

In [65]: `import pandas as pd`

In [66]: `df = pd.read_excel(r'C:\Users\ruperto\Downloads\iNeuron\Others\Attribute.xlsx')`

In [67]: `type(df)`

Out[67]: `pandas.core.frame.DataFrame`

In [68]: `df`

Out[68]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	Material
0	1006032852	Sexy	Low	4.6	M	Summer	o-neck	sleeveless	empire	NaN
1	1212192089	Casual	Low	0.0	L	Summer	o-neck	Petal	natural	microfiber
2	1190380701	vintage	High	0.0	L	Automn	o-neck	full	natural	polyster
3	966005983	Brief	Average	4.6	L	Spring	o-neck	full	natural	silk
4	876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffonfabric
...	...	...	...	...	...	...	...	...	...	...
495	713391965	Casual	Low	4.7	M	Spring	o-neck	full	natural	polyster
496	722565148	Sexy	Low	4.3	free	Summer	o-neck	full	empire	cotton
497	532874347	Casual	Average	4.7	M	Summer	v-neck	full	empire	cotton
498	655464934	Casual	Average	4.6	L	winter	boat-neck	sleeveless	empire	silk
499	919930954	Casual	Low	4.4	free	Summer	v-neck	short	empire	cotton

500 rows × 14 columns

In [69]: `df.head()`

Out[69]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	Material
0	1006032852	Sexy	Low	4.6	M	Summer	o-neck	sleeveless	empire	NaN
1	1212192089	Casual	Low	0.0	L	Summer	o-neck	Petal	natural	microfiber
2	1190380701	vintage	High	0.0	L	Automn	o-neck	full	natural	polyster
3	966005983	Brief	Average	4.6	L	Spring	o-neck	full	natural	silk
4	876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffonfabric

In [70]:

`df.tail()`

Out[70]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	Material	Fa
495	713391965	Casual	Low	4.7	M	Spring	o-neck	full	natural	polyster	
496	722565148	Sexy	Low	4.3	free	Summer	o-neck	full	empire	cotton	
497	532874347	Casual	Average	4.7	M	Summer	v-neck	full	empire	cotton	
498	655464934	Casual	Average	4.6	L	winter	boat-neck	sleeveless	empire	silk	b
499	919930954	Casual	Low	4.4	free	Summer	v-neck	short	empire	cotton	

In [71]:

`df.describe().T`

Out[71]:

	count	mean	std	min	25%	50%	75%
<b>Dress_ID</b>	500.0	9.055417e+08	1.736190e+08	444282011.0	767316420.0	908329553.0	1.039534e+09
<b>Rating</b>	500.0	3.528600e+00	2.005364e+00	0.0	3.7	4.6	4.800000e+00
<b>Recommendation</b>	500.0	4.200000e-01	4.940528e-01	0.0	0.0	0.0	1.000000e+00

In [72]:

`df.corr()`

Out[72]:

	Dress_ID	Rating	Recommendation
<b>Dress_ID</b>	1.000000	-0.561398	-0.075516
<b>Rating</b>	-0.561398	1.000000	0.040847
<b>Recommendation</b>	-0.075516	0.040847	1.000000

In [73]:

`a = pd.read_html("https://en.wikipedia.org/wiki/Data_science")`

In [74]:

`a`

Out[74]:

```
[ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"["}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" "]}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteData \
0 Augmentation Analysis Archaeology Big Cleansin...
```

```
.mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtextr{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"["}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" "]}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteData.1
0 Augmentation Analysis Archaeology Big Cleansin...
]
```

In [75]: `type(a)`

Out[75]: `list`

In [76]: `pwd`

Out[76]: `'C:\\\\Users\\\\rupto'`

In [77]: `df = pd.read_csv(r"C:\\Users\\rupto\\Downloads\\iNeuron\\Data\\sales_data_final.csv")`

In [78]: `df`

	<code>order_id</code>	<code>order_date</code>	<code>ship_date</code>	<code>ship_mode</code>	<code>customer_name</code>	<code>segment</code>	<code>state</code>	<code>country</code>	<code>r</code>
<b>0</b>	AG-2011-2040	1/1/2011	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	
<b>1</b>	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
<b>2</b>	HU-2011-1220	1/1/2011	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	
<b>3</b>	IT-2011-3647632	1/1/2011	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	
<b>4</b>	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	CA-2014-115427	12/31/2014	1/4/2015	Standard Class	Erica Bern	Corporate	California	United States	
<b>51286</b>	MO-2014-2560	12/31/2014	1/5/2015	Standard Class	Liz Preis	Consumer	Souss-Massa-Dra	Morocco	

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	r
51287	MX-2014-110527	12/31/2014	1/2/2015	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	
51288	MX-2014-114783	12/31/2014	1/6/2015	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	
51289	CA-2014-156720	12/31/2014	1/4/2015	Standard Class	Jill Matthias	Consumer	Colorado	United States	

51290 rows × 21 columns

In [79]:

df.head(10)

Out[79]:

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	1/1/2011	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	1/1/2011	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	1/1/2011	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
5	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
6	CA-2011-1510	1/2/2011	1/6/2011	Standard Class	Magdelene Morse	Consumer	Ontario	Canada	Canada
7	IN-2011-79397	1/3/2011	1/3/2011	Same Day	Kean Nguyen	Corporate	New South Wales	Australia	APAC
8	ID-2011-80230	1/3/2011	1/9/2011	Standard Class	Ken Lonsdale	Consumer	Auckland	New Zealand	APAC
9	IZ-2011-4680	1/3/2011	1/7/2011	Standard Class	Lindsay Williams	Corporate	Ninawa	Iraq	EMEA

10 rows × 21 columns

```
In [80]: df.columns
```

```
Out[80]: Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
   'segment', 'state', 'country', 'market', 'region', 'product_id',
   'category', 'sub_category', 'product_name', 'sales', 'quantity',
   'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
  dtype='object')
```

```
In [81]: df[['order_date', 'ship_date']]
```

```
Out[81]:      order_date  ship_date
```

0	1/1/2011	1/6/2011
1	1/1/2011	1/8/2011
2	1/1/2011	1/5/2011
3	1/1/2011	1/5/2011
4	1/1/2011	1/8/2011
...	...	...
51285	12/31/2014	1/4/2015
51286	12/31/2014	1/5/2015
51287	12/31/2014	1/2/2015
51288	12/31/2014	1/6/2015
51289	12/31/2014	1/4/2015

51290 rows × 2 columns

```
In [82]: df[['shipping_cost', 'region']]
```

```
Out[82]:      shipping_cost  region
```

0	35.46	Africa
1	9.72	Oceania
2	8.17	EMEA
3	4.82	North
4	4.70	Oceania
...	...	...
51285	0.89	West
51286	0.49	Africa
51287	0.35	Central

	<b>shipping_cost</b>	<b>region</b>
<b>51288</b>	0.20	North
<b>51289</b>	0.17	West

51290 rows × 2 columns

In [83]: `df.dtypes`

```
Out[83]: order_id          object
order_date        object
ship_date         object
ship_mode         object
customer_name    object
segment           object
state             object
country           object
market            object
region            object
product_id       object
category          object
sub_category     object
product_name     object
sales             object
quantity          int64
discount          float64
profit            float64
shipping_cost    float64
order_priority   object
year              int64
dtype: object
```

In [84]: `df.describe().T`

	<b>count</b>	<b>mean</b>	<b>std</b>	<b>min</b>	<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>max</b>
<b>quantity</b>	51290.0	3.476545	2.278766	1.000	2.00	3.00	5.00	14.000
<b>discount</b>	51290.0	0.142908	0.212280	0.000	0.00	0.00	0.20	0.850
<b>profit</b>	51290.0	28.641740	174.424113	-6599.978	0.00	9.24	36.81	8399.976
<b>shipping_cost</b>	51290.0	26.375915	57.296804	0.000	2.61	7.79	24.45	933.570
<b>year</b>	51290.0	2012.777208	1.098931	2011.000	2012.00	2013.00	2014.00	2014.000

In [85]: `df.dtypes=="object"`

```
Out[85]: order_id          True
order_date        True
ship_date         True
ship_mode         True
customer_name    True
segment           True
state             True
country           True
```

```
market           True
region          True
product_id      True
category         True
sub_category    True
product_name    True
sales            True
quantity        False
discount         False
profit           False
shipping_cost   False
order_priority  True
year             False
dtype: bool
```

In [86]: `df.dtypes=="float"`

```
order_id        False
order_date      False
ship_date       False
ship_mode       False
customer_name   False
segment         False
state           False
country         False
market          False
region          False
product_id      False
category         False
sub_category    False
product_name    False
sales            False
quantity        False
discount         True
profit           True
shipping_cost   True
order_priority  False
year             False
dtype: bool
```

In [87]: `df.dtypes=="int64"`

```
order_id        False
order_date      False
ship_date       False
ship_mode       False
customer_name   False
segment         False
state           False
country         False
market          False
region          False
product_id      False
category         False
sub_category    False
product_name    False
sales            False
quantity        True
discount         False
```

```
profit      False
shipping_cost  False
order_priority  False
year        True
dtype: bool
```

In [88]: `df.dtypes[df.dtypes=="int64"]`

```
Out[88]: quantity    int64
year        int64
dtype: object
```

In [89]: `df.dtypes[df.dtypes=="object"]`

```
Out[89]: order_id      object
order_date     object
ship_date      object
ship_mode      object
customer_name   object
segment        object
state          object
country        object
market          object
region          object
product_id     object
category        object
sub_category    object
product_name    object
sales           object
order_priority  object
dtype: object
```

In [90]: `df[df.dtypes[df.dtypes=="object"].index]`

	<b>order_id</b>	<b>order_date</b>	<b>ship_date</b>	<b>ship_mode</b>	<b>customer_name</b>	<b>segment</b>	<b>state</b>	<b>country</b>	<b>r</b>
<b>0</b>	AG-2011-2040	1/1/2011	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	
<b>1</b>	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
<b>2</b>	HU-2011-1220	1/1/2011	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	
<b>3</b>	IT-2011-3647632	1/1/2011	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	
<b>4</b>	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	CA-2014-115427	12/31/2014	1/4/2015	Standard Class	Erica Bern	Corporate	California	United States	

