Part 1: Data Exploration: Graduate School Admissions

The data we will be using is admission data on Grad school acceptances.

- `admit`: whether or not the applicant was admitted to grad. school
- `gpa`: undergraduate GPA
- `gre`: score of GRE test
- `rank`: prestige of undergraduate school (1 is highest prestige, ala Harvard)

We will use the GPA, GRE, and rank of the applicants to try to predict whether or not they will be accepted into graduate school.

Before we get to predictions, we should do some data exploration.

1. Load in the dataset into pandas: `data/grad.csv`.
2. Use the pandas `describe` method to get some preliminary summary statistics on the data. In particular look at the mean values of the features.
3. Use the pandas `crosstab` method to see how many applicants from each rank of school were accepted. You should get a dataframe that looks like this:

```
rank    1   2   3   4
admit
0      28  .. .. ..
1      33  .. .. ..
```

   Make a bar plot of the percent of applicants from each rank who were accepted. You can do `.plot(kind="bar")` on a pandas dataframe.

Part 2: Predicting Graduate School Admissions

Now we're ready to try to fit our data with Logistic Regression.

We're going to start with statsmodel's implementation of Logistic Regression and then move onto sklearn's LogisticRegression .

1. Use statsmodels to fit a Logistic Regression.
2. Use the `summary` method to see your results. Look at the p-values for the beta coefficients. We would like these to be significant. Are they?
3. Once we feel comfortable with our model, we can move on to cross validation. We no longer will need all the output of statsmodels so we can switch to sklearn. Use sklearn's KFold cross validation and LogisticRegression to calculate the average accuracy, precision and recall.

   Hint: Use sklearn's implementation of these scores in sklearn.metrics .
4. The `rank` column is ordinal where we assume an equal change between ranking levels, but we could also consider it to be more generally categorical. Use panda's get_dummies to binarize the column.
5. Compute the same metrics as above. Does it do better or worse with the rank column binarized?