```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
admission = pd.read_csv('admissions.csv')
fatalities = pd.read_csv('fatalities.csv')
metrics = pd.read_csv('metrics.csv')
prescriptions = pd.read_csv('prescriptions.csv')
smokers = pd.read_csv('smokers.csv')
```

```
admission.head(5)
```

	Year	ICD10 Code	ICD10 Diagnosis	Diagnosis Type	Metric	Sex	Value
0	2014/15	All codes	All admissions	All admissions	Number of admissions	NaN	11011882
1	2014/15	C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C6...	All diseases which can be caused by smoking	All diseases which can be caused by smoking	Number of admissions	NaN	1713330

```
admission.isnull().sum()
```

	0
Year	0
ICD10 Code	0
ICD10 Diagnosis	0
Diagnosis Type	0
Metric	0
Sex	693
Value	1

dtype: int64

```
admission.shape
```

```
(2079, 7)
```

```
admission.dropna(inplace =True)
```

```
admission.isnull().sum()
```

	0
Year	0
ICD10 Code	0
ICD10 Diagnosis	0
Diagnosis Type	0
Metric	0
Sex	0
Value	0

dtype: int64

```
fatalities.head(5)
```

	Year	ICD10 Code	ICD10 Diagnosis	Diagnosis Type	Metric	Sex	Value
0	2014	All codes	All deaths	All deaths	Number of observed deaths	NaN	459087
1	2014	C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C6...	All deaths which can be caused by smoking	All deaths which can be caused by smoking	Number of observed deaths	NaN	235820

```
fatalities.isnull().sum()
```

	0
Year	0
ICD10 Code	0
ICD10 Diagnosis	0
Diagnosis Type	0
Metric	0
Sex	583
Value	0

dtype: int64

```
fatalities.dropna(inplace = True)
```

```
metrics.isnull().sum()
```

	0
Year	0
Tobacco Price\nIndex	0
Retail Prices\nIndex	0
Tobacco Price Index Relative to Retail Price Index	0
Real Households' Disposable Income	0
Affordability of Tobacco Index	0
Household Expenditure on Tobacco	5
Household Expenditure Total	5
Expenditure on Tobacco as a Percentage of Expenditure	5

dtype: int64

```
prescriptions.dropna(inplace = True)
```

```
smokers.dropna(inplace = True)
```

```
admission['ICD10 Code'].value_counts()
```

ICD10 Code	count
All codes	44
J40-J43	44
S72.0-S72.2	44
H25	44
K05	44
K50	44
K25-K27	44

I70	44
I71	44
I60-I69	44
I72-I78	44
I20-I25	44
I00-I09 & I26-I51	44
J10-J18	44
J44	44
C92	44
C80	44
C25	44
C16	44
C64-C66 & C68	44
C67	44
C53	44
C32	44
C15	44
C00-C14	44
C33-C34	44
K00-K93	44
I00-I99	44
J00-J99	44
C00-D48	44
O03	44
C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C64-C66 & C68 & C16 & C25 & C80 & C92 & J40-J43 & J44 & J10-J18 & I00-I09 & I26-I51 & I20-I25 & I72-I78 & I60-I69 & I71 & I70 & K25-K27 & K50 & K05 & H25 & S72.0-S72.2 & O03	22

dtype: int64

```
admission.head()
```

	Year	ICD10 Code	ICD10 Diagnosis	Diagnosis Type	Metric	Sex	Value

63	2014/15	All codes	All admissions	All admissions	Number of admissions	Male	5141482
64	2014/15	C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C6...	All diseases which can be caused by smoking	All diseases which can be caused by smoking	Number of admissions	Male	931001

```
admission['ICD10 Diagnosis'].value_counts()
```

