25th Feb Assignment

March 9, 2023

1 Assignment 24

Consider following code to answer further questions:

```
[4]: import pandas as pd
     course_name = ['Data Science', 'Machine Learning', 'Big Data', 'Data Engineer']
     duration = [2,3,6,4]
     df = pd.DataFrame(data = {'course_name' : course_name, 'duration' : duration})
```

```
[5]: df
```

```
[5]:         course_name  duration
     0      Data Science         2
     1  Machine Learning         3
     2          Big Data         6
     3     Data Engineer         4
```

Q1. Write a code to print the data present in the second row of the dataframe, df.

```
[16]: df.loc[1]
```

```
[16]: course_name    Machine Learning
      duration                      3
      Name: 1, dtype: object
```

Q2. What is the difference between the functions loc and iloc in pandas.DataFrame?

Ans.”loc” uses label-based indexing, which means that you can select rows and columns of the DataFrame based on their labels.

```
[17]: df1 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6], 'C': [7, 8, 9]},␣
       ↪index=['a', 'b', 'c'])
```

```
[18]: df1
```

```
[18]:    A  B  C
      a  1  4  7
      b  2  5  8
      c  3  6  9
```

```
[19]: df1.loc['a', 'B']
```

```
[19]: 4
```

```
[20]: df1.loc[['a', 'c'], ['A', 'C']]
```

```
[20]:    A  C
      a  1  7
      c  3  9
```

"iloc", on the other hand, uses integer-based indexing, which means that you can select rows and columns of the DataFrame based on their integer positions.

```
[21]: df2 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6], 'C': [7, 8, 9]})
```

```
[23]: df2
```

```
[23]:    A  B  C
      0  1  4  7
      1  2  5  8
      2  3  6  9
```

```
[24]: df2.iloc[0, 1]
```

```
[24]: 4
```

```
[25]: df2.iloc[[0, 2], [0, 2]]
```

```
[25]:    A  C
      0  1  7
      2  3  9
```

Q3. Reindex the given dataframe using a variable, reindex = [3,0,1,2] and store it in the variable, new_df then find the output for both new_df.loc[2] and new_df.iloc[2].

Did you observe any difference in both the outputs? If so then explain it.

```
[28]: reindex=[3,0,1,2]
      new_df=df.reindex(index=reindex)
```

```
[29]: new_df
```

```
[29]:         course_name  duration
      3     Data Engineer         4
      0      Data Science         2
      1  Machine Learning         3
      2          Big Data         6
```

```
[30]: new_df.loc[2]
```

```
[30]: course_name    Big Data
      duration              6
      Name: 2, dtype: object
```

```
[31]: new_df.iloc[2]
```

```
[31]: course_name    Machine Learning
      duration                      3
      Name: 1, dtype: object
```

there is a difference in the output for new_df.loc[2] and new_df.iloc[2].

new_df.loc[2] returns the row of the DataFrame with the label/index value of 2, after reindexing. In this case, since the DataFrame was reindexed using the list [3, 0, 1, 2], the label 2 now refers to the row with index 1 in the original DataFrame.

Consider the below code to answer further questions:

```
[32]: import pandas as pd
      import numpy as np
      columns = ['column_1', 'column_2', 'column_3', 'column_4', 'column_5',␣
        ↪'column_6']
      indices = [1,2,3,4,5,6]
      #Creating a dataframe:
      df1 = pd.DataFrame(np.random.rand(6,6), columns = columns, index = indices)
```

```
[33]: df1
```

```
[33]:    column_1  column_2  column_3  column_4  column_5  column_6
      1  0.543344  0.136161  0.476234  0.857043  0.453316  0.875359
      2  0.939084  0.300773  0.065580  0.085652  0.298139  0.692263
      3  0.664923  0.931853  0.268545  0.890760  0.533198  0.750057
      4  0.170174  0.586955  0.027306  0.740656  0.929891  0.168259
      5  0.257238  0.500970  0.416445  0.012688  0.723541  0.873237
      6  0.657956  0.429788  0.376824  0.632009  0.760088  0.890346
```

Q4. Write a code to find the following statistical measurements for the above dataframe df1:

(i) mean of each and every column present in the dataframe.

