In [118…
```
pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages
(0.0.6)
```

In [119…
```
from ucimlrepo import fetch_ucirepo
import pandas as pd
import numpy as np


# fetch dataset
cervical_cancer_risk_factors = fetch_ucirepo(id=383)

# data (as pandas dataframes)
X = cervical_cancer_risk_factors.data.features
y = cervical_cancer_risk_factors.data.targets

# metadata
#print(cervical_cancer_risk_factors.metadata)

# variable information
#print(cervical_cancer_risk_factors.variables)

df = pd.concat([X,y], axis=1)
```

In [120…
```
df
```

Out[120…

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormon Contraceptiv
0	18	4.0	15.0	1.0	0.0	0.0	0.0	0
1	15	1.0	14.0	1.0	0.0	0.0	0.0	0
2	34	1.0	NaN	1.0	0.0	0.0	0.0	0
3	52	5.0	16.0	4.0	1.0	37.0	37.0	1
4	46	3.0	21.0	4.0	0.0	0.0	0.0	1
...	...	...	...	...	...	...	...	
853	34	3.0	18.0	0.0	0.0	0.0	0.0	0
854	32	2.0	19.0	1.0	0.0	0.0	0.0	1
855	25	2.0	17.0	0.0	0.0	0.0	0.0	1
856	33	2.0	24.0	2.0	0.0	0.0	0.0	1
857	29	2.0	20.0	1.0	0.0	0.0	0.0	1

858 rows × 36 columns

Data Cleaning

In [121…

```
# Filtering out the missing values
df.fillna(df.select_dtypes(np.number).mean(), inplace=True)
```

In [122…

```
df.isnull().sum()
```

Out[122…
```
Age                                   0
Number of sexual partners             0
First sexual intercourse              0
Num of pregnancies                    0
Smokes                                0
Smokes (years)                        0
Smokes (packs/year)                   0
Hormonal Contraceptives               0
Hormonal Contraceptives (years)       0
IUD                                   0
IUD (years)                           0
STDs                                  0
STDs (number)                         0
STDs:condylomatosis                   0
STDs:cervical condylomatosis          0
STDs:vaginal condylomatosis           0
STDs:vulvo-perineal condylomatosis    0
STDs:syphilis                         0
STDs:pelvic inflammatory disease      0
STDs:genital herpes                   0
STDs:molluscum contagiosum            0
STDs:AIDS                             0
STDs:HIV                              0
STDs:Hepatitis B                      0
STDs:HPV                              0
STDs: Number of diagnosis             0
STDs: Time since first diagnosis      0
STDs: Time since last diagnosis       0
Dx:Cancer                             0
Dx:CIN                                0
Dx:HPV                                0
Dx                                    0
Hinselmann                            0
Schiller                              0
Citology                              0
Biopsy                                0
dtype: int64
```

In [177…
```
cervical_df = df.drop(columns=['Age','STDs:cervical condylomatosis','STDs:AIDS','Sm
```

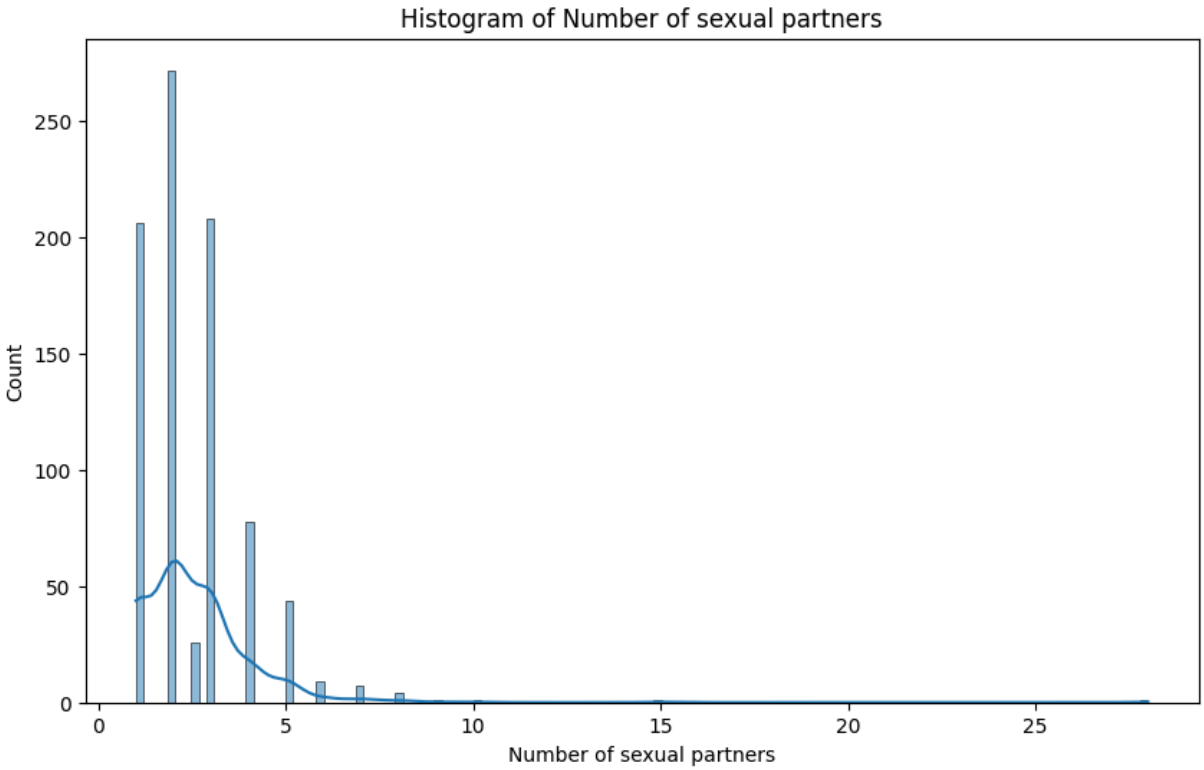
In [178…
```
cervical_df.columns
```

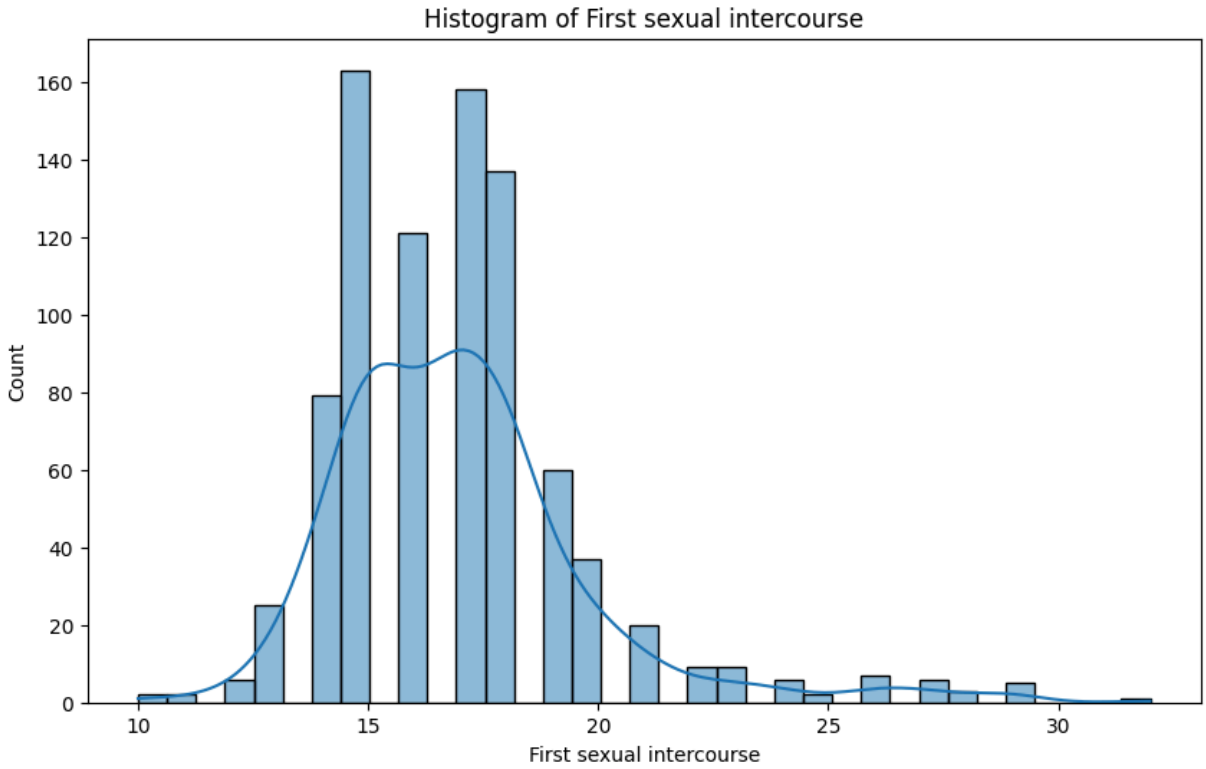
Out[178…
```
Index(['Number of sexual partners', 'First sexual intercourse',
       'Num of pregnancies', 'Smokes', 'Hormonal Contraceptives', 'IUD',
       'STDs', 'STDs:condylomatosis', 'STDs:vaginal condylomatosis',
       'STDs:vulvo-perineal condylomatosis', 'STDs:syphilis',
       'STDs:pelvic inflammatory disease', 'STDs:genital herpes',
       'STDs:molluscum contagiosum', 'STDs:HIV', 'STDs:Hepatitis B',
       'STDs:HPV', 'STDs: Number of diagnosis',
       'STDs: Time since first diagnosis', 'STDs: Time since last diagnosis',
       'Dx:Cancer', 'Dx:CIN', 'Dx:HPV', 'Dx', 'Hinselmann', 'Schiller',
       'Citology', 'Biopsy'],
      dtype='object')
```

