OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH

DUBLIN

TECHNOLOGICAL UNIVERSITY DUBLIN

DATA MINING ASSIGNMENT 2

ANALYSIS AND MODELLING OF PHARMACEUTICAL DATA FOR OPINION MINING AND DRUG CHARACTERISTICS

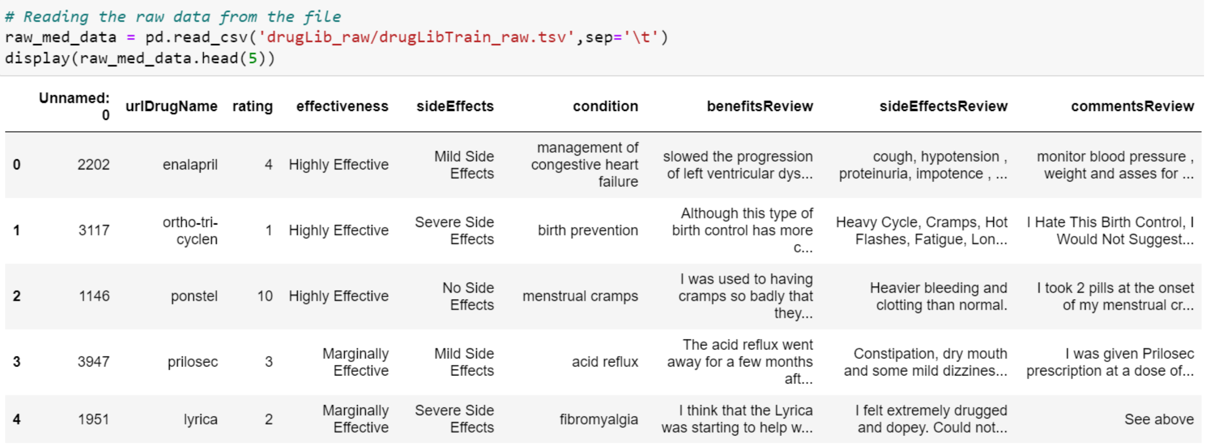
PYTHON CODE APPENDIX

Student Name	Vyoma Mohan
Student Id	D22124454
Subject	Data Mining
Subject Code	DATA 9900
Course	TU059
Stream	Data Science (DS)
Year	First year
Assignment	Assignment 2
Deadline	8th Jan 2023

This document contains only python code snippets. For rest, please refer report.

Python code related to importing and cleaning the training dataset:

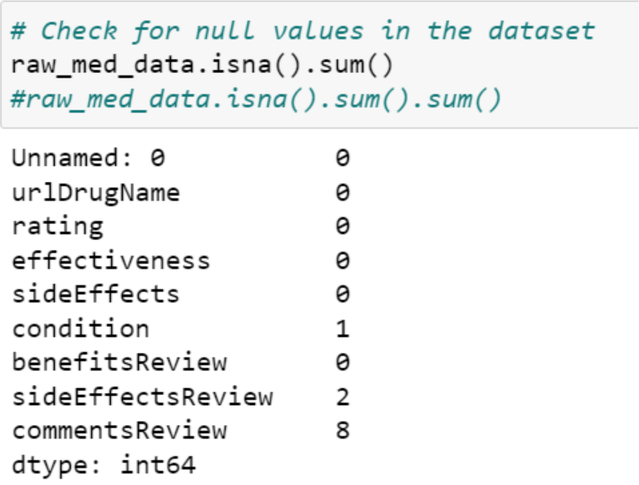
[1] Reading the excel file



```
# Reading the raw data from the file
raw_med_data = pd.read_csv('drugLib_raw/drugLibTrain_raw.tsv',sep='\t')
display(raw_med_data.head(5))
```

	Unnamed: 0	urlDrugName	rating	effectiveness	sideEffects	condition	benefitsReview	sideEffectsReview	commentsReview
0	2202	enalapril	4	Highly Effective	Mild Side Effects	management of congestive heart failure	slowed the progression of left ventricular dys...	cough, hypotension , proteinuria, impotence , ...	monitor blood pressure , weight and asses for ...
1	3117	ortho-tri-cyclen	1	Highly Effective	Severe Side Effects	birth prevention	Although this type of birth control has more c...	Heavy Cycle, Cramps, Hot Flashes, Fatigue, Lon...	I Hate This Birth Control, I Would Not Suggest...
2	1146	ponstel	10	Highly Effective	No Side Effects	menstrual cramps	I was used to having cramps so badly that they...	Heavier bleeding and clotting than normal.	I took 2 pills at the onset of my menstrual cr...
3	3947	prilosec	3	Marginally Effective	Mild Side Effects	acid reflux	The acid reflux went away for a few months aft...	Constipation, dry mouth and some mild dizzines...	I was given Prilosec prescription at a dose of...
4	1951	lyrica	2	Marginally Effective	Severe Side Effects	fibromyalgia	I think that the Lyrica was starting to help w...	I felt extremely drugged and dopey. Could not...	See above

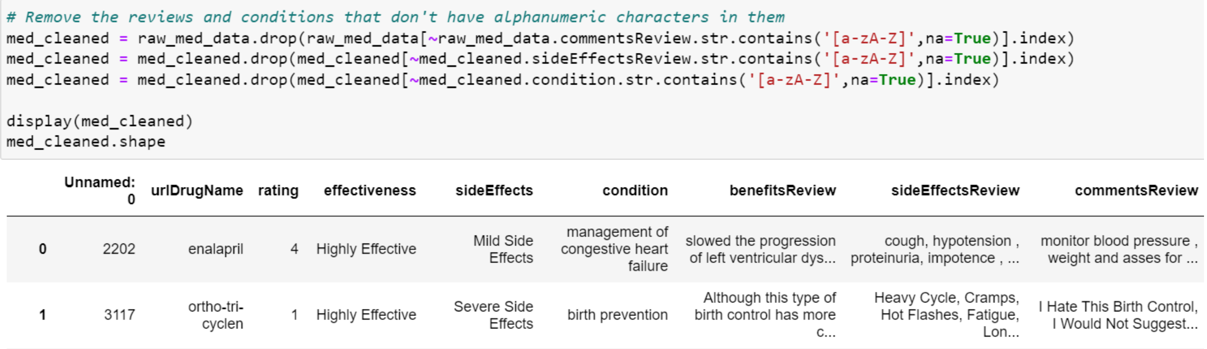
[2] Checking for null values



```
# Check for null values in the dataset
raw_med_data.isna().sum()
#raw_med_data.isna().sum()
```

```
Unnamed: 0           0
urlDrugName          0
rating               0
effectiveness        0
sideEffects          0
condition            1
benefitsReview       0
sideEffectsReview    2
commentsReview       8
dtype: int64
```

