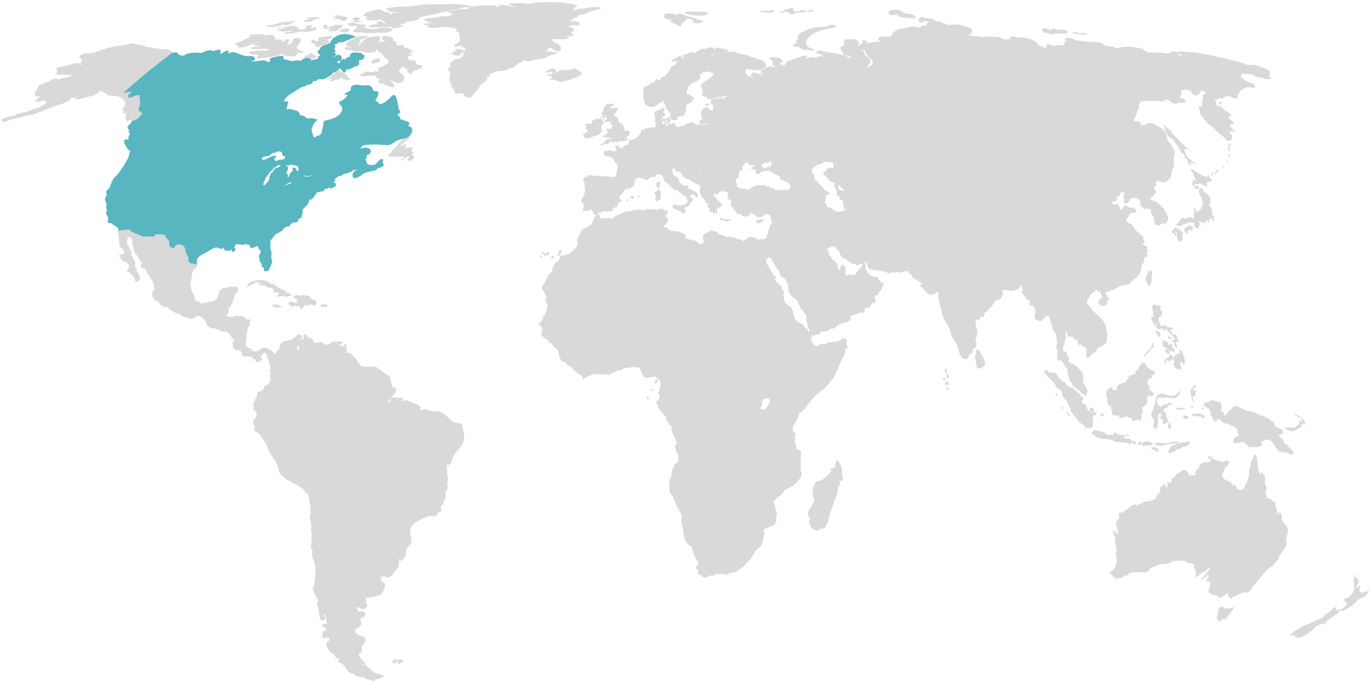
i0n Remedies

Team:
Amin Abueideh
Abed
Saleh Abbas
Fatima Twam
Kayan Abukhaizaran

Medication Non - Adherence

United States



🟦 2/3 of Non-Adherent Americans with Prescriptions

🟩 $300 BILLION Costs to the US Health Care System

🟦 125,000 Premature Deaths

⬜ 25% of Hospital Admissions

Consequences on Healthcare system

Medication non – adherence effects on healthcare system

Disease progression and complication

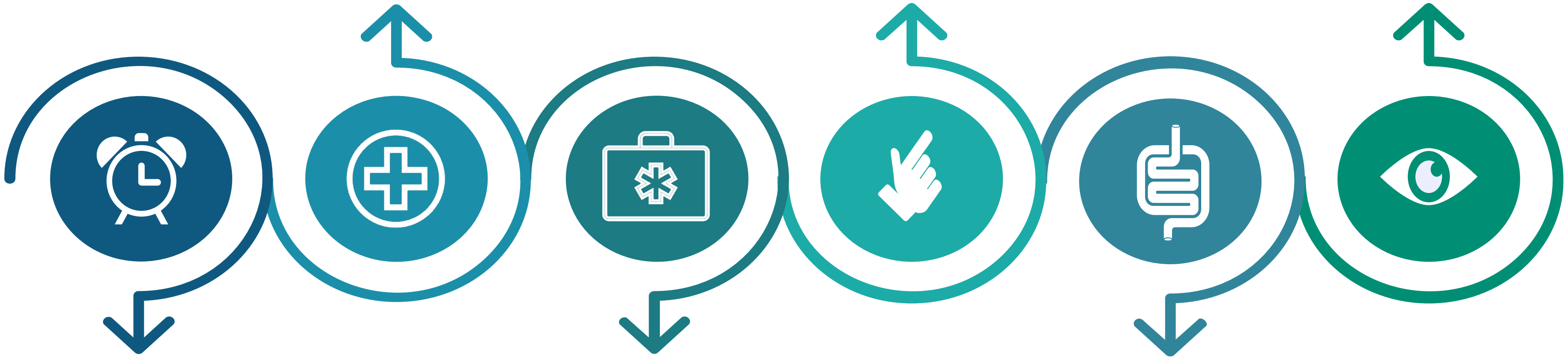
And decrease in quality of life

Increase in Healthcare Costs

Increase in Healthcare expenditure

Increase in admissions and ER visits

SOLUTIONS IN THE MARKET



Smart Prescription and Packaging

Wearable Sensors

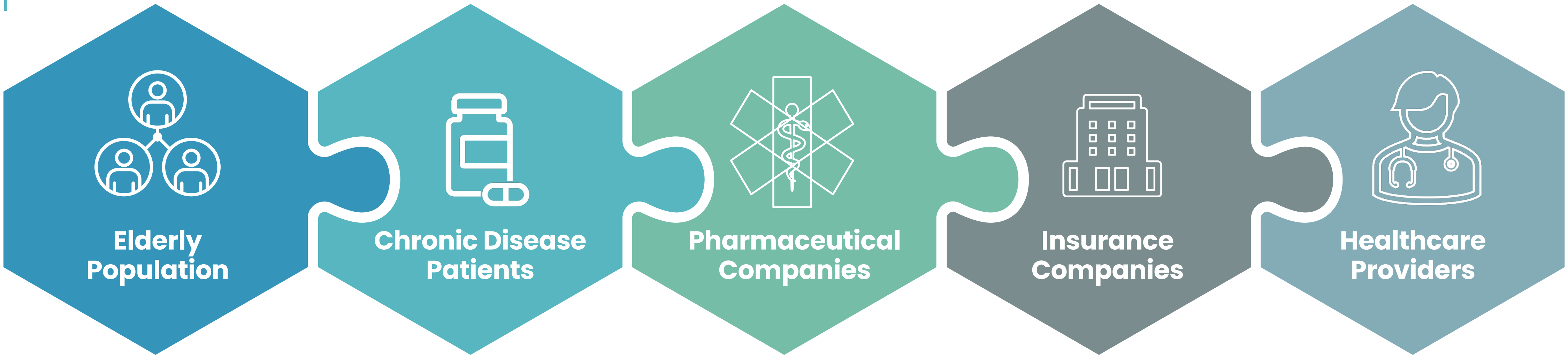
Vision Based Systems

Pharmaceutical Database & Automated Alerts

Smart Medication Cabinet

Ingestible Biosensors

TARGET MARKET



Elderly Population

Chronic Disease Patients

Pharmaceutical Companies

Insurance Companies

Healthcare Providers

Annual Health Expenditure

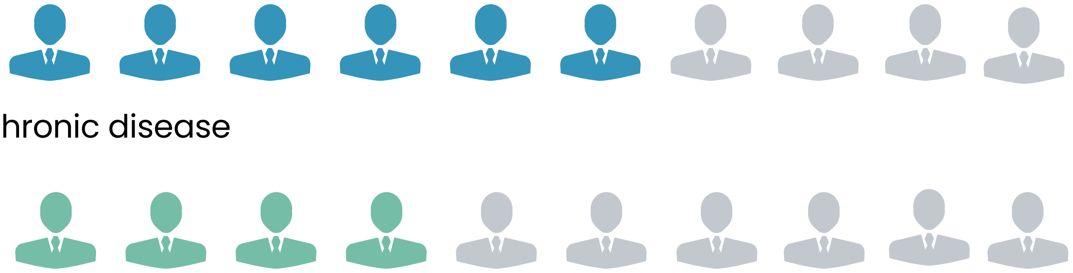
Spent on Chronic and Mental Health Diseases

● **6 in 10**

Adults have a chronic disease

● **4 in 10**

Adults have two or more chronic diseases



Our EDA Report

01

Variable Selection

Our domain expert team chose the needed variables

02

Ethical Approval was Obtained

From IAH Ethical Committee

03

EDA Pre-existing Data

04

Data Collection stage preperation

Feedback was sent from our team to hospital to ensure data quality

05

Link to EDA Report

https://drive.google.com/drive/folders/1XsJkP1InuYm_A5k5ERISvCM_qGGGRtGL?usp=sharing

Variables Selection

The data were obtained from the Health Information System (HIS) of Istishari Arab Hospital (IAH).