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	r
51286	MO-2014-2560	12/31/2014	1/5/2015	Standard Class	Liz Preis	Consumer	Souss-Massa-Dra	Morocco	
51287	MX-2014-110527	12/31/2014	1/2/2015	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	
51288	MX-2014-114783	12/31/2014	1/6/2015	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	
51289	CA-2014-156720	12/31/2014	1/4/2015	Standard Class	Jill Matthias	Consumer	Colorado	United States	

51290 rows × 16 columns

In [91]:

```
df[df.dtypes[df.dtypes=="object"].index].describe()
```

Out[91]:

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	ma
<b>count</b>	51290	51290	51290	51290	51290	51290	51290	51290	5
<b>unique</b>	25035	1430	1464	4	795	3	1094	147	
<b>top</b>	CA-2014-100111	6/18/2014	11/22/2014	Standard Class	Muhammed Yedwab	Consumer	California	United States	A
<b>freq</b>	14	135	130	30775	108	26518	2001	9994	1

In [92]:

```
df[df.dtypes[df.dtypes=="float"].index].isna()
```

Out[92]:

	discount	profit	shipping_cost
<b>0</b>	False	False	False
<b>1</b>	False	False	False
<b>2</b>	False	False	False
<b>3</b>	False	False	False
<b>4</b>	False	False	False
<b>...</b>	...	...	...
<b>51285</b>	False	False	False
<b>51286</b>	False	False	False

	discount	profit	shipping_cost
51287	False	False	False
51288	False	False	False
51289	False	False	False

51290 rows × 3 columns

In [93]: `df['order_id']`

```
Out[93]: 0      AG-2011-2040
          1      IN-2011-47883
          2      HU-2011-1220
          3      IT-2011-3647632
          4      IN-2011-47883
          ...
          51285    CA-2014-115427
          51286    MO-2014-2560
          51287    MX-2014-110527
          51288    MX-2014-114783
          51289    CA-2014-156720
Name: order_id, Length: 51290, dtype: object
```

In [94]: `df['order_id'][1:4]`

```
Out[94]: 1      IN-2011-47883
          2      HU-2011-1220
          3      IT-2011-3647632
Name: order_id, dtype: object
```

In [95]: `df.head()`

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	1/1/2011	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	1/1/2011	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	1/1/2011	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 21 columns

In [96]:

```
#convert string amount columns to correct numeric data type
df['sales'] = df['sales'].str.replace(',', '').astype(float)
```

In [97]:

```
##convert string amount columns to numeric data type but ignore all error rows
df['sales'] = pd.to_numeric(df['sales'], errors='coerce')
```

In [98]:

```
df.dtypes
```

Out[98]:

order_id	object
order_date	object
ship_date	object
ship_mode	object
customer_name	object
segment	object
state	object
country	object
market	object
region	object
product_id	object
category	object
sub_category	object
product_name	object
sales	float64
quantity	int64
discount	float64
profit	float64
shipping_cost	float64
order_priority	object
year	int64
dtype:	object

In [99]:

```
df['sales'].isnull().sum()
```

Out[99]:

0

In [100...]

```
df['total_sales'] = df['sales']*df['quantity']
```

In [101...]

```
df.head()
```

Out[101...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	1/1/2011	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
2	HU-2011-1220	1/1/2011	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	1/1/2011	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	1/1/2011	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 22 columns

In [102...]

df['order\_id'].isnull()

Out[102...]

```
0      False
1      False
2      False
3      False
4      False
...
51285  False
51286  False
51287  False
51288  False
51289  False
Name: order_id, Length: 51290, dtype: bool
```

In [103...]

df1 = pd.read\_excel(r'C:\Users\ruperto\Downloads\iNeuron\Data\Retail Sales1.xlsx')

In [104...]

df1

Out[104...]

	Dress_ID	29/8/2013	31/8/2013	2013-	2013-	2013-	2013-	2013-	2013-	
				02-09 00:00:00	04-09 00:00:00	06-09 00:00:00	08-09 00:00:00	10-09 00:00:00	12-09 00:00:00	14/
0	1006032852		2114	2274	2491	2660	2727	2887	2930	3119
1	1212192089		151	275	570	750	813	1066	1164	1558
2	1190380701		6	7	7	7	8	8	9	10
3	966005983		1005	1128	1326	1455	1507	1621	1637	1723
4	876339541		996	1175	1304	1396	1432	1559	1570	1638
...	...	...	...	...	...	...	...	...	...	...
495	713391965		0	0	0	560	554	544	537	525
496	722565148		0	0	0	875	866	861	854	850
497	532874347		0	0	0	734	728	726	715	694
498	655464934		0	0	0	254	259	261	263	268

Dress_ID	29/8/2013	31/8/2013	2013-	2013-	2013-	2013-	2013-	2013-	2013-	
			02-09 00:00:00	04-09 00:00:00	06-09 00:00:00	08-09 00:00:00	10-09 00:00:00	12-09 00:00:00	14/	
499	919930954	0	0	0	538	545	558	563	578	

500 rows × 24 columns

In [105...]

df1.columns

Out[105...]

```
Index(['Dress_ID', '29/8/2013', '31/8/2013',
       2013-02-09 00:00:00, 2013-04-09 00:00:00, 2013-06-09 00:00:00,
       2013-08-09 00:00:00, 2013-10-09 00:00:00, 2013-12-09 00:00:00,
       '14/9/2013', '16/9/2013', '18/9/2013',
       '20/9/2013', '22/9/2013', '24/9/2013',
       '26/9/2013', '28/9/2013', '30/9/2013',
       2013-02-10 00:00:00, 2013-04-10 00:00:00, 2013-06-10 00:00:00,
       2010-08-10 00:00:00, 2013-10-10 00:00:00, 2013-12-10 00:00:00],
      dtype='object')
```

In [106...]

df[df['profit'] == max(df['profit'])]

Out[106...]

order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
----------	------------	-----------	-----------	---------------	---------	-------	---------	--------

29530	CA-2013-118689	10/3/2013	10/10/2013	Standard Class	Tamara Chand	Corporate	Indiana	United States	U.S.
-------	----------------	-----------	------------	----------------	--------------	-----------	---------	---------------	------

1 rows × 22 columns

In [107...]

df[df['profit'] == max(df['profit'])]['customer\_name']

Out[107...]

```
29530    Tamara Chand
Name: customer_name, dtype: object
```

In [108...]

df[df['country'] == 'Sweden']

Out[108...]

order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
----------	------------	-----------	-----------	---------------	---------	-------	---------	--------

3	IT-2011-3647632	1/1/2011	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden
168	IT-2011-5951216	1/13/2011	1/20/2011	Standard Class	Stuart Calhoun	Consumer	Västra Götaland	Sweden
477	IT-2011-5645183	2/3/2011	2/5/2011	Second Class	Max Ludwig	Home Office	Uppsala	Sweden

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	ma
1342	IT-2011-5233011	3/31/2011	4/6/2011	Standard Class	Todd Sumrall	Corporate	Västra Götaland	Sweden	
1554	IT-2011-2419493	4/12/2011	4/13/2011	First Class	Alice McCarthy	Corporate	Stockholm	Sweden	
...	...	...	...	...	...	...	...	...	...
50948	IT-2014-5217096	12/26/2014	12/31/2014	Standard Class	Trudy Glocke	Consumer	Stockholm	Sweden	
50949	IT-2014-5217096	12/26/2014	12/31/2014	Standard Class	Trudy Glocke	Consumer	Stockholm	Sweden	
50972	IT-2014-4046179	12/27/2014	12/30/2014	Second Class	Jeremy Farry	Consumer	Stockholm	Sweden	
51086	IT-2014-2362249	12/29/2014	1/1/2015	First Class	Mitch Gastineau	Corporate	Stockholm	Sweden	
51121	IT-2014-2362249	12/29/2014	1/1/2015	First Class	Mitch Gastineau	Corporate	Stockholm	Sweden	

203 rows × 22 columns

In [109...]

df[df['shipping\_cost'] == 20]

Out[109...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country
15706	CA-2012-101154	9/18/2012	9/21/2012	First Class	Charlotte Melton	Consumer	California	United States
29060	TZ-2013-9020	9/24/2013	9/28/2013	Standard Class	Arthur Prichep	Consumer	Dar es Salaam	Tanzania
35915	SA-2014-840	3/15/2014	3/20/2014	Second Class	Thomas Thornton	Consumer	Ar Riyad	Saudi Arabia
36239	CA-2014-140151	3/24/2014	3/26/2014	First Class	Raymond Buch	Consumer	Washington	United States
36681	ES-2014-2624465	4/6/2014	4/10/2014	Standard Class	Toby Ritter	Consumer	Ile-de-France	France

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country
	47901	IN-2014-37712	11/14/2014	11/16/2014	Second Class	Berenike Kampe	Consumer	Jakarta Indonesia
	51064	IN-2014-77290	12/29/2014	12/31/2014	Second Class	Nick Radford	Consumer	South Australia Australia

7 rows × 22 columns

In [110...]

df[df['shipping\_cost'] &gt; 80]