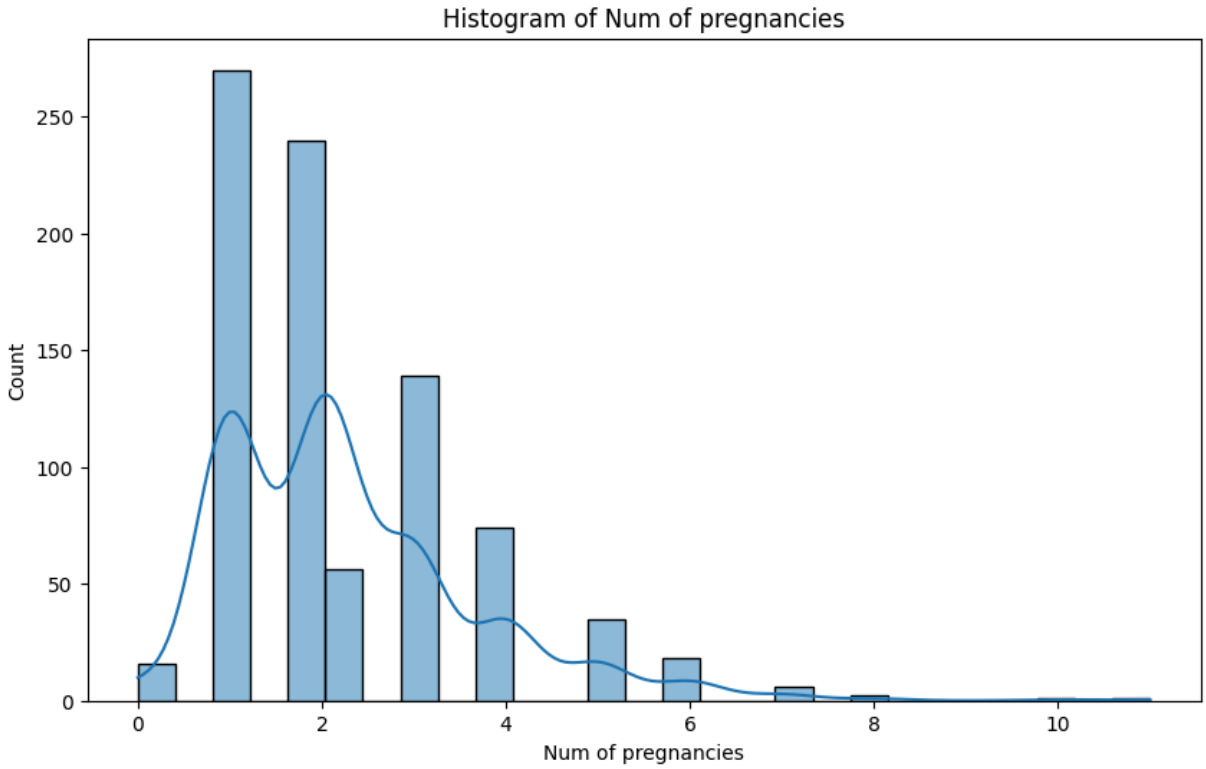
Exploratory Data Analysis

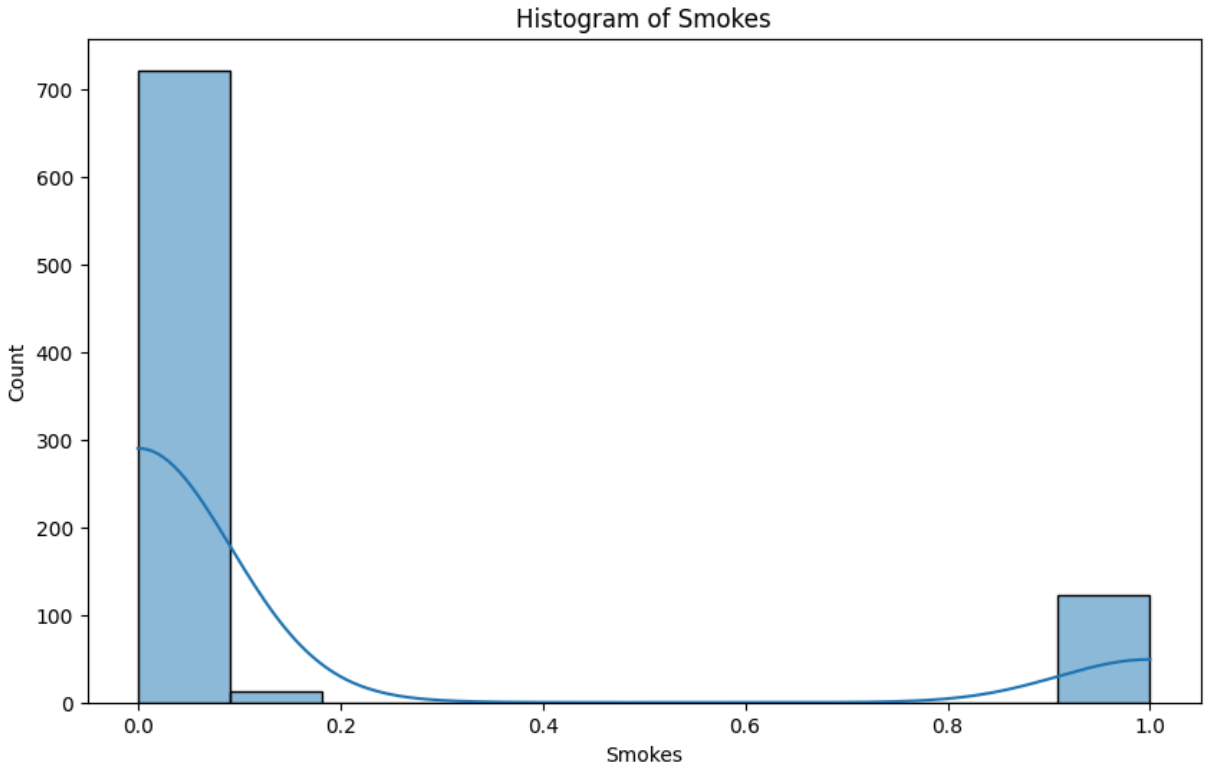
In [221…

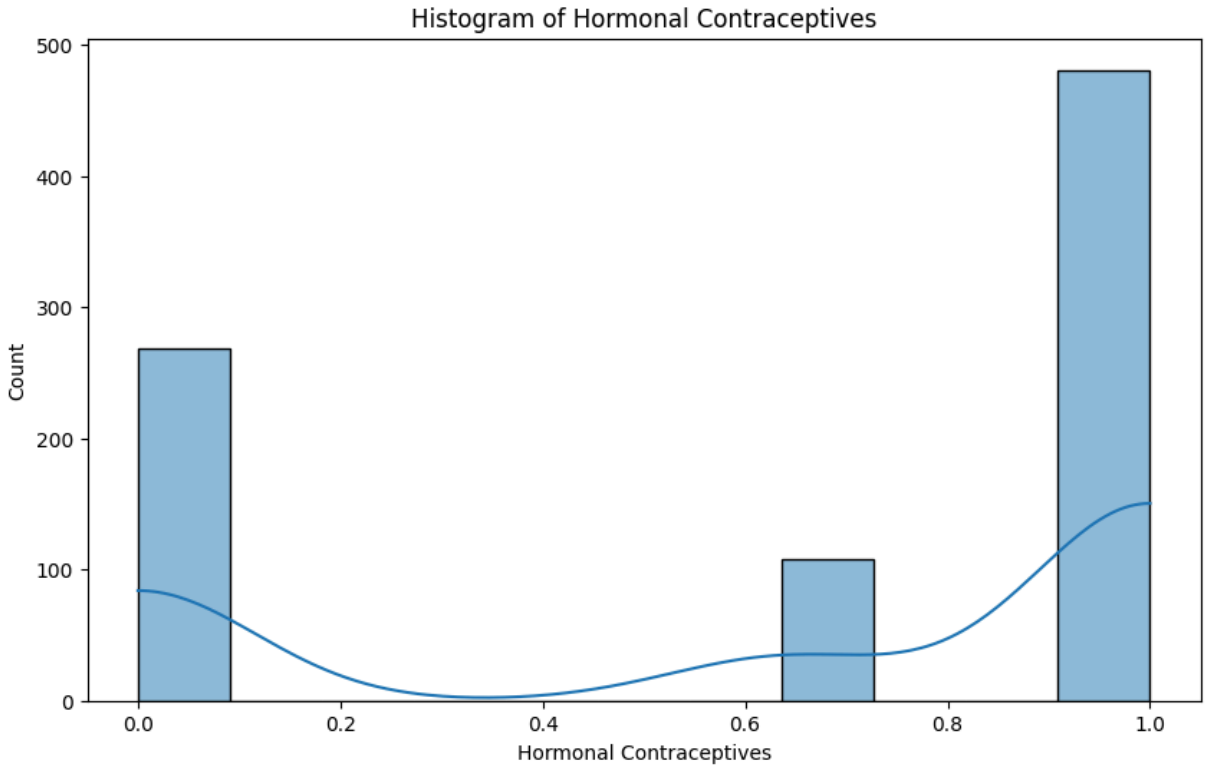
```
import matplotlib.pyplot as plt
import seaborn as sns

numerical_columns = cervical_df.select_dtypes(include=[np.number]).columns
for column in numerical_columns:
    plt.figure(figsize=(10, 6))
    sns.histplot(cervical_df[column], kde=True)
    plt.title(f'Histogram of {column}')
    plt.show()
```

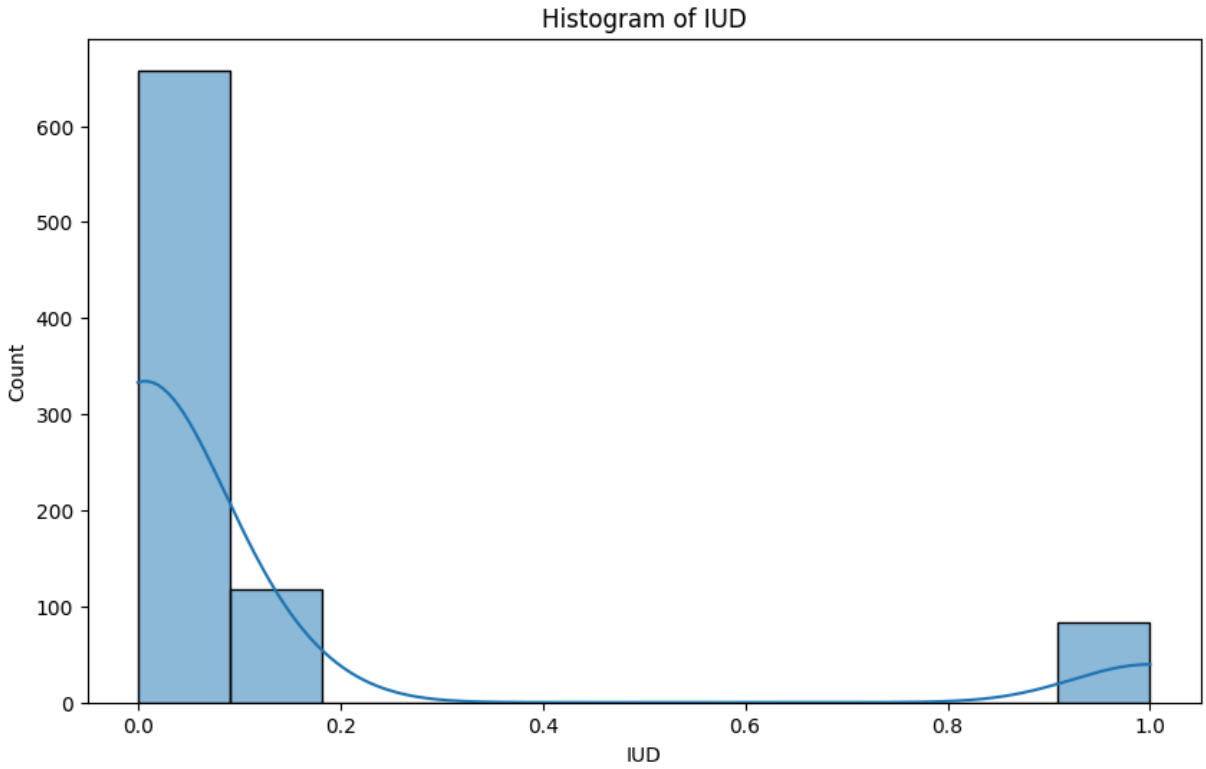


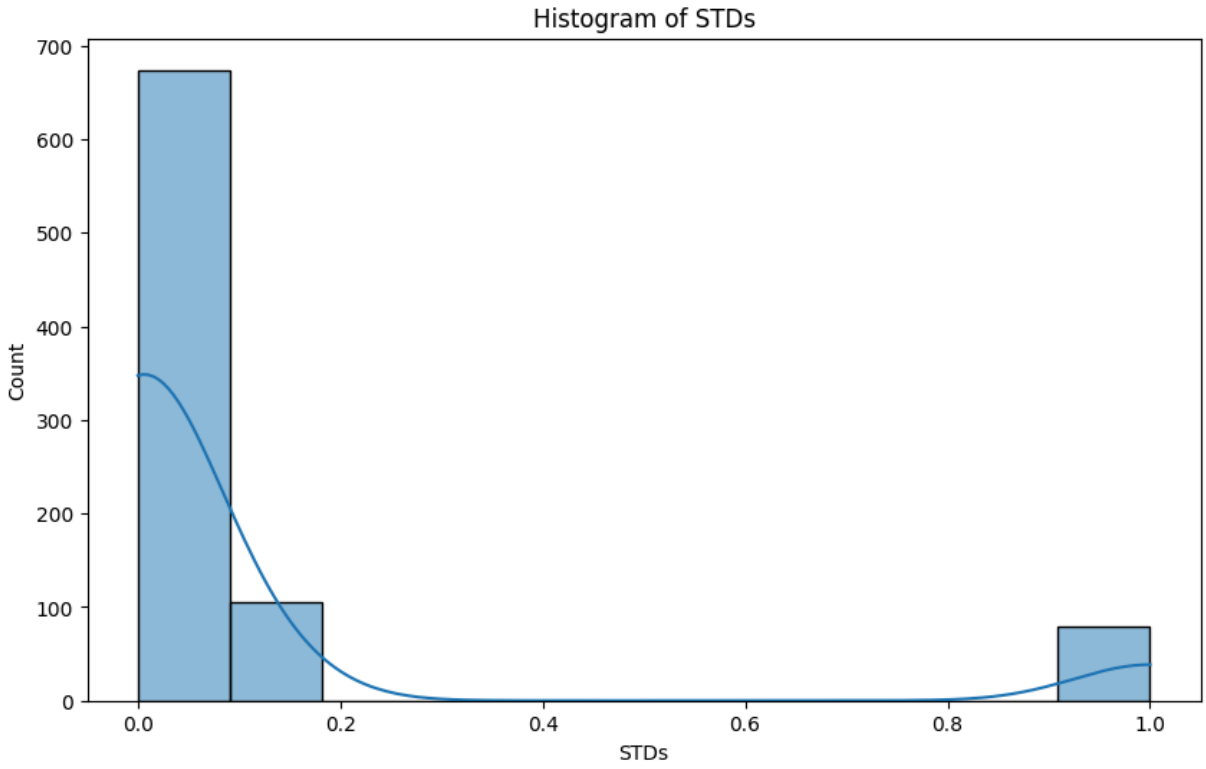

Histogram of First sexual intercourse
Count
First sexual intercourse
0
20
40
60
80
100
120
140
160
10
15
20
25
30