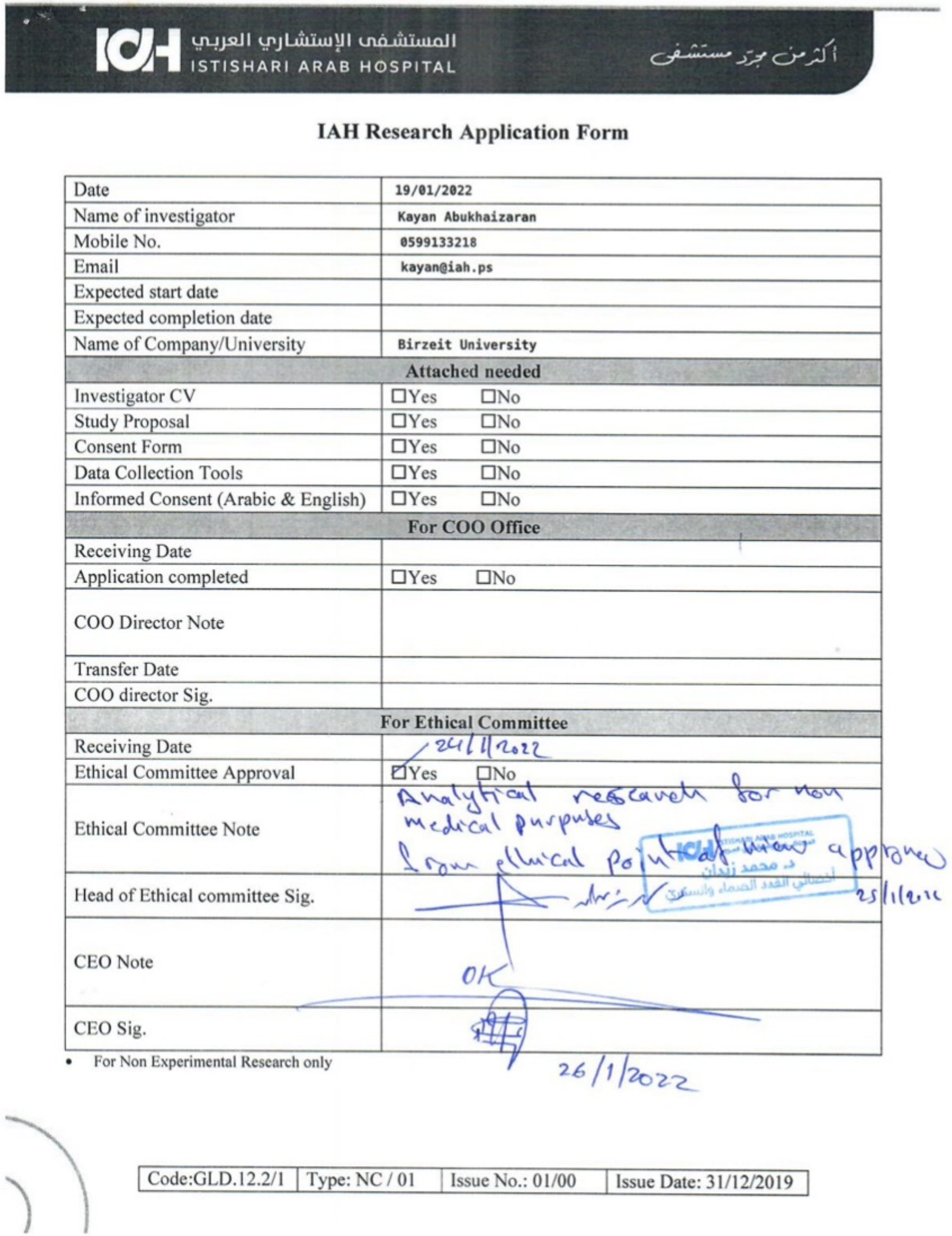
Diagnosis
Admission Date and time
Discharge Date and time
Gender
Age
Height
Weight
Past Medical History
Past Surgical History
Medication History
Smoking and Alcohol intake

Observations:
Blood Pressure,
Heart Rate,
Respiratory Rate ,
Temperature

Medication Orders

Lab Tests :
CBC, Liver Function Tests, Lipid Profile

LEGAL AND ETHICAL REQUIREMENTS



المستشفى الإستشاري العربي
ISTISHARI ARAB HOSPITAL

أكثر من مجرد مستشفى

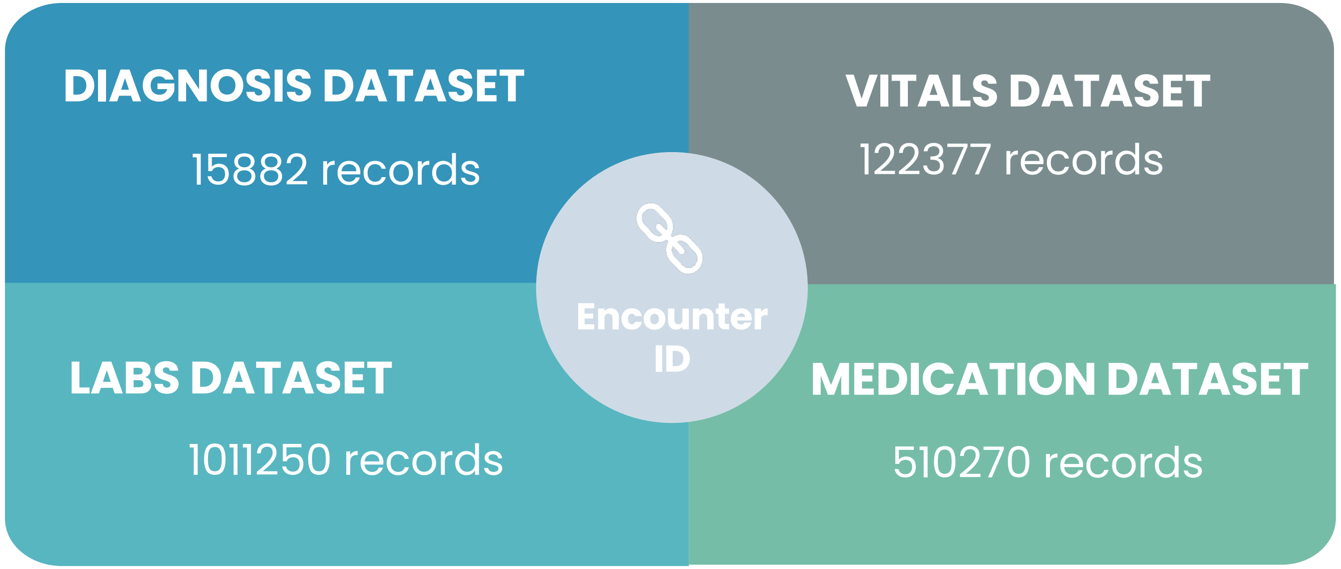
IAH Research Application Form

Date	19/01/2022
Name of investigator	Kayan Abukhaizaran
Mobile No.	0599133218
Email	kayan@iah.ps
Expected start date	
Expected completion date	
Name of Company/University	Birzeit University
Attached needed	
Investigator CV	□Yes □No
Study Proposal	□Yes □No
Consent Form	□Yes □No
Data Collection Tools	□Yes □No
Informed Consent (Arabic & English)	□Yes □No
For COO Office	
Receiving Date	
Application completed	□Yes □No
COO Director Note	
Transfer Date	
COO director Sig.	
For Ethical Committee	
Receiving Date	24/1/2022
Ethical Committee Approval	☑Yes □No
Ethical Committee Note	Analytical research for non medical purposes from ethical point of view approved
Head of Ethical committee Sig.	ISTISHARI ARAB HOSPITAL د. محمد زيدان أخصائي الغدد الصماء والسكري 25/1/2022
CEO Note	OK
CEO Sig.	26/1/2022

- For Non Experimental Research only

Code:GLD.12.2/1	Type: NC / 01	Issue No.: 01/00	Issue Date: 31/12/2019

OUR DATASETS



DIAGNOSIS DATASET

15882 records

VITALS DATASET

122377 records

Encounter ID

LABS DATASET

1011250 records

MEDICATION DATASET

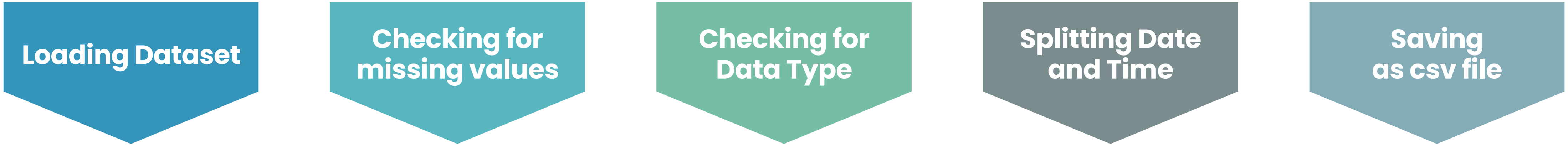
510270 records

Data collection phase:

IAH created a query so that the data will be provided to our team on a monthly basis.

Exploratory Data Analysis

The below mentioned steps were conducted on all the datasets.



