PROJECT INTRODUCTION

Herbal drugs (also flavoringism) is that the study of pharmacognosy and therefore the use of healthful plants. Plants are the premise for medical treatments through most of human history, and such ancient drugs continues to be wide practiced these days fashionable drugs makes use of the many plant-derived compounds because the basis for evidence-based pharmaceutical medicine. though herbalism might apply fashionable standards of effectiveness testing to herbs and medicines derived from natural sources, few high-quality clinical trials and standards for purity or indefinite quantity exist. The scope of flavoring drugs is typically extended to incorporate flora and bee product, similarly as minerals, shells and sure animal elements.

Herbal drugs is additionally known as phyto-medicine or therapy. Paraherbalism describes various and unscientific practices of mistreatment unrefined plant or animal extracts as unproved medicines or health-promoting agents. Paraherbalism differs from plant-derived medicines in customary materia medica as a result of it doesn't isolate or standardize biologically active compounds, however rather depends on the assumption that conserving varied substances from a given supply with less process is safer or more practical – that there's no proof flavoring dietary supplements most frequently comprise the therapy class.

The World Health Organization (WHO) estimates that eighty p.c of the population of some Asian and African countries presently use flavoring drugs for a few side of primary health care, Prescription drugs square measure prohibitively high-ticket for many of the world's population, half whom lived on but $2 U.S. compared, flavoring medicines may be mature from seed or gathered from nature for small or no price.

Many of the prescription drugs presently offered to physicians have a protracted history of use as flavoring remedies, as well as artemisin in, opium, aspirin, digitalis, and antimalarial. In keeping with the planet Health Organization, or so twenty fifth of recent medicine utilized in the u. s.a minimum of seven,000 medical compounds within the fashionable collection square measure derived from plants. Among the one hundred twenty active compounds presently isolated from the upper plants and wide utilized in fashionable drugs these days, eightieth show a direct correlation between their fashionable therapeutic use and therefore the ancient use of the plants from that they're derived.

ANALYSIS OBJECTIVES

Here are the list of question whose analysis which be given below..

1. What is the relation between the successive years and the new changes to a herbal medicine?
2. What is the relation between the successive years and the average time to store a medicine in an inventory?
3. What are the alternative medicines for a particular problem/disease?
4. Number of alternative medicines added in the last 5 years and if any new medicine is formed?
5. Number of new medicines publishes in the last 10 years.
6. Number of new medicines added to the invontory in the last 10 years.
7. Number of new medicines revised/upgraded in the last 10 years.

DATA ACQUISITION AND CLEANING

Code to read the data from Excel / CSV / HTML.

To read the dataset in xlsx format, we will load it into Pandas data frame but first let's import the pandas library and set an alias by typing **"import pandas as pd"**. After importing the library with the alias **"pd"**, let us load the .xlsx file using the following line of code:

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import numpy as np
        import warnings
        warnings.simplefilter(action='ignore', category=FutureWarning)
        import plotly.graph_objects as go

In [2]: data = pd.read_excel("Medicines_output_herbal_medicines.xlsx")

In [3]: data
```

Out[3]:

	Status	Latin name of herbal substance	Botanical name of plant	English common name of herbal substance	Combination	Use	Outcome	Date added to the inventory	Date added to the priority list	First published	Revision date	
0	F: Assessment finalised	Tanaceti parthenii herba	Tanacetum parthenium (L.) Schultz Bip.	Feverfew	no	Pain and inflammation	European Union herbal monograph	2007-10-31 01:00:00	2008-05-08 00:00:00	2010-05-28 02:00:00	2020-10-20 10:40:00	https://www.e /en/medi
1	F: Assessment finalised	Thymi aetheroleum	Thymus vulgaris L.; Thymus zygis Loefl. ex L.	Thyme oil	no	Cough and cold	European Union herbal monograph, European Unio...	2007-09-07 00:00:00	2007-09-07 00:00:00	2009-12-31 02:00:00	2020-10-15 10:06:00	https://www.e /en/medi
			Herniariae				European					

Here we have import our xlsx files and read through pandas library.

Here the xlsx file can be read through (Pandas library) and store in **data Dataframe**. The Dataframe can be shown through **.head()**. The number of rows we want to show, that number we have to pass in head parentheses as an argument.

Now if we want to describe our dataframe for our better understanding to know the stats. and other parameter that our dataset should follow

```
In [5]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 196 entries, 0 to 195
Data columns (total 12 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Status                                 196 non-null    object
 1   Latin name of herbal substance         196 non-null    object
 2   Botanical name of plant                190 non-null    object
 3   English common name of herbal substance 193 non-null  object
 4   Combination                            196 non-null    object
 5   Use                                    168 non-null    object
 6   Outcome                                186 non-null    object
 7   Date added to the inventory            196 non-null    datetime64[ns]
 8   Date added to the priority list        196 non-null    datetime64[ns]
 9   First published                        196 non-null    datetime64[ns]
 10  Revision date                          195 non-null    datetime64[ns]
 11  URL                                    196 non-null    object
dtypes: datetime64[ns](4), object(8)
memory usage: 18.5+ KB

In [6]: data.describe()
```

Out[6]:

	Status	Latin name of herbal substance	Botanical name of plant	English common name of herbal substance	Combination	Use	Outcome	Date added to the inventory	Date added to the priority list	First published	Revision date	
count	196	196	190	193	196	168	186	196	196	196	195	
unique	4	195	163	191	2	56	4	53	61	43	159	
top	F: Assessment finalised	Carvi fructus	Hamamelis virginiana L.	Caraway fruit	no	Urinary tract and genital disorders	European Union herbal monograph	2005-11-24 01:00:00	2005-11-24 01:00:00	2009-12-31 02:00:00	2009-12-31 02:00:00	https://www.em /en/medicin
freq	181	2	3	2	195	24	152	20	19	125	6	
first	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2002-07-12 02:00:00	2005-01-27 01:00:00	2008-07-03 02:00:00	2009-12-31 02:00:00	