Out[110...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country
7	IN-2011-79397	1/3/2011	1/3/2011	Same Day	Kean Nguyen	Corporate	New South Wales	Australia
8	ID-2011-80230	1/3/2011	1/9/2011	Standard Class	Ken Lonsdale	Consumer	Auckland	New Zealand
9	IZ-2011-4680	1/3/2011	1/7/2011	Standard Class	Lindsay Williams	Corporate	Ninawa	Iraq
27	MX-2011-111255	1/4/2011	1/9/2011	Second Class	Russell Applegate	Consumer	Parana	Brazil
43	MX-2011-109267	1/5/2011	1/9/2011	Standard Class	Jennifer Halladay	Consumer	Veracruz	Mexico
...	...	...	...	...	...	...	...	...
51154	IN-2014-49206	12/30/2014	1/1/2015	Second Class	Toby Ritter	Consumer	Jawa Barat	Indonesia
51228	MX-2014-116267	12/31/2014	1/3/2015	Second Class	Erica Bern	Corporate	S?o Paulo	Brazil
51229	IN-2014-43550	12/31/2014	1/1/2015	First Class	Marina Lichtenstein	Corporate	Jakarta	Indonesia
51230	RS-2014-1460	12/31/2014	1/2/2015	Second Class	Peter B?hler	Consumer	Bashkortostan	Russia
51231	IN-2014-30390	12/31/2014	1/3/2015	First Class	Justin Deggeller	Corporate	Bangkok	Thailand

3984 rows × 22 columns

```
In [111... df[df['shipping_cost'] > 80]['country']
```

```
Out[111... 7 Australia
8 New Zealand
9 Iraq
27 Brazil
43 Mexico
...
51154 Indonesia
51228 Brazil
51229 Indonesia
51230 Russia
51231 Thailand
Name: country, Length: 3984, dtype: object
```

```
In [112... #filter out
df[(df['shipping_cost'] > 100) & (df['profit'] < 10)]
```

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country
8	ID-2011-80230	1/3/2011	1/9/2011	Standard Class	Ken Lonsdale	Consumer	Auckland	New Zealand
257	MX-2011-146584	1/21/2011	1/27/2011	Standard Class	Hunter Lopez	Consumer	Managua	Nicaragua
451	ES-2011-1640672	2/2/2011	2/5/2011	First Class	Ken Heidel	Corporate	Saxony	Germany
625	ID-2011-47575	2/15/2011	2/21/2011	Standard Class	Christopher Martinez	Consumer	National Capital	Philippines
626	ID-2011-47575	2/15/2011	2/21/2011	Standard Class	Christopher Martinez	Consumer	National Capital	Philippines
...	...	...	...	...	...	...	...	...
50781	CA-2014-147956	12/25/2014	1/1/2015	Standard Class	Alan Hwang	Consumer	Washington	United States
50969	US-2014-120166	12/27/2014	12/31/2014	Standard Class	Frank Hawley	Corporate	Valle del Cauca	Colombia
50970	ES-2014-4296634	12/27/2014	1/2/2015	Standard Class	Harry Marie	Corporate	Brittany	France

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country
51034	ID-68183	2014-12-29	2014-12-31	First Class	Adam Hart	Corporate	Victoria	Australia
51153	ID-51509	2014-12-30	2015-01-01	Second Class	Naresj Patel	Consumer	Australian Capital Territory	Australia

604 rows × 22 columns

In [113... df[['profit','customer\_name']].max()

Out[113... profit 8399.976  
customer\_name Zuschuss Donatelli  
dtype: object

In [114... df.dtypes

Out[114... order\_id object  
order\_date object  
ship\_date object  
ship\_mode object  
customer\_name object  
segment object  
state object  
country object  
market object  
region object  
product\_id object  
category object  
sub\_category object  
product\_name object  
sales float64  
quantity int64  
discount float64  
profit float64  
shipping\_cost float64  
order\_priority object  
year int64  
total\_sales float64  
dtype: objectIn [115... #convert date column into datetime type  
df['order\_date'] = pd.to\_datetime(df['order\_date'])

In [116... df.dtypes

Out[116... order\_id object  
order\_date datetime64[ns]  
ship\_date object  
ship\_mode object  
customer\_name object

```

segment          object
state            object
country          object
market           object
region           object
product_id       object
category          object
sub_category     object
product_name     object
sales            float64
quantity          int64
discount          float64
profit            float64
shipping_cost    float64
order_priority   object
year              int64
total_sales      float64
dtype: object

```

In [117... df['order\_date\_year'] = df['order\_date'].dt.year

In [118... df.head()

Out[118... 

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 23 columns

In [119... df[df['order\_date\_year'] == 2014]

Out[119... 

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	m
33759	ES-2014-2774938	2014-01-01	1/3/2014	Second Class	Fred Harton	Consumer	Ile-de-France	France	

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	m
33760	IN-2014-47134	2014-01-01	1/5/2014	Standard Class	Kean Nguyen	Corporate	Victoria	Australia	
33761	IT-2014-4810734	2014-01-01	1/3/2014	Second Class	Alan Shonely	Consumer	Ile-de-France	France	
33762	US-2014-105830	2014-01-01	1/3/2014	First Class	Duane Benoit	Consumer	Ohio	United States	
33763	CA-2014-160395	2014-01-01	1/7/2014	Standard Class	Kelly Lampkin	Corporate	Nevada	United States	
...	...	...	...	...	...	...	...	...	...
51285	CA-2014-115427	2014-12-31	1/4/2015	Standard Class	Erica Bern	Corporate	California	United States	
51286	MO-2014-2560	2014-12-31	1/5/2015	Standard Class	Liz Preis	Consumer	Souss-Massa-Dra	Morocco	,
51287	MX-2014-110527	2014-12-31	1/2/2015	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	L
51288	MX-2014-114783	2014-12-31	1/6/2015	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	L
51289	CA-2014-156720	2014-12-31	1/4/2015	Standard Class	Jill Matthias	Consumer	Colorado	United States	

17531 rows × 23 columns

In [120...]

df.dtypes

Out[120...]

order_id	object
order_date	datetime64[ns]
ship_date	object
ship_mode	object
customer_name	object
segment	object
state	object
country	object
market	object
region	object
product_id	object
category	object
sub_category	object
product_name	object

```

sales           float64
quantity        int64
discount        float64
profit          float64
shipping_cost   float64
order_priority object
year            int64
total_sales    float64
order_date_year int64
dtype: object

```

In [121...]

```
#create month column
df['order_date_month'] = df['order_date'].dt.month
```

In [122...]

```
df.head()
```

Out[122...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 24 columns

In [123...]

```
#group by
df['order_date_month'].value_counts()
```

Out[123...]

```

11    6342
12    6302
9     5975
6     5331
8     5078
10    4490
5     3747
7     3166
4     3057
3     3035
1     2599
2     2168
Name: order_date_month, dtype: int64

```

In [124...]

```
df['cost_to_compnay'] = df['discount'] + df['shipping_cost']
```

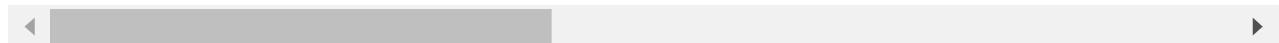
In [125...]

```
df.head()
```

Out[125...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 25 columns



In [126...]

```
df['cost_to_compnay'].max()
```

Out[126...]

933.57

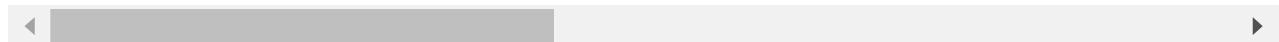
In [127...]

```
df[df['cost_to_compnay'] == max(df['cost_to_compnay'])]
```

Out[127...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
13891	CA-2012-124891	2012-07-31	7/31/2012	Same Day	Rick Hansen	Consumer	New York	United States	US

1 rows × 25 columns



In [128...]

```
df[df['cost_to_compnay'] == max(df['cost_to_compnay'])][[]]
```

```
File "C:\Users\ruperto\AppData\Local\Temp\ipykernel_74932\1398273815.py", line 1
  df[df['cost_to_compnay'] == max(df['cost_to_compnay'])][[]]
^
```

**SyntaxError:** invalid syntax

In [129... df.columns

```
Out[129... Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
   'segment', 'state', 'country', 'market', 'region', 'product_id',
   'category', 'sub_category', 'product_name', 'sales', 'quantity',
   'discount', 'profit', 'shipping_cost', 'order_priority', 'year',
   'total_sales', 'order_date_year', 'order_date_month',
   'cost_to_compnay'],
  dtype='object')
```

In [130... df[df['cost\_to\_compnay'] == max(df['cost\_to\_compnay'])]['product\_name']

```
Out[130... 13891 Plantronics CS510 - Over-the-Head monaural Wir...
Name: product_name, dtype: object
```

In [131... df['total\_sales'].max()

```
Out[131... 135828.0
```

In [132... #filter out max record  
df[df['total\_sales'] == max(df['total\_sales'])]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
<b>1103</b>	CA-2011-145317	2011-03-18	3/23/2011	Standard Class	Sean Miller	Home Office	Florida	United States	US

1 rows × 25 columns

In [133... # delete a column temporarily  
df.drop('cost\_to\_compnay', axis = 1)

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	r
<b>0</b>	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	
<b>1</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
<b>2</b>	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	
<b>3</b>	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	
<b>4</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	r
51285	CA-2014-115427	2014-12-31	1/4/2015	Standard Class	Erica Bern	Corporate	California	United States	...
51286	MO-2014-2560	2014-12-31	1/5/2015	Standard Class	Liz Preis	Consumer	Souss-Massa-Dra	Morocco	...
51287	MX-2014-110527	2014-12-31	1/2/2015	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	...
51288	MX-2014-114783	2014-12-31	1/6/2015	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	...
51289	CA-2014-156720	2014-12-31	1/4/2015	Standard Class	Jill Matthias	Consumer	Colorado	United States	...

51290 rows × 24 columns

In [134...]

df.head()

Out[134...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 25 columns

In [135...]