Loading Dataset

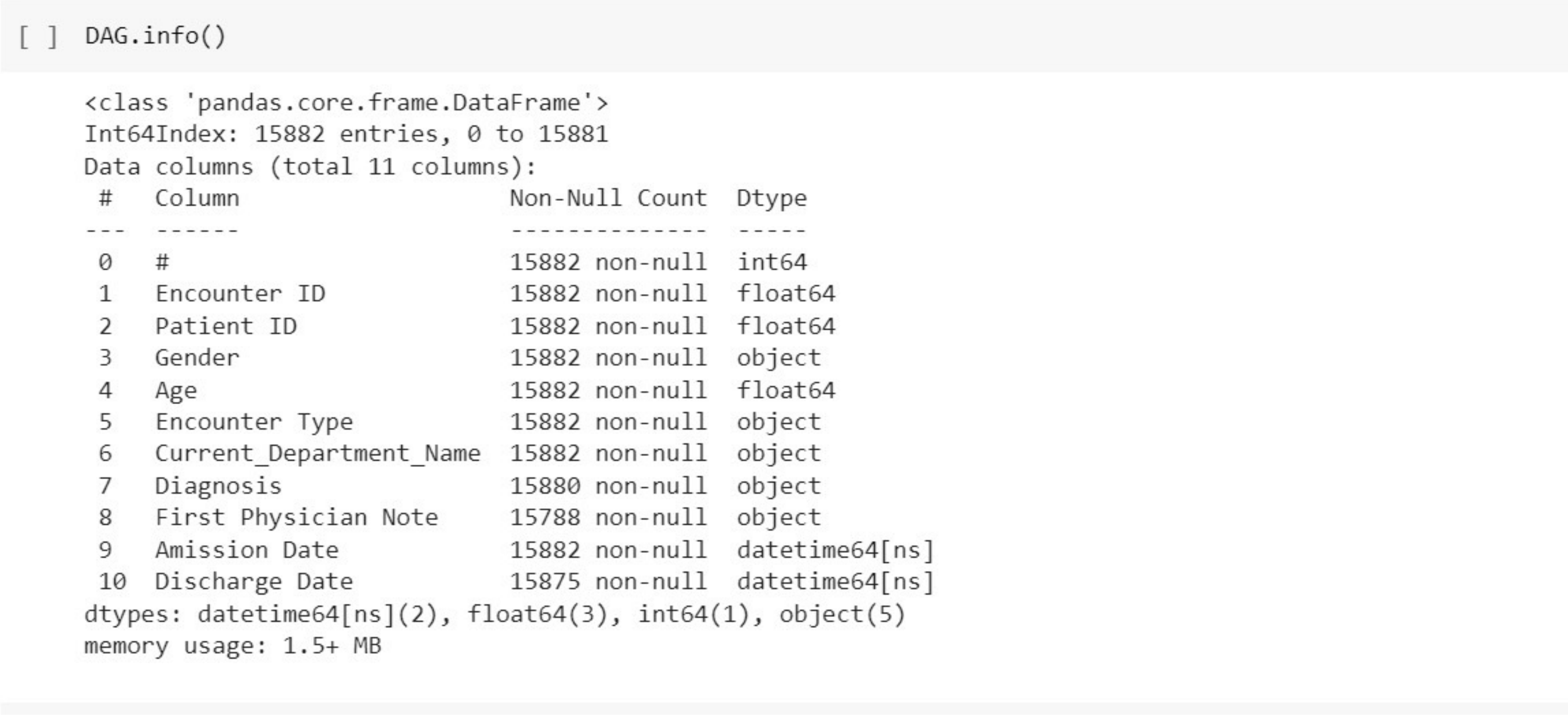
Checking for missing values

Checking for Data Type

Splitting Date and Time

Saving as csv file

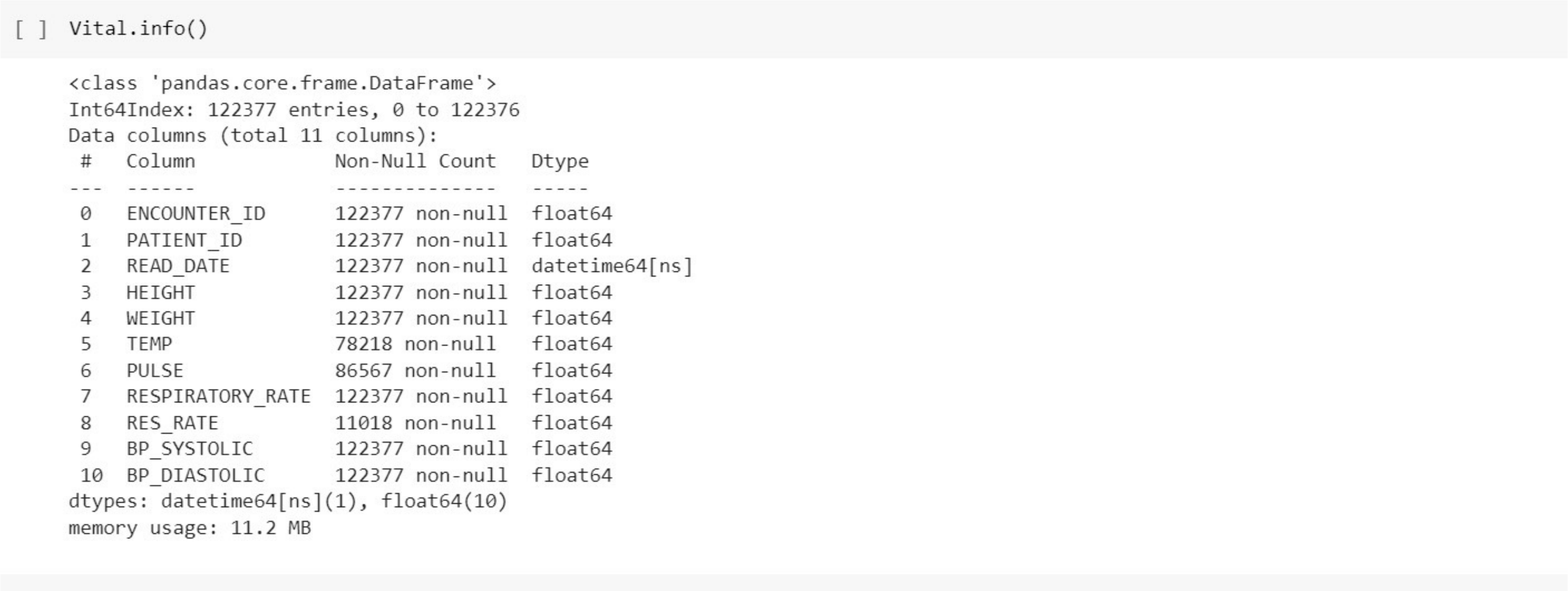
Checking for Data Type



```
[ ]  DAG.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15882 entries, 0 to 15881
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   #                        15882 non-null  int64
 1   Encounter ID             15882 non-null  float64
 2   Patient ID               15882 non-null  float64
 3   Gender                   15882 non-null  object
 4   Age                      15882 non-null  float64
 5   Encounter Type           15882 non-null  object
 6   Current_Department_Name  15882 non-null  object
 7   Diagnosis                15880 non-null  object
 8   First Physician Note     15788 non-null  object
 9   Amission Date            15882 non-null  datetime64[ns]
 10  Discharge Date           15875 non-null  datetime64[ns]
dtypes: datetime64[ns](2), float64(3), int64(1), object(5)
memory usage: 1.5+ MB
```

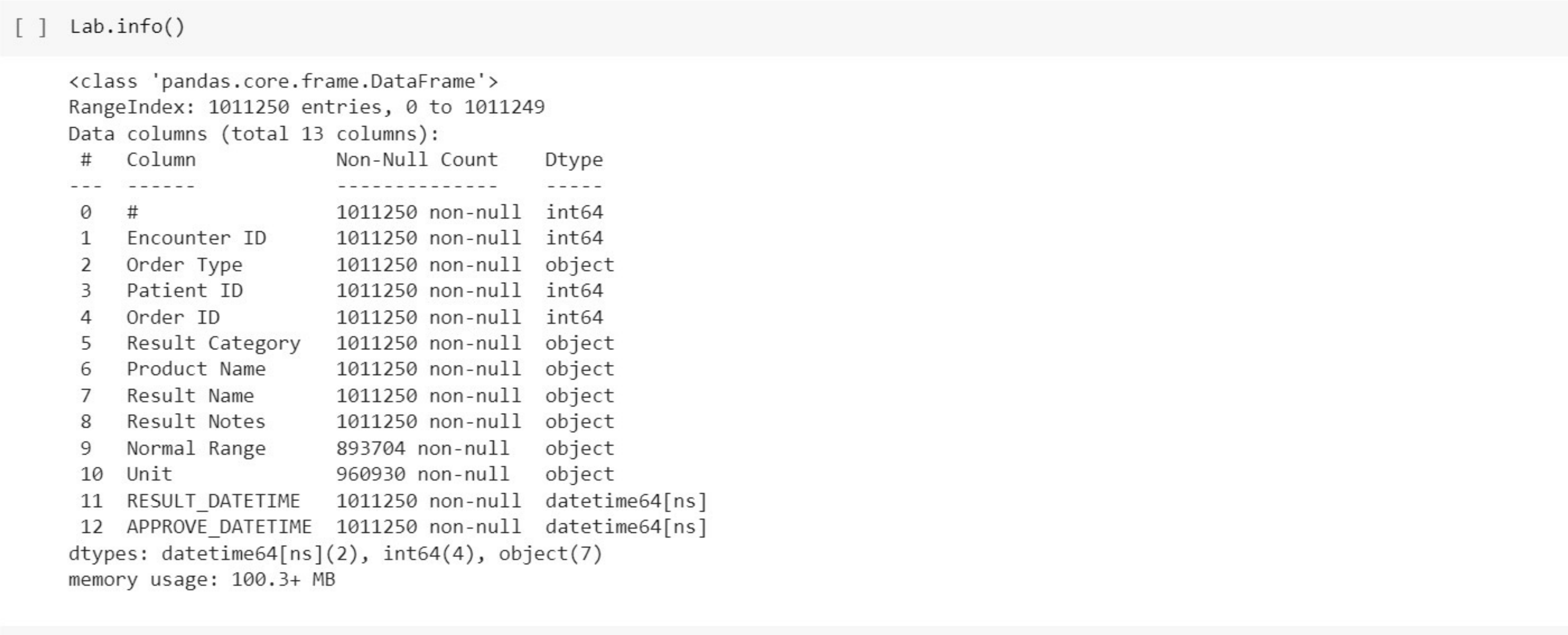
Checking for Data Type



```
[ ]  Vital.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 122377 entries, 0 to 122376
Data columns (total 11 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   ENCOUNTER_ID      122377 non-null  float64
 1   PATIENT_ID        122377 non-null  float64
 2   READ_DATE         122377 non-null  datetime64[ns]
 3   HEIGHT            122377 non-null  float64
 4   WEIGHT            122377 non-null  float64
 5   TEMP              78218 non-null   float64
 6   PULSE             86567 non-null   float64
 7   RESPIRATORY_RATE  122377 non-null  float64
 8   RES_RATE          11018 non-null   float64
 9   BP_SYSTOLIC       122377 non-null  float64
 10  BP_DIASTOLIC      122377 non-null  float64
dtypes: datetime64[ns](1), float64(10)
memory usage: 11.2 MB
```