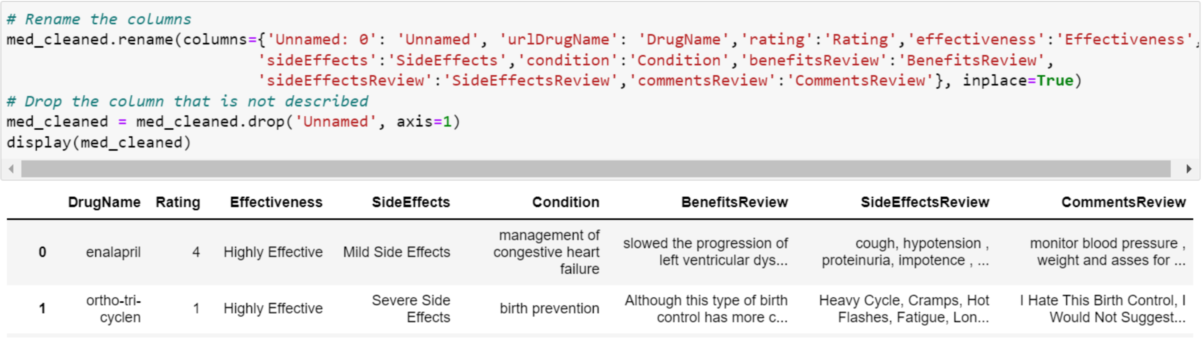
[3] Remove the reviews without alphanumeric characters



```
# Remove the reviews and conditions that don't have alphanumeric characters in them
med_cleaned = raw_med_data.drop(raw_med_data[~raw_med_data.commentsReview.str.contains('[a-zA-Z]',na=True)].index)
med_cleaned = med_cleaned.drop(med_cleaned[~med_cleaned.sideEffectsReview.str.contains('[a-zA-Z]',na=True)].index)
med_cleaned = med_cleaned.drop(med_cleaned[~med_cleaned.condition.str.contains('[a-zA-Z]',na=True)].index)

display(med_cleaned)
med_cleaned.shape
```

	Unnamed: 0	urlDrugName	rating	effectiveness	sideEffects	condition	benefitsReview	sideEffectsReview	commentsReview
0	2202	enalapril	4	Highly Effective	Mild Side Effects	management of congestive heart failure	slowed the progression of left ventricular dys...	cough, hypotension , proteinuria, impotence , ...	monitor blood pressure , weight and asses for ...
1	3117	ortho-tri-cyclen	1	Highly Effective	Severe Side Effects	birth prevention	Although this type of birth control has more c...	Heavy Cycle, Cramps, Hot Flashes, Fatigue, Lon...	I Hate This Birth Control, I Would Not Suggest...

[4] Rename columns and drop unnecessary columns

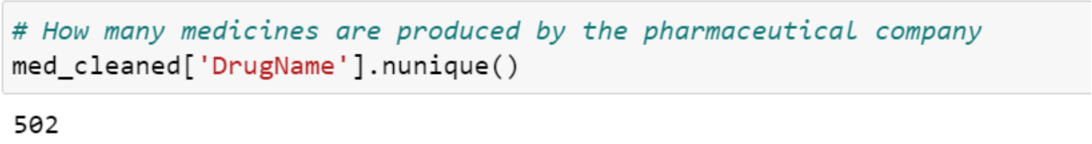


```
# Rename the columns
med_cleaned.rename(columns={'Unnamed: 0': 'Unnamed', 'urlDrugName': 'DrugName','rating':'Rating','effectiveness':'Effectiveness',
                            'sideEffects':'SideEffects','condition':'Condition','benefitsReview':'BenefitsReview',
                            'sideEffectsReview':'SideEffectsReview','commentsReview':'CommentsReview'}, inplace=True)
# Drop the column that is not described
med_cleaned = med_cleaned.drop('Unnamed', axis=1)
display(med_cleaned)
```

	DrugName	Rating	Effectiveness	SideEffects	Condition	BenefitsReview	SideEffectsReview	CommentsReview
0	enalapril	4	Highly Effective	Mild Side Effects	management of congestive heart failure	slowed the progression of left ventricular dys...	cough, hypotension , proteinuria, impotence , ...	monitor blood pressure , weight and asses for ...
1	ortho-tri-cyclen	1	Highly Effective	Severe Side Effects	birth prevention	Although this type of birth control has more c...	Heavy Cycle, Cramps, Hot Flashes, Fatigue, Lon...	I Hate This Birth Control, I Would Not Suggest...

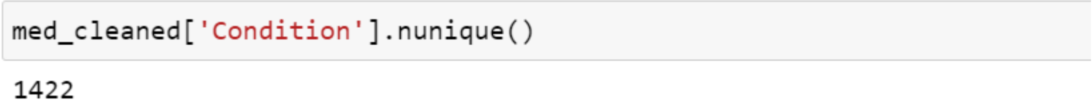
Analysis screenshots

[1] Check unique number of drugs and conditions



```
# How many medicines are produced by the pharmaceutical company
med_cleaned['DrugName'].nunique()
```

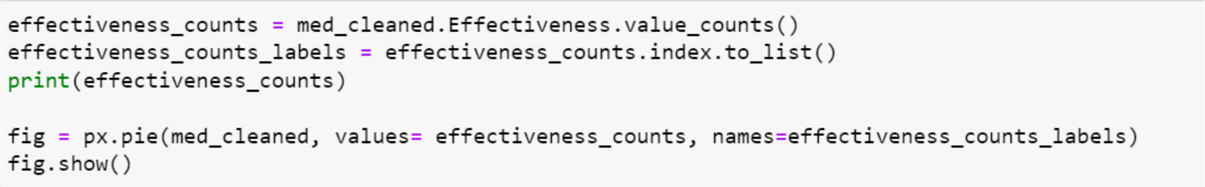
```
502
```



```
med_cleaned['Condition'].nunique()
```

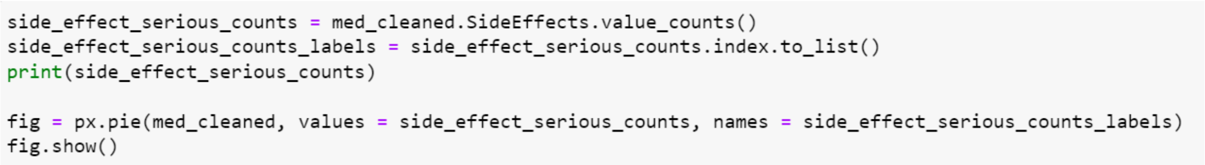
```
1422
```

[2] Distribution of effectiveness with pie chart



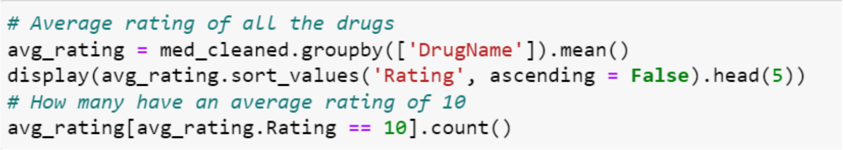
```
effectiveness_counts = med_cleaned.Effectiveness.value_counts()
effectiveness_counts_labels = effectiveness_counts.index.to_list()
print(effectiveness_counts)

fig = px.pie(med_cleaned, values= effectiveness_counts, names=effectiveness_counts_labels)
fig.show()
```

[3] Distribution of side effects with pie chart



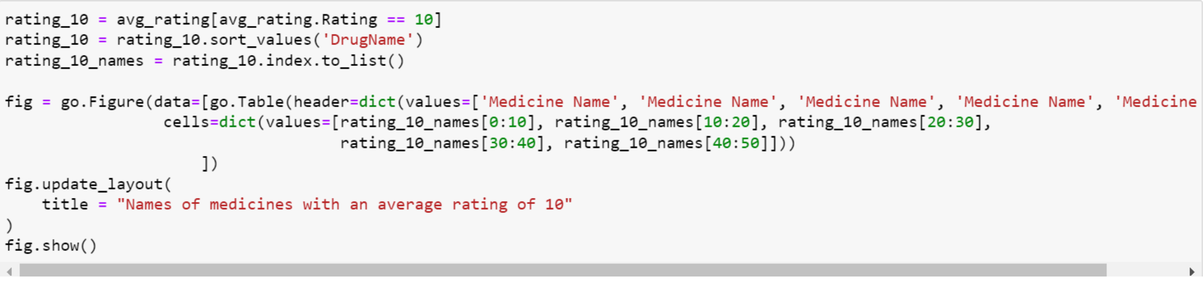
```
side_effect_serious_counts = med_cleaned.SideEffects.value_counts()
side_effect_serious_counts_labels = side_effect_serious_counts.index.to_list()
print(side_effect_serious_counts)

fig = px.pie(med_cleaned, values = side_effect_serious_counts, names = side_effect_serious_counts_labels)
fig.show()
```

