Missing Data

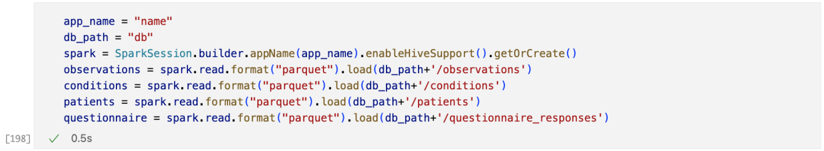
Tommaso Romano - 941796

June 13, 2022

Contents

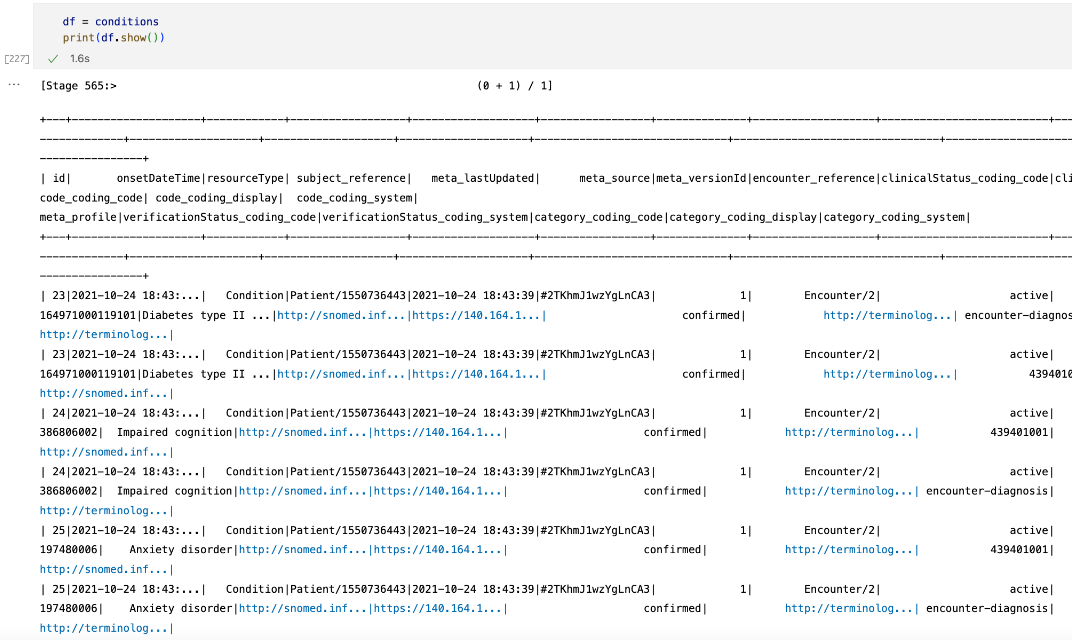
1 Introduzione

1.1 Pyspark



```
app_name = "name"
db_path = "db"
spark = SparkSession.builder.appName(app_name).enableHiveSupport().getOrCreate()
observations = spark.read.format("parquet").load(db_path+'/observations')
conditions = spark.read.format("parquet").load(db_path+'/conditions')
patients = spark.read.format("parquet").load(db_path+'/patients')
questionnaire = spark.read.format("parquet").load(db_path+'/questionnaire_responses')
```

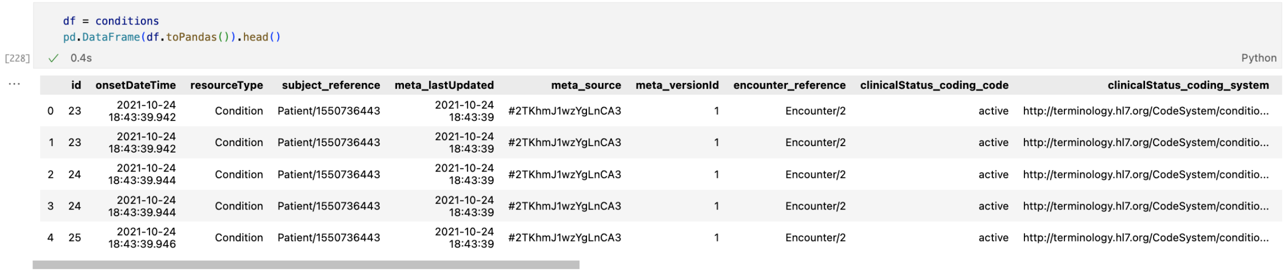
[198] ✓ 0.5s



```
df = conditions
print(df.show())
```

[227] ✓ 1.6s

```
[Stage 565:>                                                        (0 + 1) / 1]

+---+--------------------+------------+------------------+-------------------+------------------+--------------+--------------------+--------------------------+---
-------------+--------------------+--------------------+--------------------+------------------------------+---------------------------------+--------------------
----------------+
| id|       onsetDateTime|resourceType| subject_reference|   meta_lastUpdated|      meta_source|meta_versionId|encounter_reference|clinicalStatus_coding_code|cli
code_coding_code| code_coding_display|  code_coding_system|
meta_profile|verificationStatus_coding_code|verificationStatus_coding_system|category_coding_code|category_coding_display|category_coding_system|
+---+--------------------+------------+------------------+-------------------+------------------+--------------+--------------------+--------------------------+---
-------------+--------------------+--------------------+--------------------+------------------------------+---------------------------------+--------------------
----------------+
| 23|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|             1|        Encounter/2|                    active|
164971000119101|Diabetes type II ...|http://snomed.inf...|https://140.164.1...|                      confirmed|            http://terminolog...| encounter-diagnos
http://terminolog...|
| 23|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|             1|        Encounter/2|                    active|
164971000119101|Diabetes type II ...|http://snomed.inf...|https://140.164.1...|                      confirmed|            http://terminolog...|          4394010
http://snomed.inf...|
| 24|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|             1|        Encounter/2|                    active|
386806002|  Impaired cognition|http://snomed.inf...|https://140.164.1...|                      confirmed|            http://terminolog...|          439401001|
http://snomed.inf...|
| 24|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|             1|        Encounter/2|                    active|
386806002|  Impaired cognition|http://snomed.inf...|https://140.164.1...|                      confirmed|            http://terminolog...| encounter-diagnosis|
http://terminolog...|
| 25|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|             1|        Encounter/2|                    active|
197480006|    Anxiety disorder|http://snomed.inf...|https://140.164.1...|                      confirmed|            http://terminolog...|          439401001|
http://snomed.inf...|
| 25|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|             1|        Encounter/2|                    active|
197480006|    Anxiety disorder|http://snomed.inf...|https://140.164.1...|                      confirmed|            http://terminolog...| encounter-diagnosis|
http://terminolog...|
```

