

DATA ANALYSIS

In [151...]

```
import os  
os.chdir('D:\\NXTWAVE\\Dataset')  
os.getcwd()
```

Out[151...]

```
'D:\\NXTWAVE\\Dataset'
```

In [152...]

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns
```

Data set1 - leads_basic_details

In [153...]

```
d1 = pd.read_csv('leads_basic_details.csv')
```

In [154...]

```
d1.head()
```

Out[154...]

	lead_id	age	gender	current_city	current_education	parent_occupation	lead_gen_source
0	USR1001	16	FEMALE	Hyderabad	Intermediate	Private Employee	social_media
1	USR1002	20	MALE	Bengaluru	B.Tech	Business	user_referrals
2	USR1003	20	FEMALE	Visakhapatnam	B.Tech	Lawyer	user_referrals
3	USR1004	16	MALE	Mumbai	Intermediate	IT Employee	user_referrals
4	USR1005	16	MALE	Chennai	Intermediate	Government Employee	user_referrals

In [155...]

```
d1.shape
```

Out[155...]

```
(360, 7)
```

```
In [156...]: d1.dtypes
```

```
Out[156...]:
```

lead_id	object
age	int64
gender	object
current_city	object
current_education	object
parent_occupation	object
lead_gen_source	object
dtype:	object

```
In [157...]: d1.describe()
```

```
Out[157...]:
```

	age
count	360.000000
mean	21.561111
std	11.555444
min	16.000000
25%	18.000000
50%	21.000000
75%	24.000000
max	211.000000

```
In [158...]: d1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 360 entries, 0 to 359
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   lead_id          360 non-null    object 
 1   age              360 non-null    int64  
 2   gender           360 non-null    object 
 3   current_city     360 non-null    object 
 4   current_education 360 non-null    object 
 5   parent_occupation 360 non-null    object 
 6   lead_gen_source   360 non-null    object
```

```
dtypes: int64(1), object(6)
memory usage: 19.8+ KB
```

```
In [159... d1.isna().any()
```

```
Out[159... lead_id      False
          age         False
          gender       False
          current_city False
          current_education False
          parent_occupation False
          lead_gen_source  False
          dtype: bool
```

```
In [160... d1.isna().sum()
```

```
Out[160... lead_id      0
          age         0
          gender       0
          current_city 0
          current_education 0
          parent_occupation 0
          lead_gen_source  0
          dtype: int64
```

```
In [161... d1.duplicated().sum()
```

```
Out[161... 0
```

```
In [162... d1.columns
```

```
Out[162... Index(['lead_id', 'age', 'gender', 'current_city', 'current_education',
                  'parent_occupation', 'lead_gen_source'],
                  dtype='object')
```

```
In [163... d1['age'].unique()
```

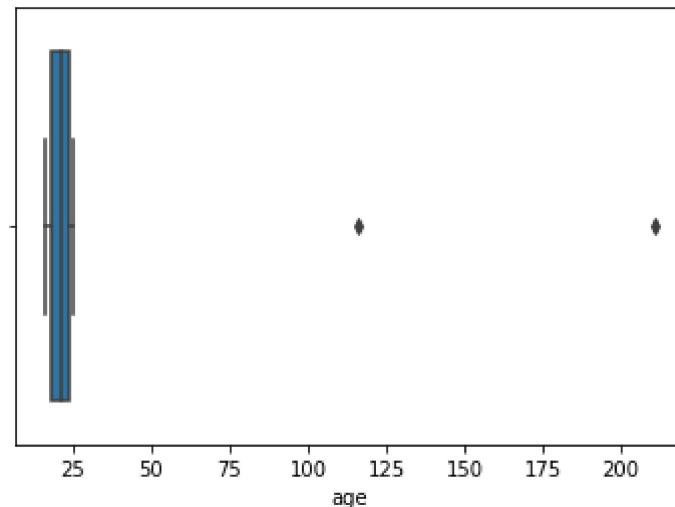
```
Out[163... array([ 16,  20,  21,  22,  25,  18,  24, 211, 116], dtype=int64)
```

Identifying the outlier

```
In [164... sns.boxplot(d1['age']);
```

C:\Users\SURESH\Pyton\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
    warnings.warn(
```



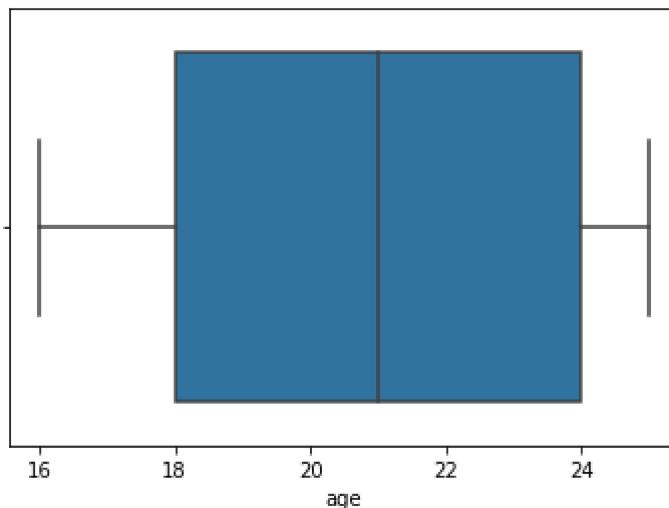
Removed the outlier

```
In [165... filter = d1['age'].values < 26  
d1_filter = d1[filter]  
sns.boxplot(d1_filter['age'])
```

C:\Users\SURESH\Pyton\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
    warnings.warn(
```

```
Out[165... <AxesSubplot:xlabel='age'>
```



```
In [166...]: d1_filter.shape
```

```
Out[166...]: (358, 7)
```

```
In [167...]: d1_filter.gender.value_counts()
```

```
Out[167...]: FEMALE    198  
MALE      160  
Name: gender, dtype: int64
```

```
In [168...]: d1_filter.current_city.value_counts()
```

```
Out[168...]: Visakhapatnam    76  
Hyderabad        75  
Kochi            67  
Bengaluru        51  
Mumbai           48  
Chennai          41  
Name: current_city, dtype: int64
```

```
In [169...]: d1_filter.current_education.value_counts()
```

```
Out[169...]: B.Tech            139  
Looking for Job       117
```

```
Degree           42
Intermediate    40
Intermediate Completed 12
10th Completed   8
Name: current_education, dtype: int64
```

In [170...]: `d1_filter.parent_occupation.value_counts()`

```
Out[170...]: Government Employee    128
Business          80
IT Employee       70
Lawyer            44
Private Employee  14
Doctor             11
Professor/Faculty 11
Name: parent_occupation, dtype: int64
```

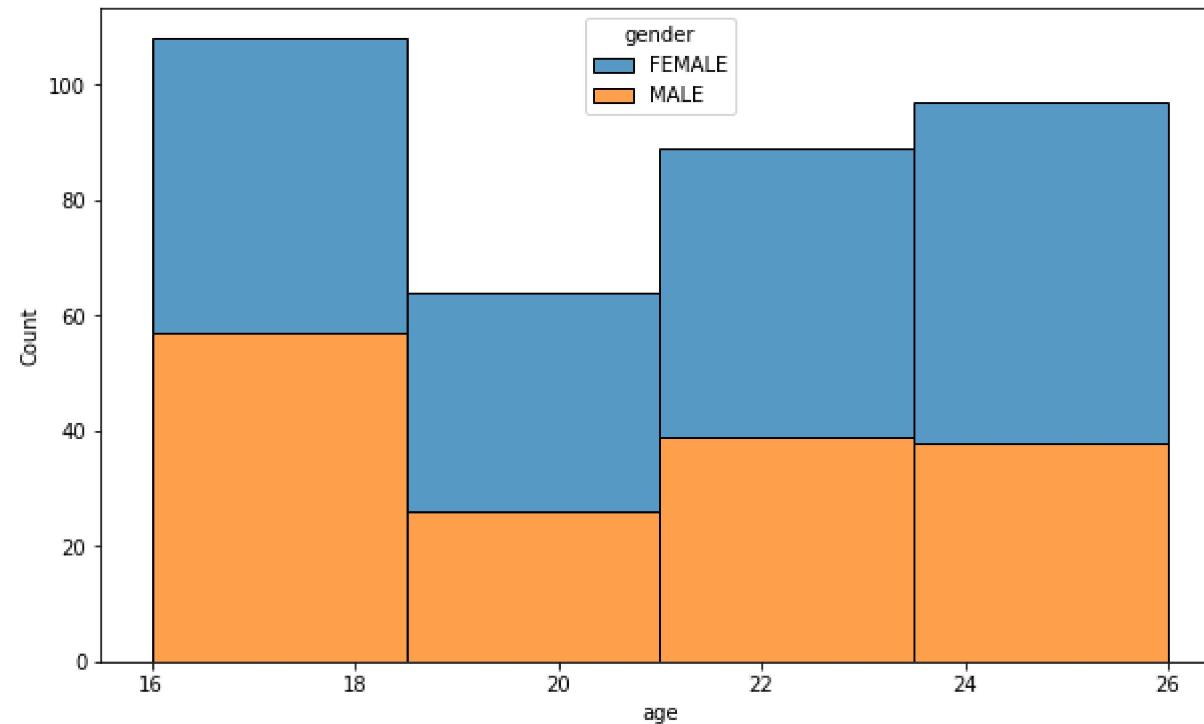
In [171...]: `d1_filter.lead_gen_source.value_counts()`