[4] How many drugs have an average rating of 10



```
# Average rating of all the drugs
avg_rating = med_cleaned.groupby(['DrugName']).mean()
display(avg_rating.sort_values('Rating', ascending = False).head(5))
# How many have an average rating of 10
avg_rating[avg_rating.Rating == 10].count()
```

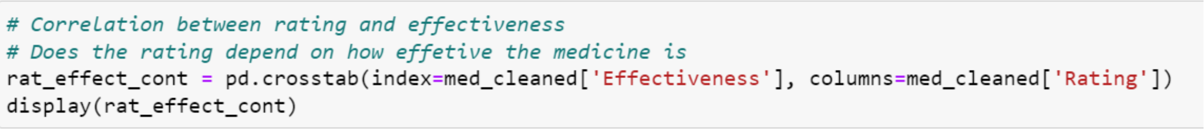
[5] Name the medicines with 10 rating



```
rating_10 = avg_rating[avg_rating.Rating == 10]
rating_10 = rating_10.sort_values('DrugName')
rating_10_names = rating_10.index.to_list()

fig = go.Figure(data=[go.Table(header=dict(values=['Medicine Name', 'Medicine Name', 'Medicine Name', 'Medicine Name', 'Medicine
                cells=dict(values=[rating_10_names[0:10], rating_10_names[10:20], rating_10_names[20:30],
                                  rating_10_names[30:40], rating_10_names[40:50]]))
                    ])
fig.update_layout(
    title = "Names of medicines with an average rating of 10"
)
fig.show()
```

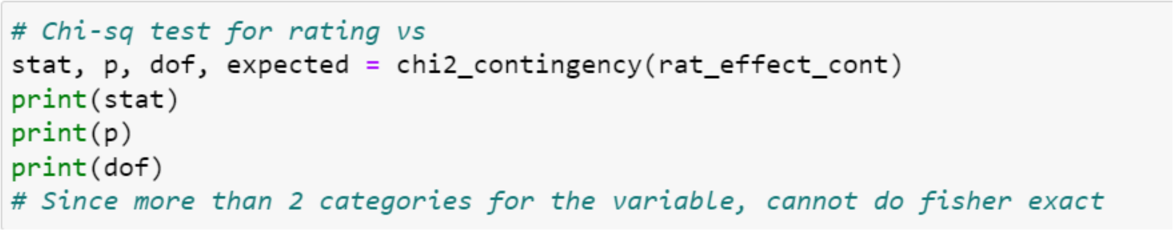
[6] Correlation between rating and effectiveness

Contingency table:



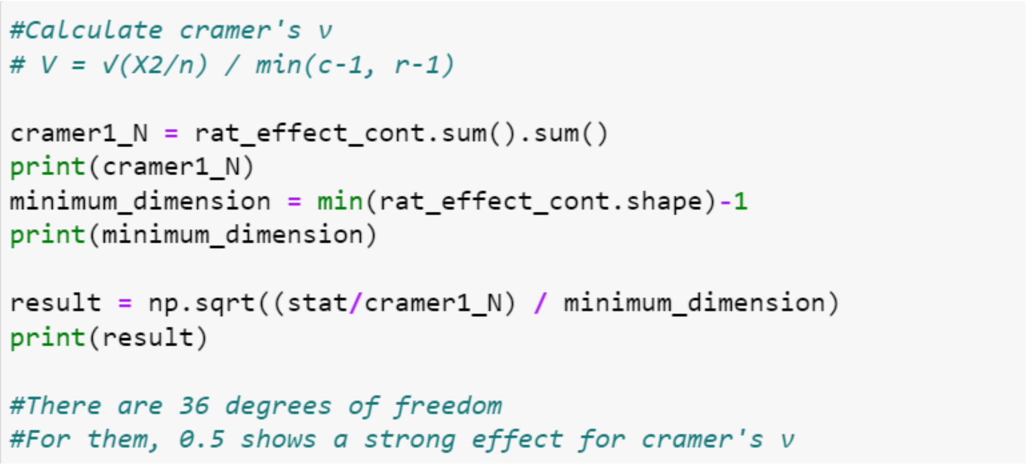
```
# Correlation between rating and effectiveness
# Does the rating depend on how effetive the medicine is
rat_effect_cont = pd.crosstab(index=med_cleaned['Effectiveness'], columns=med_cleaned['Rating'])
display(rat_effect_cont)
```

Chi square:



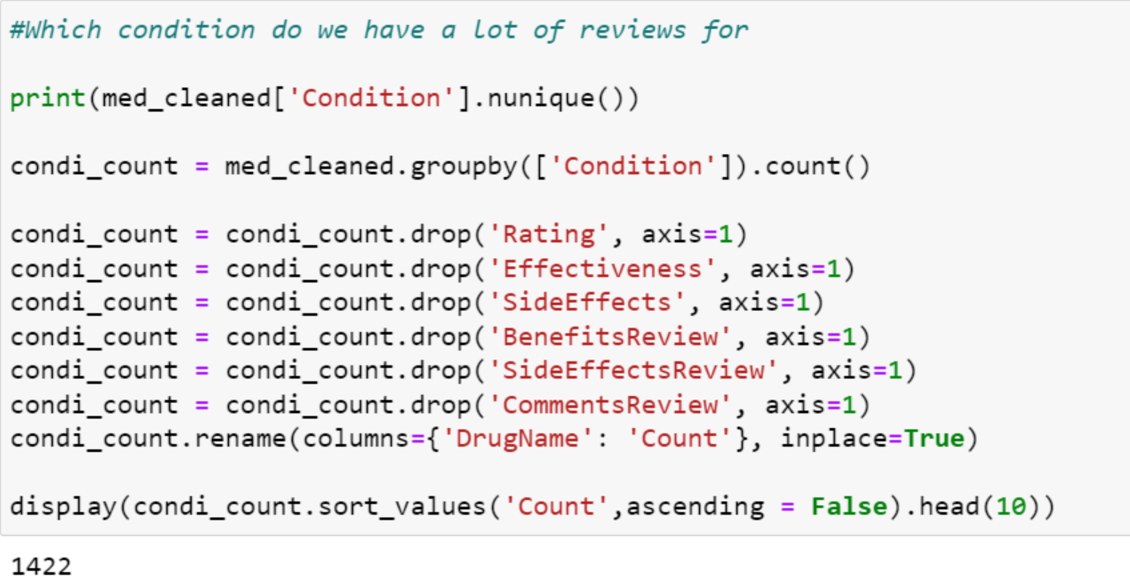
```
# Chi-sq test for rating vs
stat, p, dof, expected = chi2_contingency(rat_effect_cont)
print(stat)
print(p)
print(dof)
# Since more than 2 categories for the variable, cannot do fisher exact
```

Cramer v:



```
#Calculate cramer's v
# V = √(X2/n) / min(c-1, r-1)

cramer1_N = rat_effect_cont.sum().sum()
print(cramer1_N)
minimum_dimension = min(rat_effect_cont.shape)-1
print(minimum_dimension)

result = np.sqrt((stat/cramer1_N) / minimum_dimension)
print(result)

#There are 36 degrees of freedom
#For them, 0.5 shows a strong effect for cramer's v
```