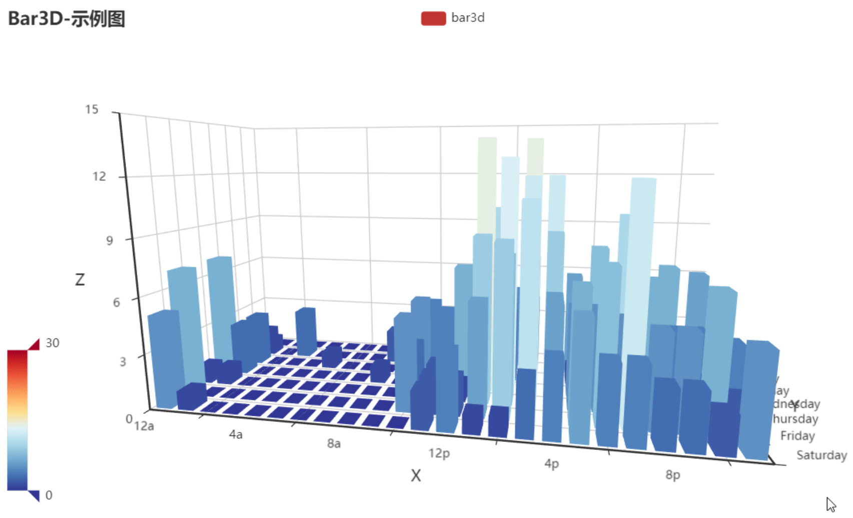


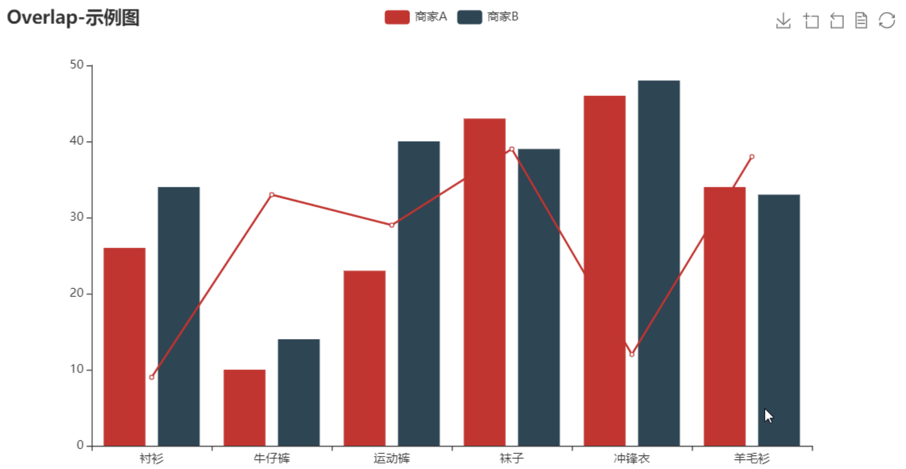
```
df = conditions
pd.DataFrame(df.toPandas()).head()
```

[228] ✓ 0.4s Python

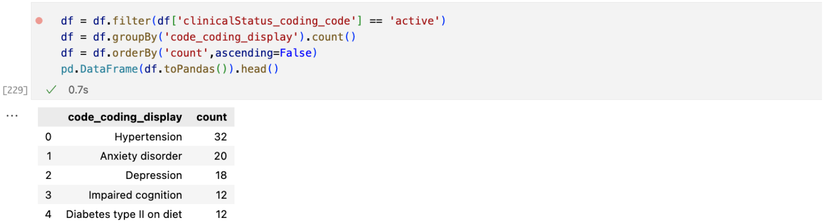
	id	onsetDateTime	resourceType	subject_reference	meta_lastUpdated	meta_source	meta_versionId	encounter_reference	clinicalStatus_coding_code	clinicalStatus_coding_system
0	23	2021-10-24 18:43:39.942	Condition	Patient/1550736443	2021-10-24 18:43:39	#2TKhmJ1wzYgLnCA3	1	Encounter/2	active	http://terminology.hl7.org/CodeSystem/conditio...
1	23	2021-10-24 18:43:39.942	Condition	Patient/1550736443	2021-10-24 18:43:39	#2TKhmJ1wzYgLnCA3	1	Encounter/2	active	http://terminology.hl7.org/CodeSystem/conditio...
2	24	2021-10-24 18:43:39.944	Condition	Patient/1550736443	2021-10-24 18:43:39	#2TKhmJ1wzYgLnCA3	1	Encounter/2	active	http://terminology.hl7.org/CodeSystem/conditio...
3	24	2021-10-24 18:43:39.944	Condition	Patient/1550736443	2021-10-24 18:43:39	#2TKhmJ1wzYgLnCA3	1	Encounter/2	active	http://terminology.hl7.org/CodeSystem/conditio...
4	25	2021-10-24 18:43:39.946	Condition	Patient/1550736443	2021-10-24 18:43:39	#2TKhmJ1wzYgLnCA3	1	Encounter/2	active	http://terminology.hl7.org/CodeSystem/conditio...

1.2 ECharts


Bar3D-示例图
bar3d
15
12
9
Z
6
3
0
30
0
12a
4a
8a
12p
4p
8p
X
ay
dnesday
Y
hursday
Friday
Saturday


Overlap-示例图
商家A
商家B
50
40
30
20
10
0
衬衫
牛仔裤
运动裤
袜子
冲锋衣
羊毛衫

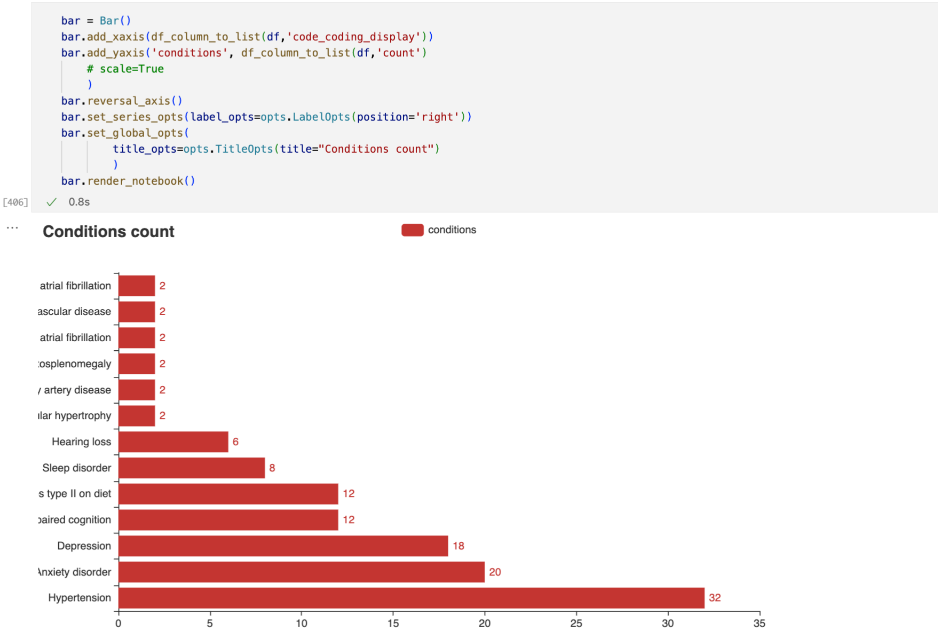
1.3 Conditions



```
df = df.filter(df['clinicalStatus_coding_code'] == 'active')
df = df.groupBy('code_coding_display').count()
df = df.orderBy('count',ascending=False)
pd.DataFrame(df.toPandas()).head()
```