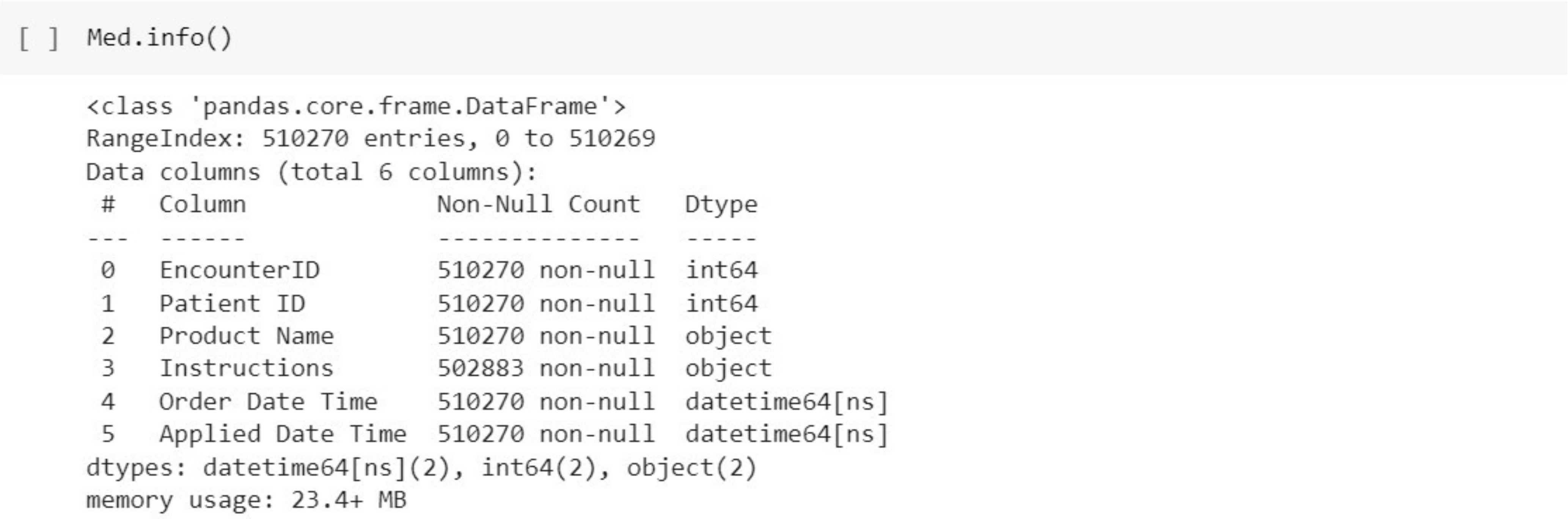
Checking for Data Type



```
[ ]  Lab.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1011250 entries, 0 to 1011249
Data columns (total 13 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   #                  1011250 non-null  int64
 1   Encounter ID       1011250 non-null  int64
 2   Order Type         1011250 non-null  object
 3   Patient ID         1011250 non-null  int64
 4   Order ID           1011250 non-null  int64
 5   Result Category    1011250 non-null  object
 6   Product Name       1011250 non-null  object
 7   Result Name        1011250 non-null  object
 8   Result Notes       1011250 non-null  object
 9   Normal Range       893704 non-null   object
 10  Unit               960930 non-null   object
 11  RESULT_DATETIME    1011250 non-null  datetime64[ns]
 12  APPROVE_DATETIME   1011250 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(4), object(7)
memory usage: 100.3+ MB
```

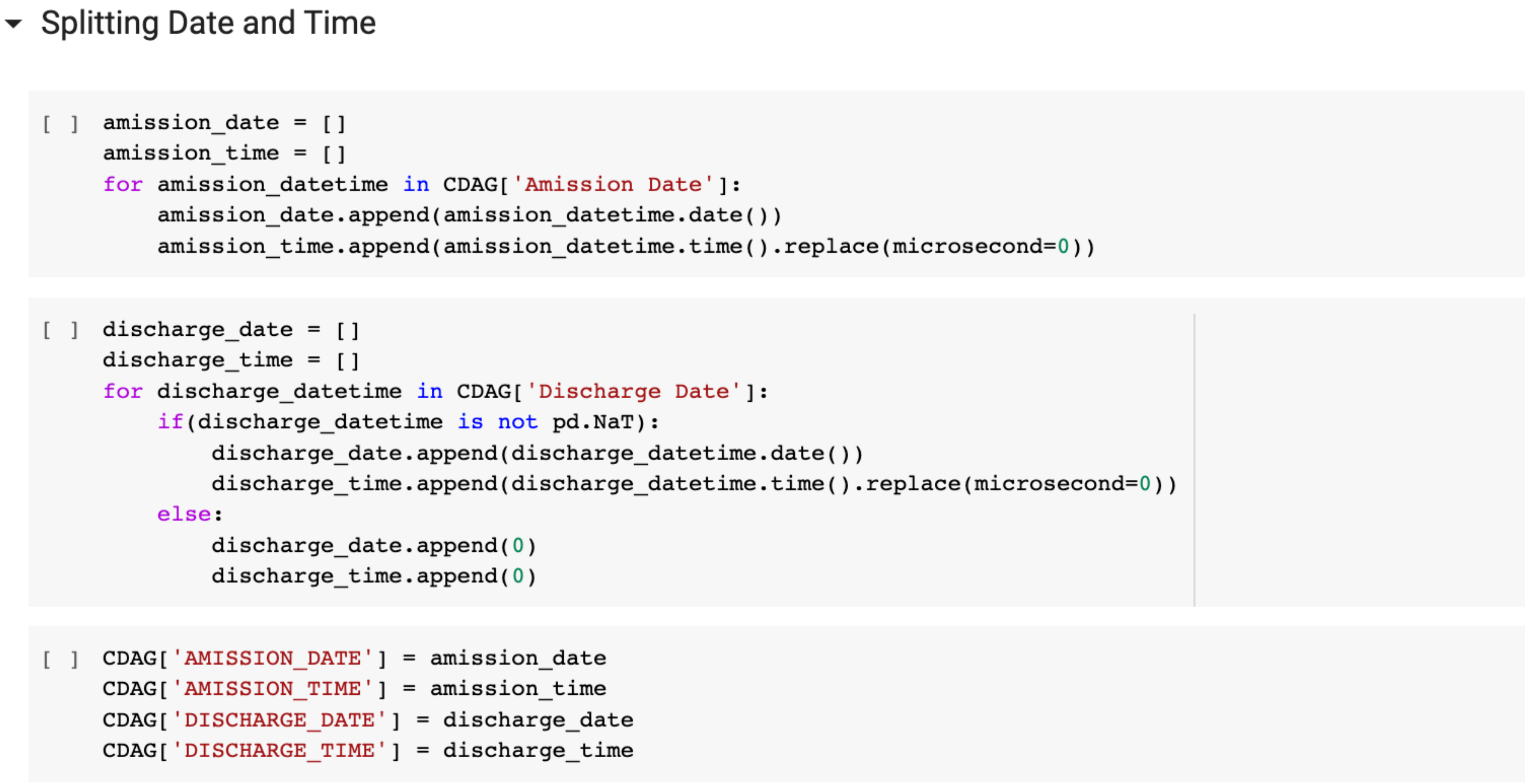
Checking for Data Type



```
[ ]  Med.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 510270 entries, 0 to 510269
Data columns (total 6 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   EncounterID        510270 non-null  int64
 1   Patient ID         510270 non-null  int64
 2   Product Name       510270 non-null  object
 3   Instructions       502883 non-null  object
 4   Order Date Time    510270 non-null  datetime64[ns]
 5   Applied Date Time  510270 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(2), object(2)
memory usage: 23.4+ MB
```

Splitting Date and Time



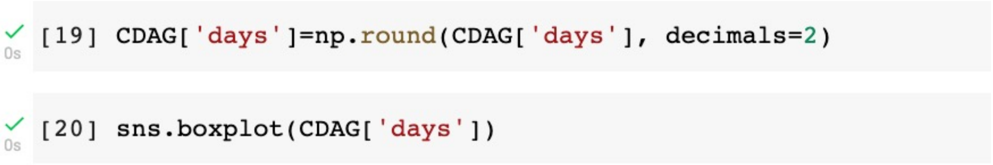
Splitting Date and Time

```
amission_date = []
amission_time = []
for amission_datetime in CDAG['Amission Date']:
    amission_date.append(amission_datetime.date())
    amission_time.append(amission_datetime.time().replace(microsecond=0))
```

```
discharge_date = []
discharge_time = []
for discharge_datetime in CDAG['Discharge Date']:
    if(discharge_datetime is not pd.NaT):
        discharge_date.append(discharge_datetime.date())
        discharge_time.append(discharge_datetime.time().replace(microsecond=0))
    else:
        discharge_date.append(0)
        discharge_time.append(0)
```

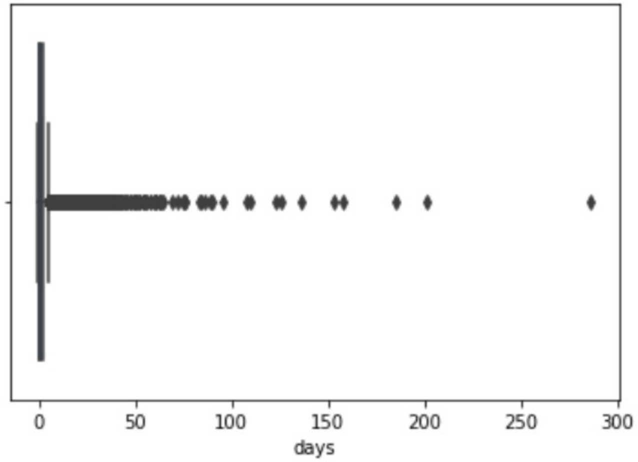
```
CDAG['AMISSION_DATE'] = amission_date
CDAG['AMISSION_TIME'] = amission_time
CDAG['DISCHARGE_DATE'] = discharge_date
CDAG['DISCHARGE_TIME'] = discharge_time
```