Ans.

```
[35]: df1.mean()
```

```
[35]: column_1    0.538787
      column_2    0.481083
      column_3    0.271822
      column_4    0.536468
      column_5    0.616362
      column_6    0.708253
      dtype: float64
```

(ii) standard deviation of column, 'column_2'

Ans.

```
[39]: df1['column_2'].std()
```

```
[39]: 0.27162911294745395
```

Q5. Replace the data present in the second row of column, 'column_2' by a string variable then find the mean of column, column_2.

If you are getting errors in executing it then explain why. [Hint: To replace the data use df1.loc[] and equate this to string data of your choice.]

```
[43]: df1.loc[2,'column_2']='jp'
```

```
[44]: df1
```

```
[44]:    column_1  column_2  column_3  column_4  column_5  column_6
      1  0.543344  0.136161  0.476234  0.857043  0.453316  0.875359
      2  0.939084        jp  0.065580  0.085652  0.298139  0.692263
      3  0.664923  0.931853  0.268545  0.890760  0.533198  0.750057
      4  0.170174  0.586955  0.027306  0.740656  0.929891  0.168259
      5  0.257238   0.50097  0.416445  0.012688  0.723541  0.873237
      6  0.657956  0.429788  0.376824  0.632009  0.760088  0.890346
```

```
[45]: df1['column_2'].mean()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 df1['column_2'].mean()

File /opt/conda/lib/python3.10/site-packages/pandas/core/generic.py:11847, in␣
  ↪NDFrame._add_numeric_operations.<locals>.mean(self, axis, skipna, level,␣
  ↪numeric_only, **kwargs)
  11829 @doc(
  11830     _num_doc,
  11831     desc="Return the mean of the values over the requested axis.",
   (…)
  11845     **kwargs,
  11846 ):
> 11847     return␣
↪NDFrame.mean(self, axis, skipna, level, numeric_only, **kwargs)

File /opt/conda/lib/python3.10/site-packages/pandas/core/generic.py:11401, in␣
↪NDFrame.mean(self, axis, skipna, level, numeric_only, **kwargs)
  11393 def mean(
  11394     self,
  11395     axis: Axis | None | lib.NoDefault = lib.no_default,
   (…)
  11399     **kwargs,
  11400 ) -> Series | float:
> 11401     return self._stat_function(
  11402␣
↪        "mean", nanops.nanmean, axis, skipna, level, numeric_only, **kwargs
  11403     )

File /opt/conda/lib/python3.10/site-packages/pandas/core/generic.py:11353, in␣
↪NDFrame._stat_function(self, name, func, axis, skipna, level, numeric_only,␣
↪**kwargs)
  11343     warnings.warn(
  11344         "Using the level keyword in DataFrame and Series aggregations i␣
↪"
  11345         "deprecated and will be removed in a future version. Use groupb␣
↪"
   (…)
  11348         stacklevel=find_stack_level(),
  11349     )
  11350     return self._agg_by_level(
  11351         name, axis=axis, level=level, skipna=skipna,␣
↪numeric_only=numeric_only
  11352     )
> 11353 return self._reduce(
  11354     func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_onl
  11355 )

File /opt/conda/lib/python3.10/site-packages/pandas/core/series.py:4816, in␣
↪Series._reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4812     raise NotImplementedError(
   4813         f"Series.{name} does not implement {kwd_name}."
   4814     )
   4815 with np.errstate(all="ignore"):
-> 4816     return op(delegate, skipna=skipna, **kwds)

File /opt/conda/lib/python3.10/site-packages/pandas/core/nanops.py:93, in␣
↪disallow.__call__.<locals>._f(*args, **kwargs)
     91 try:
```

```
     92     with np.errstate(invalid="ignore"):
---> 93         return f(*args, **kwargs)
     94 except ValueError as e:
     95     # we want to transform an object array
     96     # ValueError message to the more typical TypeError
     97     # e.g. this is normally a disallowed function on
     98     # object arrays that contain strings
     99     if is_object_dtype(args[0]):

File /opt/conda/lib/python3.10/site-packages/pandas/core/nanops.py:155, in␣
  ↪bottleneck_switch.__call__.<locals>.f(values, axis, skipna, **kwds)
    153         result = alt(values, axis=axis, skipna=skipna, **kwds)
    154 else:
--> 155     result = alt(values, axis=axis, skipna=skipna, **kwds)
    157 return result

File /opt/conda/lib/python3.10/site-packages/pandas/core/nanops.py:418, in␣
  ↪_datetimelike_compat.<locals>.new_func(values, axis, skipna, mask, **kwargs)
    415 if datetimelike and mask is None:
    416     mask = isna(values)
--> 418 result = func(values, axis=axis, skipna=skipna, mask=mask, **kwargs)
    420 if datetimelike:
    421     result = _wrap_results(result, orig_values.dtype, fill_value=iNaT)

File /opt/conda/lib/python3.10/site-packages/pandas/core/nanops.py:706, in␣
  ↪nanmean(values, axis, skipna, mask)
    703     dtype_count = dtype
    705 count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 706 the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    708 if axis is not None and getattr(the_sum, "ndim", False):
    709     count = cast(np.ndarray, count)

File /opt/conda/lib/python3.10/site-packages/numpy/core/_methods.py:48, in␣
  ↪_sum(a, axis, dtype, out, keepdims, initial, where)
     46 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     47          initial=_NoValue, where=True):
---> 48     return umr_sum(a, axis, dtype, out, keepdims, initial, where)

TypeError: unsupported operand type(s) for +: 'float' and 'str'
```

