Imports

```
In [3]: import pandas as pd
        import numpy as np
```

Load Dataset

The dataset is collected from Centers of Disease control and prevention data.cdc.gov under the category of Vaccination Coverage among Pregnancy Women

```
In [4]: # Read the data
        data = pd.read_csv('Vaccination_Coverage_among_Pregnant_Women.csv')
```

```
In [5]: # Display Features
        data.columns
```

```
Out[5]: Index(['Vaccine', 'Geography Type', 'Geography',
               'Survey Year/Influenza Season', 'Dimension Type', 'Dimension',
               'Estimate (%)', '95% CI (%)', 'Sample Size'],
              dtype='object')
```

```
In [6]: # Display Values
        data
```

Out[6]:

	Vaccine	Geography Type	Geography	Survey Year/Influenza Season	Dimension Type	Dimension	Estimate (%)	95% CI (%)	Sample Size
0	Influenza	States	Alaska	2012	Age	≥18 Years	49.2	45.3 to 53.1	852.0
1	Influenza	States	Arkansas	2012	Age	≥18 Years	46.6	40.7 to 52.5	756.0
2	Influenza	States	Colorado	2012	Age	≥18 Years	56.1	52.1 to 60.0	1170.0
3	Influenza	States	Delaware	2012	Age	≥18 Years	41.6	38.4 to 44.8	981.0
4	Influenza	States	Georgia	2012	Age	≥18 Years	33.6	29.6 to 37.7	1007.0
...	...	...	...	...	...	...	...	...	...
4132	Tdap	States	Utah	2020	Race/Ethnicity	White, Non-Hispanic	80.1	77.0 to 83.0	979.0
4133	Tdap	States	Vermont	2020	Race/Ethnicity	White, Non-Hispanic	86.4	83.6 to 88.9	696.0
4134	Tdap	States	Virginia	2020	Race/Ethnicity	White, Non-Hispanic	83.1	76.9 to 88.2	503.0
4135	Tdap	States	Washington	2020	Race/Ethnicity	White, Non-Hispanic	80.9	76.2 to 85.0	352.0
4136	Tdap	States	Wisconsin	2020	Race/Ethnicity	White, Non-Hispanic	82.8	78.4 to 86.7	364.0

4137 rows × 9 columns

```
In [7]: # Display Dimension
        data.shape
```

```
Out[7]: (4137, 9)
```

```
In [8]: # Display Information
        data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4137 entries, 0 to 4136
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Vaccine                       4137 non-null   object
 1   Geography Type                4137 non-null   object
 2   Geography                     4137 non-null   object
 3   Survey Year/Influenza Season  4137 non-null   int64
 4   Dimension Type                4137 non-null   object
 5   Dimension                     4137 non-null   object
 6   Estimate (%)                  4137 non-null   object
 7   95% CI (%)                    4137 non-null   object
 8   Sample Size                   3933 non-null   float64
dtypes: float64(1), int64(1), object(7)
memory usage: 291.0+ KB
```

```
In [9]: # Display the Unique values
        data['Vaccine'].value_counts()
```

```
Out[9]: Influenza    2891
        Tdap         1246
        Name: Vaccine, dtype: int64
```

```
In [10]: # Display top 3 values
         data.head(3)
```

Out[10]:

	Vaccine	Geography Type	Geography	Survey Year/Influenza Season	Dimension Type	Dimension	Estimate (%)	95% CI (%)	Sample Size
0	Influenza	States	Alaska	2012	Age	≥18 Years	49.2	45.3 to 53.1	852.0
1	Influenza	States	Arkansas	2012	Age	≥18 Years	46.6	40.7 to 52.5	756.0
2	Influenza	States	Colorado	2012	Age	≥18 Years	56.1	52.1 to 60.0	1170.0

```
In [11]: data['Survey Year/Influenza Season'].unique()
```

```
Out[11]: array([2012, 2020, 2013, 2014, 2015, 2016, 2017, 2018, 2019])
```

```
In [12]: # Make a new copy
         new_data = data.copy()
```

```
In [13]: # Display the presence of missing values
         new_data.isnull().sum()
```

```
Out[13]: Vaccine                           0
         Geography Type                    0
         Geography                         0
         Survey Year/Influenza Season      0
         Dimension Type                    0
         Dimension                         0
         Estimate (%)                      0
         95% CI (%)                        0
         Sample Size                     204
         dtype: int64
```

```
In [14]: new_data.describe()
```

