Read in the data

In [2]:

```
import pandas as pd
import numpy as np
import re
data_files = [
    "ap_2010.csv",
    "class_size.csv",
    "demographics.csv",
    "graduation.csv",
    "hs_directory.csv",
    "sat_results.csv"
]
data = {}
for f in data_files:
    d = pd.read_csv('schools/{0}'.format(f))
    data[f.replace('.csv', '')] = d
```

Read in the surveys

In [3]:

```
all_survey = pd.read_csv('schools/survey_all.txt', delimiter='\t', encoding='windows-1252')
d75_survey = pd.read_csv('schools/survey_d75.txt', delimiter='\t', encoding='windows-1252')
survey = pd.concat([all_survey, d75_survey], axis=0)

survey['DBN'] = survey['dbn']

survey_fields = [
    "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",
]

survey = survey.loc[:, survey_fields]
data['survey'] = survey
```

Add DBN columns

In [4]:

```
data['hs_directory']['DBN'] = data['hs_directory']['dbn']

def pad_csd(num):
    string_representation = str(num)
    if len(string_representation) > 1:
        return string_representation
    else:
        return "0" + string_representation

data['class_size']['padded_csd'] = data['class_size']['CSD'].apply(pad_csd)
data['class_size']['DBN'] = data['class_size']['padded_csd'] + data['class_size']['
'SCHOOL CODE']
```

Convert columns to numeric

In [5]:

```
cols = ['SAT Math Avg. Score', 'SAT Critical Reading Avg. Score', 'SAT Writing Avg.
Score']
for c in cols:
    data["sat_results"][c] = pd.to_numeric(data["sat_results"][c], errors="coerce")

data['sat_results']['sat_score'] = data['sat_results'][cols[0]] + data['sat_result
s'][cols[1]] + data['sat_results'][cols[2]]

def find_lat(loc):
    coords = re.findall("\(.+, .+\)", loc)
    lat = coords[0].split(",")[0].replace("(", "")
    return lat

def find_lon(loc):
    coords = re.findall("\(.+, .+\)", loc)
    lon = coords[0].split(",")[1].replace(")", "").strip()
    return lon

data['hs_directory']['lat'] = data['hs_directory']['Location 1'].apply(find_lat)
data['hs_directory']['lon'] = data['hs_directory']['Location 1'].apply(find_lon)

data['hs_directory']['lat'] = pd.to_numeric(data['hs_directory']['lat'], errors='co
erce')
data['hs_directory']['lon'] = pd.to_numeric(data['hs_directory']['lon'], errors='co
erce')
```

Condense datasets

In [6]:

```
class_size = data['class_size']
class_size = class_size[class_size['GRADE '] == '09-12']
class_size = class_size[class_size['PROGRAM TYPE'] == 'GEN ED']

class_size = class_size.groupby('DBN').agg(np.mean)
class_size.reset_index(inplace=True)
data['class_size'] = class_size

data['demographics'] = data['demographics'][data['demographics']['schoolyear'] == 20112012]

data['graduation'] = data['graduation'][data['graduation']['Cohort'] == '2006']
data['graduation'] = data['graduation'][data['graduation']['Demographic'] == 'Total Cohort']
```

Convert AP scores to numeric

In [7]:

```
cols = ['AP Test Takers ', 'Total Exams Taken', 'Number of Exams with scores 3 4 or 5']

for col in cols:
    data['ap_2010'][col] = pd.to_numeric(data['ap_2010'][col], errors='coerce')
```

Combine the datasets

In [8]:

```
combined = data['sat_results']

combined = combined.merge(data['ap_2010'], on='DBN', how='left')
combined = combined.merge(data['graduation'], on='DBN', how='left')

to_merge = ['class_size', 'demographics', 'survey', 'hs_directory']

for m in to_merge:
    combined = combined.merge(data[m], on='DBN', how='inner')

combined = combined.fillna(combined.mean())
combined = combined.fillna(0)
```

Add a school district column for mapping

In [9]:

```
def get_first_two_chars(dbn):
    return dbn[0:2]

combined['school_dist'] = combined['DBN'].apply(get_first_two_chars)
```

