TP2_ANIS

February 10, 2020

1 Chargement et prétraitements des données

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

        data= pd.read_csv("study.csv",delimiter=",")
        data.head()

        data.gender = data.gender.astype('category') # ou bien pd.Categorical(data.gender)
        data.ethnicity = data.ethnicity.astype('category')


        i = data[(data.age <= 15)].index  # Je cherche l'index de ceux dont l'age est inferieu
        data=data.drop(i)
```

```
In [2]: pd.Categorical(data.gender).describe()
```

```
Out[2]:             counts     freqs
        categories
        Female         507  0.620563
        Male           310  0.379437
```

```
In [3]: pd.Categorical(data.ethnicity).describe()
```

```
Out[3]:             counts     freqs
        categories
        Asian           27  0.033048
        Black           88  0.107711
        Dominican        1  0.001224
        Hispanic        75  0.091799
        Indian           1  0.001224
        Other            6  0.007344
        Unknown         32  0.039168
        White          587  0.718482
```

```
In [4]: data.describe()
```

```
Out[4]:              age      weight    protein   protein2    protein3   n_visits
        count  817.000000  817.000000  817.000000  817.000000  817.000000  817.000000
        mean    41.994002   67.997307  244.293758  137.565483  100.981640    2.395349
        std     21.623043   10.386467   46.767645   39.445960   29.033465    1.987492
        min     15.100000   45.800000  140.000000   30.000000   50.000000    0.000000
        25%     23.200000   60.600000  208.000000  111.000000   76.000000    1.000000
        50%     37.100000   67.200000  245.000000  139.000000  102.000000    2.000000
        75%     60.100000   74.400000  279.000000  164.000000  124.000000    3.000000
        max     94.600000   95.700000  361.000000  227.000000  150.000000    8.000000
```

```
In [ ]:
```

2 Tableau des fréquences

```
In [5]: xi, ni = np.unique(data.n_visits, return_counts=True)
        table=pd.DataFrame( data=ni, columns=["ni"],index=xi)
        N=sum(table.ni)
        table.insert(1, "fi", table.ni / N, True)
        table.insert(2,"Fi",np.cumsum(table.fi), True)
        table
```

```
Out[5]:    ni        fi        Fi
        0  154  0.188494  0.188494
        1  160  0.195838  0.384333
        2  145  0.177479  0.561812
        3  156  0.190942  0.752754
        4  111  0.135863  0.888617
        5   18  0.022032  0.910649
        6   27  0.033048  0.943696
        7   25  0.030600  0.974296
        8   21  0.025704  1.000000
```