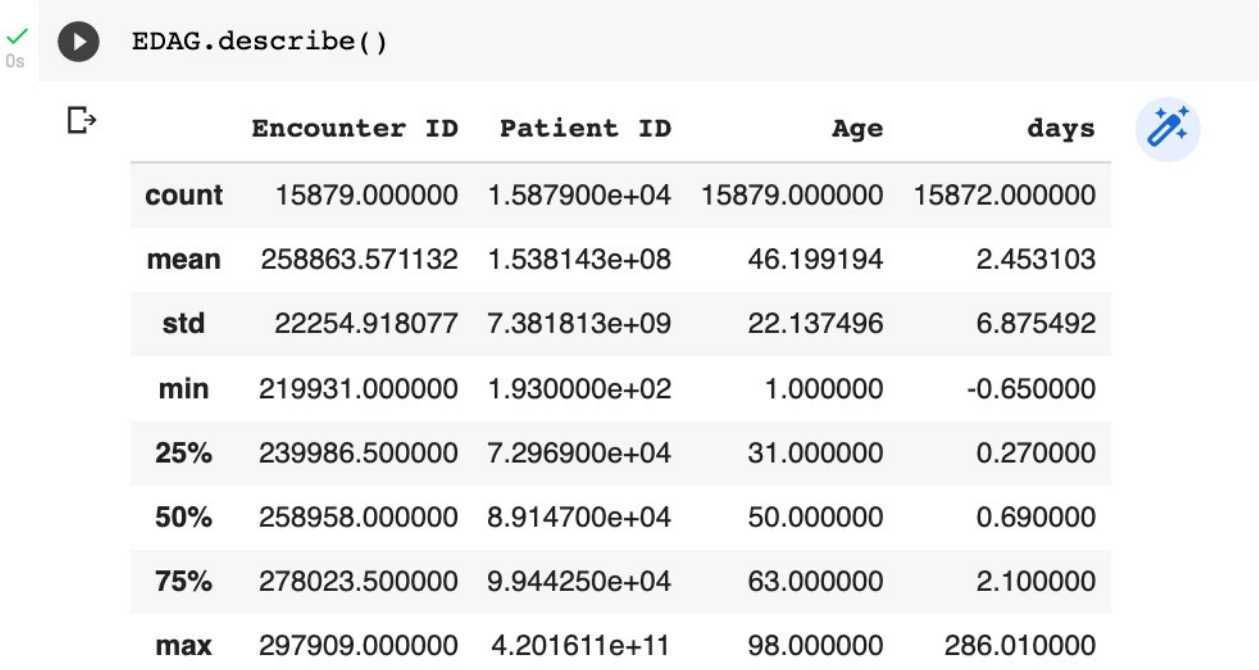
Calculating Lengths of Stay (LOS)



```
[19] CDAG['days']=np.round(CDAG['days'], decimals=2)
```

```
[20] sns.boxplot(CDAG['days'])
```

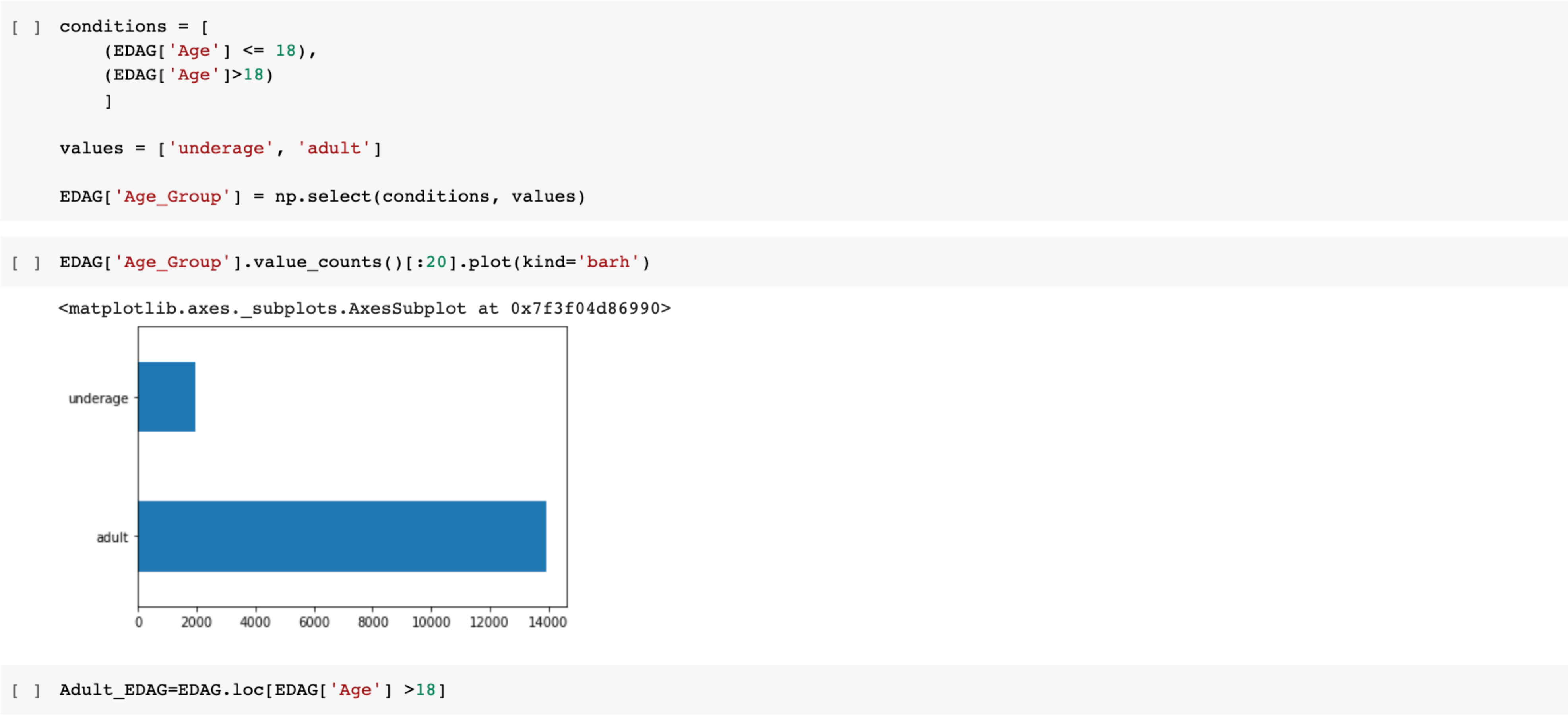




```
EDAG.describe()
```

	Encounter ID	Patient ID	Age	days
count	15879.000000	1.587900e+04	15879.000000	15872.000000
mean	258863.571132	1.538143e+08	46.199194	2.453103
std	22254.918077	7.381813e+09	22.137496	6.875492
min	219931.000000	1.930000e+02	1.000000	-0.650000
25%	239986.500000	7.296900e+04	31.000000	0.270000
50%	258958.000000	8.914700e+04	50.000000	0.690000
75%	278023.500000	9.944250e+04	63.000000	2.100000
max	297909.000000	4.201611e+11	98.000000	286.010000

Categorizing Age Groups



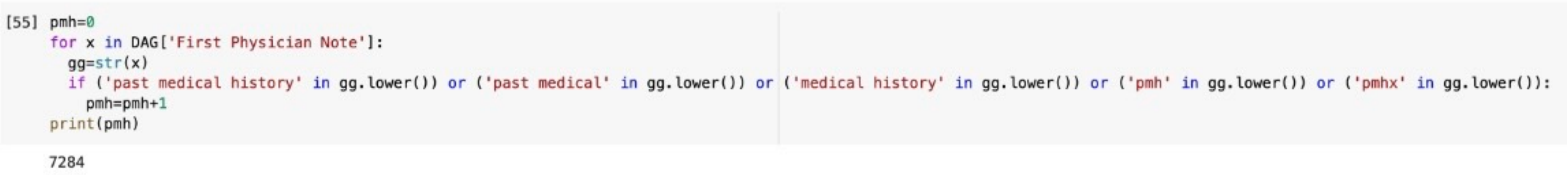
```
[ ] conditions = [
        (EDAG['Age'] <= 18),
        (EDAG['Age']>18)
        ]

    values = ['underage', 'adult']

    EDAG['Age_Group'] = np.select(conditions, values)
```

```
[ ] EDAG['Age_Group'].value_counts()[:20].plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f04d86990>
```

```
[ ] Adult_EDAG=EDAG.loc[EDAG['Age'] >18]
```

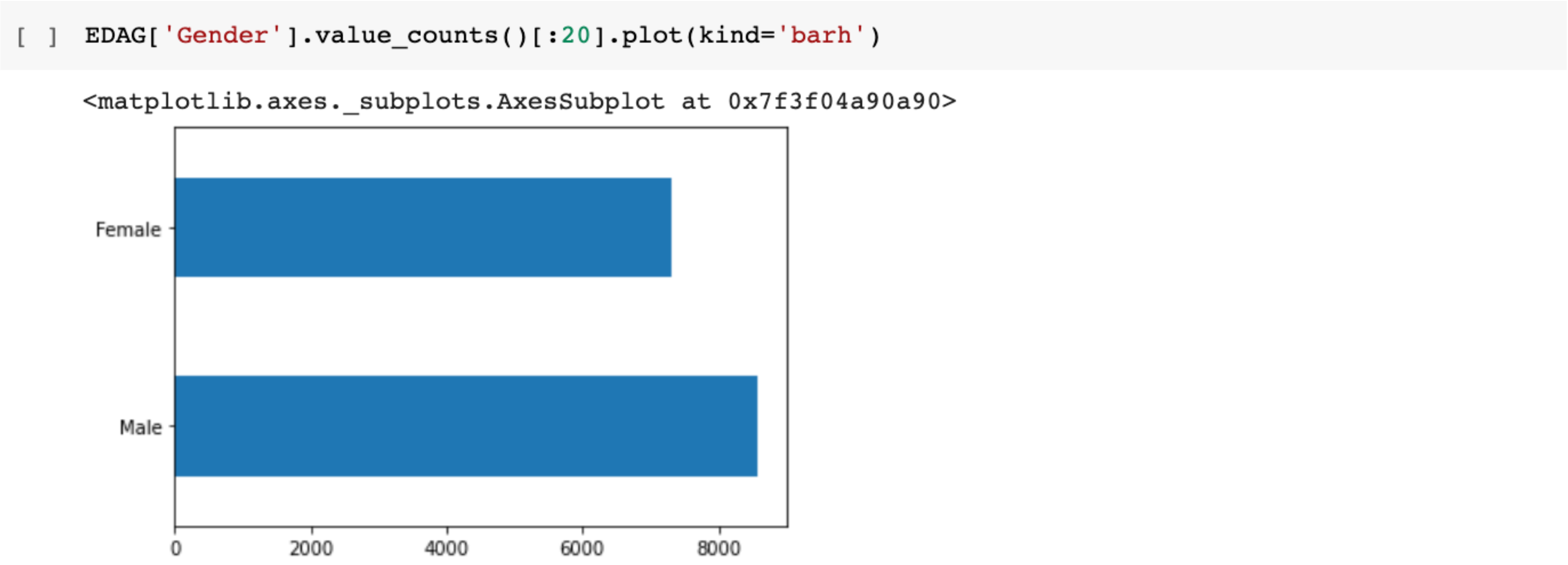
Data Extraction from “First Physician Note”



```
[55] pmh=0
     for x in DAG['First Physician Note']:
       gg=str(x)
       if ('past medical history' in gg.lower()) or ('past medical' in gg.lower()) or ('medical history' in gg.lower()) or ('pmh' in gg.lower()) or ('pmhx' in gg.lower()):
         pmh=pmh+1
     print(pmh)

     7284
```