# delete a column permanently  
df.drop('cost\_to\_comppnay', axis = 1, inplace = True)

In [136...]

df.head()

Out[136...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 24 columns

In [137...]

```
# delete a row temporarily
df.drop(1)
```

Out[137...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	r
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
5	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
...	...	...	...	...	...	...	...	...	...
51285	CA-2014-115427	2014-12-31	1/4/2015	Standard Class	Erica Bern	Corporate	California	United States	
51286	MO-2014-2560	2014-12-31	1/5/2015	Standard Class	Liz Preis	Consumer	Souss-Massa-Dra	Morocco	
51287	MX-2014-	2014-12-31	1/2/2015	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	r
	110527								
51288	MX-2014-114783	2014-12-31	1/6/2015	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	1
51289	CA-2014-156720	2014-12-31	1/4/2015	Standard Class	Jill Matthias	Consumer	Colorado	United States	

51289 rows × 24 columns

In [138...]

df[1:9]

Out[138...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
5	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
6	CA-2011-1510	2011-01-02	1/6/2011	Standard Class	Magdelene Morse	Consumer	Ontario	Canada	Canada
7	IN-2011-79397	2011-01-03	1/3/2011	Same Day	Kean Nguyen	Corporate	New South Wales	Australia	APAC
8	ID-2011-80230	2011-01-03	1/9/2011	Standard Class	Ken Lonsdale	Consumer	Auckland	New Zealand	APAC

8 rows × 24 columns

In [139...]

# row wise filter  
df.loc[[2,9]]

Out[139...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA	
9	IZ-2011-4680	2011-01-03	1/7/2011	Standard Class	Lindsay Williams	Corporate	Ninawa	Iraq	EMEA	

2 rows × 24 columns

◀	▶
---	---

In [140...]

df.loc[[2,9,3,6]]

Out[140...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA	
9	IZ-2011-4680	2011-01-03	1/7/2011	Standard Class	Lindsay Williams	Corporate	Ninawa	Iraq	EMEA	
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU	
6	CA-2011-1510	2011-01-02	1/6/2011	Standard Class	Magdelene Morse	Consumer	Ontario	Canada	Canada	

4 rows × 24 columns

◀	▶
---	---

In [141...]

df.loc[2:12]

Out[141...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	mark	
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EME	
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	E	
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	AP/	
5	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	AP/	
6	CA-2011-1510	2011-01-02	1/6/2011	Standard Class	Magdelene Morse	Consumer	Ontario	Canada	Canad	
7	IN-2011-	2011-01-03	1/3/2011	Same Day	Kean Nguyen	Corporate	New South	Australia	AP/	

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	mark
	79397							Wales	
8	ID-2011-80230	2011-01-03	1/9/2011	Standard Class	Ken Lonsdale	Consumer	Auckland	New Zealand	APAC
9	IZ-2011-4680	2011-01-03	1/7/2011	Standard Class	Lindsay Williams	Corporate	Ninawa	Iraq	EMEA
10	IN-2011-65159	2011-01-03	1/7/2011	Second Class	Larry Blacks	Consumer	National Capital	Philippines	APAC
11	IN-2011-65159	2011-01-03	1/7/2011	Second Class	Larry Blacks	Consumer	National Capital	Philippines	APAC
12	ES-2011-4869686	2011-01-03	1/7/2011	Standard Class	Dorothy Dickinson	Consumer	England	United Kingdom	E

11 rows × 24 columns

In [142...]

df.loc[2:12:4]

Out[142...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
6	CA-2011-1510	2011-01-02	1/6/2011	Standard Class	Magdelene Morse	Consumer	Ontario	Canada	Canada
10	IN-2011-65159	2011-01-03	1/7/2011	Second Class	Larry Blacks	Consumer	National Capital	Philippines	APAC

3 rows × 24 columns

In [143...]

df.loc[[0,1]]

Out[143...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

2 rows × 24 columns

In [144...]

```
# select some rows with some columns
df.loc[0:4,['order_id','order_date','ship_mode']]
```

Out[144...]

	order_id	order_date	ship_mode
<b>0</b>	AG-2011-2040	2011-01-01	Standard Class
<b>1</b>	IN-2011-47883	2011-01-01	Standard Class
<b>2</b>	HU-2011-1220	2011-01-01	Second Class
<b>3</b>	IT-2011-3647632	2011-01-01	Second Class
<b>4</b>	IN-2011-47883	2011-01-01	Standard Class

In [145...]

```
# filter rows columns choice wise
df.iloc[0:3,[2,4]]
```

Out[145...]

	ship_date	customer_name
<b>0</b>	1/6/2011	Toby Braunhardt
<b>1</b>	1/8/2011	Joseph Holt
<b>2</b>	1/5/2011	Annie Thurman

In [146...]

```
df.iloc[0:6,0:6]
```

Out[146...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment
<b>0</b>	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer
<b>1</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer
<b>2</b>	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer
<b>3</b>	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office
<b>4</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer
<b>5</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer

In [147...]

```
df.iloc[[1,4],0:6]
```

Out[147...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment
<b>1</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer
<b>4</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer

In [148...]

```
df.iloc[[1,4],[0,4]]
```

Out[148...]

	order_id	customer_name
1	IN-2011-47883	Joseph Holt
4	IN-2011-47883	Joseph Holt

In [149...]

```
#filter only numeric columns
df.dtypes[(df.dtypes == "float64") | (df.dtypes == "int64")]
```

Out[149...]

sales	float64
quantity	int64
discount	float64
profit	float64
shipping_cost	float64
year	int64
total_sales	float64
order_date_year	int64
order_date_month	int64
dtype:	object

In [150...]

```
df2 = df[df.dtypes[(df.dtypes == "float64") | (df.dtypes == "int64")].index]
```

In [151...]

```
df2
```

Out[151...]

	sales	quantity	discount	profit	shipping_cost	year	total_sales	order_date_year	order_date_
0	408.0	2	0.0	106.1400	35.46	2011	816.0	2011	
1	120.0	3	0.1	36.0360	9.72	2011	360.0	2011	
2	66.0	4	0.0	29.6400	8.17	2011	264.0	2011	
3	45.0	3	0.5	-26.0550	4.82	2011	135.0	2011	
4	114.0	5	0.1	37.7700	4.70	2011	570.0	2011	
...	...	...	...	...	...	...	...	...	...
51285	14.0	2	0.2	4.5188	0.89	2014	28.0	2014	
51286	4.0	1	0.0	0.4200	0.49	2014	4.0	2014	
51287	26.0	3	0.0	12.3600	0.35	2014	78.0	2014	
51288	7.0	1	0.0	0.5600	0.20	2014	7.0	2014	
51289	3.0	3	0.2	-0.6048	0.17	2014	9.0	2014	

51290 rows × 9 columns



In [152...]

```
df2[df2>100]
```

Out[152...]

	sales	quantity	discount	profit	shipping_cost	year	total_sales	order_date_year	order_date_m
0	408.0	NaN	NaN	106.14	NaN	2011	816.0	2011	

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>	<b>order_date_m</b>
<b>1</b>	120.0	NaN	NaN	NaN	NaN	2011	360.0	2011	
<b>2</b>	NaN	NaN	NaN	NaN	NaN	2011	264.0	2011	
<b>3</b>	NaN	NaN	NaN	NaN	NaN	2011	135.0	2011	
<b>4</b>	114.0	NaN	NaN	NaN	NaN	2011	570.0	2011	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51286</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51287</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51288</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51289</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	

51290 rows × 9 columns

In [153...]

df3 = df2[df2&gt;100]

In [154...]

df3

Out[154...]

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>	<b>order_date_m</b>
<b>0</b>	408.0	NaN	NaN	106.14	NaN	2011	816.0	2011	
<b>1</b>	120.0	NaN	NaN	NaN	NaN	2011	360.0	2011	
<b>2</b>	NaN	NaN	NaN	NaN	NaN	2011	264.0	2011	
<b>3</b>	NaN	NaN	NaN	NaN	NaN	2011	135.0	2011	
<b>4</b>	114.0	NaN	NaN	NaN	NaN	2011	570.0	2011	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51286</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51287</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51288</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51289</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	

51290 rows × 9 columns

In [155...]

# temporarily delete na values  
df3.dropna()

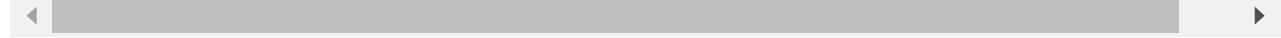
```
Out[155... sales quantity discount profit shipping_cost year total_sales order_date_year order_date_month
```

In [156... df3

Out[156... sales quantity discount profit shipping\_cost year total\_sales order\_date\_year order\_date\_m

	sales	quantity	discount	profit	shipping_cost	year	total_sales	order_date_year	order_date_m
<b>0</b>	408.0	NaN	NaN	106.14	NaN	2011	816.0	2011	
<b>1</b>	120.0	NaN	NaN	NaN	NaN	2011	360.0	2011	
<b>2</b>	NaN	NaN	NaN	NaN	NaN	2011	264.0	2011	
<b>3</b>	NaN	NaN	NaN	NaN	NaN	2011	135.0	2011	
<b>4</b>	114.0	NaN	NaN	NaN	NaN	2011	570.0	2011	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51286</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51287</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51288</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	
<b>51289</b>	NaN	NaN	NaN	NaN	NaN	2014	NaN	2014	

51290 rows × 9 columns



In [157... df3.dropna(axis = 1)

Out[157... year order\_date\_year

<b>0</b>	2011	2011
<b>1</b>	2011	2011
<b>2</b>	2011	2011
<b>3</b>	2011	2011
<b>4</b>	2011	2011
...	...	...
<b>51285</b>	2014	2014
<b>51286</b>	2014	2014
<b>51287</b>	2014	2014
<b>51288</b>	2014	2014
<b>51289</b>	2014	2014

51290 rows × 2 columns

In [158...]

df3

Out[158...]