df1['column_2'].mean() gives an error because after replacing the value with a string, the mean() function will raise a TypeError because it can only calculate the mean of numerical data.

Q6. What do you understand about the windows function in pandas and list the types of windows functions?

Ans.There are five types of windows function

- Rolling: Calculates a metric over a rolling window of a fixed size, such as the mean or the sum of the previous n rows.
- Expanding: Calculates a metric over an expanding window that includes all previous rows, such as the cumulative sum or the cumulative maximum.
- Exponentially weighted: Calculates a weighted average over a window of data, where the weights decay exponentially over time, giving more weight to recent data.
- Rolling apply: Applies a custom function to a rolling window of data, where the window size can be fixed or variable.
- Groupby rolling: Calculates a metric over a rolling window within each group of a DataFrame, such as the group mean or the group sum of the previous n rows.

Q7. Write a code to print only the current month and year at the time of answering this question. [Hint: Use pandas.datetime function]

Ans.

```
[47]: import datetime
```

```
[48]: current_date=pd.datetime.now()
```

```
/tmp/ipykernel_3168/1591506915.py:1: FutureWarning: The pandas.datetime class is
deprecated and will be removed from pandas in a future version. Import from
datetime module instead.
  current_date=pd.datetime.now()
```

```
[61]: current_month=current_date.strftime('%B')
      current_year=current_date.strftime('%Y')
```

```
[62]: current_month
```

```
[62]: 'March'
```

```
[63]: current_year
```

```
[63]: '2023'
```

Q8. Write a Python program that takes in two dates as input (in the format YYYY-MM-DD) and calculates the difference between them in days, hours, and minutes using Pandas time delta. The program should prompt the user to enter the dates and display the result.

Ans.

```
[65]: date1=pd.to_datetime(input('Enter date 1:'))
      date2=pd.to_datetime(input('Enter date 2:'))
```

```
Enter date 1: 27-05-2023

/tmp/ipykernel_3168/1270883585.py:1: UserWarning: Parsing dates in DD/MM/YYYY
format when dayfirst=False (the default) was specified. This may lead to
inconsistently parsed dates! Specify a format to ensure consistent parsing.
  date1=pd.to_datetime(input('Enter date 1:'))

Enter date 2: 31-10-2023

/tmp/ipykernel_3168/1270883585.py:2: UserWarning: Parsing dates in DD/MM/YYYY
format when dayfirst=False (the default) was specified. This may lead to
inconsistently parsed dates! Specify a format to ensure consistent parsing.
  date2=pd.to_datetime(input('Enter date 2:'))
```

```
[70]: time_diff=abs(date1-date2)
```

```
[75]: time_diff
```

```
[75]: Timedelta('157 days 00:00:00')
```

```
[76]: time_diff.days
```

```
[76]: 157
```

```
[77]: time_diff.days*24
```

```
[77]: 3768
```

```
[78]: time_diff.days*24*24
```

```
[78]: 90432
```