Data collection phase:
Our requested variables will be obtained from new screens currently in the testing phase

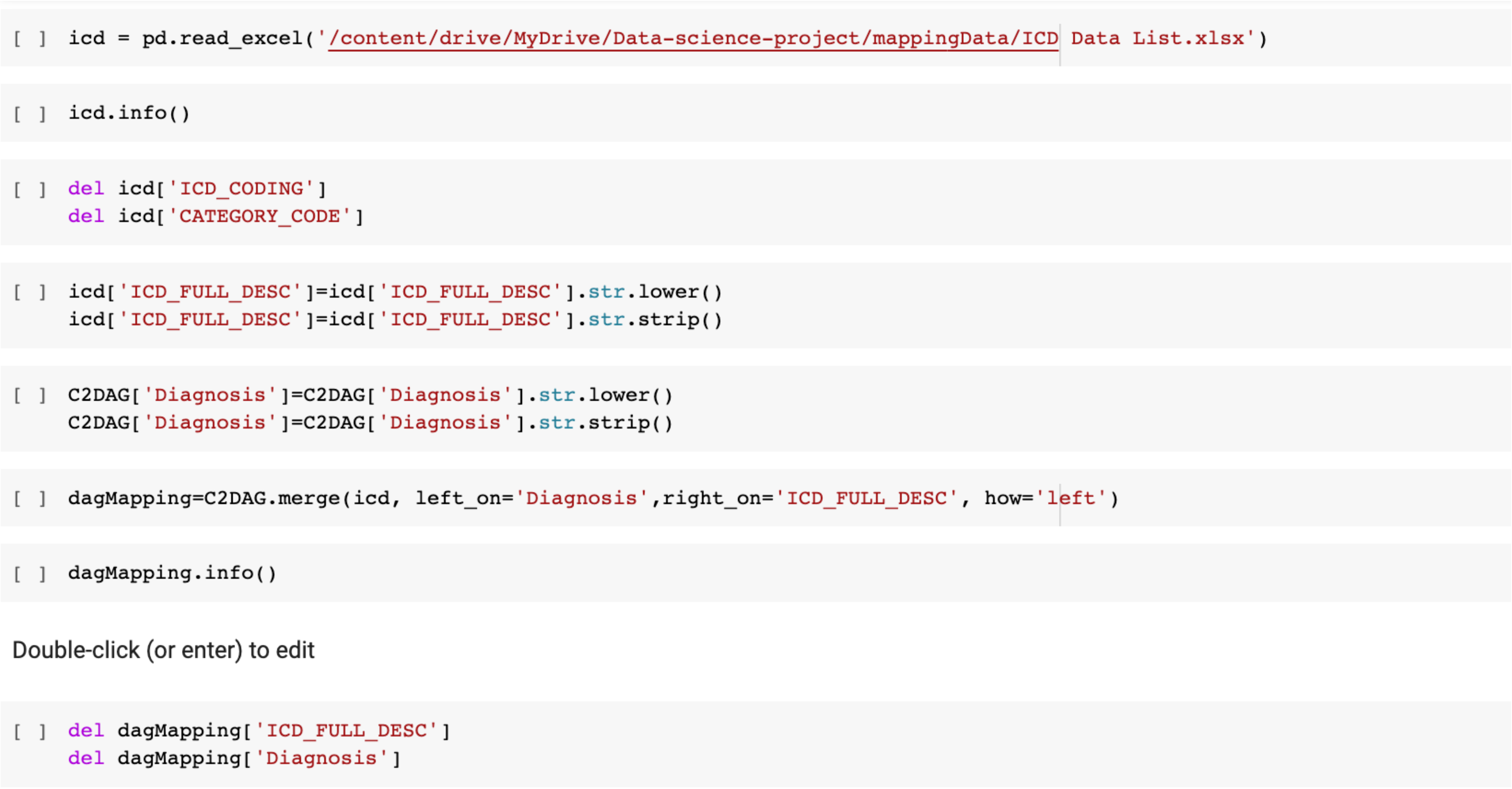
Checking for Gender values



```
[ ]  EDAG['Gender'].value_counts()[:20].plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f04a90a90>
```

Mapping “Diagnosis” with ICD-10



```
[ ] icd = pd.read_excel('/content/drive/MyDrive/Data-science-project/mappingData/ICD Data List.xlsx')
```

```
[ ] icd.info()
```

```
[ ] del icd['ICD_CODING']
    del icd['CATEGORY_CODE']
```

```
[ ] icd['ICD_FULL_DESC']=icd['ICD_FULL_DESC'].str.lower()
    icd['ICD_FULL_DESC']=icd['ICD_FULL_DESC'].str.strip()
```

```
[ ] C2DAG['Diagnosis']=C2DAG['Diagnosis'].str.lower()
    C2DAG['Diagnosis']=C2DAG['Diagnosis'].str.strip()
```

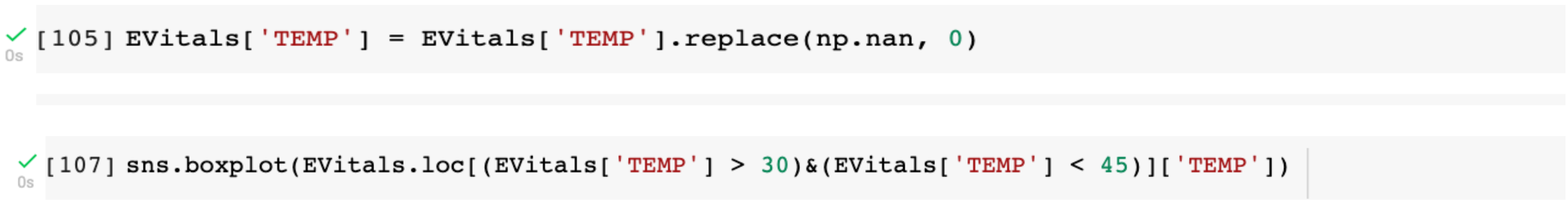
```
[ ] dagMapping=C2DAG.merge(icd, left_on='Diagnosis',right_on='ICD_FULL_DESC', how='left')
```

```
[ ] dagMapping.info()
```

Double-click (or enter) to edit

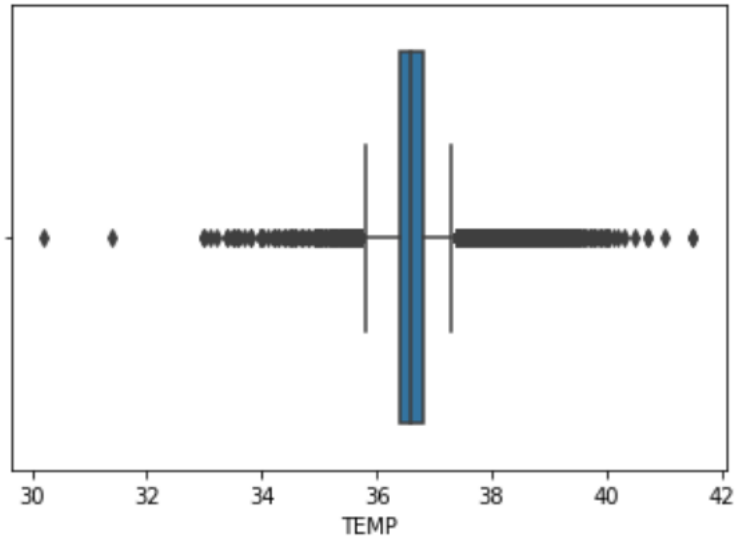
```
[ ] del dagMapping['ICD_FULL_DESC']
    del dagMapping['Diagnosis']
```

Validating the “Temp”



```
✓ [105] EVitals['TEMP'] = EVitals['TEMP'].replace(np.nan, 0)
0s
```