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>	<b>order_date_m</b>
<b>0</b>	408.0	NaN	NaN	106.14		2011	816.0	2011	
<b>1</b>	120.0	NaN	NaN	NaN		2011	360.0	2011	
<b>2</b>	NaN	NaN	NaN	NaN		2011	264.0	2011	
<b>3</b>	NaN	NaN	NaN	NaN		2011	135.0	2011	
<b>4</b>	114.0	NaN	NaN	NaN		2011	570.0	2011	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	NaN	NaN	NaN	NaN		2014	NaN	2014	
<b>51286</b>	NaN	NaN	NaN	NaN		2014	NaN	2014	
<b>51287</b>	NaN	NaN	NaN	NaN		2014	NaN	2014	
<b>51288</b>	NaN	NaN	NaN	NaN		2014	NaN	2014	
<b>51289</b>	NaN	NaN	NaN	NaN		2014	NaN	2014	

51290 rows × 9 columns



In [159...]

```
# temporarily drop na except those rows has 5 or more values
df3.dropna(thresh = 5)
```

Out[159...]

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>	<b>order_date_i</b>
<b>0</b>	408.0	NaN	NaN	106.140		2011	816.0	2011	
<b>7</b>	276.0	NaN	NaN	110.412	125.32	2011	276.0	2011	
<b>8</b>	912.0	NaN	NaN	NaN	107.10	2011	3648.0	2011	
<b>9</b>	667.0	NaN	NaN	253.320		2011	2668.0	2011	
<b>12</b>	854.0	NaN	NaN	290.430		2011	5978.0	2011	
...	...	...	...	...	...	...	...	...	...
<b>51234</b>	557.0	NaN	NaN	216.720		2014	1671.0	2014	
<b>51236</b>	530.0	NaN	NaN	180.240		2014	1060.0	2014	
<b>51245</b>	277.0	NaN	NaN	105.000		2014	1108.0	2014	
<b>51252</b>	378.0	NaN	NaN	166.440		2014	378.0	2014	
<b>51269</b>	365.0	NaN	NaN	153.090		2014	1095.0	2014	

6822 rows × 9 columns



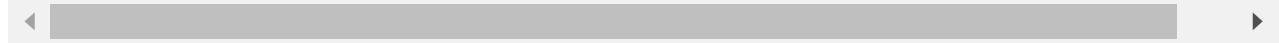
In [160...]

df3.fillna(4)

Out[160...]

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>	<b>order_date_m</b>
<b>0</b>	408.0	4.0	4.0	106.14		4.0	2011	816.0	2011
<b>1</b>	120.0	4.0	4.0	4.00		4.0	2011	360.0	2011
<b>2</b>	4.0	4.0	4.0	4.00		4.0	2011	264.0	2011
<b>3</b>	4.0	4.0	4.0	4.00		4.0	2011	135.0	2011
<b>4</b>	114.0	4.0	4.0	4.00		4.0	2011	570.0	2011
...	...	...	...	...		...	...	...	...
<b>51285</b>	4.0	4.0	4.0	4.00		4.0	2014	4.0	2014
<b>51286</b>	4.0	4.0	4.0	4.00		4.0	2014	4.0	2014
<b>51287</b>	4.0	4.0	4.0	4.00		4.0	2014	4.0	2014
<b>51288</b>	4.0	4.0	4.0	4.00		4.0	2014	4.0	2014
<b>51289</b>	4.0	4.0	4.0	4.00		4.0	2014	4.0	2014

51290 rows × 9 columns



In [161...]

df3

Out[161...]

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>	<b>order_date_m</b>
<b>0</b>	408.0	NaN	NaN	106.14		NaN	2011	816.0	2011
<b>1</b>	120.0	NaN	NaN	NaN		NaN	2011	360.0	2011
<b>2</b>	NaN	NaN	NaN	NaN		NaN	2011	264.0	2011
<b>3</b>	NaN	NaN	NaN	NaN		NaN	2011	135.0	2011
<b>4</b>	114.0	NaN	NaN	NaN		NaN	2011	570.0	2011
...	...	...	...	...		...	...	...	...
<b>51285</b>	NaN	NaN	NaN	NaN		NaN	2014	NaN	2014
<b>51286</b>	NaN	NaN	NaN	NaN		NaN	2014	NaN	2014
<b>51287</b>	NaN	NaN	NaN	NaN		NaN	2014	NaN	2014
<b>51288</b>	NaN	NaN	NaN	NaN		NaN	2014	NaN	2014
<b>51289</b>	NaN	NaN	NaN	NaN		NaN	2014	NaN	2014

51290 rows × 9 columns



In [162...]

```
# fill na values permanently
df3.fillna(value = df3['profit'].mean(), inplace = True)
```

In [163...]

df3

Out[163...]

	<b>sales</b>	<b>quantity</b>	<b>discount</b>	<b>profit</b>	<b>shipping_cost</b>	<b>year</b>	<b>total_sales</b>	<b>order_date_year</b>
<b>0</b>	408.000000	285.710477	285.710477	106.140000	285.710477	2011	816.000000	2011
<b>1</b>	120.000000	285.710477	285.710477	285.710477	285.710477	2011	360.000000	2011
<b>2</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2011	264.000000	2011
<b>3</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2011	135.000000	2011
<b>4</b>	114.000000	285.710477	285.710477	285.710477	285.710477	2011	570.000000	2011
...	...	...	...	...	...	...	...	...
<b>51285</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2014	285.710477	2014
<b>51286</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2014	285.710477	2014
<b>51287</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2014	285.710477	2014
<b>51288</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2014	285.710477	2014
<b>51289</b>	285.710477	285.710477	285.710477	285.710477	285.710477	2014	285.710477	2014

51290 rows × 9 columns



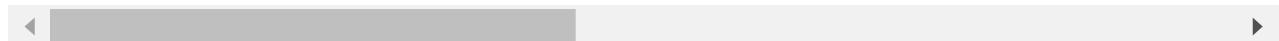
In [164...]

df.head()

Out[164...]

	<b>order_id</b>	<b>order_date</b>	<b>ship_date</b>	<b>ship_mode</b>	<b>customer_name</b>	<b>segment</b>	<b>state</b>	<b>country</b>	<b>market</b>
<b>0</b>	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
<b>1</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
<b>2</b>	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
<b>3</b>	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
<b>4</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 24 columns



In [165...]

```
# group by
df.groupby('order_date_year')['profit'].mean()
```

Out[165...]

order_date_year	profit
2011	27.666238
2012	28.043722
2013	29.604519
2014	28.758540

Name: profit, dtype: float64

In [166...]

```
df.groupby('order_date_year')['discount'].mean()
```

Out[166...]

order_date_year	discount
2011	0.148188
2012	0.141286
2013	0.140265
2014	0.143291

Name: discount, dtype: float64

In [167...]

```
df.groupby('order_date_year')['shipping_cost'].mean()
```

Out[167...]

order_date_year	shipping_cost
2011	27.147264
2012	25.861329
2013	26.418577
2014	26.268197

Name: shipping\_cost, dtype: float64

In [168...]

```
df.groupby('order_date_year')['profit'].median()
```

Out[168...]

order_date_year	profit
2011	9.0300
2012	9.5116
2013	9.2400
2014	9.2000

Name: profit, dtype: float64

In [169...]

```
df[['shipping_cost','profit']].corr()
```

Out[169...]

	shipping_cost	profit
<b>shipping_cost</b>	1.000000	0.357033
<b>profit</b>	0.357033	1.000000

In [170...]

```
df['shipping_cost']
```

Out[170...]

	shipping_cost
0	35.46
1	9.72
2	8.17
3	4.82
4	4.70
...	...
51285	0.89

```
51286    0.49
51287    0.35
51288    0.20
51289    0.17
Name: shipping_cost, Length: 51290, dtype: float64
```

```
In [171... df[['shipping_cost','profit']].describe().T
```

```
Out[171...      count      mean       std      min    25%    50%    75%      max
shipping_cost  51290.0  26.375915  57.296804  0.000  2.61  7.79  24.45  933.570
profit        51290.0  28.641740  174.424113 -6599.978  0.00  9.24  36.81  8399.976
```

```
In [172... df[['shipping_cost','profit']].mean()
```

```
Out[172... shipping_cost    26.375915
profit        28.641740
dtype: float64
```

```
In [173... df[['order_date_year','profit']].mean()
```

```
Out[173... order_date_year    2012.777208
profit        28.641740
dtype: float64
```

```
In [174... df.groupby('order_date_year')['profit'].mean()
```

```
Out[174... order_date_year
2011    27.666238
2012    28.043722
2013    29.604519
2014    28.758540
Name: profit, dtype: float64
```

```
In [175... df.groupby('order_date_year')['sales'].mean()
```

```
Out[175... order_date_year
2011    251.112581
2012    244.252235
2013    246.819335
2014    245.282129
Name: sales, dtype: float64
```

```
In [176... df.groupby('order_date_year')['sales'].sum()
```

```
Out[176... order_date_year
2011    2259511.0
2012    2677493.0
2013    3405860.0
2014    4300041.0
Name: sales, dtype: float64
```

```
In [177... df.sales
```

```
Out[177... 0      408.0
          1      120.0
          2       66.0
          3      45.0
          4     114.0
          ...
          51285   14.0
          51286    4.0
          51287   26.0
          51288    7.0
          51289    3.0
Name: sales, Length: 51290, dtype: float64
```

```
In [178... df.groupby('order_date_year')['quantity'].mean()
```

```
Out[178... order_date_year
2011      3.494443
2012      3.476647
2013      3.488369
2014      3.457989
Name: quantity, dtype: float64
```

```
In [179... df.groupby('order_date_year')['quantity'].sum()
```

```
Out[179... order_date_year
2011      31443
2012      38111
2013      48136
2014      60622
Name: quantity, dtype: int64
```

```
In [180... # Loc is Looking for written index
df.loc[5:6]
```

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	r
5	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oce
6	CA-2011-1510	2011-01-02	1/6/2011	Standard Class	Magdelene Morse	Consumer	Ontario	Canada	Canada	C

2 rows × 24 columns

```
In [181... # iloc is looking for default index
df.iloc[5:6]
```

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	r
5	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oce

1 rows × 24 columns

In [182...]

```
data1 = {'pf' : [235,56,89,8599],
         'tax' :[54,65,89,257],
         'mob' :[3665,6963,3658,9635],
         'course' :['ds','da','big', 'web']}

}
```

