DATA YOUR TURN #2

cities = []
with open('Cities.csv','rU') as f:
    rows = csv.DictReader(f)
    for r in rows:
        cities.append(r)

countries = []
with open('Countries.csv','rU') as f:
    rows = csv.DictReader(f)
    for r in rows:
        countries.append(r)

for city in cities:
    for country in countries:
        if city['country'] == country['country']:
            if country['country'] == 'United Kingdom':
                UK.append(float(city['temperature']))
            elif country['EU'] == 'yes':
                EU.append(float(city['temperature']))
            else:
                nonEU.append(float(city['temperature']))

print('EU before Brexit:'), np.average(EU+UK)
print('Non-EU before Brexit:'), np.average(nonEU)
print('EU after Brexit:'), np.average(EU)
print('Non-EU after Brexit:'), np.average(nonEU+UK)

ANSWER:
EU before Brexit: 9.69413333333
Non-EU before Brexit: 9.03047619048
EU after Brexit: 9.79321167883
Non-EU after Brexit: 8.96539473684

DATA YOUR TURN #3

countrylist = [] # list of countries in cities data
for city in cities:
    if city['country'] not in countrylist:
        countrylist.append(city['country'])

minval = 100.00
mincountry = "
maxval = -100.00
maxcountry = "
for country in countrylist:
    temps = []
    for city in cities:
if city['country'] == country:
    temps.append(float(city['temperature']))
avg = np.average(temps)
if avg < minval:
    minval = avg
    mincountry = country
if avg > maxval:
    maxval = avg
    maxcountry = country
print 'Minimum average temperature:', mincountry, 'with', minval
print 'Maximum average temperature:', maxcountry, 'with', maxval

ANSWER:
Minimum average temperature: Finland with 3.4875
Maximum average temperature: Greece with 16.9025

PLOTTING YOUR TURN #1

x = []
y = []
colors = []
for city in cities:
    x.append(city['longitude'])
    y.append(city['latitude'])
for country in countries:
    if city['country'] == country['country']:
        if country['EU'] == 'yes':
            colors.append('blue')
        else:
            colors.append('red')
plt.xlabel('longitude')
plt.ylabel('latitude')
plt.scatter(x, y, c=colors)
plt.show()

PLOTTING YOUR TURN #2

withc = 0
withoutc = 0
for country in countries:
    if country['coastline'] == 'yes':
        withc += 1
    else:
        withoutc += 1
bars = ['With Coastline', 'Without Coastline']
heights = [withc, withoutc]
plt.xticks([1,2], bars)
PLOTTING YOUR TURN #3

```python
numcold = 0
numwarm = 0
numhot = 0
for city in cities:
    if float(city['temperature']) < 8:
        numcold += 1
    elif float(city['temperature']) < 12:
        numwarm += 1
    else:
        numhot += 1
sizes = [numcold, numwarm, numhot]
lbls = ['cold', 'warm', 'hot']
plt.pie(sizes, labels=lbls, autopct='%1.1f%%')
plt.show()
```

PANDAS ELIMINATING TEMP2, TEMP2, TEMP1

```python
import pandas as pd
cities[(cities.latitude > 50) & (cities.temperature > 9)]
[['city', 'longitude']].sort_values('longitude')
```

PANDAS YOUR TURN #1

```python
f = open('Countries.csv', 'rU')
countries = pd.read_csv(f)
countries[(countries.EU == 'no') & (countries.coastline == 'no')][['country', 'population']].sort_values('population')
```

PANDAS YOUR TURN #2

```python
countries.groupby('coastline').mean()['population']
countries.groupby(['coastline', 'EU']).mean()['population']
```

PANDAS YOUR TURN #3

```python
citiesext = cities.merge(countries, on='country')
cities2 = citiesext[(citiesext.city.str.contains('k')) & (citiesext.coastline == 'yes')].sort_values('temperature')
print 'Average temperature:', cities2.temperature.mean()
print 'Second-coldest:', cities2.head(2).tail(1)[['city', 'country']]
print 'Second-warmest:', cities2.tail(2).head(1)[['city', 'country']]
```