Drugpersistency2

July 26, 2022

0.1 Drug Persistency and Medical adherence

0.1.1 Group Details

Team name: FeatureTransformers Team members: - Name: Wangu Ndungu - Email: nwangu349@gmail.com - Country: Kenya - College/Company: Kenyatta University - Specialization: Data Science

- Name: Nikola Andrejić
- Email: nikola.ing.nl@gmail.com
- Country: Serbia
- College/Company: University of Niš
- Specialization: Data Science

0.1.2 PROBLEM STATEMENT

According to the World Health Organisation, only 50-70% of patients adhere properly to prescribed drugs during therapy. This is especially true among those with long term medication. This worrying statistic is caused by various factors, for example: patient's condition or disease, their socio-economic status, confusion by the schedule, forgetting, discontinuing because they feel better, just to name a few. Medical non-adherence can lead to devastating consequences on one's health, especially those with chronic illnesses.

The purpose of this project is to study trends among patients in a sample and build a model that'll classify a new patient as Persistent or Non-Persistent.

This project will give medical practitioners(especially pharmaceuticals) insight on which patients might require more rigorous follow-ups to ensure they'll adhere to their prescriptions.

[98]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew
%matplotlib inline
```

0.1.3 1. Read in and inspect the data

[93]:
```
#Read in the data
data = pd.ExcelFile("Data/Healthcare_dataset.xlsx")
data.sheet_names
```

[93]:
```
['Feature Description', 'Dataset']
```

[94]:
```
#Parse the dataset from data
df = data.parse(sheet_name="Dataset")
```

[95]:
```
#See the head
df.head()
```

[95]:
```
  Ptid Persistency_Flag  Gender           Race     Ethnicity   Region  \
0   P1       Persistent    Male      Caucasian  Not Hispanic     West
1   P2   Non-Persistent    Male          Asian  Not Hispanic     West
2   P3   Non-Persistent  Female  Other/Unknown      Hispanic  Midwest
3   P4   Non-Persistent  Female      Caucasian  Not Hispanic  Midwest
4   P5   Non-Persistent  Female      Caucasian  Not Hispanic  Midwest

  Age_Bucket        Ntm_Speciality Ntm_Specialist_Flag  \
0        >75  GENERAL PRACTITIONER              Others
1      55-65  GENERAL PRACTITIONER              Others
2      65-75  GENERAL PRACTITIONER              Others
3        >75  GENERAL PRACTITIONER              Others
4        >75  GENERAL PRACTITIONER              Others

      Ntm_Speciality_Bucket  … Risk_Family_History_Of_Osteoporosis  \
0  OB/GYN/Others/PCP/Unknown  …                                   N
1  OB/GYN/Others/PCP/Unknown  …                                   N
2  OB/GYN/Others/PCP/Unknown  …                                   N
3  OB/GYN/Others/PCP/Unknown  …                                   N
4  OB/GYN/Others/PCP/Unknown  …                                   N

  Risk_Low_Calcium_Intake  Risk_Vitamin_D_Insufficiency  \
0                       N                             N
1                       N                             N
2                       Y                             N
3                       N                             N
4                       N                             N

  Risk_Poor_Health_Frailty Risk_Excessive_Thinness  \
0                        N                       N
1                        N                       N
2                        N                       N
3                        N                       N
4                        N                       N
```

```
  Risk_Hysterectomy_Oophorectomy Risk_Estrogen_Deficiency Risk_Immobilization  \
0                              N                        N                   N
1                              N                        N                   N
2                              N                        N                   N
3                              N                        N                   N
4                              N                        N                   N

  Risk_Recurring_Falls Count_Of_Risks
0                    N              0
1                    N              0
2                    N              2
3                    N              1
4                    N              1

[5 rows x 69 columns]
```

[97]:
```
#See the info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3424 entries, 0 to 3423
Data columns (total 69 columns):
 #   Column                                                              Non-
Null Count  Dtype
---  ------
--------------  -----
 0   Ptid                                                                3424
non-null   object
 1   Persistency_Flag                                                    3424
non-null   object
 2   Gender                                                              3424
non-null   object
 3   Race                                                                3424
non-null   object
 4   Ethnicity                                                           3424
non-null   object
 5   Region                                                              3424
non-null   object
 6   Age_Bucket                                                          3424
non-null   object
 7   Ntm_Speciality                                                      3424
non-null   object
 8   Ntm_Specialist_Flag                                                 3424
non-null   object
 9   Ntm_Speciality_Bucket                                               3424
non-null   object
 10  Gluco_Record_Prior_Ntm                                              3424
```