Find correlations

In [10]:

```
correlations = combined.corr()
correlations = correlations['sat_score']
correlations
```

Out[10]:

```
SAT Critical Reading Avg. Score    0.986820
SAT Math Avg. Score                0.972643
SAT Writing Avg. Score             0.987771
sat_score                          1.000000
AP Test Takers                     0.523140
                                     ...
Census Tract                       0.048737
BIN                                0.052232
BBL                                0.044427
lat                               -0.121029
lon                               -0.132222
Name: sat_score, Length: 85, dtype: float64
```

Plotting survey correlations

In [11]:

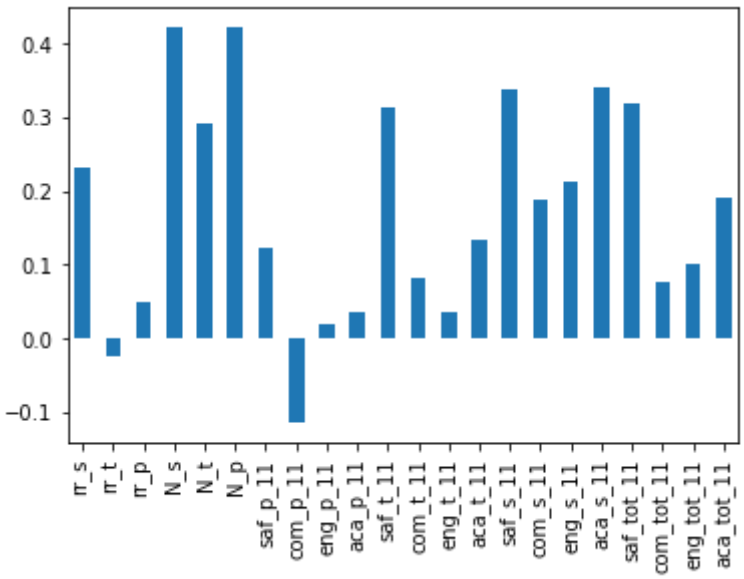
```
survey_fields.remove('DBN')
```

In [12]:

```
%matplotlib inline
combined.corr()['sat_score'][survey_fields].plot.bar()
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7b9fddc0>
```



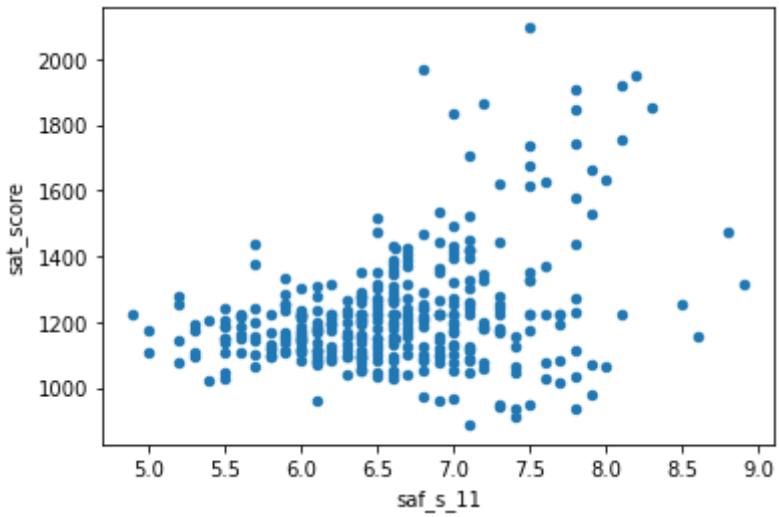
Exploring safety

In [13]:

```
combined.plot.scatter(x='saf_s_11', y='sat_score')
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7bb29fd0>
```