Q9. Write a Python program that reads a CSV file containing categorical data and converts a specified column to a categorical data type. The program should prompt the user to enter the file path, column name, and category order, and then display the sorted data.

```
[80]: file_path=input('Enter file path:')
```

```
Enter file path: ./Python and Its libraries/25th  Feb Assignment/services.csv
```

```
[81]: col_name = input("Enter the column name :")
```

```
Enter the column name : status
```

```
[82]: cat_order = input("Enter the cat order :")
```

```
Enter the cat order : active,inactive,defunct
```

```
[83]: df= pd.read_csv(file_path)
```

[86]:
```
df[col_name] = pd.Categorical(df[col_name], categories=cat_order.split(","))
```

[89]:
```
df_sorted = df.sort_values(by=col_name)
```

[90]:
```
df_sorted
```

[90]:
```
    id  location_id  program_id       accepted_payments  \
0    1            1         NaN                     NaN   
18  19           19         NaN                     NaN   
17  18           18         NaN                     NaN   
16  17           17         NaN                     NaN   
15  16           16         NaN                     NaN   
14  15           15         NaN                     NaN   
13  14           14         NaN                     NaN   
12  13           13         NaN                     NaN   
21  22           22         NaN  Cash, Check, Credit Card   
10  11           11         NaN                     NaN   
11  12           12         NaN                     NaN   
8    9            9         NaN                     NaN   
7    8            8         NaN                     NaN   
6    7            7         NaN                     NaN   
5    6            6         NaN                     NaN   
4    5            5         NaN                     NaN   
3    4            4         NaN                     NaN   
2    3            3         NaN                     NaN   
1    2            2         NaN                     NaN   
9   10           10         NaN                     NaN   
22  23           22         NaN                     NaN   
20  21           21         NaN                     NaN   
19  20           20         NaN                     NaN   

           alternate_name                                application_process  \
0                     NaN                         Walk in or apply by phone.   
18                    NaN     Call for screening appointment (650-347-3648).   
17                    NaN  Call for screening appointment. Medical visits…   
16                    NaN                              Call for information.   
15                    NaN  Walk in. Written application, identification r…   
14                    NaN          Walk in or through other agency referral.   
13                    NaN  Call for appointment. Referral from human serv…   
12                    NaN  Walk in. Proof of California residency require…   
21  Fotos para pasaportes                  Walk in or apply by phone or mail   
10                    NaN  Walk in. Proof of California residency require…   
11                    NaN  Walk in or apply by phone, email or webpage re…   
8                     NaN  Walk in. Proof of residency in California requ…   
7                     NaN                                    Apply by phone.   
6                     NaN  Apply by phone or be referred by a doctor, soc…   
5                     NaN  Walk in or apply by phone for membership appli…   
```

```
4                     NaN                             Phone for information.
3                     NaN                                    Apply by phone.
2                     NaN        Phone for information (403-4300 Ext. 4322).
1                     NaN                 Apply by phone for an appointment.
9                     NaN  Walk in. Proof of California residency to rece…
22                    NaN                  Walk in or apply by phone or mail
20                    NaN                    By phone during business hours.
19                    NaN                                           Walk in.

                                             audience  \
0   Older adults age 55 or over, ethnic minorities…
18                                                NaN
17                                                NaN
16                                                NaN
15                                                NaN
14  Adult alcoholic/drug addictive men and women w…
13  Individuals or families with low or no income …
12                                                NaN
21  Profit and nonprofit businesses, the public, m…
10                                                NaN
11  Adults, parents, children in 1st-12th grades i…
8      Ethnic minorities, especially Spanish speaking
7   Senior citizens age 60 or over, disabled indiv…
6   Older adults who have memory or sensory loss, …
5                                             Any age
4   Low-income working families with children tran…
3   Parents, children, families with problems of c…
2   Older adults age 55 or over who can benefit fr…
1        Residents of San Mateo County age 55 or over
9                                                 NaN
22  Second service and nonprofit businesses, the p…
20                                                NaN
19                                                NaN

                                          description  \
0   A walk-in center for older adults that provide…
18  Provides free medical and dental care to those…
17  By appointment only, Project Smile provides a …
16  Provides emergency food, clothing and furnitur…
15  Provides emergency assistance including food a…
14  Provides a long-term (6-12 month) residential …
13  Provides food, clothing, bus tokens and shelte…
12  Provides general reading materials, including …
