Solutions to Lab 3

Lab 3a Solution

This is a quiz given in Roger Peng Coursera (https://www.coursera.org) class Computing for Data Analysis (https://www.coursera.org/course/compdata).

In [1]:
```
import pandas as pd
import os

data = pd.read_csv(os.path.join('data', 'ozone.csv'))
```

In [2]:
```
print data.head()
```

```
   Ozone  Solar.R  Wind  Temp  Month  Day
0     41      190   7.4    67      5    1
1     36      118   8.0    72      5    2
2     12      149  12.6    74      5    3
3     18      313  11.5    62      5    4
4    NaN      NaN  14.3    56      5    5
```

Print the column names of the dataset to the screen, one column name per line.

In [3]:
```
for x in data.columns.values:
    print x
```

```
Ozone
Solar.R
Wind
Temp
Month
Day
```

Extract the first 2 rows of the data frame and print them to the console. What does the output look like?

In [4]:
```
tmp = data.ix[0:1] # or data.head(2)
print tmp.head()
```

```
   Ozone  Solar.R  Wind  Temp  Month  Day
0     41      190   7.4    67      5    1
1     36      118   8.0    72      5    2
```

How many observations (i.e. rows) are in this data frame?

In [5]:
```
print len(data)
```

```
153
```

Extract the last 2 rows of the data frame and print them to the console. What does the output look like?

In [6]:
```
tmp = data.tail(2)
print tmp.head()
```

```
     Ozone  Solar.R  Wind  Temp  Month  Day
151     18      131   8.0    76      9   29
152     20      223  11.5    68      9   30
```

What is the value of Ozone in the 47th row?

In [7]:
```
print data.ix[46:48,]
```

```
    Ozone  Solar.R  Wind  Temp  Month  Day
46     21      191  14.9    77      6   16
47     37      284  20.7    72      6   17
48     20       37   9.2    65      6   18
```

How many missing values are in the Ozone column of this data frame?

In [8]:
```
print data['Ozone'].isnull().sum()
print len(data) - len(data['Ozone'].dropna())
```

```
37
37
```

What is the mean of the Ozone column in this dataset? Exclude missing values (coded as NA) from this calculation.

In [9]:
```
print data['Ozone'].mean()
```

```
42.1293103448
```

Extract the subset of rows of the data frame where Ozone values are above 31 and Temp values are above 90. What is the mean of Solar.R in this subset?

In [10]:
```
print data[(data.Ozone > 31) & (data.Temp > 90)].head()
```

```
     Ozone  Solar.R  Wind  Temp  Month  Day
68      97      267   6.3    92      7    8
69      97      272   5.7    92      7    9
119     76      203   9.7    97      8   28
```

```
120    118      225   2.3    94      8   29
121     84      237   6.3    96      8   30
```

In [11]:
```
print data[(data.Ozone > 31) & (data.Temp > 90)]['Solar.R'].mean()
```

```
212.8
```

What is the mean of "Temp" when "Month" is equal to 6?

In [12]:
```
print data[ data.Month==6 ].Temp.mean()
print data[ data.Month==6 ]['Temp'].mean()
```

```
79.1
79.1
```

What was the maximum ozone value in the month of May (i.e. Month = 5)?

In [13]:
```
print data[ data.Month==5 ].Ozone.max()
```

```
115.0
```

Lab 3b Solution:

Via Pandas

In [ ]:
```
# Read Data
df = pd.read_csv(data_path)
print df.head()
```

In [ ]:
```
# What percent survived?
df.groupby('survived')['survived'].count()
```

In [ ]:
```
#Write a function
def titanic_function(data):
    cnts =  data.groupby('survived')['survived'].count()
    return {'survived': cnts[1], 'not survived': cnts[0]}
```

In [ ]:
```
print titanic_function(df)
```

In [ ]:
```
# What percent of males survived? Females?
```

In []:
```
# What percent of males survived? Females?
print df.groupby(['sex','survived'])['survived'].count()
```

San-Pandas (for comparison)

In []:
```
import os, csv
data_path = os.path.join('data','titanic.csv')
print data_path

with open(data_path,'r') as infile:
    reader = csv.reader(infile)
    data = list(reader)


print len(data)
```

In []:
```
### What percent of the people survied?

survived = 0
for d in data:
    try:
        if int(d[0]) == 1:
            survived+=1
    except ValueError:
        pass

print survived/float(len(data))*100
```

In []:
```
### Function

def titanic_function(data):
    tmp = {}
    for d in data:
        try:
            if int(d[0]) == 1:
                tmp['survived'] = tmp.get('survived',0) + 1
            else:
                tmp['not survived'] = tmp.get('not survived',0) + 1
        except ValueError:
                tmp['unknown'] = tmp.get('unknown',0) + 1
    return tmp

print titanic_function(data)
```

In []:
```
###What percent of males survived? Females?
```

```
def titanic_function(data):
    M = {}
    F = {}
    for d in data:
        try:
            if d[3] == 'male':
                if int(d[0]) == 1:
                    M['survived'] = M.get('survived',0) + 1
                else:
                    M['not survived'] = M.get('not survived',0) + 1
            elif d[3] == 'female':
                if int(d[0]) == 1:
                    F['survived'] = F.get('survived',0) + 1
                else:
                    F['not survived'] = F.get('not survived',0) + 1
            else:
                pass
        except ValueError:
            pass
    return {'male': M, 'female': F}

val = titanic_function(data)

print val['male']['survived']/float( val['male']['survived'] + val['male']['n
ot survived'])*100
print val['female']['survived']/float(val['female']['survived'] + val['female
']['not survived'])*100
```