```
Out[171...]: social_media      85
SEO                 75
email_marketing    73
user_referrals     66
website            59
Name: lead_gen_source, dtype: int64
```

Visualization

In [172...]: `plt.figure(figsize=(10,6))
sns.histplot(d1_filter ,x='age',bins=4,hue='gender',multiple="stack", binrange=(16,26))`

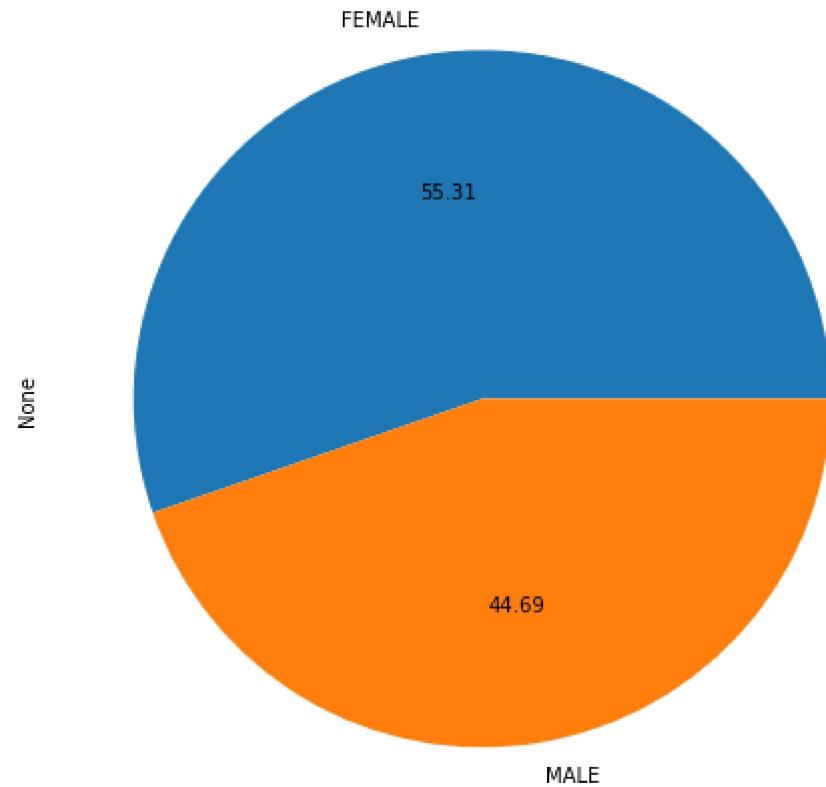
Out[172...]: <AxesSubplot:xlabel='age', ylabel='Count'>



In [173...]

```
plt.figure(figsize=(15,8))
d1_filter.groupby('gender').size().plot(kind='pie', autopct='%.2f')
```

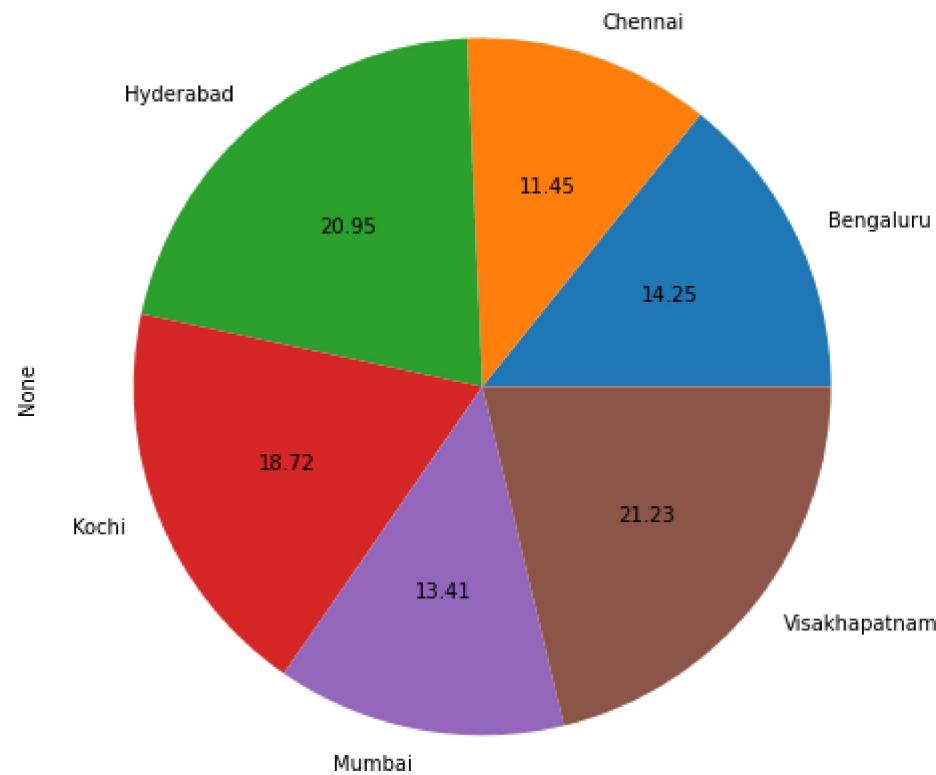
Out[173... <AxesSubplot:ylabel='None'>



In [174...]

```
plt.figure(figsize=(15,8))
d1_filter.groupby('current_city').size().plot(kind='pie', autopct='%.2f')
```

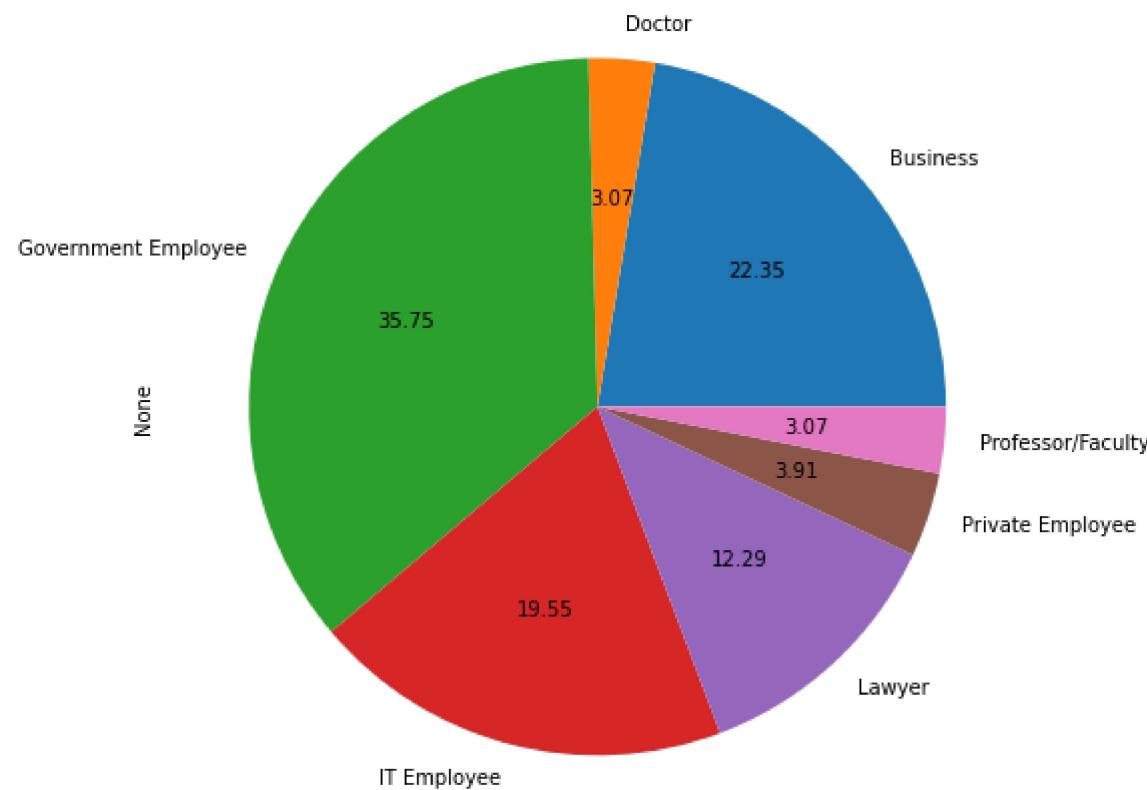
Out[174... <AxesSubplot:ylabel='None'>



In [175...]