```
In [5]: 1 covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54650 entries, 0 to 54649
Data columns (total 12 columns):
dateRep                                                    54650 non-null object
day                                                        54650 non-null int64
month                                                      54650 non-null int64
year                                                       54650 non-null int64
cases                                                      54650 non-null int64
deaths                                                     54650 non-null int64
countriesAndTerritories                                    54650 non-null object
geoId                                                      54409 non-null object
countryterritoryCode                                       54561 non-null object
popData2019                                                54561 non-null float64
continentExp                                               54650 non-null object
Cumulative_number_for_14_days_of_COVID-19_cases_per_100000    51818 non-null float64
dtypes: float64(2), int64(5), object(5)
memory usage: 5.0+ MB
```

```
In [6]: 1 covid.describe()
```

Out[6]:

	day	month	year	cases	deaths	popData2019	Cumulative_number_for_14_days_of_COVID-19_cases_per_100000
count	54650.000000	54650.000000	54650.000000	54650.000000	54650.000000	5.456100e+04	51818.000000
mean	15.693907	6.492333	2019.998774	933.105489	23.130412	4.161009e+07	49.615371
std	8.921747	2.649498	0.034993	5205.737043	124.130190	1.547119e+08	124.680587
min	1.000000	1.000000	2019.000000	-8261.000000	-1918.000000	8.150000e+02	-147.419587
25%	8.000000	4.000000	2020.000000	0.000000	0.000000	1.324820e+06	0.588243
50%	16.000000	7.000000	2020.000000	13.000000	0.000000	7.813207e+06	5.886045
75%	23.000000	9.000000	2020.000000	214.000000	3.000000	2.860872e+07	41.316290
max	31.000000	12.000000	2020.000000	138901.000000	4928.000000	1.433784e+09	1900.836210

Clean the unnecessary data, by removing, replace the missing data and renaming the columns.

Dataset generally contains some null value, which is generally caused by misplacing some values. So its necessary to clean this mess from our dataset for better visualization

In the above diagram we see that some of the columns of this dataset contains null value.

In the process of data cleaning we first drop the particular column name **Combination** and **URL** which contains atmost same value. Then subsequently we fill the null values column with its mean value or 0, so that the values will not much more effect. Finally after all process our dataset is almost clean.

why data clean needed (for your data)

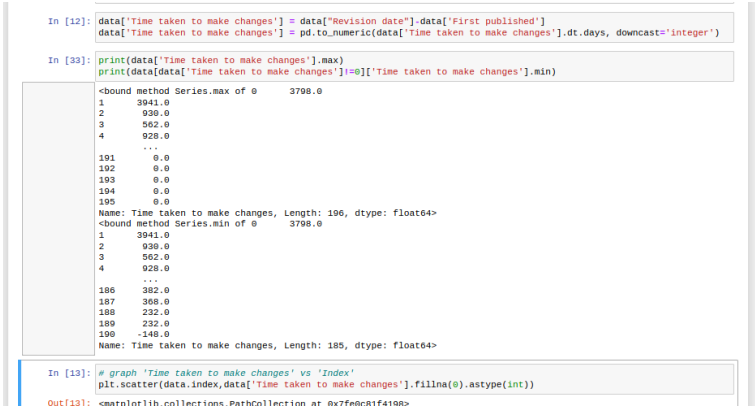
Data cleansing or scrubbing or appending is the procedure of correcting or removing inaccurate and corrupt data. This process is crucial and emphasized because wrong data can drive a business to wrong decisions, conclusions, and poor analysis, especially if the huge quantities of big data are into the picture.

DATA AND EXPLORATORY ANALYSIS

Code and its output with Explanation

1. **What is the relation between the successive years and the new changes to a herbal medicine?**

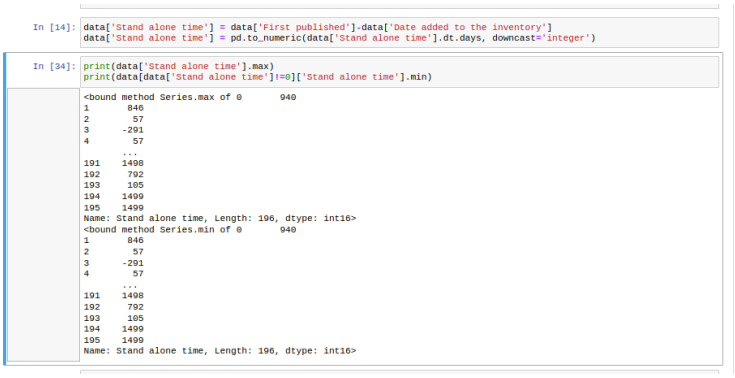
This counts gives the estimated difference to make a change (upgrade most prob.). As the output is given in screenshot.