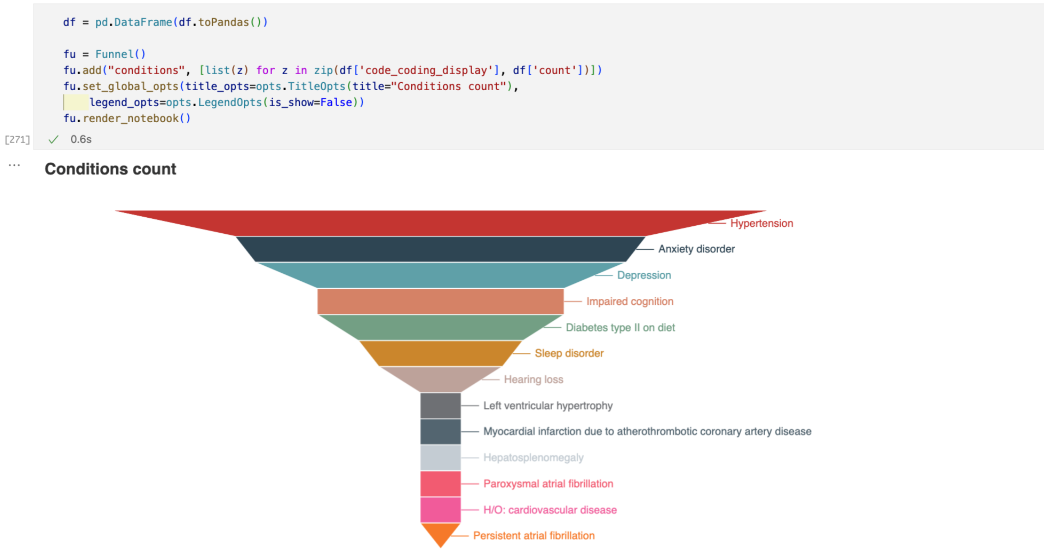
[229] ✓ 0.7s

	code_coding_display	count
0	Hypertension	32
1	Anxiety disorder	20
2	Depression	18
3	Impaired cognition	12
4	Diabetes type II on diet	12



```
bar = Bar()
bar.add_xaxis(df_column_to_list(df,'code_coding_display'))
bar.add_yaxis('conditions', df_column_to_list(df,'count')
    # scale=True
    )
bar.reversal_axis()
bar.set_series_opts(label_opts=opts.LabelOpts(position='right'))
bar.set_global_opts(
        title_opts=opts.TitleOpts(title="Conditions count")
        )
bar.render_notebook()
```

[406] ✓ 0.8s



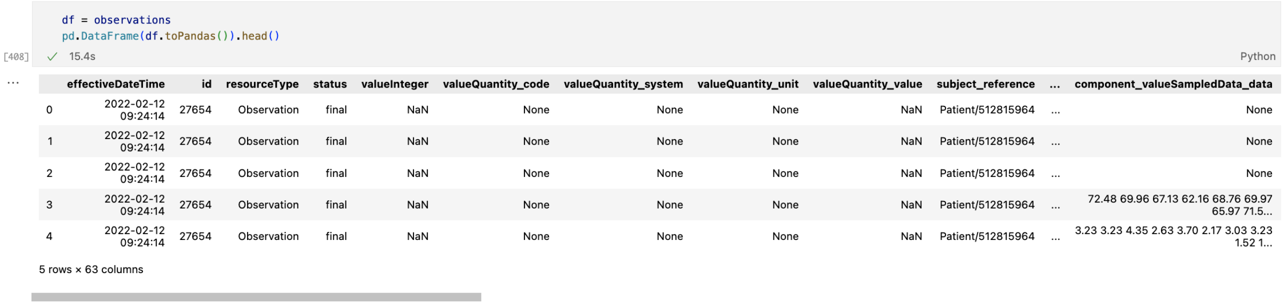
```
df = pd.DataFrame(df.toPandas())

fu = Funnel()
fu.add("conditions", [list(z) for z in zip(df['code_coding_display'], df['count'])])
fu.set_global_opts(title_opts=opts.TitleOpts(title="Conditions count"),
    legend_opts=opts.LegendOpts(is_show=False))
fu.render_notebook()
```

[271] ✓ 0.6s

Conditions count

Hypertension
Anxiety disorder
Depression
Impaired cognition
Diabetes type II on diet
Sleep disorder
Hearing loss
Left ventricular hypertrophy
Myocardial infarction due to atherothrombotic coronary artery disease
Hepatosplenomegaly
Paroxysmal atrial fibrillation
H/O: cardiovascular disease
Persistent atrial fibrillation

2 Missing data



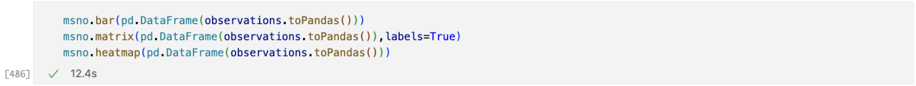
```
df = observations
pd.DataFrame(df.toPandas()).head()
```

[408] ✓ 15.4s Python

	effectiveDateTime	id	resourceType	status	valueInteger	valueQuantity_code	valueQuantity_system	valueQuantity_unit	valueQuantity_value	subject_reference	...	component_valueSampledData_data
0	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	None
1	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	None
2	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	None
3	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	72.48 69.96 67.13 62.16 68.76 69.97 65.97 71.5...
4	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	3.23 3.23 4.35 2.63 3.70 2.17 3.03 3.23 1.52 1...