```
plt.figure(figsize=(15,8))
d1_filter.groupby('parent_occupation').size().plot(kind='pie', autopct='%.2f')
```

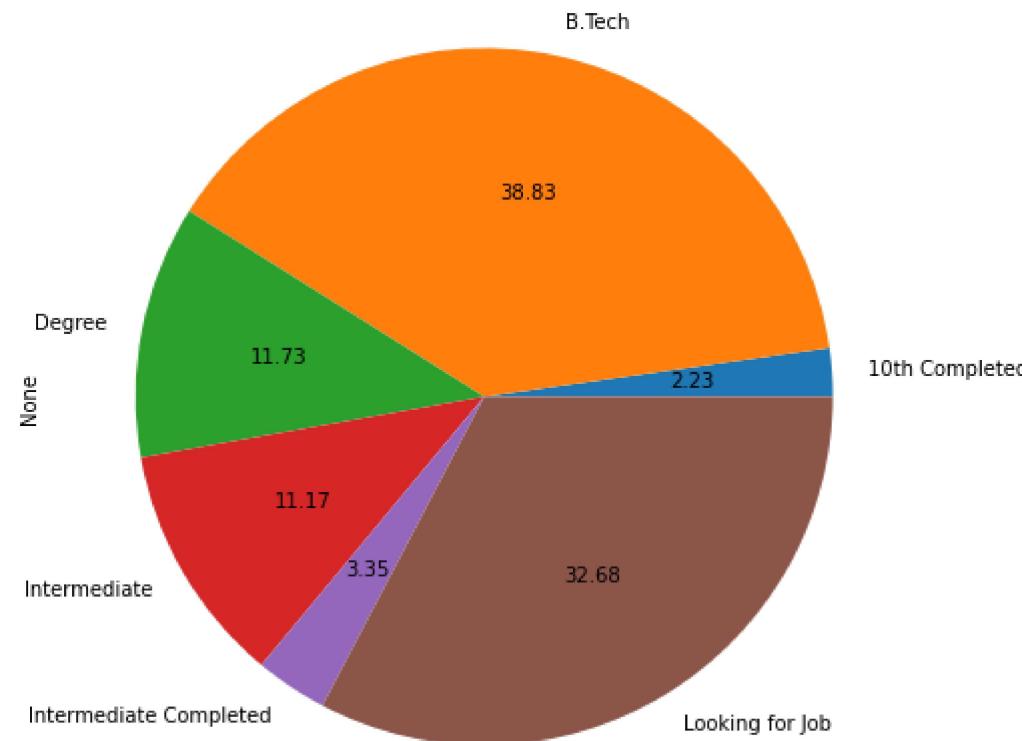
Out[175... <AxesSubplot:ylabel='None'>



In [176...]

```
plt.figure(figsize=(15,8))
d1_filter.groupby('current_education').size().plot(kind='pie', autopct='%.2f')
```

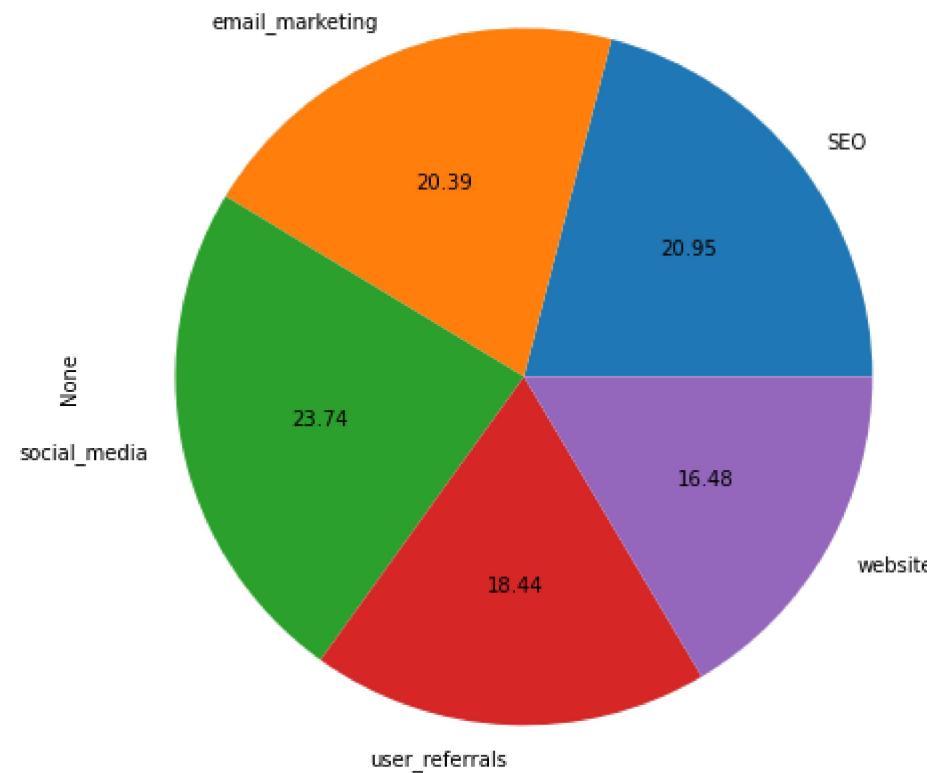
Out[176... <AxesSubplot:ylabel='None'>



In [177...]

```
plt.figure(figsize=(15,8))
d1_filter.groupby('lead_gen_source').size().plot(kind='pie', autopct='%.2f')
```

Out[177... <AxesSubplot:ylabel='None'>

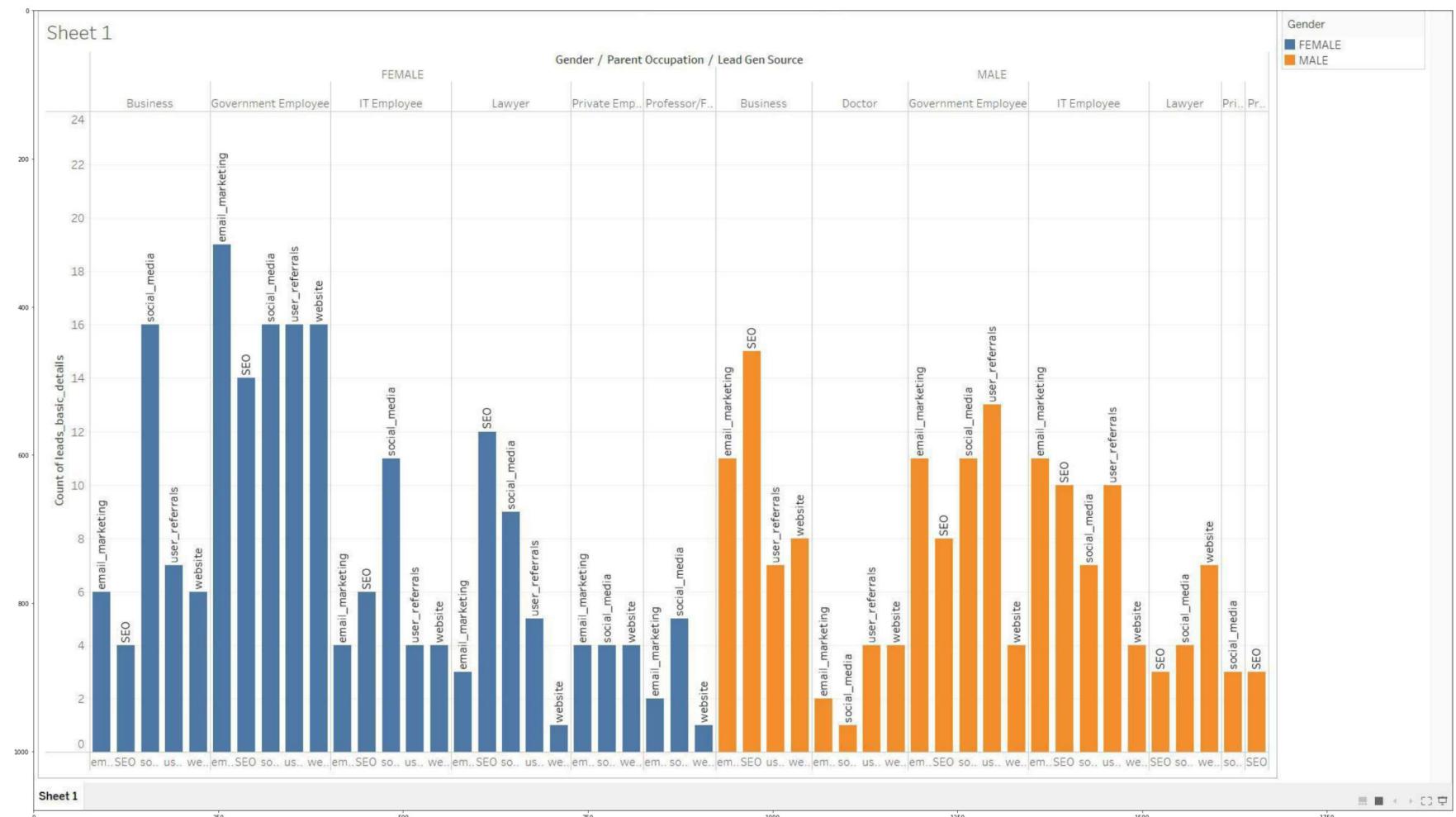


Visualization using Tableau

In [178...]