ICD10 Diagnosis	count
All admissions	44
Chronic Obstructive Lung Disease	44
Hip Fracture 55+	44
Age Related Cataract 45+	44
Periodontal Disease / Periodonitis	44
Crohns Disease	44
Stomach / Duodenal Ulcer	44
Atherosclerosis	44
Aortic Aneurysm	44
Cerebrovascular Disease	44
Other arterial disease	44
Ischaemic Heart Disease	44
Other Heart Disease	44
Pneumonia, Influenza	44
Chronic Airway Obstruction	44
Myeloid Leukaemia	44
Unspecified Site	44
Pancreas	44
Stomach	44
Kidney and Renal Pelvis	44
Bladder	44
Cervical	44

Larynx	44
Oesophagus	44
Upper Respiratory Sites	44
Trachea, Lung, Bronchus	44
All diseases of the digestive system	44
All circulatory diseases	44
All respiratory diseases	44
All cancers	44
Spontaneous Abortion	44
All diseases which can be caused by smoking	22

dtype: int64

```
prescriptions.columns
```

```
Index(['Year', 'All Pharmacotherapy Prescriptions',
       'Nicotine Replacement Therapy (NRT) Prescriptions',
       'Bupropion (Zyban) Prescriptions',
       'Varenicline (Champix) Prescriptions',
       'Net Ingredient Cost of All Pharmacotherapies',
       'Net Ingredient Cost of Nicotine Replacement Therapies (NRT)',
       'Net Ingredient Cost of Bupropion (Zyban)',
       'Net Ingredient Cost of Varenicline (Champix)'],
      dtype='object')
```

```
metrics.columns
```

```
Index(['Year', 'Tobacco Price\nIndex', 'Retail Prices\nIndex',
       'Tobacco Price Index Relative to Retail Price Index',
       'Real Households' Disposable Income', 'Affordability of Tobacco Index',
       'Household Expenditure on Tobacco', 'Household Expenditure Total',
       'Expenditure on Tobacco as a Percentage of Expenditure'],
      dtype='object')
```

```
fatalities.columns
```

```
Index(['Year', 'ICD10 Code', 'ICD10 Diagnosis', 'Diagnosis Type', 'Metric',
       'Sex', 'Value'],
      dtype='object')
```

```
print('n\ Statistics for Admission')
print(admission.describe())
```

```
n\ Statistics for Admission
           Year ICD10 Code ICD10 Diagnosis  \
count      1386       1386            1386
unique       11         32              32
top     2014/15  All codes  All admissions
freq        126         44              44

                                  Diagnosis Type                Metric   Sex  \
count                                       1386                  1386  1386
unique                                        11                     2     2
top     Cancers which can be caused by smoking  Number of admissions  Male
freq                                         484                   704   693

       Value
count   1386
unique   953
top        .
freq      40
```

```
print('n\ Statistics for fatalities')
print(fatalities.describe())
```

```
n\ Statistics for fatalities
              Year
count  1166.000000
mean   2009.000000
std       3.163635
min    2004.000000
25%    2006.000000
50%    2009.000000
75%    2012.000000
max    2014.000000
```

```
print('n\ Statistics for metrics')
print(metrics.describe())
```

```
n\ Statistics for metrics
              Year  Tobacco Price\nIndex  Retail Prices\nIndex  \
count    36.000000             36.000000             36.000000
mean   1997.500000            520.744444            239.483333
std      10.535654            336.517379             83.155536
min    1980.000000            100.000000            100.000000
25%    1988.750000            219.550000            169.200000
50%    1997.500000            445.700000            239.650000
75%    2006.250000            723.150000            299.575000
max    2015.000000           1294.300000            386.700000

       Tobacco Price Index Relative to Retail Price Index  \
count                                          36.000000
mean                                          195.652778
std                                            66.149810
min                                           100.000000
25%                                           135.000000
```

```
50%                                                185.850000
75%                                                241.375000
max                                                334.700000

       Real Households' Disposable Income  Affordability of Tobacco Index  \
count                           36.000000                       36.000000
mean                           154.591667                       81.913889
std                             35.899251                       10.276357
min                             98.700000                       58.700000
25%                            123.300000                       78.725000
50%                            157.200000                       81.400000
75%                            190.075000                       87.225000
max                            196.400000                      103.500000

       Household Expenditure on Tobacco  Household Expenditure Total  \
count                         31.000000                 3.100000e+01
mean                       13417.451613                 6.520081e+05
std                         3796.825216                 2.859747e+05
min                         7006.000000                 2.144490e+05
25%                        10519.500000                 4.085960e+05
50%                        14047.000000                 6.394050e+05
75%                        15822.500000                 9.050715e+05
max                        19411.000000                 1.152387e+06

       Expenditure on Tobacco as a Percentage of Expenditure
count                                          31.000000
mean                                            2.241935
std                                             0.447045
min                                             1.700000
25%                                             1.800000
50%                                             2.200000
75%                                             2.500000
max                                             3.300000
```

```
print('n\ Statistics for prescriptions')
print(prescriptions.describe())
```

```
n\ Statistics for prescriptions
       All Pharmacotherapy Prescriptions  \
count                           9.000000
mean                         2191.666667
std                           406.265923
min                          1348.000000
25%                          2079.000000
50%                          2263.000000
75%                          2483.000000
max                          2564.000000

       Nicotine Replacement Therapy (NRT) Prescriptions  \
count                                          9.000000
mean                                        1441.555556
std                                          353.597621
min                                          766.000000
25%                                         1318.000000
```

```
50%                                          1541.000000
75%                                          1559.000000
max                                          1938.000000

       Bupropion (Zyban) Prescriptions  Varenicline (Champix) Prescriptions  \
count                         9.000000                             9.000000
mean                         51.777778                           698.444444
std                          36.829940                           294.100370
min                          21.000000                            22.000000
25%                          26.000000                           612.000000
50%                          36.000000                           714.000000
75%                          58.000000                           877.000000
max                         119.000000                           987.000000

       Net Ingredient Cost of All Pharmacotherapies  \
count                                      9.000000
mean                                   55856.555556
std                                     9738.846930
min                                    38145.000000
25%                                    48767.000000
50%                                    58121.000000
75%                                    63425.000000
max                                    65883.000000

       Net Ingredient Cost of Nicotine Replacement Therapies (NRT)  \
count                                          9.000000
mean                                       30003.444444
std                                         6218.555904
min                                        18208.000000
25%                                        28069.000000
50%                                        30808.000000
75%                                        31429.000000
max                                        39743.000000

       Net Ingredient Cost of Bupropion (Zyban)  \
count                                  9.000000
mean                                1984.777778
std                                 1295.995735
min                                  807.000000
25%                                  994.000000
50%                                 1581.000000
```

```
print('n\ Statistics for smokers')
print(smokers.describe())
```