Out[14]:

	Survey Year/Influenza Season	Sample Size
count	4137.000000	3933.000000
mean	2016.230602	805.390796
std	2.663058	2658.305251
min	2012.000000	30.000000
25%	2014.000000	169.000000
50%	2017.000000	325.000000
75%	2019.000000	659.000000
max	2020.000000	43737.000000

```
In [15]: type(new_data['Estimate (%)'][0])
```

```
Out[15]: str
```

```
In [16]: new_data['Estimate (%)'].unique()
```

```
Out[16]: array(['49.2', '46.6', '56.1', '41.6', '33.6', '42.0', '49.1', '53.0',
       '47.9', '66.1', '42.8', '66.0', '45.5', '58.8', '38.6', '37.8',
       '39.5', '43.6', '54.3', '47.3', '47.4', '60.4', '44.1', '48.9',
       '57.5', '60.9', '43.4', '58.6', '39.1', '49.6', '64.0', '57.3',
       '58.2', '74.7', '73.5', '62.9', '73.1', '41.3', '40.9', '68.7',
       '65.1', '77.1', '71.2', '53.7', '72.4', '70.5', '79.8', '58.4',
       '68.2', '48.1', '64.8', '68.5', '76.0', '75.0', '60.8', '67.6',
       '75.0', '74.1', '67.1', '64.5', '34.8', '61.1', '76.1', '54.9',
       '67.2', '64.4', '74.0', '64.7', '70.7', '63.3', '39.0', '51.3',
       '33.4', '27.6', '38.5', '42.5', '49.0', '41.0', '51.0', '37.0',
       '59.4', '32.6', '51.6', '30.7', '50.6', '29.5', '50.0', '38.1',
       '39.8', '56.7', '38.9', '41.9', '51.7', '53.2', '39.7', '52.4',
       '32.0', '40.5', '66.8', '56.4', '67.8', '63.5', '51.1', '49.7',
       '25.9', '29.9', '72.5', '50.3', '73.3', '43.9', '40.4', '69.5',
       '58.3', '67.0', '45.1', 'NR*', '37.3', '60.1', '65.3', '55.2',
       '65.6', '58.0', '69.9', '62.8', '38.3', '50.2', '65.2', '51.5',
       '70.2', '59.1', '58.7', '61.7', '48.3', '46.9', '35.7', '42.6',
       '48.5', '53.1', '49.9', '45.4', '66.4', '51.8', '60.7', '40.1',
       '48.7', '37.7', '44.4', '46.7', '54.7', '49.8', '61.4', '45.2',
       '58.5', '64.9', '57.9', '53.4', '62.7', '57.8', '74.6', '74.8',
       '64.2', '67.4', '42.9', '72.6', '57.4', '57.1', '72.2', '79.1',
       '61.6', '68.6', '65.9', '69.8', '77.9', '76.8', '68.4', '67.5',
       '31.2', '75.6', '69.4', '73.2', '73.8', '65.5', '39.4', '37.4',
       '44.7', '60.0', '50.5', '68.1', '72.9', '54.1', '40.7', '46.2',
       '45.6', '52.1', '62.5', '64.1', '57.2', '61.0', '82.0', '76.4',
       '71.1', '90.8', '50.9', '50.8', '71.6', '72.1', '81.7', '77.6',
       '59.8', '75.1', '69.6', '85.8', '63.8', '57.6', '66.7', '75.9',
       '77.7', '68.8', '77.2', '65.4', '80.5', '66.9', '76.9', '78.5',
       '70.0', '28.3', '20.3', '37.5', '39.6', '33.8', '60.2', '35.3',
       '50.1', '33.1', '26.5', '41.7', '48.8', '56.0', '36.1', '56.5',
       '29.2', '63.4', '64.3', '54.4', '45.7', '24.8', '23.3', '36.2',
       '62.6', '53.5', '59.2', '34.7', '53.8', '44.6', '55.9', '48.4',
       '77.0', '35.6', '45.9', '53.3', '30.6', '47.7', '35.0', '46.8',
       '41.4', '36.3', '51.9', '67.7', '59.5', '61.5', '57.0', '68.3',
       '90.9', '36.0', '32.9', '61.9', '76.3', '71.3', '80.4', '34.5',