In [183...]

```
df01 = pd.DataFrame(data1)
```

In [184...]

```
df01
```

Out[184...]

	pf	tax	mob	course
<b>0</b>	235	54	3665	ds
<b>1</b>	56	65	6963	da
<b>2</b>	89	89	3658	big
<b>3</b>	8599	257	9635	web

In [185...]

```
data23 = {'name' : ['sudh','raha','jab','ron'],
          'salary' :[54,65,89,257],
          'id' :[3665,6963,3658,9635],
          'add' :['kol','blr','blr', 'del']}

}
```

In [186...]

```
df02 = pd.DataFrame(data23)
```

In [187...]

```
df02
```

Out[187...]

	name	salary	id	add
<b>0</b>	sudh	54	3665	kol
<b>1</b>	raha	65	6963	blr
<b>2</b>	jab	89	3658	blr
<b>3</b>	ron	257	9635	del

In [188...]

```
# merge or concat row wise
pd.concat([df01,df02])
```

Out[188...]

	pf	tax	mob	course	name	salary	id	add
0	235.0	54.0	3665.0	ds	NaN	NaN	NaN	NaN
1	56.0	65.0	6963.0	da	NaN	NaN	NaN	NaN
2	89.0	89.0	3658.0	big	NaN	NaN	NaN	NaN
3	8599.0	257.0	9635.0	web	NaN	NaN	NaN	NaN
0	NaN	NaN	NaN	NaN	sudh	54.0	3665.0	kol
1	NaN	NaN	NaN	NaN	raha	65.0	6963.0	blr
2	NaN	NaN	NaN	NaN	jab	89.0	3658.0	blr
3	NaN	NaN	NaN	NaN	ron	257.0	9635.0	del

In [189...]

```
# merge or concat column wise - horizontal
pd.concat([df01,df02], axis = 1)
```

Out[189...]

	pf	tax	mob	course	name	salary	id	add
0	235	54	3665	ds	sudh	54	3665	kol
1	56	65	6963	da	raha	65	6963	blr
2	89	89	3658	big	jab	89	3658	blr
3	8599	257	9635	web	ron	257	9635	del

In [190...]

```
data40 = {'id' :[3665,6963,3658,9635],
          'salary' :[54,65,89,257],
          'add' :['kol','blr','blr', 'del']}
}
```

In [191...]

```
df40 = pd.DataFrame(data40)
```

In [192...]

```
df40
```

Out[192...]

	id	salary	add
0	3665	54	kol
1	6963	65	blr
2	3658	89	blr
3	9635	257	del

In [193...]

```
data41 = {'id' :[3665,69463,3658,9635],
          'mob' :[4534,64545,8439,25457],
          'pf' :[345,568,587,258]}
```

```
In [194... df41 = pd.DataFrame(data41)
```

```
In [195... df41
```

```
Out[195...      id    mob    pf
0   3665  4534  345
1  69463  64545  568
2   3658  8439  587
3   9635  25457  258
```

```
In [196... # merge operation
pd.merge(df41,df40)
```

```
Out[196...      id    mob    pf  salary  add
0   3665  4534  345      54  kol
1   3658  8439  587      89  blr
2   9635  25457  258     257  del
```

```
In [197... pd.merge(df41,df40, how = 'right', on = 'id')
```

```
Out[197...      id    mob    pf  salary  add
0   3665  4534.0  345.0      54  kol
1   6963      NaN      NaN      65  blr
2   3658  8439.0  587.0      89  blr
3   9635  25457.0  258.0     257  del
```

```
In [198... data51 = {'id1' :[3665,69463,3658,9635],
             'mob' :[4534,64545,8439,25457],
             'pf' :[345,568,587,258]}
}
```

```
In [199... df51 = pd.DataFrame(data51)
```

```
In [200... df51
```

Out[200...]

	<b>id1</b>	<b>mob</b>	<b>pf</b>
<b>0</b>	3665	4534	345
<b>1</b>	69463	64545	568
<b>2</b>	3658	8439	587
<b>3</b>	9635	25457	258

In [201...]

```
data48 = {'id' :[3665,6963,3658,9635],
          'salary' :[54,65,89,257],
          'add' :['kol','blr','blr', 'del']}

}
```

In [202...]

```
df48 = pd.DataFrame(data48)
```

In [203...]

```
df48
```

Out[203...]

	<b>id</b>	<b>salary</b>	<b>add</b>
<b>0</b>	3665	54	kol
<b>1</b>	6963	65	blr
<b>2</b>	3658	89	blr
<b>3</b>	9635	257	del

In [204...]

```
# merge operation even there is no common column
pd.merge(df48, df51)
```

```
MergeError
```

```
Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_74932\1100746926.py in <module>
```

```
    1 # merge operation even there is no common column
```

```
--> 2 pd.merge(df48, df51)
```

```
~\anaconda3\lib\site-packages\pandas\core\reshape\merge.py in merge(left, right, how, on, left_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
    104     validate: str | None = None,
    105 ) -> DataFrame:
--> 106     op = _MergeOperation(
    107         left,
    108         right,
```

```
~\anaconda3\lib\site-packages\pandas\core\reshape\merge.py in __init__(self, left, right, how, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, copy, indicator, validate)
    679             warnings.warn(msg, FutureWarning, stacklevel=3)
    680
--> 681             self._validate_specification()
    682
```

```
683      cross_col = None
```

```
~\anaconda3\lib\site-packages\pandas\core\reshape\merge.py in _validate_specification(self)
    1344         common_cols = left_cols.intersection(right_cols)
    1345         if len(common_cols) == 0:
-> 1346             raise MergeError(
    1347                 "No common columns to perform merge on. "
    1348                 f"Merge options: left_on={self.left_on}, "
```

**MergeError:** No common columns to perform merge on. Merge options: left\_on=None, right\_on=None, left\_index=False, right\_index=False

In [ ]:

```
# merge operation even there is no common column
pd.merge(df48, df51, left_on = 'id', right_on = 'id1')
```

In [205...]

df48

Out[205...]

	id	salary	add
0	3665	54	kol
1	6963	65	blr
2	3658	89	blr
3	9635	257	del

In [206...]

df51

Out[206...]

	id1	mob	pf
0	3665	4534	345
1	69463	64545	568
2	3658	8439	587
3	9635	25457	258

In [207...]

```
pd.merge(df41, df40, on ='id')
```

Out[207...]

	id	mob	pf	salary	add
0	3665	4534	345	54	kol
1	3658	8439	587	89	blr
2	9635	25457	258	257	del

In [208...]

```
# join operation is done by index matching
df48.join(df51)
```

Out[208...]

	<b>id</b>	<b>salary</b>	<b>add</b>	<b>id1</b>	<b>mob</b>	<b>pf</b>
<b>0</b>	3665	54	kol	3665	4534	345
<b>1</b>	6963	65	blr	69463	64545	568
<b>2</b>	3658	89	blr	3658	8439	587
<b>3</b>	9635	257	del	9635	25457	258

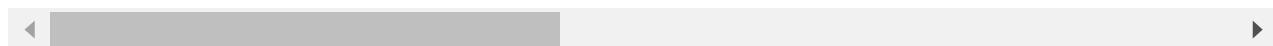
In [209...]

df

Out[209...]

	<b>order_id</b>	<b>order_date</b>	<b>ship_date</b>	<b>ship_mode</b>	<b>customer_name</b>	<b>segment</b>	<b>state</b>	<b>country</b>	<b>r</b>
<b>0</b>	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	
<b>1</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
<b>2</b>	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	
<b>3</b>	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	
<b>4</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	
...	...	...	...	...	...	...	...	...	...
<b>51285</b>	CA-2014-115427	2014-12-31	1/4/2015	Standard Class	Erica Bern	Corporate	California	United States	
<b>51286</b>	MO-2014-2560	2014-12-31	1/5/2015	Standard Class	Liz Preis	Consumer	Souss-Massa-Dra	Morocco	
<b>51287</b>	MX-2014-110527	2014-12-31	1/2/2015	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	
<b>51288</b>	MX-2014-114783	2014-12-31	1/6/2015	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	
<b>51289</b>	CA-2014-156720	2014-12-31	1/4/2015	Standard Class	Jill Matthias	Consumer	Colorado	United States	

51290 rows × 24 columns



In [210...]

```
def profit_flag(a):
    if a >= 0:
```

```

        return 'positive'
else:
    return 'negative'

```

In [211... profit\_flag(58)

Out[211... 'positive'

In [212... profit\_flag(-5)

Out[212... 'negative'

In [213... # application of function

```
df['flag_profit'] = df['profit'].apply(profit_flag)
```

In [214... df.head()

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 25 columns

In [215... df['len\_cust\_name'] = df['customer\_name'].apply(len)

In [216... df.head()

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 26 columns

In [217...]

```
def quantity_flag(a):
    if a < 10:
        return 'low'
    elif a > 10 and a < 20:
        return 'medium'
    else:
        return 'high'
```

In [218...]

```
df['flag_quantity'] = df['quantity'].apply(quantity_flag)
```

In [219...]

```
df.head()
```

Out[219...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 27 columns

```
In [220... df['square_quantity'] = df['quantity'].apply(lambda a : a**2)
```

```
In [221... df.head()
```

Out[221...]

