MDT Validation Notebook

Validated on Synthea +MDT population vs MEPS for Pediatric Asthma

In [29]:
```
import pandas as pd
import datetime as dt
import numpy as np
from scipy.stats import chi2_contingency
```

Grab medication RXCUI of interest

Grabs the MEPS product RXCUI lists for filtering of Synthea to medications of interest. Path to this will be MDT module - log - rxcui_ndc_df_output.csv

In [113…
```
rxcui_df = pd.read_csv(r"") # MDT produced medication list
rxcui_df = rxcui_df[['medication_product_name','medication_product_rxcui']].drop_duplicates()
rxcui_df['medication_product_rxcui'] = rxcui_df['medication_product_rxcui'].astype(int)
```

Read Synthea Population

Reads Synthea Medication file and filters on medications of interest

The path for this will be synthea -> output -> csv -> medications.csv

In [115…
```
col_list = ['START','PATIENT','CODE']

syn_med_df = pd.DataFrame(columns = ['START','PATIENT','CODE','medication_product_rxcui','medication_product_name'])

for x in pd.read_csv(r"", usecols=col_list, chunksize=100000):
    x['CODE'] = x['CODE'].astype(int)
    temp_df = x.merge(rxcui_df, how="inner", left_on='CODE', right_on='medication_product_rxcui')
    syn_med_df = syn_med_df.append(temp_df)
```

Synthea Patient Population Filtering

Reads and merges Synthea patient data to allow for patient management. The path for this will be synthea -> output -> csv -> patients.csv

This step can be skipped if not filtering by patient. For the pediatic use case we limited to patients who received medications when they were < 6 years of age

In [76]:
```
syn_pat_df = pd.read_csv(r"")
syn_pat_df = syn_pat_df.merge(syn_med_df, how='inner', left_on='Id', right_on='PATIENT')

syn_pat_df['START'] = pd.to_datetime(syn_pat_df['START']).dt.date
syn_pat_df['BIRTHDATE'] = pd.to_datetime(syn_pat_df['BIRTHDATE']).dt.date
syn_pat_df['age_in_days'] = (syn_pat_df['START'] - syn_pat_df['BIRTHDATE']).dt.days

syn_med_df = syn_pat_df[syn_pat_df['age_in_days'] < 2191]
```

Synthea distributions

Gets total patient counts and medication distributions from Synthea population

In [116…
```
syn_med_df = syn_med_df.groupby(['medication_product_name']).agg(patient_count=('CODE','count')).reset_index()
total_patients = syn_med_df['patient_count'].sum()
syn_med_df['percent'] = syn_med_df['patient_count']/total_patients
syn_med_df
```

Out[116…

	medication_product_name	patient_count	percent
0	120 ACTUAT fluticasone propionate 0.044 MG/ACT...	2378	0.341618

	medication_product_name	patient_count	percent
1	120 ACTUAT fluticasone propionate 0.11 MG/ACTU...	1070	0.153714
2	Breath-Actuated 120 ACTUAT beclomethasone dipr...	203	0.029162
3	budesonide 0.125 MG/ML Inhalation Suspension	977	0.140353
4	budesonide 0.125 MG/ML Inhalation Suspension [...	513	0.073696
5	budesonide 0.25 MG/ML Inhalation Suspension	1819	0.261313
6	budesonide 0.5 MG/ML Inhalation Suspension	1	0.000144

MEPS Expected

generates the expected MEPS patient counts for chi squared goodness of fit test

Path to file will be in you MDT module - log - validation_df.csv

In [108…
```
meps_df = pd.read_csv(r"")
meps_df = meps_df[meps_df['age'] == '0-5'][['medication_product_name','validation_percent_product_patients']]
meps_df['patient_count'] = meps_df['validation_percent_product_patients'] * total_patients
meps_df['patient_count'] = meps_df['patient_count'].round(0)
meps_df
```

Out[108…

	medication_product_name	validation_percent_product_patients	patient_count
0	120_Actuat_Fluticasone_Propionate_0_044_Mg_Act...	0.335052	2332.0
1	120_Actuat_Fluticasone_Propionate_0_11_Mg_Actu...	0.156948	1093.0
16	Budesonide_0_125_Mg_Ml_Inhalation_Suspension	0.140715	980.0
17	Budesonide_0_125_Mg_Ml_Inhalation_Suspension_P...	0.072027	501.0
18	Budesonide_0_25_Mg_Ml_Inhalation_Suspension	0.263781	1836.0
19	Breath_Actuated_120_Actuat_Beclomethasone_Dipr...	0.031000	216.0

Run Chi Squared

Runs chi squared test for two different populations Take the values for patient count from syn_med_df and meps_df for this.

Numbers used are for the pediatric asthma use case of Synthea +MDT vs MEPS

In [117…
```
obs = np.array([[203, 216],
                [977, 979],
                [513, 489],
                [1819, 1836],
                [1, 0],
                [2378, 2332],
                [1070, 1093]])


chi2, p, df, ob = chi2_contingency(obs)
print(f"""X2 = {chi2}
p-value = {p}
degrees of freedom = {df}
observatrions = {ob}""")
```

```
X2 = 2.7347252762386036
p-value = 0.8413287112519282
degrees of freedom = 6
observatrions = [[2.09741047e+02 2.09258953e+02]
 [9.79125270e+02 9.76874730e+02]
 [5.01576442e+02 5.00423558e+02]
 [1.82960269e+03 1.82539731e+03]
 [5.00575291e-01 4.99424709e-01]
 [2.35770962e+03 2.35229038e+03]
 [1.08274435e+03 1.08025565e+03]]
```