Missing Values Visualization

Tommaso Romano'
941796

01

Introduction

PySpark, Echarts, Conditions

02

Missing Values

Missingno, Filtering

03

Imputation

Time Series, Simple Imputer, Extension

PySpark Init

```
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

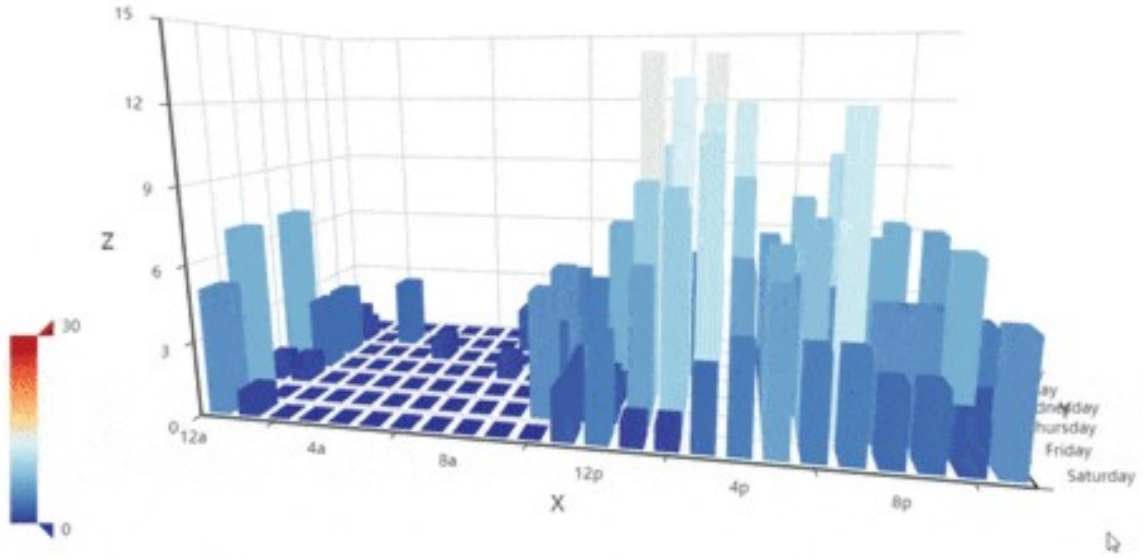
PySpark DataFrame Visualization