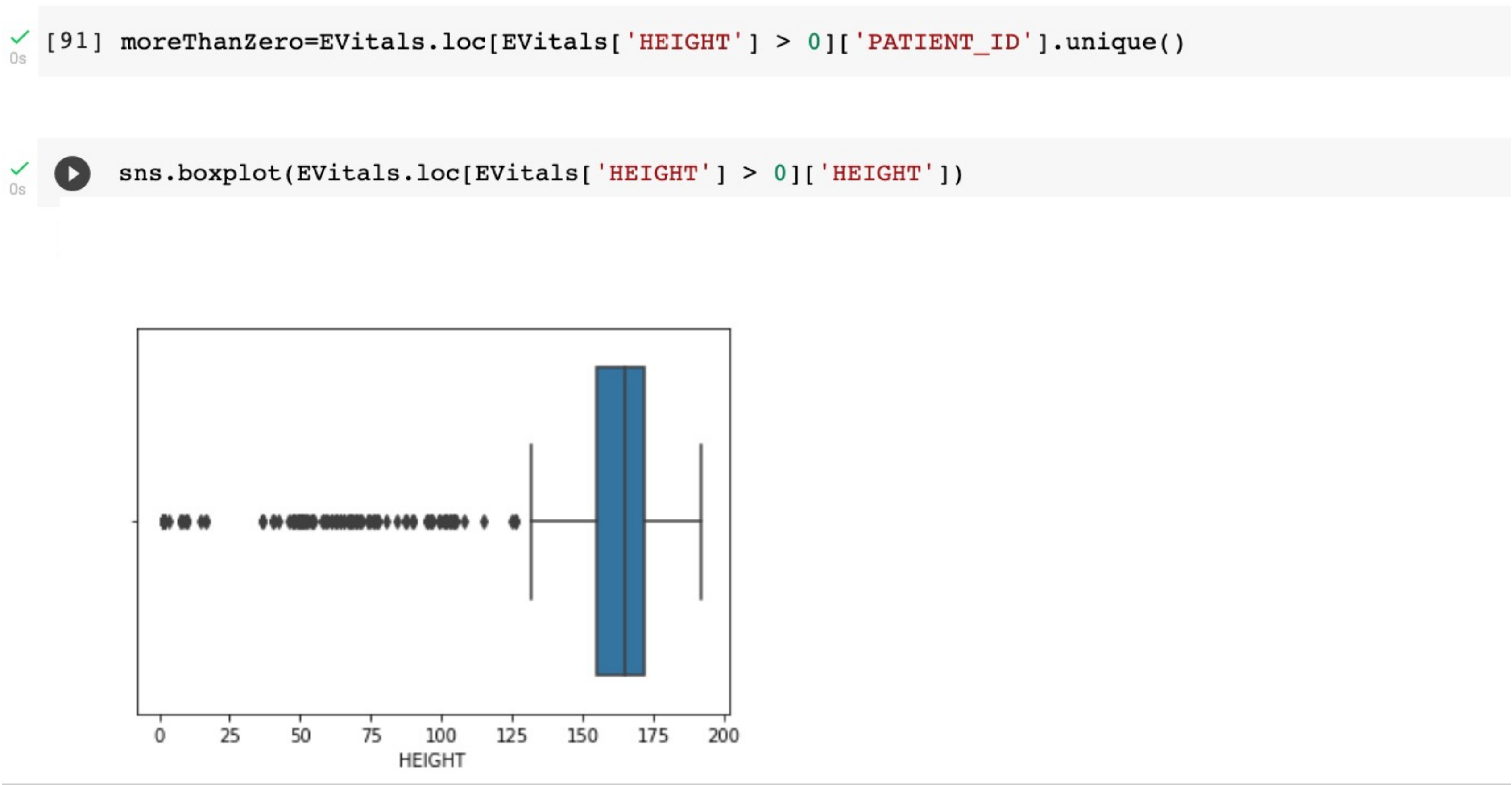
```
✓ [107] sns.boxplot(EVitals.loc[(EVitals['TEMP'] > 30)&(EVitals['TEMP'] < 45)]['TEMP'])
0s
```



Data collection phase:

To overcome the outliers and fault values due to human errors. We provided the IAH with validation tools to be applied on the observation fields.

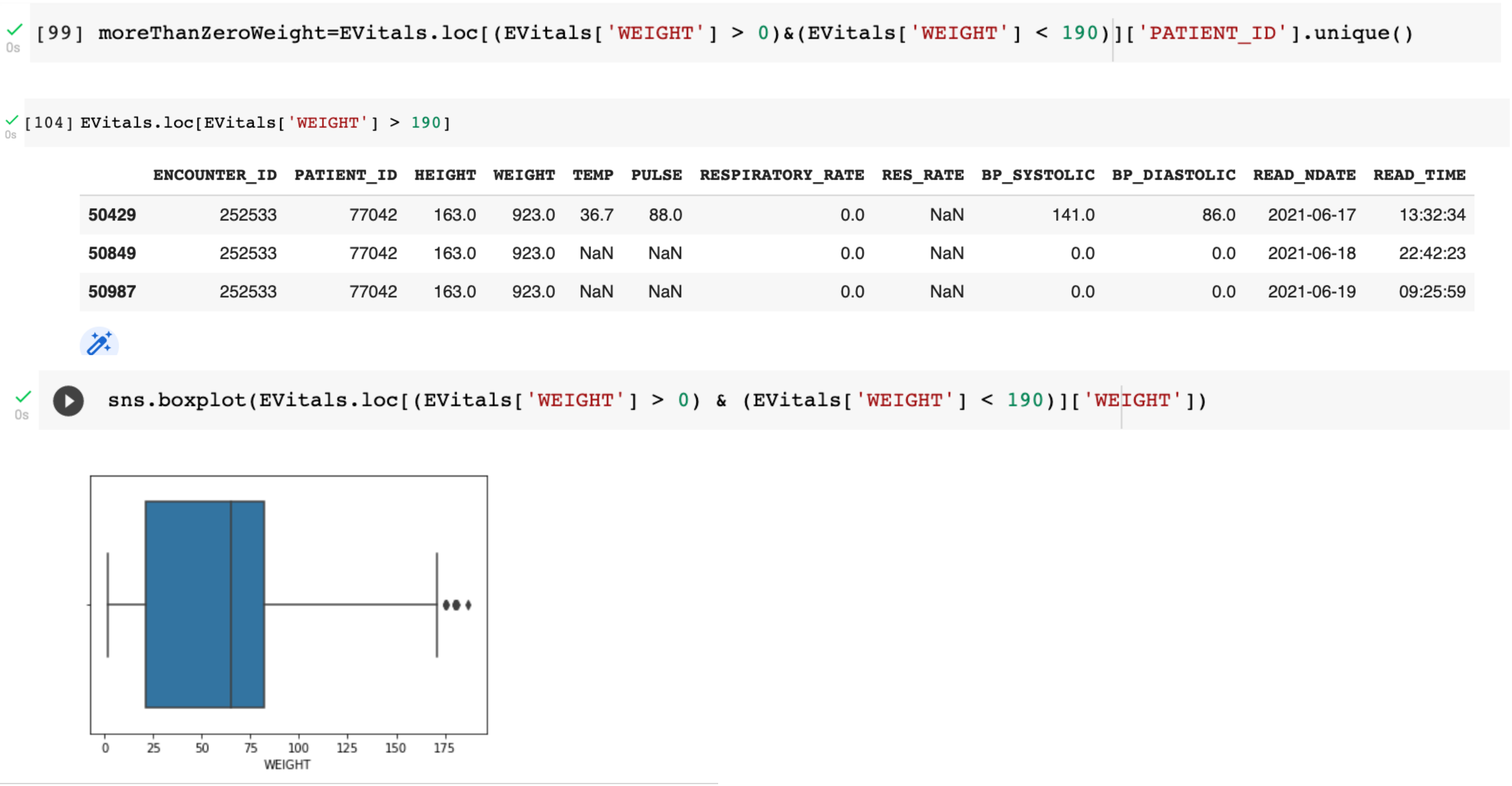
Validating the “Height”



```
[91] moreThanZero=EVitals.loc[EVitals['HEIGHT'] > 0]['PATIENT_ID'].unique()
```

```
sns.boxplot(EVitals.loc[EVitals['HEIGHT'] > 0]['HEIGHT'])
```

Validating the “Weight”



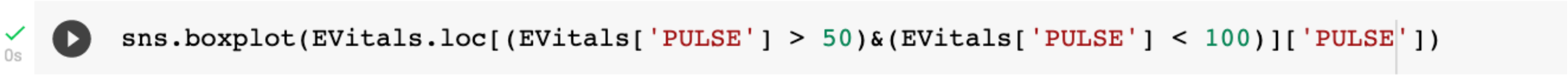
```
[99] moreThanZeroWeight=EVitals.loc[(EVitals['WEIGHT'] > 0)&(EVitals['WEIGHT'] < 190)]['PATIENT_ID'].unique()
```

```
[104] EVitals.loc[EVitals['WEIGHT'] > 190]
```

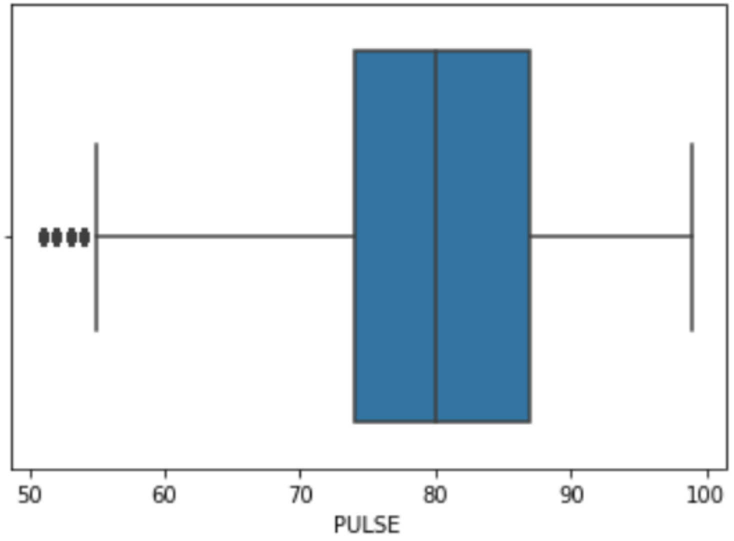
	ENCOUNTER_ID	PATIENT_ID	HEIGHT	WEIGHT	TEMP	PULSE	RESPIRATORY_RATE	RES_RATE	BP_SYSTOLIC	BP_DIASTOLIC	READ_NDATE	READ_TIME
50429	252533	77042	163.0	923.0	36.7	88.0	0.0	NaN	141.0	86.0	2021-06-17	13:32:34
50849	252533	77042	163.0	923.0	NaN	NaN	0.0	NaN	0.0	0.0	2021-06-18	22:42:23
50987	252533	77042	163.0	923.0	NaN	NaN	0.0	NaN	0.0	0.0	2021-06-19	09:25:59

```
sns.boxplot(EVitals.loc[(EVitals['WEIGHT'] > 0) & (EVitals['WEIGHT'] < 190)]['WEIGHT'])
```

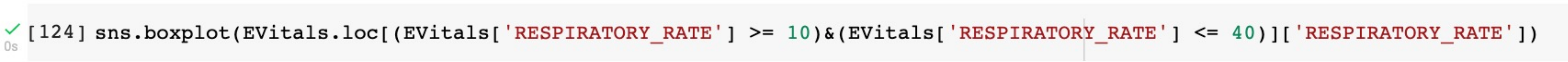
Validating the “Pulse”



