Lab Report

Course Code: CSE 316

Course Name: Artificial Intelligence Lab

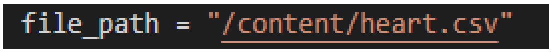
Lab Report No: 01

Date of submission : 12-08-2023

Submitted To :	**Submitted By:**
Md. Sadekur Rahman	**Name:** Rayhan Rafin
Assistant Professor	**ID**: 213-15-4278
Department of CSE	**Section:** 60_B
Daffodil International University	

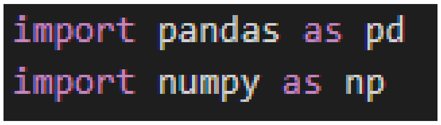
Insertion and Naming Data Frame:

In order to insert the data frame we have to specify its file path. To do that we need to write the code :



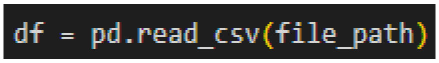
```
file_path = "/content/heart.csv"
```

This will add the file to the environment. Next we need to import python libraries to manipulate and view the data in the csv file. We can import pandas and numpy library to the job



```
import pandas as pd
import numpy as np
```

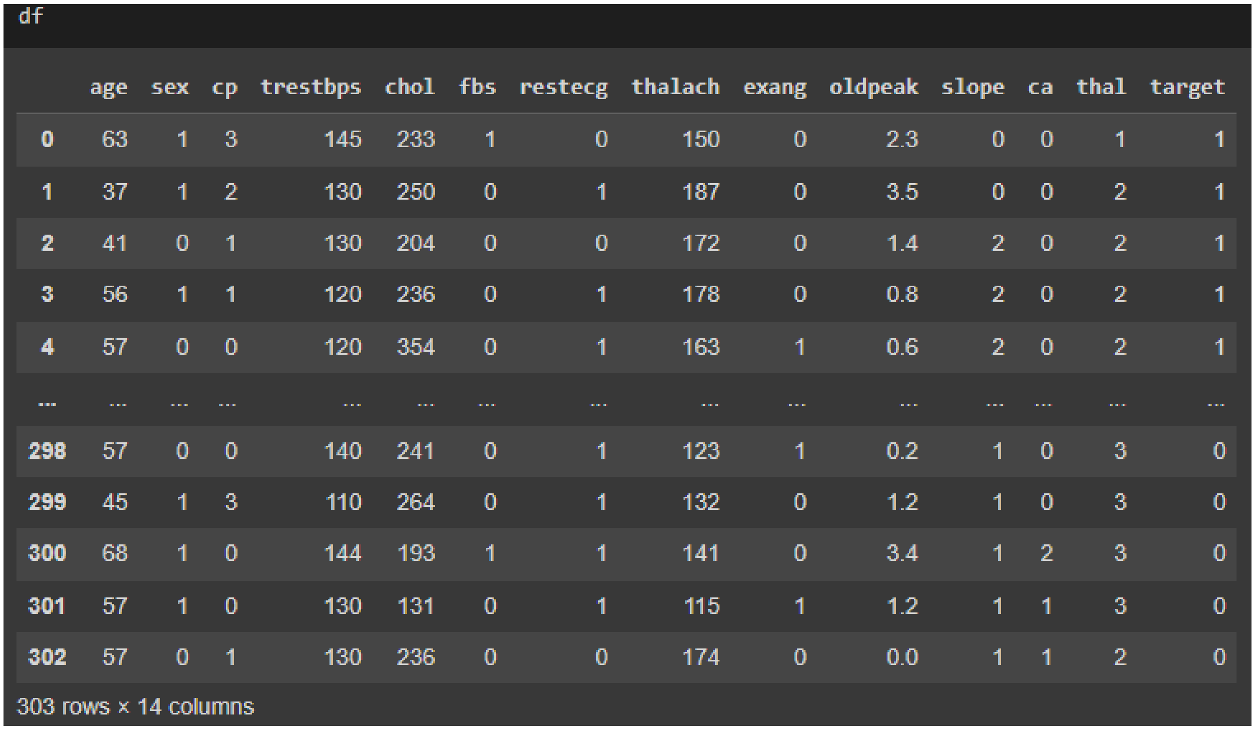
We can name it as df instead of writing the file path everytime.



```
df = pd.read_csv(file_path)
```

Viewing file data:

We can view the file data as a whole or partial part. To see all the data in the file we just need to write the file path name (df). This will show us all the data stored in the file.