```
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_codi
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|
164971000119101|Diabetes type II ...|http://snomed.inf...|https://140.164.1...|                    confirmed|           http://terminolog...| encount
http://terminolog...|
| 23|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|            1|        Encounter/2|
164971000119101|Diabetes type II ...|http://snomed.inf...|https://140.164.1...|                    confirmed|           http://terminolog...|
http://snomed.inf...|
| 24|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|            1|        Encounter/2|
386806002|  Impaired cognition|http://snomed.inf...|https://140.164.1...|                  confirmed|           http://terminolog...|          439
http://snomed.inf...|
| 24|2021-10-24 18:43:...|   Condition|Patient/1550736443|2021-10-24 18:43:39|#2TKhmJ1wzYgLnCA3|            1|        Encounter/2|
```

Echarts Examples



Conditions Init

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

[228] ✓ 0.4s

	id	onsetDateTime	resourceType	subject_reference	meta_lastUpdated	meta_source	meta_versionId	encounter_reference	clinicalStatus_coding_code	clinica
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.or
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.or
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.or
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.or
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.or

Conditions Grouping

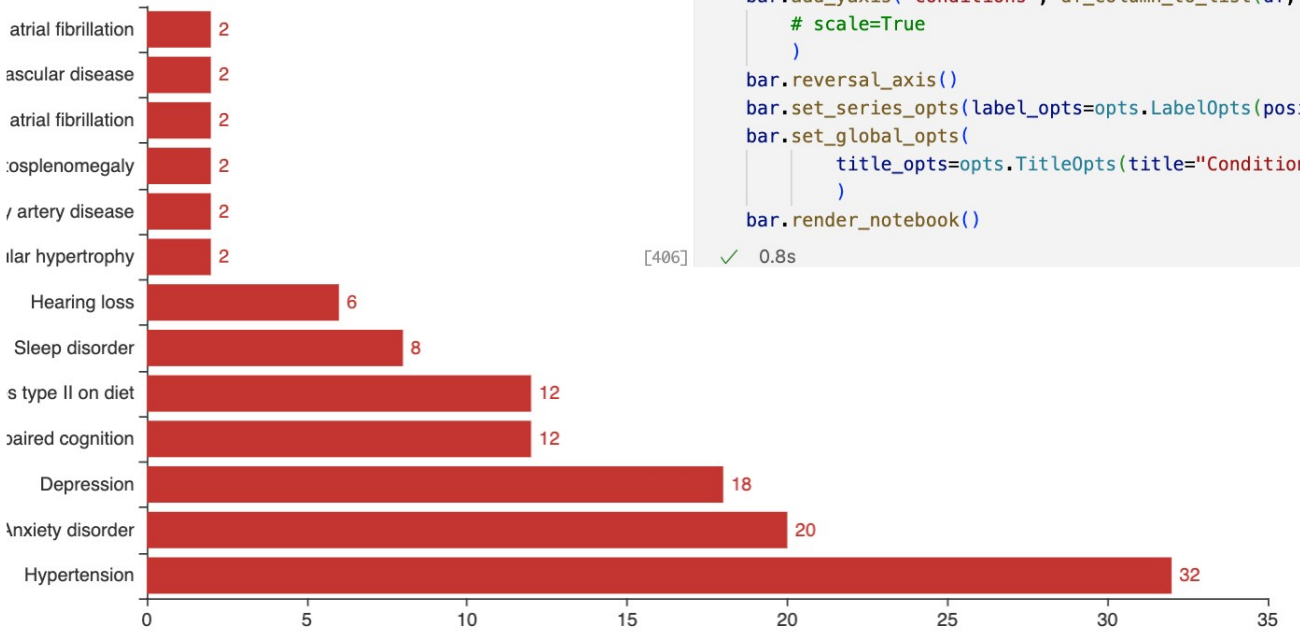
```
df = df.filter(df['clinicalStatus_coding_code'] == 'active')
df = df.groupBy('code_coding_display').count()
df = df.orderBy('count',ascending=False)
pd.DataFrame(df.toPandas()).head()
```

[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

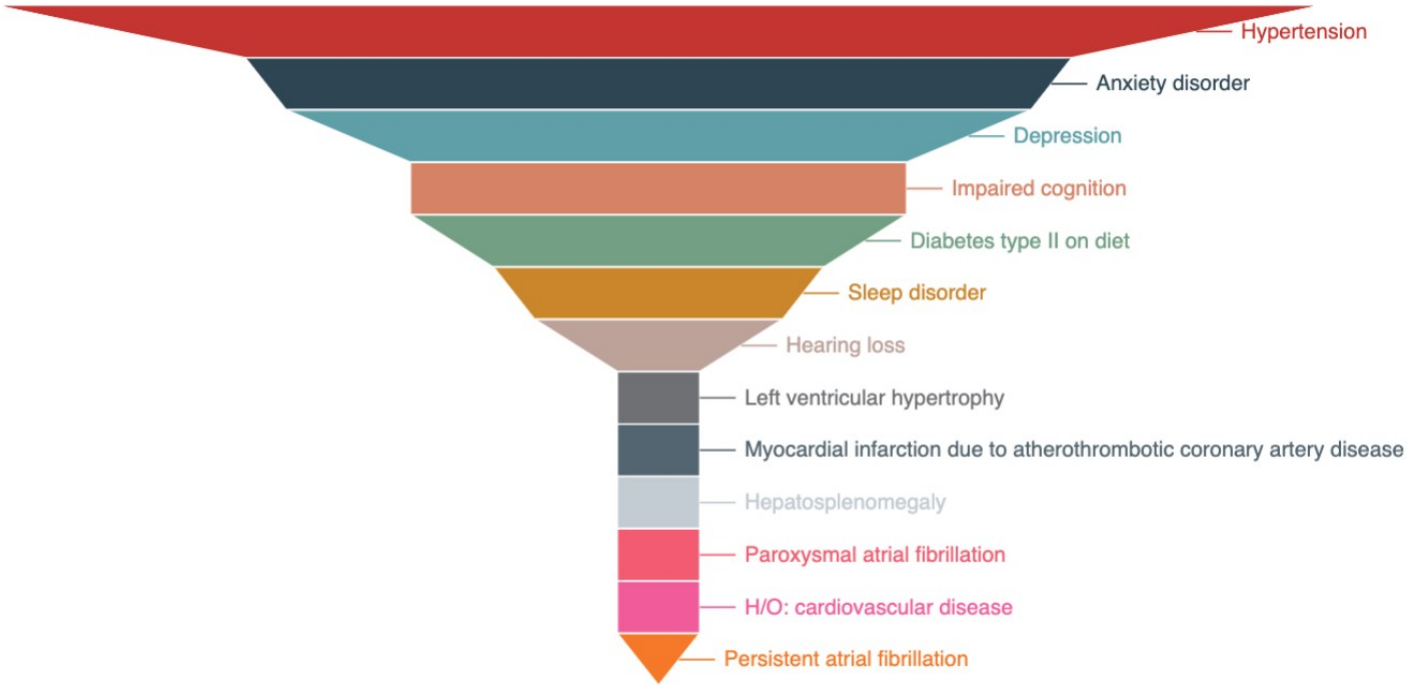
Conditions Visualizing (Bar)




```
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

Conditions Visualizing (Funnel)