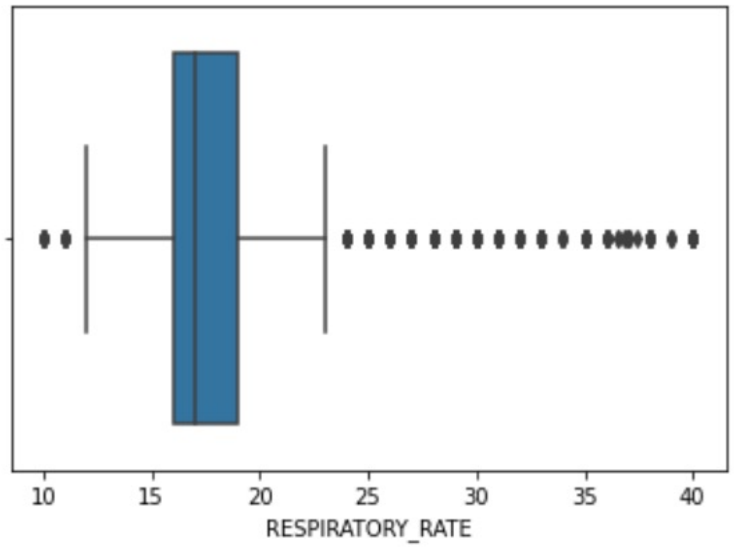
```
sns.boxplot(EVitals.loc[(EVitals['PULSE'] > 50)&(EVitals['PULSE'] < 100)]['PULSE'])
```



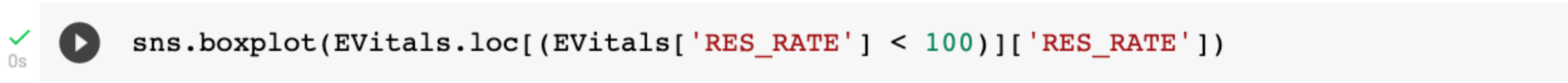
Validating the “Respiratory Rate”



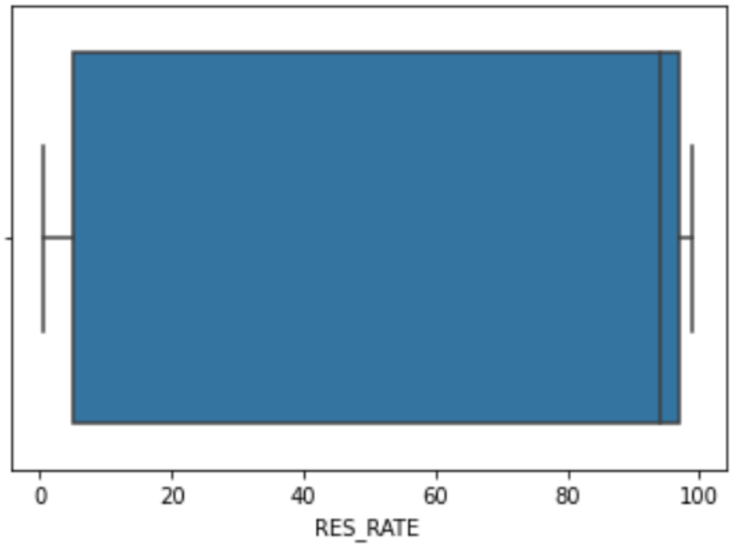
```
✓ [124] sns.boxplot(EVitals.loc[(EVitals['RESPIRATORY_RATE'] >= 10)&(EVitals['RESPIRATORY_RATE'] <= 40)]['RESPIRATORY_RATE'])
0s
```



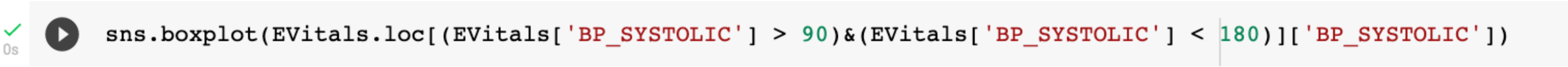
Validating the “O2_Saturation”



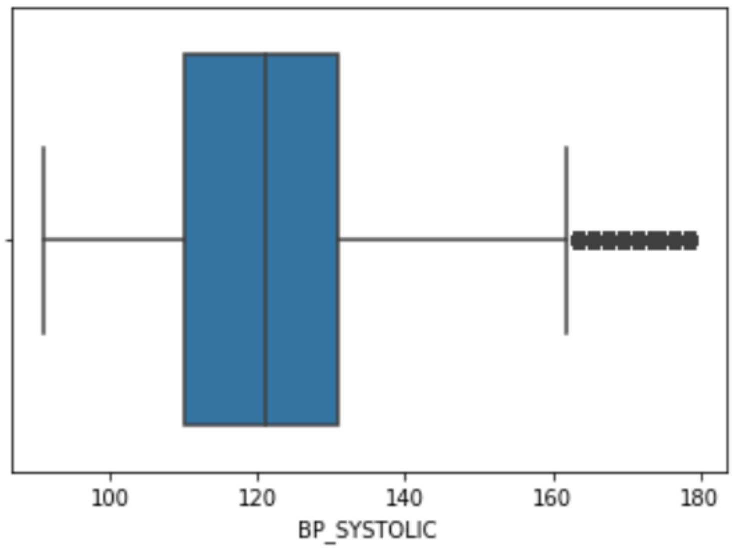
```
sns.boxplot(EVitals.loc[(EVitals['RES_RATE'] < 100)]['RES_RATE'])
```



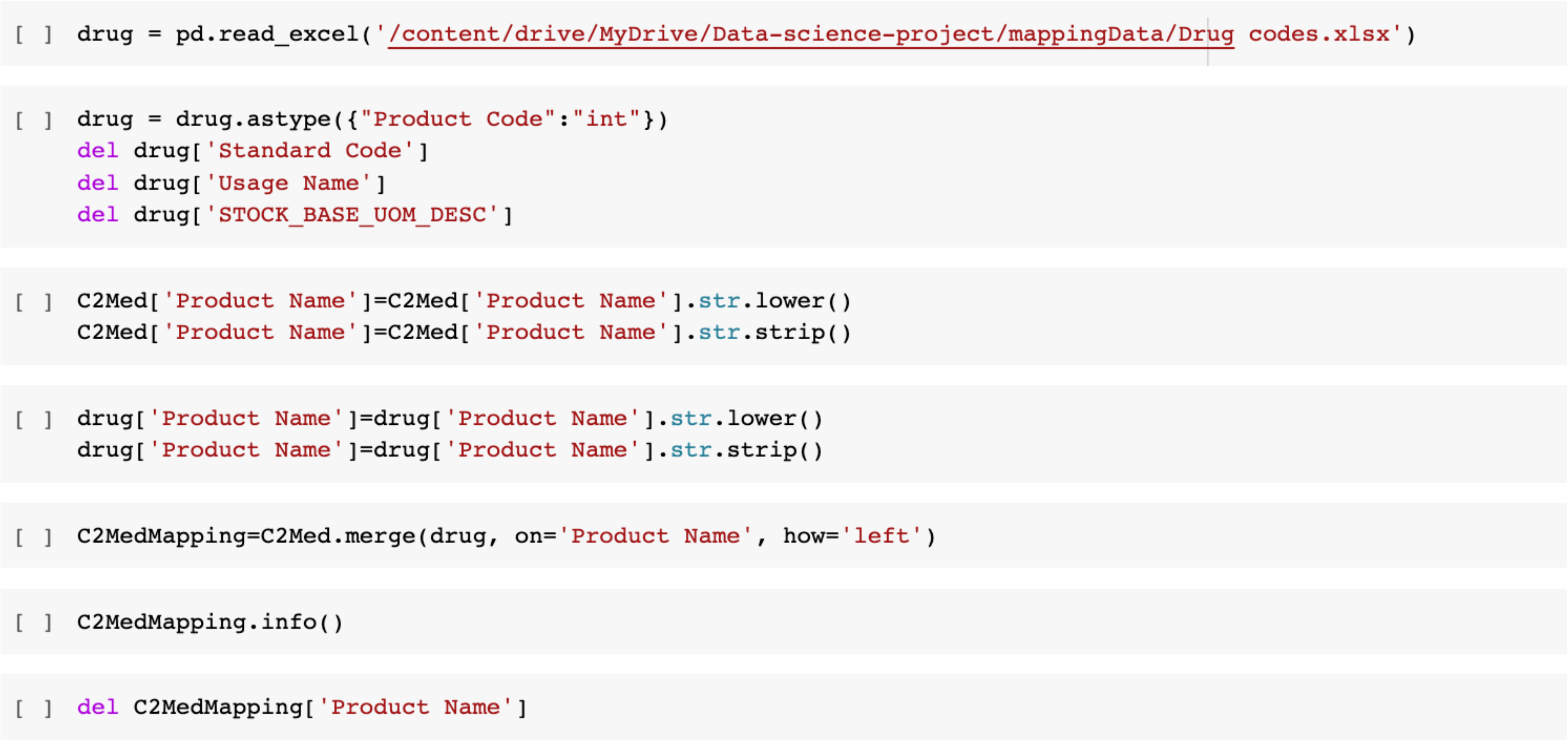
Validating the “Blood Pressure”



```
sns.boxplot(EVitals.loc[(EVitals['BP_SYSTOLIC'] > 90)&(EVitals['BP_SYSTOLIC'] < 180)]['BP_SYSTOLIC'])
```



Mapping the “Product Code”



```
[ ] drug = pd.read_excel('/content/drive/MyDrive/Data-science-project/mappingData/Drug codes.xlsx')
```

```
[ ] drug = drug.astype({"Product Code":"int"})
    del drug['Standard Code']
    del drug['Usage Name']
    del drug['STOCK_BASE_UOM_DESC']
```

```
[ ] C2Med['Product Name']=C2Med['Product Name'].str.lower()
    C2Med['Product Name']=C2Med['Product Name'].str.strip()
```

```
[ ] drug['Product Name']=drug['Product Name'].str.lower()
    drug['Product Name']=drug['Product Name'].str.strip()
```

```
[ ] C2MedMapping=C2Med.merge(drug, on='Product Name', how='left')
```

```
[ ] C2MedMapping.info()
```

```
[ ] del C2MedMapping['Product Name']
```

Data Management Plan

01

DATA DESCRIPTION, COLLECTION AND REUSING EXISTING DATA

02

DOCUMENTATION AND DATA QUALITY

03

LEGAL AND ETHICAL REQUIREMENTS

04

DATA SHARING AND LONG-TERM PRESERVATION

05

DATA MANAGEMENT RESPONSIBILITIES AND RESOURCES