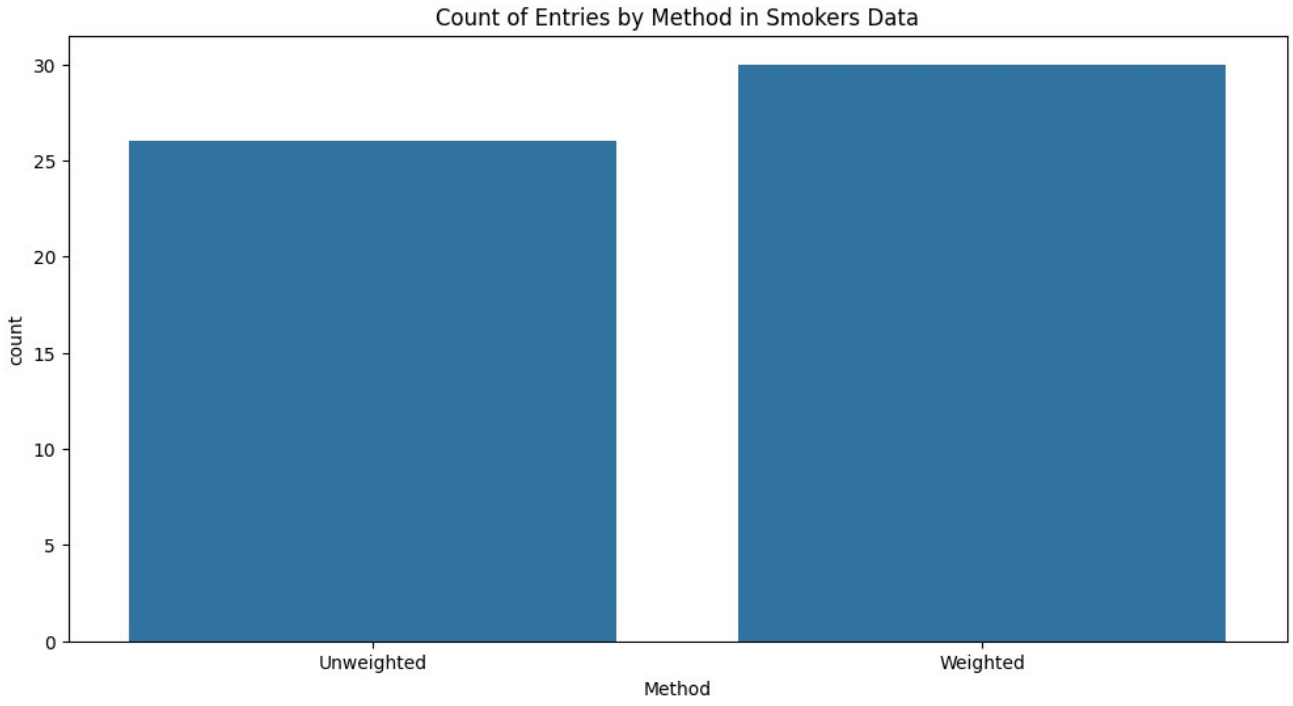
```
     n\ Statistics for smokers
                  Year  16 and Over      16-24      25-34      35-49      50-59  \
     count    56.000000    56.000000  56.000000  56.000000  56.000000  56.000000
     mean   1997.250000    27.928571  31.357143  33.732143  31.303571  28.946429
     std      12.179342     7.641649   5.992203   7.460176   8.232794   9.455003
     min    1974.000000    17.000000  20.000000  20.000000  20.000000  18.000000
     25%    1987.500000    22.000000  26.000000  28.750000  25.000000  22.000000
     50%    2000.500000    26.500000  33.000000  34.000000  29.500000  26.000000
     75%    2007.250000    31.250000  35.000000  37.250000  35.250000  34.250000
     max    2014.000000    51.000000  47.000000  55.000000  55.000000  53.000000
```

```
       60 and Over
count    56.000000
mean     18.875000
std       7.848712
min      10.000000
25%      13.000000
50%      16.000000
75%      23.000000
max      44.000000
```