Observations Init

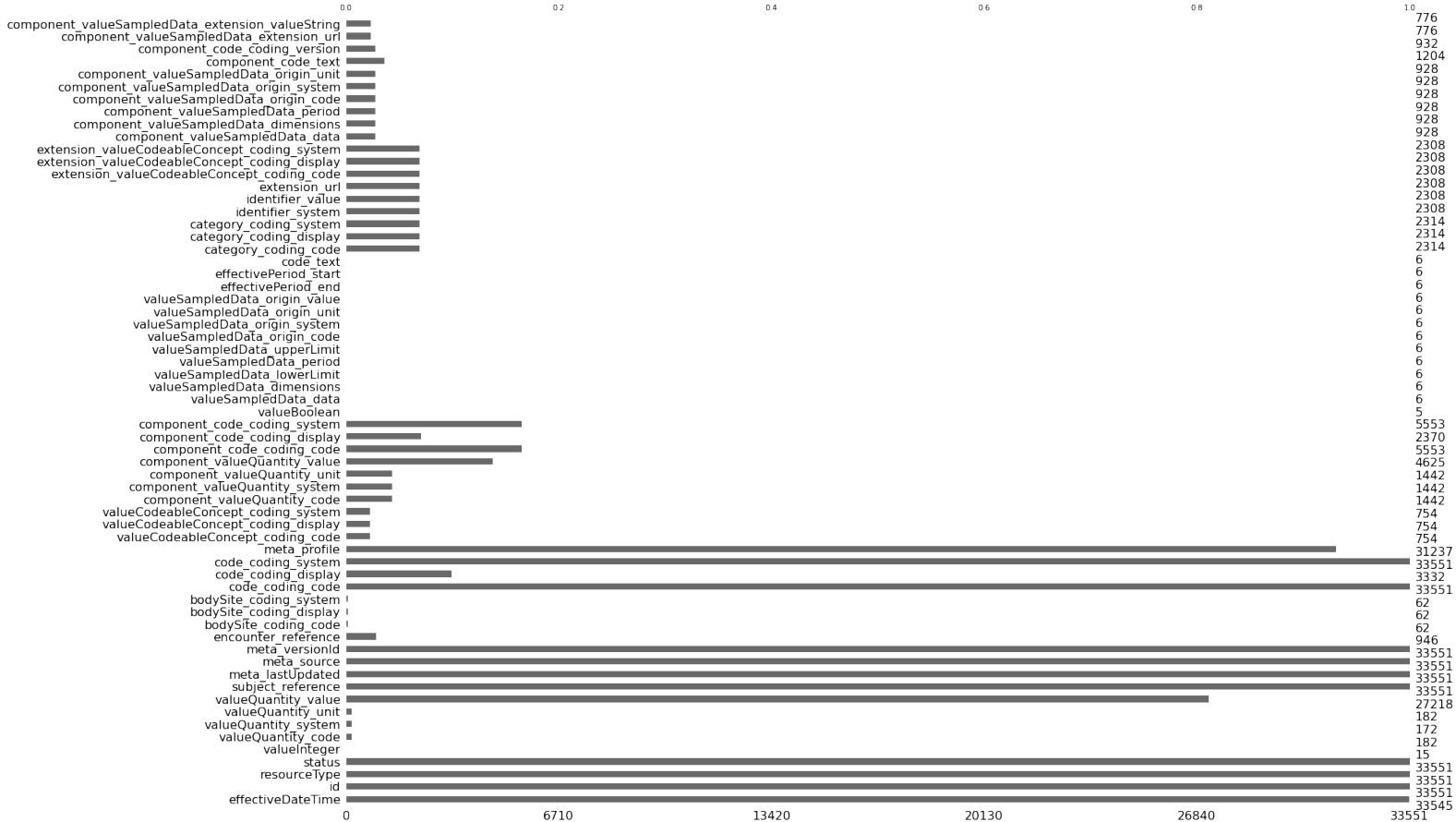
```
df = observations
pd.DataFrame(df.toPandas()).head()
```

[408] ✓ 15.4s

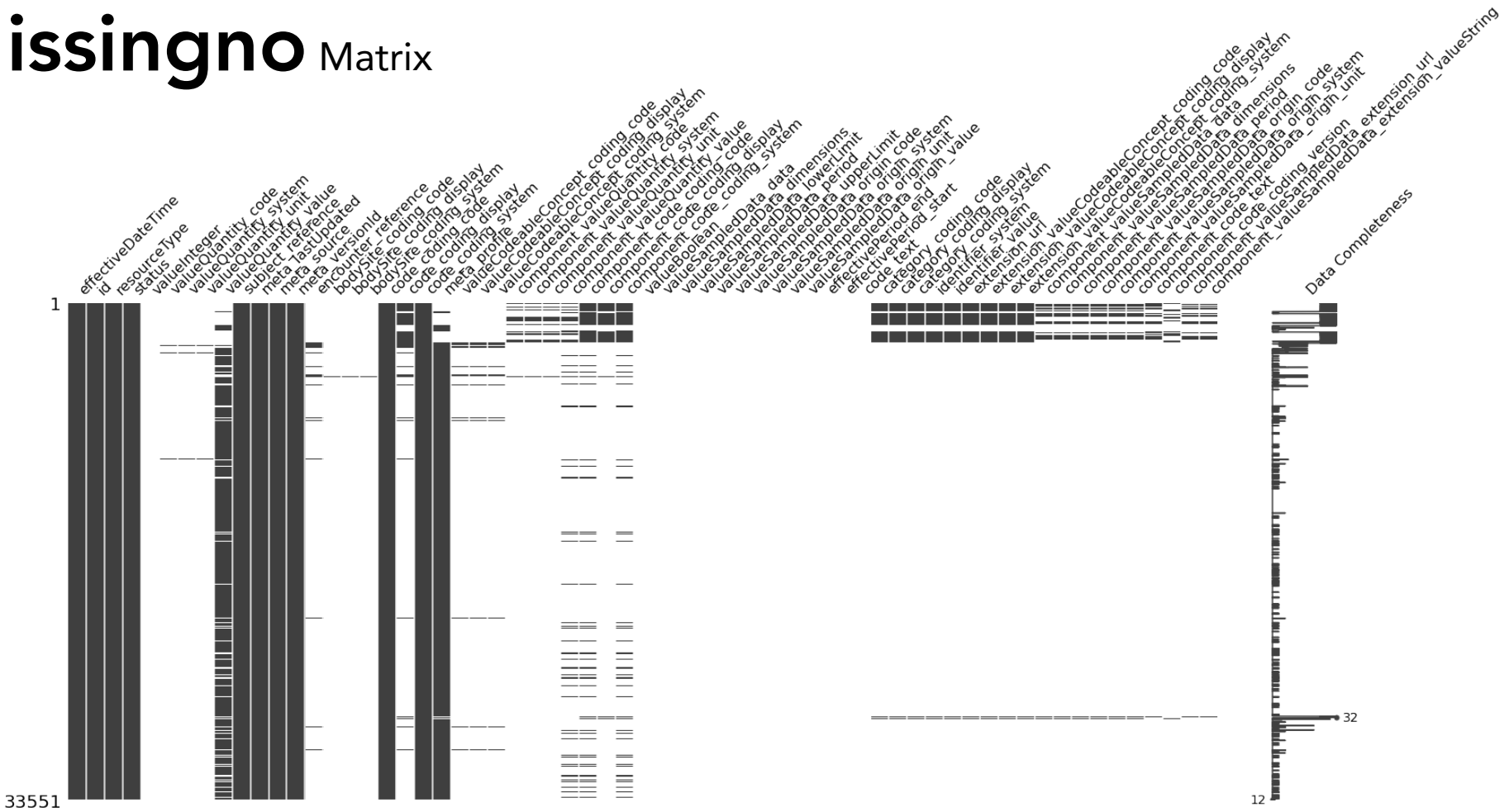
	effectiveDateTime	id	resourceType	status	valueInteger	valueQuantity_code	valueQuantity_system	valueQuantity_unit	valueQuantity_value	subject_reference	...	component_
0	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	
1	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	
2	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	
3	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	72.48 69.9
4	2022-02-12 09:24:14	27654	Observation	final	NaN	None	None	None	NaN	Patient/512815964	...	3.23 3.23 4.3