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
<b>0</b>	AG-2011-2040	2011-01-01	1/6/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
<b>1</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
<b>2</b>	HU-2011-1220	2011-01-01	1/5/2011	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
<b>3</b>	IT-2011-3647632	2011-01-01	1/5/2011	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
<b>4</b>	IN-2011-47883	2011-01-01	1/8/2011	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 28 columns

```
In [222... import numpy as np
```

```
In [223... np.array([2,3,5,6,8])
```

Out[223... array([2, 3, 5, 6, 8])

```
In [224... type(np.array([2,3,5,6,8]))
```

Out[224... numpy.ndarray

```
In [225... np.array([[1,2,3],[56,85,95]])
```

Out[225... array([[ 1, 2, 3],  
[56, 85, 95]])

```
In [226... a = np.array([5,3,5,6], ndmin = 5)
```

```
In [227... a
```

Out[227... array([[[[5, 3, 5, 6]]]])

```
In [228... arr = np.array ([[1,2,3],[4,5,6],[7,8,9]])
```

```
In [229... arr
```

```
Out[229... array([[1, 2, 3],  
[4, 5, 6],  
[7, 8, 9]])
```

```
In [230... arr.ndim
```

```
Out[230... 2
```

```
In [231... np.random.randint(1,10,(4,4))
```

```
Out[231... array([[8, 3, 9, 2],  
[1, 7, 3, 3],  
[2, 2, 2, 3],  
[6, 1, 1, 1]])
```

```
In [232... np.random.exponential(scale = 10)
```

```
Out[232... 19.375680532928946
```

```
In [233... # random integer  
np.random.randint(1,10,(4,4,2))
```

```
Out[233... array([[[9, 2],  
[6, 9],  
[5, 1],  
[4, 3]],  
  
[[8, 4],  
[8, 3],  
[4, 5],  
[7, 6]],  
  
[[2, 8],  
[3, 6],  
[6, 6],  
[5, 4]],  
  
[[6, 4],  
[9, 1],  
[7, 9],  
[9, 4]]])
```

```
In [234... pwd
```

```
Out[234... 'C:\\\\Users\\\\rupto'
```

```
In [235... l = [1,2,3,4,5,6,7,8,9]
```

```
In [236... a = np.array(1)
```

```
In [237... a
```

```
Out[237... array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [238... type(a)
```

```
Out[238... numpy.ndarray
```

```
In [239... l1 = [2,3,5,6, True, 'sgaf',35.25]
```

```
In [240... a1 = np.array(l1)
```

```
In [241... # 2 dimension  
a2 = np.array([[1,2,3], [4,5,6]])
```

```
In [242... # 3 dimension  
a3 = np.array([[[1,2,3], [4,5,6], [7,8,9]]])
```

```
In [243... a1
```

```
Out[243... array(['2', '3', '5', '6', 'True', 'sgaf', '35.25'], dtype='<U32')
```

```
In [244... a2
```

```
Out[244... array([[1, 2, 3],  
 [4, 5, 6]])
```

```
In [245... a3
```

```
Out[245... array([[[1, 2, 3],  
 [4, 5, 6],  
 [7, 8, 9]]])
```

```
In [246... #dimension  
a1.ndim
```

```
Out[246... 1
```

```
In [247... a2.ndim
```

```
Out[247... 2
```

```
In [248... a3.ndim
```

```
Out[248... 3
```

```
In [249... x = np.random.randint(1,10,(2,3))
```

```
In [250... x.ndim
```

```
Out[250... 2
```

```
In [251... x
```

```
Out[251... array([[1, 4, 8],  
[2, 2, 1]])
```

```
In [252... a1.size
```

```
Out[252... 7
```

```
In [253... a2.size
```

```
Out[253... 6
```

```
In [254... a3.size
```

```
Out[254... 9
```

```
In [255... a1.shape
```

```
Out[255... (7,)
```

```
In [256... a3.shape
```

```
Out[256... (1, 3, 3)
```

```
In [257... a2.shape
```

```
Out[257... (2, 3)
```

```
In [258... a2
```

```
Out[258... array([[1, 2, 3],  
[4, 5, 6]])
```

```
In [259... a3
```

```
Out[259... array([[1, 2, 3],  
   [4, 5, 6],  
   [7, 8, 9]]])
```

```
In [260... np.random.randint(2,50)
```

```
Out[260... 13
```

```
In [261... np.random.randint(2,50,(2,3,4))
```

```
Out[261... array([[[36, 37, 21, 2],  
   [26, 26, 21, 35],  
   [46, 49, 28, 8]],  
  
  [[42, 5, 33, 2],  
   [41, 32, 18, 37],  
   [16, 10, 43, 28]]])
```

```
In [262... np.random.randint(2,50,(3,3,4))
```

```
Out[262... array([[[ 6, 5, 31, 14],  
   [41, 24, 3, 19],  
   [25, 46, 39, 35]],  
  
  [[15, 20, 39, 40],  
   [40, 39, 37, 38],  
   [ 5, 3, 47, 24]],  
  
  [[21, 46, 26, 2],  
   [ 2, 2, 37, 26],  
   [ 2, 21, 15, 11]]])
```

```
In [263... np.random.rand(5,4)
```

```
Out[263... array([[ 0.02648188,  0.53207958,  0.9164909 ,  0.41715934],  
   [ 0.39090151,  0.05687278,  0.13845919,  0.07236484],  
   [ 0.95163107,  0.26393736,  0.52852584,  0.70148755],  
   [ 0.703076 ,  0.06388578,  0.1585603 ,  0.09517653],  
   [ 0.63227364,  0.70555558,  0.28124545,  0.61692908]])
```

```
In [264... a4 = np.random.randn(5,4)  
a4
```

```
Out[264... array([[ 0.12694963, -0.12813859, -0.63242722,  0.53873989],  
   [ 0.41204806, -1.08078878, -0.06024934,  0.97512916],  
   [ 0.70996501, -1.97242266,  0.34776041, -0.46519898],  
   [ 1.12990331,  0.76334065,  0.29931082,  0.64125664],  
   [-1.24288446,  0.21006209, -0.46382568,  1.7452034 ]])
```

```
In [265... a4.reshape(10,2)
```

```
Out[265... array([[ 0.12694963, -0.12813859],  
   [-0.63242722,  0.53873989],  
   [ 0.41204806, -1.08078878],
```

```
[[-0.06024934,  0.97512916],
 [ 0.70996501, -1.97242266],
 [ 0.34776041, -0.46519898],
 [ 1.12990331,  0.76334065],
 [ 0.29931082,  0.64125664],
 [-1.24288446,  0.21006209],
 [-0.46382568,  1.7452034 ]])
```

In [266...]

a4.reshape(20,1)

Out[266...]

```
array([[ 0.12694963],
       [-0.12813859],
       [-0.63242722],
       [ 0.53873989],
       [ 0.41204806],
       [-1.08078878],
       [-0.06024934],
       [ 0.97512916],
       [ 0.70996501],
       [-1.97242266],
       [ 0.34776041],
       [-0.46519898],
       [ 1.12990331],
       [ 0.76334065],
       [ 0.29931082],
       [ 0.64125664],
       [-1.24288446],
       [ 0.21006209],
       [-0.46382568],
       [ 1.7452034 ]])
```

In [267...]

a4.reshape(2,-1)

Out[267...]

```
array([[ 0.12694963, -0.12813859, -0.63242722,  0.53873989,  0.41204806,
       -1.08078878, -0.06024934,   0.97512916,  0.70996501, -1.97242266],
       [ 0.34776041, -0.46519898,   1.12990331,  0.76334065,  0.29931082,
       0.64125664, -1.24288446,   0.21006209, -0.46382568,  1.7452034 ]])
```

In [268...]

a4.reshape(2,5,2)

Out[268...]

```
array([[[ 0.12694963, -0.12813859],
       [-0.63242722,  0.53873989],
       [ 0.41204806, -1.08078878],
       [-0.06024934,  0.97512916],
       [ 0.70996501, -1.97242266]],

      [[ 0.34776041, -0.46519898],
       [ 1.12990331,  0.76334065],
       [ 0.29931082,  0.64125664],
       [-1.24288446,  0.21006209],
       [-0.46382568,  1.7452034 ]]])
```

In [269...]

a4.reshape(2,2,5)

Out[269...]

```
array([[[ 0.12694963, -0.12813859, -0.63242722,  0.53873989,
       0.41204806],
       [-1.08078878, -0.06024934,   0.97512916,  0.70996501,
```

```
-1.97242266]],  
[[ 0.34776041, -0.46519898,  1.12990331,  0.76334065,  
 0.29931082],  
[ 0.64125664, -1.24288446,  0.21006209, -0.46382568,  
1.7452034 ]]])
```

In [270...]

a2

Out[270...]

```
array([[1, 2, 3],  
[4, 5, 6]])
```

In [271...]

a2[:,1:]

Out[271...]

```
array([[2, 3],  
[5, 6]])
```

In [272...]

a2[1:,1:]

Out[272...]

```
array([[5, 6]])
```

In [273...]

a2[:,[1,2]]

Out[273...]

```
array([[2, 3],  
[5, 6]])
```

In [274...]

a2[[0,1]]

Out[274...]

```
array([[1, 2, 3],  
[4, 5, 6]])
```

In [275...]

a2

Out[275...]

```
array([[1, 2, 3],  
[4, 5, 6]])
```

In [276...]

a2[[0,1],1:]

Out[276...]

```
array([[2, 3],  
[5, 6]])
```

In [277...]

a5 = np.random.randint(2,90,(6,5))

In [278...]