```
In [12]: data['Time taken to make changes'] = data["Revision date"]-data['First published']
         data['Time taken to make changes'] = pd.to_numeric(data['Time taken to make changes'].dt.days, downcast='integer')

In [33]: print(data['Time taken to make changes'].max)
         print(data[data['Time taken to make changes']!=0]['Time taken to make changes'].min)

<bound method Series.max of 0      3798.0
1      3941.0
2       930.0
3       562.0
4       928.0
        ...
191       0.0
192       0.0
193       0.0
194       0.0
195       0.0
Name: Time taken to make changes, Length: 196, dtype: float64>
<bound method Series.min of 0      3798.0
1      3941.0
2       930.0
3       562.0
4       928.0
        ...
186     382.0
187     368.0
188     232.0
189     232.0
190    -148.0
Name: Time taken to make changes, Length: 185, dtype: float64>

In [13]: # graph 'Time taken to make changes' vs 'Index'
         plt.scatter(data.index,data['Time taken to make changes'].fillna(0).astype(int))

Out[13]: <matplotlib.collections.PathCollection at 0x7fe0c81f4198>
```

2. What is the relation between the successive years and the average time to store a medicine in an inventory?

This counts gives the estimated time difference to take an item from repository to the publishing area. As the output is given in screenshot.



```
In [14]: data['Stand alone time'] = data['First published']-data['Date added to the inventory']
         data['Stand alone time'] = pd.to_numeric(data['Stand alone time'].dt.days, downcast='integer')

In [34]: print(data['Stand alone time'].max)
         print(data[data['Stand alone time']!=0]['Stand alone time'].min)

<bound method Series.max of 0       940
1       846
2        57
3      -291
4        57
       ...
191    1498
192     792
193     105
194    1499
195    1499
Name: Stand alone time, Length: 196, dtype: int16>
<bound method Series.min of 0       940
1       846
2        57
3      -291
4        57
       ...
191    1498
192     792
193     105
194    1499
195    1499
Name: Stand alone time, Length: 196, dtype: int16>
```

3. What are the alternative medicines for a particular problem/disease?

This counts gives the number of alternative medicines to a particular disease/condition. As the output is given in screenshot.

```
In [16]: # alternate meds
diseases = data[['English common name of herbal substance','Use']].groupby('Use').nunique().reset_index()
diseases.columns = ['Use', 'Count']
diseases
```

Out[16]:

	Use	Count
0	Circulatory disorders	5
1	Circulatory disorders, Gastrointestinal disorders	2
2	Circulatory disorders, Mental stress and mood ...	2
3	Constipation	9
4	Constipation, Gastrointestinal disorders	1
5	Cough and cold	13
6	Cough and cold, Circulatory disorders	1
7	Cough and cold, Constipation	1
8	Cough and cold, Gastrointestinal disorders	6
9	Cough and cold, Mental stress and mood disorders	1
10	Cough and cold, Mouth and throat disorders	1
11	Fatigue and weakness	6
12	Fatigue and weakness, Mental stress and mood d...	1
13	Fatigue and weakness, Urinary tract and genita...	1
14	Gastrointestinal disorders	16
15	Loss of appetite	1
16	Loss of appetite, Cough and cold, Gastrointest...	1
17	Loss of appetite, Gastrointestinal disorders	4
18	Loss of appetite, Mouth and throat disorders, ...	1
19	Loss of appetite, Pain and inflammation, Gastr...	1

4. Number of alternative medicines added in the last 5 years and if any new medicine is formed?

This shows that there are only 9 diseases/conditions on which the herbal medicines are manufacturing. Out of which urinary track and genital disorders tops the research.

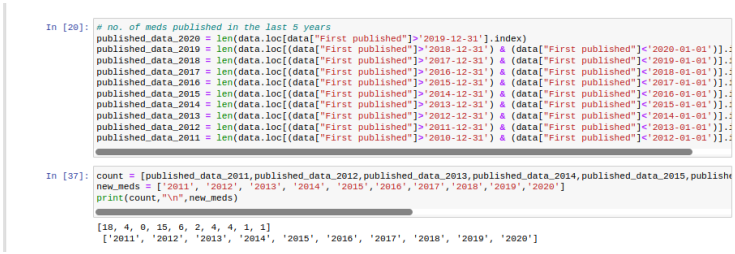
```
In [18]: # diseases corresponding to new med added in the last 5 years with their no. of previous alternatives

med = data[['Use']].loc[data["First published"]>'2015-12-31'].reset_index().drop(columns = ["index"])
USE = diseases[['Use', 'Count']].reset_index().drop(columns = ['index'])
df = pd.merge(USE, med , how = 'inner')
df
```

Out[18]:

	Use	Count
0	Gastrointestinal disorders	16
1	Loss of appetite	1
2	Loss of appetite, Gastrointestinal disorders	4
3	Loss of appetite, Urinary tract and genital di...	2
4	Pain and inflammation	5
5	Skin disorders and minor wounds	7
6	Sleep disorders and temporary insomnia, Mental...	9
7	Urinary tract and genital disorders	24
8	Urinary tract and genital disorders	24

5. **Number of new medicines publishes in the last 10 years.**

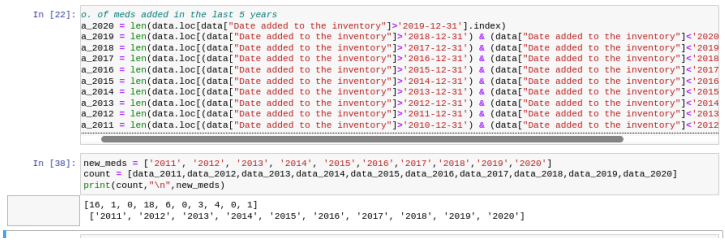
In the last 10 years, 18, 4, 0, 15, 6, 2, 4, 4, 1, 1 herbal medicines are published respectively.