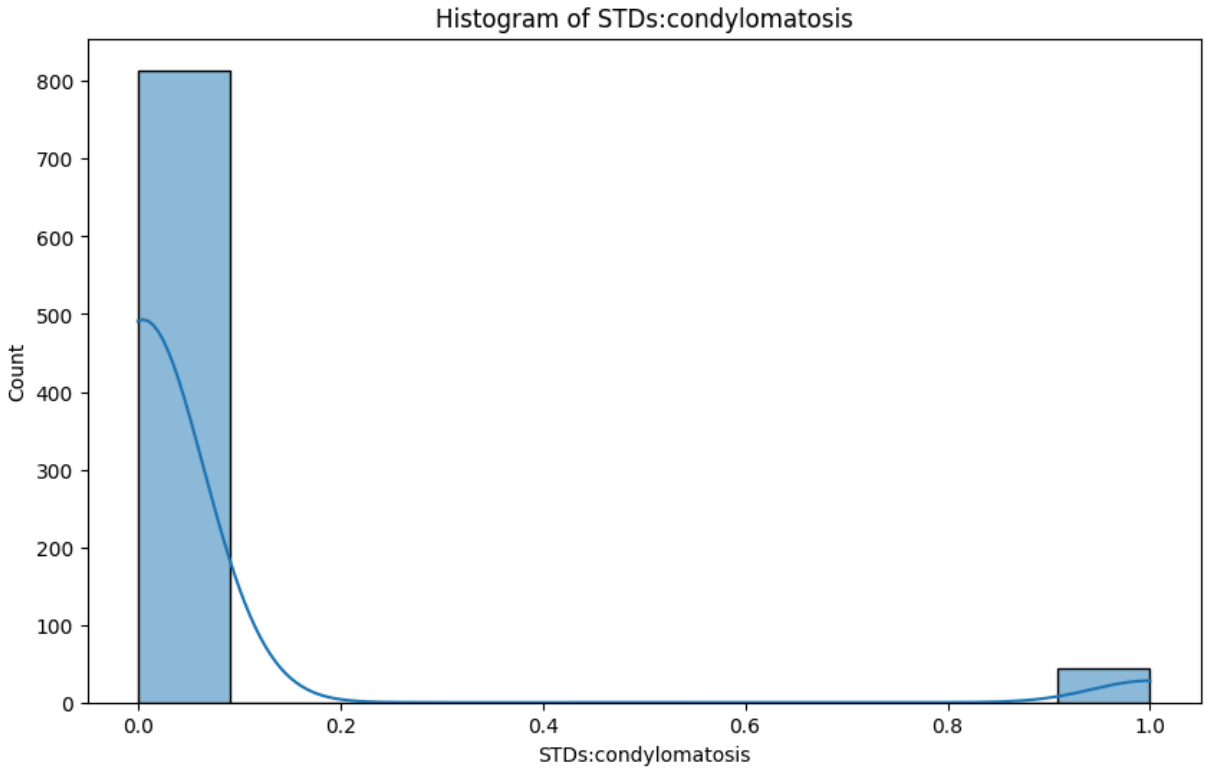

Histogram of Num of pregnancies
Count
Num of pregnancies
0
50
100
150
200
250
0
2
4
6
8
10

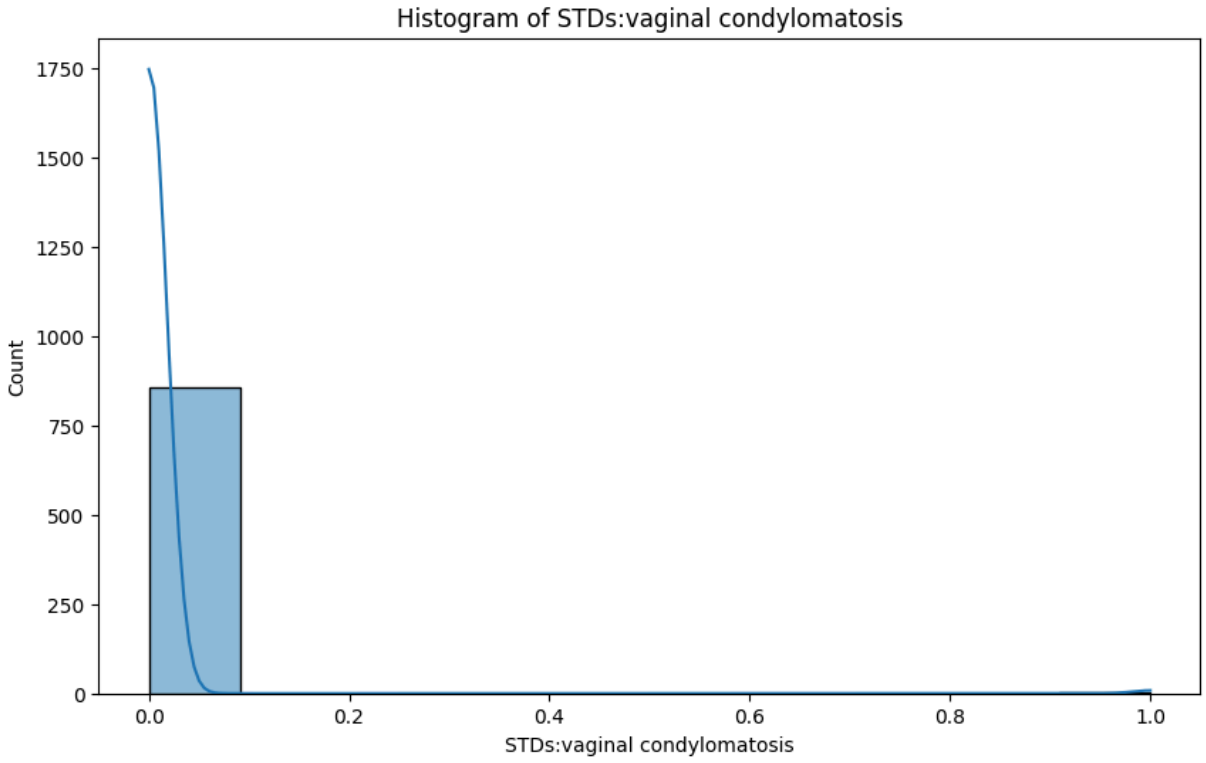

Histogram of Smokes
Count
Smokes
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0

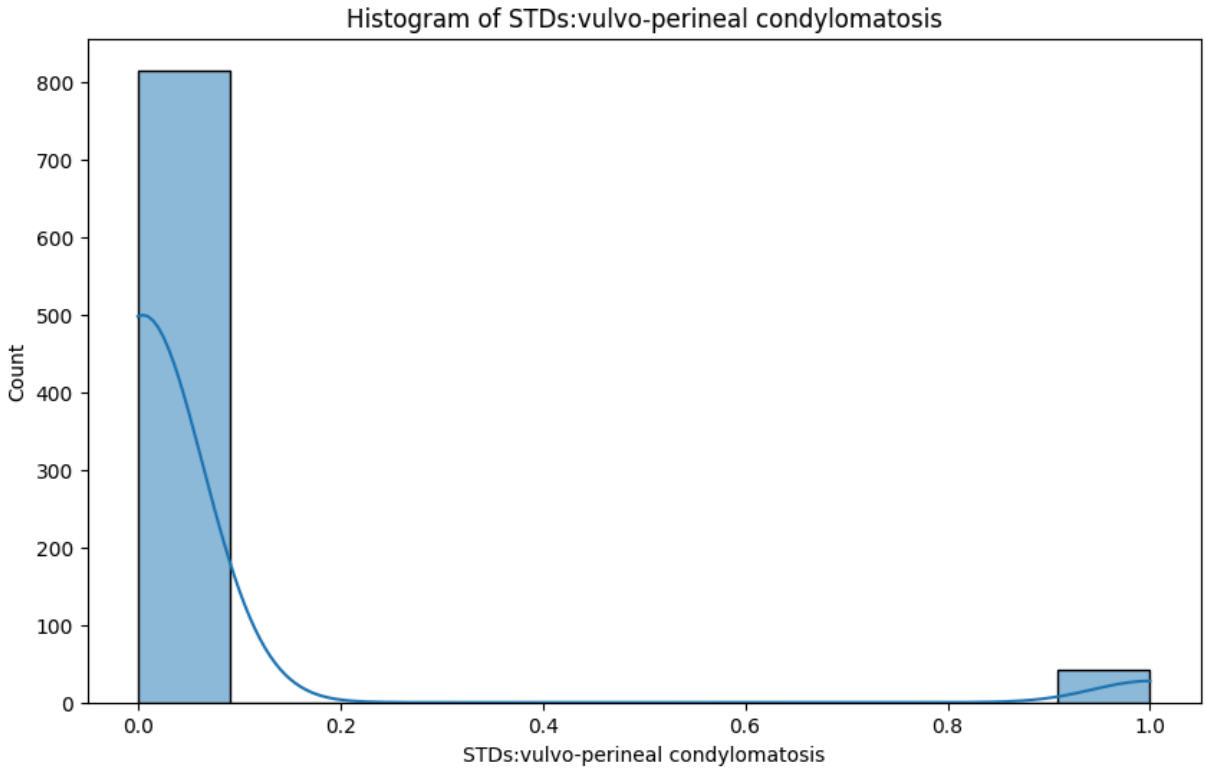

Histogram of Hormonal Contraceptives
Count
Hormonal Contraceptives
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0