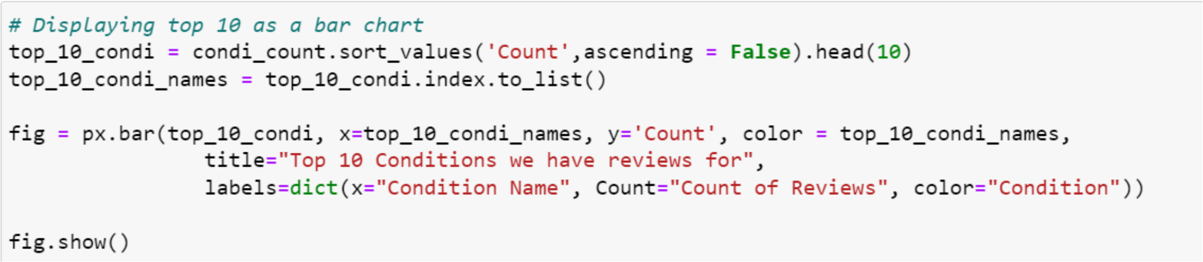
[7] Which condition do we have a lot of reviews for:



```
#Which condition do we have a lot of reviews for

print(med_cleaned['Condition'].nunique())

condi_count = med_cleaned.groupby(['Condition']).count()

condi_count = condi_count.drop('Rating', axis=1)
condi_count = condi_count.drop('Effectiveness', axis=1)
condi_count = condi_count.drop('SideEffects', axis=1)
condi_count = condi_count.drop('BenefitsReview', axis=1)
condi_count = condi_count.drop('SideEffectsReview', axis=1)
condi_count = condi_count.drop('CommentsReview', axis=1)
condi_count.rename(columns={'DrugName': 'Count'}, inplace=True)

display(condi_count.sort_values('Count',ascending = False).head(10))
```

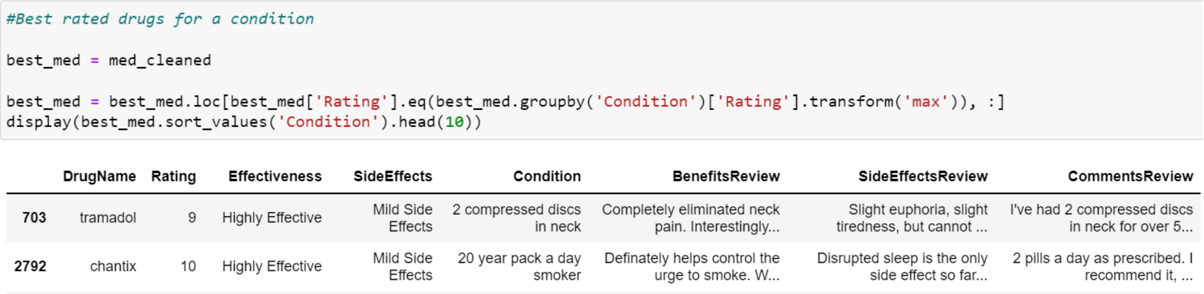
```
1422
```

Displaying top 10 as a bar chart:



```
# Displaying top 10 as a bar chart
top_10_condi = condi_count.sort_values('Count',ascending = False).head(10)
top_10_condi_names = top_10_condi.index.to_list()

fig = px.bar(top_10_condi, x=top_10_condi_names, y='Count', color = top_10_condi_names,
                title="Top 10 Conditions we have reviews for",
                labels=dict(x="Condition Name", Count="Count of Reviews", color="Condition"))

fig.show()
```

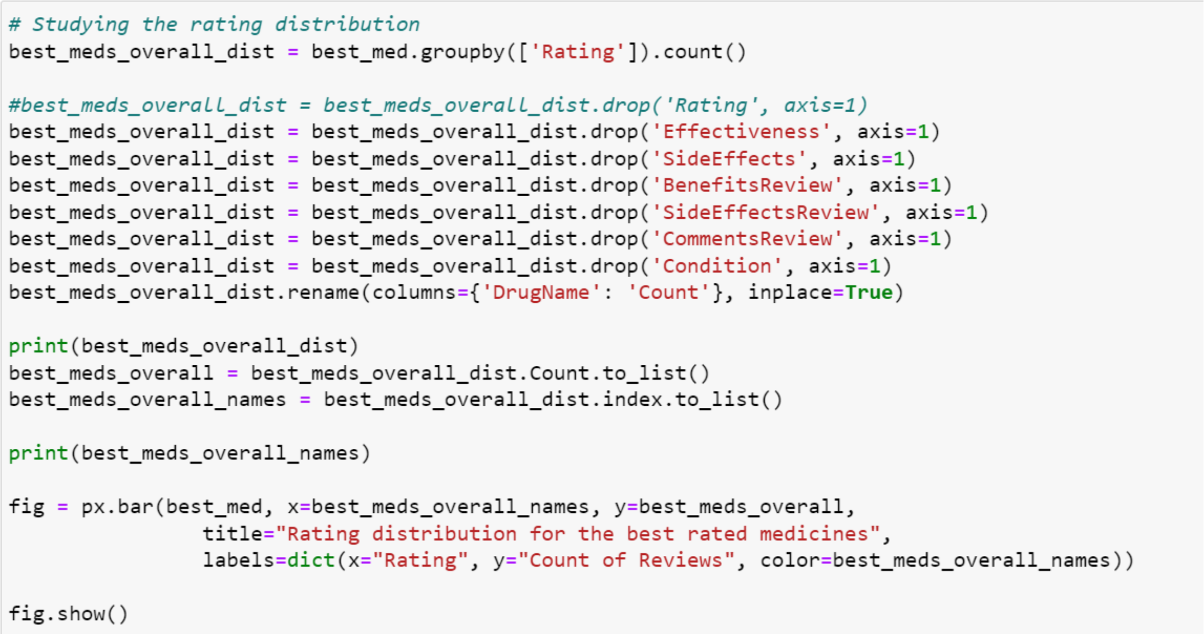
[8] Finding the best drugs for a condition



```
#Best rated drugs for a condition

best_med = med_cleaned

best_med = best_med.loc[best_med['Rating'].eq(best_med.groupby('Condition')['Rating'].transform('max')), :]
display(best_med.sort_values('Condition').head(10))
```

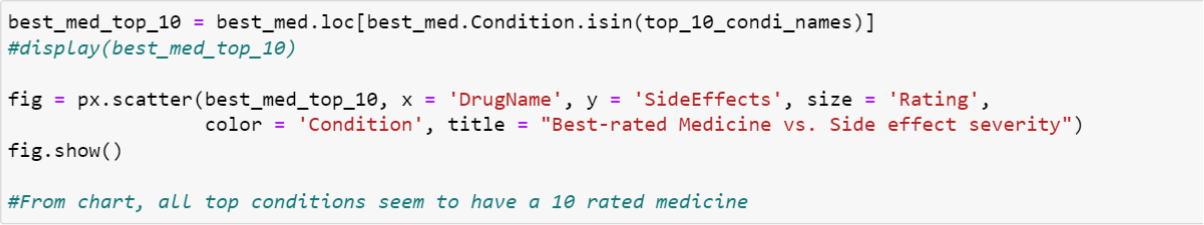
	DrugName	Rating	Effectiveness	SideEffects	Condition	BenefitsReview	SideEffectsReview	CommentsReview
703	tramadol	9	Highly Effective	Mild Side Effects	2 compressed discs in neck	Completely eliminated neck pain. Interestingly...	Slight euphoria, slight tiredness, but cannot ...	I've had 2 compressed discs in neck for over 5...
2792	chantix	10	Highly Effective	Mild Side Effects	20 year pack a day smoker	Definately helps control the urge to smoke. W...	Disrupted sleep is the only side effect so far...	2 pills a day as prescribed. I recommend it, ...

