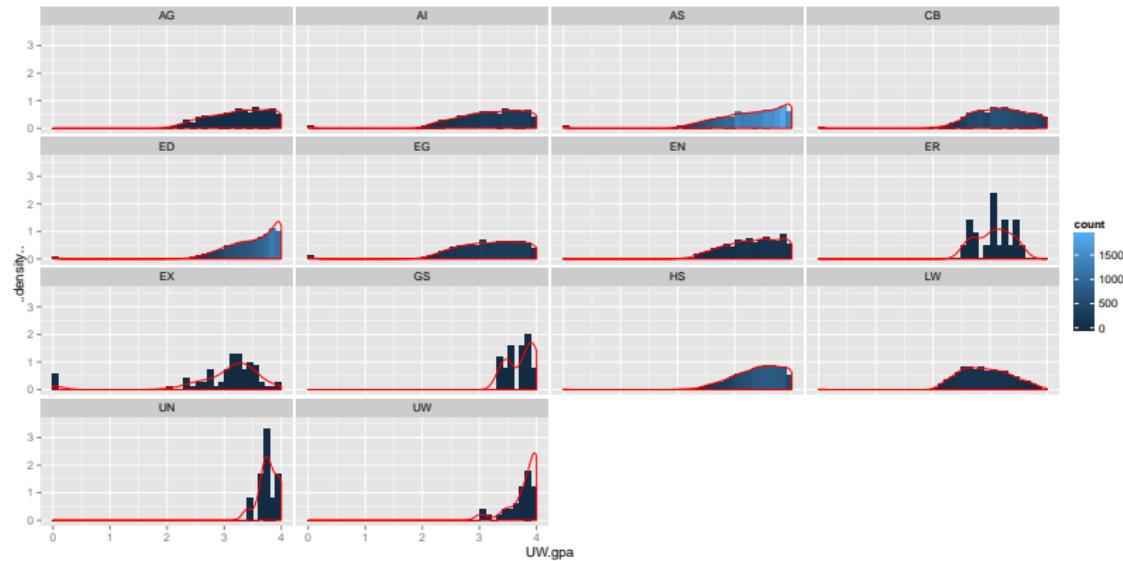
## Boxplot of GPA in different degrees



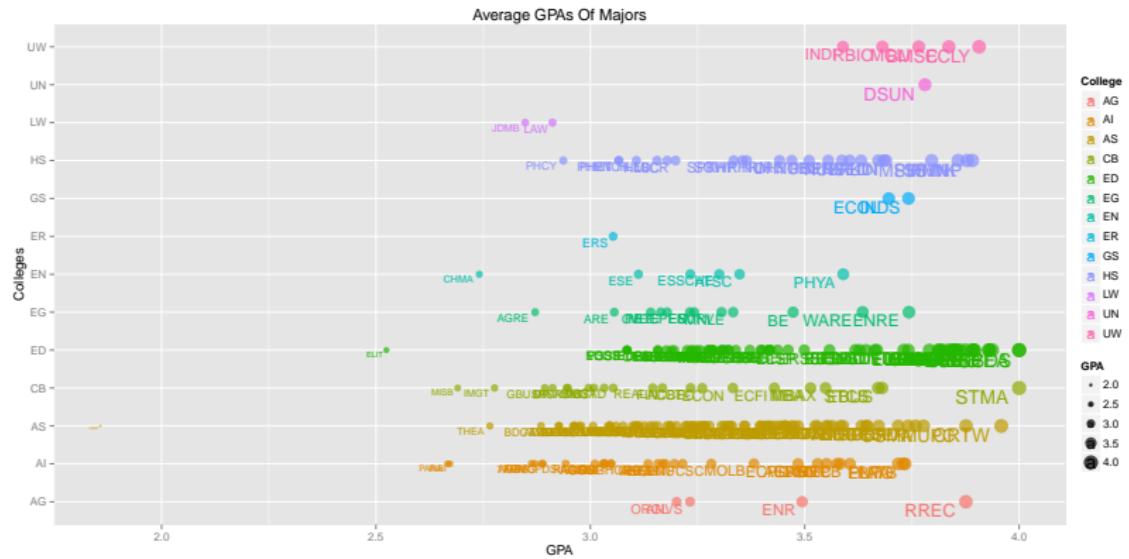
# A more intuitive look



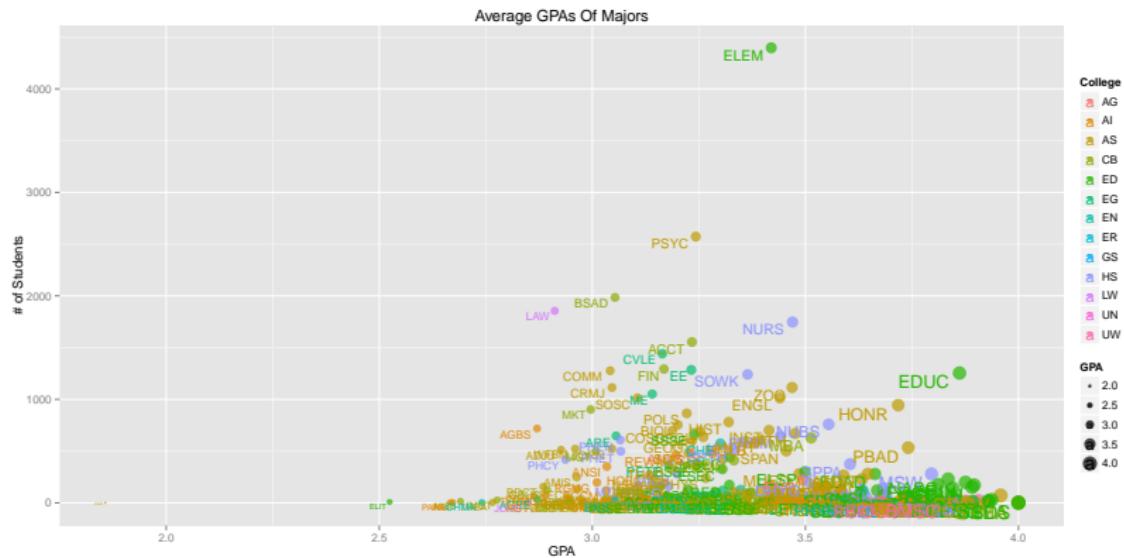
# Different colleges



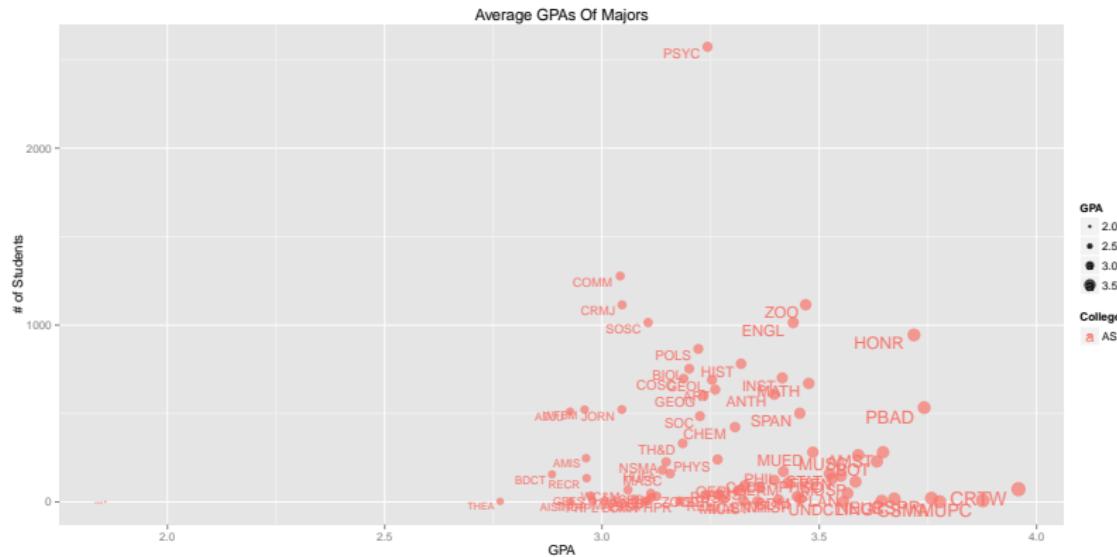
# Different majors



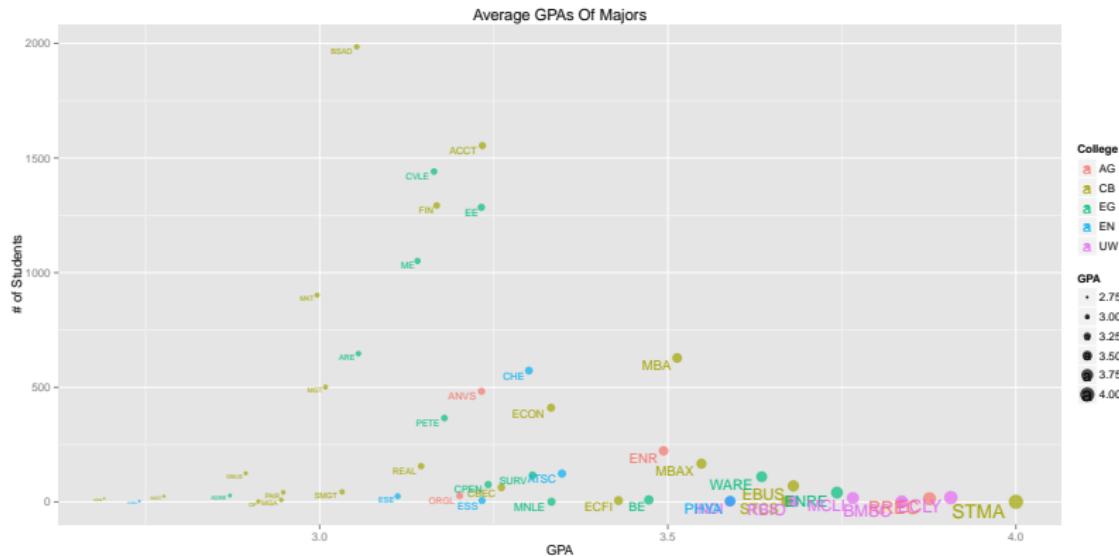
# Considering the number of students



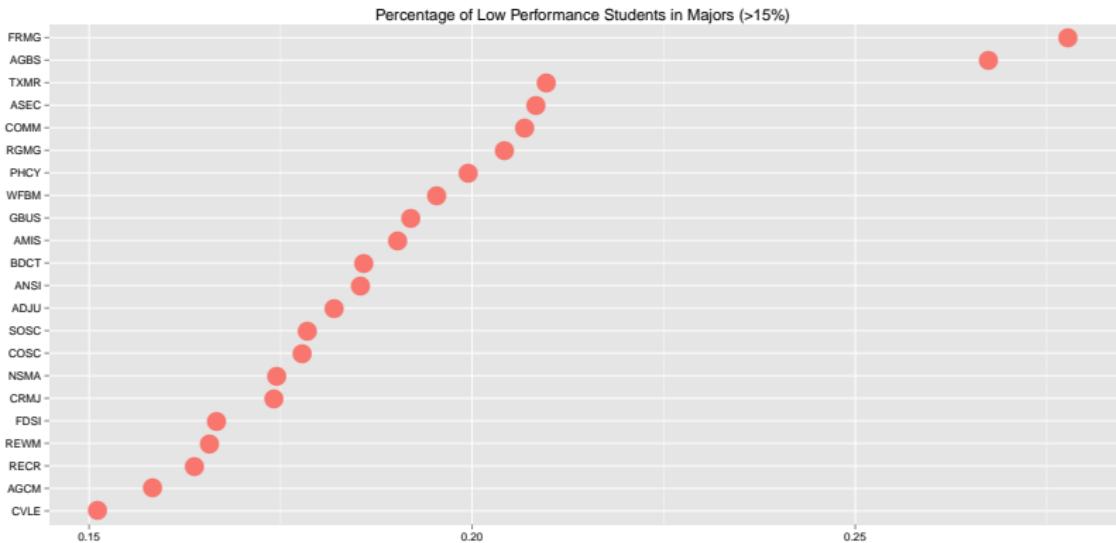
# Only looking at Arts and Science majors