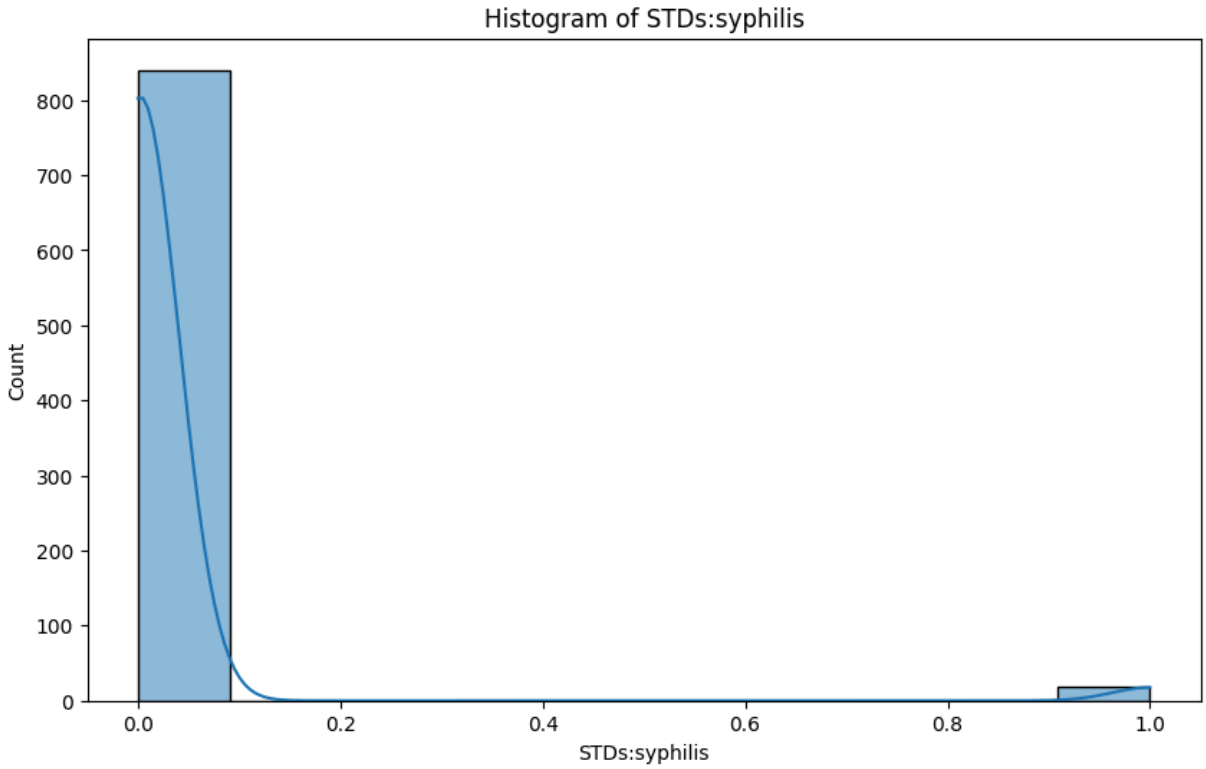

Histogram of IUD
Count
IUD
0
100
200
300
400
500
600
0.0
0.2
0.4
0.6
0.8
1.0

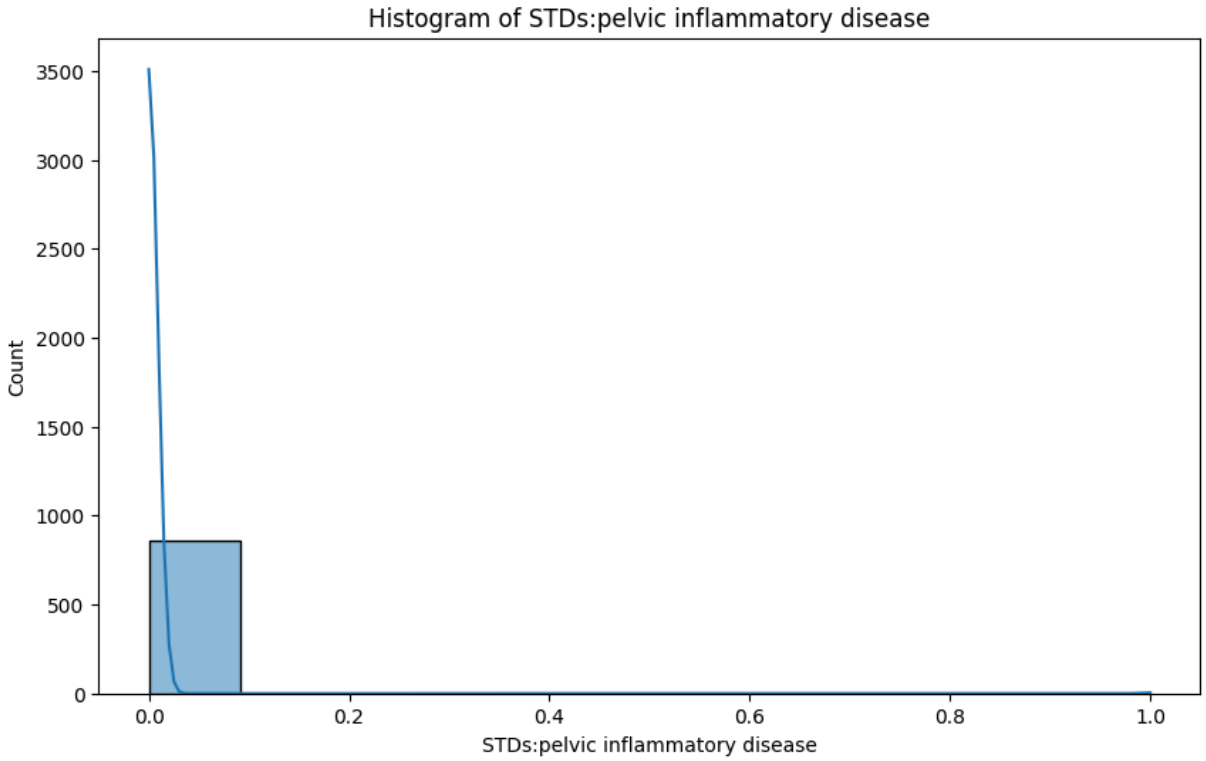

Histogram of STDs
Count
STDs
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0

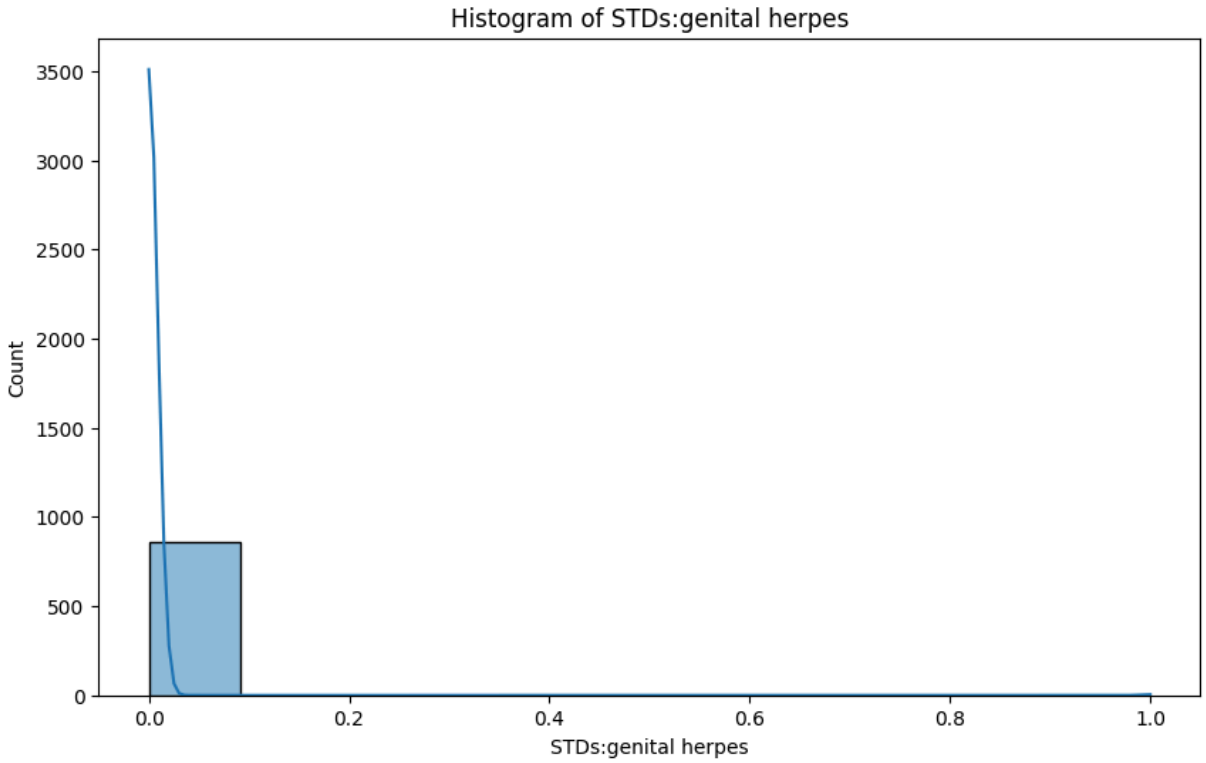

Histogram of STDs:condylomatosis
Count
STDs:condylomatosis
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

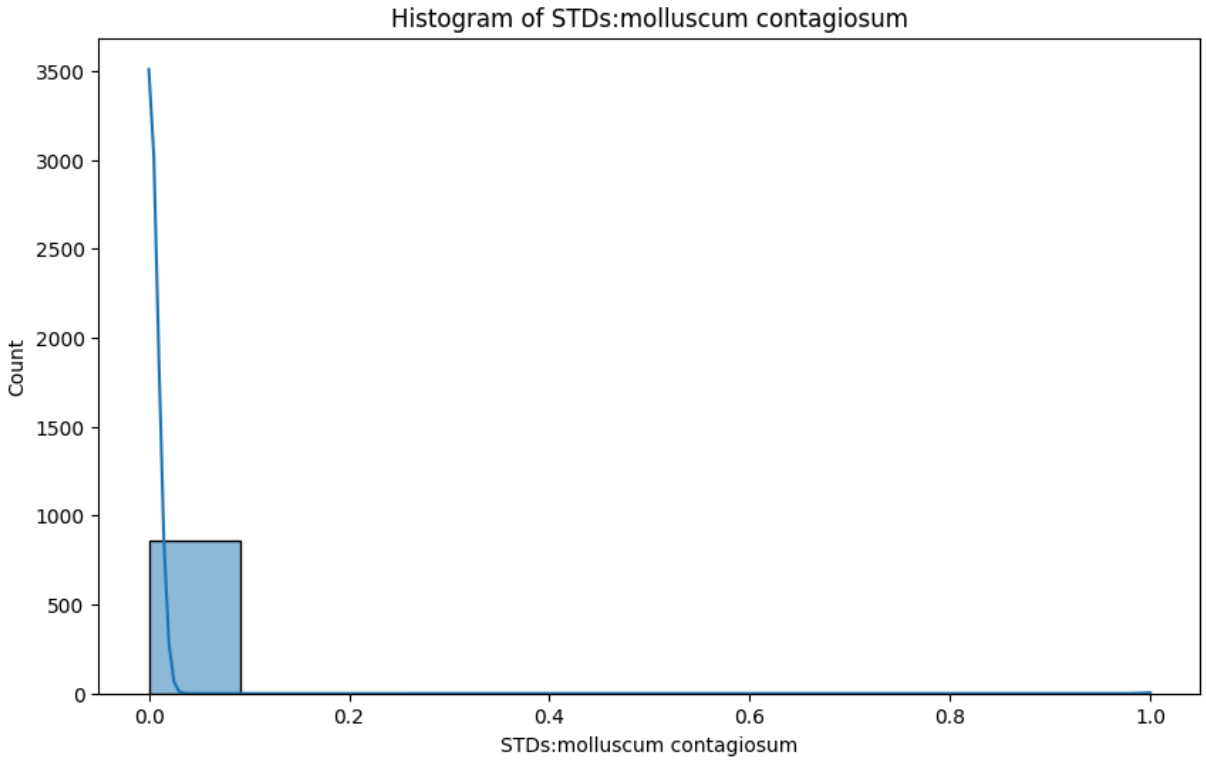

Histogram of STDs:vaginal condylomatosis
Count
STDs:vaginal condylomatosis
0
250
500
750
1000
1250
1500
1750
0.0
0.2
0.4
0.6
0.8
1.0

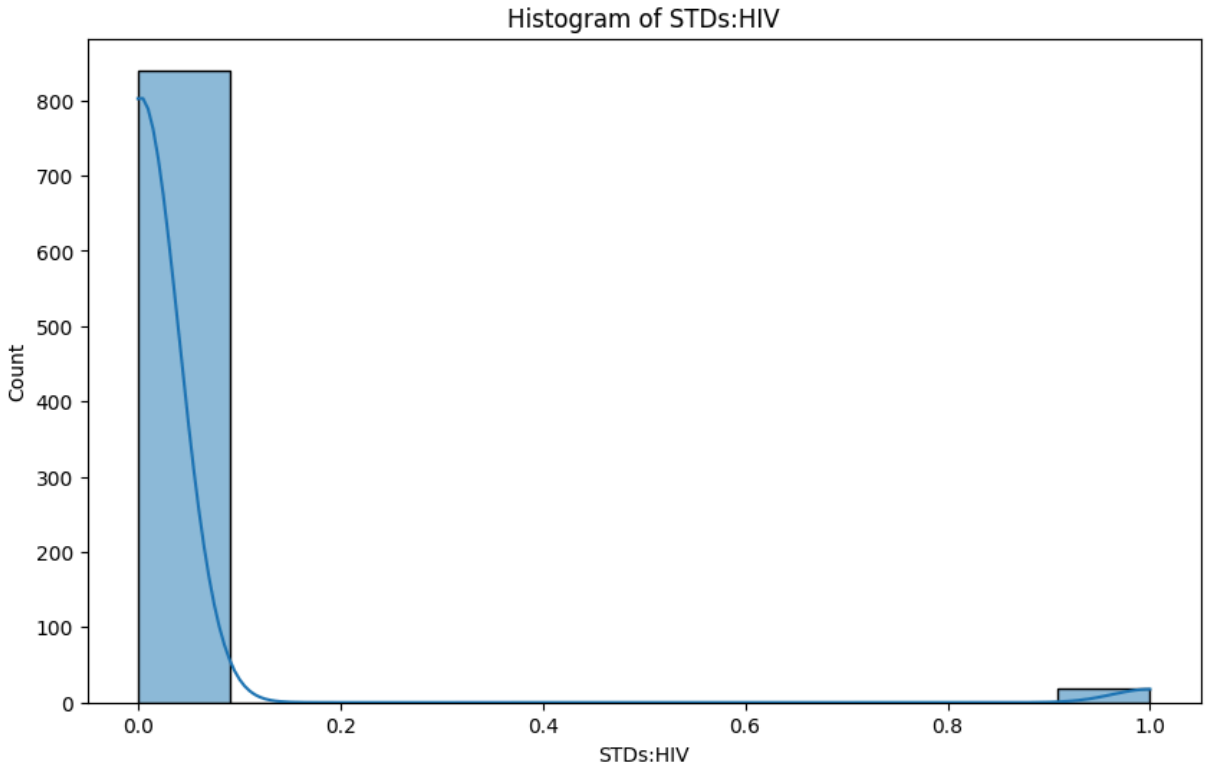

Histogram of STDs:vulvo-perineal condylomatosis
Count
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0
STDs:vulvo-perineal condylomatosis