```
In [6]: data.n_visits.describe()
```

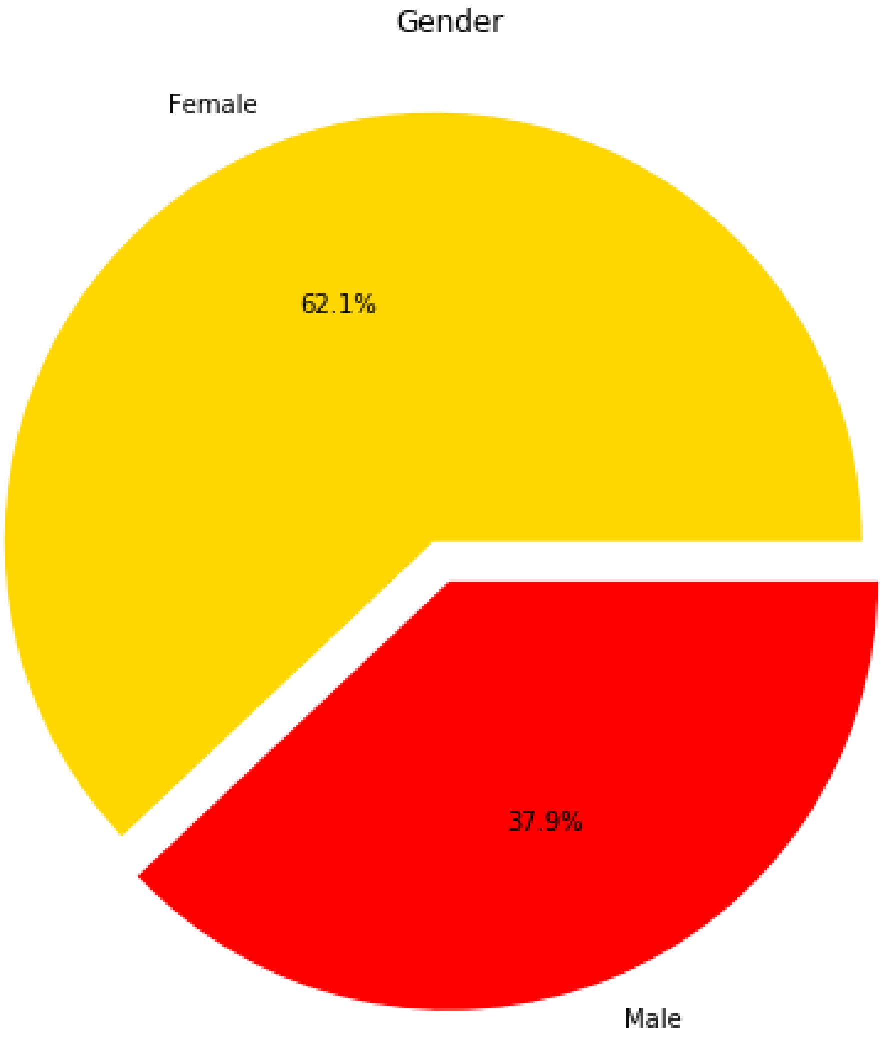
```
Out[6]: count    817.000000
        mean       2.395349
        std        1.987492
        min        0.000000
        25%        1.000000
        50%        2.000000
        75%        3.000000
        max        8.000000
        Name: n_visits, dtype: float64
```

3 Représentation graphique unidimensionelle