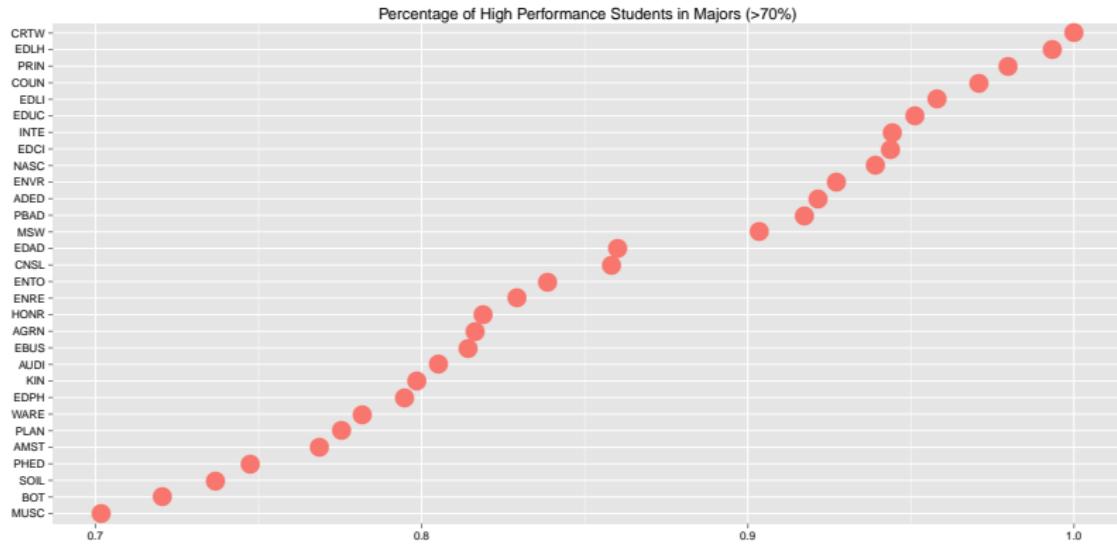
# Colleges with few majors



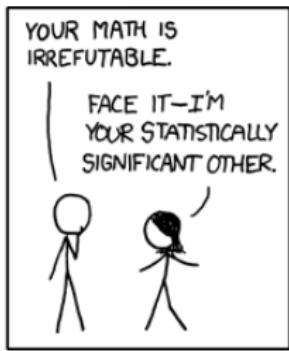
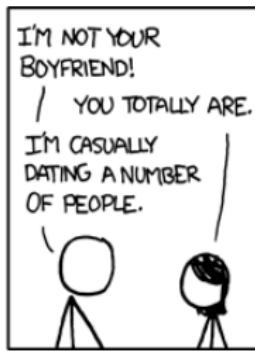
# Hardest majors? (< 2.5)



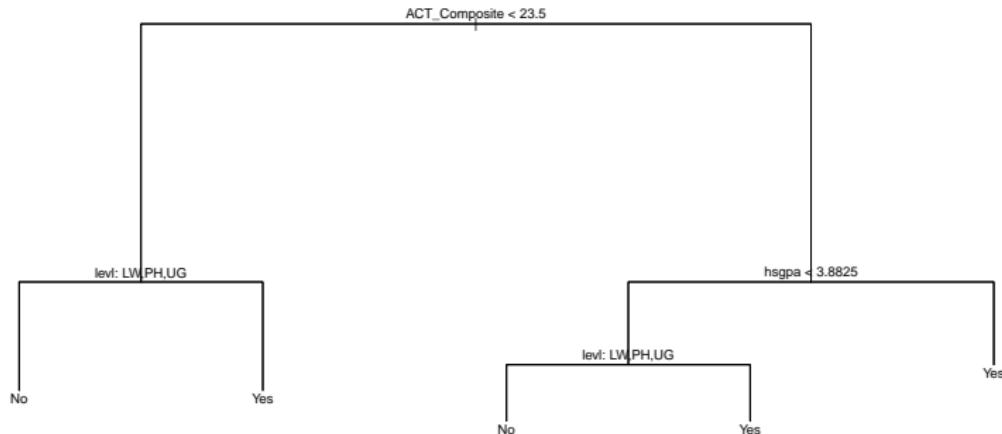
# Easiest majors? (or they just work harder) (> 3.5)



# About outliers



# A simple way to look at it



Students with a  $> 3.5$  GPA

Misclassification error rate:  $0.2656 = 3053 / 11495$

# What random forest thinks are most important

Predictors	Relative Influence
Highschool GPA	37.3484049

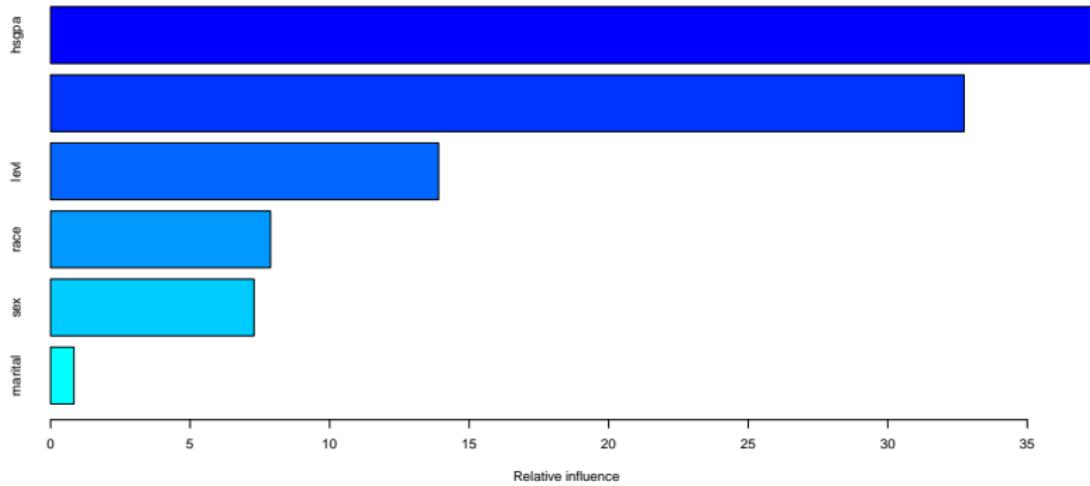
# What random forest thinks are most important

Predictors	Relative Influence
Highschool GPA	37.3484049
ACT Composite	32.7361483

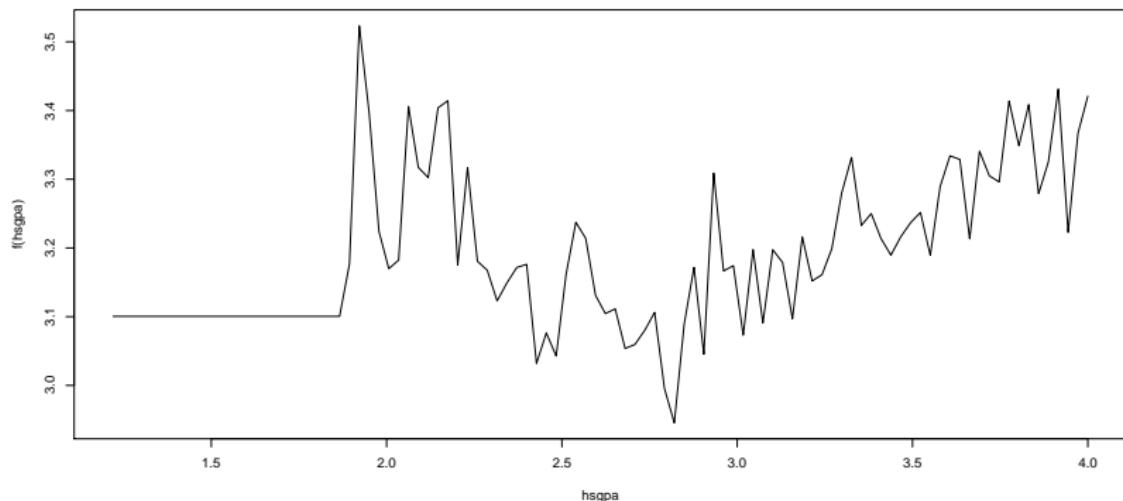
# What random forest thinks are most important

Predictors	Relative Influence
Highschool GPA	37.3484049
ACT Composite	32.7361483
Level	13.9047409
Race	7.8814707
Sex	7.2903646
Marital Status	0.8388707

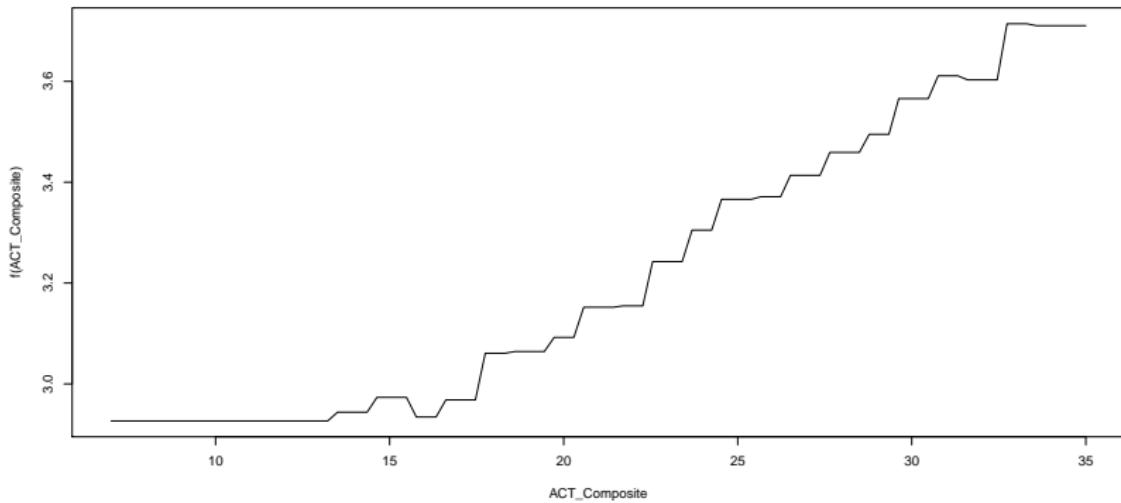
The method used here is called boosting algorithm