```
non-null   object
 11  Gluco_Record_During_Rx                                            3424
non-null   object
 12  Dexa_Freq_During_Rx                                               3424
non-null   int64
 13  Dexa_During_Rx                                                    3424
non-null   object
 14  Frag_Frac_Prior_Ntm                                               3424
non-null   object
 15  Frag_Frac_During_Rx                                               3424
non-null   object
 16  Risk_Segment_Prior_Ntm                                            3424
non-null   object
 17  Tscore_Bucket_Prior_Ntm                                           3424
non-null   object
 18  Risk_Segment_During_Rx                                            3424
non-null   object
 19  Tscore_Bucket_During_Rx                                           3424
non-null   object
 20  Change_T_Score                                                    3424
non-null   object
 21  Change_Risk_Segment                                               3424
non-null   object
 22  Adherent_Flag                                                     3424
non-null   object
 23  Idn_Indicator                                                     3424
non-null   object
 24  Injectable_Experience_During_Rx                                   3424
non-null   object
 25  Comorb_Encounter_For_Screening_For_Malignant_Neoplasms            3424
non-null   object
 26  Comorb_Encounter_For_Immunization                                 3424
non-null   object
 27  Comorb_Encntr_For_General_Exam_W_O_Complaint,_Susp_Or_Reprtd_Dx     3424
non-null   object
 28  Comorb_Vitamin_D_Deficiency                                       3424
non-null   object
 29  Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified              3424
non-null   object
 30  Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx  3424
non-null   object
 31  Comorb_Long_Term_Current_Drug_Therapy                             3424
non-null   object
 32  Comorb_Dorsalgia                                                  3424
non-null   object
 33  Comorb_Personal_History_Of_Other_Diseases_And_Conditions          3424
non-null   object
 34  Comorb_Other_Disorders_Of_Bone_Density_And_Structure              3424
```

```
non-null   object
 35  Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias     3424
non-null   object
 36  Comorb_Osteoporosis_without_current_pathological_fracture           3424
non-null   object
 37  Comorb_Personal_history_of_malignant_neoplasm                       3424
non-null   object
 38  Comorb_Gastro_esophageal_reflux_disease                             3424
non-null   object
 39  Concom_Cholesterol_And_Triglyceride_Regulating_Preparations         3424
non-null   object
 40  Concom_Narcotics                                                    3424
non-null   object
 41  Concom_Systemic_Corticosteroids_Plain                               3424
non-null   object
 42  Concom_Anti_Depressants_And_Mood_Stabilisers                        3424
non-null   object
 43  Concom_Fluoroquinolones                                             3424
non-null   object
 44  Concom_Cephalosporins                                               3424
non-null   object
 45  Concom_Macrolides_And_Similar_Types                                 3424
non-null   object
 46  Concom_Broad_Spectrum_Penicillins                                   3424
non-null   object
 47  Concom_Anaesthetics_General                                         3424
non-null   object
 48  Concom_Viral_Vaccines                                               3424
non-null   object
 49  Risk_Type_1_Insulin_Dependent_Diabetes                              3424
non-null   object
 50  Risk_Osteogenesis_Imperfecta                                        3424
non-null   object
 51  Risk_Rheumatoid_Arthritis                                           3424
non-null   object
 52  Risk_Untreated_Chronic_Hyperthyroidism                              3424
non-null   object
 53  Risk_Untreated_Chronic_Hypogonadism                                 3424
non-null   object
 54  Risk_Untreated_Early_Menopause                                      3424
non-null   object
 55  Risk_Patient_Parent_Fractured_Their_Hip                             3424
non-null   object
 56  Risk_Smoking_Tobacco                                                3424
non-null   object
 57  Risk_Chronic_Malnutrition_Or_Malabsorption                          3424
non-null   object
 58  Risk_Chronic_Liver_Disease                                          3424
```

```
non-null   object
 59  Risk_Family_History_Of_Osteoporosis                                3424
non-null   object
 60  Risk_Low_Calcium_Intake                                            3424
non-null   object
 61  Risk_Vitamin_D_Insufficiency                                       3424
non-null   object
 62  Risk_Poor_Health_Frailty                                           3424
non-null   object
 63  Risk_Excessive_Thinness                                            3424
non-null   object
 64  Risk_Hysterectomy_Oophorectomy                                     3424
non-null   object
 65  Risk_Estrogen_Deficiency                                           3424
non-null   object
 66  Risk_Immobilization                                                3424
non-null   object
 67  Risk_Recurring_Falls                                               3424
non-null   object
 68  Count_Of_Risks                                                     3424
non-null   int64
dtypes: int64(2), object(67)
memory usage: 1.8+ MB
```

[100]:
```python
#Parse the feature descriptions from the excel file
feature_description = data.parse("Feature Description",index_col=[0,1])
feature_description
```

[100]:
```
           Variable Description
Bucket                   Variable
Unique Row Id            Patient ID
Unique ID of each patient
Target Variable          Persistency_Flag                     Flag indicating if
a patient was persistent or…
Demographics             Age                                            Age of
the patient during their therapy
                         Race                                      Race of the
patient from the patient table
                         Region                                  Region of the
patient from the patient table
                         Ethnicity                            Ethnicity of the
patient from the patient table
                         Gender                                  Gender of the
patient from the patient table
                         IDN Indicator                                    Flag
indicating patients mapped to IDN
Provider Attributes      NTM - Physician Specialty            Specialty of the
```