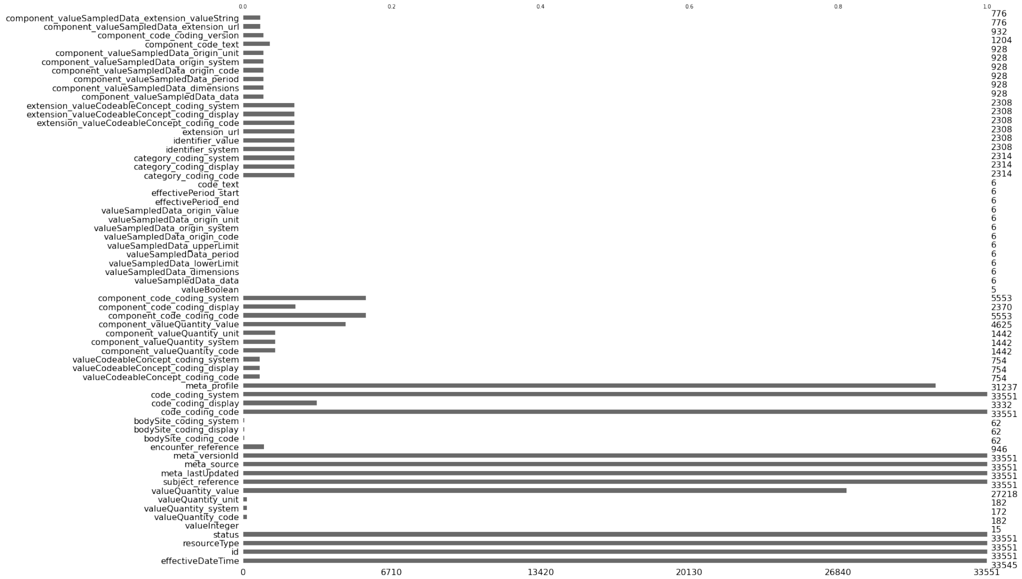
5 rows × 63 columns

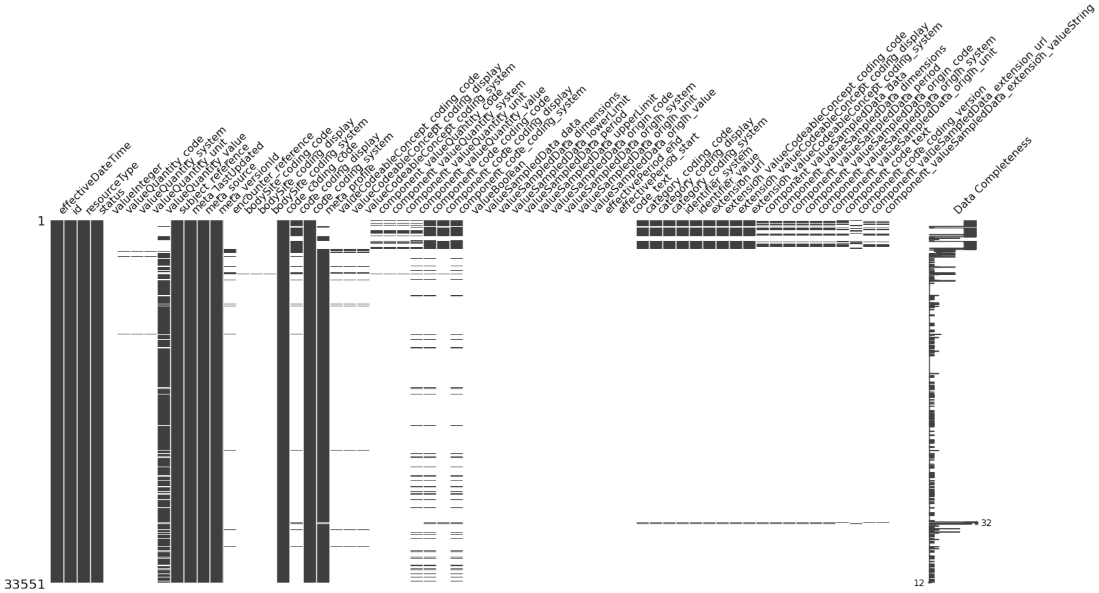
2.1 Missingno

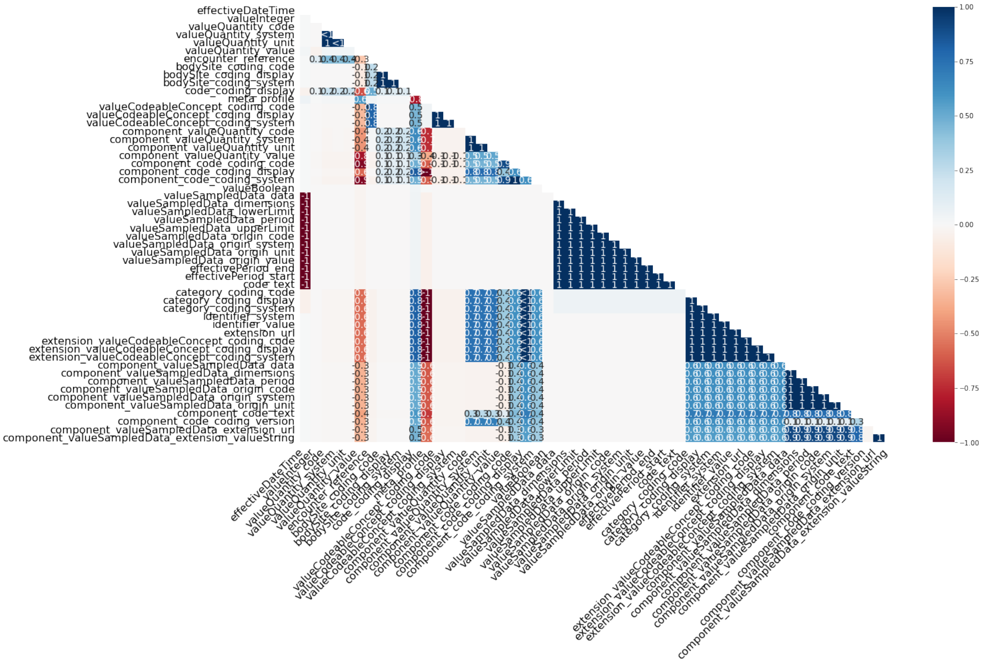


```
msno.bar(pd.DataFrame(observations.toPandas()))
msno.matrix(pd.DataFrame(observations.toPandas()),labels=True)
msno.heatmap(pd.DataFrame(observations.toPandas()))
```

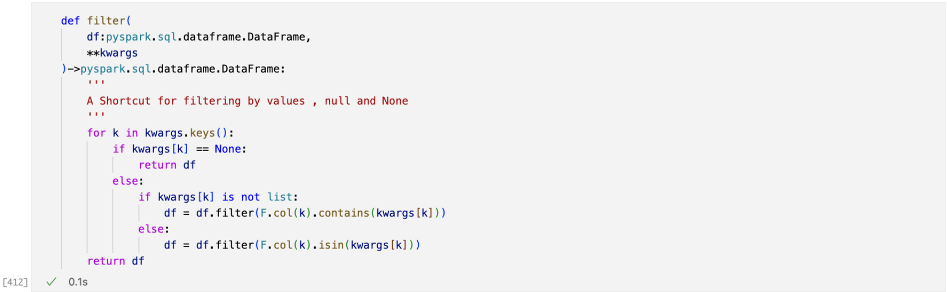
[486] ✓ 12.4s


0.0
0.2
0.4
0.6
0.8
1.0
component_valueSampledData_extension_valueString 776
component_valueSampledData_extension_url 776
component_code_coding_version 932
component_code_text 1204
component_valueSampledData_origin_unit 928
component_valueSampledData_origin_system 928
component_valueSampledData_origin_code 928
component_valueSampledData_period 928
component_valueSampledData_dimensions 928
component_valueSampledData_data 928
extension_valueCodeableConcept_coding_system 2308
extension_valueCodeableConcept_coding_display 2308
extension_valueCodeableConcept_coding_code 2308
extension_url 2308
identifier_value 2308
identifier_system 2308
category_coding_system 2314
category_coding_display 2314
category_coding_code 2314
code_text 6
effectivePeriod_start 6
effectivePeriod_end 6
valueSampledData_origin_value 6
valueSampledData_origin_unit 6
valueSampledData_origin_system 6
valueSampledData_origin_code 6
valueSampledData_upperLimit 6
valueSampledData_period 6
valueSampledData_lowerLimit 6
valueSampledData_dimensions 6
valueSampledData_data 6
valueBoolean 5
component_code_coding_system 5553
component_code_coding_display 2370
component_code_coding_code 5553
component_valueQuantity_value 4625
component_valueQuantity_unit 1442
component_valueQuantity_system 1442
component_valueQuantity_code 1442
valueCodeableConcept_coding_system 754
valueCodeableConcept_coding_display 754
valueCodeableConcept_coding_code 754
meta_profile 31237
code_coding_system 33551
code_coding_display 3332
code_coding_code 33551
bodySite_coding_system 62
bodySite_coding_display 62
bodySite_coding_code 62
encounter_reference 946
meta_versionId 33551
meta_source 33551
meta_lastUpdated 33551
subject_reference 33551
valueQuantity_value 27218
valueQuantity_unit 182
valueQuantity_system 172
valueQuantity_code 182
valueInteger 15
status 33551
resourceType 33551
id 33551
effectiveDateTime 33545
0
6710
13420
20130
26840
33551