```
import matplotlib.image as mpimg  
  
img = mpimg.imread('leads_basic_details.jpg')  
plt.figure(figsize=(50,25))  
plt.imshow(img)
```

Out[178...]



Data set2 - leads_demo_watched_details

In [179...]

```
d2 = pd.read_csv('leads_demo_watched_details.csv')
```

In [180...]

```
d2.head()
```

Out[180...]

	lead_id	demo_watched_date	language	watched_percentage
--	---------	-------------------	----------	--------------------

0 USR1002

1/4/2022

Telugu

42

	lead_id	demo_watched_date	language	watched_percentage
1	USR1003	1/7/2022	Telugu	81
2	USR1004	1/2/2022	Telugu	35
3	USR1005	1/3/2022	Hindi	38
4	USR1006	1/12/2022	Hindi	54

In [181...]

d2.shape

Out[181...](194, 4)

In [182...]

d2.dtypes

Out[182...]

	lead_id	object
	demo_watched_date	object
	language	object
	watched_percentage	int64
	dtype:	object

In [183...]

d2.describe()

Out[183...]

	watched_percentage
count	194.000000
mean	56.634021
std	43.555635
min	2.000000
25%	35.000000
50%	55.500000
75%	75.750000
max	510.000000

In [184...]

```
d2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194 entries, 0 to 193
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   lead_id          194 non-null    object  
 1   demo_watched_date 194 non-null    object  
 2   language          194 non-null    object  
 3   watched_percentage 194 non-null    int64  
dtypes: int64(1), object(3)
memory usage: 6.2+ KB
```

In [185...]

```
d2.isna().sum()
```

Out[185...]

```
lead_id          0
demo_watched_date 0
language          0
watched_percentage 0
dtype: int64
```

In [186...]

```
d2.duplicated().sum()
```

Out[186...]

```
0
```

In [187...]

```
d2.columns
```

Out[187...]

```
Index(['lead_id', 'demo_watched_date', 'language', 'watched_percentage'], dtype='object')
```

In [188...]

```
d2['watched_percentage'].unique()
```

Out[188...]

```
array([ 42,  81,  35,  38,  54,  52,  41,  50,  48,  44,  46,  51,  47,
       40,  36, 100,   5,  53,   8,  10,  11,  14,  17,  20,  23,  55,
       73,  26,  29,  32,  56,  59,  62,  60,  65,  70,  74,  75,  68,
       85,  71,  86,  87,  88,  89,   2,  90,  91,  92,  77,  80,  63,
       83,  25,   3,   4,  30,  45,  64, 510,  37,  66,  67,  49,  57,
       69,  61,  95,  15,  12,  13,  72,  19, 233,  27,  21,  16,  76,
      18,  79,  78,  22,  24,  28,  82,  84], dtype=int64)
```

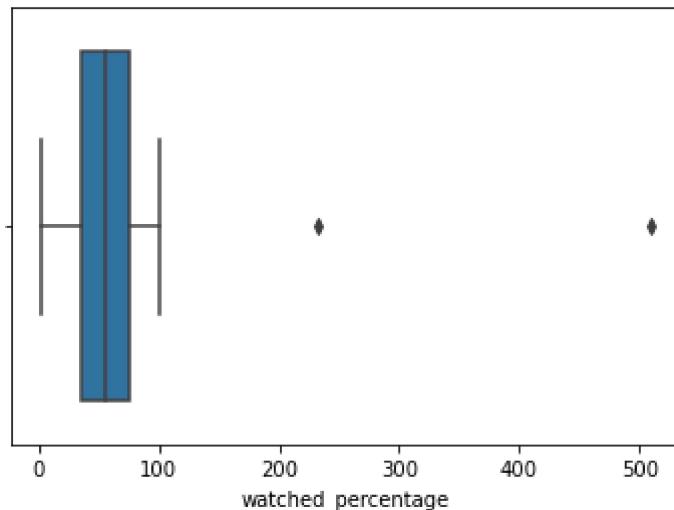
In [189...]

```
sns.boxplot(d2['watched_percentage'])
```

```
C:\Users\SURESH\Python\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword argument: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
```

```
    warnings.warn(
```

```
Out[189... <AxesSubplot:xlabel='watched_percentage'>
```



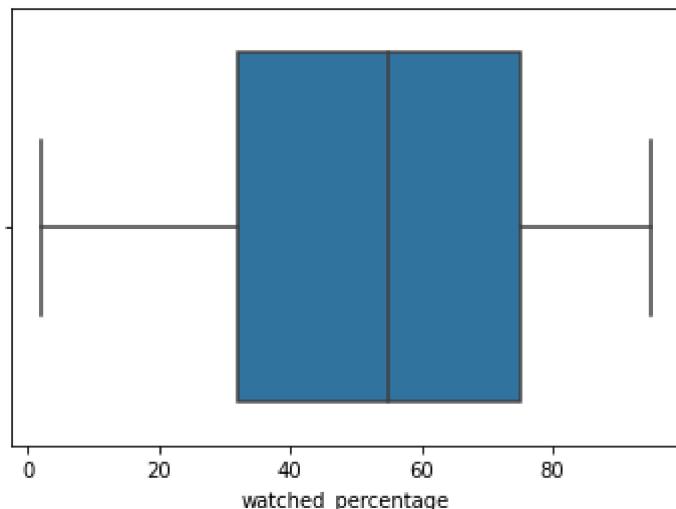
```
In [190... 
```

```
filter = d2['watched_percentage'].values<100
d2_filter = d2[filter]
sns.boxplot(d2_filter['watched_percentage'])
```

```
C:\Users\SURESH\Python\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword argument: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
```

```
    warnings.warn(
```

```
Out[190... <AxesSubplot:xlabel='watched_percentage'>
```



```
In [191... d2_filter.shape
```

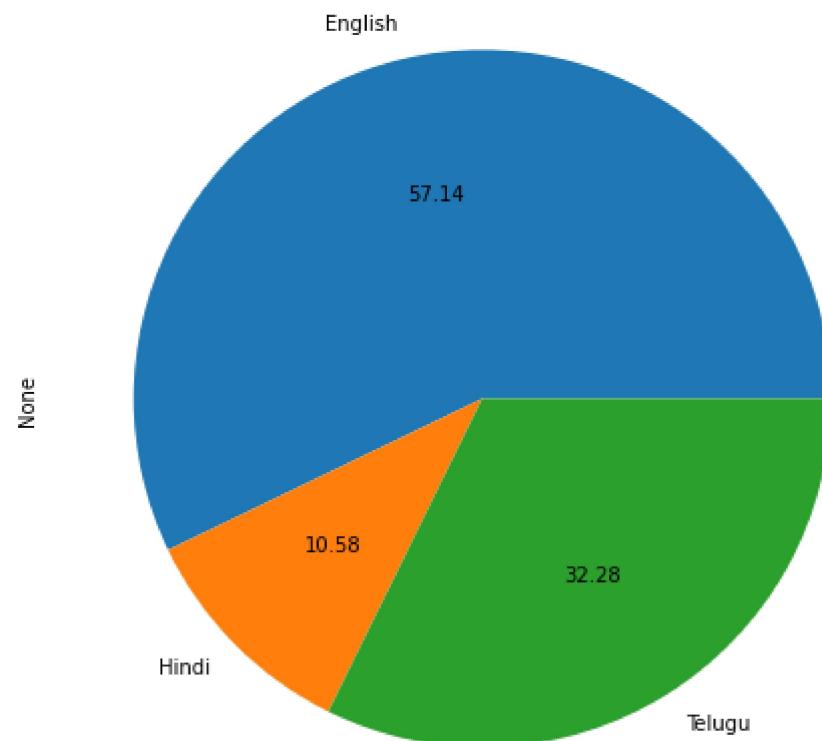
```
Out[191... (189, 4)
```

```
In [192... d2_filter.language.value_counts()
```

```
Out[192... English    108  
Telugu      61  
Hindi       20  
Name: language, dtype: int64
```

```
In [193... plt.figure(figsize=(15,8))  
d2_filter.groupby('language').size().plot(kind='pie', autopct='%.2f')
```

```
Out[193... <AxesSubplot:ylabel='None'>
```



In [194...]

```
import datetime
d2_filter['demo_watched_date']=pd.to_datetime(d2_filter['demo_watched_date'])
d2_filter['month']=d2_filter['demo_watched_date'].dt.month
d2_filter['year']= d2_filter['demo_watched_date'].dt.year
```

<ipython-input-194-64a7ad3e6e8c>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
d2_filter['demo_watched_date']=pd.to_datetime(d2_filter['demo_watched_date'])
<ipython-input-194-64a7ad3e6e8c>:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
d2_filter['month']=d2_filter['demo_watched_date'].dt.month  
<ipython-input-194-64a7ad3e6e8c>:4: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
d2_filter['year']= d2_filter['demo_watched_date'].dt.year
```

In [195...]

```
d2_filter.head(15)
```

Out[195...]