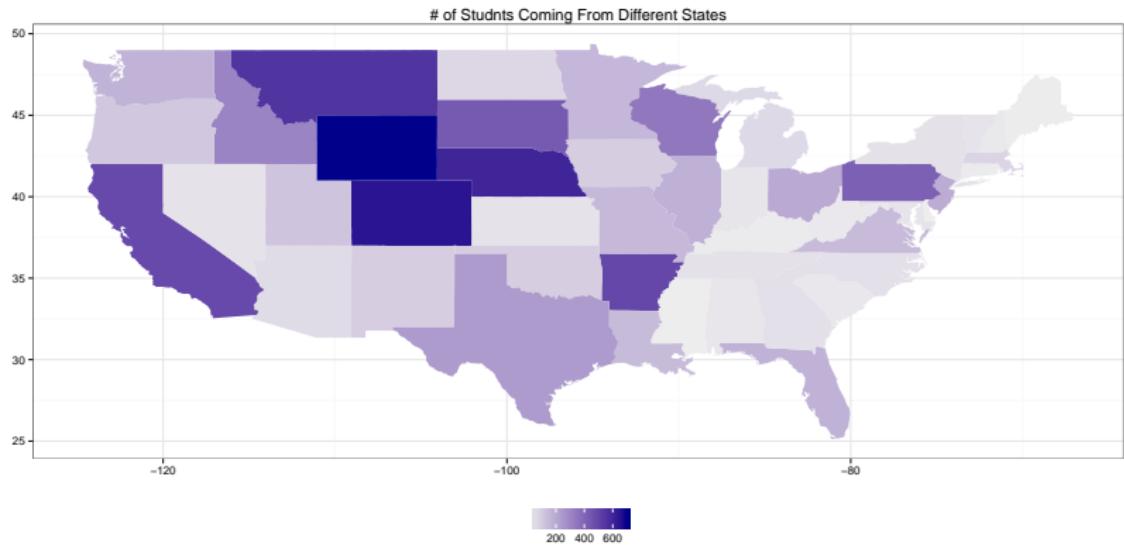
# How does highschool GPA relate to UW GPA?



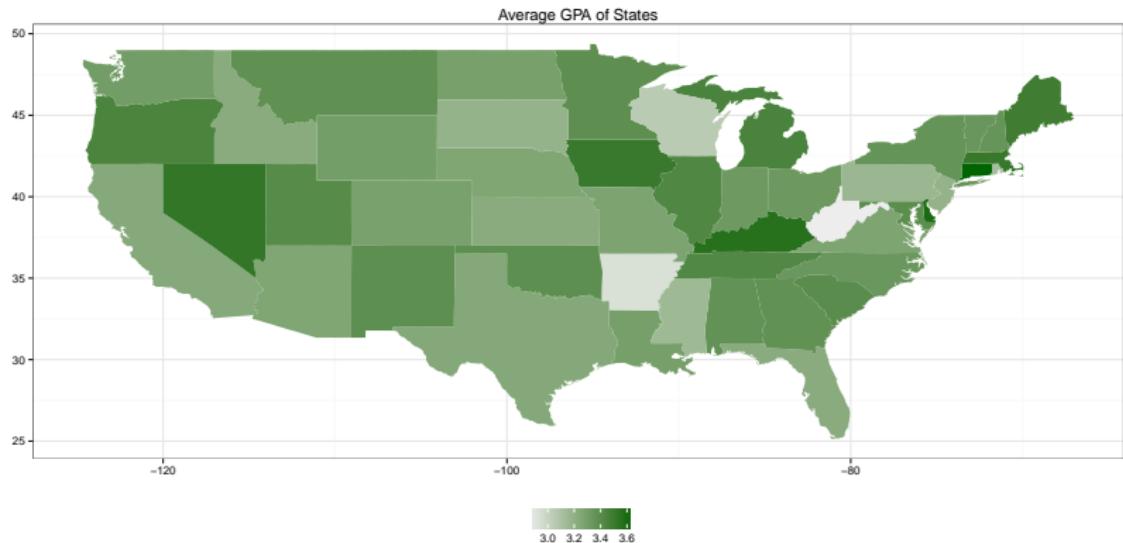
# What about ACT Composite?



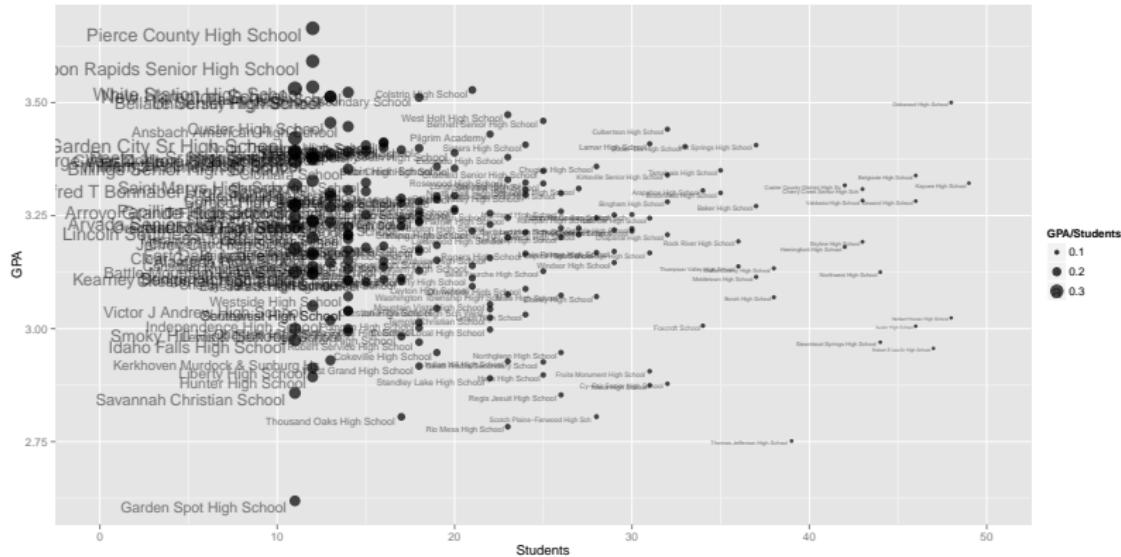
# Most of our students come from...



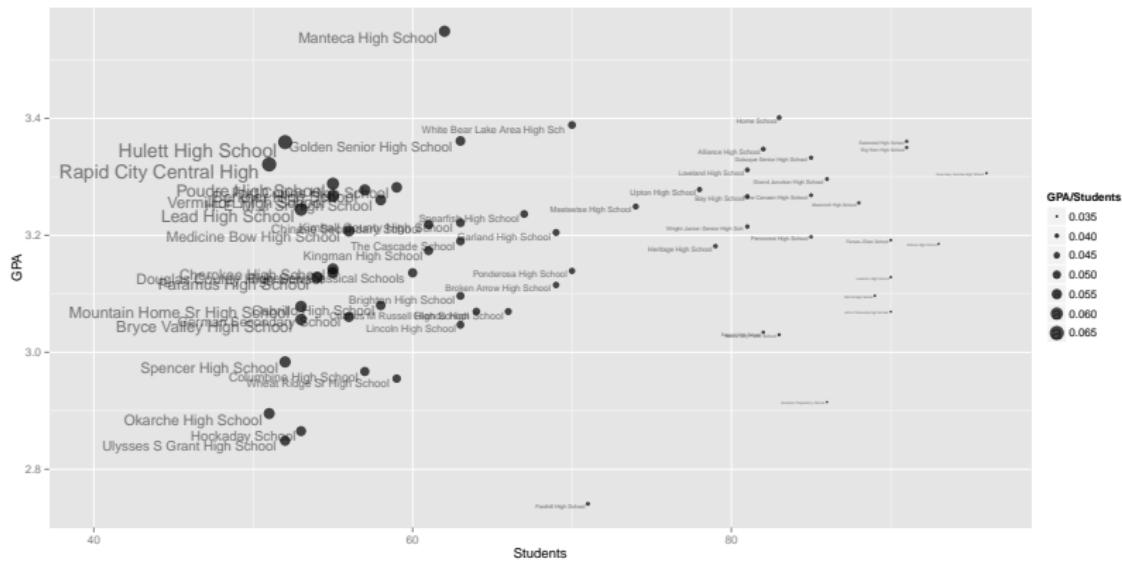
# The best students come from...



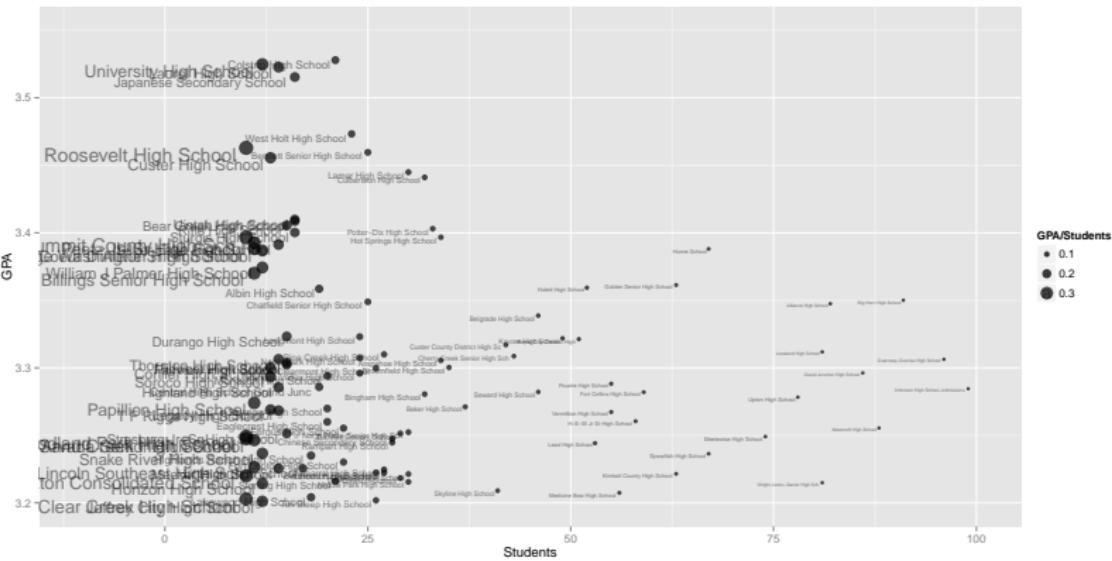
# So many high schools!