5 rows × 63 columns

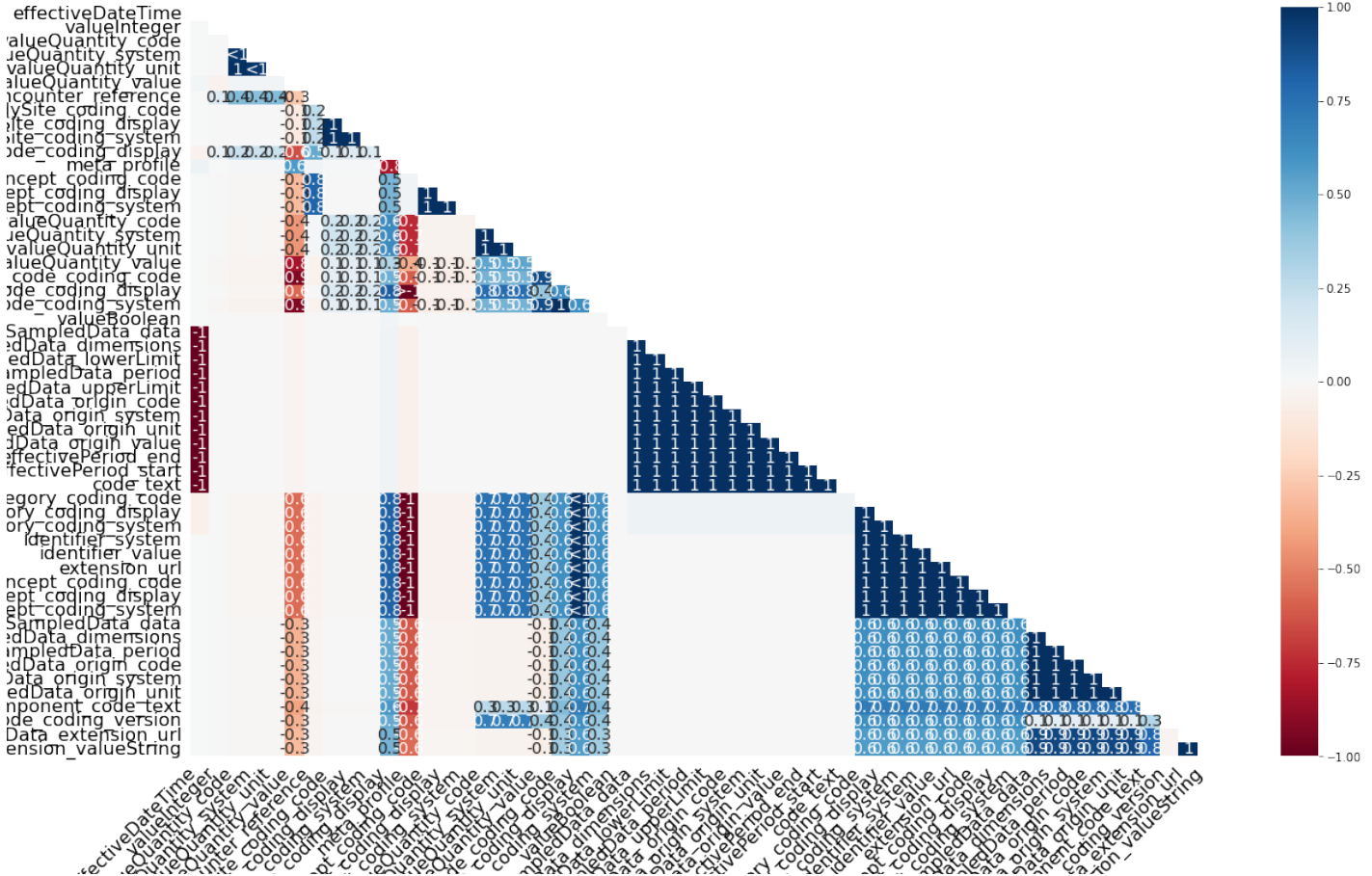
Missingno Bar



Missingno Matrix



Missingno Heatmap



Filtering Observations

```
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

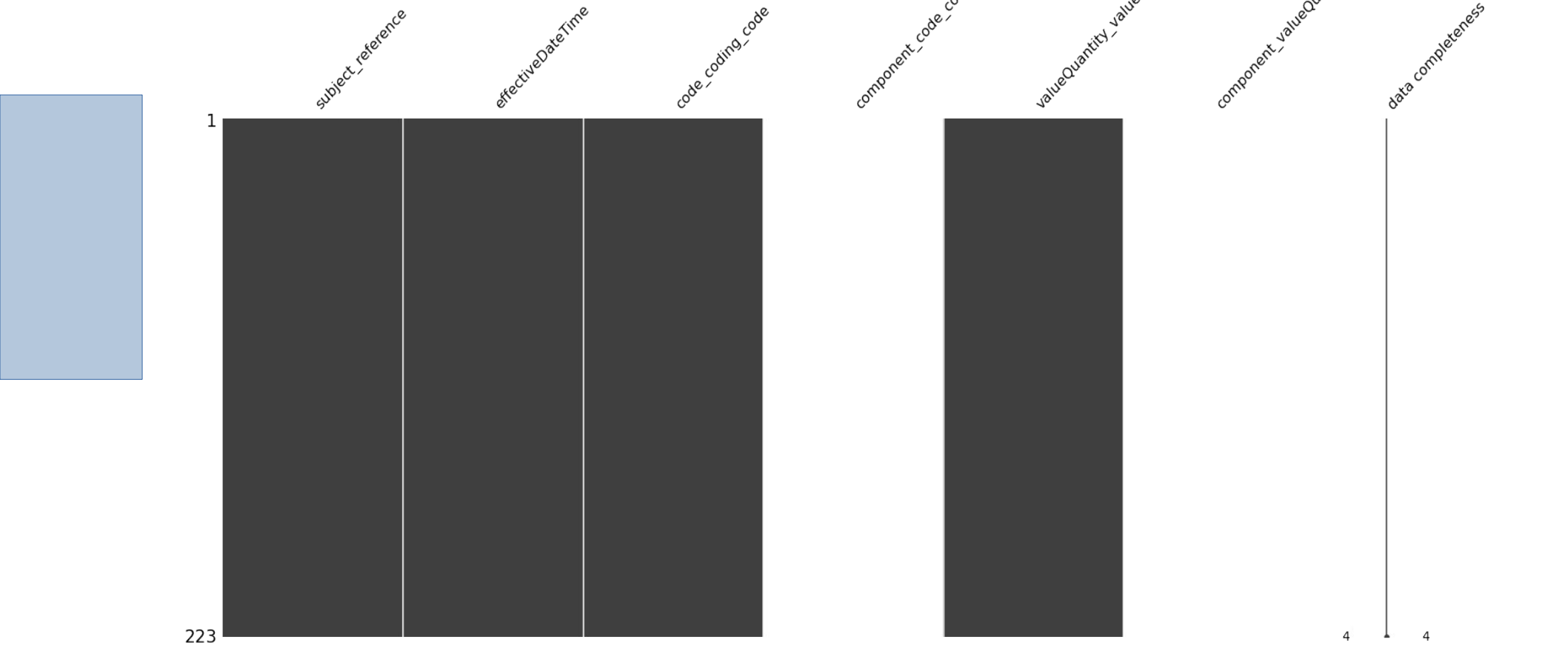
```
df = filter_observations('621892226','55425-3',None)
pd.DataFrame(df.toPandas()).head()
```

[414] ✓ 0.6s

...

	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

Filtering Matrix



Imputation 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	valueQuantit
104	Patient/621892226	2021-11-10 01:00:00	55425-3	None	
108	Patient/621892226	2021-11-11 01:00:00	55425-3	None	
107	Patient/621892226	2021-11-12 01:00:00	55425-3	None	
109	Patient/621892226	2021-11-13 01:00:00	55425-3	None	
106	Patient/621892226	2021-11-14 01:00:00	55425-3	None	
105	Patient/621892226	2021-11-15 01:00:00	55425-3	None	
115	Patient/621892226	2021-12-10 01:00:00	55425-3	None	
110	Patient/621892226	2021-12-10 01:00:00	55425-3	None	
114	Patient/621892226	2021-12-11 01:00:00	55425-3	None	
111	Patient/621892226	2021-12-11 01:00:00	55425-3	None	

Imputation Trunc Time

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

[578] ✓ 1.5s

	code_coding_code	new_effectiveDateTime	a
0	55425-3	2021-11-10	
1	55425-3	2021-11-11	
2	55425-3	2021-12-12	
3	55425-3	2021-11-13	
4	55425-3	2021-11-14	
5	55425-3	2021-11-15	
6	55425-3	2021-12-10	
7	55425-3	2021-12-11	
8	55425-3	2021-12-12	
9	55425-3	2021-12-13	

```
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

Imputation Fill Missing

```
df2 = fill_missing_time_values(df, TIME, [CODE]
pd.DataFrame(df2.toPandas()).head(10)
```

[612] ✓ 5.9s

...