In [14]:

```
combined['saf_s_11'].mean()
```

Out[14]:

```
6.611666666666661
```

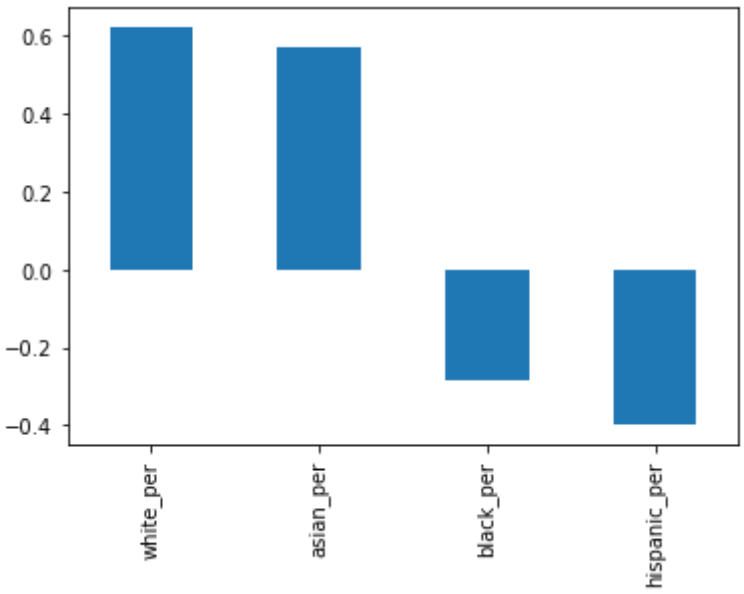
Exploring Race and SAT Scores

In [15]:

```
race_fields = ['white_per', 'asian_per', 'black_per', 'hispanic_per']
combined.corr()['sat_score'][race_fields].plot.bar()
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7bd87b50>
```

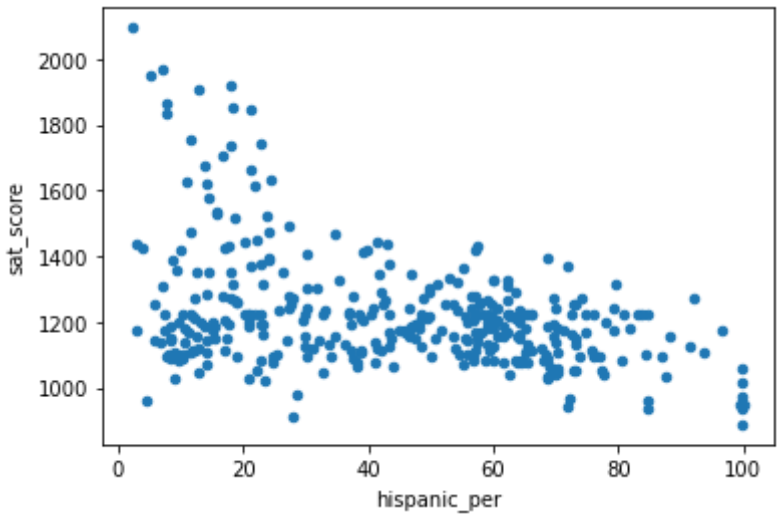


In [16]:

```
combined.plot.scatter(x='hispanic_per', y='sat_score')
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7bb5eaf0>
```



In [17]:

```
combined[combined['hispanic_per'] > 95]['SCHOOL NAME']
```

Out[17]:

```
44                          MANHATTAN BRIDGES HIGH SCHOOL
82       WASHINGTON HEIGHTS EXPEDITIONARY LEARNING SCHOOL
89     GREGORIO LUPERON HIGH SCHOOL FOR SCIENCE AND M...
125                  ACADEMY FOR LANGUAGE AND TECHNOLOGY
141                INTERNATIONAL SCHOOL FOR LIBERAL ARTS
176     PAN AMERICAN INTERNATIONAL HIGH SCHOOL AT MONROE
253                             MULTICULTURAL HIGH SCHOOL
286               PAN AMERICAN INTERNATIONAL HIGH SCHOOL
Name: SCHOOL NAME, dtype: object
```

In [18]:

```
(combined[(combined['hispanic_per'] < 10) & (combined['sat_score'] > 1800)]['SCHOOL
NAME'])
```

Out[18]:

```
37                                STUYVESANT HIGH SCHOOL
151                         BRONX HIGH SCHOOL OF SCIENCE
187                       BROOKLYN TECHNICAL HIGH SCHOOL
327    QUEENS HIGH SCHOOL FOR THE SCIENCES AT YORK CO...
356                   STATEN ISLAND TECHNICAL HIGH SCHOOL
Name: SCHOOL NAME, dtype: object
```