       '71.0', '45.3', '40.2', '63.7', '37.6', '43.5', '47.0', '69.3',
       '61.3', '78.0', '83.9', '82.6', '81.3', '71.5', '70.8', '56.2',
       '43.3', '44.2', '29.8', '52.2', '51.4', '44.0', '59.7', '40.8',
       '46.5', '55.3', '46.4', '49.5', '43.8', '59.3', '38.7', '74.9',
       '78.4', '95.1', '69.7', '79.6', '54.0', '80.8', '70.3', '53.9',
       '70.9', '76.6', '63.9', '77.3', '64.6', '73.4', '23.4', '55.4',
       '48.2', '43.2', '54.6', '72.7', '58.1', '62.4', '50.4', '65.7',
       '42.4', '61.8', '52.3', '65.8', '60.6', '52.5', '63.2', '55.5',
       '52.8', '54.8', '75.7', '59.6', '66.3', '59.9', '66.5', '57.7',
       '45.8', '52.9', '52.0', '56.3', '58.9', '47.5', '62.3', '73.7',
       '51.2', '75.5', '55.0', '59.0', '70.4', '70.1', '25.0', '75.3',
       '72.0', '40.3', '61.2', '55.7', '49.4', '70.6', '31.0', '74.3',
       '72.3', '48.0', '69.2', '56.8', '66.2', '73.9', '82.2', '14.7',
       '44.3', '60.3', '40.6', '43.0', '46.3', '42.2', '60.5', '39.2',
       '56.9', '36.5', '54.2', '47.6', '38.2', '68.0', '35.4', '53.6',
       '54.5', '36.6', '63.0', '33.5', '55.1', '30.5', '41.1', '35.8',
       '71.4', '29.3', '36.8', '36.9', '69.0', '44.8', '76.7', '55.8',
       '41.5', '25.3', '71.8', '62.1', '79.2', '69.1', '47.2', '50.7',
       '73.0', '67.9', '75.4', '62.2', '52.6', '52.7', '55.6', '75.2',
       '67.3', '63.1', '46.1', '63.6', '21.7', '68.9', '41.8', '78.1',
       '56.6', '71.7', '32.3', '81.5', '20.1', '76.5', '73.6', '81.6',
       '77.4', '74.5', '20.7', '44.9', '82.1', '49.3', '81.4', '78.7',
       '17.1', '83.3', '81.9', '78.2', '82.9', '62.0', '87.1', '78.8',
       '72.8', '16.6', '39.9', '40.0', '30.8', '79.7', '34.0', '48.6',
       '35.9', '83.5', '79.4', '38.0', '43.1', '24.7', '42.1', '27.8',
       '26.6', '38.4', '44.5', '66.6', '75.8', '74.2', '25.2', '86.8',
       '71.9', '76.2', '84.4', '42.3', '84.6', '85.0', '79.3', '83.0',
       '80.3', '81.8', '84.2', '78.6', '80.2', '29.6', '87.0', '91.5',
       '42.7', '79.5', '82.7', '20.5', '8.2', '15.5', '14.0', '15.8',
       '9.1', '18.8', '25.6', '31.3', '26.4', '27.1', '28.0', '32.8',
       '47.8', '84.1', '86.5', '86.0', '74.4', '81.2', '83.6', '88.6',
       '85.6', '83.2', '90.2', '86.9', '85.9', '86.3', '89.5', '85.1',
       '82.8', '78.9', '88.4', '86.1', '87.7', '89.2', '87.2', '90.4',
       '81.0', '86.2', '83.7', '21.6', '7.7', '12.1', '18.3', '14.5',
       '15.1', '25.5', '20.4', '25.4', '29.7', '39.3', '84.7', '86.7',
       '77.5', '91.6', '87.9', '77.8', '80.9', '87.3', '85.3', '83.8',
       '89.3', '87.4', '87.6', '89.7', '90.1', '18.7', '8.4', '15.9',
       '10.8', '18.0', '8.1', '14.3', '23.1', '26.8', '31.6', '27.0',
       '46.0', '84.0', '80.7', '83.4', '88.1', '84.5', '81.1', '90.0',
       '80.0', '88.0', '84.8', '84.3', '82.3', '89.1', '85.5', '89.0',