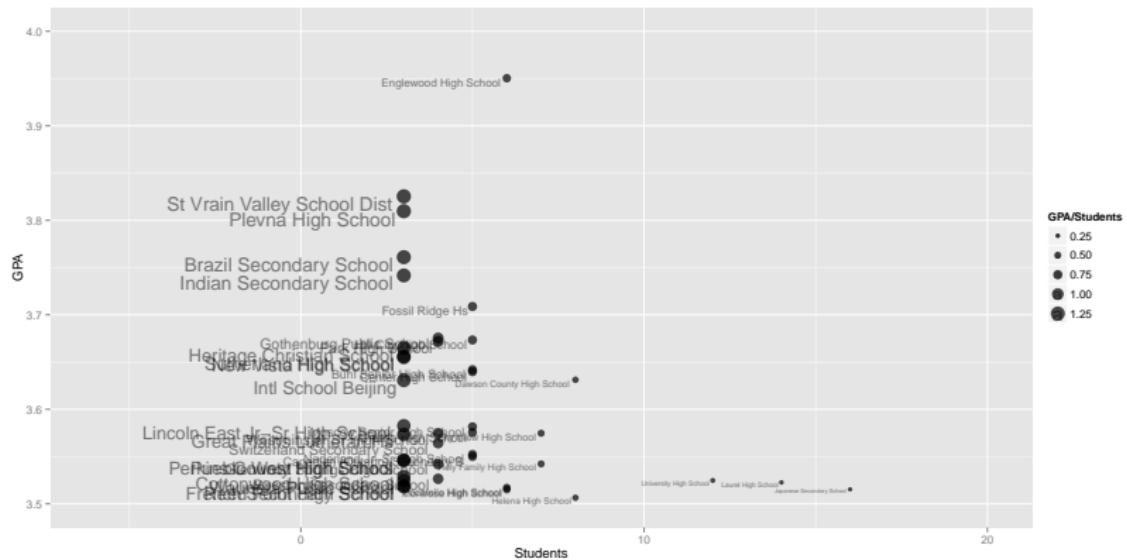
# Let's take a closer look



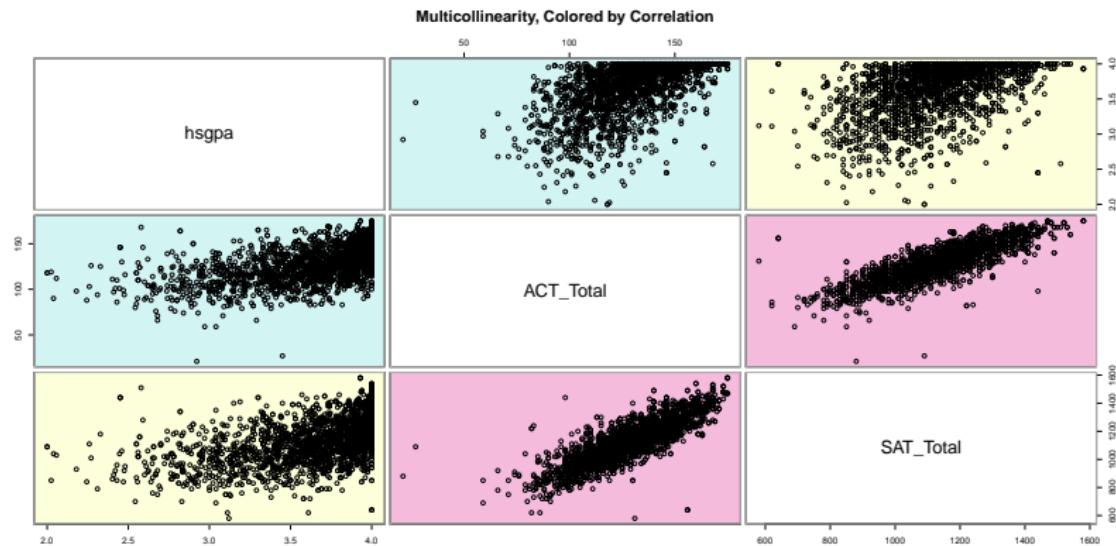
# The high schools near UW



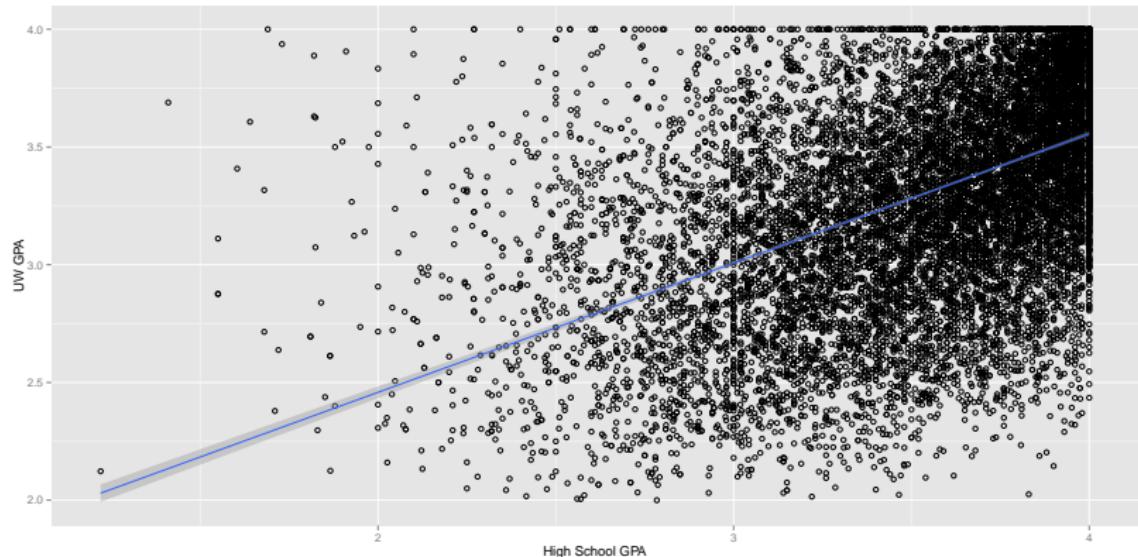
# A closer look shows us...