effectiveDateTime
id
resourceType
status
valueInteger
valueQuantity_code
valueQuantity_system
valueQuantity_unit
valueQuantity_value
subject_reference
meta_lastUpdated
meta_source
meta_versionId
encounter_reference
bodySite_coding_code
bodySite_coding_display
bodySite_coding_system
code_coding_code
code_coding_display
code_coding_system
meta_profile
valueCodeableConcept_coding_code
valueCodeableConcept_coding_display
valueCodeableConcept_coding_system
component_valueQuantity_code
component_valueQuantity_system
component_valueQuantity_unit
component_valueQuantity_value
component_code_coding_code
component_code_coding_display
component_code_coding_system
valueBoolean
valueSampledData_data
valueSampledData_dimensions
valueSampledData_lowerLimit
valueSampledData_period
valueSampledData_upperLimit
valueSampledData_origin_code
valueSampledData_origin_system
valueSampledData_origin_unit
valueSampledData_origin_value
effectivePeriod_end
effectivePeriod_start
code_text
category_coding_code
category_coding_display
category_coding_system
identifier_system
identifier_value
extension_url
extension_valueCodeableConcept_coding_code
extension_valueCodeableConcept_coding_display
extension_valueCodeableConcept_coding_system
component_valueSampledData_data
component_valueSampledData_dimensions
component_valueSampledData_period
component_valueSampledData_origin_code
component_valueSampledData_origin_system
component_valueSampledData_origin_unit
component_code_text
component_code_coding_version
component_valueSampledData_extension_url
component_valueSampledData_extension_valueString
Data Completeness
1
33551
32
12

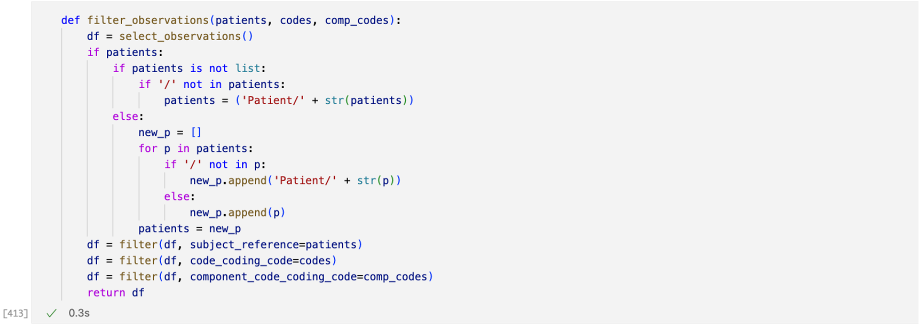


2.2 Filter



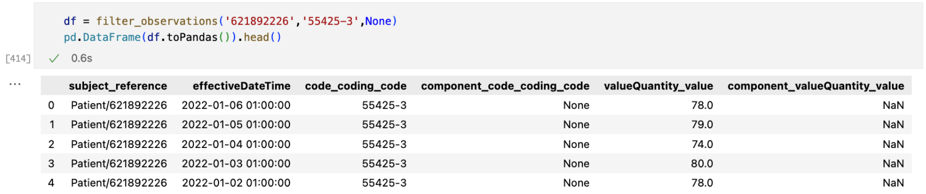
```
def filter(
    df:pyspark.sql.dataframe.DataFrame,
    **kwargs
)->pyspark.sql.dataframe.DataFrame:
    '''
    A Shortcut for filtering by values , null and None
    '''
    for k in kwargs.keys():
        if kwargs[k] == None:
            return df
        else:
            if kwargs[k] is not list:
                df = df.filter(F.col(k).contains(kwargs[k]))
            else:
                df = df.filter(F.col(k).isin(kwargs[k]))
    return df
```

[412] ✓ 0.1s



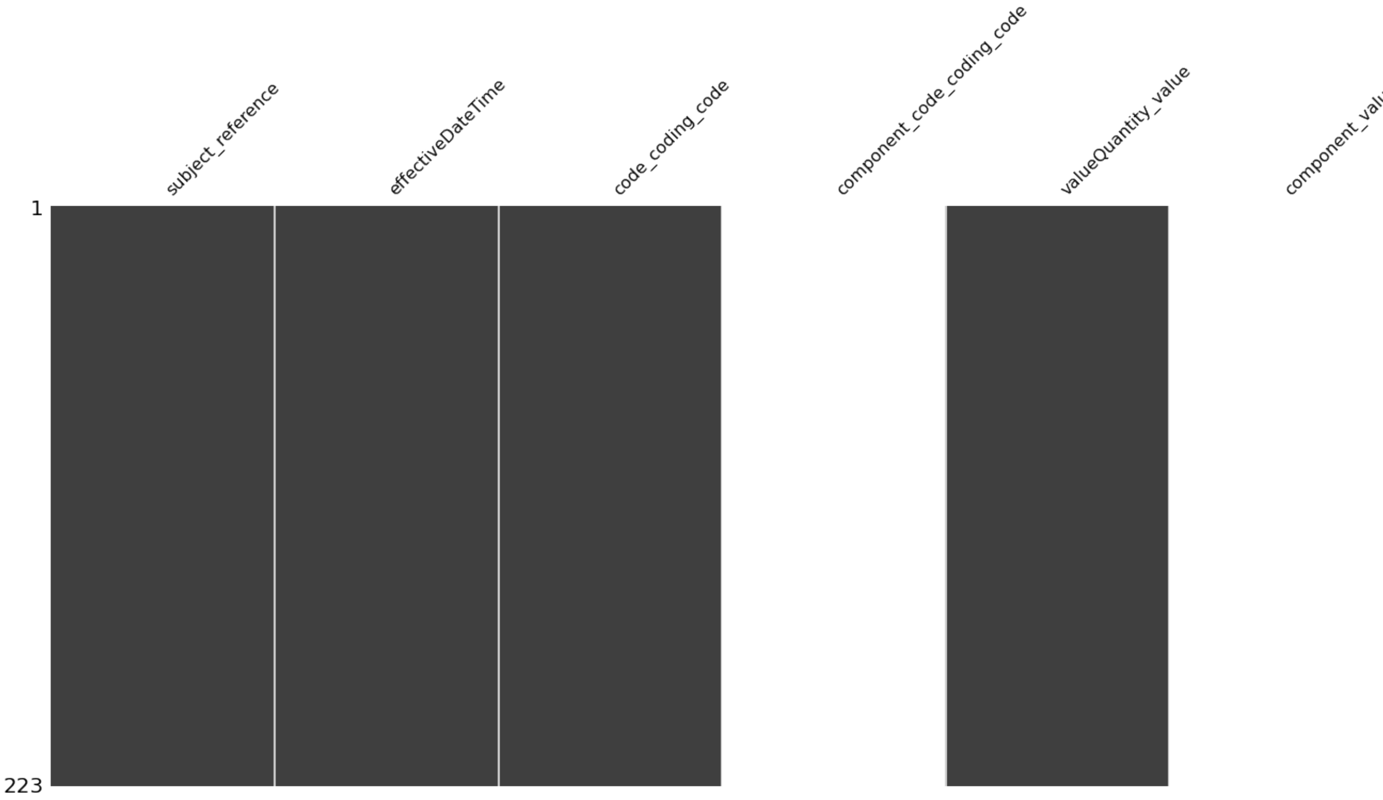
```
def filter_observations(patients, codes, comp_codes):
    df = select_observations()
    if patients:
        if patients is not list:
            if '/' not in patients:
                patients = ('Patient/' + str(patients))
        else:
            new_p = []
            for p in patients:
                if '/' not in p:
                    new_p.append('Patient/' + str(p))
                else:
                    new_p.append(p)
            patients = new_p
    df = filter(df, subject_reference=patients)
    df = filter(df, code_coding_code=codes)
    df = filter(df, component_code_coding_code=comp_codes)
    return df
```