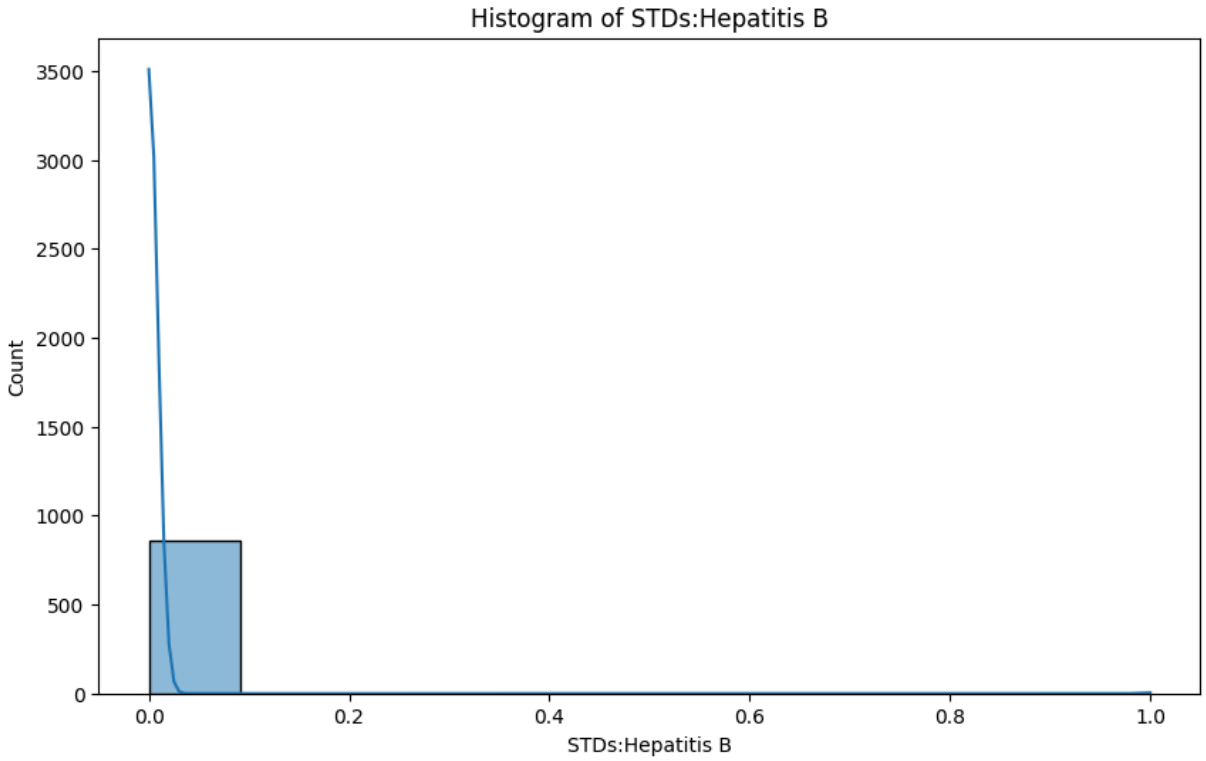

Histogram of STDs:syphilis
Count
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0
STDs:syphilis

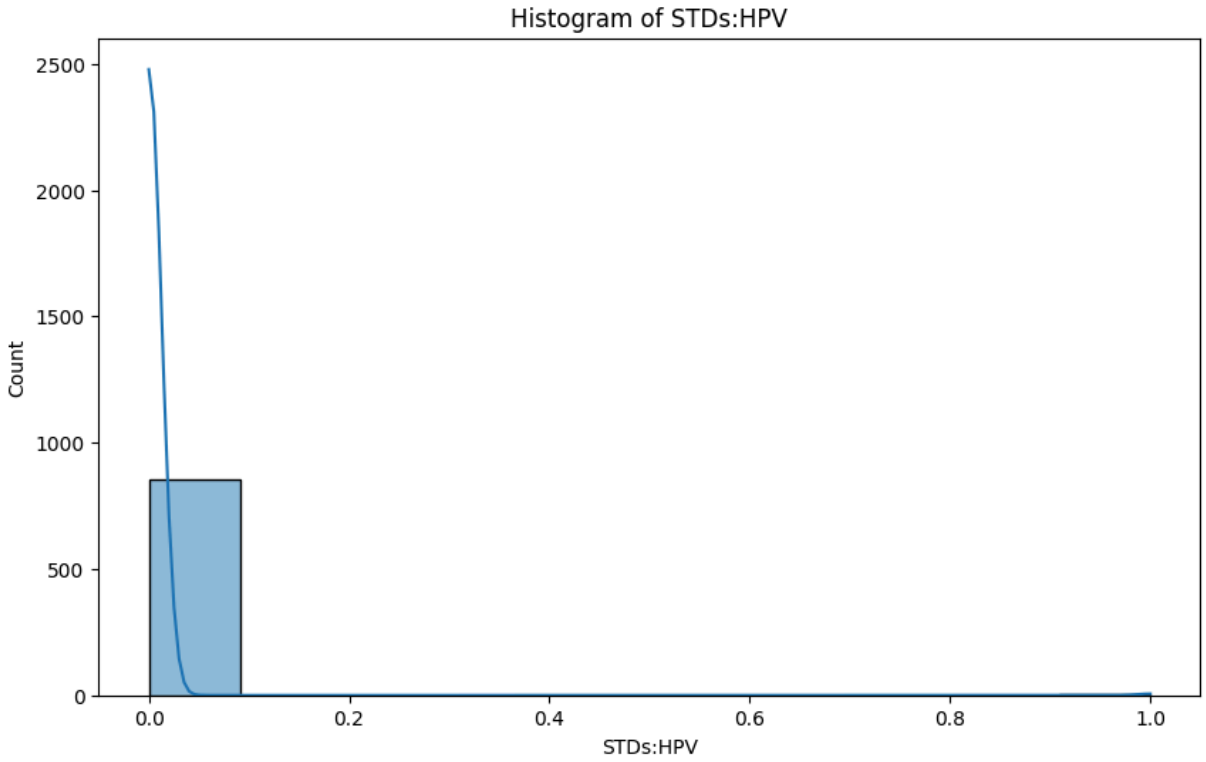

Histogram of STDs:pelvic inflammatory disease
Count
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
STDs:pelvic inflammatory disease

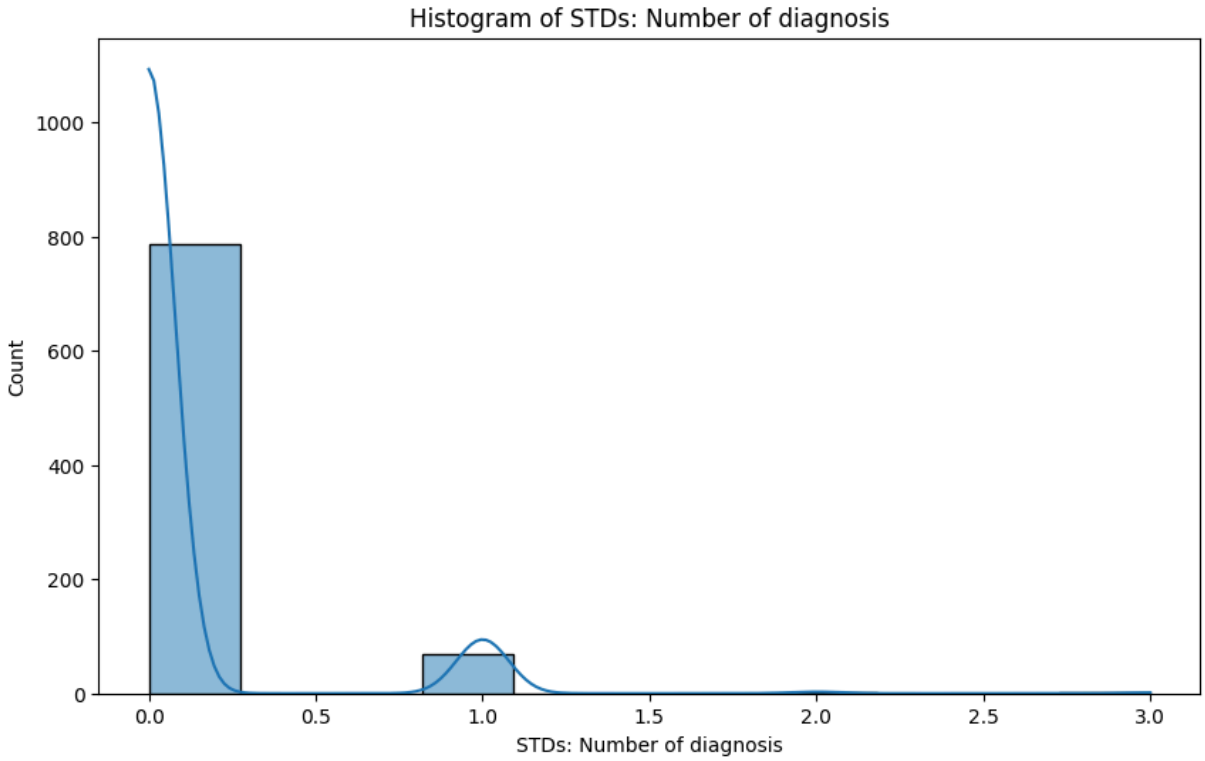

Histogram of STDs:genital herpes
Count
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
STDs:genital herpes

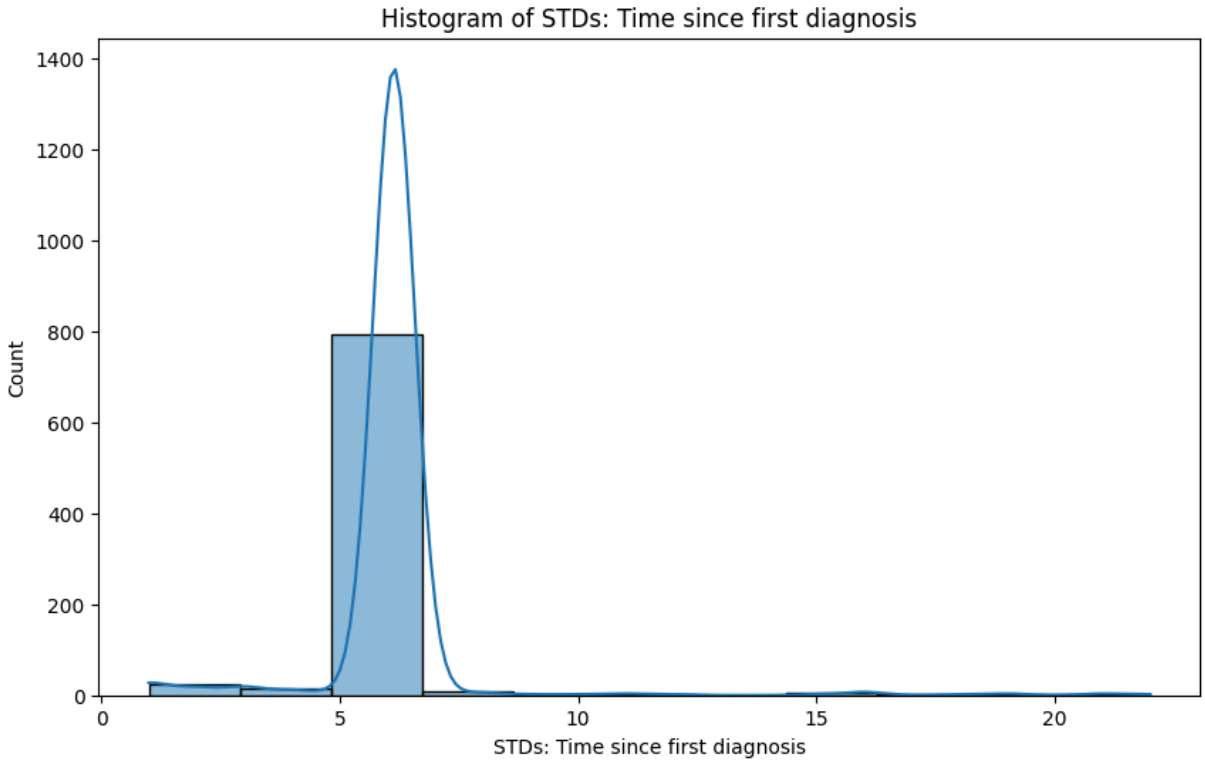

Histogram of STDs:molluscum contagiosum
Count
STDs:molluscum contagiosum
0
500
1000
1500
2000
2500
3000
3500
0.0
0.2
0.4
0.6
0.8
1.0