```
HCP that prescribed the NTM Rx
Clinical Factors         NTM - T-Score                        T Score of the
patient at the time of the NTM …
                         Change in T Score                    Change in Tscore
before starting with any ther…
                         NTM - Risk Segment                   Risk Segment of
the patient at the time of the…
                         Change in Risk Segment               Change in Risk
Segment before starting with an…
                         NTM - Multiple Risk Factors          Flag indicating if
patient falls under multip…
                         NTM - Dexa Scan Frequency            Number of DEXA
scans taken prior to the first …
                         NTM - Dexa Scan Recency              Flag indicating
the presence of Dexa Scan befo…
                         Dexa During Therapy                  Flag indicating if
the patient had a Dexa Scan…
                         NTM - Fragility Fracture Recency     Flag indicating if
the patient had a recent fr…
                         Fragility Fracture During Therapy    Flag indicating if
the patient had fragility f…
                         NTM - Glucocorticoid Recency         Flag indicating
usage of Glucocorticoids (>=7…
                         Glucocorticoid Usage During Therapy  Flag indicating if
the patient had a Glucocort…
Disease/Treatment Factor NTM - Injectable Experience          Flag indicating
any injectable drug usage in t…
                         NTM - Risk Factors                   Risk Factors that
the patient is falling into…
                         NTM - Comorbidity                    Comorbidities are
divided into two main catego…
                         NTM - Concomitancy                   Concomitant drugs
recorded prior to starting w…
                         Adherence
Adherence for the therapies
```

0.1.4 2. Check for missing values

[43]:
```
#Check for null values
df.isna().sum()
```

[43]:
```
Ptid                                  0
Persistency_Flag                      0
Gender                                0
Race                                  0
Ethnicity                             0
                                     ..
```

```
Risk_Hysterectomy_Oophorectomy    0
Risk_Estrogen_Deficiency          0
Risk_Immobilization               0
Risk_Recurring_Falls              0
Count_Of_Risks                    0
Length: 69, dtype: int64
```

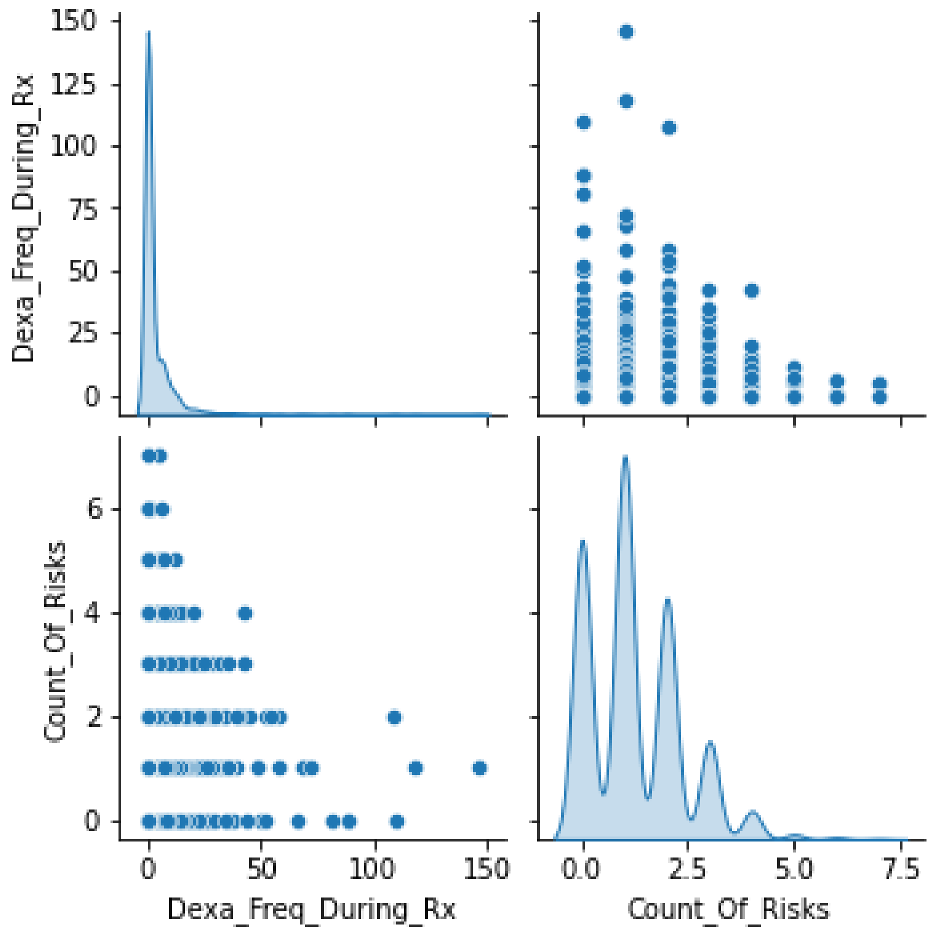
[44]:
```python
df.isna().sum().max()
#There are no missing values
```

[44]:
```
0
```

0.1.5 3. Check for outliers

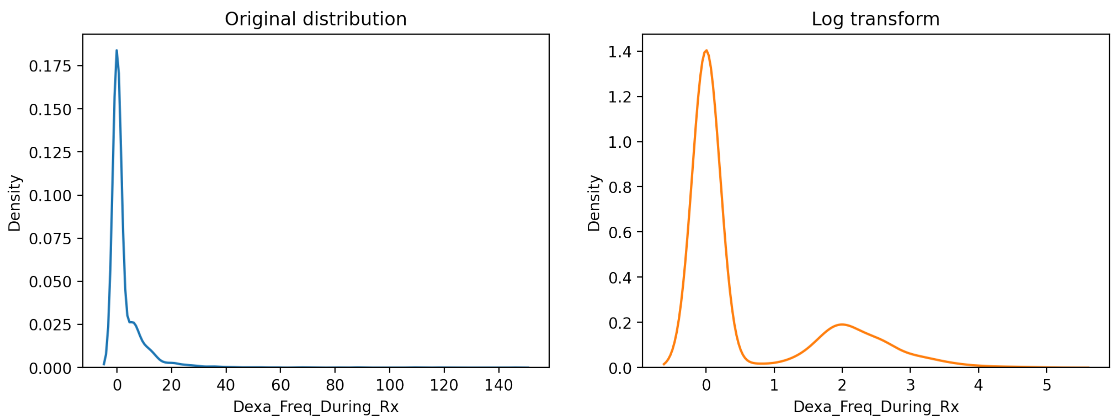
[142]:
```python
#Define the function that calculates upper and lower bounds for outliers based␣
↪on IQR
def get_outliers(series):
    q1 = series.quantile(0.25)
    q3 = series.quantile(0.75)