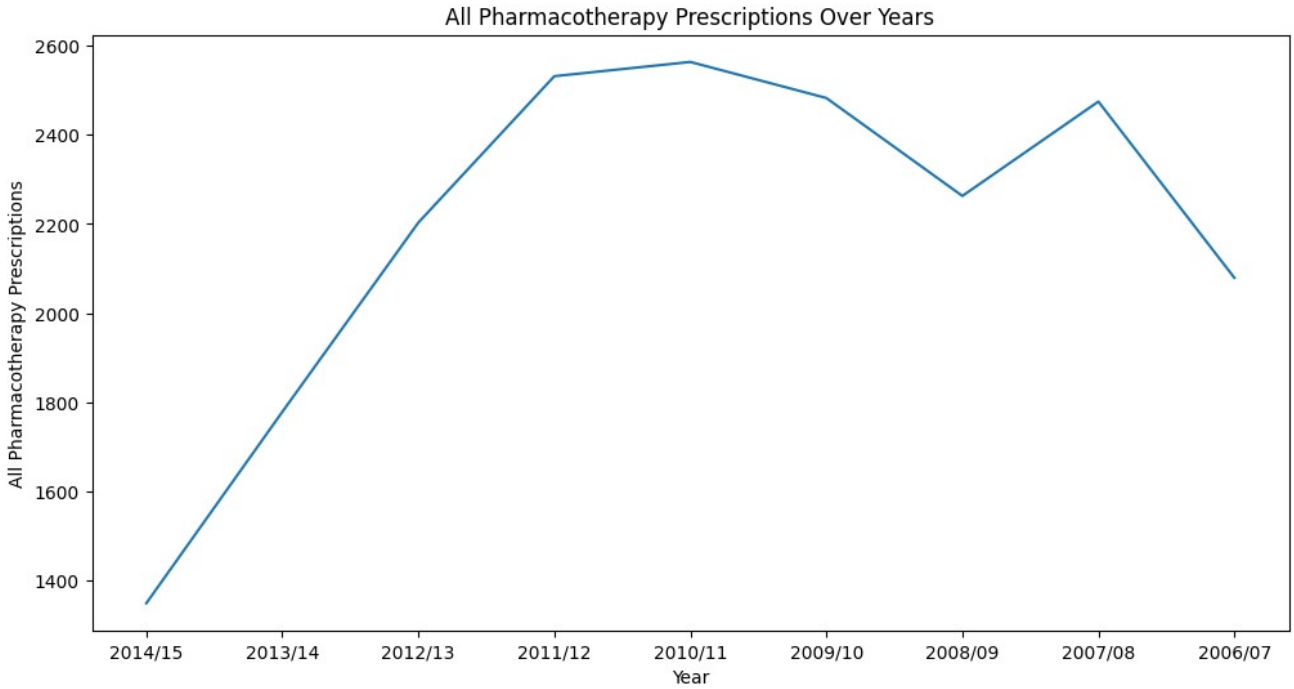
```
smokers.columns
```

```
Index(['Year', 'Method', 'Sex', '16 and Over', '16-24', '25-34', '35-49',
       '50-59', '60 and Over'],
      dtype='object')
```