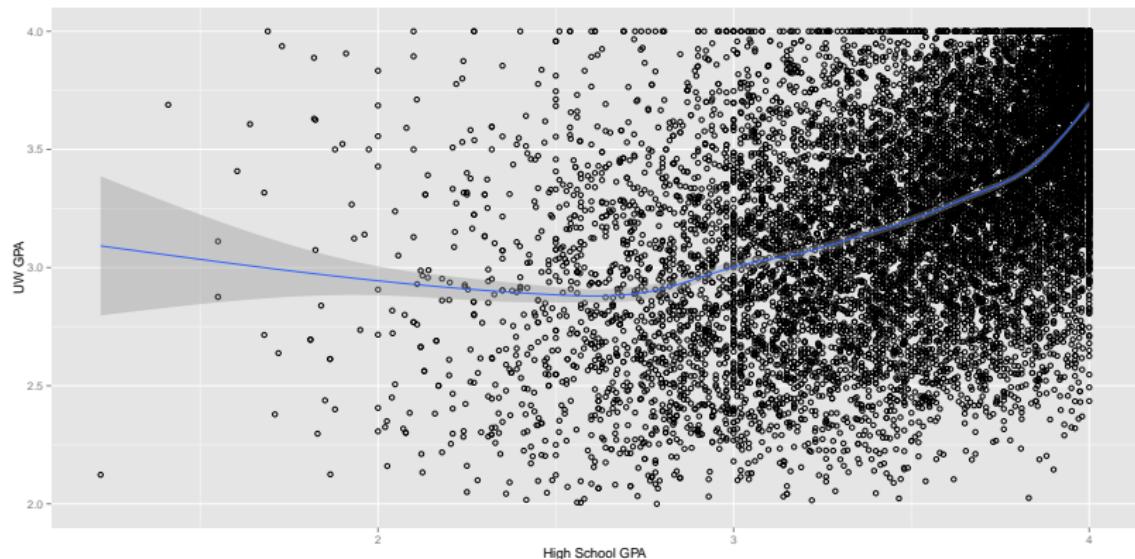
# Our continuous predictors



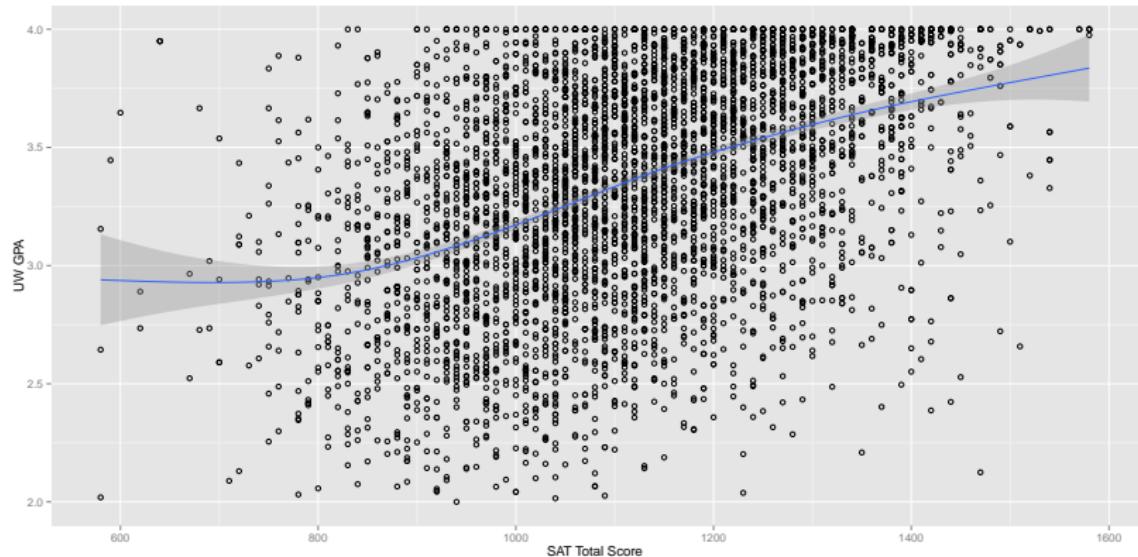
There is a positive strong correlation



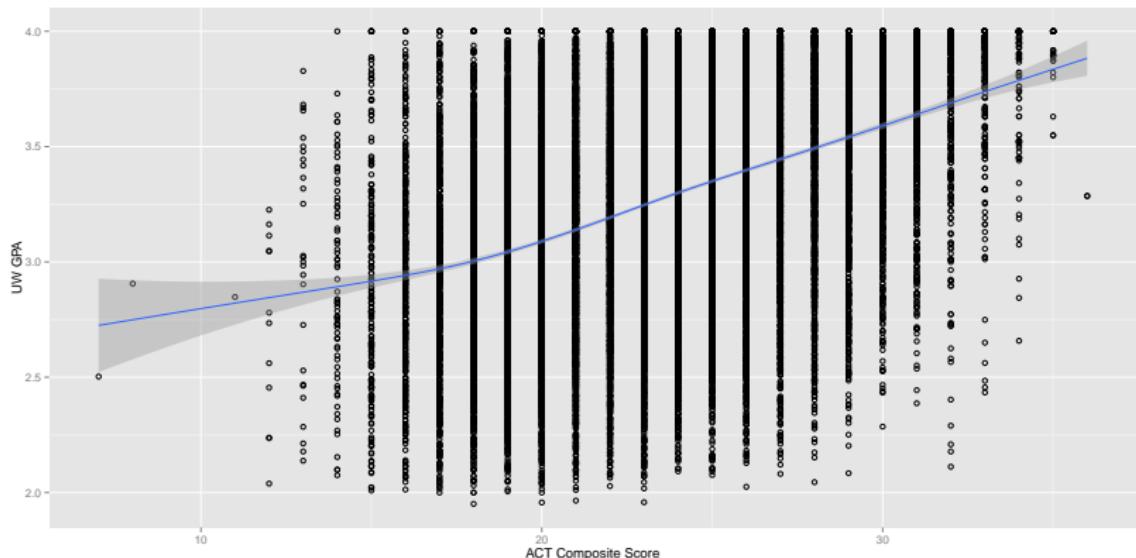
# Another way to look at it



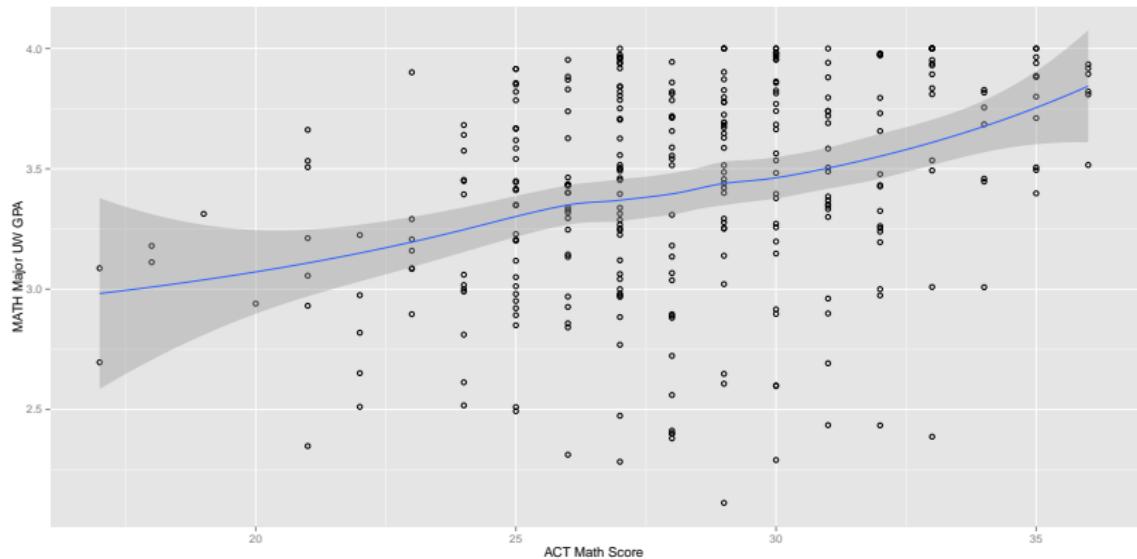
## SAT



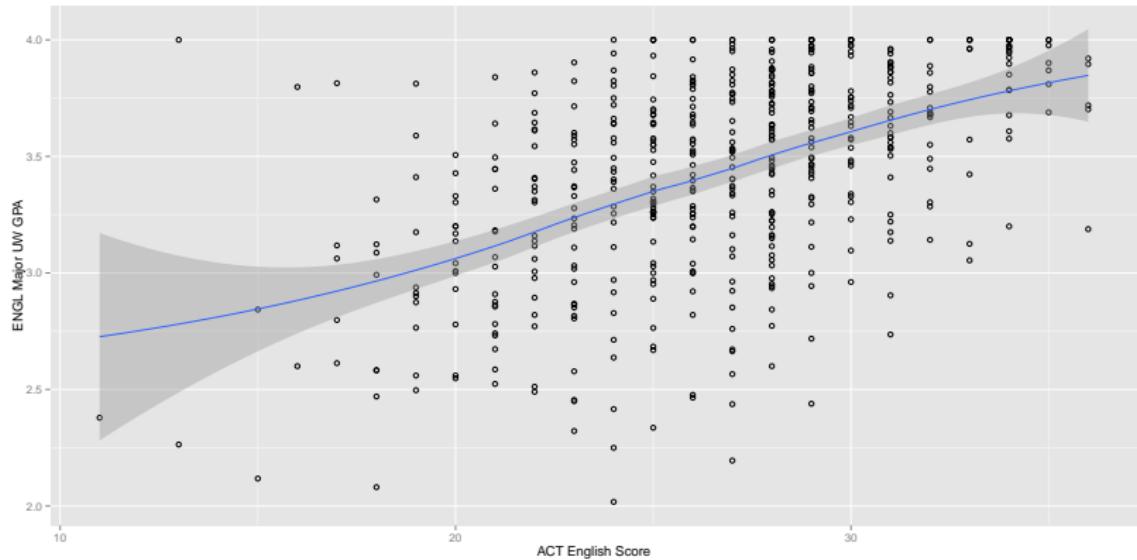
## ACT



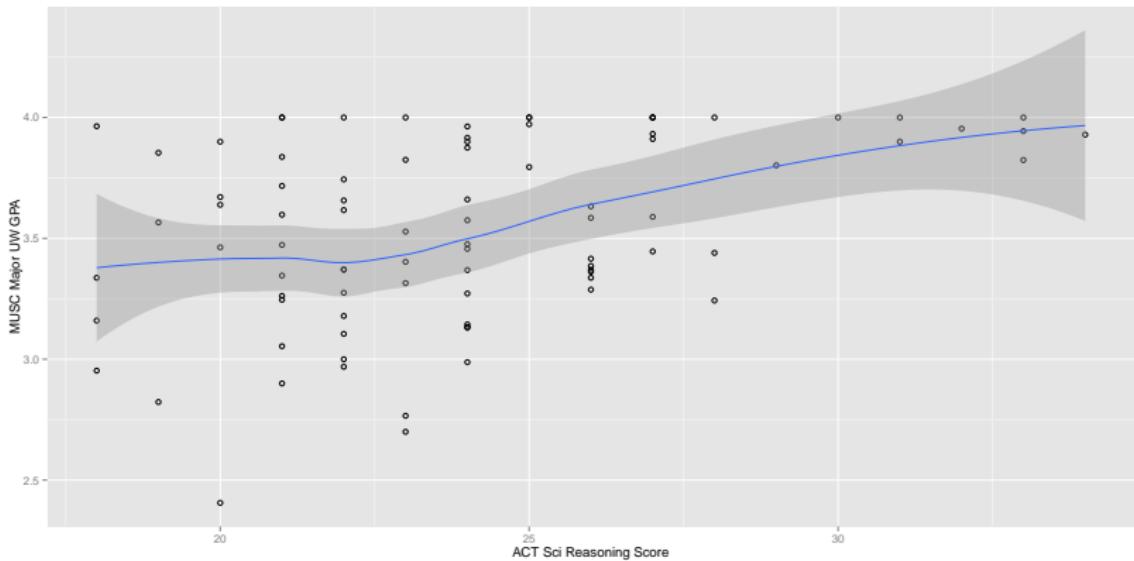
# Do Math majors really need Math?



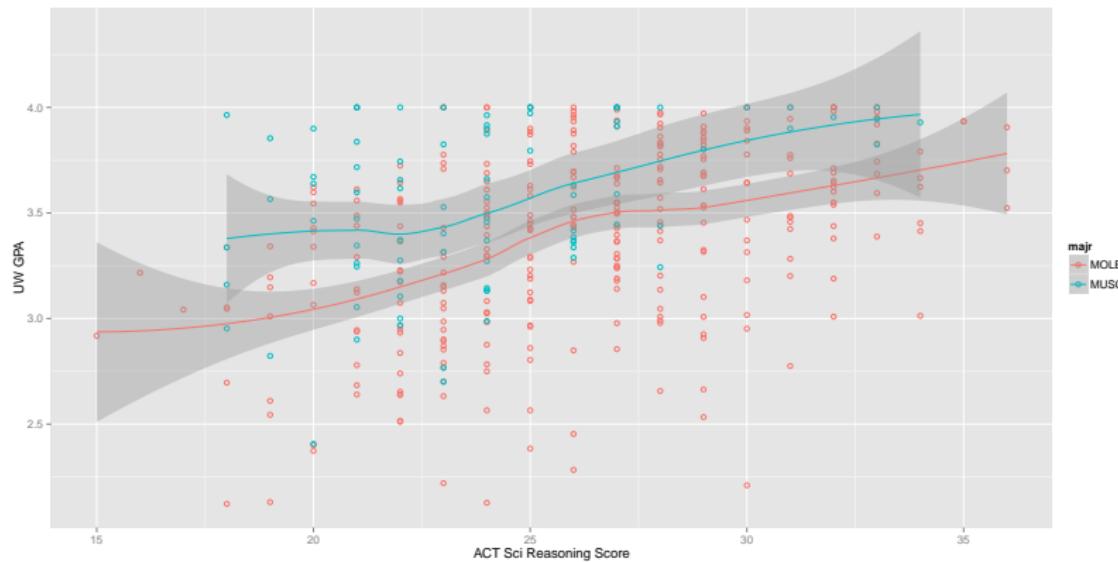
# Do English majors really need English?



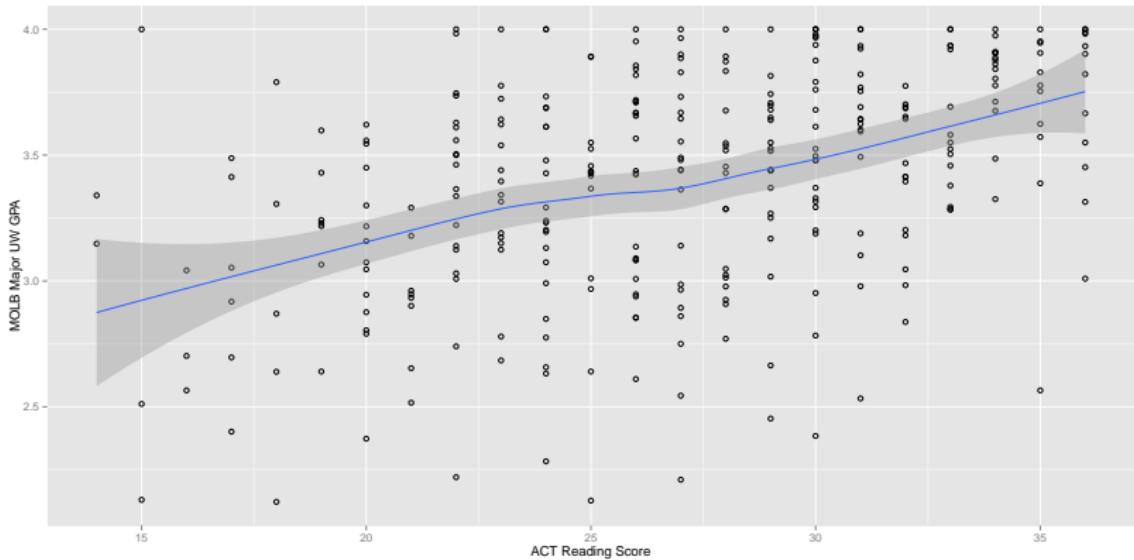
# Do Music majors need science reasoning?