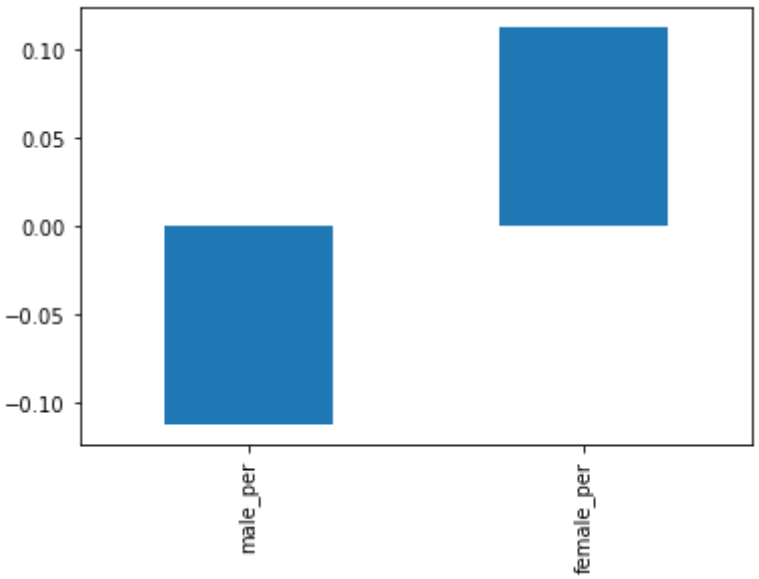
Exploring Gender and SAT Scores

In [19]:

```
gender_fields = ['male_per', 'female_per']
combined.corr()['sat_score'][gender_fields].plot.bar()
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7bed4df0>
```

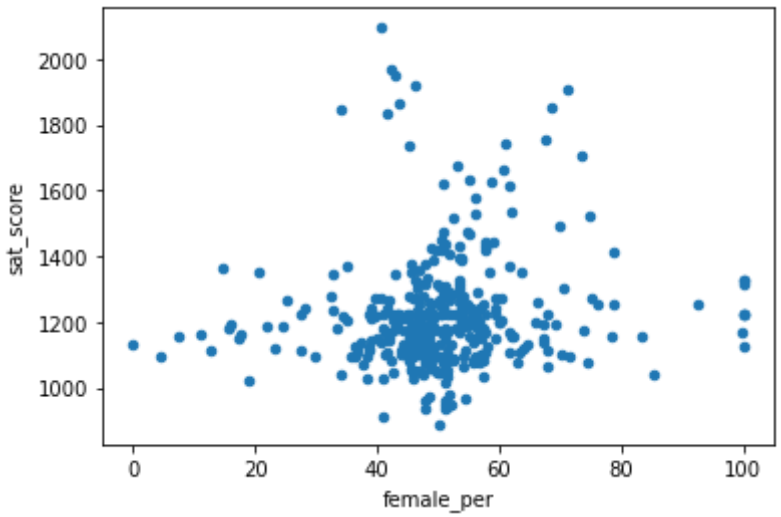


In [20]:

```
combined.plot.scatter(x='female_per', y='sat_score')
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7c02ea30>
```



In [21]:

```
(combined[(combined['female_per'] > 60) & (combined['sat_score'] > 1700)]['SCHOOL NAME'])
```

Out[21]:

```
5                          BARD HIGH SCHOOL EARLY COLLEGE
26                          ELEANOR ROOSEVELT HIGH SCHOOL
60                                       BEACON HIGH SCHOOL
61     FIORELLO H. LAGUARDIA HIGH SCHOOL OF MUSIC & A...
302                           TOWNSEND HARRIS HIGH SCHOOL
Name: SCHOOL NAME, dtype: object
```

Exploring AP Scores vs. SAT Scores

In [22]:

```
combined['ap_per'] = combined['AP Test Takers ']/combined['total_enrollment']
```

In [23]:

```
combined.plot.scatter(x='ap_per', y='sat_score')
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe7c1052b0>
```