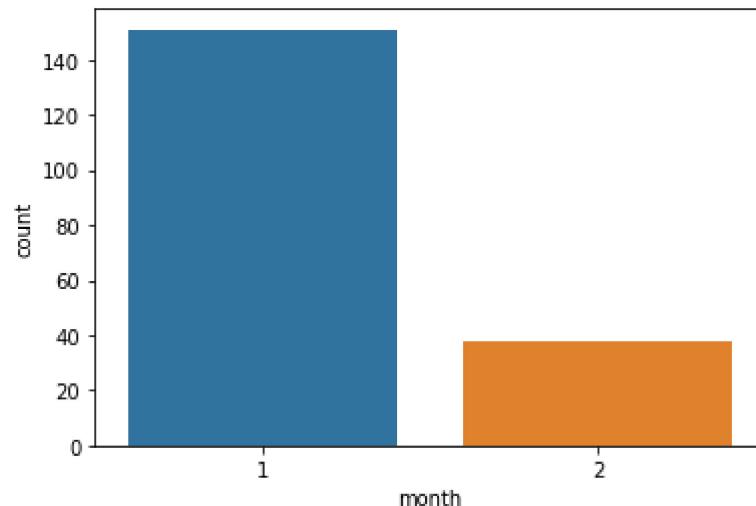
	lead_id	demo_watched_date	language	watched_percentage	month	year
0	USR1002	2022-01-04	Telugu	42	1	2022
1	USR1003	2022-01-07	Telugu	81	1	2022
2	USR1004	2022-01-02	Telugu	35	1	2022
3	USR1005	2022-01-03	Hindi	38	1	2022
4	USR1006	2022-01-12	Hindi	54	1	2022
5	USR1007	2022-01-11	English	52	1	2022
6	USR1008	2022-01-08	Telugu	41	1	2022
7	USR1009	2022-01-10	Telugu	50	1	2022
8	USR1010	2022-01-12	Telugu	48	1	2022
9	USR1011	2022-01-17	Telugu	44	1	2022
10	USR1012	2022-01-16	Telugu	46	1	2022
11	USR1013	2022-01-18	Telugu	50	1	2022
12	USR1014	2022-01-18	Telugu	51	1	2022
13	USR1015	2022-01-18	Hindi	44	1	2022
14	USR1016	2022-02-21	English	47	2	2022

In [196...]

```
sns.countplot(d2_filter['month'])
```

```
C:\Users\SURESH\Pyton\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword argument: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
warnings.warn(
```

```
Out[196... <AxesSubplot:xlabel='month', ylabel='count'>
```



```
In [197... d2_filter.month.value_counts()
```

```
Out[197... 1    151  
2     38  
Name: month, dtype: int64
```

```
In [198... d2_filter.year.value_counts()
```

```
Out[198... 2022    189  
Name: year, dtype: int64
```

Merging Data set1 - leads_basic_details and Data set2 - leads_demo_watched_details

where lead_id is primary key

```
In [203... d1_d2 = pd.merge(d1_filter,d2_filter ,how='inner', on = 'lead_id')
```

```
d1_d2 .head(5)
```

Out[203...]

	lead_id	age	gender	current_city	current_education	parent_occupation	lead_gen_source	demo_watched_date	language	watched_percentage
0	USR1002	20	MALE	Bengaluru	B.Tech	Business	user_referrals	2022-01-04	Telugu	
1	USR1003	20	FEMALE	Visakhapatnam	B.Tech	Lawyer	user_referrals	2022-01-07	Telugu	
2	USR1004	16	MALE	Mumbai	Intermediate	IT Employee	user_referrals	2022-01-02	Telugu	
3	USR1005	16	MALE	Chennai	Intermediate	Government Employee	user_referrals	2022-01-03	Hindi	
4	USR1006	16	MALE	Kochi	Intermediate	Doctor	user_referrals	2022-01-12	Hindi	

In [204...]

```
d1_d2.shape
```

Out[204...]

(188, 12)

In [205...]

```
d1_d2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 188 entries, 0 to 187
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   lead_id          188 non-null    object 
 1   age              188 non-null    int64  
 2   gender            188 non-null    object 
 3   current_city     188 non-null    object 
 4   current_education 188 non-null    object 
 5   parent_occupation 188 non-null    object 
 6   lead_gen_source   188 non-null    object 
 7   demo_watched_date 188 non-null    datetime64[ns]
 8   language          188 non-null    object 
 9   watched_percentage 188 non-null    int64  
 10  month             188 non-null    int64  
 11  year              188 non-null    int64  
dtypes: datetime64[ns](1), int64(4), object(7)
memory usage: 19.1+ KB
```

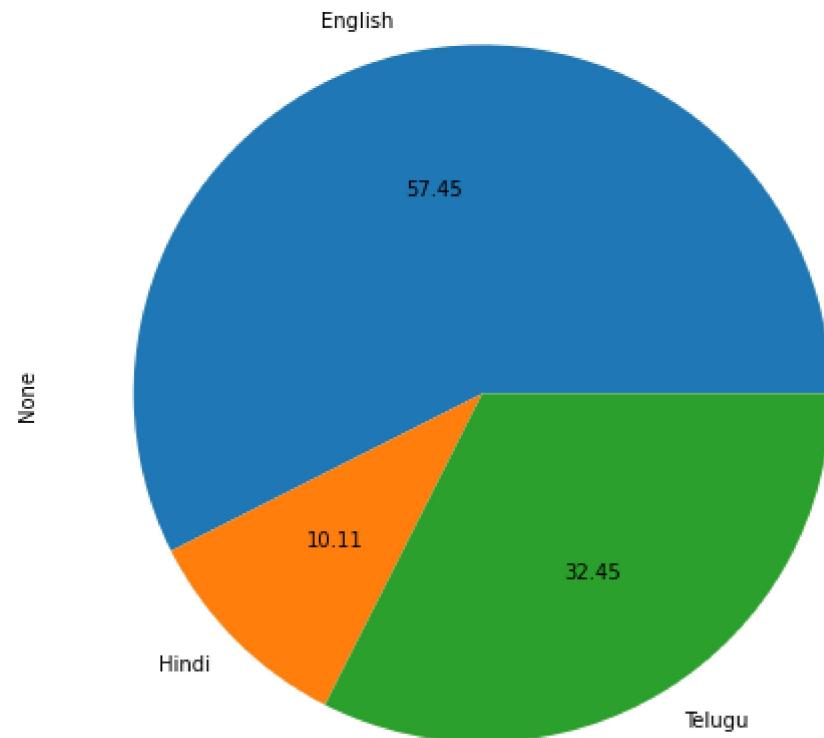
In [206...]

```
d1_d2.groupby(['language', 'current_city']).size()
```

```
Out[206... language current_city
English  Bengaluru      22
          Chennai        16
          Hyderabad     19
          Kochi         19
          Mumbai        10
          Visakhapatnam 22
Hindi    Bengaluru      3
          Chennai        6
          Hyderabad     3
          Kochi         4
          Mumbai        1
          Visakhapatnam 2
Telugu   Bengaluru      7
          Chennai        7
          Hyderabad     17
          Kochi         8
          Mumbai        7
          Visakhapatnam 15
dtype: int64
```

```
In [216... plt.figure(figsize=(15,8))
d1_d2.groupby('language').size().plot(kind='pie', autopct='%.2f')
```

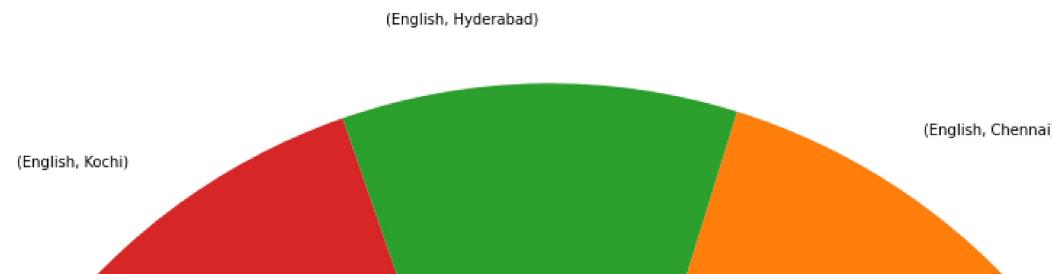
```
Out[216... <AxesSubplot:ylabel='None'>
```