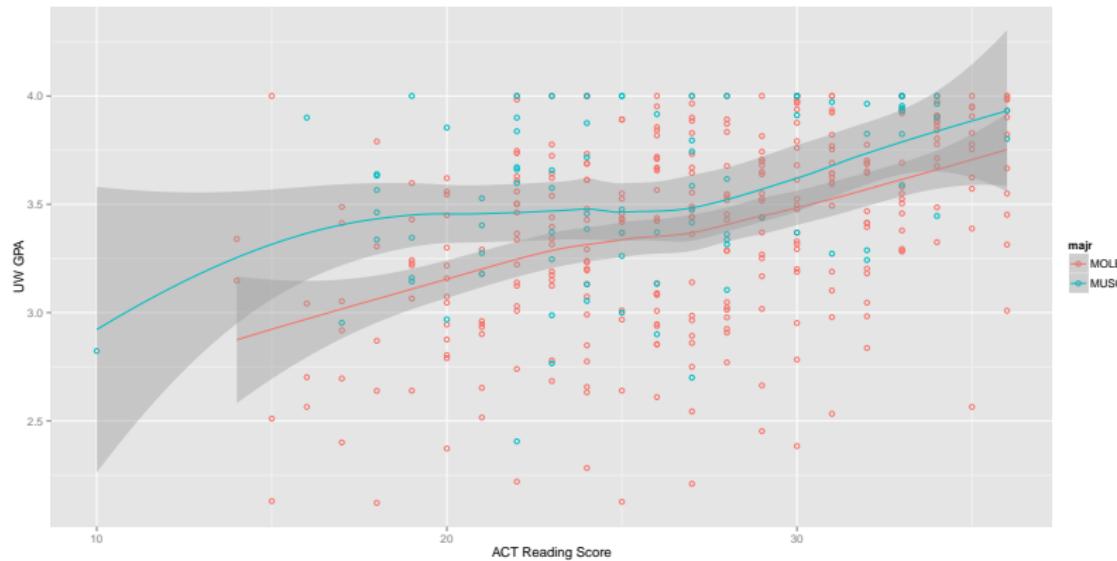
# When comparing to Molecular Biology



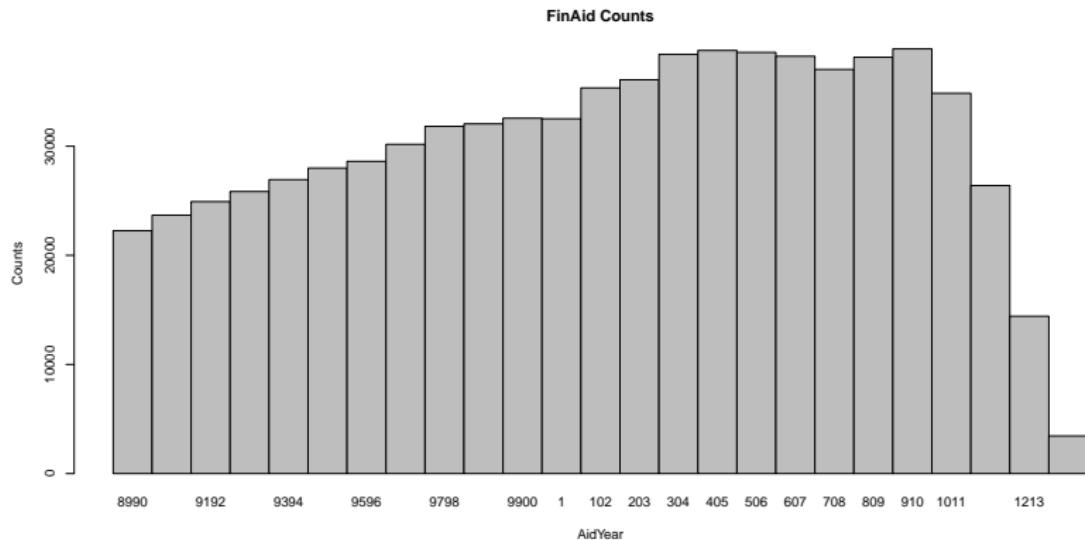
# Molecular Biology again



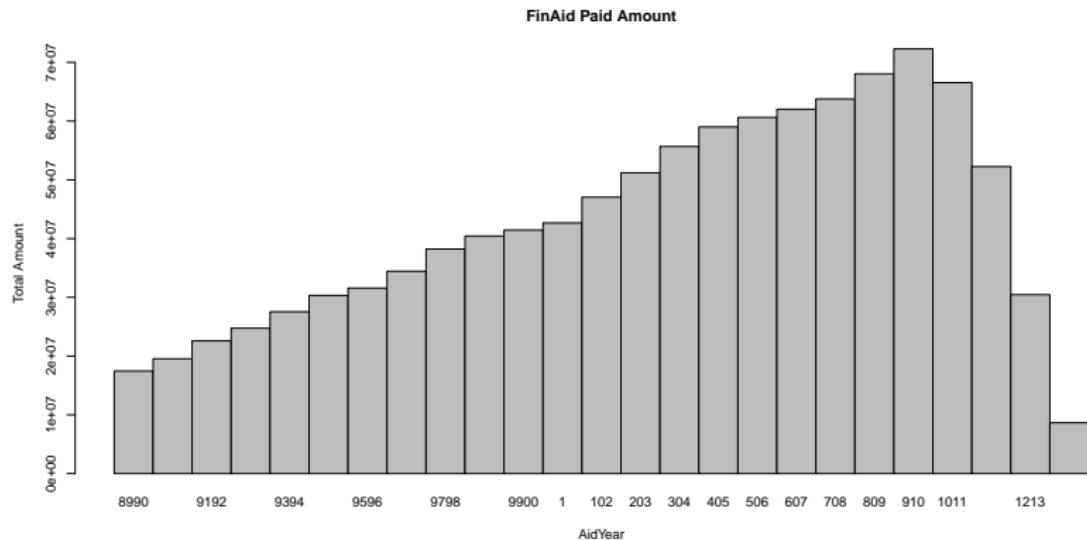
# Let's add Music again



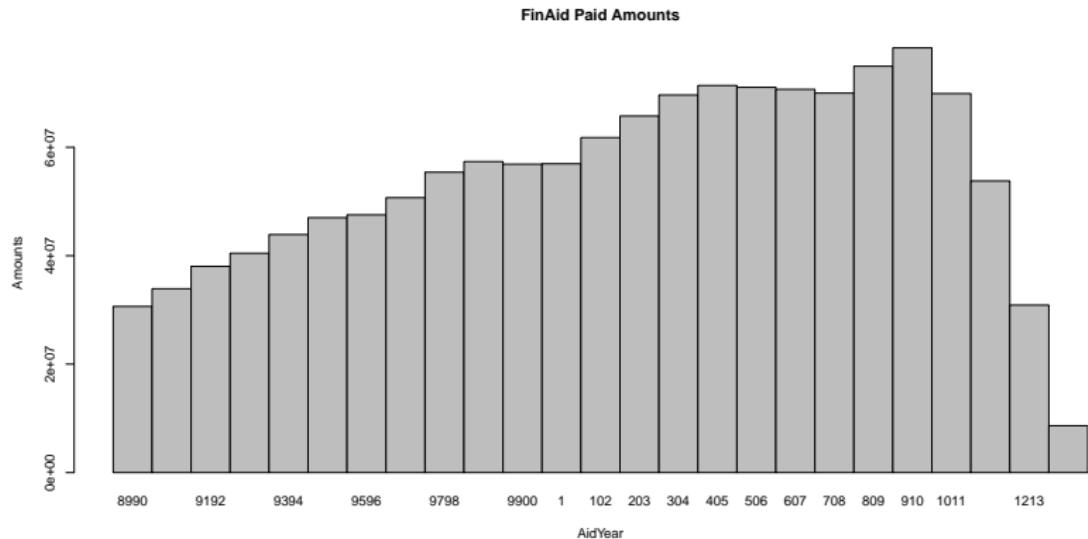
# Are UW students receiving more and more FinAid?