```
In [20]: # no. of meds published in the last 5 years
published_data_2020 = len(data.loc[data["First published"]>'2019-12-31'].index)
published_data_2019 = len(data.loc[(data["First published"]>'2018-12-31') & (data["First published"]<'2020-01-01')].i
published_data_2018 = len(data.loc[(data["First published"]>'2017-12-31') & (data["First published"]<'2019-01-01')].i
published_data_2017 = len(data.loc[(data["First published"]>'2016-12-31') & (data["First published"]<'2018-01-01')].i
published_data_2016 = len(data.loc[(data["First published"]>'2015-12-31') & (data["First published"]<'2017-01-01')].i
published_data_2015 = len(data.loc[(data["First published"]>'2014-12-31') & (data["First published"]<'2016-01-01')].i
published_data_2014 = len(data.loc[(data["First published"]>'2013-12-31') & (data["First published"]<'2015-01-01')].i
published_data_2013 = len(data.loc[(data["First published"]>'2012-12-31') & (data["First published"]<'2014-01-01')].i
published_data_2012 = len(data.loc[(data["First published"]>'2011-12-31') & (data["First published"]<'2013-01-01')].i
published_data_2011 = len(data.loc[(data["First published"]>'2010-12-31') & (data["First published"]<'2012-01-01')].i

In [37]: count = [published_data_2011,published_data_2012,published_data_2013,published_data_2014,published_data_2015,publishe
new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
print(count,"\n",new_meds)

[18, 4, 0, 15, 6, 2, 4, 4, 1, 1]
 ['2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020']
```

6. **Number of new medicines added to the invontory in the last 10 years.**

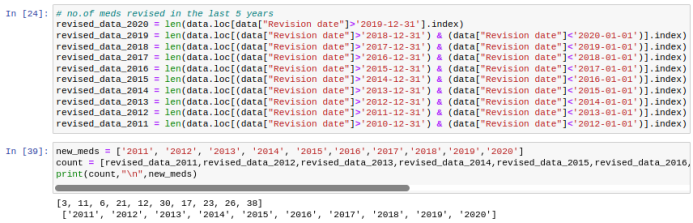
In the last 10 years, 16, 1, 0, 18, 6, 0, 3, 4, 0, 1 herbal medicines are published respectively.



```
In [22]: o. of meds added in the last 5 years
a_2020 = len(data.loc[data["Date added to the inventory"]>'2019-12-31'].index)
a_2019 = len(data.loc[(data["Date added to the inventory"]>'2018-12-31') & (data["Date added to the inventory"]<'2020
a_2018 = len(data.loc[(data["Date added to the inventory"]>'2017-12-31') & (data["Date added to the inventory"]<'2019
a_2017 = len(data.loc[(data["Date added to the inventory"]>'2016-12-31') & (data["Date added to the inventory"]<'2018
a_2016 = len(data.loc[(data["Date added to the inventory"]>'2015-12-31') & (data["Date added to the inventory"]<'2017
a_2015 = len(data.loc[(data["Date added to the inventory"]>'2014-12-31') & (data["Date added to the inventory"]<'2016
a_2014 = len(data.loc[(data["Date added to the inventory"]>'2013-12-31') & (data["Date added to the inventory"]<'2015
a_2013 = len(data.loc[(data["Date added to the inventory"]>'2012-12-31') & (data["Date added to the inventory"]<'2014
a_2012 = len(data.loc[(data["Date added to the inventory"]>'2011-12-31') & (data["Date added to the inventory"]<'2013
a_2011 = len(data.loc[(data["Date added to the inventory"]>'2010-12-31') & (data["Date added to the inventory"]<'2012

In [38]: new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
count = [data_2011,data_2012,data_2013,data_2014,data_2015,data_2016,data_2017,data_2018,data_2019,data_2020]
print(count,"\n",new_meds)

[16, 1, 0, 18, 6, 0, 3, 4, 0, 1]
 ['2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020']
```

7. **Number of new medicines revised/upgraded in the last 10 years.**

In the last 10 years, 16, 1, 0, 18, 6, 0, 3, 4, 0, 1 herbal medicines are published respectively.



```
In [24]: # no.of meds revised in the last 5 years
revised_data_2020 = len(data.loc[data["Revision date"]>'2019-12-31'].index)
revised_data_2019 = len(data.loc[(data["Revision date"]>'2018-12-31') & (data["Revision date"]<'2020-01-01')].index)
revised_data_2018 = len(data.loc[(data["Revision date"]>'2017-12-31') & (data["Revision date"]<'2019-01-01')].index)
revised_data_2017 = len(data.loc[(data["Revision date"]>'2016-12-31') & (data["Revision date"]<'2018-01-01')].index)
revised_data_2016 = len(data.loc[(data["Revision date"]>'2015-12-31') & (data["Revision date"]<'2017-01-01')].index)
revised_data_2015 = len(data.loc[(data["Revision date"]>'2014-12-31') & (data["Revision date"]<'2016-01-01')].index)
revised_data_2014 = len(data.loc[(data["Revision date"]>'2013-12-31') & (data["Revision date"]<'2015-01-01')].index)
revised_data_2013 = len(data.loc[(data["Revision date"]>'2012-12-31') & (data["Revision date"]<'2014-01-01')].index)
revised_data_2012 = len(data.loc[(data["Revision date"]>'2011-12-31') & (data["Revision date"]<'2013-01-01')].index)
revised_data_2011 = len(data.loc[(data["Revision date"]>'2010-12-31') & (data["Revision date"]<'2012-01-01')].index)

In [39]: new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
count = [revised_data_2011,revised_data_2012,revised_data_2013,revised_data_2014,revised_data_2015,revised_data_2016,
print(count,"\n",new_meds)

[3, 11, 6, 21, 12, 30, 17, 23, 26, 38]
 ['2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020']
```

DATA ANALYSIS – VISUALIZATION

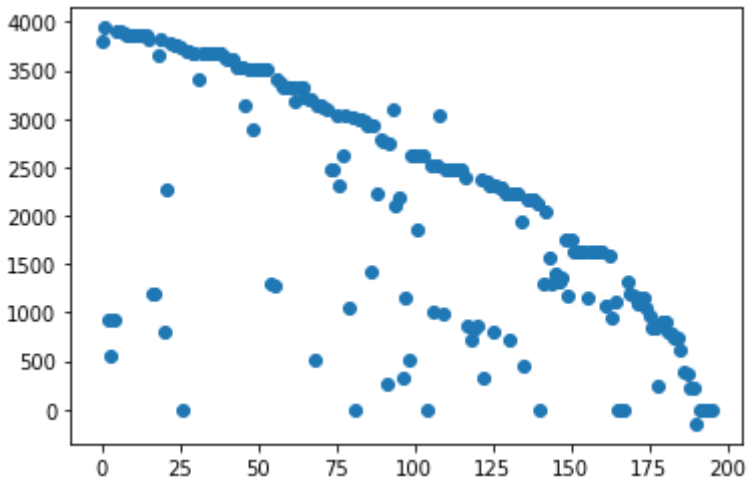
Code and its output with vizualization

1. **What is the relation between the successive years and the new changes to a herbal medicine?**

   Code:-

```
In [12]: data['Time taken to make changes'] = data["Revision date"]-data['First published']
         data['Time taken to make changes'] = pd.to_numeric(data['Time taken to make changes'].dt.days, downcast='integer')

In [33]: print(data['Time taken to make changes'].max)
         print(data[data['Time taken to make changes']!=0]['Time taken to make changes'].min)

                                                                                    ...

In [13]: # graph 'Time taken to make changes' vs 'Index'
         plt.scatter(data.index,data['Time taken to make changes'].fillna(0).astype(int))
```

   Output:-



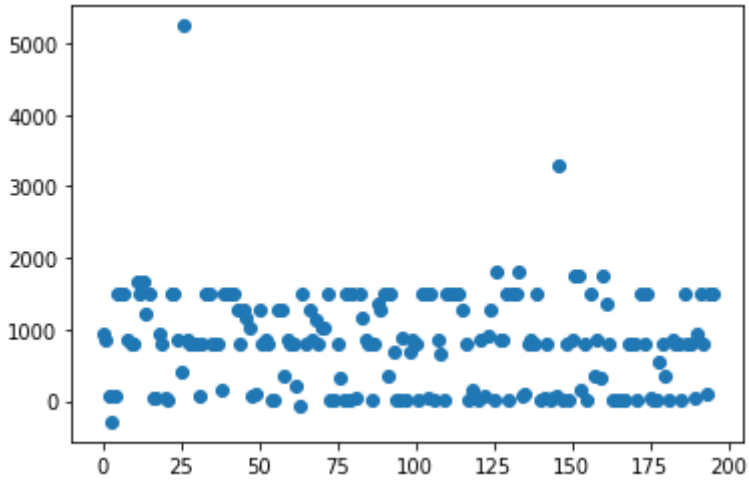
   This visualization shows that as time increases, the waiting time to produce a new medicine decreases.

2. **What is the relation between the successive years and the average time to store a medicine in an inventory?**

Code:-

```
In [14]: data['Stand alone time'] = data['First published']-data['Date added to the inventory']
         data['Stand alone time'] = pd.to_numeric(data['Stand alone time'].dt.days, downcast='integer')

In [34]: print(data['Stand alone time'].max)
         print(data[data['Stand alone time']!=0]['Stand alone time'].min)

In [15]: # graph 'Stand alone time' vs 'Index'
         plt.scatter(data.index,data['Stand alone time'].fillna(0).astype(int))
```

Output:-



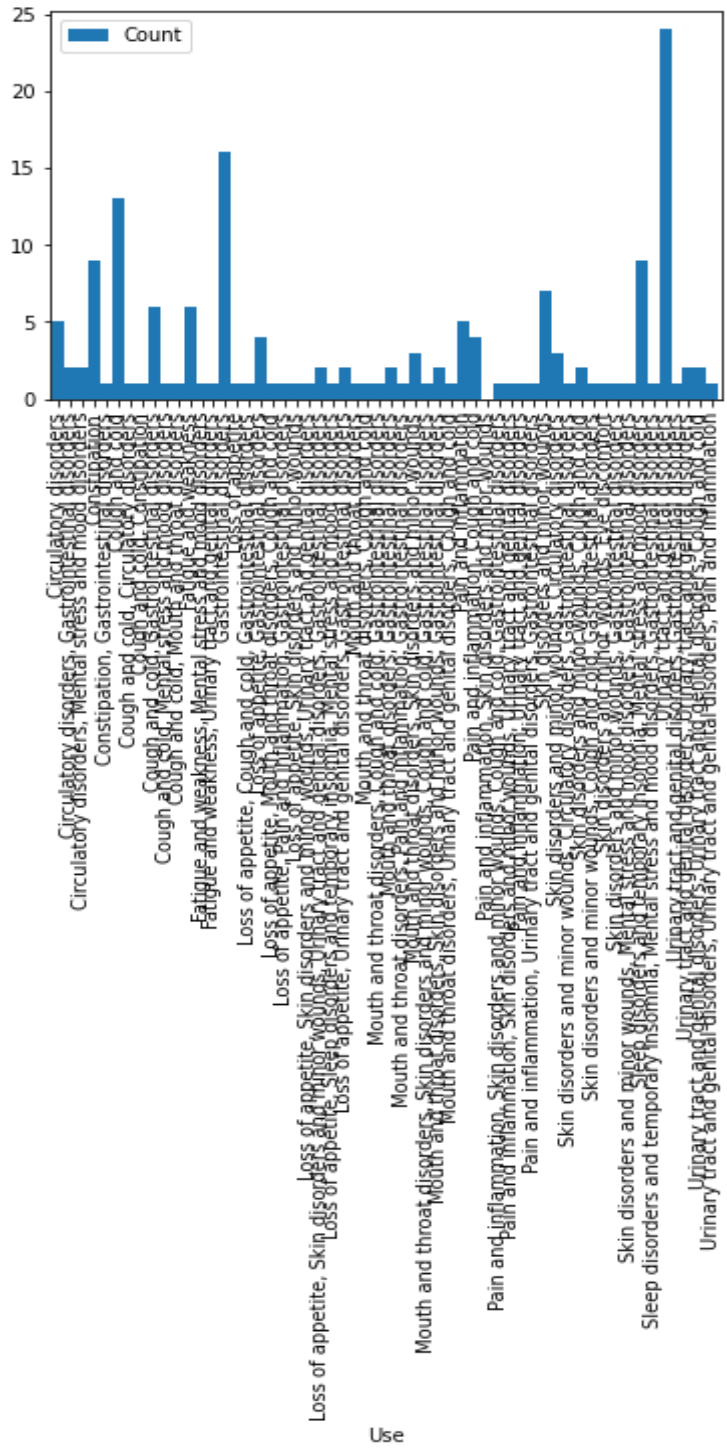
This is the graphical representation of total time taken to store a medicine raw material before it is passed to publish the medicine. Here, the graph is between the amount of days(y-axis) and year from 2010-01-01 till now(x-axis).

3. **What are the alternative medicines for a particular problem/disease?**

Code:-

```
In [16]: # alternate meds
         diseases = data[['English common name of herbal substance','Use']].groupby('Use').nunique().reset_index()
         diseases.columns = ['Use', 'Count']
         diseases

In [27]: # graph of df diseases
         plt.figure(figsize = (15,27))
         ax = diseases.plot(x = 'Use', y= 'Count',kind = "bar" ,width =1)
```

Output:-



From this output, we can see that some disease like **Gastrointestinal disorders** have many alternative medicines, but some like **Mouth and throat disorders, Cough and cold** have very few alternatives or none at all.

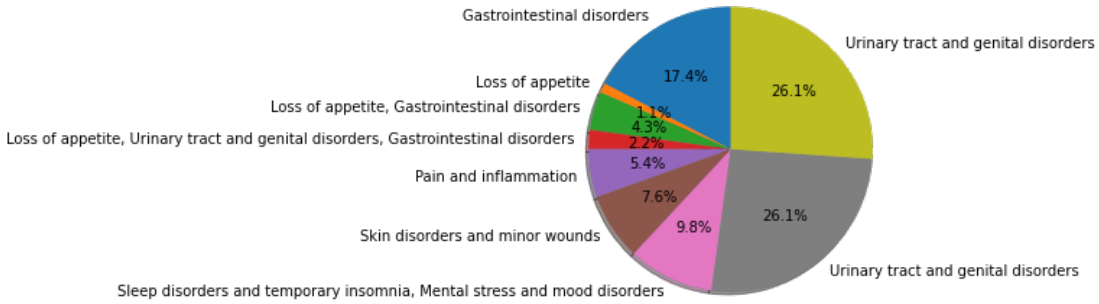
4. **Number of alternative medicines added in the last 5 years and if any new medicine is formed?**

Code:-

```
In [18]: # diseases corresponding to new med added in the last 5 years with their no. of previous alternatives

         med = data[['Use']].loc[data["First published"]>'2015-12-31'].reset_index().drop(columns = ["index"])
         USE = diseases[['Use', 'Count']].reset_index().drop(columns = ['index'])
         df = pd.merge(USE, med , how = 'inner')
         df
```

```
In [19]: #graph
         fig1, ax1 = plt.subplots()
         ax1.pie(df.Count, labels=df.Use, autopct='%1.1f%%',shadow=True, startangle=90)
         ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
         plt.show()
```

Output:-



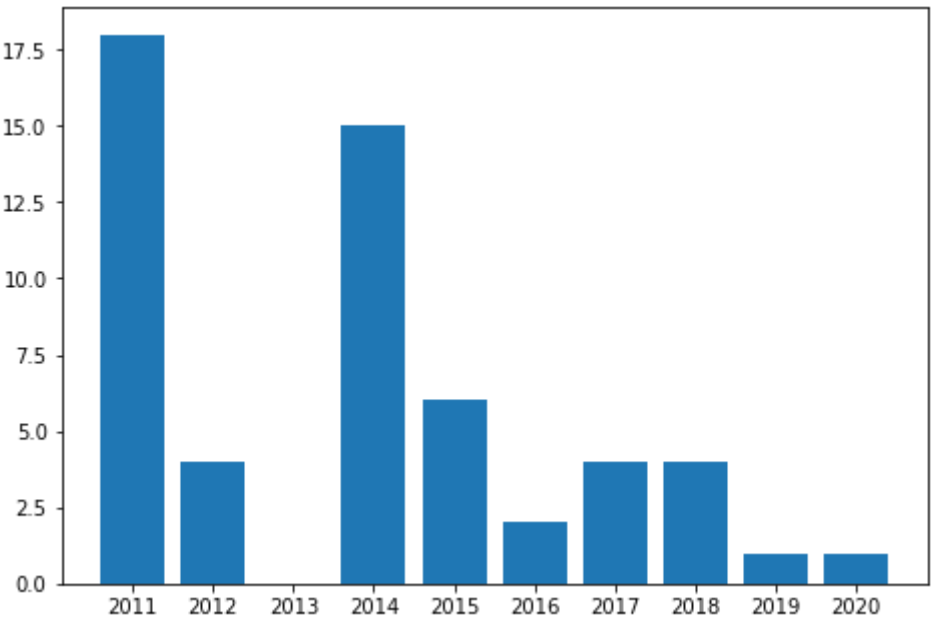
From this pie chart, we can see that medicines for some problems like urinary infection is in high demands, while some like problems loss of appetite is comparable to none at all.

5. **Number of new medicines publishes in the last 10 years.**

Code:-

```
In [20]: # no. of meds published in the last 5 years
published_data_2020 = len(data.loc[data["First published"]>'2019-12-31'].index)
published_data_2019 = len(data.loc[(data["First published"]>'2018-12-31') & (data["First published"]<'2020-01-01')].
published_data_2018 = len(data.loc[(data["First published"]>'2017-12-31') & (data["First published"]<'2019-01-01')].
published_data_2017 = len(data.loc[(data["First published"]>'2016-12-31') & (data["First published"]<'2018-01-01')].
published_data_2016 = len(data.loc[(data["First published"]>'2015-12-31') & (data["First published"]<'2017-01-01')].
published_data_2015 = len(data.loc[(data["First published"]>'2014-12-31') & (data["First published"]<'2016-01-01')].
published_data_2014 = len(data.loc[(data["First published"]>'2013-12-31') & (data["First published"]<'2015-01-01')].
published_data_2013 = len(data.loc[(data["First published"]>'2012-12-31') & (data["First published"]<'2014-01-01')].
published_data_2012 = len(data.loc[(data["First published"]>'2011-12-31') & (data["First published"]<'2013-01-01')].
published_data_2011 = len(data.loc[(data["First published"]>'2010-12-31') & (data["First published"]<'2012-01-01')].

In [21]: # graph
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
count = [published_data_2011,published_data_2012,published_data_2013,published_data_2014,published_data_2015,publishe
ax.bar(new_meds,count)
plt.show()
```

Output:-



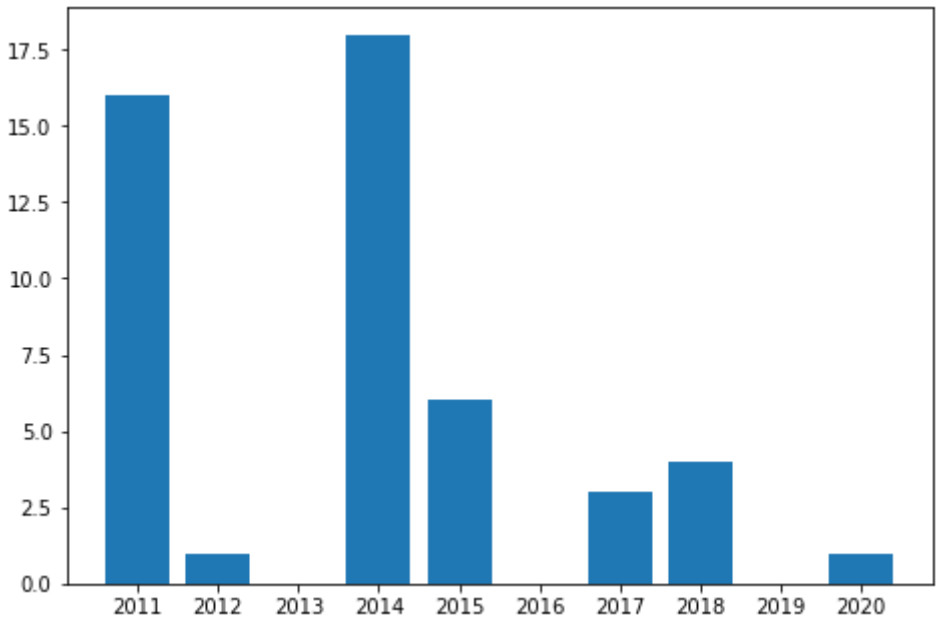
From this bar chart, we can see that compared to years like 2011, 2014, the years like 2019,2020 have not published much medicines. In 2020, even with the rise of covid-19, the publishing is so less i.e., either the sufficient data is not received yet, or company is going through some sort of issues.

6. **Number of new medicines added to the invontory in the last 10 years.**

Code:-

```
In [22]: no. of meds added in the last 5 years
         a_2020 = len(data.loc[data["Date added to the inventory"]>'2019-12-31'].index)
         a_2019 = len(data.loc[(data["Date added to the inventory"]>'2018-12-31') & (data["Date added to the inventory"]<'2020
         a_2018 = len(data.loc[(data["Date added to the inventory"]>'2017-12-31') & (data["Date added to the inventory"]<'2019
         a_2017 = len(data.loc[(data["Date added to the inventory"]>'2016-12-31') & (data["Date added to the inventory"]<'2018
         a_2016 = len(data.loc[(data["Date added to the inventory"]>'2015-12-31') & (data["Date added to the inventory"]<'2017
         a_2015 = len(data.loc[(data["Date added to the inventory"]>'2014-12-31') & (data["Date added to the inventory"]<'2016
         a_2014 = len(data.loc[(data["Date added to the inventory"]>'2013-12-31') & (data["Date added to the inventory"]<'2015
         a_2013 = len(data.loc[(data["Date added to the inventory"]>'2012-12-31') & (data["Date added to the inventory"]<'2014
         a_2012 = len(data.loc[(data["Date added to the inventory"]>'2011-12-31') & (data["Date added to the inventory"]<'2013
         a_2011 = len(data.loc[(data["Date added to the inventory"]>'2010-12-31') & (data["Date added to the inventory"]<'2012

In [23]: # graph
         fig = plt.figure()
         ax = fig.add_axes([0,0,1,1])
         new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
         count = [data_2011,data_2012,data_2013,data_2014,data_2015,data_2016,data_2017,data_2018,data_2019,data_2020]
         ax.bar(new_meds,count)
         plt.show()
```

Output:-



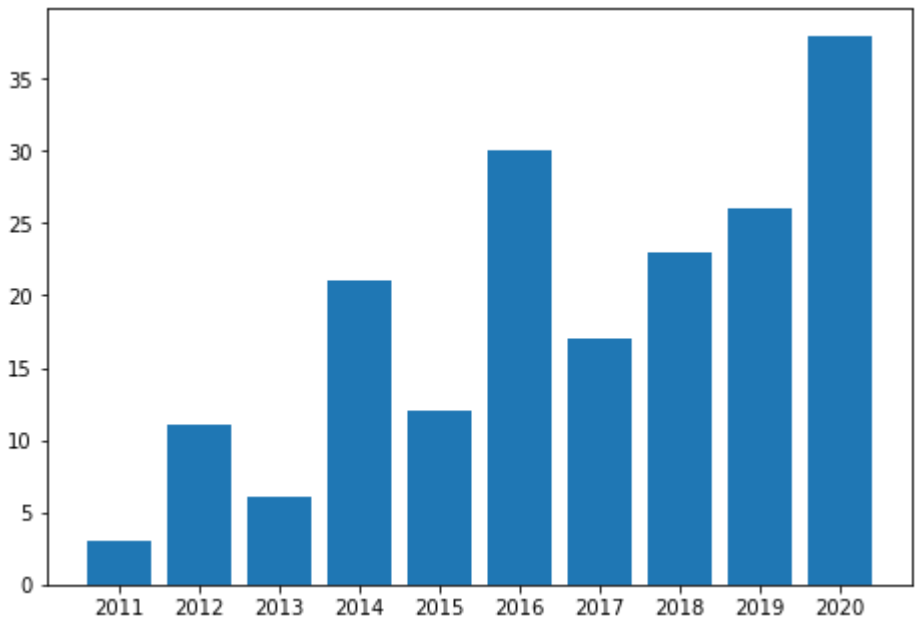
From point 5, we conclude that in 2019, 2020 there's been some reason through which enough medicines is not published. From this graph, we can say that due to lack of material in the inventory, the company is unable to publish. The lack of availability can be due to any investment issue or has not much sufficient man power.

7. **Number of new medicines revised/upgraded in the last 10 years**

Code:-

```
In [24]: # no.of meds revised in the last 5 years
revised_data_2020 = len(data.loc[data["Revision date"]>'2019-12-31'].index)
revised_data_2019 = len(data.loc[(data["Revision date"]>'2018-12-31') & (data["Revision date"]<'2020-01-01')].index)
revised_data_2018 = len(data.loc[(data["Revision date"]>'2017-12-31') & (data["Revision date"]<'2019-01-01')].index)
revised_data_2017 = len(data.loc[(data["Revision date"]>'2016-12-31') & (data["Revision date"]<'2018-01-01')].index)
revised_data_2016 = len(data.loc[(data["Revision date"]>'2015-12-31') & (data["Revision date"]<'2017-01-01')].index)
revised_data_2015 = len(data.loc[(data["Revision date"]>'2014-12-31') & (data["Revision date"]<'2016-01-01')].index)
revised_data_2014 = len(data.loc[(data["Revision date"]>'2013-12-31') & (data["Revision date"]<'2015-01-01')].index)
revised_data_2013 = len(data.loc[(data["Revision date"]>'2012-12-31') & (data["Revision date"]<'2014-01-01')].index)
revised_data_2012 = len(data.loc[(data["Revision date"]>'2011-12-31') & (data["Revision date"]<'2013-01-01')].index)
revised_data_2011 = len(data.loc[(data["Revision date"]>'2010-12-31') & (data["Revision date"]<'2012-01-01')].index)

In [39]: new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
count = [revised_data_2011,revised_data_2012,revised_data_2013,revised_data_2014,revised_data_2015,revised_data_2016,
print(count,"\n",new_meds)

...

In [25]: # graph
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
new_meds = ['2011', '2012', '2013', '2014', '2015','2016','2017','2018','2019','2020']
count = [revised_data_2011,revised_data_2012,revised_data_2013,revised_data_2014,revised_data_2015,revised_data_2016,
ax.bar(new_meds,count)
plt.show()
```

Output:-



In spite of lack of inventory and publishing of new medicines, the up-gradation / revision of previous medicines are up to date. Which means the staff is doing their job very well.

EXECUTIVE SUMMARY

CONCLUSION

From the above observations, we concluded:

1. With the increase in technology, the time to upgrade any medicine decreases but the average time requires to kept the raw product in the inventory remains the same.

2. In the last 5 years, we have discovered medicine for only one condition: Loss of appetite, and different medicine are manufactured for urinary track and genital disorders.

3. As compared to 10 years age, we are producing and publishing very less, but revising a lot. It shows that either no new disease/condition is being produced, or we are unable to make a medicine for that.

REFERENCES

- www.globaltimes.cn
- https://data.europa.eu/euodp/en/data
- https://en.wikipedia.org/wiki/Herbal_medicine