Studying the rating distribution of best drugs:



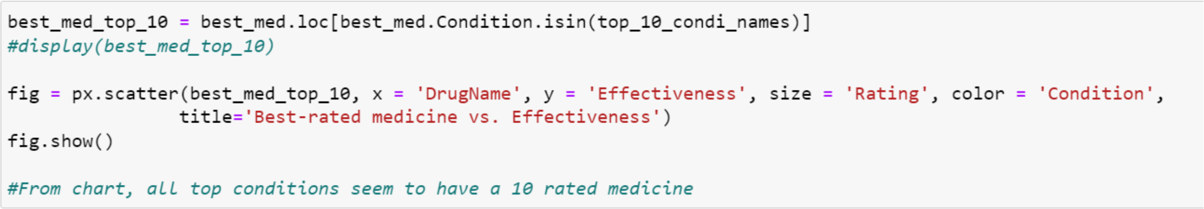
```
# Studying the rating distribution
best_meds_overall_dist = best_med.groupby(['Rating']).count()

#best_meds_overall_dist = best_meds_overall_dist.drop('Rating', axis=1)
best_meds_overall_dist = best_meds_overall_dist.drop('Effectiveness', axis=1)
best_meds_overall_dist = best_meds_overall_dist.drop('SideEffects', axis=1)
best_meds_overall_dist = best_meds_overall_dist.drop('BenefitsReview', axis=1)
best_meds_overall_dist = best_meds_overall_dist.drop('SideEffectsReview', axis=1)
best_meds_overall_dist = best_meds_overall_dist.drop('CommentsReview', axis=1)
best_meds_overall_dist = best_meds_overall_dist.drop('Condition', axis=1)
best_meds_overall_dist.rename(columns={'DrugName': 'Count'}, inplace=True)

print(best_meds_overall_dist)
best_meds_overall = best_meds_overall_dist.Count.to_list()
best_meds_overall_names = best_meds_overall_dist.index.to_list()

print(best_meds_overall_names)

fig = px.bar(best_med, x=best_meds_overall_names, y=best_meds_overall,
             title="Rating distribution for the best rated medicines",
             labels=dict(x="Rating", y="Count of Reviews", color=best_meds_overall_names))

fig.show()
```

Plot best medicine vs. severity:



```
best_med_top_10 = best_med.loc[best_med.Condition.isin(top_10_condi_names)]
#display(best_med_top_10)

fig = px.scatter(best_med_top_10, x = 'DrugName', y = 'SideEffects', size = 'Rating',
                 color = 'Condition', title = "Best-rated Medicine vs. Side effect severity")
fig.show()

#From chart, all top conditions seem to have a 10 rated medicine
```

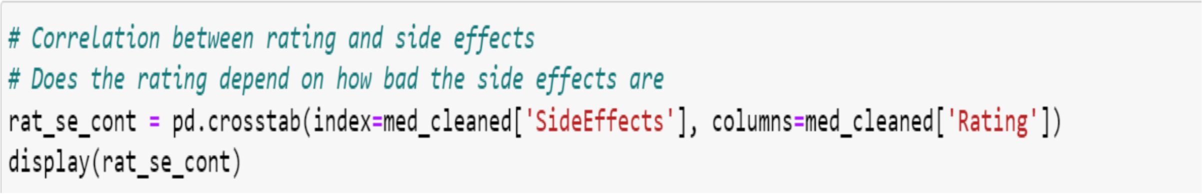
Plot best medicine vs. Effectiveness:



```
best_med_top_10 = best_med.loc[best_med.Condition.isin(top_10_condi_names)]
#display(best_med_top_10)

fig = px.scatter(best_med_top_10, x = 'DrugName', y = 'Effectiveness', size = 'Rating', color = 'Condition',
                 title='Best-rated medicine vs. Effectiveness')
fig.show()

#From chart, all top conditions seem to have a 10 rated medicine
```

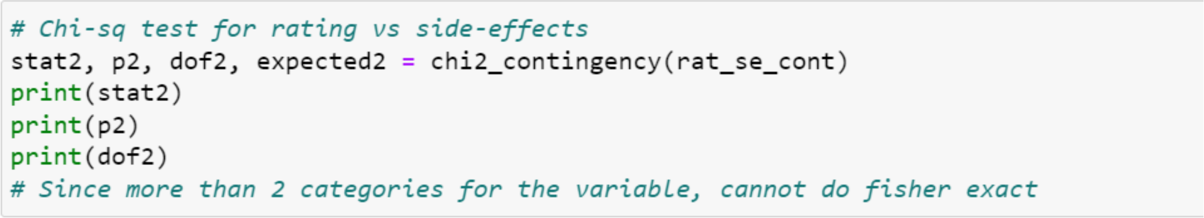
[9] Correlation: rating and side effects

Contingency table:



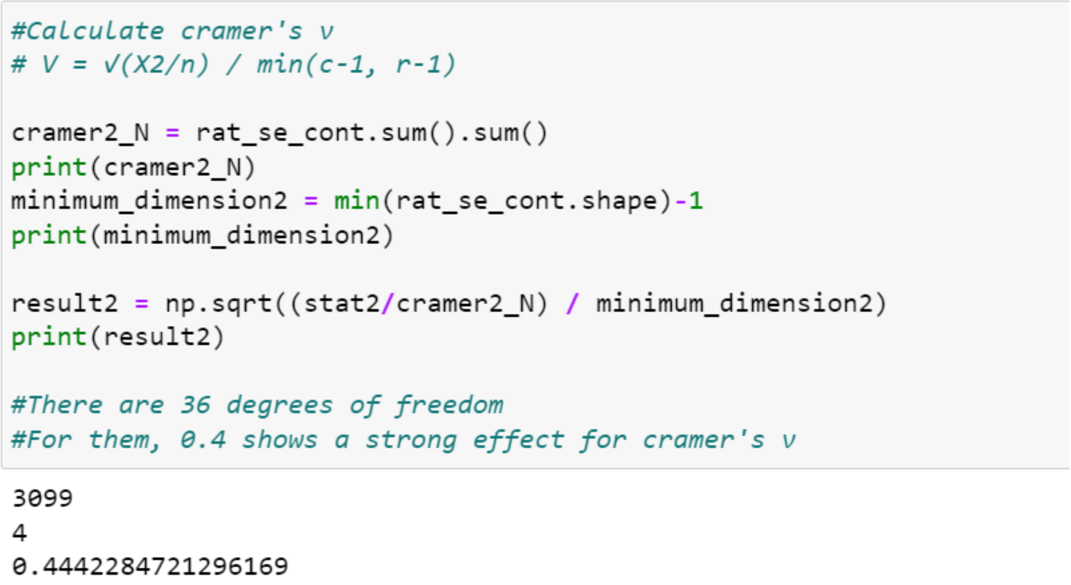
```
# Correlation between rating and side effects
# Does the rating depend on how bad the side effects are
rat_se_cont = pd.crosstab(index=med_cleaned['SideEffects'], columns=med_cleaned['Rating'])
display(rat_se_cont)
```

Chisq:



```
# Chi-sq test for rating vs side-effects
stat2, p2, dof2, expected2 = chi2_contingency(rat_se_cont)
print(stat2)
print(p2)
print(dof2)
# Since more than 2 categories for the variable, cannot do fisher exact
```

Cramer V:

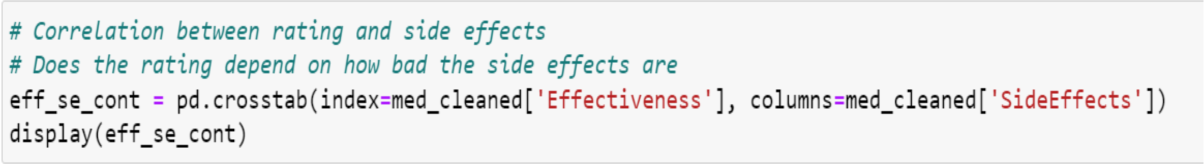


```
#Calculate cramer's v
# V = √(X2/n) / min(c-1, r-1)

cramer2_N = rat_se_cont.sum().sum()
print(cramer2_N)
minimum_dimension2 = min(rat_se_cont.shape)-1
print(minimum_dimension2)

result2 = np.sqrt((stat2/cramer2_N) / minimum_dimension2)
print(result2)

#There are 36 degrees of freedom
#For them, 0.4 shows a strong effect for cramer's v
```

```
3099
4
0.4442284721296169
```

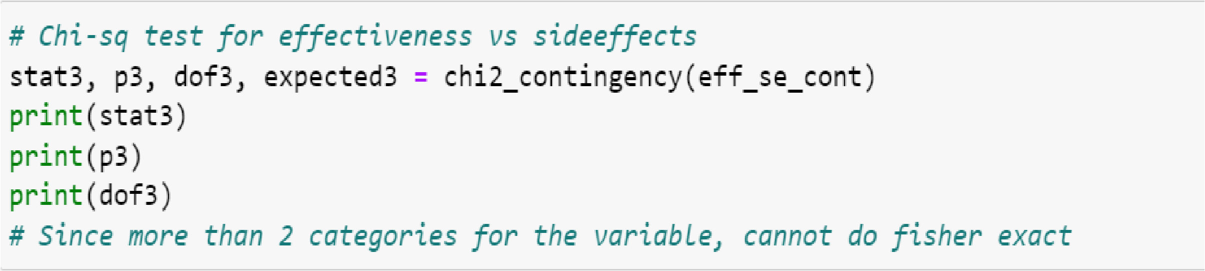
[10] Correlation: Effectiveness and side effects

Contingency table:



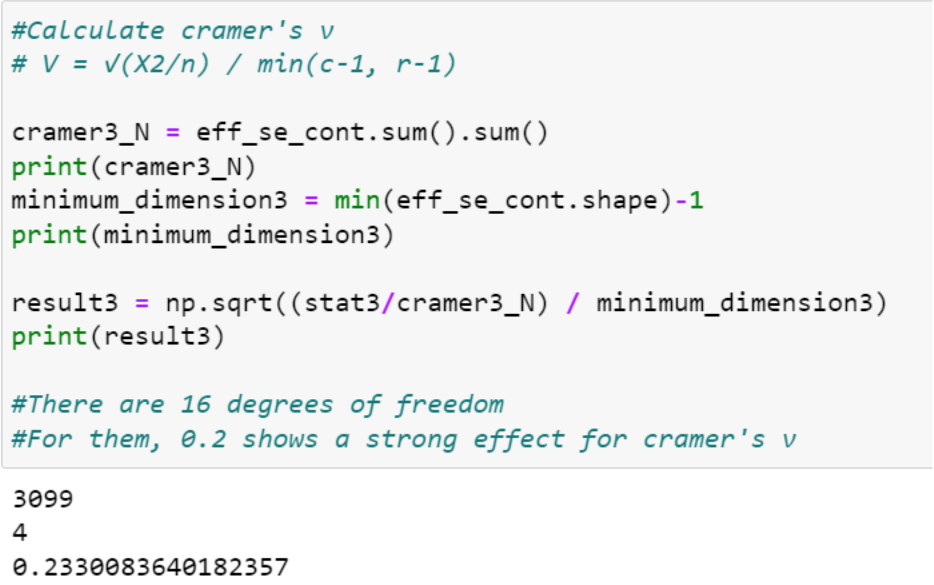
```
# Correlation between rating and side effects
# Does the rating depend on how bad the side effects are
eff_se_cont = pd.crosstab(index=med_cleaned['Effectiveness'], columns=med_cleaned['SideEffects'])
display(eff_se_cont)
```

Chisq:



```
# Chi-sq test for effectiveness vs sideeffects
stat3, p3, dof3, expected3 = chi2_contingency(eff_se_cont)
print(stat3)
print(p3)
print(dof3)
# Since more than 2 categories for the variable, cannot do fisher exact
```

Cramer V:

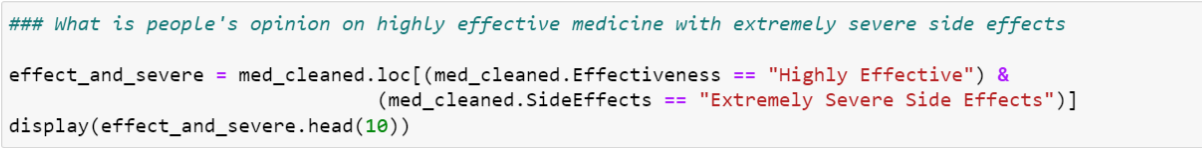


```
#Calculate cramer's v
# V = √(X2/n) / min(c-1, r-1)

cramer3_N = eff_se_cont.sum().sum()
print(cramer3_N)
minimum_dimension3 = min(eff_se_cont.shape)-1
print(minimum_dimension3)

result3 = np.sqrt((stat3/cramer3_N) / minimum_dimension3)
print(result3)

#There are 16 degrees of freedom
#For them, 0.2 shows a strong effect for cramer's v
```

```
3099
4
0.2330083640182357
```

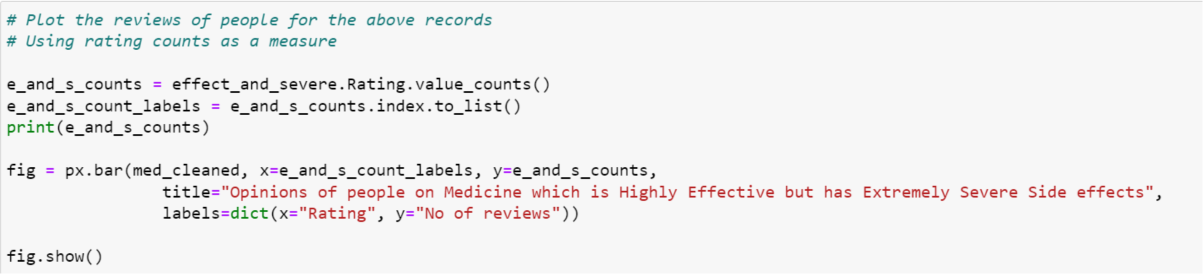
[11] Analyse highly effective but extremely severe side effects

Filter medicines:



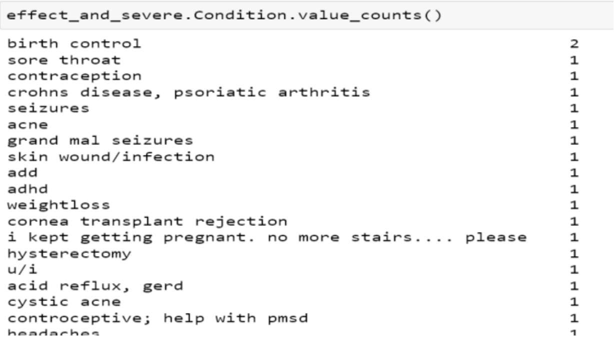
```
### What is people's opinion on highly effective medicine with extremely severe side effects

effect_and_severe = med_cleaned.loc[(med_cleaned.Effectiveness == "Highly Effective") &
                                    (med_cleaned.SideEffects == "Extremely Severe Side Effects")]
display(effect_and_severe.head(10))
```