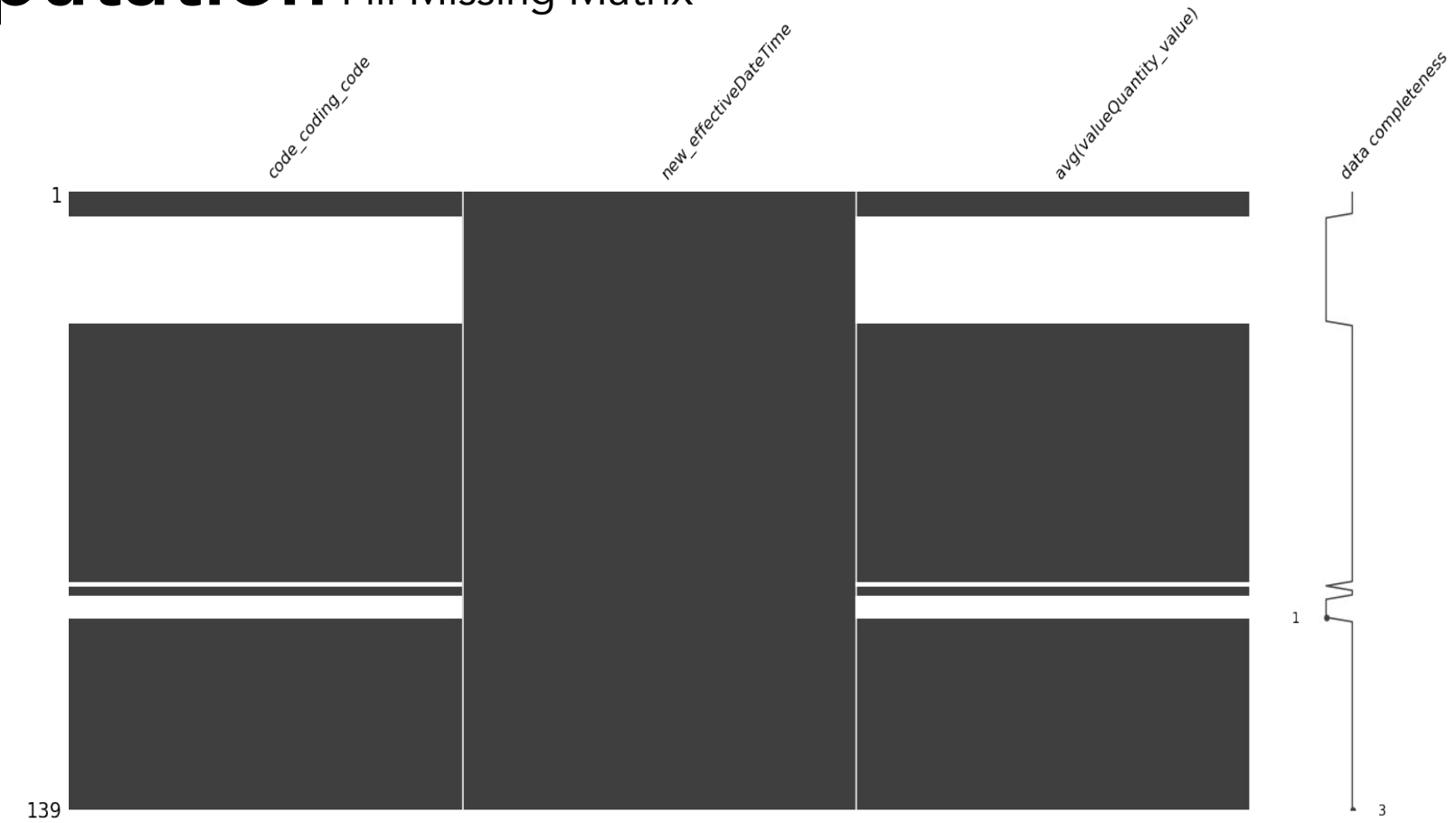
</>

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

```
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,
                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

Imputation Fill Missing Matrix



Imputation Echarts

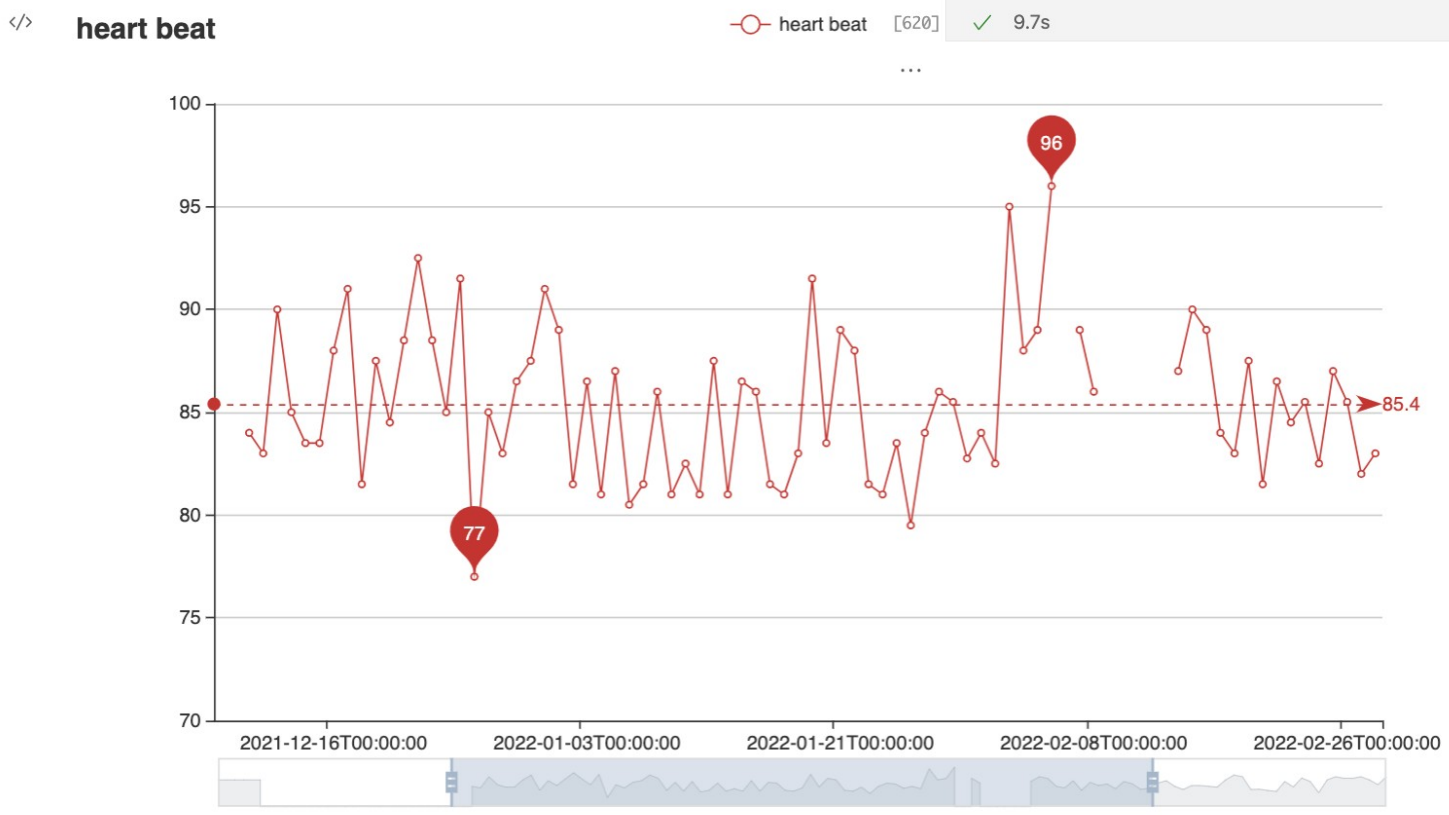
```
def heartbeat_graph(x, y, title):
    hb_line = Line()
    hb_line.add_xaxis(x)
    hb_line.add_yaxis(title, y,
        label_opts=opts.LabelOpts(is_show=False),
        markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_='average')]),
        markpoint_opts=opts.MarkPointOpts(data=[opts.MarkPointItem(type_='min'), 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

Imputation Visualization

```
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()
```