```
In [7]: plt.figure(figsize=(8,8))
        vals, counts= np.unique(data.gender, return_counts=True)
```

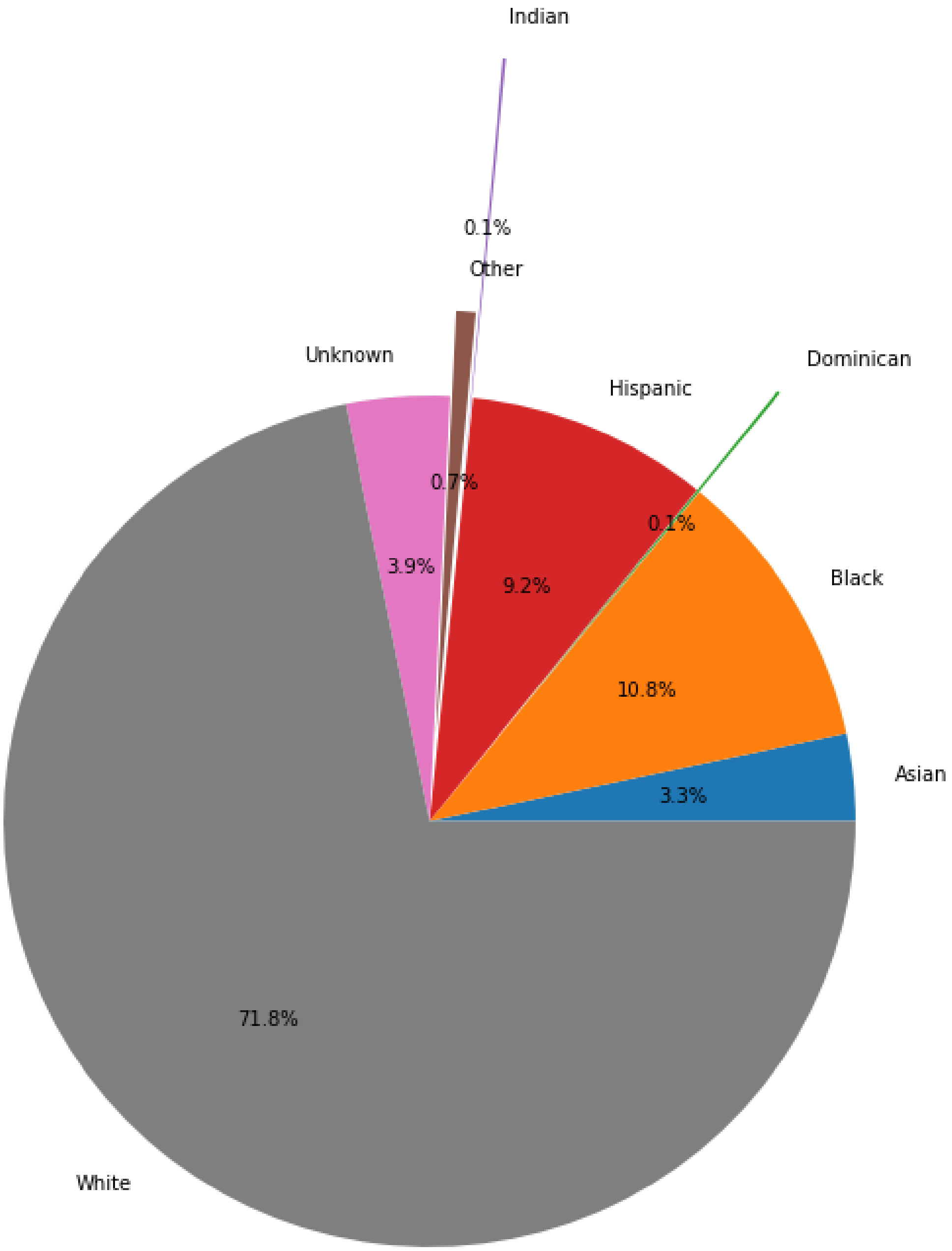
```
labels = vals
sizes = counts
colors = ['gold', "red"]
explode = (0.1, 0)  # explode 1st slice

# Plot
plt.title('Gender')
plt.pie(sizes,explode=explode, labels=labels, colors=colors,autopct='%1.1f%%');
```



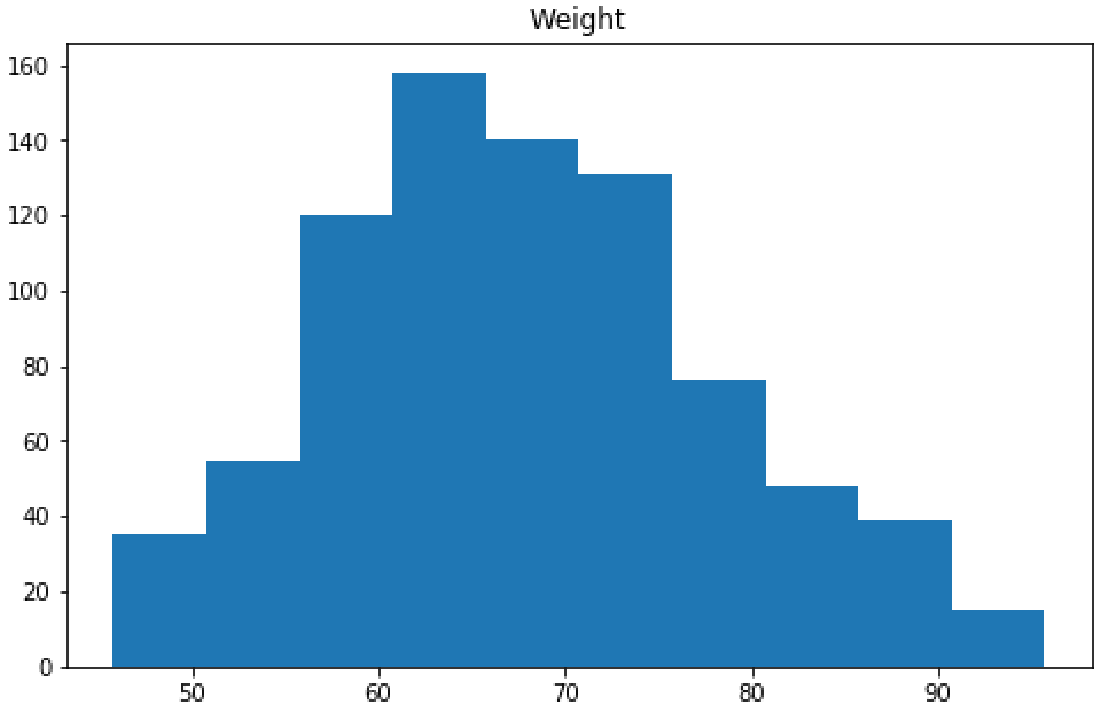
```
In [8]: plt.figure(figsize=(15,10))
        vals, counts= np.unique(data.ethnicity, return_counts=True)
```