Plot ratings as bar graph:



```
# Plot the reviews of people for the above records
# Using rating counts as a measure

e_and_s_counts = effect_and_severe.Rating.value_counts()
e_and_s_count_labels = e_and_s_counts.index.to_list()
print(e_and_s_counts)

fig = px.bar(med_cleaned, x=e_and_s_count_labels, y=e_and_s_counts,
                title="Opinions of people on Medicine which is Highly Effective but has Extremely Severe Side effects",
                labels=dict(x="Rating", y="No of reviews"))

fig.show()
```

Count of conditions in this category:

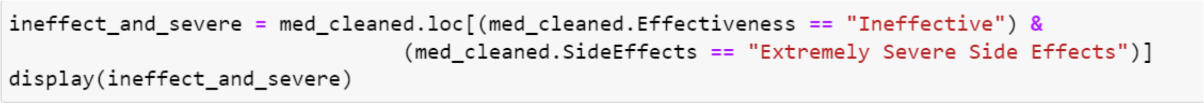


```
effect_and_severe.Condition.value_counts()
```

```
birth control                                           2
sore throat                                             1
contraception                                           1
crohns disease, psoriatic arthritis                     1
seizures                                                1
acne                                                    1
grand mal seizures                                      1
skin wound/infection                                    1
add                                                     1
adhd                                                    1
weightloss                                              1
cornea transplant rejection                             1
i kept getting pregnant. no more stairs.... please      1
hysterectomy                                            1
u/i                                                     1
acid reflux, gerd                                       1
cystic acne                                             1
controceptive; help with pmsd                           1
headaches                                               1
```

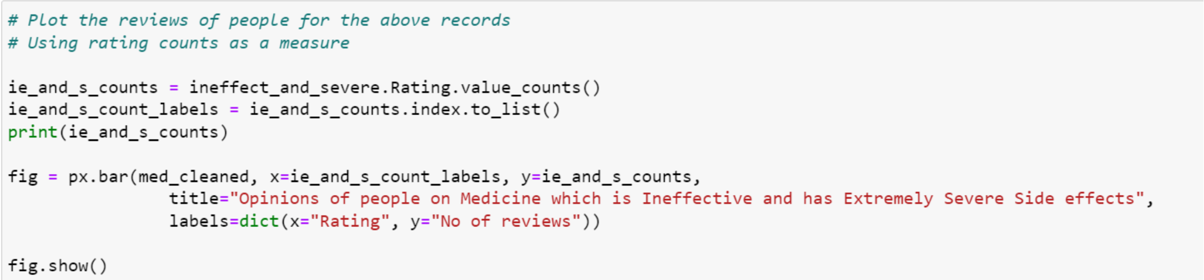
[12] Ineffective and extremely severe medicines:

Filter medicines:



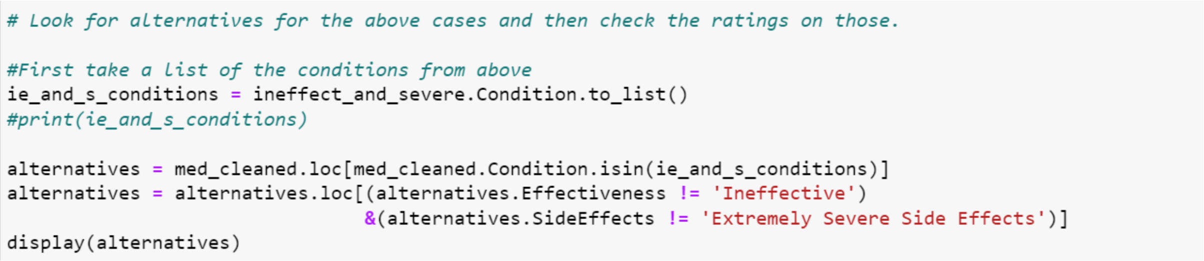
```
ineffect_and_severe = med_cleaned.loc[(med_cleaned.Effectiveness == "Ineffective") &
                                      (med_cleaned.SideEffects == "Extremely Severe Side Effects")]
display(ineffect_and_severe)
```

Plot ratings as bar graphs:



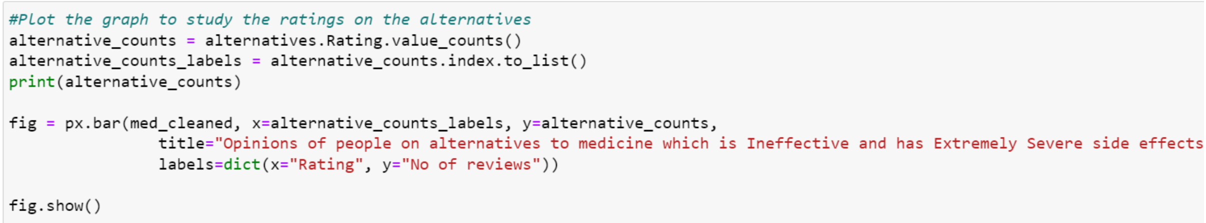
```
# Plot the reviews of people for the above records
# Using rating counts as a measure

ie_and_s_counts = ineffect_and_severe.Rating.value_counts()
ie_and_s_count_labels = ie_and_s_counts.index.to_list()
print(ie_and_s_counts)

fig = px.bar(med_cleaned, x=ie_and_s_count_labels, y=ie_and_s_counts,
             title="Opinions of people on Medicine which is Ineffective and has Extremely Severe Side effects",
             labels=dict(x="Rating", y="No of reviews"))

fig.show()
```

Look for alternatives for such medicines:



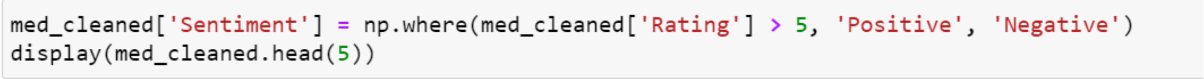
```
# Look for alternatives for the above cases and then check the ratings on those.

#First take a list of the conditions from above
ie_and_s_conditions = ineffect_and_severe.Condition.to_list()
#print(ie_and_s_conditions)

alternatives = med_cleaned.loc[med_cleaned.Condition.isin(ie_and_s_conditions)]
alternatives = alternatives.loc[(alternatives.Effectiveness != 'Ineffective')
                                &(alternatives.SideEffects != 'Extremely Severe Side Effects')]
display(alternatives)
```

Plot ratings as bar graph:



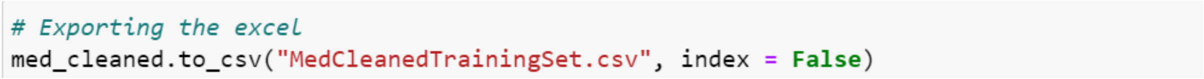
```
#Plot the graph to study the ratings on the alternatives
alternative_counts = alternatives.Rating.value_counts()
alternative_counts_labels = alternative_counts.index.to_list()
print(alternative_counts)

fig = px.bar(med_cleaned, x=alternative_counts_labels, y=alternative_counts,
             title="Opinions of people on alternatives to medicine which is Ineffective and has Extremely Severe side effects
             labels=dict(x="Rating", y="No of reviews"))

fig.show()
```

[13] Adding sentiment column for ML:



```
med_cleaned['Sentiment'] = np.where(med_cleaned['Rating'] > 5, 'Positive', 'Negative')
display(med_cleaned.head(5))
```