✓ 9.7s

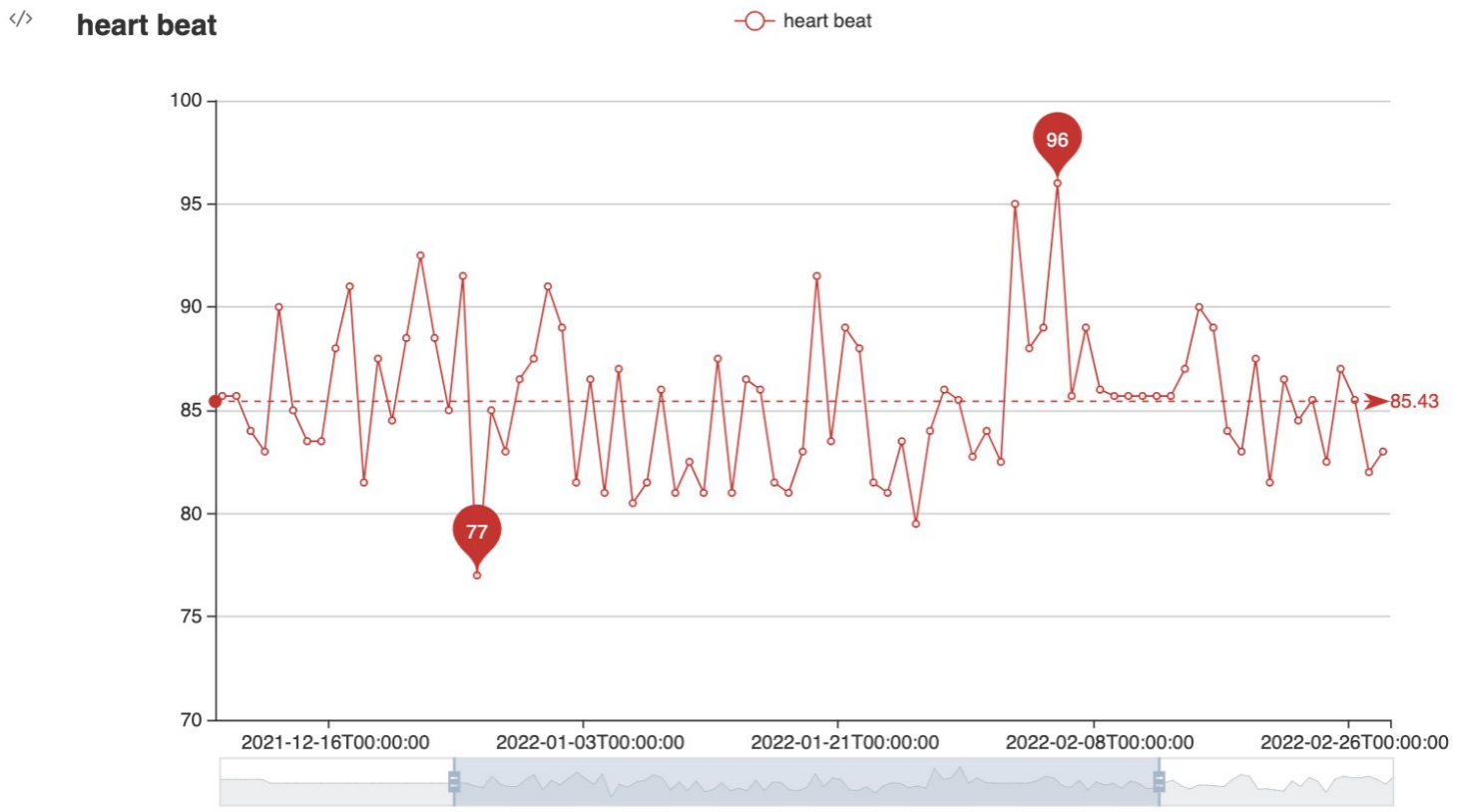


Imputation Simple Imputer

```
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

Imputation Visualization



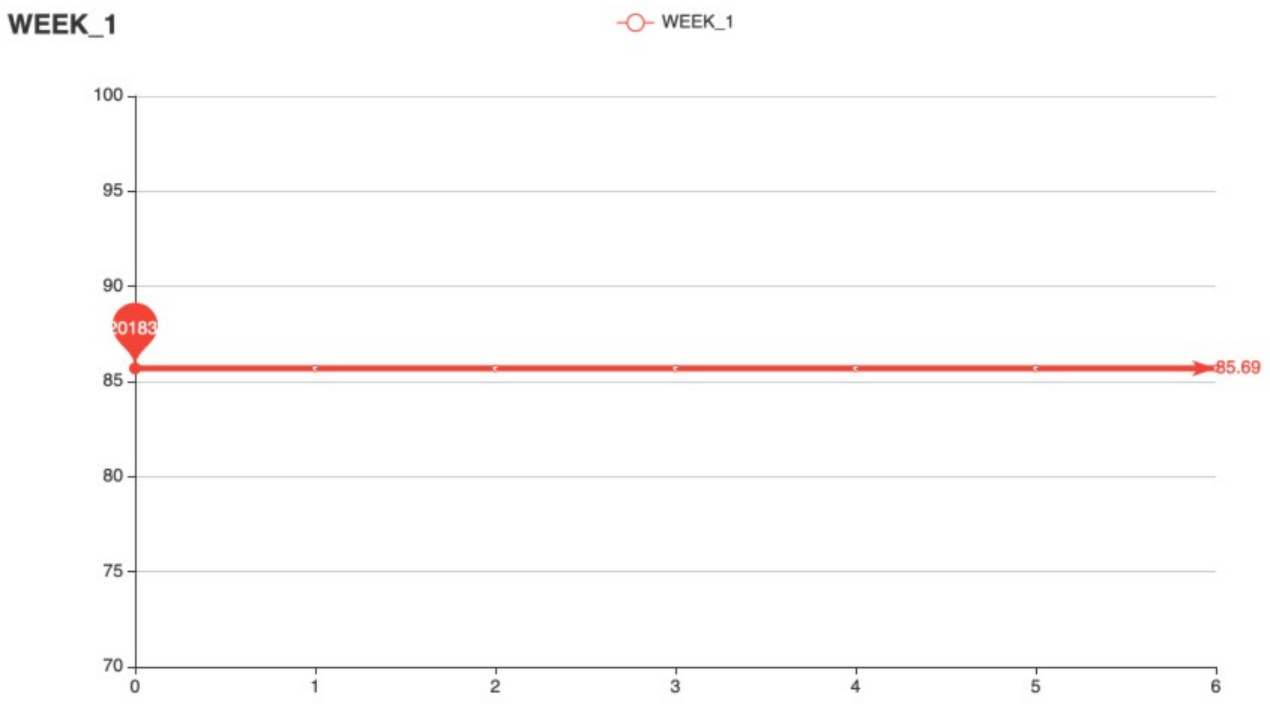
Imputation Trend GIF

```
import os
import imageio
from PIL import Image

png_dir = 'renders/weekly/'
images = []
for file_name in sorted(os.listdir(png_dir)):
    prev = None
    if file_name.endswith('.png'):
        file_path = os.path.join(png_dir, file_name)
        images.append(imageio.imread(file_path))
        prev = file_path

imageio.mimsave('renders/weekly/HB_MOVE.gif', images)
print("Done")
```

[45] ✓ 1.3s

```
Done
```

Imputation GIF



Imputation History GIF

```
prev = df4.iloc[0:7]
prevs = [prev['avg('+VALUE+')']]
for i in range(7, len(df4), 7):
    week = df4.iloc[i:i+7]
    x = range(0, len(week))
    prevs.append(week['avg('+VALUE+')'])
    y = prevs
    bar = heartbeat_graph(x, y, 'WEEK_'+str(int(i/7)))
    bar.render('renders/weekly/sum/week_'+str(int(i/7))+'.html')
```

[81] ✓ 0.2s

Imputation History GIF