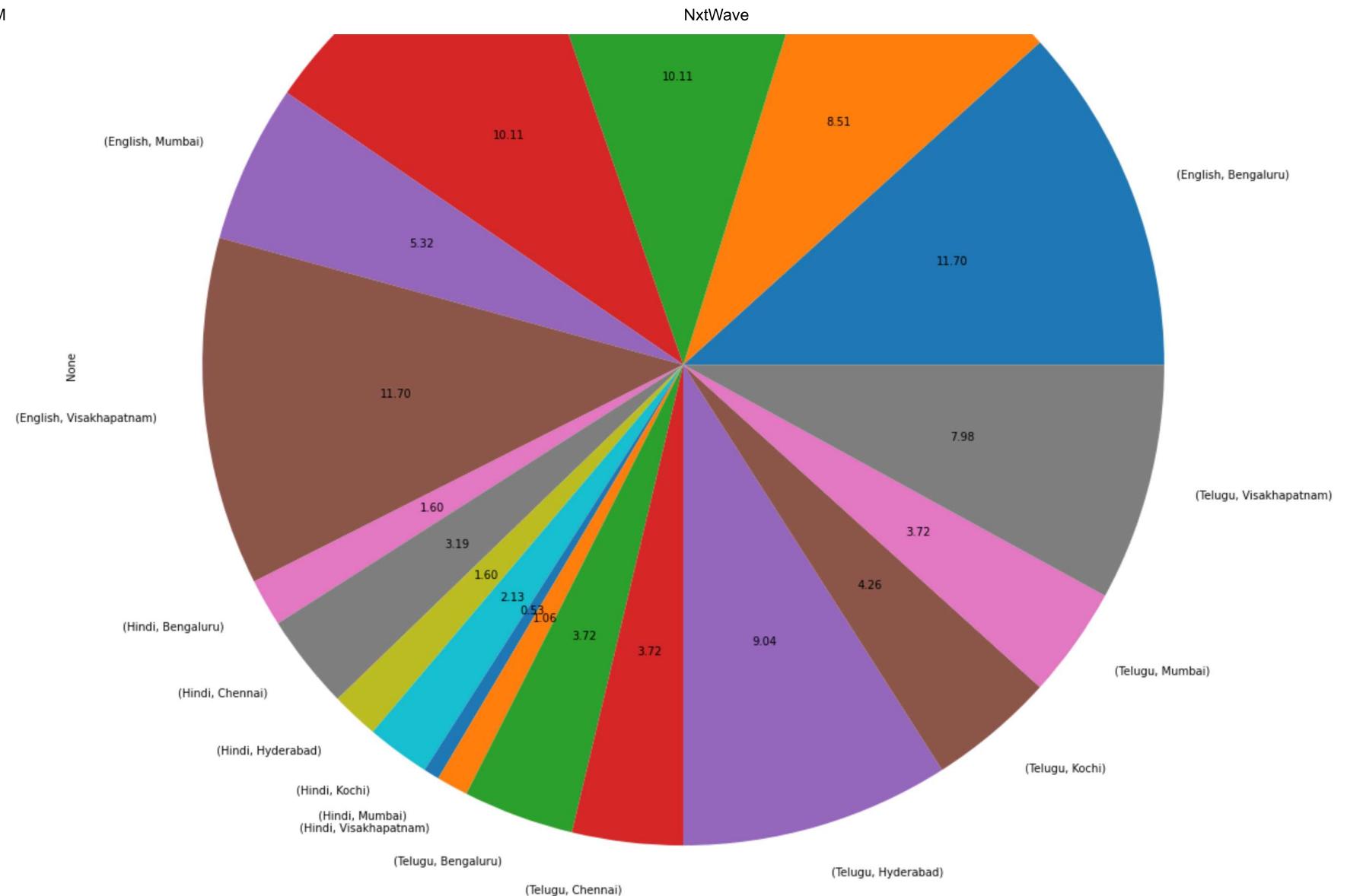


In [317...]

```
plt.figure(figsize=(25,20))
d1_d2.groupby(['language', 'current_city']).size().plot(kind='pie', autopct='%.2f')
```

Out[317... <AxesSubplot:ylabel='None'>





Data set3 - leads_interaction_details

```
In [218]: d3 = pd.read_csv('leads_interaction_details.csv')
```

In [219...]

d3.head()

Out[219...]

	jnr_sm_id	lead_id	lead_stage	call_done_date	call_status	call_reason
0	JNR1001MG	USR1001	lead	1/2/2022	successful	lead_introduction
1	JNR1001MG	USR1001	lead	1/2/2022	successful	demo_schedule
2	JNR1001MG	USR1002	lead	1/3/2022	successful	lead_introduction
3	JNR1001MG	USR1002	lead	1/4/2022	successful	demo_schedule
4	JNR1001MG	USR1002	awareness	1/5/2022	successful	post_demo_followup

In [220...]

d3.shape

Out[220...](2192, 6)

In [221...]

d3.describe()

Out[221...]

	jnr_sm_id	lead_id	lead_stage	call_done_date	call_status	call_reason
count	2192	2192	2192	2192	2192	2192
unique	16	358	4	40	2	8
top	JNR1012MG	USR1311	lead	1/8/2022	successful	demo_not_attended
freq	164	11	1547	156	1866	701

In [222...]

d3.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2192 entries, 0 to 2191
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   jnr_sm_id        2192 non-null   object  
 1   lead_id          2192 non-null   object  
 2   lead_stage       2192 non-null   object  
 3   call_done_date   2192 non-null   object  

```

```
4    call_status      2192 non-null   object
5    call_reason      2192 non-null   object
dtypes: object(6)
memory usage: 102.9+ KB
```

```
In [223...]: d3.isna().sum()
```

```
Out[223...]: jnr_sm_id      0
lead_id          0
lead_stage       0
call_done_date  0
call_status      0
call_reason      0
dtype: int64
```

```
In [229...]: d3.nunique()
```

```
Out[229...]: jnr_sm_id      16
lead_id          358
lead_stage       4
call_done_date  40
call_status      2
call_reason      8
dtype: int64
```

```
In [230...]: d3.duplicated().sum()
```

```
Out[230...]: 438
```

```
In [276...]: d3.drop_duplicates(inplace=True)
```

```
In [277...]: d3.shape
```

```
Out[277...]: (1754, 6)
```

```
In [278...]: d3.groupby(['lead_stage', 'call_status', 'call_reason']).size()
```

```
Out[278...]: lead_stage      call_status      call_reason
           awareness      successful      followup_for_consideration      30
                           post_demo_followup      194
```

	unsuccessful	followup_for_consideration	5
consideration	successful	followup_for_conversion	103
		interested_for_conversion	114
conversion	unsuccessful	followup_for_conversion	20
		interested_for_conversion	20
lead	successful	successful_conversion	64
	successful	demo_not_attended	277
		demo_schedule	321
		lead_introduction	329
unsuccessful	demo_not_attended		83
	demo_schedule		25
		lead_introduction	169

dtype: int64

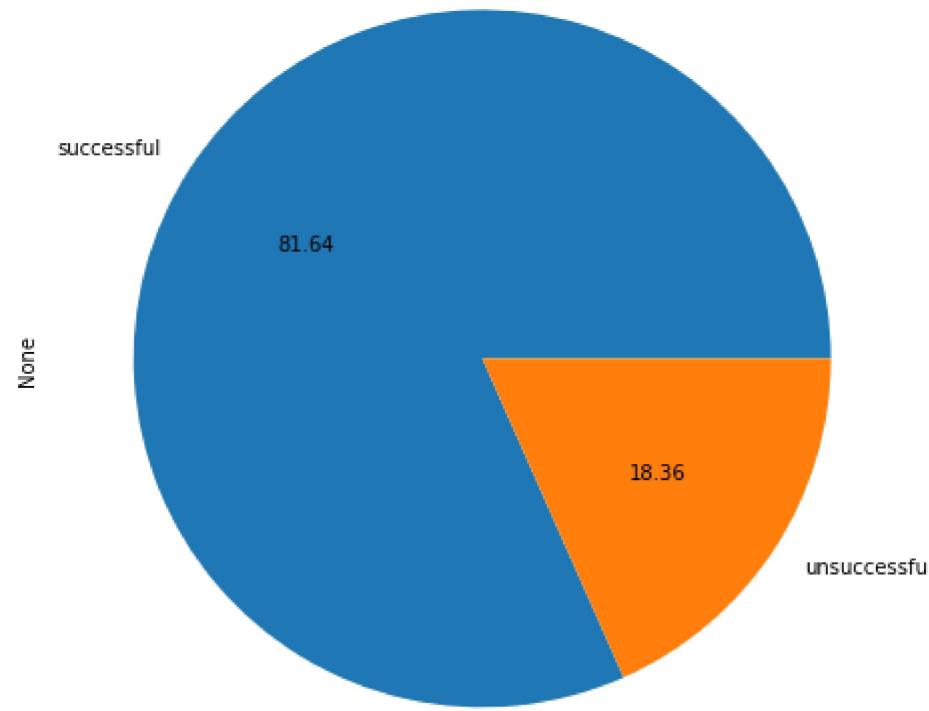
Where succesful call is higher than unsuccessful call throughtout the process.

In [279...]