```
sizes=counts
labels= 'Asian', 'Black', 'Dominican', 'Hispanic', 'Indian', 'Other', 'Unknown', 'White
explode=(0, 0,0.3,0,0.8,0.2,0,0)
plt.pie(sizes,explode=explode,labels=labels,autopct='%1.1f%%');
```

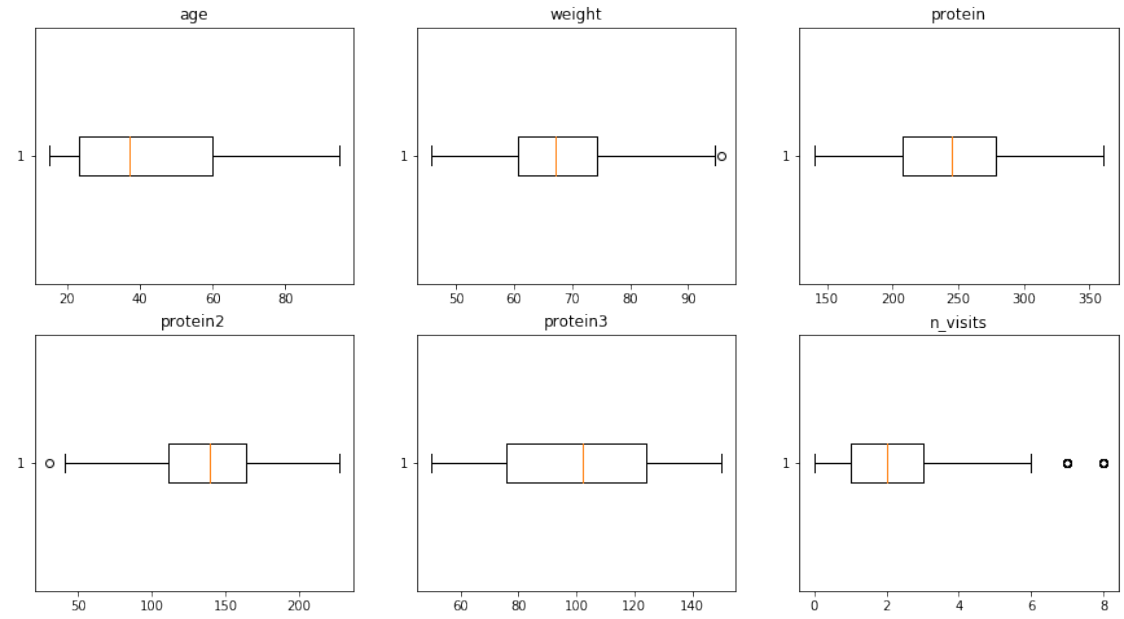