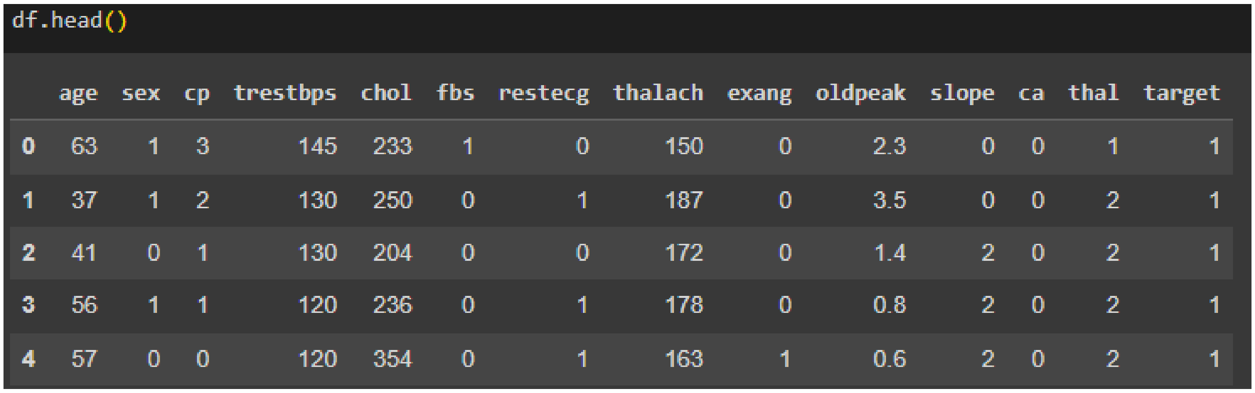


```
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

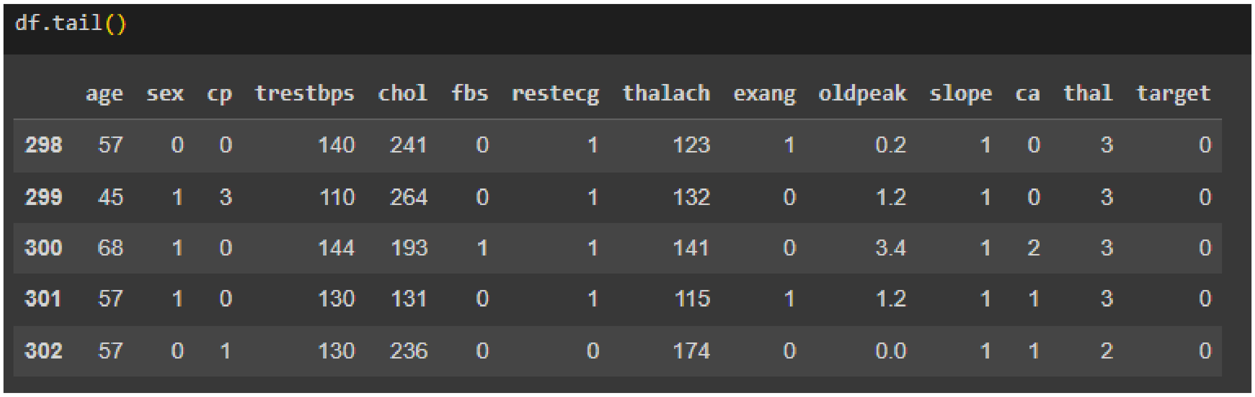
This can be viewed from first or last rows. The following command shows the first 5 rows:



```
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Similarly we can view the data from last 5 rows:

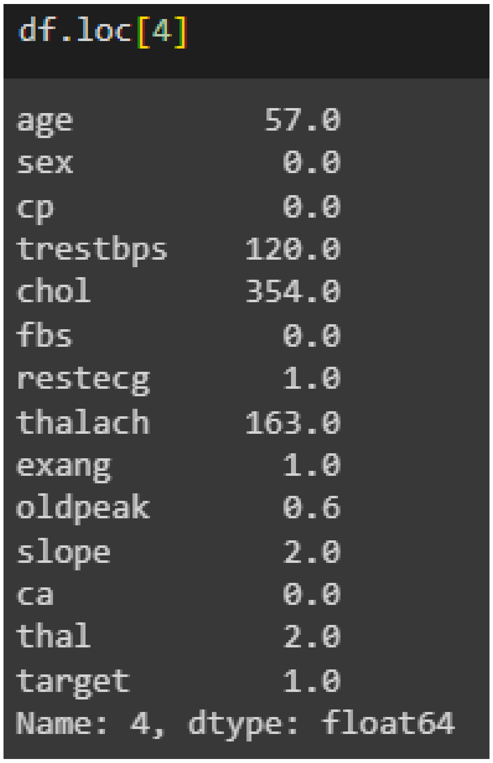


```
df.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

Specific Row Search:

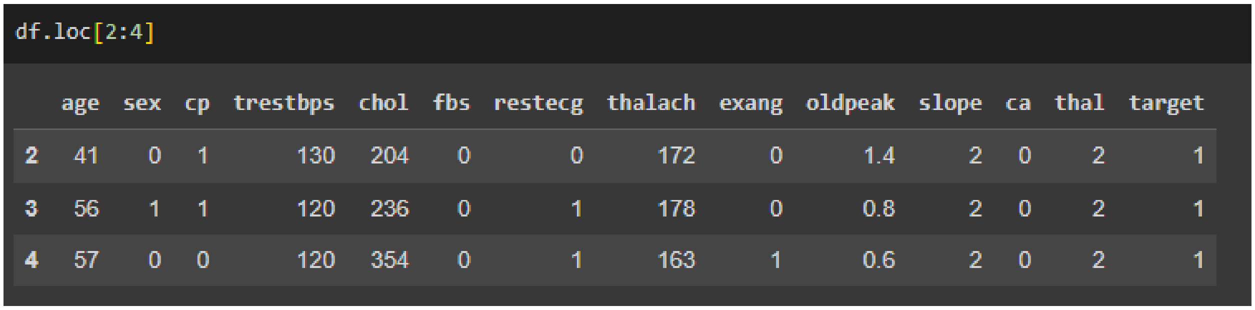
We can also search for data in specific location.



```
df.loc[4]
```

```
age          57.0
sex           0.0
cp            0.0
trestbps    120.0
chol        354.0
fbs           0.0
restecg       1.0
thalach     163.0
exang         1.0
oldpeak       0.6
slope         2.0
ca            0.0
thal          2.0
target        1.0
Name: 4, dtype: float64
```

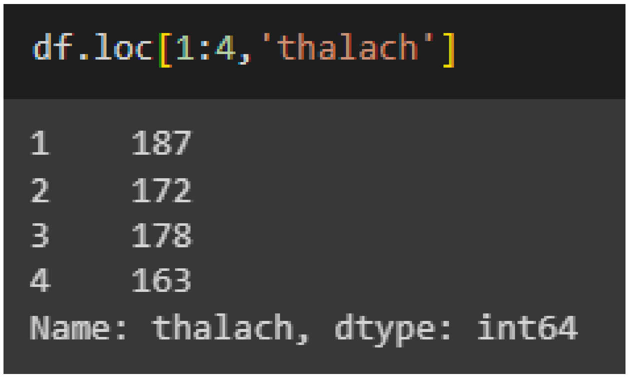
The data within a range can also be viewed using start_location : end_location format



```
df.loc[2:4]
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Adding a column name after range with a comma will show us only the column value of the given range. Let’s say we want to see the thalach column from row 1 to 4, here’s how to view it

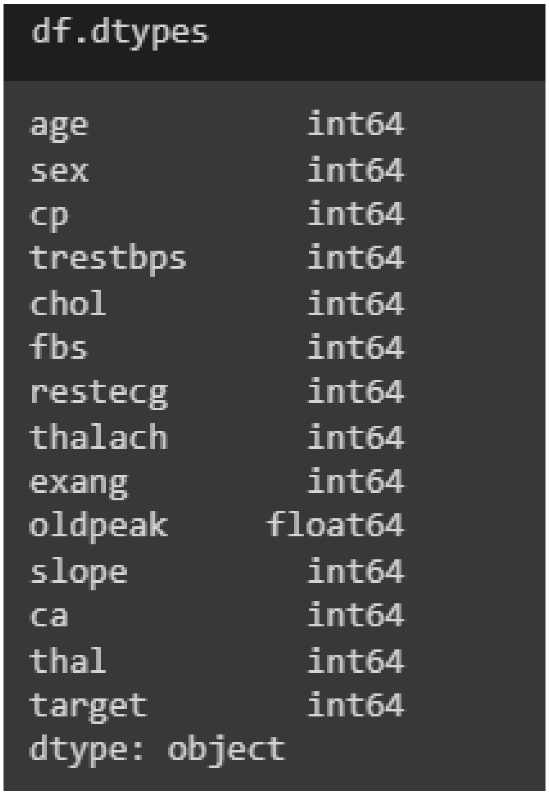


```
df.loc[1:4,'thalach']
```

```
1    187
2    172
3    178
4    163
Name: thalach, dtype: int64
```