```
plt.figure(figsize=(15,8))
d3.groupby('call_status').size().plot(kind='pie', autopct='%.2f')
```

Out[279...]

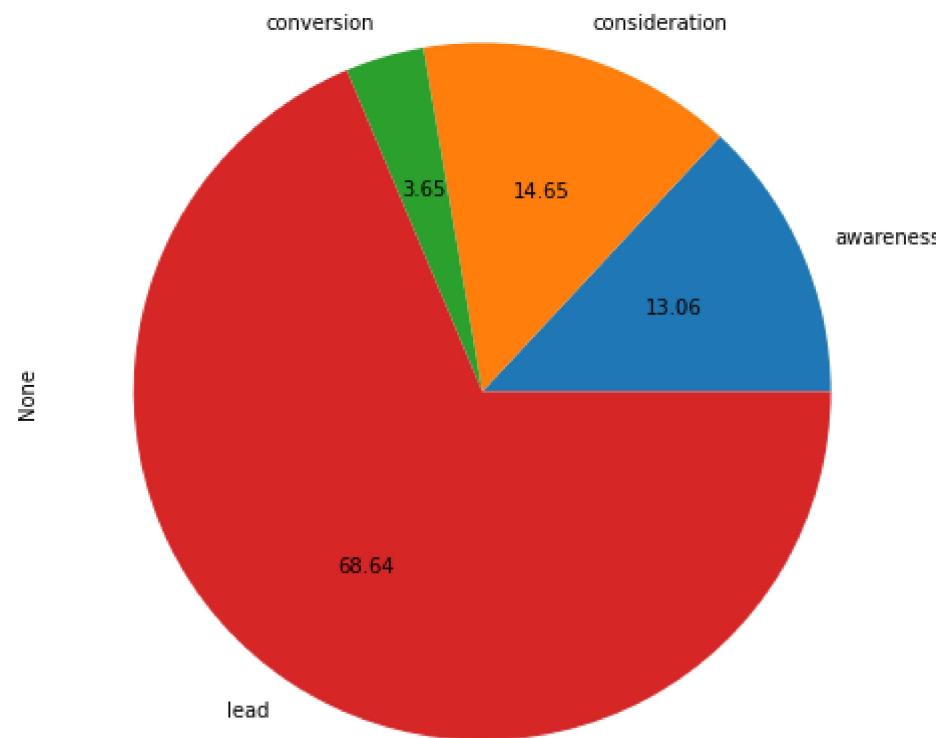
<AxesSubplot:ylabel='None'>



In [280...]

```
plt.figure(figsize=(15,8))
d3.groupby('lead_stage').size().plot(kind='pie', autopct='%.2f')
```

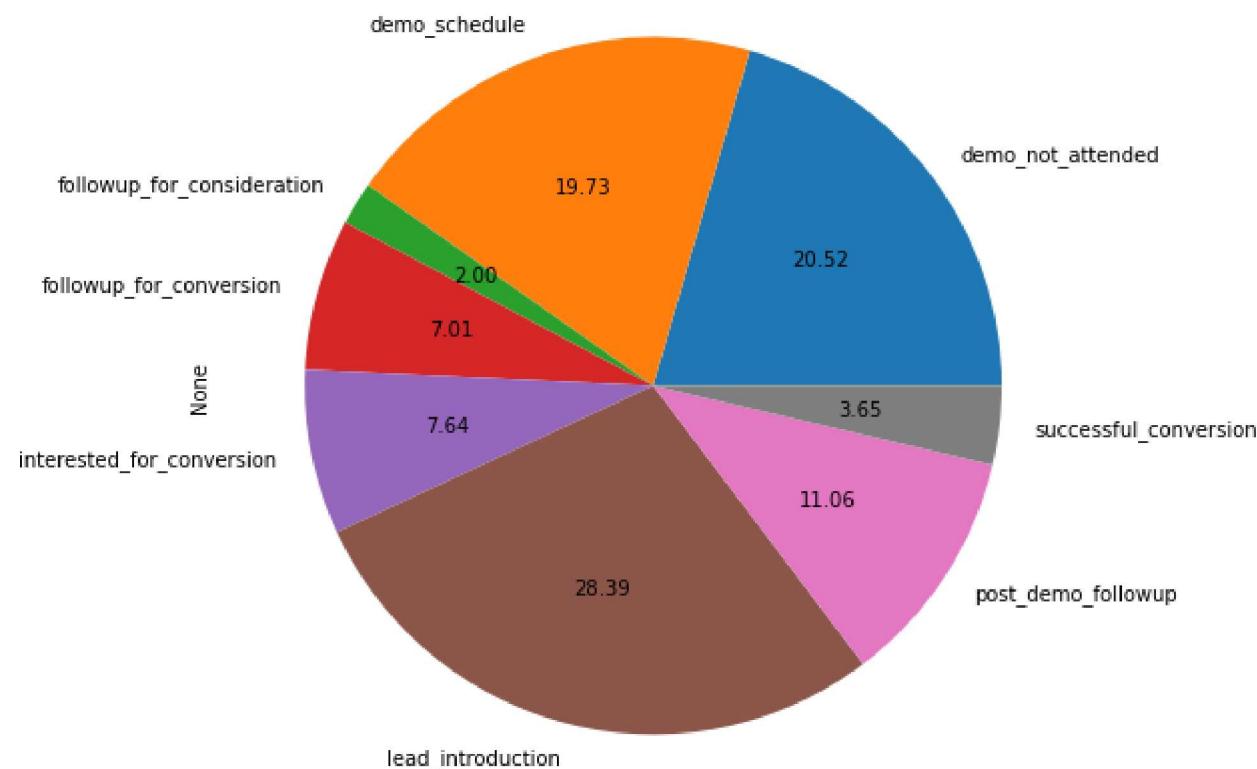
Out[280... <AxesSubplot:ylabel='None'>



In [281...]

```
plt.figure(figsize=(15,8))
d3.groupby('call_reason').size().plot(kind='pie', autopct='%.2f')
```

Out[281... <AxesSubplot:ylabel='None'>



Most of the call reason is lead_introduction and demo not attended

Data set4 - leads_reasons_for_no_interest

In [282...]

```
d4 = pd.read_csv('leads_reasons_for_no_interest.csv')
```

In [283...]

```
d4.head()
```

Out[283...]

	lead_id	reasons_for_not_interested_in_demo	reasons_for_not_interested_to_consider	reasons_for_not_interested_to_convert
0	USR1001	No time for student	NaN	NaN

	lead_id	reasons_for_not_interested_in_demo	reasons_for_not_interested_to_consider	reasons_for_not_interested_to_convert
1	USR1003	NaN	No time for student	NaN
2	USR1004	NaN	Wants offline classes	NaN
3	USR1005	NaN	Can't afford	NaN
4	USR1006	NaN	Student not interested in domain	NaN

In [284...]

d4.shape

Out[284...]: (294, 4)

In [285...]

d4.describe()

Out[285...]

	lead_id	reasons_for_not_interested_in_demo	reasons_for_not_interested_to_consider	reasons_for_not_interested_to_convert
count	294	164	79	51
unique	294	6	5	5
top	USR1110	Wants offline classes	Can't afford	Can't afford
freq	1	56	32	19

In [286...]

d4.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   lead_id          294 non-null    object 
 1   reasons_for_not_interested_in_demo 164 non-null    object 
 2   reasons_for_not_interested_to_consider 79 non-null    object 
 3   reasons_for_not_interested_to_convert 51 non-null    object 
dtypes: object(4)
memory usage: 9.3+ KB
```

In [287...]

d4.isna().sum()