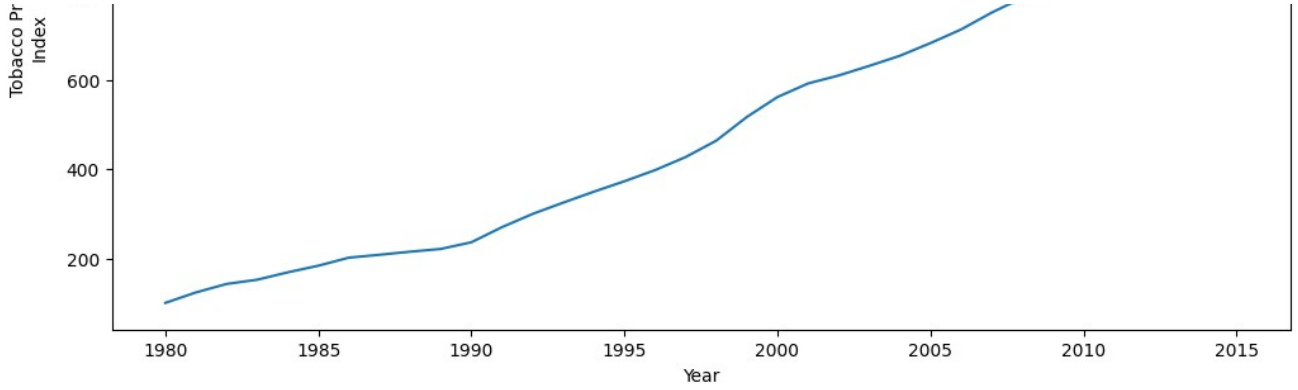
```
plt.figure(figsize=(12, 6))
sns.countplot(data=smokers, x='Method')
plt.title('Count of Entries by Method in Smokers Data')
plt.show()
```