```
In [9]: plt.figure(figsize=(8,5))
```

```
plt.hist(data.weight);
plt.title("Weight");
```

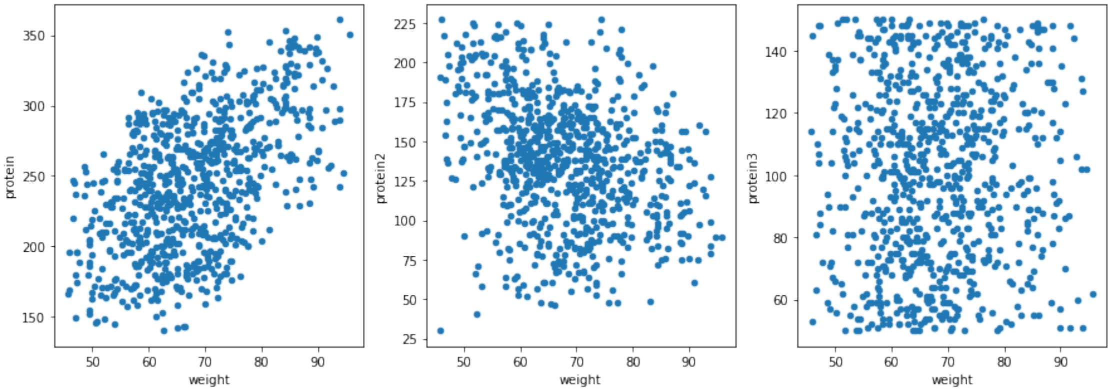


```
In [10]: plt.figure(figsize=(15,8))
         plt.subplot(231)
         plt.title('age')
         plt.boxplot(data.age,vert=False);
         plt.subplot(232)
         plt.title('weight')
         plt.boxplot(data.weight,vert=False);
         plt.subplot(233)
         plt.title('protein')
         plt.boxplot(data.protein,vert=False);
         plt.subplot(234)
         plt.title('protein2')
         plt.boxplot(data.protein2,vert=False);
         plt.subplot(235)
         plt.title('protein3')
         plt.boxplot(data.protein3,vert=False);
         plt.subplot(236)
         plt.title('n_visits')
         plt.boxplot(data.n_visits,vert=False);
```



4 Représentation graphique multidimensionelle

```
In [11]: fig = plt.figure(figsize=(15,5));
         ax1 = fig.add_subplot(131);
         ax2 = fig.add_subplot(132);
         ax3 = fig.add_subplot(133);
         data.plot(kind='scatter',x='weight', y='protein', ax=ax1, legend=False);
         data.plot(kind='scatter',x='weight', y='protein2', ax=ax2, legend=False);
         data.plot(kind='scatter',x='weight', y='protein3', ax=ax3, legend=False);
```

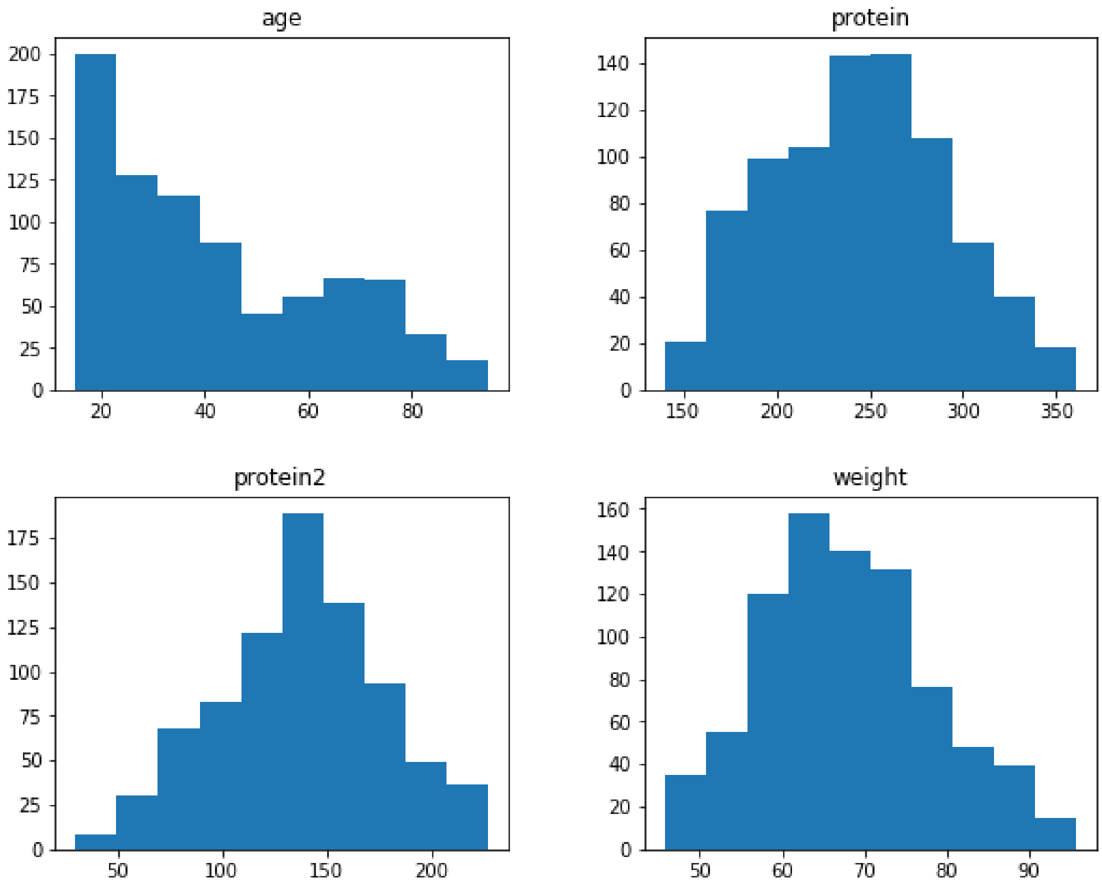