    iqr = q3-q1

    ub = q1-1.5*iqr
    lb = q3+1.5*iqr

    return series[(series<ub) | (series>lb)]
```

[128]:
```python
#See the pairplot
sns.pairplot(df,diag_kind="kde");
```



3.1. The "Dexa_Freq_During_Rx" column

[129]:
```
#Compare the original distribution and the log transform of␣
↪"Dexa_Freq_During_Rx"
fig,axes=plt.subplots(nrows=1,ncols=2)
fig.set_size_inches((12,4))
fig.set_dpi(200)
sns.kdeplot(x=df["Dexa_Freq_During_Rx"],ax=axes[0],color="tab:blue")
sns.kdeplot(x=np.log(df["Dexa_Freq_During_Rx"]+1),ax=axes[1],color="tab:
↪orange");
axes[0].set_title("Original distribution")
axes[1].set_title("Log transform");
```



[130]:
```
#Skewness of the original distribution
skew(df['Dexa_Freq_During_Rx'])
```

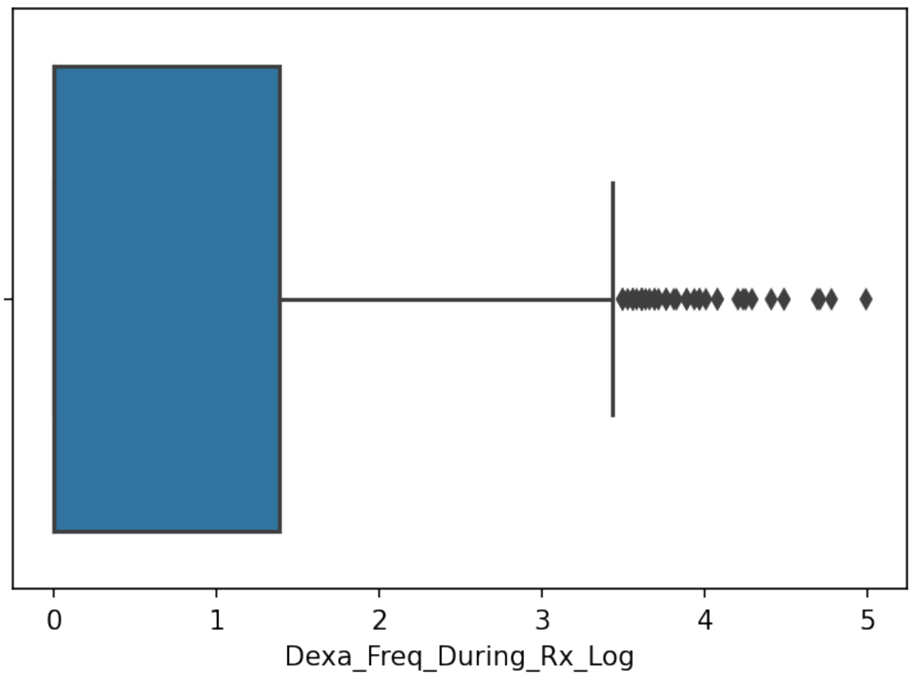
[130]: 6.805747051718919

Notice that the probability density over the column “Dexa_Freq_During_Rx” is exponential-like fat tailed distribution. Also, this distribution has large positive skewness. Therefore, it is suitable to apply the log transform to this column. Notice that the mininum value in this column is zero so to avoid the divergence problems we will apply the transformation log(x+1). The two distributions are compared side by side in the figure above.

[145]:
```
#Apply the log transform
df["Dexa_Freq_During_Rx_Log"] = df['Dexa_Freq_During_Rx'].apply(lambda x: np.
    ↪log(1+x))
```

[148]:
```
#Drop the original column
df=df.drop("Dexa_Freq_During_Rx",axis=1)
```

[149]:
```
#We check for outliers using the boxplot
plt.figure(figsize=(6,4),dpi=150)
sns.boxplot(x=df["Dexa_Freq_During_Rx_Log"]);
```