       '88.2', '92.1', '83.1', '24.9', '8.5', '19.0', '17.7', '15.7',
       '21.4', '13.8', '25.7', '23.9', '30.9', '24.1', '88.5', '91.7',
       '90.6', '87.5', '91.8', '82.4', '84.9', '87.8', '89.6', '80.6',
       '11.2', '21.8', '31.1', '29.0', '19.9', '32.7', '79.0', '18.5',
       '26.7', '22.9', '30.3', '37.2', '85.7', '88.7', '86.4', '91.3',
       '79.9', '85.4', '14.9', '23.6', '80.1', '93.7', '82.5', '93.0',
       '98.2', '88.8', '18.6', '5.2', '13.7', '13.6', '13.4', '26.2',
       '27.9', '31.4', '47.1', '91.0', '91.4', '78.3', '89.9'],
      dtype=object)
```

```
In [18]: # There is some unwanted values so replacing it with NaN for easy cleaning of data
         new_data['Estimate (%)'] = new_data['Estimate (%)'].replace('NR*', 'NaN')
```

```
In [19]: new_data['Estimate (%)'].unique()
```

```
Out[19]: array(['49.2', '46.6', '56.1', '41.6', '33.6', '42.0', '49.1', '53.0',
       '47.9', '66.1', '42.8', '66.0', '45.5', '58.8', '38.6', '37.8',
       '39.5', '43.6', '54.3', '47.3', '47.4', '60.4', '44.1', '48.9',
       '57.5', '60.9', '43.4', '58.6', '39.1', '49.6', '64.0', '57.3',
       '58.2', '74.7', '73.5', '62.9', '73.1', '41.3', '40.9', '68.7',
       '65.1', '77.1', '71.2', '53.7', '72.4', '70.5', '79.8', '58.4',
       '68.2', '48.1', '64.8', '68.5', '76.0', '75.0', '60.8', '67.6',
       '75.0', '74.1', '67.1', '64.5', '34.8', '61.1', '76.1', '54.9',
       '67.2', '64.4', '74.0', '64.7', '70.7', '63.3', '39.0', '51.3',
       '33.4', '27.6', '38.5', '42.5', '49.0', '41.0', '51.0', '37.0',
       '59.4', '32.6', '51.6', '30.7', '50.6', '29.5', '50.0', '38.1',
       '39.8', '56.7', '38.9', '41.9', '51.7', '53.2', '39.7', '52.4',
       '32.0', '40.5', '66.8', '56.4', '67.8', '63.5', '51.1', '49.7',
       '25.9', '29.9', '72.5', '50.3', '73.3', '43.9', '40.4', '69.5',
       '58.3', '67.0', '45.1', 'NaN', '37.3', '60.1', '65.3', '55.2',
       '65.6', '58.0', '69.9', '62.8', '38.3', '50.2', '65.2', '51.5',
       '70.2', '59.1', '58.7', '61.7', '48.3', '46.9', '35.7', '42.6',
       '48.5', '53.1', '49.9', '45.4', '66.4', '51.8', '60.7', '40.1',
       '48.7', '37.7', '44.4', '46.7', '54.7', '49.8', '61.4', '45.2',
       '58.5', '64.9', '57.9', '53.4', '62.7', '57.8', '74.6', '74.8',
       '64.2', '67.4', '42.9', '72.6', '57.4', '57.1', '72.2', '79.1',
       '61.6', '68.6', '65.9', '69.8', '77.9', '76.8', '68.4', '67.5',
       '31.2', '75.6', '69.4', '73.2', '73.8', '65.5', '39.4', '37.4',
       '44.7', '60.0', '50.5', '68.1', '72.9', '54.1', '40.7', '46.2',
       '45.6', '52.1', '62.5', '64.1', '57.2', '61.0', '82.0', '76.4',
       '71.1', '90.8', '50.9', '50.8', '71.6', '72.1', '81.7', '77.6',
       '59.8', '75.1', '69.6', '85.8', '63.8', '57.6', '66.7', '75.9',
       '77.7', '68.8', '77.2', '65.4', '80.5', '66.9', '76.9', '78.5',
       '70.0', '28.3', '20.3', '37.5', '39.6', '33.8', '60.2', '35.3',