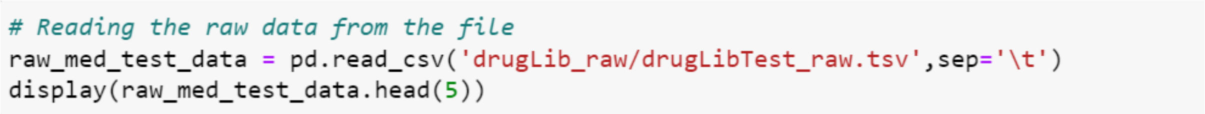
Exporting the training data after cleaning:



```
# Exporting the excel
med_cleaned.to_csv("MedCleanedTrainingSet.csv", index = False)
```

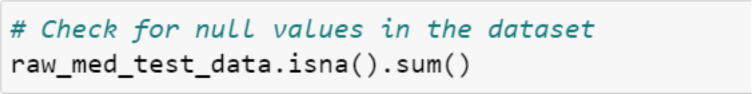
Cleaning and exporting the test set:

Reading the data:



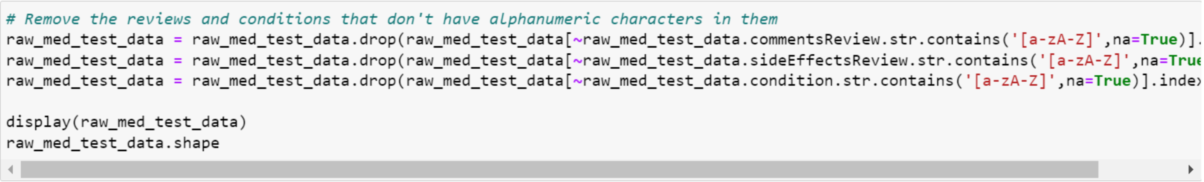
```
# Reading the raw data from the file
raw_med_test_data = pd.read_csv('drugLib_raw/drugLibTest_raw.tsv',sep='\t')
display(raw_med_test_data.head(5))
```

Null checks on data:



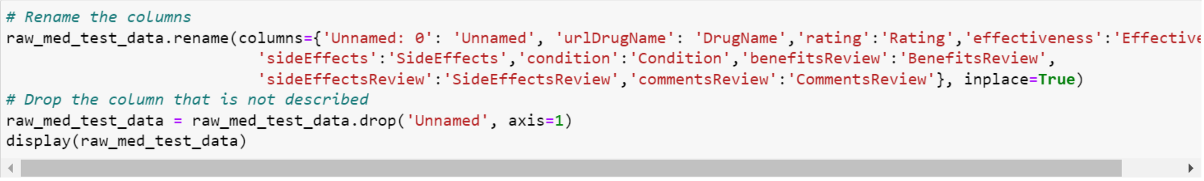
```
# Check for null values in the dataset
raw_med_test_data.isna().sum()
```

Remove non alpha numeric reviews:



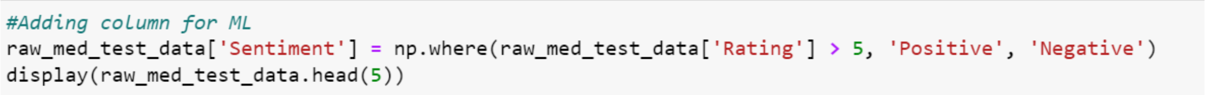
```
# Remove the reviews and conditions that don't have alphanumeric characters in them
raw_med_test_data = raw_med_test_data.drop(raw_med_test_data[~raw_med_test_data.commentsReview.str.contains('[a-zA-Z]',na=True)].
raw_med_test_data = raw_med_test_data.drop(raw_med_test_data[~raw_med_test_data.sideEffectsReview.str.contains('[a-zA-Z]',na=True
raw_med_test_data = raw_med_test_data.drop(raw_med_test_data[~raw_med_test_data.condition.str.contains('[a-zA-Z]',na=True)].index

display(raw_med_test_data)
raw_med_test_data.shape
```

Rename columns/ Drop unnecessary:



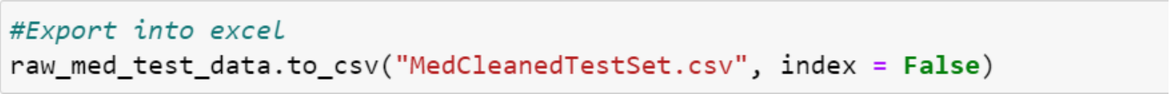
```
# Rename the columns
raw_med_test_data.rename(columns={'Unnamed: 0': 'Unnamed', 'urlDrugName': 'DrugName','rating':'Rating','effectiveness':'Effective
                                'sideEffects':'SideEffects','condition':'Condition','benefitsReview':'BenefitsReview',
                                'sideEffectsReview':'SideEffectsReview','commentsReview':'CommentsReview'}, inplace=True)
# Drop the column that is not described
raw_med_test_data = raw_med_test_data.drop('Unnamed', axis=1)
display(raw_med_test_data)
```

Add sentiment column:



```
#Adding column for ML
raw_med_test_data['Sentiment'] = np.where(raw_med_test_data['Rating'] > 5, 'Positive', 'Negative')
display(raw_med_test_data.head(5))
```

Export:

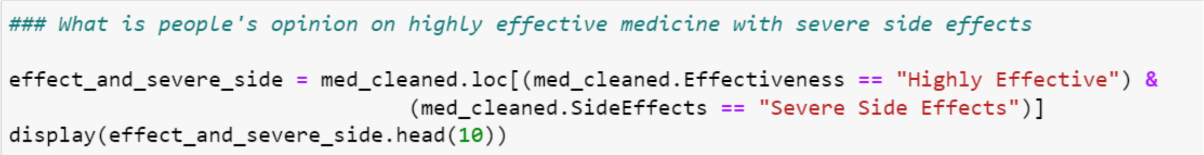


```
#Export into excel
raw_med_test_data.to_csv("MedCleanedTestSet.csv", index = False)
```

Additional snippets for conclusions:

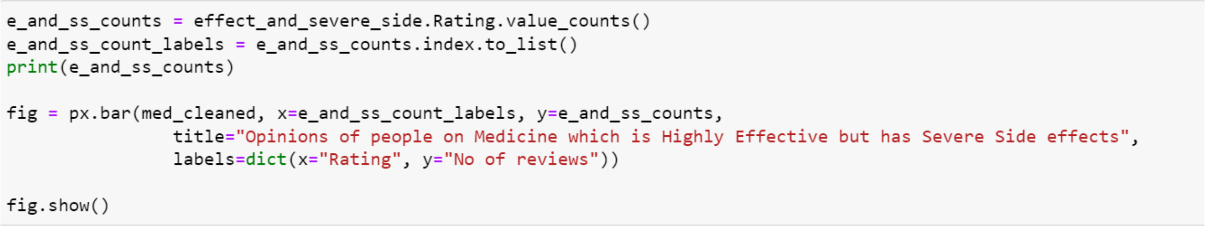
[1] Highly effective medicines with severe side effects rating distribution

Get such medicines:



```
### What is people's opinion on highly effective medicine with severe side effects

effect_and_severe_side = med_cleaned.loc[(med_cleaned.Effectiveness == "Highly Effective") &
                                         (med_cleaned.SideEffects == "Severe Side Effects")]
display(effect_and_severe_side.head(10))
```

Plot rating distribution as bar graph:



```
e_and_ss_counts = effect_and_severe_side.Rating.value_counts()
e_and_ss_count_labels = e_and_ss_counts.index.to_list()
print(e_and_ss_counts)

fig = px.bar(med_cleaned, x=e_and_ss_count_labels, y=e_and_ss_counts,
                title="Opinions of people on Medicine which is Highly Effective but has Severe Side effects",
                labels=dict(x="Rating", y="No of reviews"))

fig.show()
```

Conclusion:

This concludes the code appendix where the snippets of the python code are pasted. For full interpretation, plots and results please refer to the report in the respective sections.