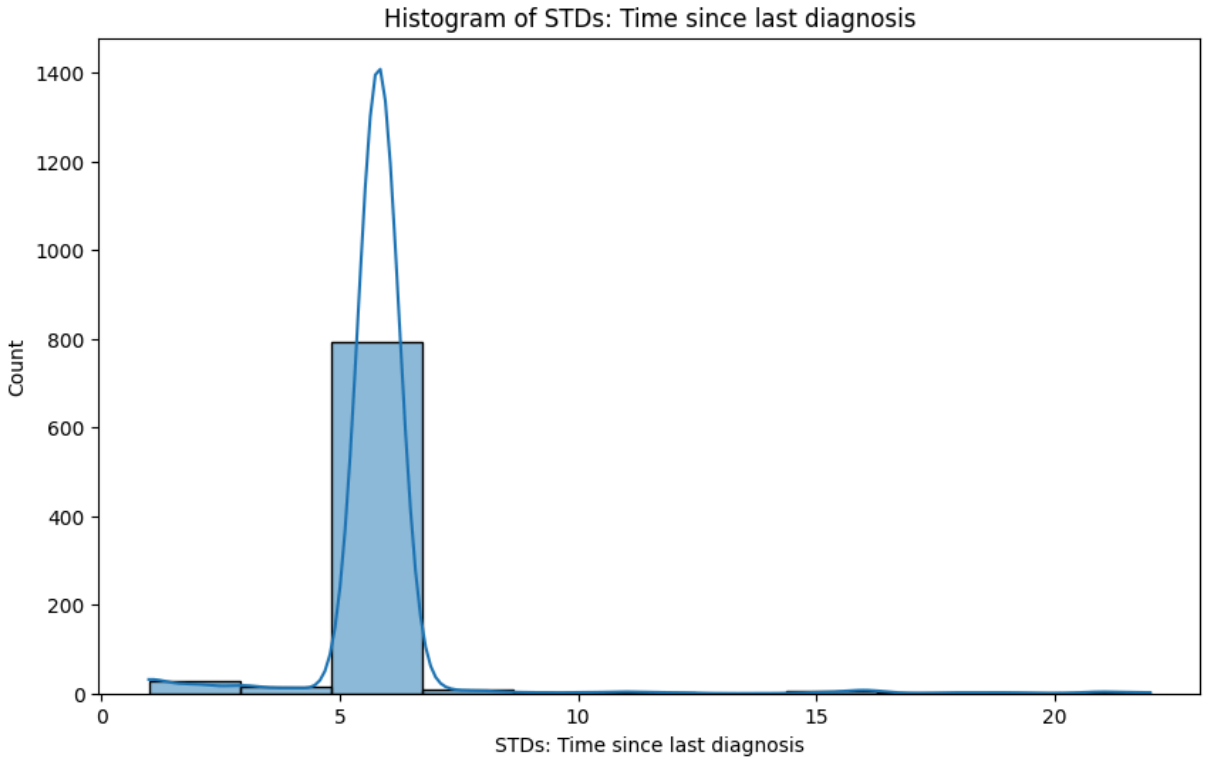

Histogram of STDs:HIV
Count
STDs:HIV
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

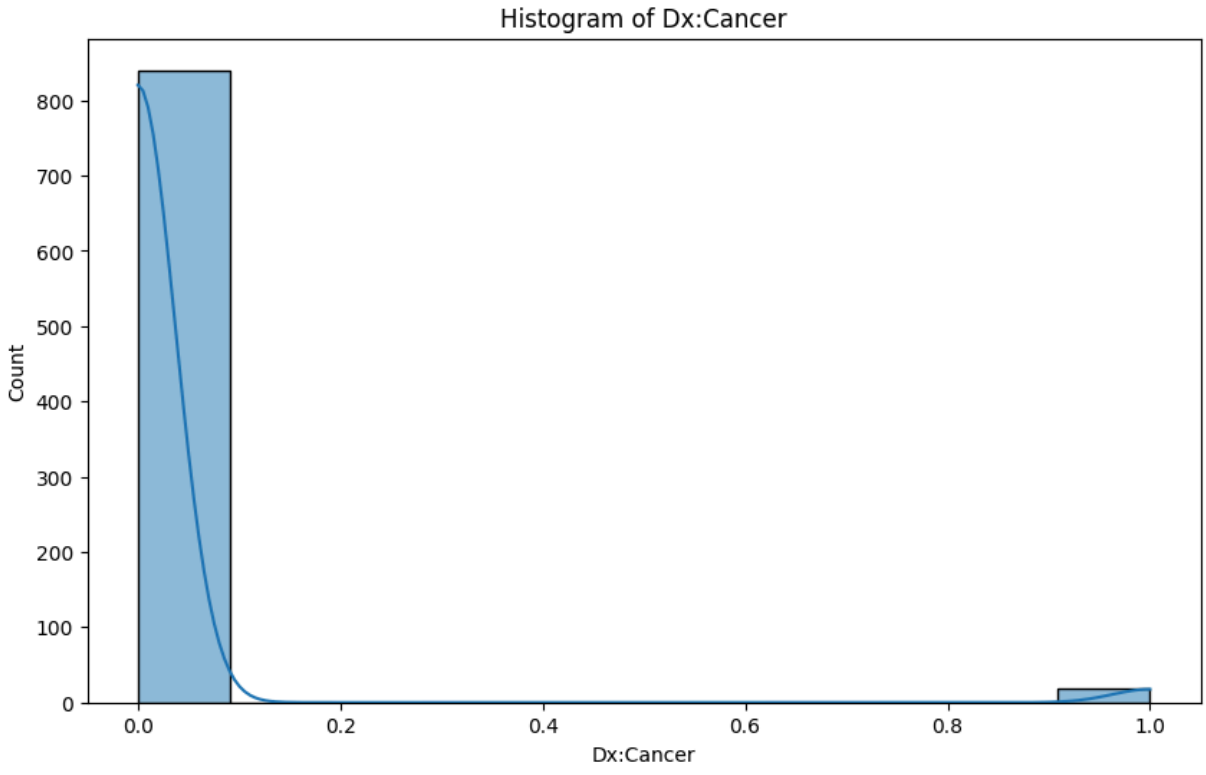

Histogram of STDs:Hepatitis B
Count
STDs:Hepatitis B
0
500
1000
1500
2000
2500
3000
3500
0.0
0.2
0.4
0.6
0.8
1.0

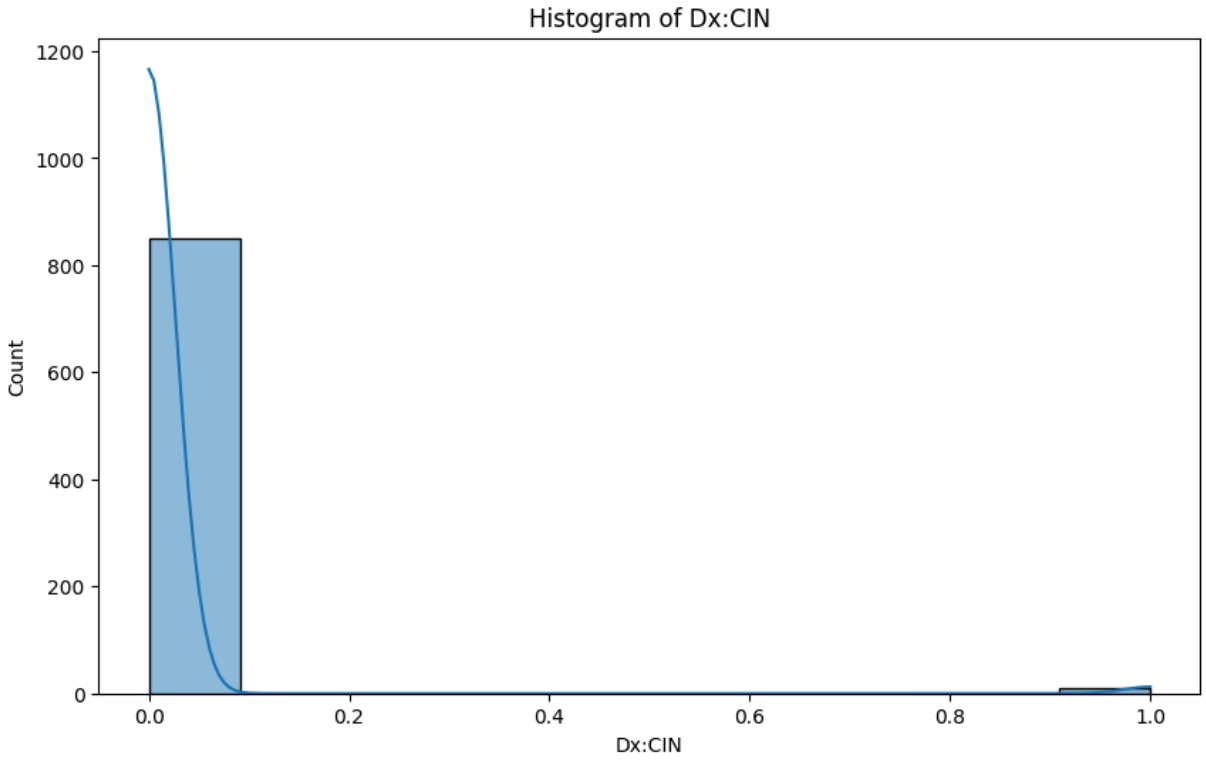

Histogram of STDs:HPV
Count
STDs:HPV
0
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0

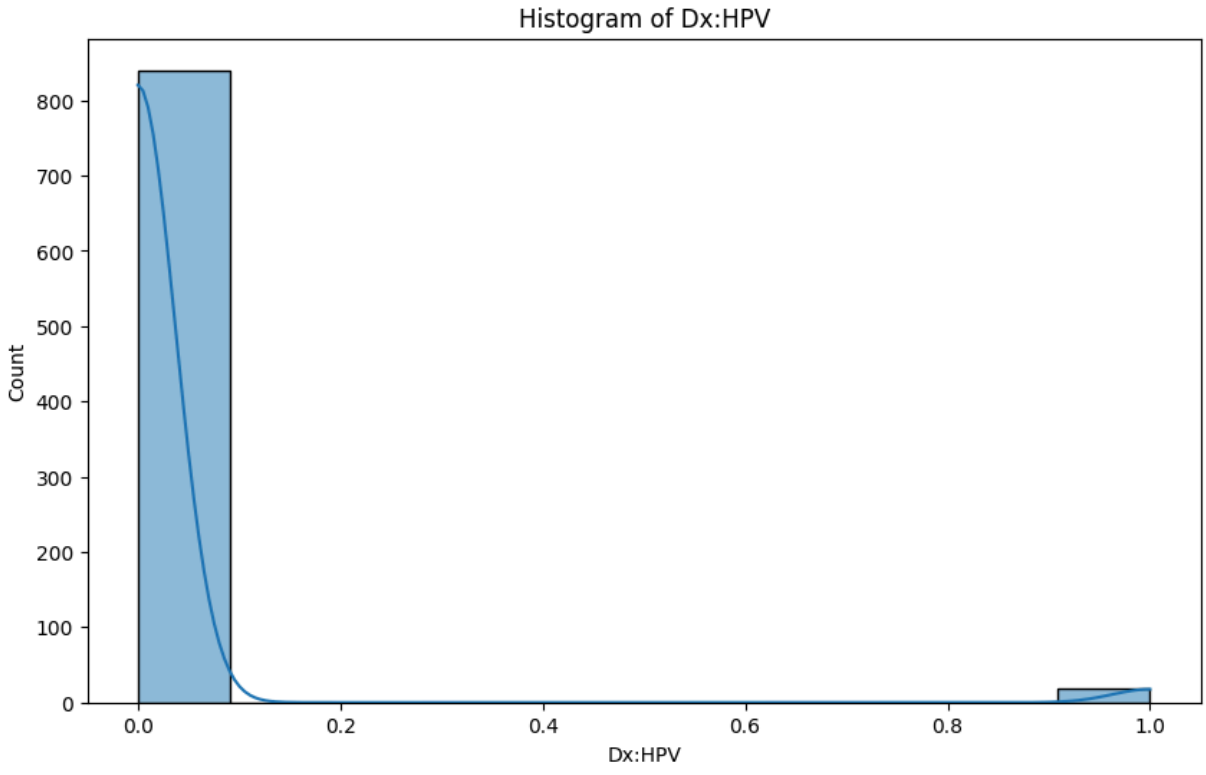

Histogram of STDs: Number of diagnosis
Count
1000
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
STDs: Number of diagnosis