       '50.1', '33.1', '26.5', '41.7', '48.8', '56.0', '36.1', '56.5',
       '29.2', '63.4', '64.3', '54.4', '45.7', '24.8', '23.3', '36.2',
       '62.6', '53.5', '59.2', '34.7', '53.8', '44.6', '55.9', '48.4',
       '77.0', '35.6', '45.9', '53.3', '30.6', '47.7', '35.0', '46.8',
       '41.4', '36.3', '51.9', '67.7', '59.5', '61.5', '57.0', '68.3',
       '90.9', '36.0', '32.9', '61.9', '76.3', '71.3', '80.4', '34.5',
       '71.0', '45.3', '40.2', '63.7', '37.6', '43.5', '47.0', '69.3',
       '61.3', '78.0', '83.9', '82.6', '81.3', '71.5', '70.8', '56.2',
       '43.3', '44.2', '29.8', '52.2', '51.4', '44.0', '59.7', '40.8',
       '46.5', '55.3', '46.4', '49.5', '43.8', '59.3', '38.7', '74.9',
       '78.4', '95.1', '69.7', '79.6', '54.0', '80.8', '70.3', '53.9',
       '70.9', '76.6', '63.9', '77.3', '64.6', '73.4', '23.4', '55.4',
       '48.2', '43.2', '54.6', '72.7', '58.1', '62.4', '50.4', '65.7',
       '42.4', '61.8', '52.3', '65.8', '60.6', '52.5', '63.2', '55.5',
       '52.8', '54.8', '75.7', '59.6', '66.3', '59.9', '66.5', '57.7',
       '45.8', '52.9', '52.0', '56.3', '58.9', '47.5', '62.3', '73.7',
       '51.2', '75.5', '55.0', '59.0', '70.4', '70.1', '25.0', '75.3',
       '72.0', '40.3', '61.2', '55.7', '49.4', '70.6', '31.0', '74.3',
       '72.3', '48.0', '69.2', '56.8', '66.2', '73.9', '82.2', '14.7',
       '44.3', '60.3', '40.6', '43.0', '46.3', '42.2', '60.5', '39.2',
       '56.9', '36.5', '54.2', '47.6', '38.2', '68.0', '35.4', '53.6',
       '54.5', '36.6', '63.0', '33.5', '55.1', '30.5', '41.1', '35.8',
       '71.4', '29.3', '36.8', '36.9', '69.0', '44.8', '76.7', '55.8',
       '41.5', '25.3', '71.8', '62.1', '79.2', '69.1', '47.2', '50.7',
       '73.0', '67.9', '75.4', '62.2', '52.6', '52.7', '55.6', '75.2',
       '67.3', '63.1', '46.1', '63.6', '21.7', '68.9', '41.8', '78.1',
       '56.6', '71.7', '32.3', '81.5', '20.1', '76.5', '73.6', '81.6',
       '77.4', '74.5', '20.7', '44.9', '82.1', '49.3', '81.4', '78.7',
       '17.1', '83.3', '81.9', '78.2', '82.9', '62.0', '87.1', '78.8',
       '72.8', '16.6', '39.9', '40.0', '30.8', '79.7', '34.0', '48.6',
       '35.9', '83.5', '79.4', '38.0', '43.1', '24.7', '42.1', '27.8',
       '26.6', '38.4', '44.5', '66.6', '75.8', '74.2', '25.2', '86.8',
       '71.9', '76.2', '84.4', '42.3', '84.6', '85.0', '79.3', '83.0',
       '80.3', '81.8', '84.2', '78.6', '80.2', '29.6', '87.0', '91.5',
       '42.7', '79.5', '82.7', '20.5', '8.2', '15.5', '14.0', '15.8',
       '9.1', '18.8', '25.6', '31.3', '26.4', '27.1', '28.0', '32.8',
       '47.8', '84.1', '86.5', '86.0', '74.4', '81.2', '83.6', '88.6',
       '85.6', '83.2', '90.2', '86.9', '85.9', '86.3', '89.5', '85.1',
       '82.8', '78.9', '88.4', '86.1', '87.7', '89.2', '87.2', '90.4',
       '81.0', '86.2', '83.7', '21.6', '7.7', '12.1', '18.3', '14.5',
       '15.1', '25.5', '20.4', '25.4', '29.7', '39.3', '84.7', '86.7',
       '77.5', '91.6', '87.9', '77.8', '80.9', '87.3', '85.3', '83.8',
       '89.3', '87.4', '87.6', '89.7', '90.1', '18.7', '8.4', '15.9',
       '10.8', '18.0', '8.1', '14.3', '23.1', '26.8', '31.6', '27.0',
       '46.0', '84.0', '80.7', '83.4', '88.1', '84.5', '81.1', '90.0',
       '80.0', '88.0', '84.8', '84.3', '82.3', '89.1', '85.5', '89.0',
       '88.2', '92.1', '83.1', '24.9', '8.5', '19.0', '17.7', '15.7',
       '21.4', '13.8', '25.7', '23.9', '30.9', '24.1', '88.5', '91.7',
       '90.6', '87.5', '91.8', '82.4', '84.9', '87.8', '89.6', '80.6',
       '11.2', '21.8', '31.1', '29.0', '19.9', '32.7', '79.0', '18.5',
       '26.7', '22.9', '30.3', '37.2', '85.7', '88.7', '86.4', '91.3',
       '79.9', '85.4', '14.9', '23.6', '80.1', '93.7', '82.5', '93.0',
       '98.2', '88.8', '18.6', '5.2', '13.7', '13.6', '13.4', '26.2',
       '27.9', '31.4', '47.1', '91.0', '91.4', '78.3', '89.9'],
      dtype=object)
```

```
In [20]: new_data['Estimate (%)'] = new_data['Estimate (%)'].dropna()
```

```
In [21]: # The column is in str so typecasting it to float for numerical computations
         new_data['Estimate (%)'] = new_data['Estimate (%)'].astype(float)
```

```
In [22]: new_data['Estimate (%)'][0]
```

```
Out[22]: 49.2
```

```
In [23]: new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4137 entries, 0 to 4136
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Vaccine                       4137 non-null   object
 1   Geography Type                4137 non-null   object
 2   Geography                     4137 non-null   object
 3   Survey Year/Influenza Season  4137 non-null   int64
 4   Dimension Type                4137 non-null   object
 5   Dimension                     4137 non-null   object
 6   Estimate (%)                  3715 non-null   float64
 7   95% CI (%)                    4137 non-null   object
 8   Sample Size                   3933 non-null   float64
dtypes: float64(2), int64(1), object(6)
memory usage: 291.0+ KB
```

```
In [24]: # Dropping all the null values
         new_data = new_data.dropna()
```

```
In [25]: new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3715 entries, 0 to 4136
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Vaccine                       3715 non-null   object
 1   Geography Type                3715 non-null   object
 2   Geography                     3715 non-null   object
 3   Survey Year/Influenza Season  3715 non-null   int64
 4   Dimension Type                3715 non-null   object
 5   Dimension                     3715 non-null   object
 6   Estimate (%)                  3715 non-null   float64
 7   95% CI (%)                    3715 non-null   object
 8   Sample Size                   3715 non-null   float64
dtypes: float64(2), int64(1), object(6)
memory usage: 290.2+ KB
```

```
In [26]: new_data.describe()
```

Out[26]:

	Survey Year/Influenza Season	Estimate (%)	Sample Size
count	3715.000000	3715.000000	3715.000000
mean	2016.241184	61.275074	849.202153
std	2.648391	15.169773	2728.862515
min	2012.000000	5.200000	32.000000
25%	2014.000000	52.000000	191.000000
50%	2017.000000	62.700000	346.000000
75%	2019.000000	72.100000	685.500000
max	2020.000000	98.200000	43737.000000

Central Tendency Measures

```
In [27]: # Calculate Mean
         new_data['Estimate (%)'].mean()
```

```
Out[27]: 61.27507402422611 4
```

```
In [28]: # Calculate Median
         new_data['Estimate (%)'].median()
```

```
Out[28]: 62.7
```

```
In [29]: # Calculate Mode
         new_data['Estimate (%)'].mode()
```

```
Out[29]: 0    57.8
         Name: Estimate (%), dtype: float64
```

```
In [30]: # Calculate Quantile
         new_data['Estimate (%)'].quantile()
```

```
Out[30]: 62.7
```

Measure of dispersion

```
In [31]: # Calculate Variance
         new_data['Estimate (%)'].var()
```

```
Out[31]: 230.12201180792763
```

```
In [32]: # Calculate Standard Deviation
         new_data['Estimate (%)'].std()
```

```
Out[32]: 15.169772964943398
```

```
In [33]: # Correlation
         # Calculate Variance
         new_data['Estimate (%)'].corr(new_data['Sample Size'])
```

```
Out[33]: -0.019988084872987543
```

```
In [35]: new_data.sample(2)
```

Out[35]:

	Vaccine	Geography Type	Geography	Survey Year/Influenza Season	Dimension Type	Dimension	Estimate (%)	95% CI (%)	Sample Size
3513	Tdap	National	United States	2020	Age	≥35 Years	78.7	76.8 to 80.5	4108.0
1258	Influenza	States	Arkansas	2015	Age	25-34 Years	66.9	60.6 to 72.8	428.0

```
In [37]: new_data['Vaccine'].value_counts()
```

```
Out[37]: Influenza    2573
         Tdap         1142
         Name: Vaccine, dtype: int64
```

```
In [38]: # Create Frequency Table
         pd.crosstab(index=new_data['Vaccine'], columns='count')
```

Out[38]:

col_0	count
Vaccine	
Influenza	2573
Tdap	1142

```
In [39]: # Create two-way table
         pd.crosstab(index=new_data['Geography Type'], columns=new_data['Vaccine'])
```

Out[39]:

Vaccine	Influenza	Tdap
Geography Type		
National	80	72
States	2493	1070

```
In [40]: # Two-way table probability
         pd.crosstab(index=new_data['Geography Type'], columns=new_data['Vaccine'], normalize=True)
```

Out[40]:

Vaccine	Influenza	Tdap
Geography Type		
National	0.021534	0.019381
States	0.671063	0.288022

```
In [41]: # Two-way table conditional probability
         pd.crosstab(index=new_data['Geography Type'], columns=new_data['Vaccine'], normalize='index', margins=True)
```

Out[41]:

Vaccine	Influenza	Tdap
Geography Type		
National	0.526316	0.473684
States	0.699691	0.300309
All	0.692598	0.307402

```
In [42]: # Two-way table conditional probability
         pd.crosstab(index=new_data['Geography Type'], columns=new_data['Vaccine'], normalize='columns', margins=True)
```

Out[42]:

Vaccine	Influenza	Tdap	All
Geography Type			
National	0.031092	0.063047	0.040915
States	0.968908	0.936953	0.959085

```
In [43]: num_data = new_data.select_dtypes(exclude=[object])
```

```
In [44]: num_data.columns
```

```
Out[44]: Index(['Survey Year/Influenza Season', 'Estimate (%)', 'Sample Size'], dtype='object')
```

```
In [45]: # correlation matrix
         corr_data = num_data.corr()
         corr_data
```

Out[45]:

	Survey Year/Influenza Season	Estimate (%)	Sample Size
Survey Year/Influenza Season	1.00000	0.534680	-0.013620
Estimate (%)	0.53468	1.000000	-0.019988
Sample Size	-0.01362	-0.019988	1.000000

Additional Tasks

```
In [51]: # kurtosis
         kurt_data = new_data['Estimate (%)'].kurt(axis=0)
         kurt_data
```

```
Out[51]: 0.31221625194292413
```

```
In [63]: # Skewness of the data
         skew_data = new_data['Sample Size'].skew(axis=0)
         skew_data
```

```
Out[63]: 9.487462004182909
```

```
In [54]: new_data.describe()
```

Out[54]:

	Survey Year/Influenza Season	Estimate (%)	Sample Size
count	3715.000000	3715.000000	3715.000000
mean	2016.241184	61.275074	849.202153
std	2.648391	15.169773	2728.862515
min	2012.000000	5.200000	32.000000
25%	2014.000000	52.000000	191.000000
50%	2017.000000	62.700000	346.000000
75%	2019.000000	72.100000	685.500000
max	2020.000000	98.200000	43737.000000

0 quartile = 0 quantile = 0 percentile

1 quartile = 0.25 quantile = 25 percentile

2 quartile = .5 quantile = 50 percentile (median)

3 quartile = .75 quantile = 75 percentile

4 quartile = 1 quantile = 100 percentile

Percentiles go from 0 to 100.

Quartiles go from 1 to 4 (or 0 to 4).

Quantiles can go from anything to anything.

Percentiles and quartiles are examples of quantiles.

```
In [56]: # Quartile
         new_data.quantile(1)
```

```
/var/folders/nd/q79n0hd51vqcq6swmwgkc95w0000gn/T/ipykernel_57145/957225423.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  new_data.quantile(1)
```

```
Out[56]: Survey Year/Influenza Season     2020.0
         Estimate (%)                       98.2
         Sample Size                     43737.0
         Name: 1.0, dtype: float64
```

```
In [73]: # Percentile
         new_data.quantile(0.75)
```

```
/var/folders/nd/q79n0hd51vqcq6swmwgkc95w0000gn/T/ipykernel_57145/2618097402.py:2: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  new_data.quantile(0.75)
```

```
Out[73]: Survey Year/Influenza Season    2019.0
         Estimate (%)                      72.1
         Sample Size                      685.5
         Name: 0.75, dtype: float64
```

```
In [72]: np.percentile(new_data['Sample Size'], 75)
```

```
Out[72]: 685.5
```

```
In [62]: # Quantile
         new_data.quantile()
```

```
/var/folders/nd/q79n0hd51vqcq6swmwgkc95w0000gn/T/ipykernel_57145/3307333788.py:2: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  new_data.quantile()
```

```
Out[62]: Survey Year/Influenza Season    2017.0
         Estimate (%)                      62.7
         Sample Size                      346.0
         Name: 0.5, dtype: float64
```