[413] ✓ 0.3s



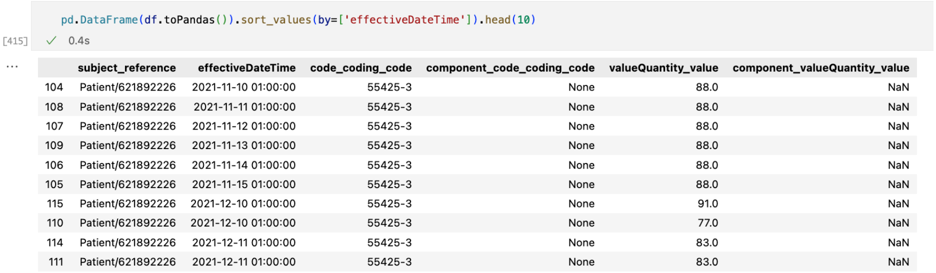
```
df = filter_observations('621892226','55425-3',None)
pd.DataFrame(df.toPandas()).head()
```

	subject_reference	effectiveDateTime	code_coding_code	component_code_coding_code	valueQuantity_value	component_valueQuantity_value
0	Patient/621892226	2022-01-06 01:00:00	55425-3	None	78.0	NaN
1	Patient/621892226	2022-01-05 01:00:00	55425-3	None	79.0	NaN
2	Patient/621892226	2022-01-04 01:00:00	55425-3	None	74.0	NaN
3	Patient/621892226	2022-01-03 01:00:00	55425-3	None	80.0	NaN
4	Patient/621892226	2022-01-02 01:00:00	55425-3	None	78.0	NaN



3 Imputation

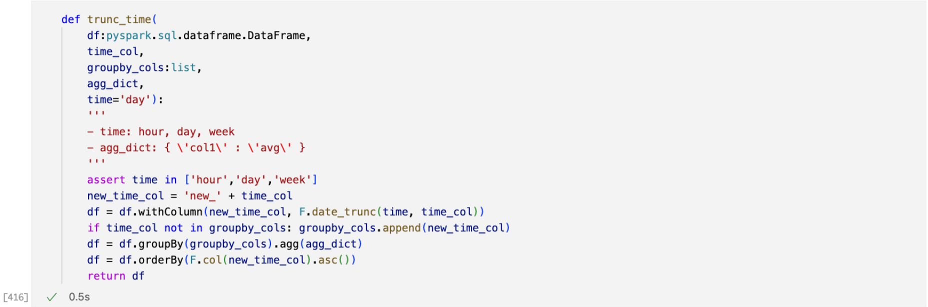
3.1 Time Series



```
pd.DataFrame(df.toPandas()).sort_values(by=['effectiveDateTime']).head(10)
```

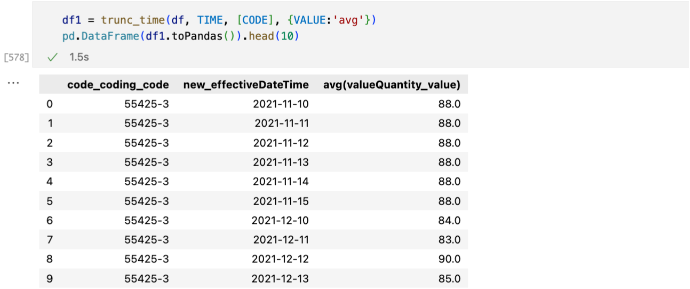
[415] ✓ 0.4s

	subject_reference	effectiveDateTime	code_coding_code	component_code_coding_code	valueQuantity_value	component_valueQuantity_value
104	Patient/621892226	2021-11-10 01:00:00	55425-3	None	88.0	NaN
108	Patient/621892226	2021-11-11 01:00:00	55425-3	None	88.0	NaN
107	Patient/621892226	2021-11-12 01:00:00	55425-3	None	88.0	NaN
109	Patient/621892226	2021-11-13 01:00:00	55425-3	None	88.0	NaN
106	Patient/621892226	2021-11-14 01:00:00	55425-3	None	88.0	NaN
105	Patient/621892226	2021-11-15 01:00:00	55425-3	None	88.0	NaN
115	Patient/621892226	2021-12-10 01:00:00	55425-3	None	91.0	NaN
110	Patient/621892226	2021-12-10 01:00:00	55425-3	None	77.0	NaN
114	Patient/621892226	2021-12-11 01:00:00	55425-3	None	83.0	NaN
111	Patient/621892226	2021-12-11 01:00:00	55425-3	None	83.0	NaN



```
def trunc_time(
    df:pyspark.sql.dataframe.DataFrame,
    time_col,
    groupby_cols:list,
    agg_dict,
    time='day'):
    '''
    - time: hour, day, week
    - agg_dict: { \'col1\' : \'avg\' }
    '''
    assert time in ['hour','day','week']
    new_time_col = 'new_' + time_col
    df = df.withColumn(new_time_col, F.date_trunc(time, time_col))
    if time_col not in groupby_cols: groupby_cols.append(new_time_col)
    df = df.groupBy(groupby_cols).agg(agg_dict)
    df = df.orderBy(F.col(new_time_col).asc())
    return df
```

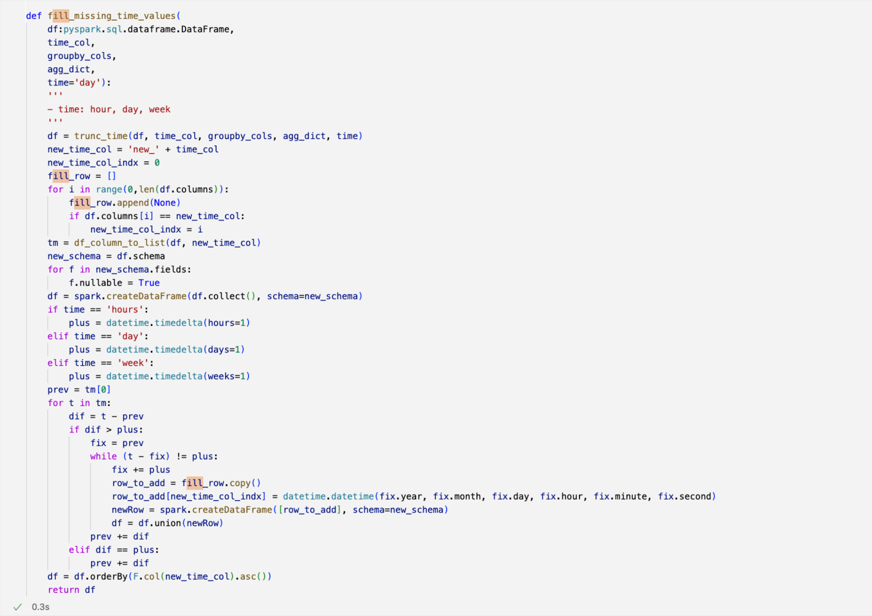
[416] ✓ 0.5s



```
df1 = trunc_time(df, TIME, [CODE], {VALUE:'avg'})
pd.DataFrame(df1.toPandas()).head(10)
```