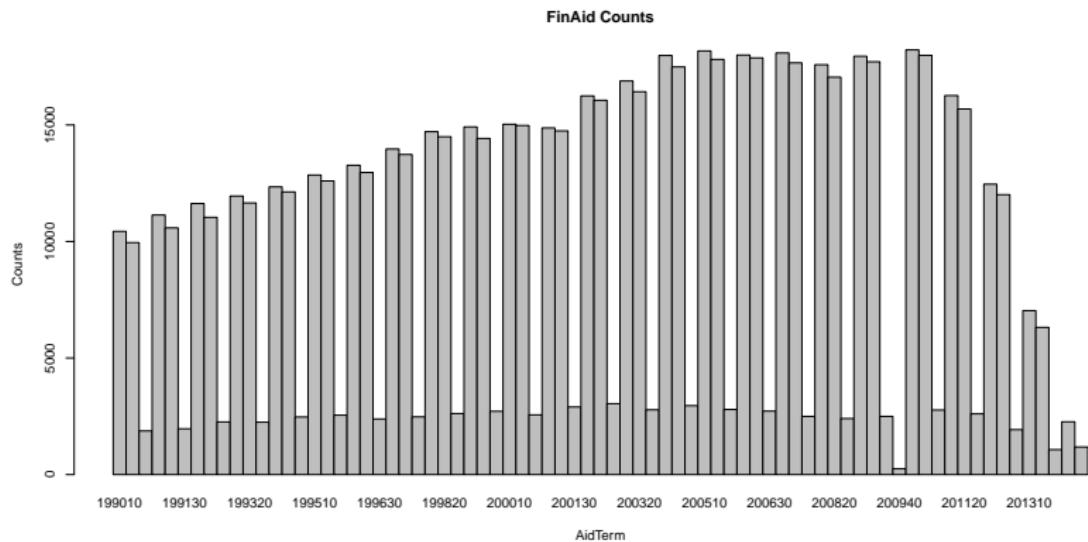
# From the amount of money perspective



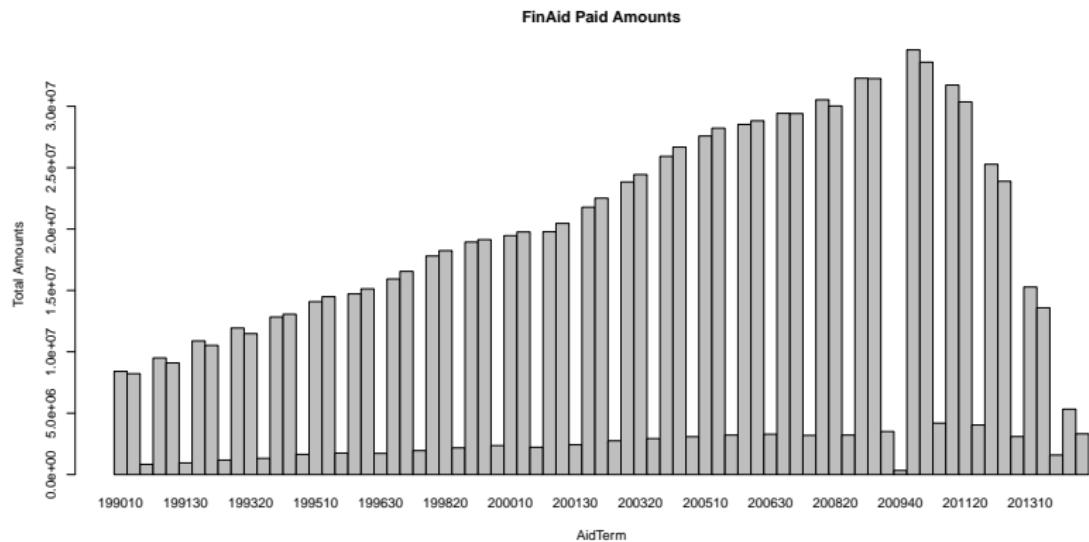
# Taking into account the inflation



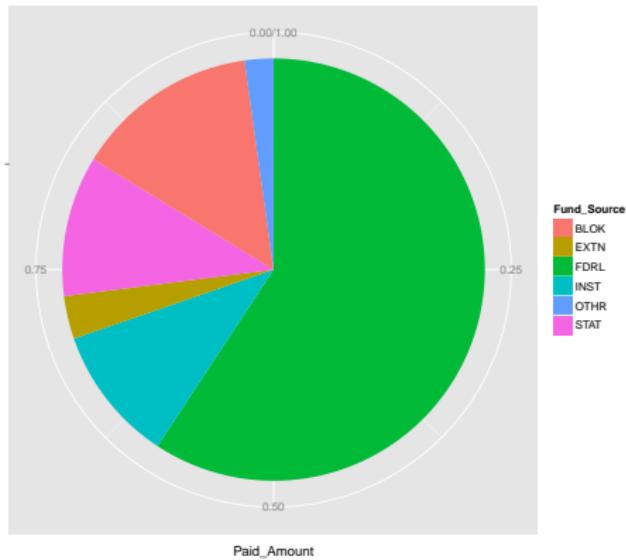
# Look at it in semester terms



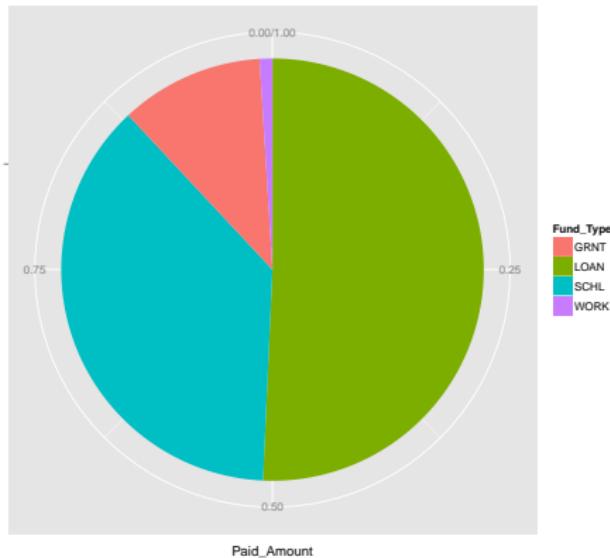
# And amounts



# Sources of funding



# Types of funding



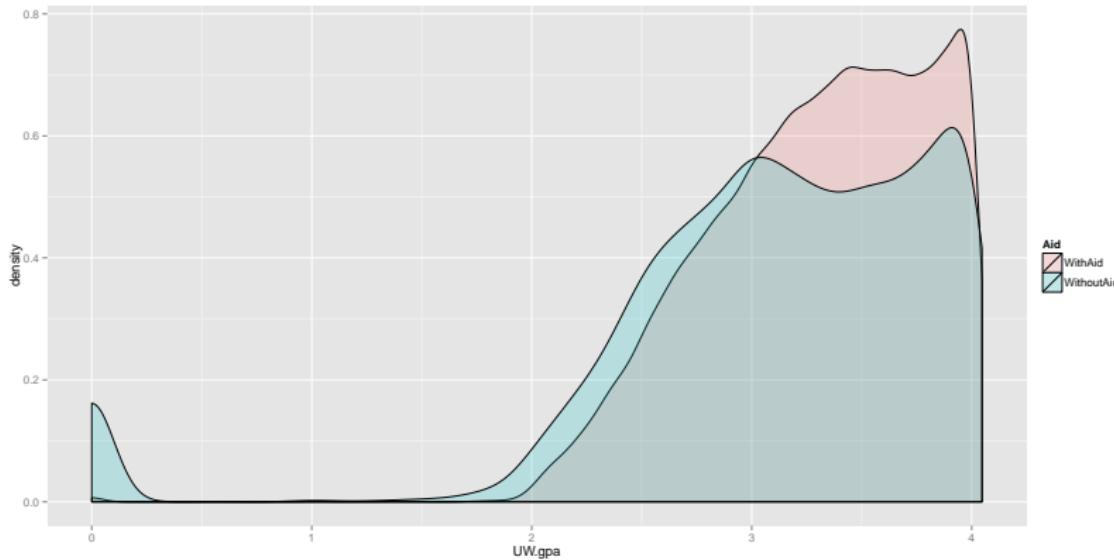
# Students that have a 0 GPA

- 455 students. 47 of them have received Financial Aid.
- Or in another way, 47 of 44439 (0.106%) FinAid students withdrew.
- Among these 47, the most received \$22252 (Scholarship and Loan), the least received \$35 (T&F Reduction).
- Totally \$155351.43.

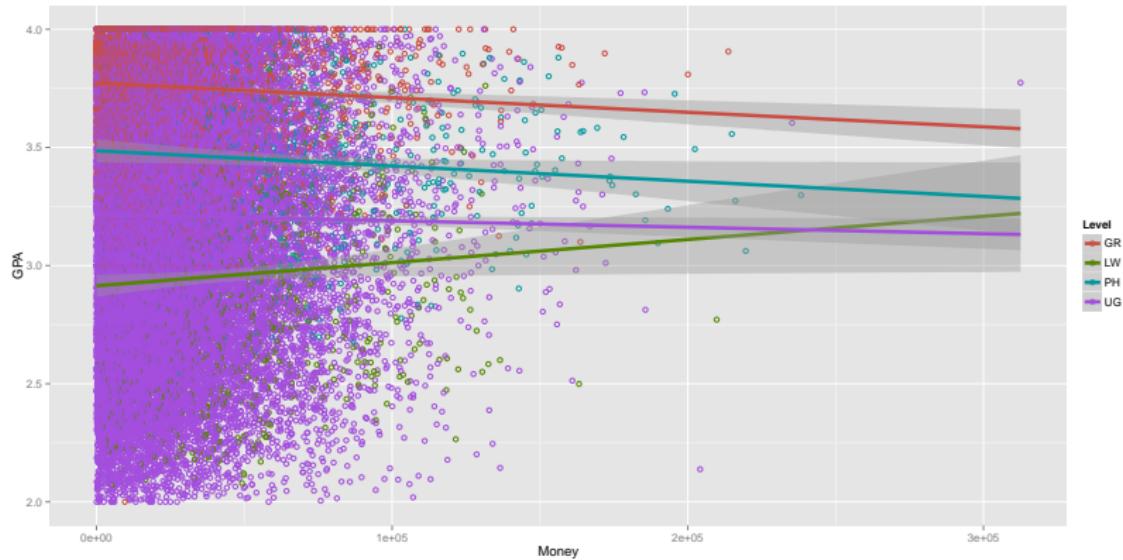
# Students with a < 2.5 (Including 0) GPA

- 4697 students. 3169 of them have received Financial Aid.
- Or in another way, 3169 of 44439 (7.13%) FinAid students did not performe well.
- Among these 3169, the most received \$213708.31 (Loan and many Scholarships). Students received great amounts often have different degrees.
- Totally \$81595467.76.

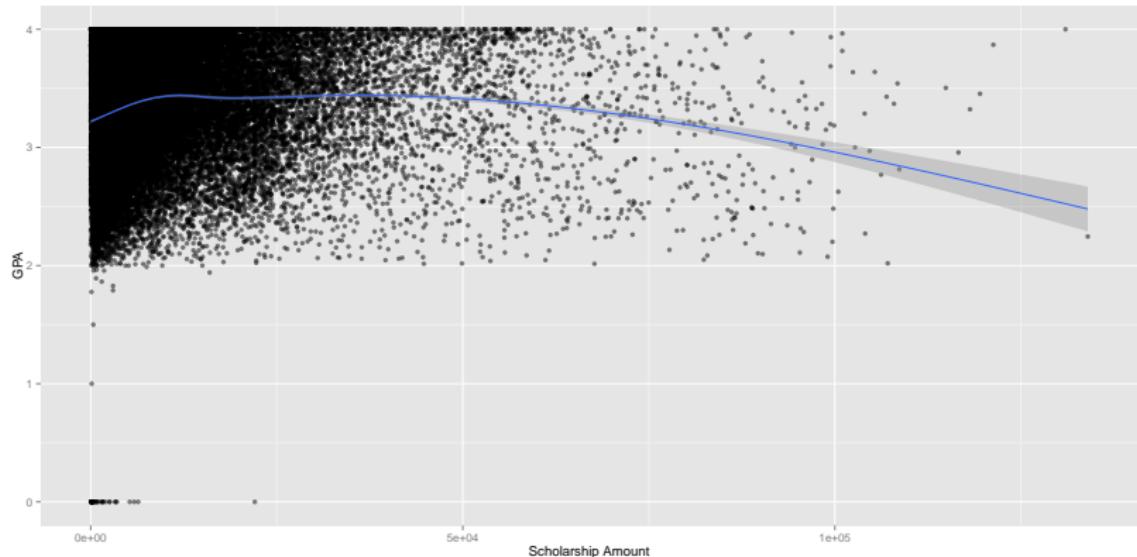
# Money helps students perform better



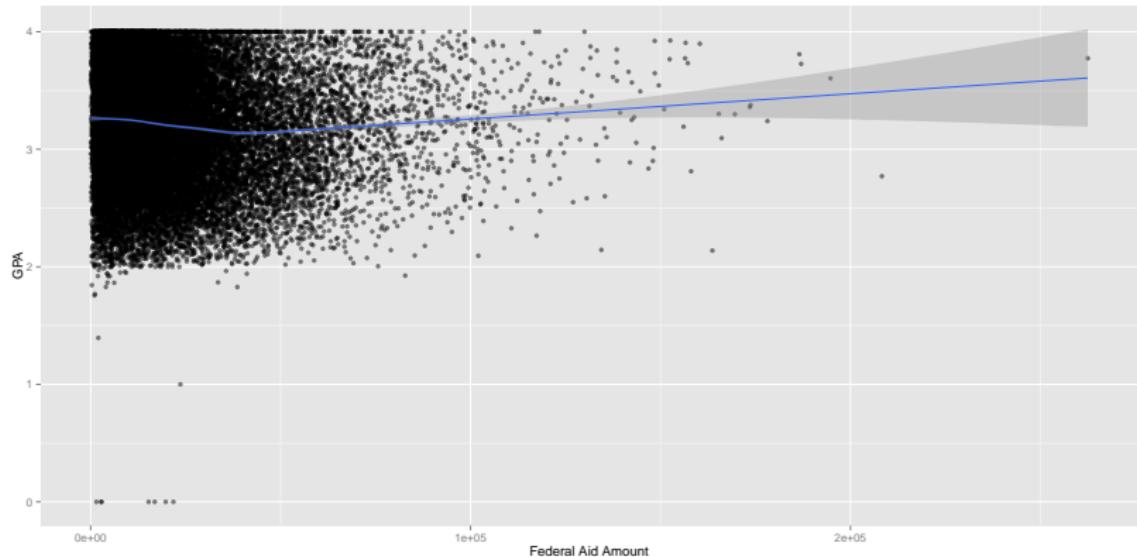
# Depending on which level you are in



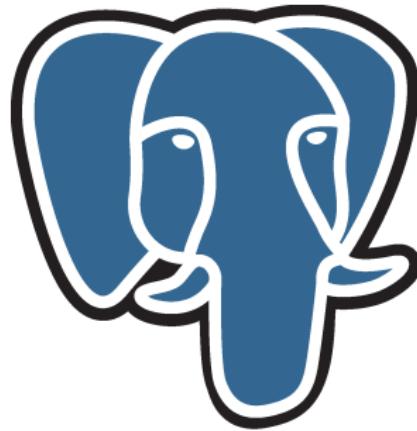
# If we only look at Scholarship



# If we only look at Federal Aid



# PostgreSQL



# PostgreSQL

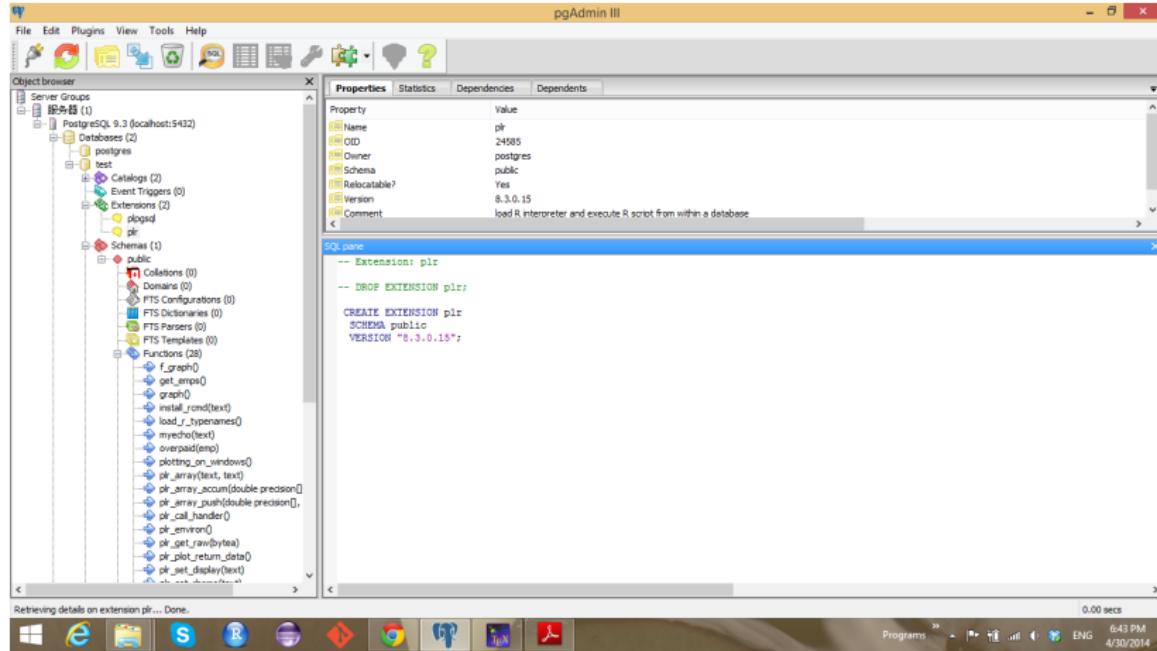
the world's most advanced open source database

# Procedural Language for PostgreSQL

- PL/Java
- PL/Perl
- PL/php
- PL/Python
- And... our PL/R!

Created, updated and supported by one man . . . our hero  
Joseph E Conway <http://www.joeconway.com/plr/>

# The graphical user interface



# Why do we do this?

## What SQL Can Do

- Count
- Sum, Average
- Max, Min
- First, Last
- ... (not much left)

## What SQL Can Not Do

- Median, Quantile
- Standard Deviation, Correlation
- Analysis of Variance
- (Generalized) Linear Regression
- All Kinds of Plotting
- Time Series Analysis
- Classification Tree, Random Forrest
- ... basically, almost everything that R can do

# Let's find the medians!

The screenshot shows the pgAdmin III interface. The top bar displays the title "Query - test1 on postgres@localhost:5432". The menu bar includes File, Edit, Query, Favourites, Macros, View, Help. Below the menu is a toolbar with various icons. The main window has tabs for "SQL Editor" and "Graphical Query Builder", with "SQL Editor" selected. A "Previous queries" dropdown is open. The SQL code area contains the following:

```
1  create or replace function r_median(_float8) returns float as '$1'
2  median(arg1);
3  ' language 'plr';
4  ;
5  CREATE AGGREGATE median ($1
6  sfunc = plr_array_accum,
7  basetype = float8,
8  stype = _float8,
9  finalfunc = r_median
10 );
11 ;
12 create table foo(f0 int, f1 text, f2 float8);
13 insert into foo values(1,'cat1',1.21);
14 insert into foo values(2,'cat1',1.24);
15 insert into foo values(3,'cat1',1.18);
16 insert into foo values(4,'cat1',1.26);
17 insert into foo values(5,'cat1',1.15);
18 insert into foo values(6,'cat1',1.15);
19 insert into foo values(7,'cat2',1.26);
20 insert into foo values(8,'cat2',1.32);
21 insert into foo values(9,'cat2',1.30);
22 ;
23 select f1, median(f2) from foo group by f1 order by f1;
```

The "Output pane" at the bottom shows the results of the query:

f1	median	
text	double precision	
1	cat1	1.195
2	cat2	1.3

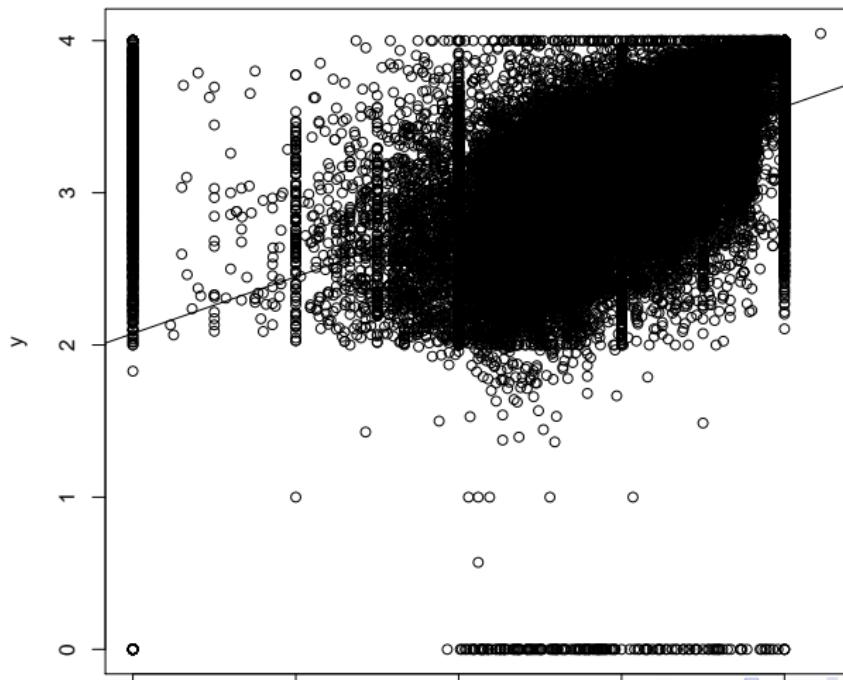


# Coding is fun and easy

```
create or replace function boxplot(data float[])
returns text as
,
pdf("D:/ boxplot.pdf");
boxplot(data);
dev.off();
print("done");
,
language 'plr' strict;

select boxplot(array_agg(uw_gpa)) from all_grads;
```

# SQL can plot linear regression line now



# Even generalized linear regression is possible

The screenshot shows the pgAdmin III interface. The title bar says "Query - test1 on postgres@localhost:5432 \*". The menu bar includes File, Edit, Query, Favourites, Macros, View, Help. The toolbar has various icons for file operations. A connection status bar at the top right says "test1 on postgres@localhost:5432". The main window has tabs for "SQL Editor" (which is selected) and "Graphical Query Builder". In the SQL Editor, there is a "Previous queries" section and a code editor containing the following SQL:

```
1 select glm_plr(array_agg(uw_gpa),array_agg(first_term_gpa)) from all_grads;
```

Below the SQL editor is an "Output pane" tab. Under "Data Output", the results of the query are displayed in a table:

	glm_plr	lm_type
1	("(Intercept)", 2.32304221041427, 0.0079082184744897, 293.750383592452, 0)	
2	(x, 0.31375901628002, 0.00247234384490469, 126.907516091119, 0)	

At the bottom of the output pane, it says "OK." and there is a set of navigation icons.

# The truncated regression you saw before

The screenshot shows the pgAdmin III interface with a query window titled "Query - test1 on postgres@localhost:5432 \*". The query editor contains the following SQL code:

```
1 select truncreg_plr(array_agg(uw_gpa),array_agg(first_term_gpa)) from all_grads;
```

The output pane displays the results of the query, which are the parameters for a truncated regression model:

	truncreg_plr
	lm_type
1	("(Intercept)", 2.32304211210073, 0.00790815323299613, 293.752794572572, 0)
2	(x, 0.313759013081575, 0.00247232175435176, 126.908648734454, 0)
3	(sigma, 0.509241840392061, 0.00141317848275043, 360.352104569933, 0)

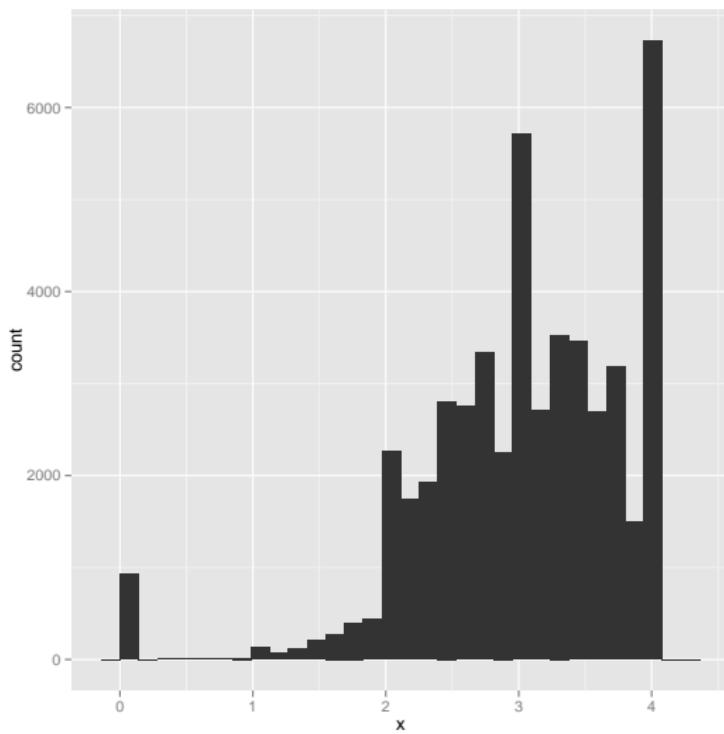
At the bottom of the pgAdmin window, there are several navigation icons.

# Installing R packages

```
CREATE TABLE plr_modules (
  modseq int4 ,
  modsrd text
);
```

```
INSERT INTO plr_modules
VALUES (0, 'library(Kendall)');
```

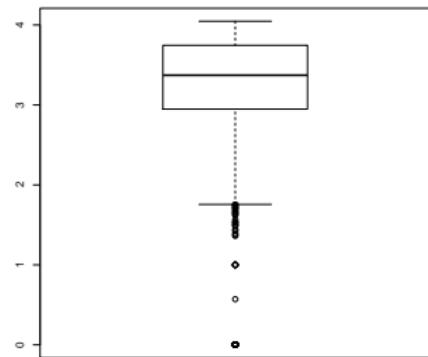
# ggplot2 is working now!



# Just do it!

## Let's Run Some Examples!

- Correlation
- Boxplot
- Histogram by ggplot2
- Install a Package



# Back to the future

## Possible Future Works

- Dig deeper into the data ...
- Complete constructing and refining our R/SQL language.

# Finally...

- Thank You All For Listening To Us!
- Thank You, Professor Douglas!