21  [NOTE THIS IS NOT A REAL SERVICE--THIS IS FOR …
10  Provides general reading materials, including …
11  Offers an intergenerational literacy program f…
8   Provides general reading material, including b…
```

```
7   Delivers a hot meal to the home of persons age…
6   Rosener House is a day center for older adults…
5   A multipurpose center offering a wide variety …
4   Provides fixed 8% short term loans to eligible…
3   Provides supervised visitation services and a …
2   Offers supportive counseling services to San M…
1   Provides training and job placement to eligibl…
9   Provides general reading and media materials, …
22  [NOTE THIS IS NOT A REAL ORGANIZATION--THIS IS…
20                                just a test service
19              no unrequired fields for this service

                                          eligibility                  email  \
0   Age 55 or over for most programs, age 60 or ov…                   NaN
18    Low-income person without access to health care                    NaN
17    Low-income person without access to health care                    NaN
16                                Low-income families                    NaN
15                      None for emergency assistance                    NaN
14  Age 21-60, detoxed, physically able and willin…                   NaN
13  None for most services. For emergency assistan…                   NaN
12    Resident of California to obtain a library card                    NaN
21                                               None  passports@example.org
10  Resident of California to obtain a library car…                   NaN
11  English-speaking adult reading at or below 7th…                   NaN
8     Resident of California to obtain a library card                    NaN
7             Homebound person unable to cook or shop                    NaN
6                                      Age 18 or over                    NaN
5                                                None                    NaN
4   Eligibility: Low-income family with legal cust…                   NaN
3                                                None                    NaN
2          Resident of San Mateo County age 55 or over                    NaN
1   Age 55 or over, county resident and willing an…                   NaN
9             Resident of California to obtain a card                    NaN
22                                               None                    NaN
20                                                NaN                    NaN
19                                                NaN                    NaN

    …                            interpretation_services  \
0   …                                                NaN
18  …                                                NaN
17  …                                                NaN
16  …                                                NaN
15  …                                                NaN
14  …                                                NaN
13  …                                                NaN
12  …                                                NaN
21  …  We offer 3-way interpretation services over th…
```

```
10  …                                                 NaN
11  …                                                 NaN
8   …                                                 NaN
7   …                                                 NaN
6   …                                                 NaN
5   …                                                 NaN
4   …                                                 NaN
3   …                                                 NaN
2   …                                                 NaN
1   …                                                 NaN
9   …                                                 NaN
22  …                                                 NaN
20  …                                                 NaN
19  …                                                 NaN

                                              keywords languages  \
0   ADULT PROTECTION AND CARE SERVICES, Meal Sites…       NaN
18  HEALTH SERVICES, Outpatient Care, Community Cl…       NaN
17  HEALTH SERVICES, Outpatient Care, Community Cl…       NaN
16  COMMODITY SERVICES, Clothing/Personal Items, C…       NaN
15  COMMODITY SERVICES, Clothing/Personal Items, C…       NaN
14  ALCOHOLISM SERVICES, Residential Care, DRUG AB…       NaN
13  COMMUNITY SERVICES, Interpretation/Translation…       NaN
12  EDUCATION SERVICES, Library, Libraries, Public…       NaN
21                                   Salud, Medicina   Spanish
10  EDUCATION SERVICES, Library, Libraries, Public…       NaN
11  EDUCATION SERVICES, Adult, Alternative, Litera…       NaN
8   EDUCATION SERVICES, Library, Libraries, Public…       NaN
7   ADULT PROTECTION AND CARE SERVICES, Meal Sites…       NaN
6   ADULT PROTECTION AND CARE SERVICES, Adult Day …       NaN
5   ADULT PROTECTION AND CARE SERVICES, In-Home Su…       NaN
4      COMMUNITY SERVICES, Speakers, Automobile Loans       NaN
3   INDIVIDUAL AND FAMILY DEVELOPMENT SERVICES, Gr…       NaN
2   Geriatric Counseling, Older Adults, Gay, Lesbi…       NaN
1   EMPLOYMENT/TRAINING SERVICES, Job Development,…       NaN
9   EDUCATION SERVICES, Library, Libraries, Public…       NaN
22          Ruby on Rails/Postgres/Redis, testing, wic       NaN
20                                                 NaN       NaN
19                                                 NaN       NaN

                                    name  \
0        Fair Oaks Adult Activity Center
18         San Mateo Free Medical Clinic
17                         Project Smile
16     South San Francisco Citadel Corps
15                       Sunnyvale Corps
14           Adult Rehabilitation Center
```

```
13                  Redwood City Corps
12               Redwood Shores Branch
