In the last step, the expected output was:



N\_p  N\_s  N\_t  aca\_p\_11  aca\_s\_11  aca\_t\_11  aca\_tot\_11    bn  com\_p\_11  \

0   90  NaN   22       7.8       NaN       7.9         7.9  M015       7.6

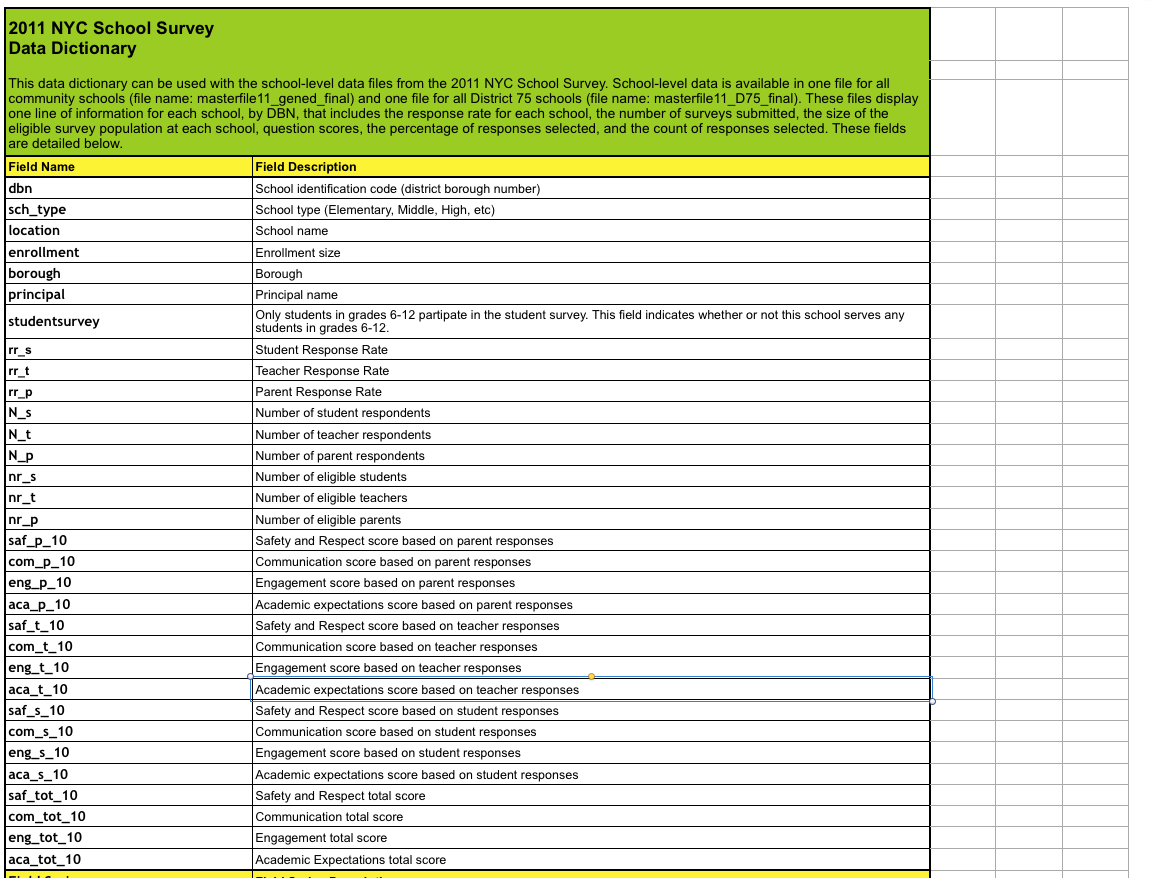
1  161  NaN   34       7.8       NaN       9.1         8.4  M019       7.6

There are two immediate facts that we can see in the data:

* There are over 2000 columns, nearly all of which we don't need. We'll have to filter the data to remove the unnecessary ones. Working with fewer columns will make it easier to print the dataframe out and find correlations within it.
* The survey data has a dbn column that we'll want to convert to uppercase (DBN). The conversion will make the column name consistent with the other data sets.

First, we'll need filter the columns to remove the ones we don't need. Luckily, there's a data dictionary at the [original data download location](https://data.cityofnewyork.us/Education/NYC-School-Survey-2011/mnz3-dyi8). The dictionary tells us what each column represents. Based on our knowledge of the problem and the analysis we're trying to do, we can use the data dictionary to determine which columns to use.

Here's a preview of the data dictionary:



Based on the dictionary, it looks like these are the relevant columns:



["dbn", "rr\_s", "rr\_t", "rr\_p", "N\_s", "N\_t", "N\_p", "saf\_p\_11", "com\_p\_11", "eng\_p\_11", "aca\_p\_11", "saf\_t\_11", "com\_t\_11", "eng\_t\_11", "aca\_t\_11", "saf\_s\_11", "com\_s\_11", "eng\_s\_11", "aca\_s\_11", "saf\_tot\_11", "com\_tot\_11", "eng\_tot\_11", "aca\_tot\_11"]

These columns will give us aggregate survey data about how parents, teachers, and students feel about school safety, academic performance, and more. It will also give us the DBN, which allows us to uniquely identify the school.

Before we filter columns out, we'll want to copy the data from the dbn column into a new column called DBN. We can copy columns like this:



survey["new\_column"] = survey["old\_column"]

Instructions

* Copy the data from the dbn column of survey into a new column in survey called DBN.
* Filter survey so it only contains the columns we listed above. You can do this using [pandas.DataFrame.loc[]](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.loc.html" \t "_blank).
  + Remember that we renamed dbn to DBN; be sure to change the list of columns we want to keep accordingly.
* Assign the dataframe survey to the key survey in the dictionary data.
* When you're finished, the value in data["survey"] should be a dataframe with 23columns and 1702 rows.

The answer ¨

# First of all import pandas

import pandas

# second explore how many lines and columns the dataset has

Count\_Row=survey.shape[0] #gives number of row count

Count\_Col=survey.shape[1] #gives number of col count

# print these numbers

print(Count\_Row)

print("###################################################")

print(Count\_Col)

# thired rename the dbn column to DBN with this code below

# survey["new\_column"] = survey["old\_column"]

survey["DBN"] = survey["dbn"]

# fourth, store the relevant columns to filter the dataset in a next step

# hhmm please change dbn to DBN here OTHERWISE you will miss DBN

relevant\_columns = ["DBN", "rr\_s", "rr\_t", "rr\_p", "N\_s", "N\_t", "N\_p", "saf\_p\_11", "com\_p\_11", "eng\_p\_11", "aca\_p\_11", "saf\_t\_11", "com\_t\_11", "eng\_t\_11", "aca\_t\_11", "saf\_s\_11", "com\_s\_11", "eng\_s\_11", "aca\_s\_11", "saf\_tot\_11", "com\_tot\_11", "eng\_tot\_11", "aca\_tot\_11"]

# fifth finally filter the dataset with loc hmmm remember the difference with iloc

survey = survey.loc[:,relevant\_columns]

# create an empty dictionary

data = {}

# here you add a key named survey in the dictionary with value the whole survey fields

data["survey"] = survey

print(survey.head)

# print these numbers after the transformation

Count\_Row\_last=survey.shape[0] #gives number of row count

Count\_Col\_last=survey.shape[1] #gives number of col count

print(Count\_Row\_last)

print("###################################################")

print(Count\_Col\_last)

# optional

# print(data)

print("###################################################")

# confirm that you have added a key

print(data.keys())

# When you're finished, the value in data["survey"] should be a dataframe with 23columns and 1702 row