Data information:

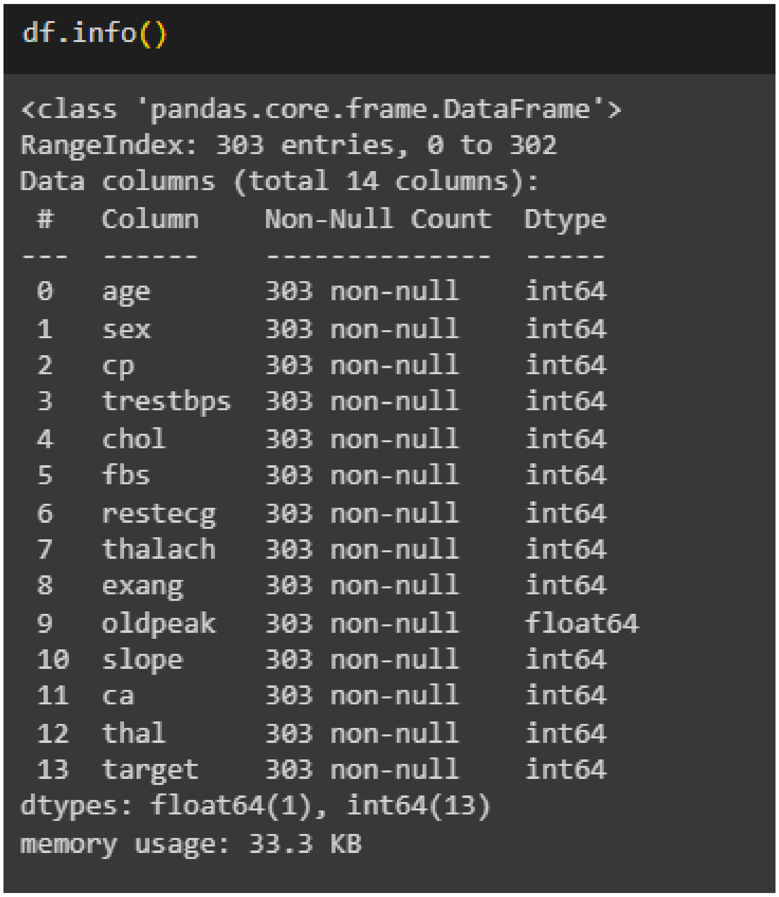
We can also check the data types of the columns:



```
df.dtypes
```

```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

If we want to get a complete information about the data in the file we can type name.info() to find it

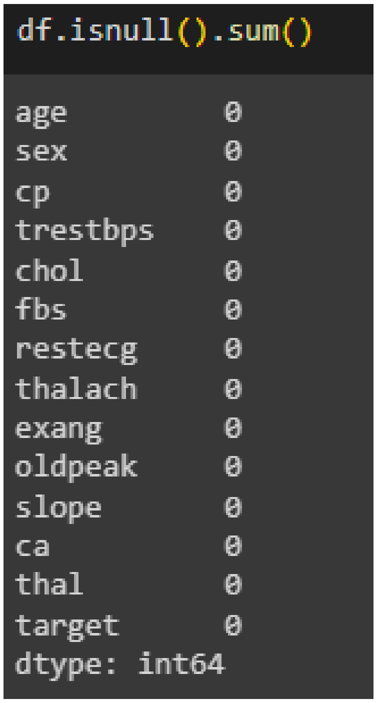


```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Counting operations:

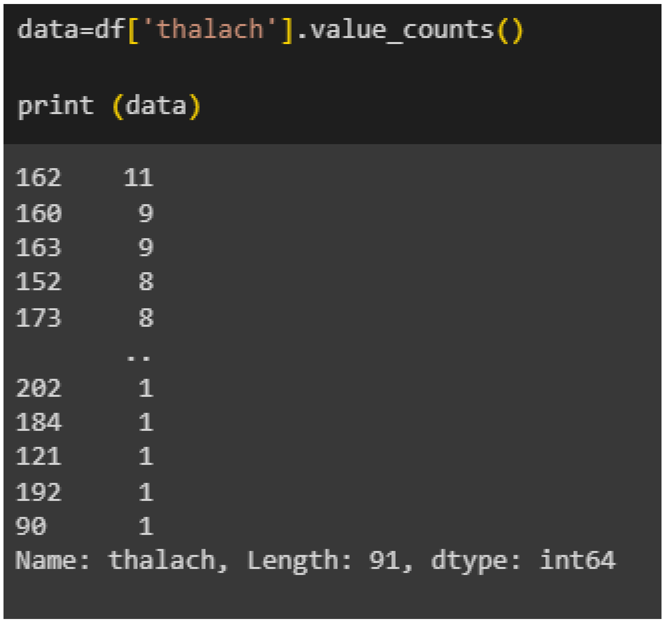
We can perform different counting operation on the given file. We can check the number of null values on each column



```
df.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

We can also count the appearance of a value in the file and store it in a variable. Then print the variable to show the result



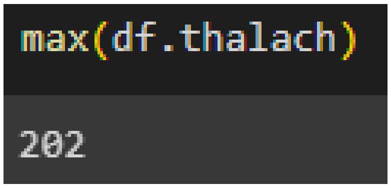
```
data=df['thalach'].value_counts()

print (data)
```

```
162    11
160     9
163     9
152     8
173     8
       ..
202     1
184     1
121     1
192     1
90      1
Name: thalach, Length: 91, dtype: int64
```

Minimum and maximum:

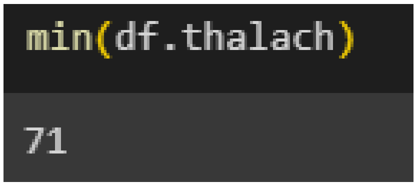
The max(file_name.column) returns the max value for the column



```
max(df.thalach)
```

```
202
```

Similarly we can find the minimum of the column

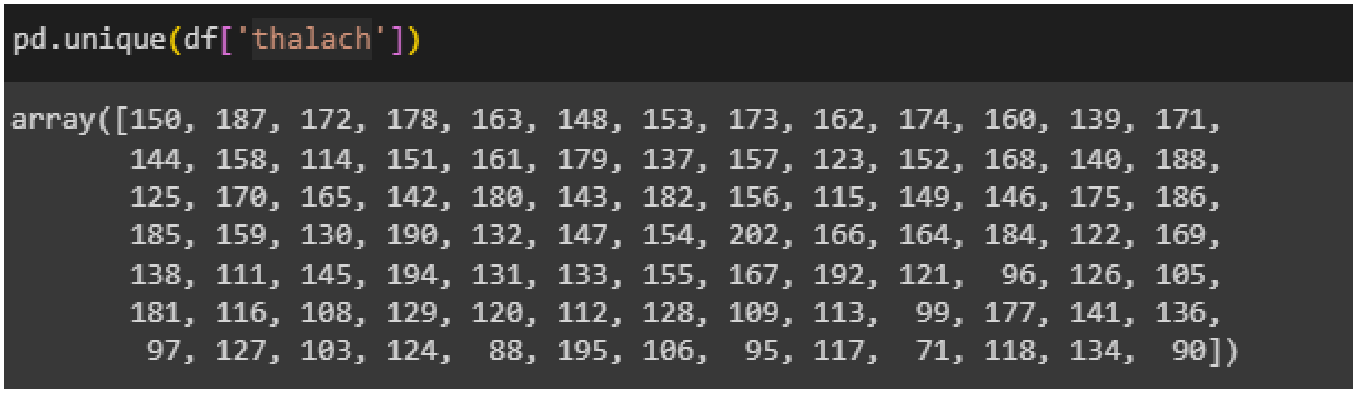


```
min(df.thalach)
```

```
71
```

Other operations:

We can find unique values on a column with the following code:



```
pd.unique(df['thalach'])
```

```
array([150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 171,
       144, 158, 114, 151, 161, 179, 137, 157, 123, 152, 168, 140, 188,
       125, 170, 165, 142, 180, 143, 182, 156, 115, 149, 146, 175, 186,
       185, 159, 130, 190, 132, 147, 154, 202, 166, 164, 184, 122, 169,
       138, 111, 145, 194, 131, 133, 155, 167, 192, 121,  96, 126, 105,
       181, 116, 108, 129, 120, 112, 128, 109, 113,  99, 177, 141, 136,
        97, 127, 103, 124,  88, 195, 106,  95, 117,  71, 118, 134,  90])
```