```
[150]: get_outliers(df["Dexa_Freq_During_Rx_Log"])
```

```
[150]: 241     3.637586
       541     4.406719
       651     3.761200
       1370    3.761200
       1398    3.828641
       1734    4.077537
       1838    3.526361
       1854    4.077537
       1901    4.709530
       1909    3.610918
       1920    3.555348
       1993    4.488636
       2013    4.204693
       2024    3.496508
       2028    4.488636
       2033    4.779123
       2044    3.891820
       2065    3.610918
       2132    4.248495
       2134    3.663562
       2151    3.713572
       2168    3.496508
```

```
2176    4.234107
2197    3.970292
2205    3.931826
2215    4.990433
2275    3.555348
2278    3.970292
2298    3.891820
2314    3.806662
2379    3.496508
2393    3.761200
2503    3.583519
2557    3.688879
2558    3.610918
2603    4.691348
2608    3.610918
2681    4.007333
2686    3.688879
2713    3.555348
2751    4.290459
2799    3.610918
Name: Dexa_Freq_During_Rx_Log, dtype: float64
```

[155]:
```
#Calculate what percentage of the data are classified as outliers in this column
print(f'{(len(get_outliers(df["Dexa_Freq_During_Rx_Log"]))/len(df))*100} %')
```

```
1.2266355140186915 %
```

This is a small percentage of the data so we can drop these rows.

[156]:
```
df = df.drop(index = get_outliers(df["Dexa_Freq_During_Rx_Log"]).index)
```

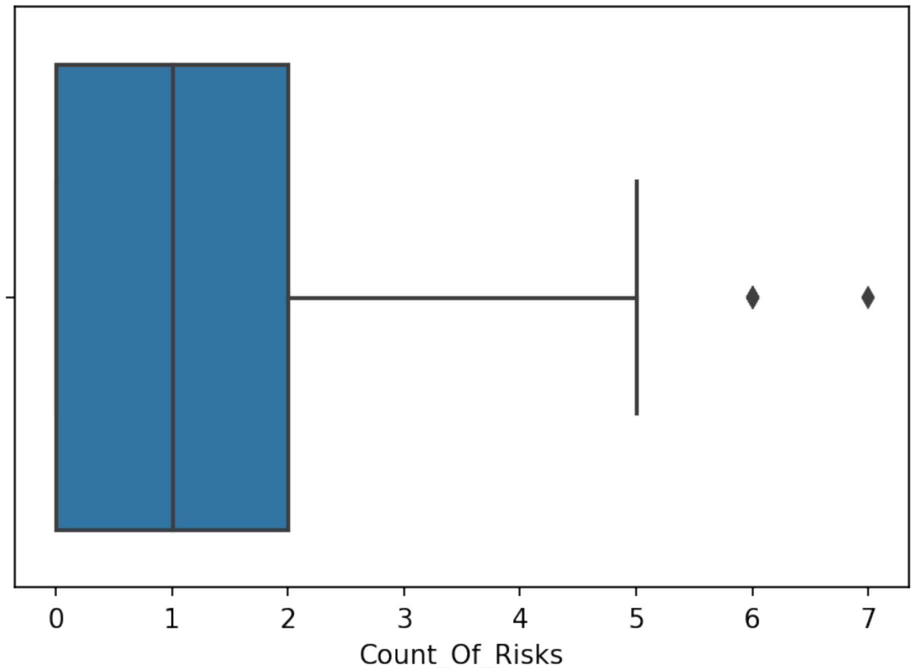
3.2. The "Count_Of_Risks" column

[161]:
```
skew(df["Count_Of_Risks"])
```

[161]:
```
0.734052404983274
```

The distribution has a positive skewness.

[159]:
```
#We check for outliers using the boxplot
plt.figure(figsize=(6,4),dpi=150)
sns.boxplot(x=df["Count_Of_Risks"]);
```

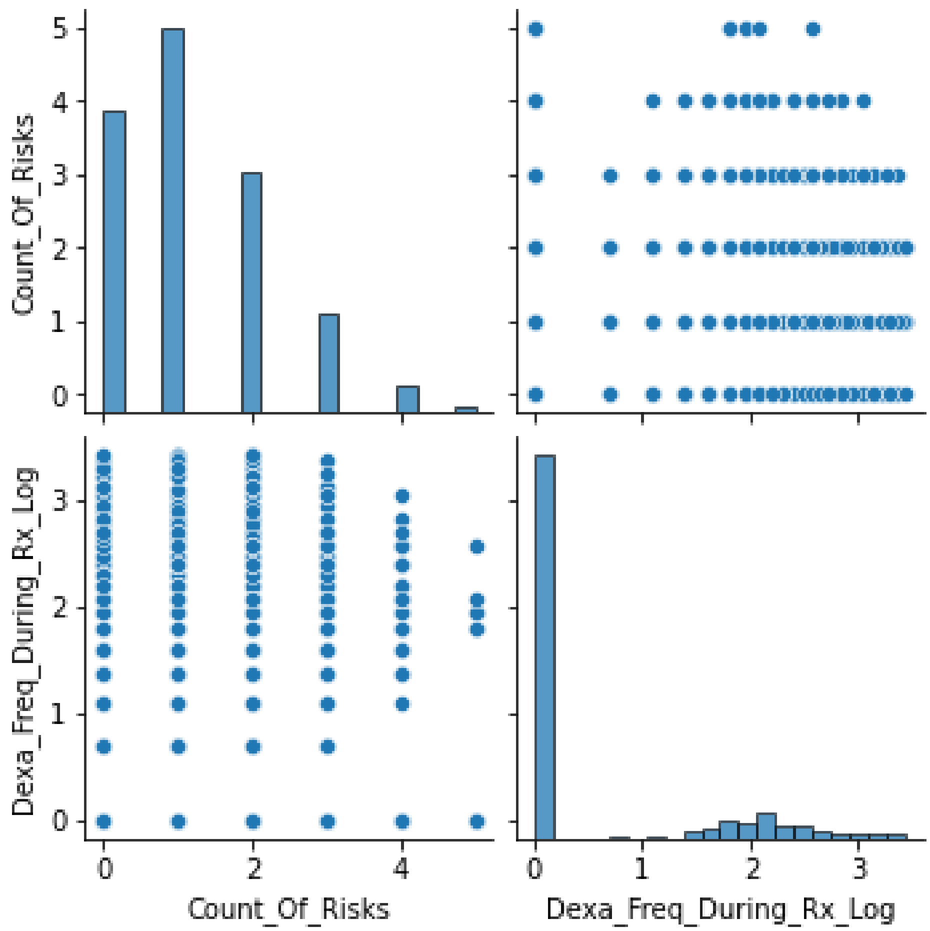


Only two points are classified as outliers so we can drop them.

[160]:
```
df = df.drop(index = get_outliers(df["Count_Of_Risks"]).index)
```

3.3. Check the pairplot once again