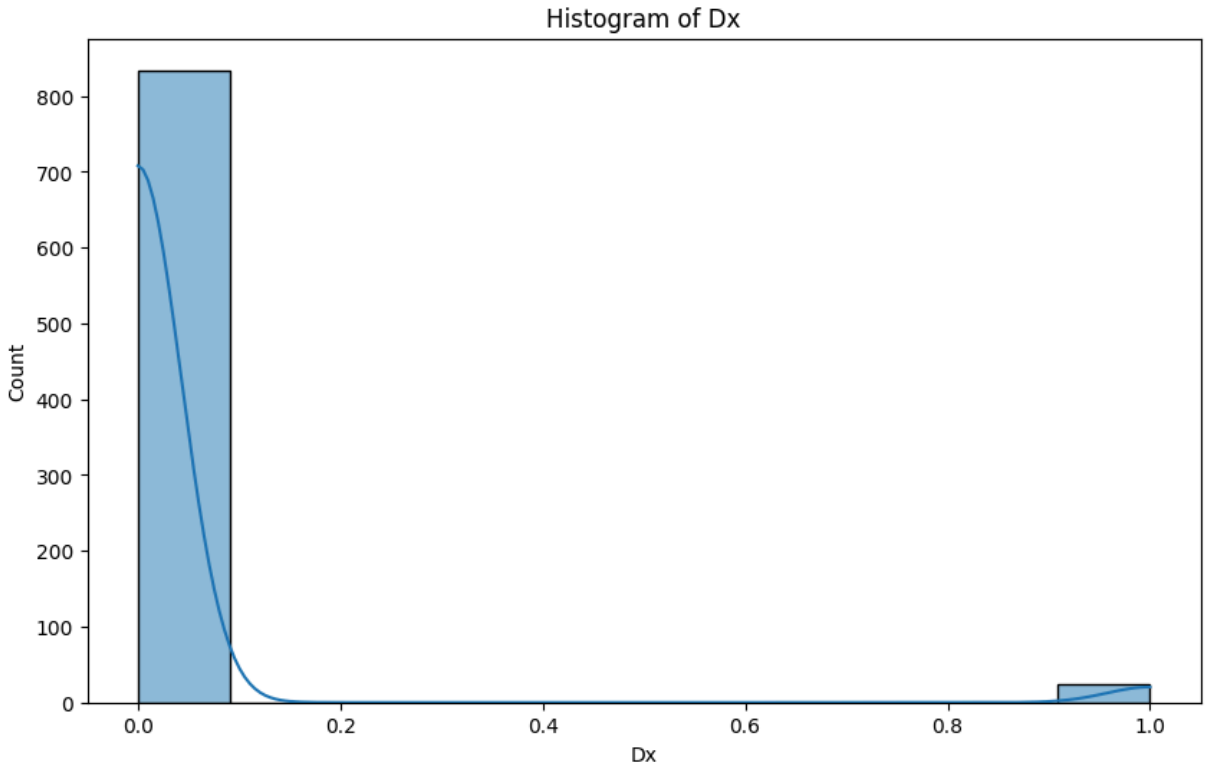

Histogram of STDs: Time since first diagnosis
Count
1400
1200
1000
800
600
400
200
0
0
5
10
15
20
STDs: Time since first diagnosis

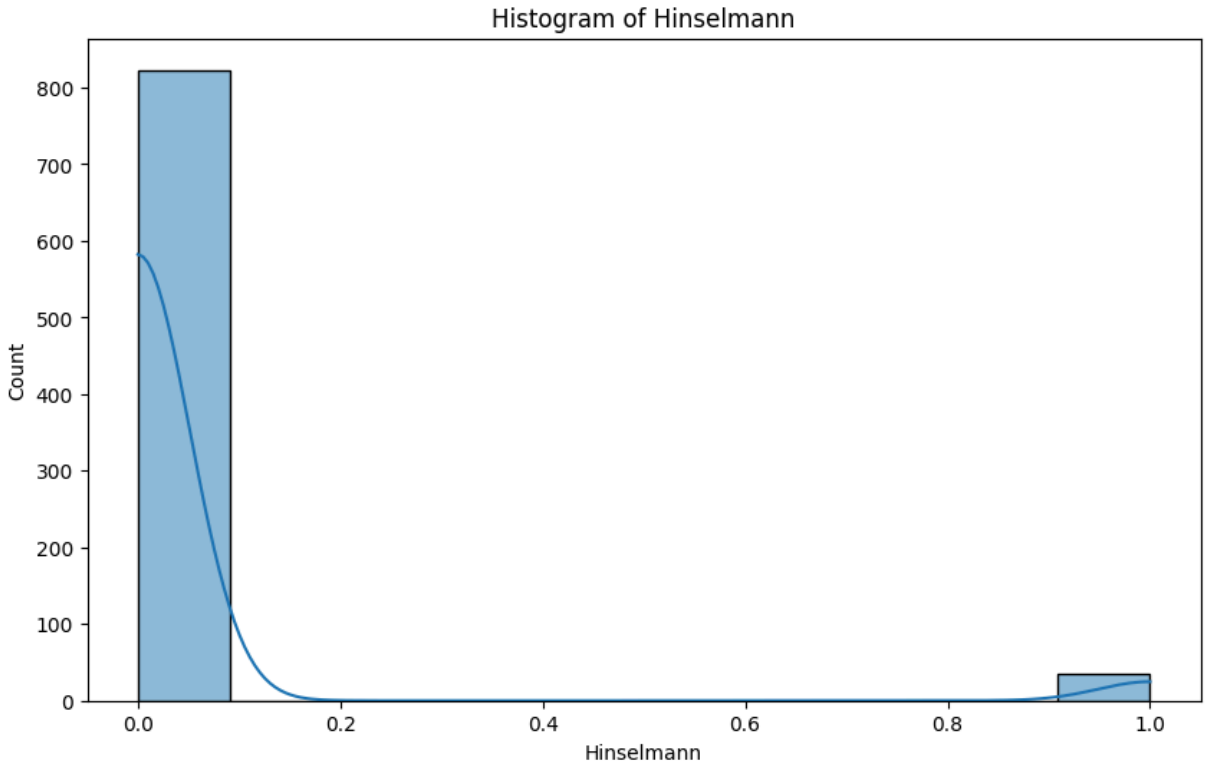

Histogram of STDs: Time since last diagnosis
Count
STDs: Time since last diagnosis
0
200
400
600
800
1000
1200
1400
0
5
10
15
20

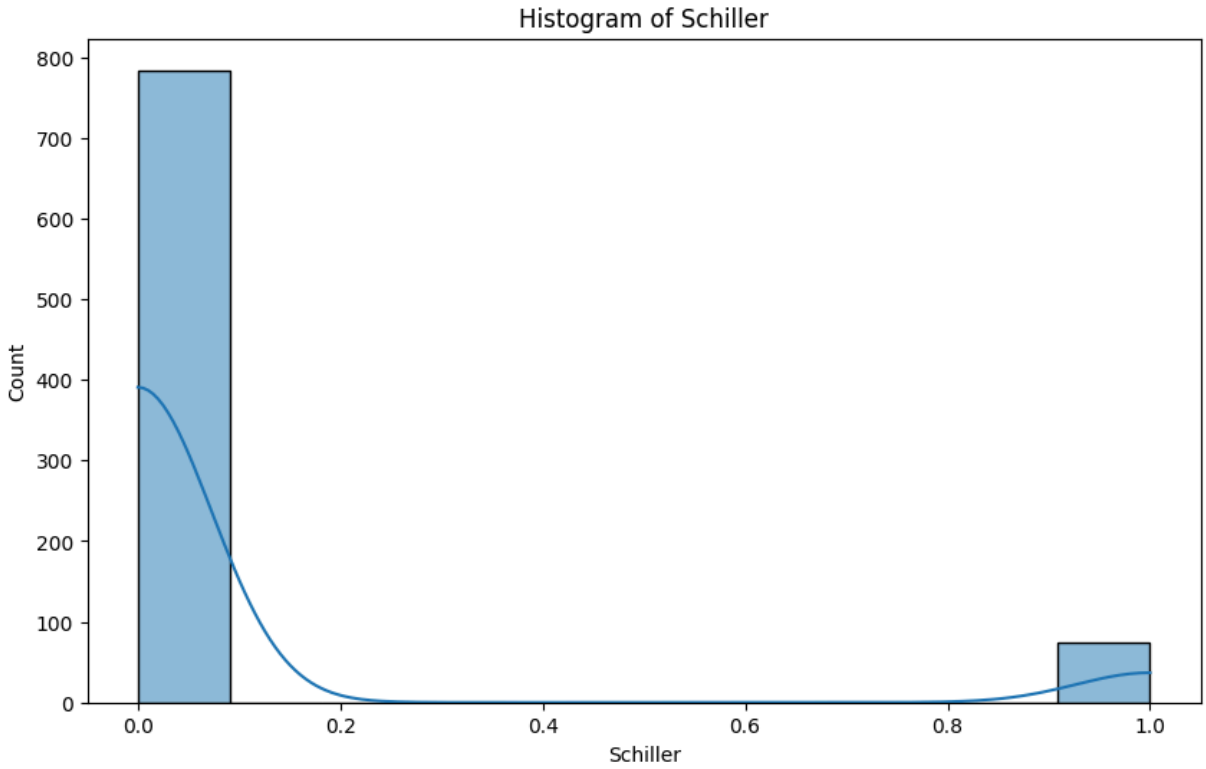

Histogram of Dx:Cancer
Count
Dx:Cancer
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

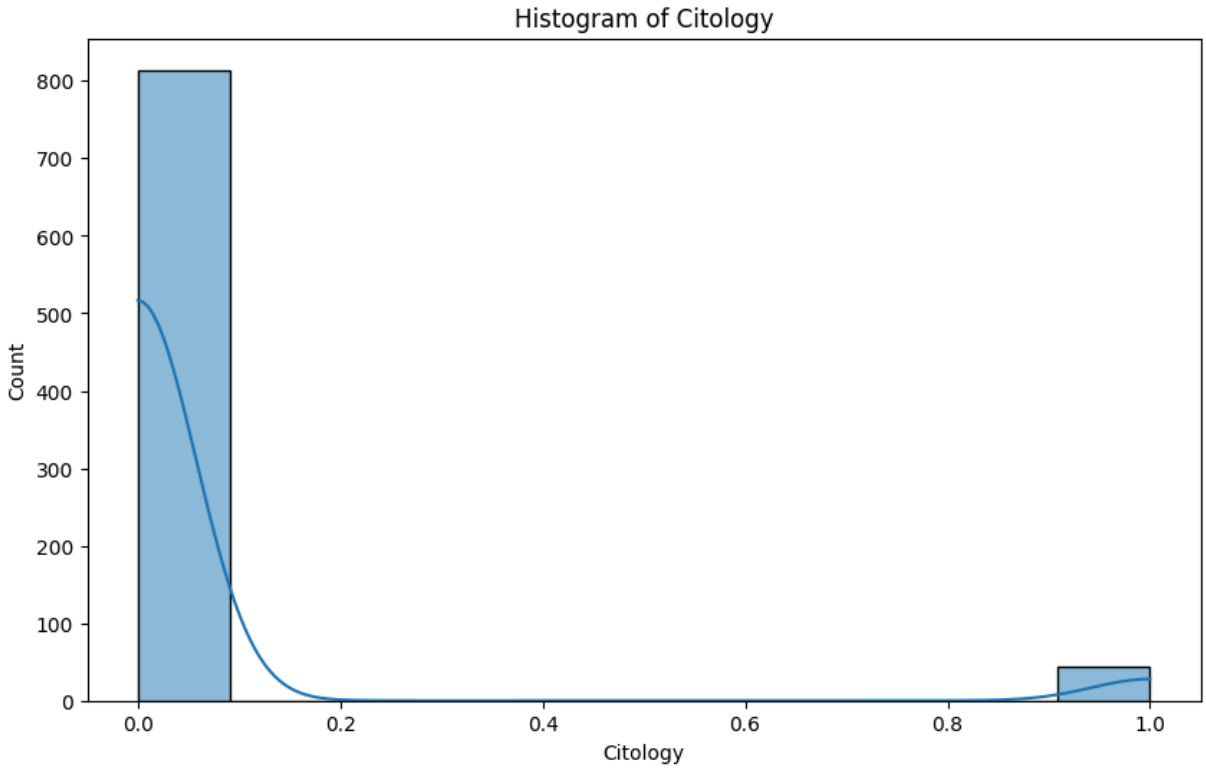

Histogram of Dx:CIN
Count
Dx:CIN
0
200
400
600
800
1000
1200
0.0
0.2
0.4
0.6
0.8
1.0