21                     Passport Photos
10                     Schaberg Branch
11                        Project Read
8                     Fair Oaks Branch
7       Meals on Wheels - South County
6     Rosener House Adult Day Services
5   Little House Recreational Activities
4    Economic Self-Sufficiency Program
3             Family Visitation Center
2               Senior Peer Counseling
1     Second Career Employment Program
9                         Main Library
22                Example Service Name
20     Service for Admin Test Location
19           Service with blank fields

                          required_documents  \
0                                        NaN
18                                       NaN
17                                       NaN
16                                       NaN
15                                       NaN
14                                       NaN
13                                       NaN
12                                       NaN
21  Government-issued picture identification
10                                       NaN
11                                       NaN
8                                        NaN
7                                        NaN
6                                        NaN
5                                        NaN
4                                        NaN
3                                        NaN
2                                        NaN
1                                        NaN
9                                        NaN
22                                       NaN
20                                       NaN
19                                       NaN

                            service_areas    status  \
0                                   Colma    active
18                    Belmont, Burlingame    active
17                         East Palo Alto    active
```

```
16  Colma, Daly City, South San Francisco    active
15                                    NaN    active
14       Alameda County, San Mateo County    active
13    Belmont, Burlingame, East Palo Alto    active
12                     San Mateo County    active
21       Alameda County, San Mateo County    active
10                     San Mateo County    active
11                            Daly City    active
8                      San Mateo County    active
7                 Belmont, East Palo Alto    active
6     Belmont, Burlingame, East Palo Alto    active
5                      San Mateo County    active
4                      San Mateo County    active
3                      San Mateo County    active
2                      San Mateo County    active
1                      San Mateo County    active
9                      San Mateo County    active
22       San Mateo County, Alameda County    active
20                     San Mateo County  inactive
19                                    NaN   defunct

                                             wait_time                 website  \
0                                             No wait.                     NaN
18                                             Varies.                     NaN
17                                             Varies.                     NaN
16                                                 NaN                     NaN
15                                            No wait.                     NaN
14  Varies according to available beds for men and…                     NaN
13                                   Up to 20 minutes.                     NaN
12                                            No wait.                     NaN
21                                 No wait to 2 weeks.  http://www.example.com
10                                            No wait.                     NaN
11  Depends on availability of tutors for small gr…                     NaN
8                                             No wait.                     NaN
7                                             No wait.                     NaN
6                                             No wait.                     NaN
5                                             No wait.                     NaN
4                                                  NaN                     NaN
3                                             No wait.                     NaN
2                                              Varies.                     NaN
1                                              Varies.                     NaN
9                                             No wait.                     NaN
22                                  No wait to 2 weeks  http://www.example.com
20                                                 NaN                     NaN
19                                                 NaN                     NaN

                                taxonomy_ids
```

```
0                                        NaN
18                                       NaN
17                                       NaN
16                                       NaN
15                                       NaN
14                                       NaN
13                                       NaN
12                                       NaN
21  105, 108, 108-05, 108-05-01, 111, 111-05
10                                       NaN
11                                       NaN
8                                        NaN
7                                        NaN
6                                        NaN
5                                        NaN
4                                        NaN
3                                        NaN
2                                        NaN
1                                        NaN
9                                        NaN
22                                       NaN
20                                       NaN
19                                       NaN

[23 rows x 22 columns]
```