[163]:
```
sns.pairplot(df);
```



0.1.6 4. Categorical colums

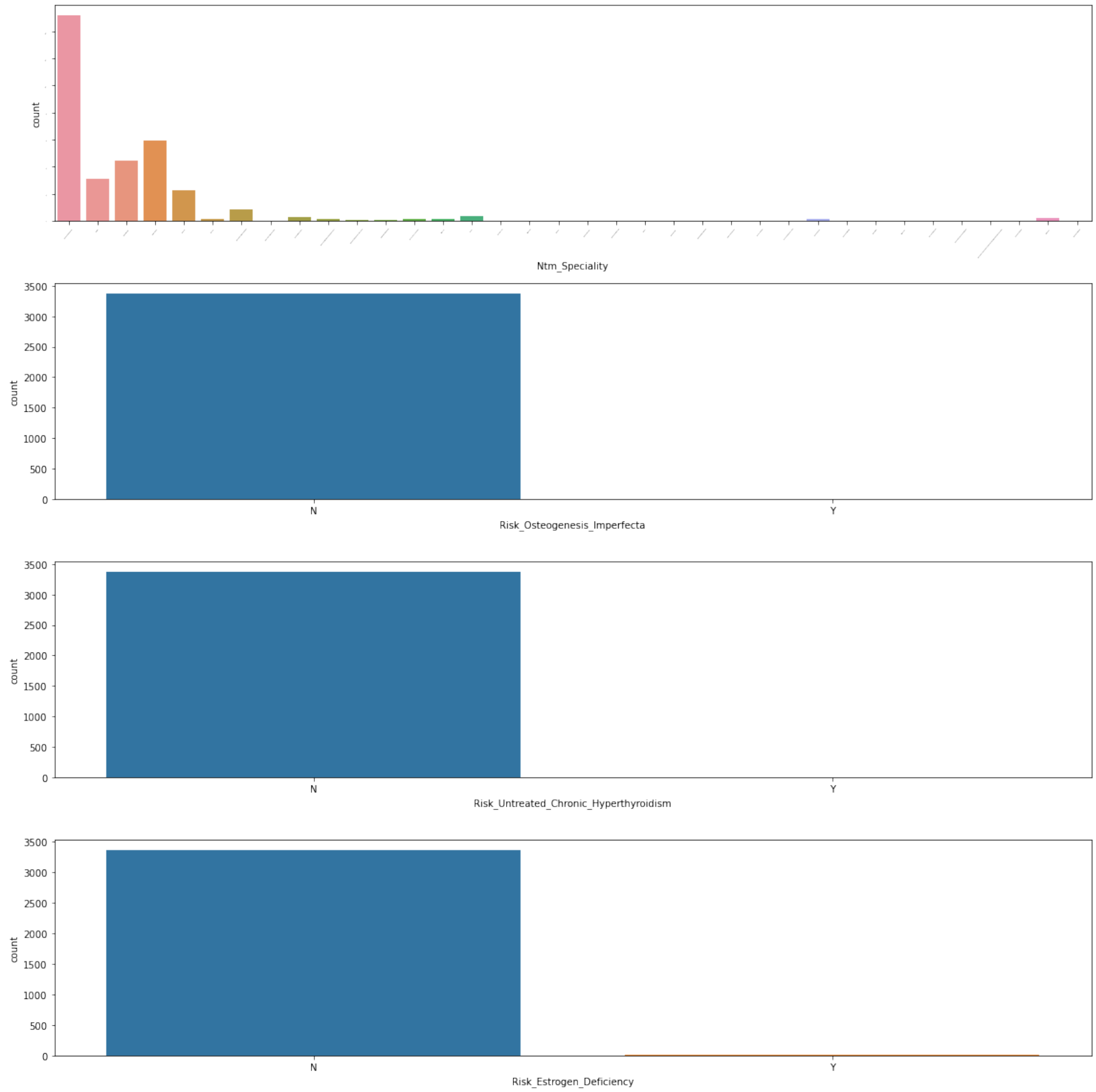
Let's check for imbalances in the categorical columns

Categorical columns can be highly imbalanced. If we take a cutoff at about 0.3% of the lenght of the df we can classify as outliers any categories that have 10 or less datapoints contained in them. If the column with outliers is binary we will drop both the column and the datapoints and if the column has more than two categories we will drop only the datapoints.

[216]:
```
#Create a list of categorical columns
cat_cols=df.select_dtypes("object").drop("Ptid",axis=1).columns
#Create a list of categorical columns with outliers
cat_cols_outliers = cat_cols[[any(df[col].value_counts()<=10) for col in␣
↪cat_cols]]
```

[215]:
```
#Visualize the imbalance of categorical columns with outliers
fig,axes=plt.subplots(nrows=len(cat_cols_outliers))
fig.set_size_inches((16,4*4))
```