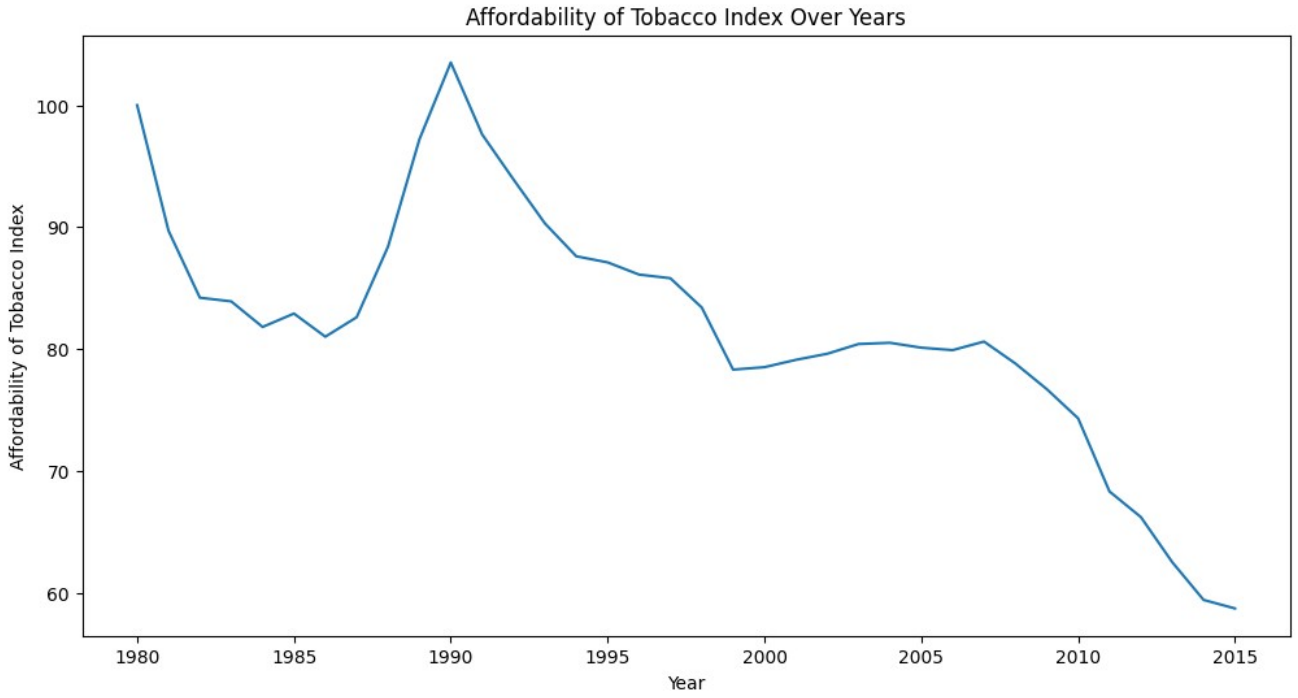
```
plt.figure(figsize=(12, 6))
sns.lineplot(data=prescriptions, x='Year', y='All Pharmacotherapy Prescriptions')
plt.title('All Pharmacotherapy Prescriptions Over Years')
plt.show()
```




```
plt.figure(figsize=(12, 6))
sns.lineplot(data=metrics, x='Year', y=metrics['Tobacco Price\nIndex'])
plt.title('Tobacco Price Index Over Years')
plt.show()
```