Q10. Write a Python program that reads a CSV file containing sales data for different products and visualizes the data using a stacked bar chart to show the sales of each product category over time. The program should prompt the user to enter the file path and display the chart.

```
[92]: file_path=input('Enter file path')
```

```
Enter file path ./Python and Its libraries/25th  Feb Assignment/sales.csv
```

```
[93]: sales_data=pd.read_csv(file_path)
```

```
[94]: sales_data['Date']=pd.to_datetime(sales_data['Date'])
```

```
[101]: sales_group=sales_data.groupby(['Product','Date']).sum().reset_index()
```

```
[102]: sales_group
```

```
[102]:    Product       Date  Sales
       0  OnePlus 2022-06-07    150
       1  OnePlus 2022-06-10    200
       2  OnePlus 2022-06-12    170
```

```
3   Pixel6a 2022-06-06     70
4   Pixel6a 2022-06-09    130
5   Pixel6a 2022-06-11     60
6   Pixel6a 2022-06-14    130
7   Samsung 2022-06-08    120
8   Samsung 2022-06-10    100
9   Samsung 2022-06-12    100
```

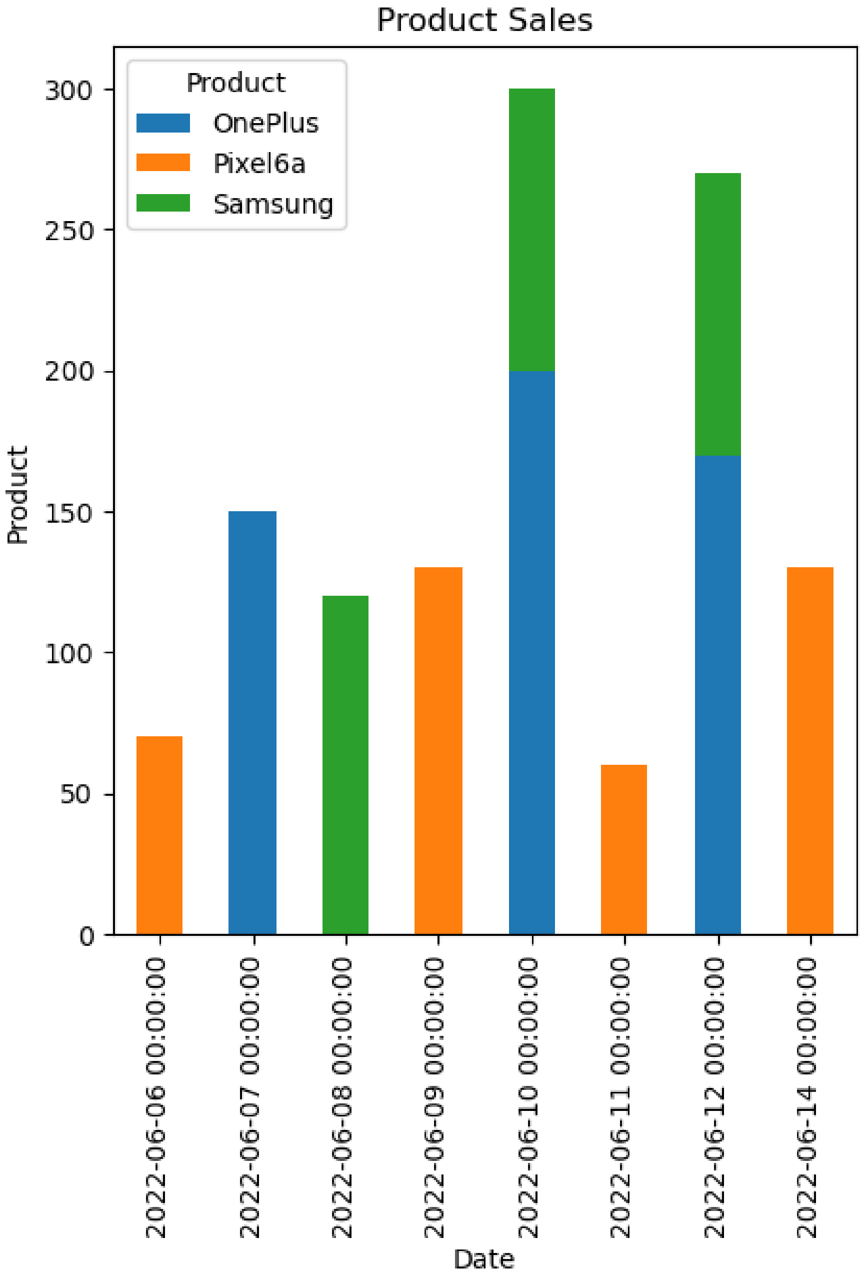
[103]:
```
sales_pivot = sales_group.pivot(index='Date',columns='Product', values='Sales')
```