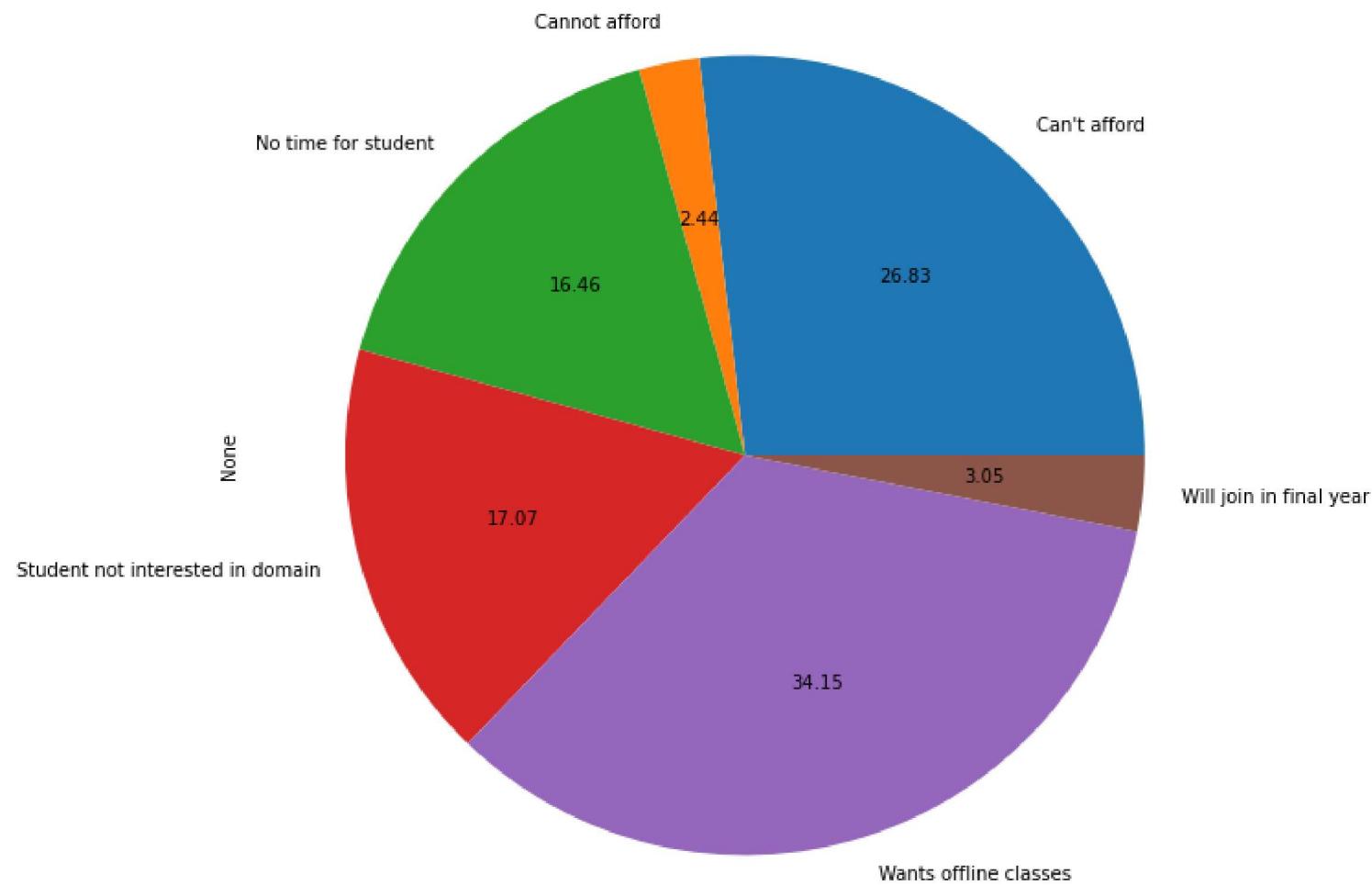
```
Out[287... lead_id          0  
reasons_for_not_interested_in_demo 130  
reasons_for_not_interested_to_consider 215  
reasons_for_not_interested_to_convert 243  
dtype: int64
```

```
In [289... d4.groupby(['reasons_for_not_interested_in_demo']).size()
```

```
Out[289... reasons_for_not_interested_in_demo  
Can't afford           44  
Cannot afford          4  
No time for student   27  
Student not interested in domain 28  
Wants offline classes 56  
Will join in final year 5  
dtype: int64
```

```
In [309... plt.figure(figsize=(25,10))  
d4.groupby('reasons_for_not_interested_in_demo').size().plot(kind='pie', autopct='%.2f')
```

```
Out[309... <AxesSubplot:ylabel='None'>
```



most of the leads no interest because they required offline classes

```
In [293...]: d5 = pd.read_csv('sales_managers_assigned_leads_details.csv')
```

```
In [296...]: d5.head()
```

```
Out[296...]
```

	snr_sm_id	jnr_sm_id	assigned_date	cycle	lead_id
0	SNR501MG	JNR1001MG	1/1/2022	1	USR1001
1	SNR501MG	JNR1001MG	1/1/2022	1	USR1002
2	SNR501MG	JNR1001MG	1/1/2022	1	USR1003
3	SNR501MG	JNR1001MG	1/1/2022	1	USR1004
4	SNR501MG	JNR1001MG	1/1/2022	1	USR1005

In [295...]

d5.shape

Out[295...](360, 5)

In [299...]

d5.describe()

Out[299...]

	cycle
count	360.000000
mean	2.469444
std	1.124139
min	1.000000
25%	1.000000
50%	2.000000
75%	3.000000
max	4.000000

In [300...]

d5.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 360 entries, 0 to 359
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ----- 
 0   snr_sm_id  360 non-null    object 
 1   jnr_sm_id  360 non-null    object 
 2   assigned_date 360 non-null    datetime64[ns]
 3   cycle       360 non-null    int64  
 4   lead_id     360 non-null    object 
```

```
0    snr_sm_id      360 non-null   object
1    jnr_sm_id      360 non-null   object
2  assigned_date    360 non-null   object
3     cycle         360 non-null  int64
4    lead_id        360 non-null   object
dtypes: int64(1), object(4)
memory usage: 14.2+ KB
```

In [301...]

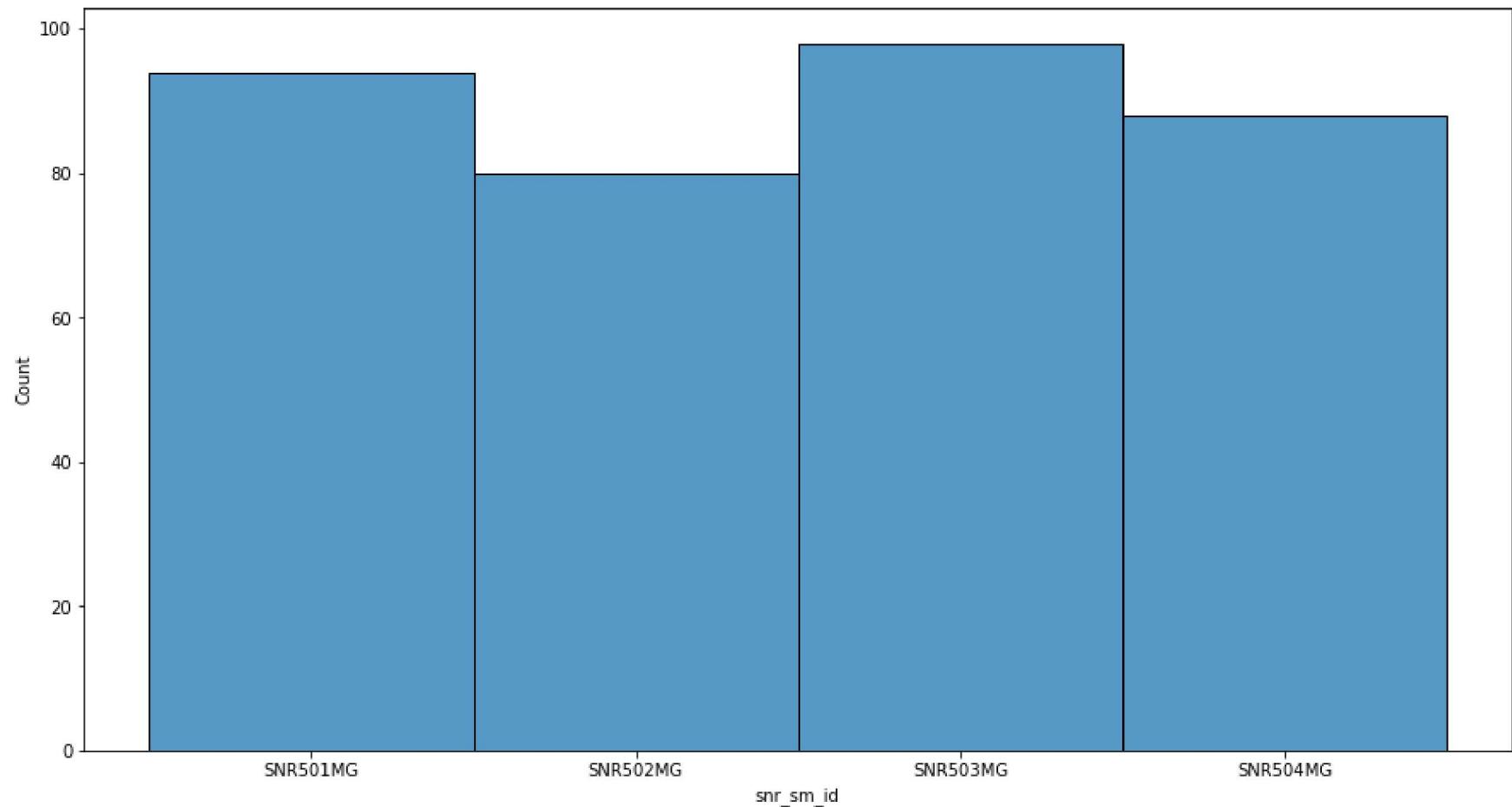
```
d5.groupby(['snr_sm_id','jnr_sm_id']).size()
```

Out[301...]

```
snr_sm_id  jnr_sm_id
SNR501MG    JNR1001MG    34
                  JNR1002MG    20
                  JNR1003MG    20
                  JNR1004MG    20
SNR502MG    JNR1005MG    20
                  JNR1006MG    20
                  JNR1007MG    20
                  JNR1008MG    20
SNR503MG    JNR1009MG    20
                  JNR1010MG    21
                  JNR1011MG    26
                  JNR1012MG    31
SNR504MG    JNR1013MG    20
                  JNR1014MG    20
                  JNR1015MG    21
                  JNR1016MG    27
dtype: int64
```

In [316...]

```
plt.figure(figsize=(15,8))
ax=sns.histplot(data=d5["snr_sm_id"],bins=5)
```



Number of Junior Sales Managers are evenly assigned among Senior Sales Managers.

In []: