

Predicting Economic Indicators

Grace Gee, Eugene Wang

Our previous notebook is attached below.

Background and Motivation.







Initial Plan

Initially, we planned to explore prediction of sovereign bond ratings, using various quantitative measures of political situations, economic status, and historical resilience amongst many other factors.

We first envisioned that our project would consist of 2 components:

1. Analyses - We will explore different classifiers, such as SVMs and neural networks that are well suited for predicting bond ratings.
2. Visual Exploration – Enable the user to input various prediction parameters and explore the accuracy of their prediction model.

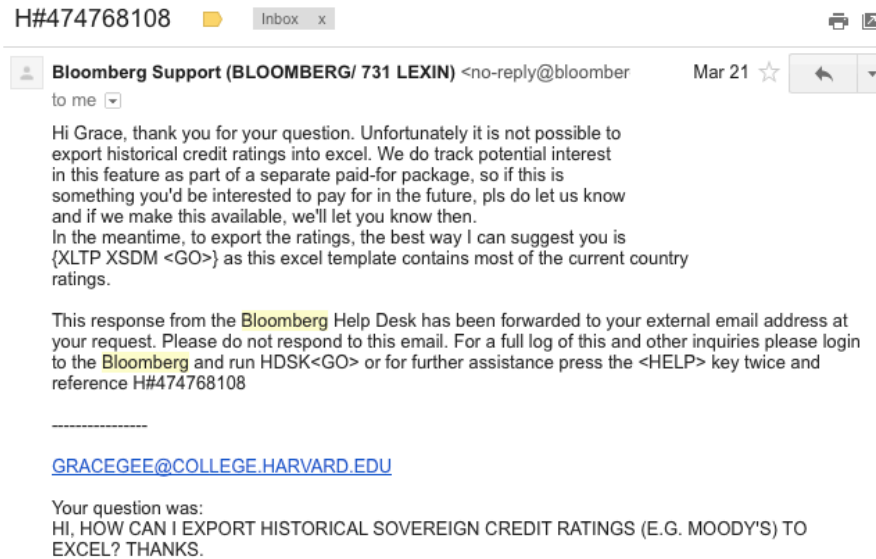
However, we pivoted after not being able to find a complete historical data set on sovereign bond indicators. The only sovereign bond indicators public dataset was from Moody's, which has sporadic data for different countries and does not include the most recent values. We thought of innovative ways to extract bond ratings data such as scrapping historical archive of wikipedia (http://en.wikipedia.org/wiki/List_of_countries_by_credit_rating).

Country	Rating	Outlook	Date	Ref.
 Abu Dhabi, UAE	AA	Stable	2012-02-20	[4][5]
 Albania	B+	Stable	2012-02-20	[4][5]
 Andorra	A-	Negative	2012-02-20	[4][5]
 Angola	BB-	Stable	2012-02-20	[4][5]
 Argentina	CCC+	Negative	2013-09-11	[6]
 Aruba	A-	Stable	2012-02-20	[4][5]

Example of bond ratings table that can be scrapped for historical data

However the free internet archive only included data as far back as 2010. This doesn't really work since most bond ratings are only updated semi-annually, which gives us a dearth of data to visualize.

We even made two trips down to Baker library at Harvard Business School to access the Bloomberg terminal (a data terminal for financial professionals) for sovereign bond ratings data. After a couple of hours of fiddling around, we decided to contact the Bloomberg help desk. Here's a correspondence with them and basically, they told us that there is no historical data for sovereign bond ratings.



After hitting the wall, we decided to switch gears.

Current Plan

In light of this, we switched to related economic time series, specifically 6 indicators that are related to the sovereign bonds ratings:

- Budget balance as a % of GDP
Difference between government spending and revenue
- Foreign debt as a % of GDP
Money owed by government to other nations
- Real GDP Growth
Growth of the country's economy net of inflation
- Inflation
Rate of price increases
- Current account as a % of GDP
Exports net of imports
- Unemployment rate
Proportion of workforce who are not employed

These indicators are primarily used to show the fiscal and monetary health of a country, which are close to our original intent for the project on sovereign bond ratings. Specifically, they tend to be very inter-related. For example, the unemployment rate of a country is famously modeled by the Phillip's curve (inverse relationship between the two).

The data is also readily available (although quite a bit of preprocessing was needed). More on the data collection part can be found in later sections.

Related Work

We both have extensive experience in machine learning and data science. Previously we built a model for predicting Supreme Court outcomes based on attorney arguments (datahacked.com) and we really want to bring the joy of prediction and modeling to the masses. This project will be able getting out users' hands dirty with fitting coefficients for a multi-linear regression model.

Project Objectives/ Questions. Provide the primary questions you are trying to answer with your visualization. What would you like to learn and accomplish? List the benefits.

The purpose of the project is to allow our users to manually construct their own multi-linear model to predict their chosen time series data (out of the 6) using the other 5 unselected time series. Oftentimes in econometrics, we take advanced tools like R for granted in automatically fitting regression equations for us. The following is an example of the R code that can perform multi-linear regression:

```
Call:
lm(formula = ROLL ~ UNEM + HGRAD + INC, data = datavar)

Residuals:
    Min       1Q   Median       3Q      Max
-1148.840  -489.712   -1.876   387.400  1425.753

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.153e+03  1.053e+03  -8.691 5.02e-09 ***
UNEM         4.501e+02  1.182e+02   3.809 0.000807 ***
HGRAD        4.065e-01  7.602e-02   5.347 1.52e-05 ***
INC          4.275e+00  4.947e-01   8.642 5.59e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 670.4 on 25 degrees of freedom
Multiple R-squared:  0.9621, Adjusted R-squared:  0.9576
F-statistic: 211.5 on 3 and 25 DF, p-value: < 2.2e-16
```

But manually fitting a regression model can give us some intuition about the relationships between the variables. For example, if I increase the coefficient for one of the independent variable and obtain a better fit, I can be more confident that there is a strong positive relationship between the dependent and independent variable. As such, we are determined to produce a tool that allows users to carry out such exploratory analysis – all with elegant visualizations.

On a deeper level, we also want to show our users that it is not easy getting a fit for a multi-linear regression model manually. Intuitively, we want to show that the coefficients for a linear regression model with more than 1 independent variable are hard to obtain by inspection.

Understandably, our other objective is to allow the user to carry out all these in a fun setting. After all, fitting linear regressions is an activity that only few masochistic Econ PHD students in the basement of Littauer will enjoy. In our design, we prioritize the fun element in the user interactions. We specifically shape the webpage up like a game, where users are challenged to minimize the error of their linear regression model.

Data. From where and how are you collecting your data? If appropriate, provide a link to your data sources.

We will collect time series data on

- Budget balance as a % of GDP
- Foreign debt as a % of GDP
- Real GDP Growth
- Inflation
- Current account as a % of GDP
- Unemployment rate

The data spans 1990 - 2013 and will be collected from the PRS (Political Risk Services) group. The PRS group is founded in 1979 and is one of the most reliable data sources for tracking countries' financial health. Their database is available to Harvard students via the HBS and can be obtained from the link below:

<https://www-countrydata-com.ezp-prod1.hul.harvard.edu/index.php/customer/countrydata/>

Note that we only pick the 6 most commonly used economic time series to ensure that most of our audience can understand the purpose of the project. The data contains all the 6 time series for all the major countries in the world. By ensuring that we obtain data on different countries from the same source, we can minimize discrepancies between numbers when we are comparing between countries, since they use the same methodology.

Exploratory Data Analysis

We initially used Excel to visualize some trends and do basic data cleaning (e.g. replace missing values). We also realize that data such as GDP has low variability even across countries while indicators such as inflation are sensitive to outliers (Zimbabwe's inflation rate went through the roof).

Data Processing. Do you expect to do substantial data cleanup? What quantities do you plan to derive from your data? How will data processing be implemented?

When we first obtain the data, it is in csv format with the first column as country names and second column as the indicator time series. Though the data is already in a pretty format, we still did the following wrangling in javascript (check data_wrangling.html)

1. Convert country names to alpha3 code. This part of the wrangling was carried out manually. We obtain a list of country names and alpha3 code from (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-3) and convert the names accordingly.
2. We used javascript to convert the flat csv data into json. The top node of the json are the countries' alpha3 code followed by the indicators they have. Lastly, each date is a key attached to a specific value for the indicator. Here's a screenshot of the data we obtained after we switch out the alpha3 code.

	A	B	C	D	E	F	G	H	I	J	K
1	Country	Variable	1990	1991	1992	1993	1994	1995	1996	1997	1998
2	DZA	Budget Balance as % of GDP	-4.4	-1.4	-1.6	-8.8	-4.1	-1.4	2.9	2.5	0
3	DZA	Current Account as % of GDP	2.3	5.2	3.3	1.6	-4.3	-5.3	2.7	2.5	4
4	DZA	Foreign Debt as % GDP	42.3	54.3	48.9	46.4	69.5	75	71.7	87	65
5	DZA	Inflation (%)	16.7	25.9	31.7	20.6	29.2	32.1	21.6	7	8
6	DZA	Real GDP Growth (%)	2.6	2.1	2.2	-2.9	-1.8	3.9	4.3	4	-1
7	DZA	Unemployment Rate (%)	19.7	21.1	23.8	38.7	38.6	38.8	38.9	38.8	38.9
8	AGO	Budget Balance as % of GDP	-20	-20.2	-12.5	-9.5	-12.9	-27.4	-10.4	-31	-8.5
9	AGO	Current Account as % of GDP	-3.2	-7.7	-10.6	-12.2	-5.7	-6.6	50.1	-7.5	-8.5
10	AGO	Foreign Debt as % GDP	106.7	112.1	138.7	177	188.2	149.5	147	188	134.5
11	AGO	Inflation (%)	11	176	496	1838	800	11011.1	905.3	200	80
12	AGO	Real GDP Growth (%)	-5.3	-1.6	3.3	-25	8.6	2.5	12.8	3	5
13	AGO	Unemployment Rate (%)									
14	ARG	Budget Balance as % of GDP	-0.3	-0.5	0	-0.7	-0.7	-0.6	-1.9	-2.5	-1
15	ARG	Current Account as % of GDP	3.2	-0.3	-2.4	-3.4	-4.3	-2	-2.5	-2	-4.3
16	ARG	Foreign Debt as % GDP	47	34.5	31.4	25.5	33.4	38.4	40.6	34.7	25.4
17	ARG	Inflation (%)	2313.7	172	24.6	10.6	4.3	3.3	0.2	0.6	1.1
18	ARG	Real GDP Growth (%)	0.1	10.5	9.6	5.8	5.8	-2.8	5.5	8.6	4.5
19	ARG	Unemployment Rate (%)	9.2	6.3	7.2	9.1	11.7	15.9	16.6	13.4	12.1
20	AUS	Budget Balance as % of GDP	2.1	0.5	-2.3	-3.3	-2.9	-2.4	-0.9	-2	-0.1
21	AUS	Current Account as % of GDP	-5.4	-3.8	-3.7	-3.3	-5.1	-5.4	-3.9	-4.5	-4.2
22	AUS	Foreign Debt as % GDP	34.1	35	37.1	47.7	52.4	56.3	56.6	40	55.4
23	AUS	Inflation (%)	7.3	3.2	1	1.8	1.9	4.6	2.6	0.3	0.7
24	AUS	Real GDP Growth (%)	1.4	-1.3	2.7	3.8	5.2	3.8	3.9	3	2.3
25	AUS	Unemployment Rate (%)	6.9	9.6	10.8	10.9	9.7	8.2	8.2	8.3	7.8

The following is a screenshot of a portion of the json output. Note that the first level node is the alpha 3 code for the country followed by indicator name then the date value pairs.

```
{
  "DZA": {
    "Budget Balance as % of GDP": {
      "1990": "-4.4",
      "1991": "-1.4",
      "1992": "-1.6",
      "1993": "-8.8",
    },
    "Current Account as % of GDP": {
      "1990": "2.3",
      "1991": "5.2",
      "1992": "3.3",
      "1993": "1.6",
    },
    "Foreign Debt as % GDP": {
      "1990": "42.3",
      "1991": "54.3",
      "1992": "48.9",
      "1993": "46.4",
    },
  },
}
```

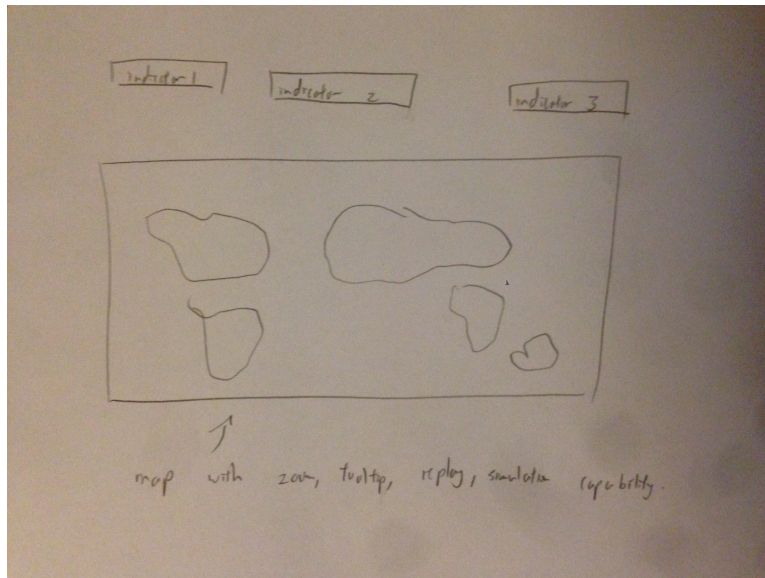
But the data wrangling process are not that smooth. We realize that some of the countries have a lot of missing values. Imputing missing values is a big problem in data science and we use the default way of address it: replace the missing values with the average. Admittedly this is less than ideal but this method will allow the user to build a prediction model without any exceptions thrown. Most of the replacement of missing values is done with simple programming in Excel.

Design Evolution and Implementation.

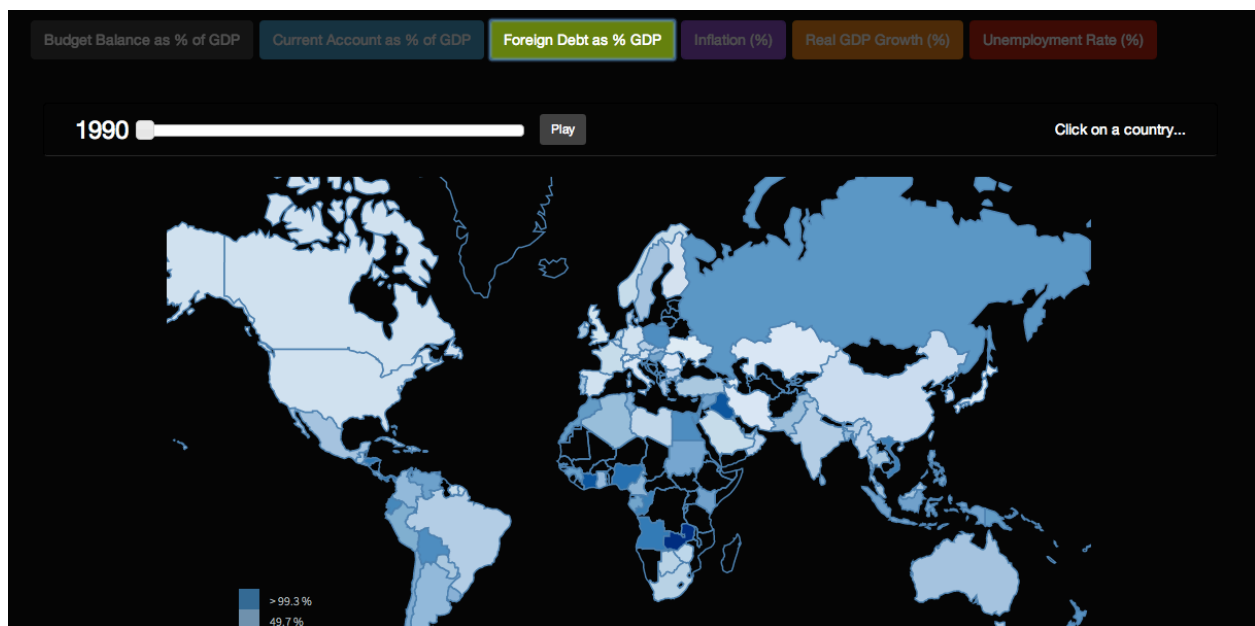
We had a few crazy ideas here and there.

1. Visual map -

Since this includes data for countries, we used a world map. We included zoom, tooltip, clickable, and also showed a heat map depending on the relative value of the indicator. A rough sketch of our preliminary idea:



Most importantly, since the indicators' values change across time, we added a simulation component when users can click play and see how the relative values (shown as colors) change across time in a continuous simulation! When users click on a country, we show a button asking them if they want to predict indicator values for the specific country.



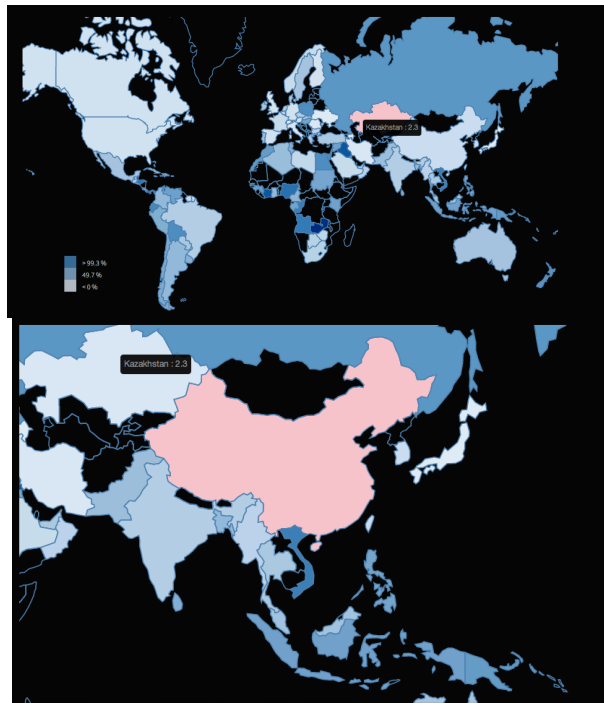
Screenshot of our final product

Along the way, we had a few challenges:

- Animating the graphs. We used the `setTimeout` function to color the graph recursively and handled the case where the values will loop around after the years go past 2013. We eventually decide to have a continuous animation so that users can look at the values changing over the years, even when there are just 13 values

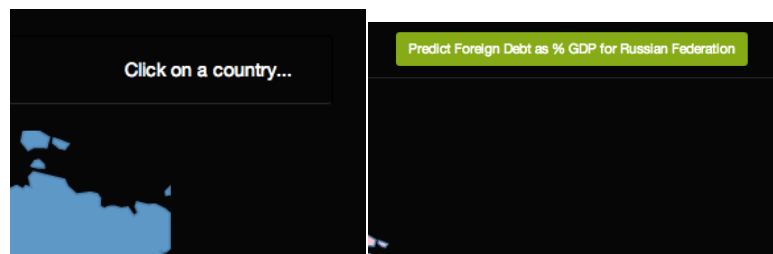
from 1990 to 2013.

- Coloring. Since we are displaying ordinal data, we decided to use different shades of the same color (instead of different colors) to represent the values for the specific indicator. As our TF Francis pointed out, we have decided to add a legend to make the colors more intelligible.
- Tooltip and zooming. These are pretty standard. But we paid particular attention to allow the countries to turn pink when we hover over it but once clicked, the selected country remains pink and hovering is disabled until the user click again to zoom out. Here are a few screenshots to illustrate this:



Notice that China still remains pink even though I am hovering over Kazakhstan.

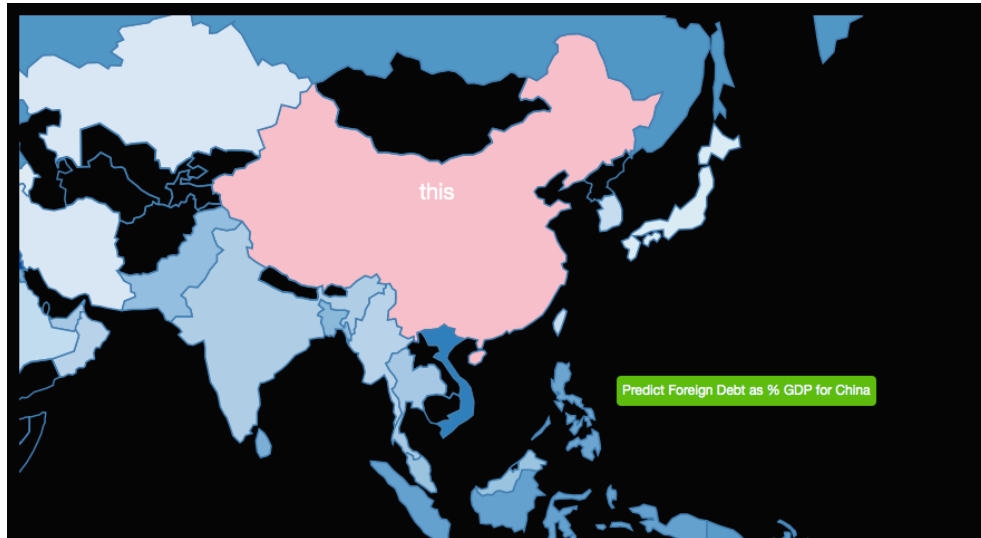
- Instructions. Much to the consternation of our TF, we used the blinking text 'Click on a country' to capture our audiences' attention.



This may be reminiscent of the HTML blink tag centuries ago, but after exploring multiple options, we decided to stick with our original implementation. We agree with Francis Kei that this method is suboptimal, especially since our users will focus their attention on the country that is clicked instead of the top right corner of the map. But after spending more than 5 hours in vain, we decide to ditch this improvement in

favor of other more pressing visualization especially for the results part later. Some of the methods we tried include:

- Putting instructions to proceed within the country as text. This works fine for some of the countries (especially large ones) but for small countries, the text size gets blown out of proportion, as in Thailand's case.
- We tried to place a div (kind of like an improvisation of the tooltip) over the map. But after zooming, the x, y position of the country gets jumbled up. As in the image, the green button is just a little off to the right of China but in the Thailand's case, it almost stretched to Indonesia.



Most importantly, most of these changes, destroy the aesthetics of the map, which forces us to stick with the original implementation of blinking instructions and button to proceed on the top right.

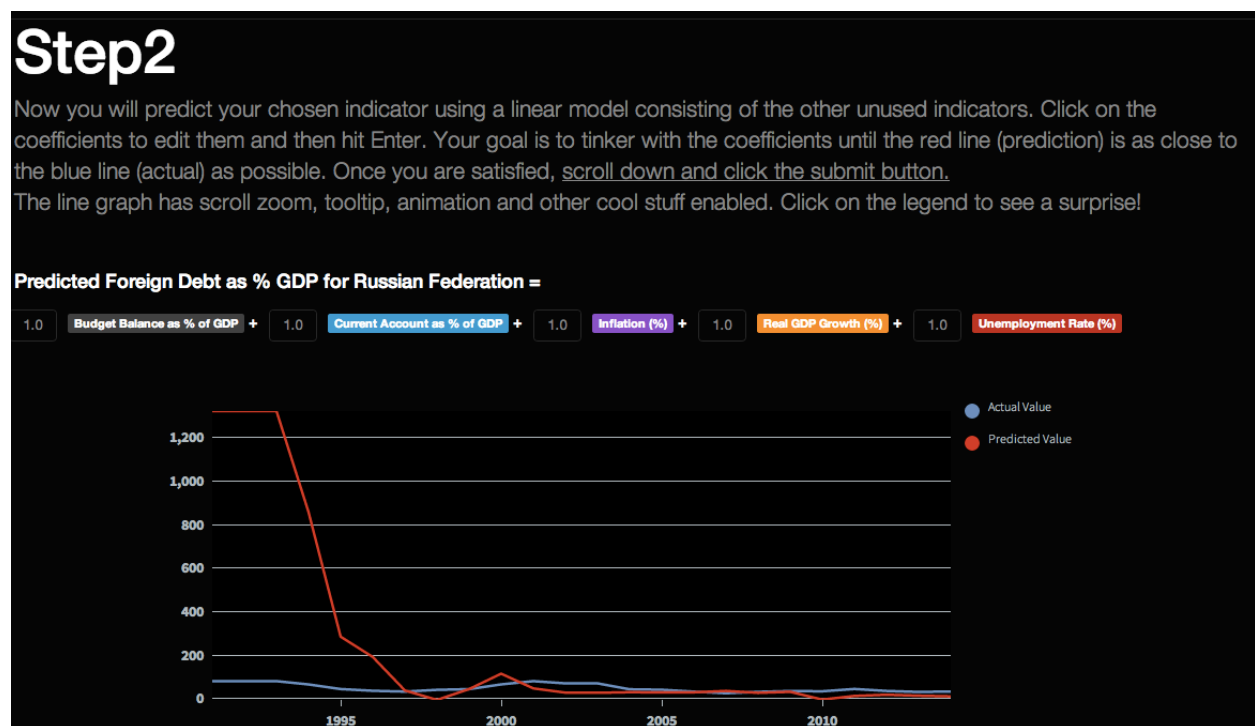
2. Line graph -

We include a line graph that shows both the actual value for the specific country's economic indicator across time as well as a predicted value. They will be distinguished by color and has awesome features like scroll zoom, click zoom, tooltip, highlight the line and even clickable legends!

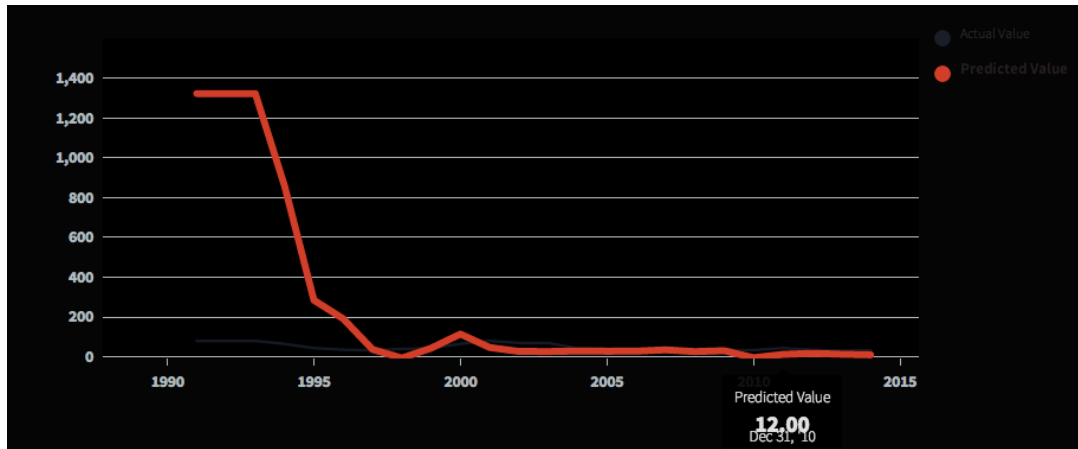
Most importantly, the line graphs have awesome transition animations!

We also have a multi-linear regression model equation above the line graph. Users can manually change the coefficients, hit enter and see how the predicted line graph changes. The goal is for the users to try all sorts of coefficients and try to bring the predicted value as close as possible to the actual value graph.

Here's a screenshot of the final product:



The implementation of it was tricky. Even though we adapted code from <https://gist.github.com/Matthew-Weber/5645518>, we have to do substantial work beyond just changing the colors and background to fit our theme. For one, the data source was in a different format so changing the data adapter took up a few hours. Secondly, we have to refactor their code to update the graph every time the user hits enter while changing the coefficients and maintain the animations. We also made sure that the graph can be zoomed using scrolling and that the tooltip (using the library of jquery.tipsy.js) function ensures that the entire line is highlighted as shown below.



Also note that there is JavaScript involved to ensure that the 5 unselected indicators are used in the model (i.e. nothing is hard coded) and every time the user hits enter, the JavaScript recalculates the predicted value from the model and replots the line.

Alternative ideas:

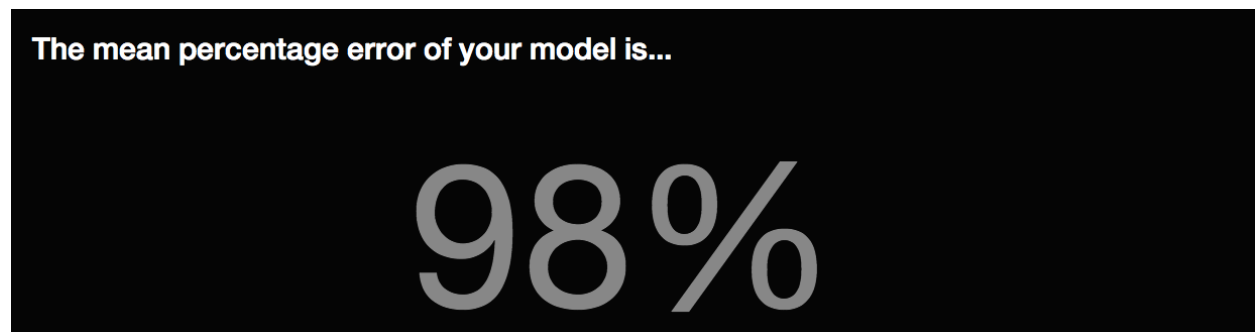
Here we also thought about having other visualizations like scatterplot where the user can explore simple linear relationships between 2 variables. We even thought about using different tabs so that the user can toggle between the line graph and the scatter plot. However, we finally decided on not including the scatter plot since it may add more complexity and hinder with the story telling aspect of our project.

4. Results Section!

After users are happy with their model, they can click submit and we will show them their mean percentage error calculated by the absolute difference for each point between their model prediction and actual value divided by the actual value.

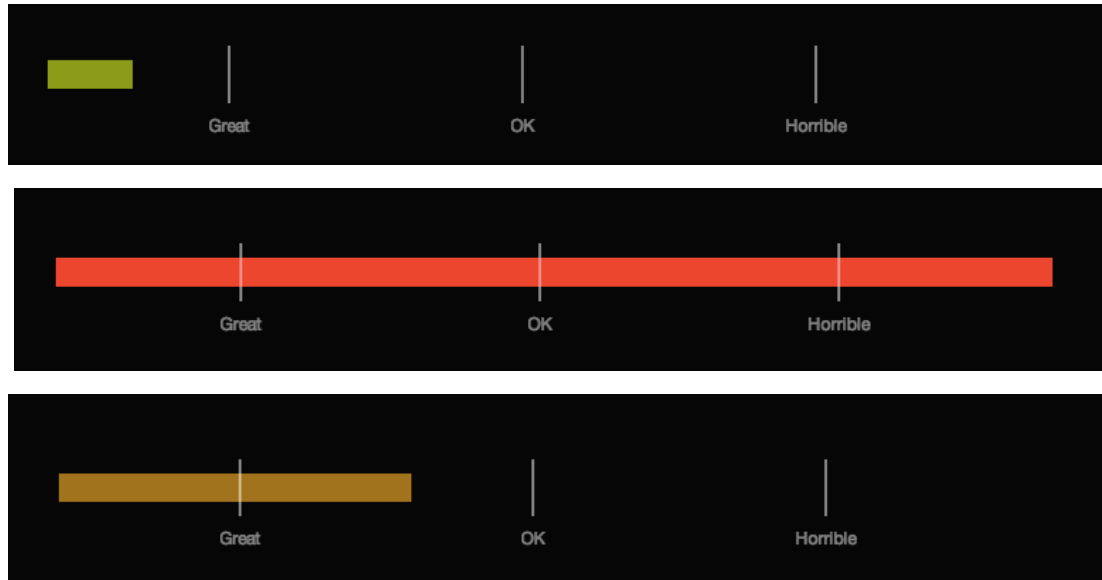
We brainstormed several visualizations including an animated donut chart telling them how much mean percentage error their model has. However, we scrapped the donut chart idea in the end since there is theoretically no upper bound on the error you can get but the donut chart can only show up to a certain limit (assuming that one whole donut means 100%).

In our first milestone, the version we submitted showed just a number:



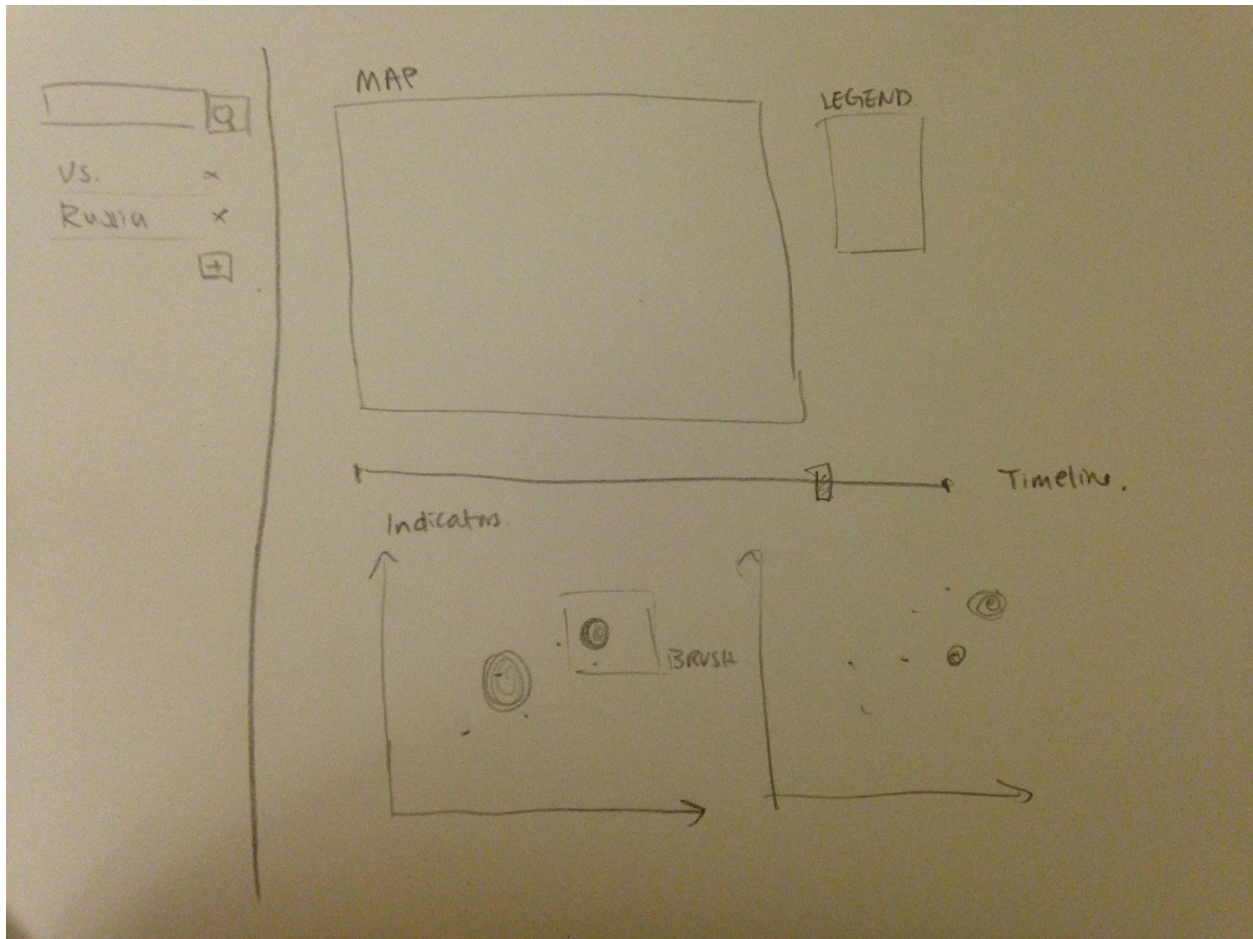
Francis grudgingly pointed out that this was 'fine'. Indeed, it is kind of a lackluster finale when the user has taken so much trouble to go through all the steps to get a mean percentage error. To spice it up, we introduced two features:

- Animation of the number from 0% to your percentage error. This adds more suspense as you see the number creep upwards.
- Animation of status bar. This is programmed from scratch using rectangles, lines, texts and transition animations (including easing). As the status bar grows from left to right, the color changes from green (for good models) to red when the error is too high!



- Humor (well we tried!)
Just a little comment on you intellectual capacity based on the your model performance. It's just a joke so don't take it too seriously if you don't score well. I mean... since when is life easy?

This sketch is from the previous process book but the general gist of the idea is still in place in our new design.



Must-Have Features. These are features without which you would consider your project to be a failure.

1. Map of countries. Hovering gives more country-specific detail (eg country name, significant indicator numbers). Users should be able to toggle between different indicator maps.
2. A 'play' button to simulate and show animations of the changes in map values (through colors) over time.
3. For each indicator, a line graph of the actual value vs predicted value for the indicators.
4. Prediction accuracy of the multi-linear regression model, as shown in a animated donut chart.
- 5.

Optional Features. Those features which you consider would be nice to have, but not critical.

1. We can explore even more time series indicators or models such as neural networks or regression based decision tree. However, admittedly, this may be less interactive since some of these classifiers may be quite foreign to the user.

Evaluation.

We learned more about fitting multi-linear regression than information about the data. In fact, after letting some of our friends play around with the site, we realized how hard a task it is to fit

a regression model. It's fortunate that we have computers today to do it!

Perhaps we have a few points of further improvements. By the end of this, we have spent more than 100 hours combined on the project but nonetheless there is room for improvements. We can explore having a detail view beside the map that gives us information about the countries. We can also have tooltips that gives us more information about the economic indicators. We tried implementing this for a couple of hours but there was some name clash problems with the JavaScript that causes jQuery tooltip not to work.

Project Schedule. Make sure that you plan your work so that you can avoid a big rush right before the final project deadline, and delegate different modules and responsibilities among your team members. Write this in terms of weekly deadlines.

April 6 - April 13

*prototype due before april 10

Nov 26 - Dec 3

Build Classifiers

Eugene - Run words through Linear Regression classifier

Grace - Run words through SVM or PCA

John - Run words through Neural Networks

April 13 - April 20

April 20 - April 27

April 27 - May 4

May 4 - May 8

Analysis and Writeup

Older Process Book

Sovereign Bond Rating Prediction

Grace Gee, Eugene Wang

Background and Motivation. Discuss your motivations and reasons for choosing this project,

especially any background or research interests that may have influenced your decision.

Sovereign bond ratings signal the amount of risk in investing in a particular country. They tend to incorporate quantitative measures of political situations, economic status, historical resilience amongst many other factors.

Our project will be composed of 2 components:

1. Analyses - Prediction of sovereign bond ratings given historical time series data. We will explore different classifiers, such as SVMs and neural networks that are well suited for classifying fixed categories.

2. Visual Exploration - Since we will have just 2 strong indicators of interest

We are very interested in the field of financial indicators and have done previous work in sentimental analysis of FOMC statements on gold. Our previous work involved building a classifier that predicts the directionality of price movements. However, this project will be our first attempt to use time series to predict fixed classes.

If time allows,

1. model accuracy
2. maps
3. more detailed country info such as matching dates.

Project Objectives. Provide the primary questions you are trying to answer with your visualization. What would you like to learn and accomplish? List the benefits.

What economic indicators and time series to predict sovereign bond ratings.

Countries with bond ratings over time

Data. From where and how are you collecting your data? If appropriate, provide a link to your data sources.

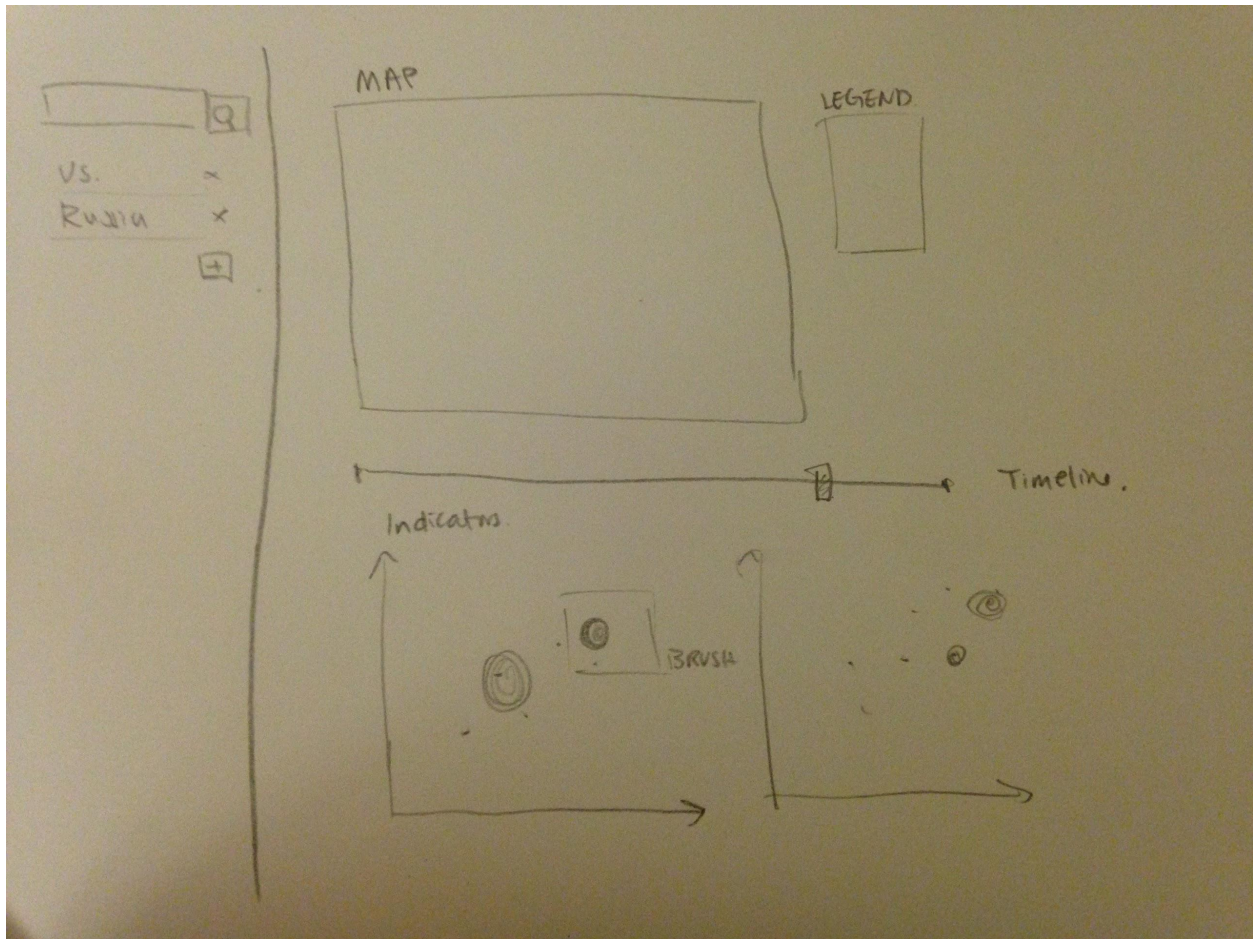
Sovereign debt ratings are provided by Moody's, Standard & Poor's and Fitch. We will be collecting Moody's ratings (the leading rating agency) from Bloomberg at HBS.

Data Processing. Do you expect to do substantial data cleanup? What quantities do you plan to derive from your data? How will data processing be implemented?

3. Features: Government debt-GDP ratio, Household debt-income, GDP, Inflation, Bond yields, Unemployment rate, PE ratio, Debt-Equity ratio
4. Target: Sovereign bond ratings are categorized into 9 sets of (AAA, AA, A, BAA, BA, B, CAA, CA, C).
5. Methodologies:

Visualization. How will you display your data? Provide some general ideas that you have for

the visualization design. Include sketches of your design.



Must-Have Features. These are features without which you would consider your project to be a failure.

Analyses

1. Test methods of interpolation of monthly, quarterly, weekly data & methods of filling missing data
2. Preprocessing features such as SVD and PCA to reduce the dimensionality of the feature set to circumvent the problem of curse of dimensionality.
3. Categorical classifiers such as decision trees, clustering, neural network and nearest neighbors.

Visualizations

6. Map of countries with bond ratings highlighted in key. Hovering gives more country-specific detail (eg country name, significant indicator numbers). Users should be able to toggle between "Actual" Moody's ratings and "Predicted" ratings maps.
7. For each feature, scatter plots of countries (eg feature vs rating).

8. Prediction accuracy of algorithm in predicting sovereign bond ratings. This number changes with slider event.
9. Timeline slider to show prediction and actual rating differences over time. Event triggers scatter plots and model accuracy.

Optional Features. Those features which you consider would be nice to have, but not critical.

2. If time permits, we will explore the relationship between the sovereign bond rating and significant economic events over time.
3. We can explore even more time series features or classifiers such as kernel-based functions including RBF neural networks.
4. Features sidebar to allow users to add/remove indices. Map and accuracy will reflect the changes in prediction from included features.

Project Schedule. Make sure that you plan your work so that you can avoid a big rush right before the final project deadline, and delegate different modules and responsibilities among your team members. Write this in terms of weekly deadlines.

March 23 - March 30

Data preprocessing & Set up Feature Analysis

Eugene - Fill in missing data/interpolate data; Literature of Classifiers used in rating classifications

Grace - Bloomberg download; Build classifier modules that take in time series features

March 30 - April 6

Visualization of results

Eugene - Build map

Grace - Build scatter plot queries

April 6 - April 13

*prototype due before april 10

Nov 26 - Dec 3

Build Classifiers

Eugene - Run words through Linear Regression classifier

Grace - Run words through SVM or PCA

John - Run words through Neural Networks

April 13 - April 20

April 20 - April 27

April 27 - May 4

May 4 - May 8

Analysis and Writeup