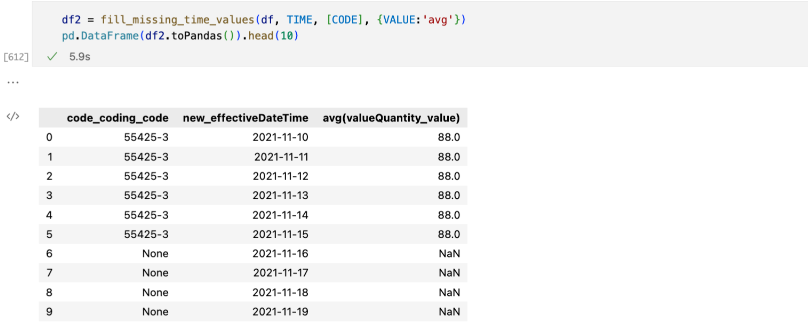
[578] ✓ 1.5s

	code_coding_code	new_effectiveDateTime	avg(valueQuantity_value)
0	55425-3	2021-11-10	88.0
1	55425-3	2021-11-11	88.0
2	55425-3	2021-11-12	88.0
3	55425-3	2021-11-13	88.0
4	55425-3	2021-11-14	88.0
5	55425-3	2021-11-15	88.0
6	55425-3	2021-12-10	84.0
7	55425-3	2021-12-11	83.0
8	55425-3	2021-12-12	90.0
9	55425-3	2021-12-13	85.0



```
def fill_missing_time_values(
    df:pyspark.sql.dataframe.DataFrame,
    time_col,
    groupby_cols,
    agg_dict,
    time='day'):
    '''
    - time: hour, day, week
    '''
    df = trunc_time(df, time_col, groupby_cols, agg_dict, time)
    new_time_col = 'new_' + time_col
    new_time_col_indx = 0
    fill_row = []
    for i in range(0,len(df.columns)):
        fill_row.append(None)
        if df.columns[i] == new_time_col:
            new_time_col_indx = i
    tm = df_column_to_list(df, new_time_col)
    new_schema = df.schema
    for f in new_schema.fields:
        f.nullable = True
    df = spark.createDataFrame(df.collect(), schema=new_schema)
    if time == 'hours':
        plus = datetime.timedelta(hours=1)
    elif time == 'day':
        plus = datetime.timedelta(days=1)
    elif time == 'week':
        plus = datetime.timedelta(weeks=1)
    prev = tm[0]
    for t in tm:
        dif = t - prev
        if dif > plus:
            fix = prev
            while (t - fix) != plus:
                fix += plus
                row_to_add = fill_row.copy()
                row_to_add[new_time_col_indx] = datetime.datetime(fix.year, fix.month, fix.day, fix.hour, fix.minute, fix.second)
                newRow = spark.createDataFrame([row_to_add], schema=new_schema)
                df = df.union(newRow)
            prev += dif
        elif dif == plus:
            prev += dif
    df = df.orderBy(F.col(new_time_col).asc())
    return df
```

[451] ✓ 0.3s

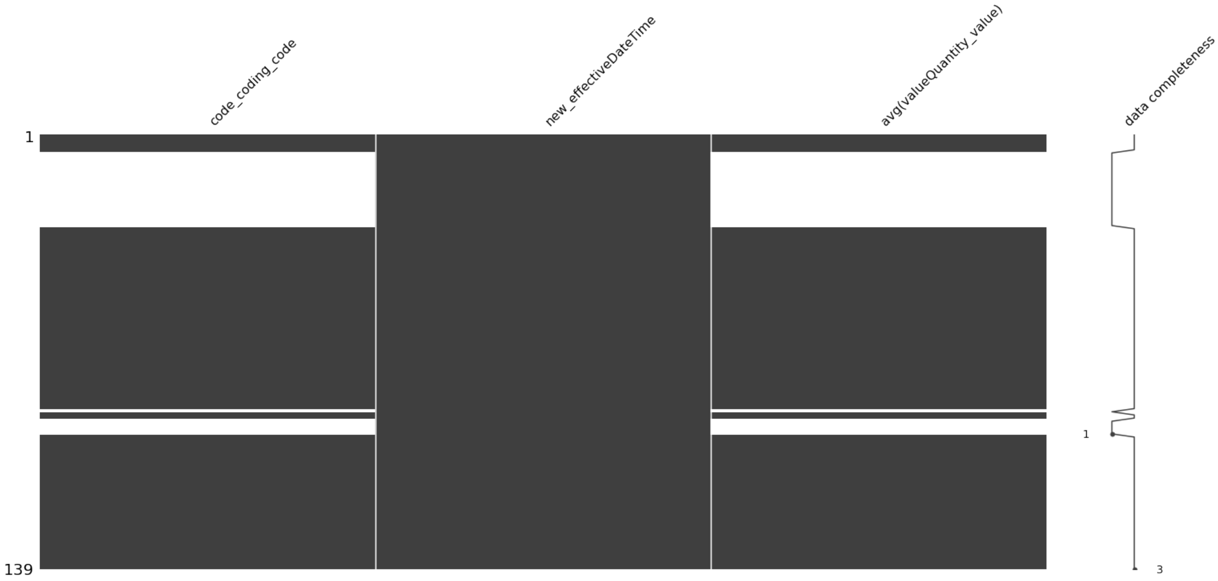


```
df2 = fill_missing_time_values(df, TIME, [CODE], {VALUE:'avg'})
pd.DataFrame(df2.toPandas()).head(10)
```

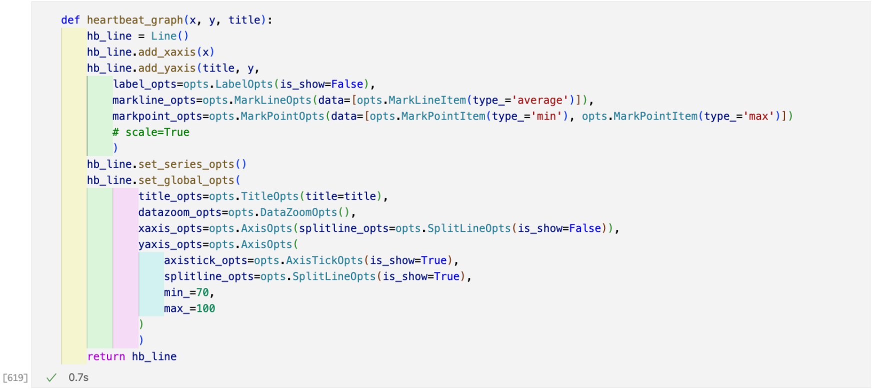
[612] ✓ 5.9s

...

	code_coding_code	new_effectiveDateTime	avg(valueQuantity_value)
0	55425-3	2021-11-10	88.0
1	55425-3	2021-11-11	88.0
2	55425-3	2021-11-12	88.0
3	55425-3	2021-11-13	88.0
4	55425-3	2021-11-14	88.0
5	55425-3	2021-11-15	88.0
6	None	2021-11-16	NaN
7	None	2021-11-17	NaN
8	None	2021-11-18	NaN
9	None	2021-11-19	NaN