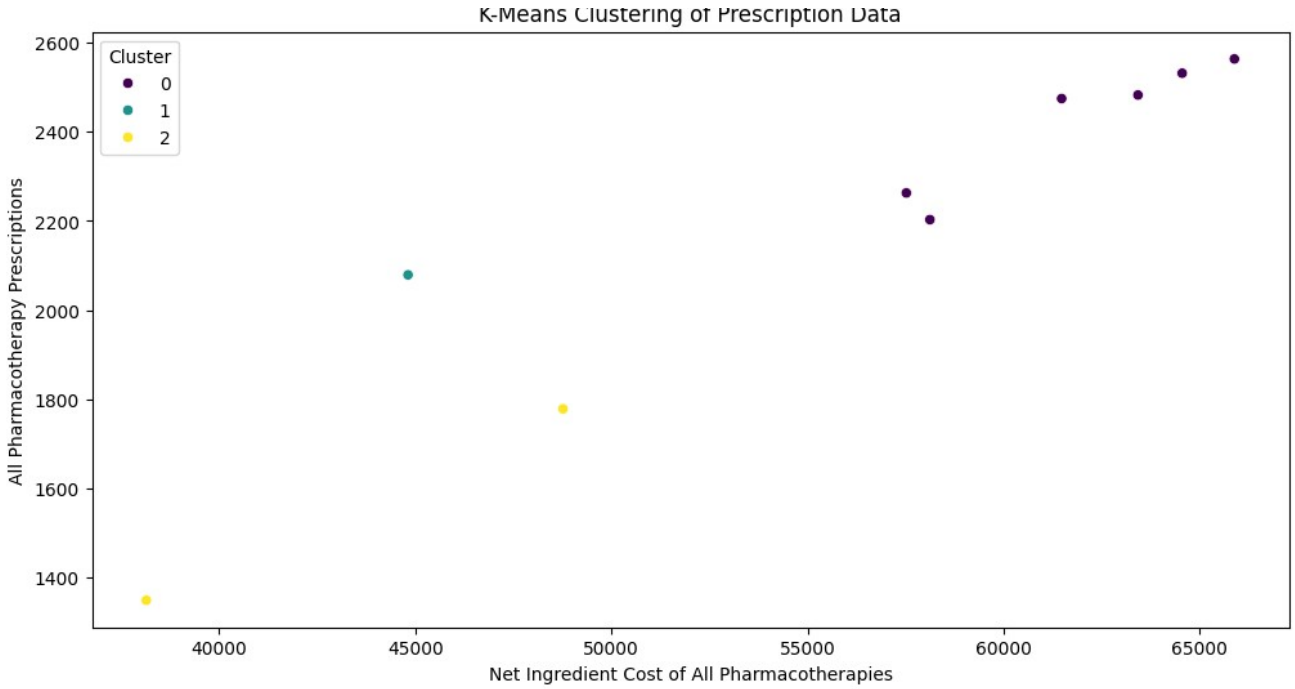

```
plt.figure(figsize=(12, 6))
sns.lineplot(data=metrics, x='Year', y='Affordability of Tobacco Index')
plt.title('Affordability of Tobacco Index Over Years')
plt.show()
```



```
from sklearn.model_selection import train_test_split

# Define features and target variable
X = prescriptions.drop(columns=['Year'])
y = prescriptions['Year']

print(y.value_counts())
```

```
Year
2014/15    1
2013/14    1
2012/13    1
2011/12    1
2010/11    1
2009/10    1
2008/09    1
2007/08    1
2006/07    1
Name: count, dtype: int64
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

X = prescriptions.drop(columns=['Year'])

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_scaled)

# Add cluster labels to the dataset
prescriptions['Cluster'] = kmeans.labels_

# Visualize the clusters
plt.figure(figsize=(12, 6))
sns.scatterplot(data=prescriptions, x='Net Ingredient Cost of All Pharmacotherapies', y='
plt.title('K-Means Clustering of Prescription Data')
plt.show()
```




```
from sklearn.model_selection import train_test_split

# Define features and target variable
X = prescriptions.drop(columns=['Year'])
y = prescriptions['Year']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train a logistic regression model
model = LogisticRegression()
```

```
model.fit(X_train_scaled, y_train)
```

```
from sklearn.metrics import classification_report, confusion_matrix

# Make predictions on the test set
y_pred = model.predict(X_test_scaled)

# Evaluate the model
print("Classification Report:")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

     2007/08       0.00      0.00      0.00       1.0
     2009/10       0.00      0.00      0.00       0.0
     2013/14       0.00      0.00      0.00       1.0
     2014/15       0.00      0.00      0.00       0.0

    accuracy                           0.00       2.0
   macro avg       0.00      0.00      0.00       2.0
weighted avg       0.00      0.00      0.00       2.0

Confusion Matrix:
[[0 1 0 0]
 [0 0 0 0]
 [0 0 0 1]
 [0 0 0 0]]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```