```
In [12]: data.drop(["n_visits",'age','gender','ethnicity'], axis=1).corr()
```

```
Out[12]:             weight   protein  protein2  protein3
         weight    1.000000  0.527940 -0.304254  0.043547
         protein   0.527940  1.000000 -0.221852  0.009820
         protein2 -0.304254 -0.221852  1.000000 -0.019180
         protein3  0.043547  0.009820 -0.019180  1.000000
```

```
In [13]: fig = plt.figure(figsize = (10,8));
         ax = fig.gca();
         data.drop(["n_visits",'protein3'], axis=1).hist(grid=False,ax=ax);
```

```
/Users/macbookpro/anaconda/lib/python3.5/site-packages/IPython/core/interactiveshell.py:2961: U
  exec(code_obj, self.user_global_ns, self.user_ns)
```



```
In [14]: data.drop(["n_visits",'gender','ethnicity'], axis=1).skew(axis = 0, skipna = True)
```

```
Out[14]: age          0.589952
         weight       0.280581
         protein      0.052732
         protein2    -0.064582
         protein3    -0.039009
         dtype: float64
```

5 Recodage d’une variable

```
In [15]: data.age.describe()

Out[15]: count    817.000000
         mean      41.994002
         std       21.623043
         min       15.100000
         25%       23.200000
         50%       37.100000
         75%       60.100000
         max       94.600000
         Name: age, dtype: float64

In [16]: age5_cat=pd.qcut(data.age, 5)
         data.insert(8,"age5_cat",age5_cat, True)

In [17]: data.head()

Out[17]:    gender ethnicity   age  weight  protein  protein2  protein3  n_visits  \
         0  Female     White  72.0    76.0      246        88       136         8
         1  Female     Black  84.1    59.8      210        85        86         6
         2  Female     Black  79.7    56.0      205        91       110         7
         3  Female     White  75.7    66.7      286        68        54         2
         4  Female     White  74.6    72.1      171        81        99         8

                age5_cat
         0  (66.78, 94.6]
         1  (66.78, 94.6]
         2  (66.78, 94.6]
         3  (66.78, 94.6]
         4  (66.78, 94.6]

In [ ]:

In [ ]:

In [ ]:
```