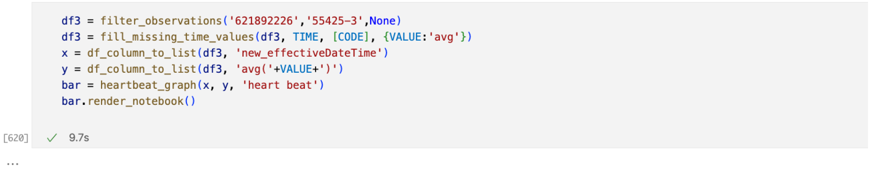


3.2 Simple Imputer



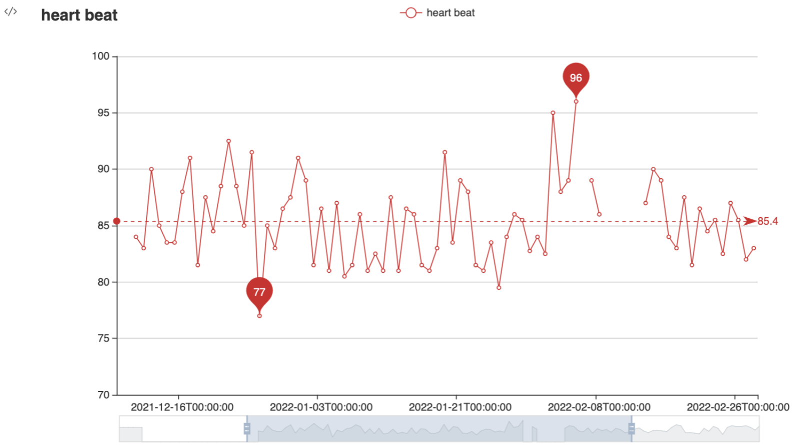
```
def heartbeat_graph(x, y, title):
    hb_line = Line()
    hb_line.add_xaxis(x)
    hb_line.add_yaxis(title, y,
        label_opts=opts.LabelOpts(is_show=False),
        markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_='average')]),
        markpoint_opts=opts.MarkPointOpts(data=[opts.MarkPointItem(type_='min'), opts.MarkPointItem(type_='max')])
        # scale=True
        )
    hb_line.set_series_opts()
    hb_line.set_global_opts(
            title_opts=opts.TitleOpts(title=title),
            datazoom_opts=opts.DataZoomOpts(),
            xaxis_opts=opts.AxisOpts(splitline_opts=opts.SplitLineOpts(is_show=False)),
            yaxis_opts=opts.AxisOpts(
                axistick_opts=opts.AxisTickOpts(is_show=True),
                splitline_opts=opts.SplitLineOpts(is_show=True),
                min_=70,
                max_=100
            )
        )
    return hb_line
```

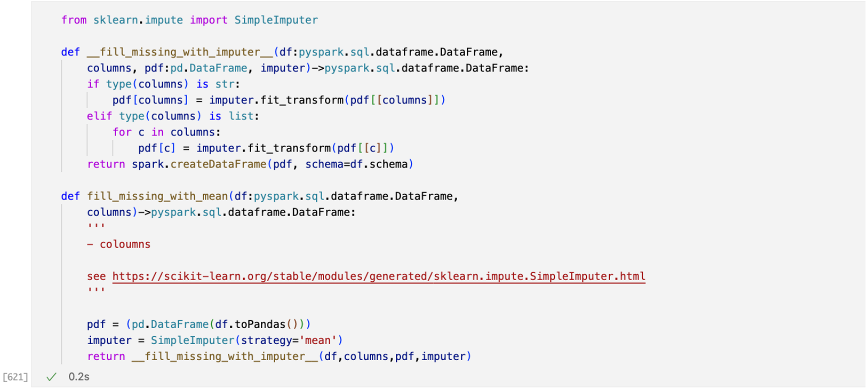
[619] ✓ 0.7s



```
df3 = filter_observations('621892226','55425-3',None)
df3 = fill_missing_time_values(df3, TIME, [CODE], {VALUE:'avg'})
x = df_column_to_list(df3, 'new_effectiveDateTime')
y = df_column_to_list(df3, 'avg('+VALUE+')')
bar = heartbeat_graph(x, y, 'heart beat')
bar.render_notebook()
```

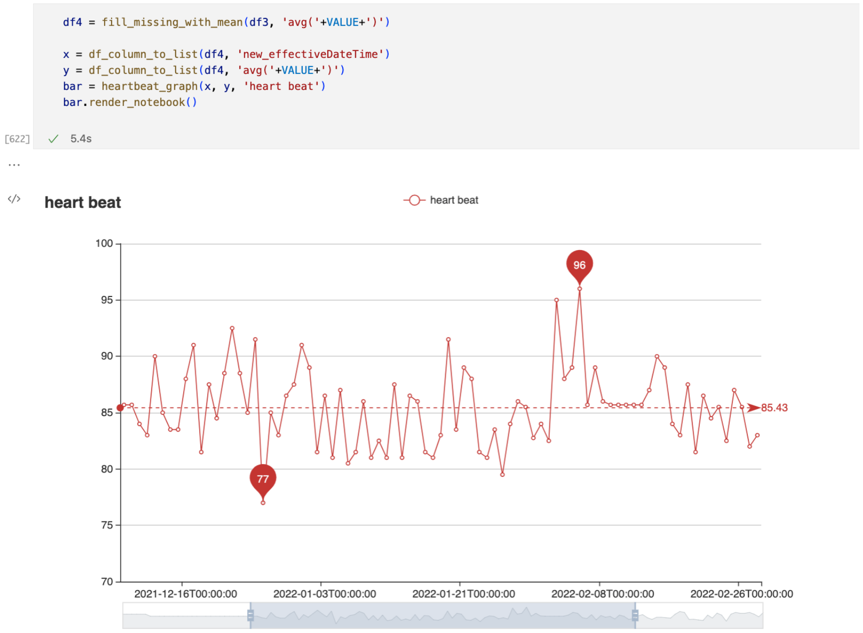
[620] ✓ 9.7s





```
from sklearn.impute import SimpleImputer

def __fill_missing_with_imputer__(df:pyspark.sql.dataframe.DataFrame,
    columns, pdf:pd.DataFrame, imputer)->pyspark.sql.dataframe.DataFrame:
    if type(columns) is str:
        pdf[columns] = imputer.fit_transform(pdf[[columns]])
    elif type(columns) is list:
        for c in columns:
            pdf[c] = imputer.fit_transform(pdf[[c]])
    return spark.createDataFrame(pdf, schema=df.schema)

def fill_missing_with_mean(df:pyspark.sql.dataframe.DataFrame,
    columns)->pyspark.sql.dataframe.DataFrame:
    '''
    - coloumns

    see https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html
    '''

    pdf = (pd.DataFrame(df.toPandas()))
    imputer = SimpleImputer(strategy='mean')
    return __fill_missing_with_imputer__(df,columns,pdf,imputer)
```

[621] ✓ 0.2s



```
df4 = fill_missing_with_mean(df3, 'avg('+VALUE+')')

x = df_column_to_list(df4, 'new_effectiveDateTime')
y = df_column_to_list(df4, 'avg('+VALUE+')')
bar = heartbeat_graph(x, y, 'heart beat')
bar.render_notebook()
```

[622] ✓ 5.4s

Lorem Ipsum

- item1.
- item2.