```
i=0
for col in cat_cols_outliers:
    sns.countplot(x=df[col],ax=axes[i])
    i+=1
axes[0].tick_params(rotation=50,labelsize=0)
plt.tight_layout()
```



[230]:
```
#Define the function that will get us the outliers
def get_outliers_cat(series,threshold):
    val_counts=series.value_counts()
    outlier_cat = val_counts[val_counts<=threshold].index

    if len(outlier_cat)>0:
```

```
        return pd.concat([series[series==cat] for cat in outlier_cat])
    else:
        return pd.Series([],dtype="object")
```

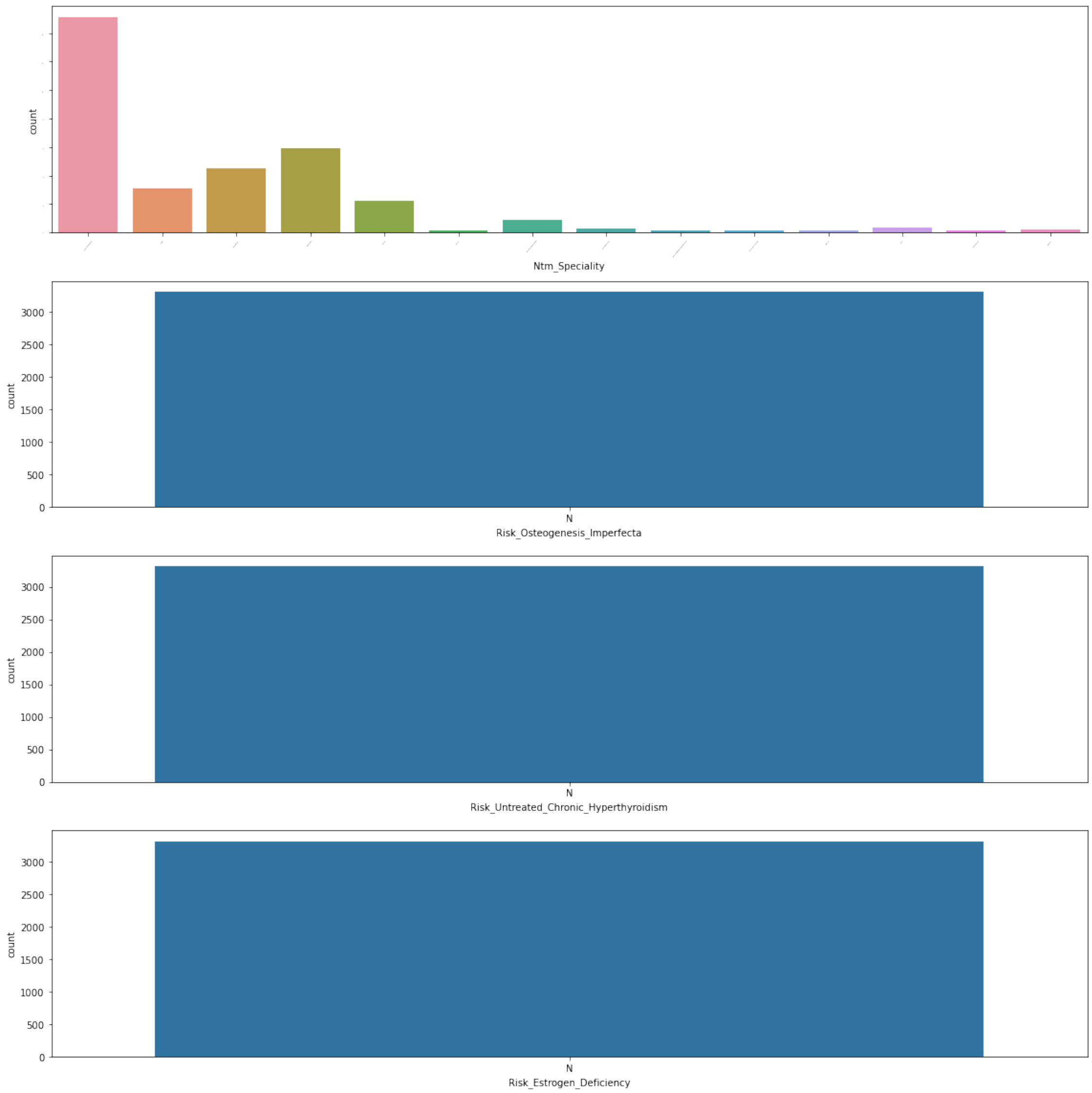
[232]:
```
get_outliers_cat(df["Risk_Estrogen_Deficiency"],10)
```

[232]:
```
242     Y
249     Y
754     Y
1220    Y
1660    Y
1669    Y
1720    Y
1727    Y
2047    Y
2956    Y
Name: Risk_Estrogen_Deficiency, dtype: object
```

[238]:
```
#Create the drop index
drop_ind = pd.concat([get_outliers_cat(df[col],10) for col in␣
↪cat_cols_outliers]).index.unique()
```

[240]:
```
#Drop the datapoints
df = df.drop(index=drop_ind)
```

[241]:
```
#Visualize the imbalance of categorical columns that previously contained the␣
↪outliers
fig,axes=plt.subplots(nrows=len(cat_cols_outliers))
fig.set_size_inches((16,4*4))
i=0
for col in cat_cols_outliers:
    sns.countplot(x=df[col],ax=axes[i])
    i+=1
axes[0].tick_params(rotation=50,labelsize=0)
plt.tight_layout()
```



[246]:
```
#Drop the colums with only one category left
df = df.drop(columns=cat_cols_outliers.difference(["Ntm_Speciality"]))
```

[249]:
```
((3424-len(df))/3424)*100
```

[249]: 3.212616822429906

We have dropped a little bit more than 3% of the data in the process.