[105]:
```
sales_pivot
```

[105]:
```
Product      OnePlus  Pixel6a  Samsung
Date
2022-06-06       NaN     70.0      NaN
2022-06-07     150.0      NaN      NaN
2022-06-08       NaN      NaN    120.0
2022-06-09       NaN    130.0      NaN
2022-06-10     200.0      NaN    100.0
2022-06-11       NaN     60.0      NaN
2022-06-12     170.0      NaN    100.0
2022-06-14       NaN    130.0      NaN
```

[108]:
```
sales_pivot.plot(kind='bar',␣
  ↪stacked=True,xlabel='Date',ylabel='Product',title='Product␣
  ↪Sales',figsize=(5,6))
```

[108]:
```
<AxesSubplot: title={'center': 'Product Sales'}, xlabel='Date',
ylabel='Product'>
```



Q11. You are given a CSV file containing student data that includes the student ID and their test score. Write a Python program that reads the CSV file, calculates the mean, median, and mode of the test scores, and displays the results in a table.

The program should do the following

- Prompt the user to enter the file path of the CSV file containing the student dataR
- Read the CSV file into a Pandas DataFrameR

- Calculate the mean, median, and mode of the test scores using Pandas tools
- Display the mean, median, and mode in a table. #### Assume the CSV file contains the following columns
- Student ID: The ID of the student
- Test Score: The score of the student’s test. #### Example usage of the program: #### Enter the file path of the CSV file containing the student data: student_data.csv | Statistic | Value | |————|———| | Mean | 79.6 | | Median | 82 | | Mode | 85, 90 | #### Assume that the CSV file student_data.csv contains the following data: |Student ID|Test Score| |———-|——— |1 |85 |2 |90 |3 |80 |4 |75 |5 |85 |6 |82 |7 |78 |8 |85 |9 |90 |10 |85

The program should calculate the mean, median, and mode of the test scores and display the results in a table.

Ans.

[109]:
```
file_path=input('Enter file path')
```

```
Enter file path ./Python and Its libraries/25th  Feb Assignment/student_data.csv
```

[110]:
```
df=pd.read_csv(file_path)
```

[111]:
```
mean=df['Test Score'].mean()
```

[113]:
```
median=df['Test Score'].median()
```

[117]:
```
mode=df['Test Score'].mode()[0]
```

[119]:
```
new_df=pd.DataFrame({'Statistics':['Mean','Median','Mode'],'Value':
 ↪[mean,median,mode]})
```

[124]:
```
new_df
```

[124]:
```
  Statistics  Value
0       Mean   83.5
1     Median   85.0
2       Mode   85.0
```