a5

Out[278...]

```
array([[12, 52, 59, 66, 17],  
[40, 10, 74, 11, 74],  
[83, 9, 2, 42, 40],  
[30, 45, 24, 69, 9],  
[62, 69, 71, 15, 83],  
[65, 41, 87, 9, 44]])
```

```
In [279... # filter array in numpy  
a5[a5>50]
```

```
Out[279... array([52, 59, 66, 74, 74, 83, 69, 62, 69, 71, 83, 65, 87])
```

```
In [280... # change value in numpy  
a5[0,4]
```

```
Out[280... 17
```

```
In [281... a5[0,4] = 1111
```

```
In [ ]:
```

```
In [282... a6 = np.random.randint(0,3,(3,3))
```

```
In [283... a7 = np.random.randint(0,3,(3,3))
```

```
In [284... a6
```

```
Out[284... array([[1, 1, 1],  
[1, 1, 0],  
[2, 0, 2]])
```

```
In [285... a7
```

```
Out[285... array([[0, 0, 2],  
[0, 2, 2],  
[2, 2, 0]])
```

```
In [286... a6*a7
```

```
Out[286... array([[0, 0, 2],  
[0, 2, 0],  
[4, 0, 0]])
```

```
In [287... a6@a7
```

```
Out[287... array([[2, 4, 4],  
[0, 2, 4],  
[4, 4, 4]])
```

```
In [288... b = np.array([[2,3],[2,3]])
```

```
In [289... b
```

```
Out[289... array([[2, 3],  
[2, 3]])
```

```
In [290... b1 = np.array([[3,2],[3,2]])
```

```
In [291... b1
```

```
Out[291... array([[3, 2],  
[3, 2]])
```

```
In [292... # element wise multiplication  
b*b1
```

```
Out[292... array([[6, 6],  
[6, 6]])
```

```
In [293... # matrix multiplication  
b@b1
```

```
Out[293... array([[15, 10],  
[15, 10]])
```

```
In [294... a6+100
```

```
Out[294... array([[101, 101, 101],  
[101, 101, 100],  
[102, 100, 102]])
```

```
In [295... a6*2
```

```
Out[295... array([[2, 2, 2],  
[2, 2, 0],  
[4, 0, 4]])
```

```
In [296... a6/0
```

```
C:\Users\rupto\AppData\Local\Temp\ipykernel_74932/1058679823.py:1: RuntimeWarning: divide by zero encountered in true_divide  
a6/0
```

```
C:\Users\rupto\AppData\Local\Temp\ipykernel_74932/1058679823.py:1: RuntimeWarning: invalid value encountered in true_divide  
a6/0
```

```
Out[296... array([[inf, inf, inf],  
[inf, inf, nan],  
[inf, nan, inf]])
```

```
In [297... a8 = np.zeros((4,4))
```

```
In [298... a8
```

```
Out[298... array([[0., 0., 0., 0.],  
[0., 0., 0., 0.],  
[0., 0., 0., 0.],  
[0., 0., 0., 0.]])
```

```
[0., 0., 0., 0.],  
[0., 0., 0., 0.]])
```

In [299...]

```
a9 = np.ones((4,4))
```

In [300...]

```
a9
```

Out[300...]

```
array([[1., 1., 1., 1.],  
       [1., 1., 1., 1.],  
       [1., 1., 1., 1.],  
       [1., 1., 1., 1.]])
```

In [301...]

```
a9+10
```

Out[301...]

```
array([[11., 11., 11., 11.],  
       [11., 11., 11., 11.],  
       [11., 11., 11., 11.],  
       [11., 11., 11., 11.]])
```

In [302...]

```
a9 + np.array([1,2,3,4])
```

Out[302...]

```
array([[2., 3., 4., 5.],  
       [2., 3., 4., 5.],  
       [2., 3., 4., 5.],  
       [2., 3., 4., 5.]])
```

In [303...]

```
# Transpose  
np.array([[1,2,3,4]]).T + a9
```

Out[303...]

```
array([[2., 2., 2., 2.],  
       [3., 3., 3., 3.],  
       [4., 4., 4., 4.],  
       [5., 5., 5., 5.]])
```

In [304...]

```
a9
```

Out[304...]

```
array([[1., 1., 1., 1.],  
       [1., 1., 1., 1.],  
       [1., 1., 1., 1.],  
       [1., 1., 1., 1.]])
```

In [305...]

```
a6
```

Out[305...]

```
array([[1, 1, 1],  
       [1, 1, 0],  
       [2, 0, 2]])
```

In [306...]

```
a6.T
```

Out[306...]

```
array([[1, 1, 2],  
       [1, 1, 0],  
       [1, 0, 2]])
```

```
In [307... np.array([1,2,3,4]).T
```

```
Out[307... array([1, 2, 3, 4])
```

```
In [308... np.array([[1,2,3,4]]).T
```

```
Out[308... array([[1],  
[2],  
[3],  
[4]])
```

```
In [309... np.sqrt(a6)
```

```
Out[309... array([[1.          , 1.          , 1.          ],  
[1.          , 1.          , 0.          ],  
[1.41421356, 0.          , 1.41421356]])
```

```
In [310... a6
```

```
Out[310... array([[1, 1, 1],  
[1, 1, 0],  
[2, 0, 2]])
```

```
In [311... np.exp(a6)
```

```
Out[311... array([[2.71828183, 2.71828183, 2.71828183],  
[2.71828183, 2.71828183, 1.          ],  
[7.3890561 , 1.          , 7.3890561 ]])
```

```
In [312... np.log1p(a6)
```

```
Out[312... array([[0.69314718, 0.69314718, 0.69314718],  
[0.69314718, 0.69314718, 0.          ],  
[1.09861229, 0.          , 1.09861229]])
```

```
In [313... np.log10(a6)
```

```
C:\Users\ruperto\AppData\Local\Temp\ipykernel_74932/3143830726.py:1: RuntimeWarning: divide by zero encountered in log10  
    np.log10(a6)  
Out[313... array([[0.          , 0.          , 0.          ],  
[0.          , 0.          , -inf],  
[0.30103,      -inf, 0.30103]])
```

```
In [314... # range  
np.arange(20)
```

```
Out[314... array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,  
17, 18, 19])
```

```
In [315... np.arange(1,20,3)
```

```
Out[315... array([ 1,  4,  7, 10, 13, 16, 19])
```

```
In [316... np.arange(1,15.9,3)
```

```
Out[316... array([ 1.,  4.,  7., 10., 13.])
```

```
In [317... np.arange(1.8,15.9,3.5)
```

```
Out[317... array([ 1.8,  5.3,  8.8, 12.3, 15.8])
```

```
In [318... # generate 100 numbers between 2 and 3
np.linspace(2,3,num=100)
```

```
Out[318... array([2.          , 2.01010101, 2.02020202, 2.03030303, 2.04040404,
       2.05050505, 2.06060606, 2.07070707, 2.08080808, 2.09090909,
       2.1010101 , 2.11111111, 2.12121212, 2.13131313, 2.14141414,
       2.15151515, 2.16161616, 2.17171717, 2.18181818, 2.19191919,
       2.2020202 , 2.21212121, 2.22222222, 2.23232323, 2.24242424,
       2.25252525, 2.26262626, 2.27272727, 2.28282828, 2.29292929,
       2.3030303 , 2.31313131, 2.32323232, 2.33333333, 2.34343434,
       2.35353535, 2.36363636, 2.37373737, 2.38383838, 2.39393939,
       2.4040404 , 2.41414141, 2.42424242, 2.43434343, 2.44444444,
       2.45454545, 2.46464646, 2.47474747, 2.48484848, 2.49494949,
       2.50505051, 2.51515152, 2.52525253, 2.53535354, 2.54545455,
       2.55555556, 2.56565657, 2.57575758, 2.58585859, 2.5959596 ,
       2.60606061, 2.61616162, 2.62626263, 2.63636364, 2.64646465,
       2.65656566, 2.66666667, 2.67676768, 2.68686869, 2.6969697 ,
       2.70707071, 2.71717172, 2.72727273, 2.73737374, 2.74747475,
       2.75757576, 2.76767677, 2.77777778, 2.78787879, 2.7979798 ,
       2.80808081, 2.81818182, 2.82828283, 2.83838384, 2.84848485,
       2.85858586, 2.86868687, 2.87878788, 2.88888889, 2.8989899 ,
       2.90909091, 2.91919192, 2.92929293, 2.93939394, 2.94949495,
       2.95959596, 2.96969697, 2.97979798, 2.98989899, 3.        ])
```

```
In [319... np.linspace(2,3,retstep = True)
```

```
Out[319... (array([2.          , 2.02040816, 2.04081633, 2.06122449, 2.08163265,
       2.10204082, 2.12244898, 2.14285714, 2.16326531, 2.18367347,
       2.20408163, 2.22444898 , 2.24489796, 2.26530612, 2.28571429,
       2.30612245, 2.32653061, 2.34693878, 2.36734694, 2.3877551 ,
       2.40816327, 2.42857143, 2.44897959, 2.46938776, 2.48979592,
       2.51020408, 2.53061224, 2.55102041, 2.57142857, 2.59183673,
       2.6122449 , 2.63265306, 2.65306122, 2.67346939, 2.69387755,
       2.71428571, 2.73469388, 2.75510204, 2.7755102 , 2.79591837,
       2.81632653, 2.83673469, 2.85714286, 2.87755102, 2.89795918,
       2.91836735, 2.93877551, 2.95918367, 2.97959184, 3.        ]),
       0.02040816326530612)
```

```
In [320... np.logspace(2,4, num = 4, base = 10)
```

```
Out[320... array([ 100.          , 464.15888336, 2154.43469003, 10000.        ])
```

```
In [321... np.logspace(2,4)
```

```
Out[321... array([ 100.          , 109.8541142 , 120.67926406, 132.57113656,
   145.63484775, 159.98587196, 175.75106249, 193.06977289,
   212.09508879, 232.99518105, 255.95479227, 281.1768698 ,
   308.88435965, 339.32217719, 372.75937203, 409.49150624,
   449.8432669 , 494.17133613, 542.86754393, 596.36233166,
   655.12855686, 719.685673 , 790.60432109, 868.51137375,
   954.09547635, 1048.11313415, 1151.39539933, 1264.85521686,
  1389.49549437, 1526.41796718, 1676.83293681, 1842.06996933,
  2023.58964773, 2222.99648253, 2442.05309455, 2682.69579528,
  2947.05170255, 3237.45754282, 3556.48030622, 3906.93993705,
  4291.93426013, 4714.86636346, 5179.47467923, 5689.86602902,
  6250.55192527, 6866.48845004, 7543.12006335, 8286.42772855,
  9102.98177992, 10000.          ])
```

```
In [322... np.eye(5)
```

```
Out[322... array([[1., 0., 0., 0., 0.],
   [0., 1., 0., 0., 0.],
   [0., 0., 1., 0., 0.],
   [0., 0., 0., 1., 0.],
   [0., 0., 0., 0., 1.]])
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```