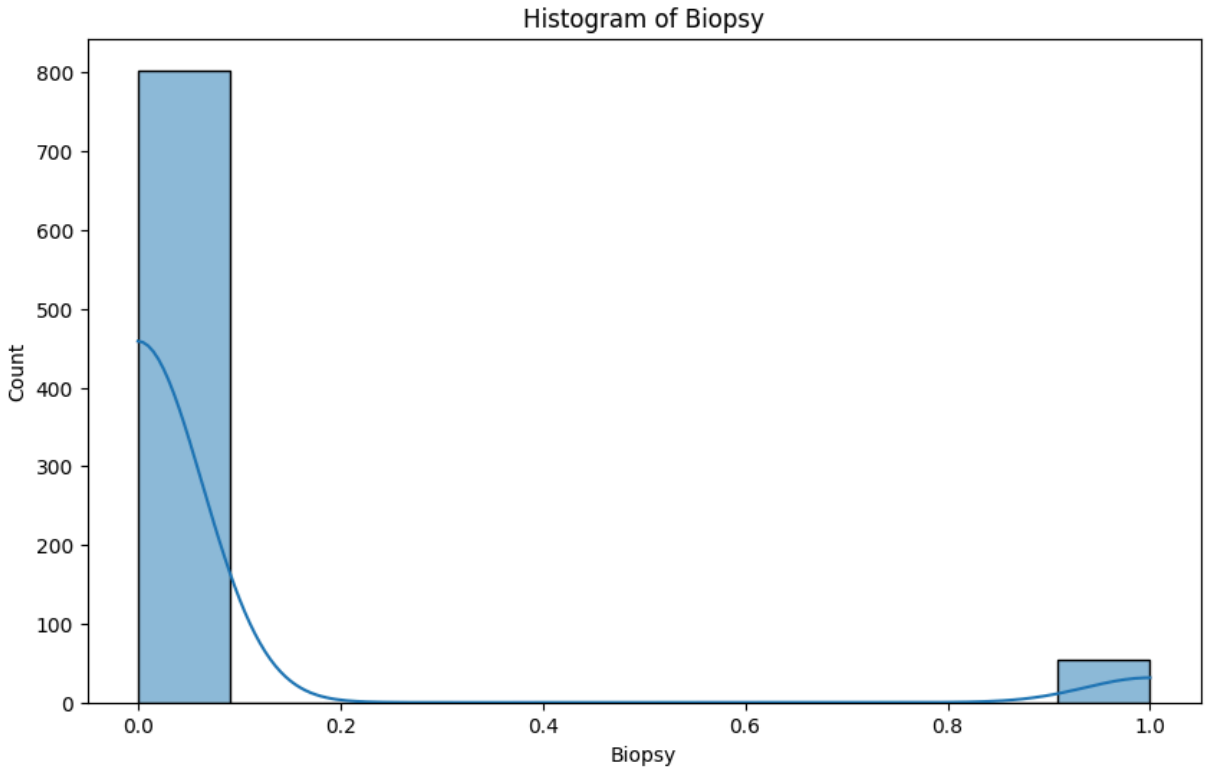

Histogram of Dx:HPV
Count
Dx:HPV
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of Dx
Count
Dx
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of Hinselmann
Count
Hinselmann
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of Schiller
Count
Schiller
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of Citology
Count
Citology
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0



In [180…

```
cervical_df.corr()
```

Out[180…

	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Hormonal Contraceptives	I
Number of sexual partners	1.000000	-0.147937	0.076719	0.238078	0.006342	0.030
First sexual intercourse	-0.147937	1.000000	-0.058223	-0.123602	0.018344	-0.020
Num of pregnancies	0.076719	-0.058223	1.000000	0.080768	0.142858	0.198
Smokes	0.238078	-0.123602	0.080768	1.000000	-0.002165	-0.051
Hormonal Contraceptives	0.006342	0.018344	0.142858	-0.002165	1.000000	0.033
IUD	0.030005	-0.020975	0.198550	-0.051184	0.033729	1.000
STDs	0.053754	-0.013133	0.044250	0.116676	-0.032105	0.053
STDs:condylomatosis	0.034646	0.026777	-0.037999	0.059919	-0.009284	0.077
STDs:vaginal condylomatosis	-0.042924	0.071425	-0.003166	0.069631	-0.059222	0.032
STDs:vulvo-perineal condylomatosis	0.036750	0.031082	-0.037204	0.062775	-0.013714	0.061
STDs:syphilis	0.027178	-0.100999	0.141728	0.082684	-0.003624	-0.022
STDs:pelvic inflammatory disease	0.030616	-0.001089	-0.056542	-0.014059	0.027587	-0.013
STDs:genital herpes	-0.031826	0.023398	-0.032114	-0.014059	0.027587	-0.013
STDs:molluscum contagiosum	0.030616	-0.013332	0.041168	-0.014059	-0.048598	-0.013
STDs:HIV	0.019871	-0.013430	0.009384	0.059412	-0.076278	0.008
STDs:Hepatitis B	-0.011012	0.011154	-0.032114	0.083551	-0.048598	-0.013
STDs:HPV	0.013871	0.033112	-0.028162	0.049171	0.039040	-0.018
STDs: Number of diagnosis	0.051559	-0.013327	0.033514	0.095433	-0.050660	0.029
STDs: Time since first diagnosis	0.018451	0.018214	0.059202	0.022888	0.022702	0.058
STDs: Time since last diagnosis	0.027509	0.025524	0.075320	0.030891	0.039616	0.067
Dx:Cancer	0.022309	0.067283	0.035123	-0.011027	0.026407	0.110

	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Hormonal Contraceptives	I
Dx:CIN	0.015691	-0.032626	0.007344	-0.042822	-0.003334	0.051
Dx:HPV	0.027264	0.043966	0.046753	0.012210	0.038038	0.058
Dx	0.022982	0.035750	0.019025	-0.067614	-0.001723	0.138
Hinselmann	-0.039273	-0.016546	0.038685	0.034527	0.033551	0.044
Schiller	-0.008899	0.003493	0.087687	0.053613	-0.004247	0.084
Citology	0.021839	-0.010971	-0.029656	-0.003913	-0.011030	0.007
Biopsy	-0.001429	0.007262	0.043460	0.029091	0.007711	0.051

28 rows × 28 columns

Logistic Regression Model

In [181…
```
X = cervical_df.drop(columns = ['Biopsy'])
```

In [182…
```
y = cervical_df['Biopsy']
```

In [183…
```
from sklearn.model_selection import train_test_split
```

In [204…
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.5, random_s
```

In [205…
```
X_train
```

Out[205…

	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Hormonal Contraceptives	IUD	STDs	STDs
647	2.527644	15.0	3.000000	0.0	1.000000	0.000000	0.000000	
698	4.000000	16.0	2.275561	0.0	0.000000	0.000000	0.000000	
559	1.000000	16.0	2.000000	0.0	0.000000	0.000000	0.000000	
250	2.000000	18.0	2.000000	0.0	1.000000	0.000000	0.000000	
605	2.000000	18.0	1.000000	0.0	1.000000	0.000000	0.000000	
…	…	…	…	…	…	…	…	
715	2.000000	14.0	1.000000	0.0	0.000000	0.000000	0.000000	
767	2.000000	13.0	1.000000	0.0	0.641333	0.112011	0.104914	
72	2.000000	21.0	2.000000	0.0	1.000000	0.000000	0.000000	
235	2.000000	17.0	1.000000	0.0	1.000000	0.000000	0.000000	
37	2.527644	18.0	1.000000	0.0	1.000000	0.000000	0.000000	

429 rows × 27 columns

In [206…

```
X_test
```

Out[206…

	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Hormonal Contraceptives	IUD	STDs	STDs:condyl
255	2.0	18.0	2.0	0.000000	1.0	0.0	0.0	
56	5.0	15.0	4.0	0.000000	0.0	0.0	0.0	
479	1.0	24.0	2.0	0.000000	1.0	0.0	0.0	
84	2.0	15.0	3.0	0.000000	0.0	0.0	1.0	
589	3.0	18.0	4.0	0.000000	1.0	0.0	0.0	
...	...	...	...	...	...	...	...	
42	3.0	18.0	3.0	1.000000	1.0	0.0	0.0	
669	3.0	22.0	2.0	0.145562	1.0	1.0	0.0	
133	4.0	18.0	2.0	1.000000	1.0	0.0	0.0	
640	4.0	18.0	3.0	0.000000	1.0	0.0	0.0	
389	1.0	17.0	1.0	0.000000	0.0	0.0	0.0	

429 rows × 27 columns

In [207…
```
from sklearn.preprocessing import StandardScaler
```

In [208…
```
scaler = StandardScaler()
```

In [209…
```
X_train_scaled = scaler.fit_transform(X_train)
```

In [210…
```
X_test_scaled = scaler.transform(X_test)
```

In [211…
```
X_train_scaled
```

Out[211…
```
array([[ 0.07516296, -0.72738414,  0.55092521, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [ 1.19369986, -0.37596966,  0.03607382, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [-1.08537615, -0.37596966, -0.15976465, ..., -0.20313083,
        -0.27420425, -0.21527067],
       ...,
       [-0.32568415,  1.38110271, -0.15976465, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [-0.32568415, -0.02455519, -0.87045452, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [ 0.07516296,  0.32685929, -0.87045452, ..., -0.20313083,
        -0.27420425, -0.21527067]])
```

In [212…
```
X_test_scaled
```

Out[212…
```
array([[-0.32568415,  0.32685929, -0.15976465, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [ 1.95339186, -0.72738414,  1.26161507, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [-1.08537615,  2.43534614, -0.15976465, ..., -0.20313083,
        -0.27420425, -0.21527067],
       ...,
       [ 1.19369986,  0.32685929, -0.15976465, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [ 1.19369986,  0.32685929,  0.55092521, ..., -0.20313083,
        -0.27420425, -0.21527067],
       [-1.08537615, -0.02455519, -0.87045452, ..., -0.20313083,
        -0.27420425, -0.21527067]])
```

In [213…
```
from sklearn.linear_model import LogisticRegression
```

In [214…
```
log_reg = LogisticRegression(random_state= 0).fit(X_train_scaled,y_train)
```

In [215…
```
log_reg.predict(X_train_scaled)
```

Out[215…
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [216…
```
log_reg.score(X_train_scaled, y_train)
```

Out[216…
```
0.9673659673659674
```

In [217…
```
log_reg.score(X_test_scaled, y_train)
```

Out[217…
```
0.8601398601398601
```

In [218…
```
log_reg1 = LogisticRegression(random_state =0,
                              C=1,
```

```
                                fit_intercept = True,
                                ).fit(X_train_scaled,y_train)
```

In [219…
```
log_reg1.score(X_train_scaled, y_train)
```

Out[219…
```
0.9673659673659674
```

In [220…
```
log_reg1.score(X_test_scaled, y_test)
```

Out[220…
```
0.9463869463869464
```

Findings

The accuracy score of the logistic regression model that predicts the target variable Biopsy, the train scale has a score of 0.96 or 96% which indicates a strong performance and also the test scale has a score of 0.94 or 94%, the logistic regression model is performing well based on the accuracy scores provided.

In [ ]:
```

```