In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
C:\ProgramData\Anaconda3\envs\lib\site-packages\scipy\__init__.py:146: UserWa
rning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of S
ciPy (detected version 1.26.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
data1 = pd.read_csv(r"C:\Users\Sinha Rahul\Downloads\ecommerce-purchases.csv.z
```

In [3]:
```
data1
```

Out[3]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	Se
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/19	
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/24	
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20 IE	AM	Opera/9.58.(X11; Linux x86_64; it-IT) Presto/2...	Brown, Watson and Andrews	6011456623207998	10/25	
...	...	...	...	...	...	...	...	
9995	966 Castaneda Locks\nWest Juliafurt, CO 96415	92 XI	PM	Mozilla/5.0 (Windows NT 5.1) AppleWebKit/5352 ...	Randall-Sloan	342945015358701	03/22	
9996	832 Curtis Dam Suite 785\nNorth Edwardburgh, T...	41 JY	AM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Hale, Collins and Wilson	210033169205009	07/25	
9997	Unit 4434 Box 6343\nDPO AE 28026-0283	74 Zh	AM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...	Anderson Ltd	6011539787356311	05/21	
9998	0096 English Rest\nRoystad, IA 12457	74 cL	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_8;...	Cook Inc	180003348082930	11/17	
9999	40674 Barrett Stravenue\nGrimesville, WI 79682	64 Hr	AM	Mozilla/5.0 (X11; Linux i686; rv:1.9.5.20) Gec...	Greene Inc	4139972901927273	02/19	

10000 rows × 14 columns

Display Top 10 row in dataset

In [4]:
```
data1.head(10)
```

Out[4]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	C Ex Dat
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/2
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/1
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/1
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/2

In [5]:
```
data1.head(-5)
```

Out[5]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56. (X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	900	
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	561	M
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/19	699	
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/24	384	
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20 IE	AM	Opera/9.58. (X11; Linux x86_64; it-IT) Presto/2...	Brown, Watson and Andrews	6011456623207998	10/25	678	
...	...	...	...	...	...	...	...	...	
9990	75731 Molly Springs\nWest Danielle, VT 96934-5102	93 ty	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4;...	Pace, Vazquez and Richards	869968197049750	04/24	877	
9991	PSC 8165, Box 8498\nAPO AP 60327-0346	50 dA	AM	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT ...	Snyder Inc	4221582137197481	02/24	969	
9992	885 Allen Mountains Apt. 230\nWallhaven, LA 16995	40 vH	PM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_6_5) A...	Wells Ltd	4664825258997302	10/20	431	
9993	7555 Larson Locks Suite 229\nEllisburgh, MA 34...	72 jg	PM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_8...	Colon and Sons	30025560104631	10/25	629	
9994	6276 Rojas Hollow\nLake Louis, WY 56410-7837	93 Ex	PM	Opera/9.68. (X11; Linux x86_64; sl-SI) Presto/2...	Ritter-Smith	3112186784121077	01/25	1823	

9995 rows × 14 columns

check last 10 ROW IN DATASET

In [6]:
```
data1.tail(10)
```

Out[6]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date
9990	75731 Molly Springs\nWest Danielle, VT 96934-5102	93 ty	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4;...	Pace, Vazquez and Richards	869968197049750	04/24
9991	PSC 8165, Box 8498\nAPO AP 60327-0346	50 dA	AM	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT ...	Snyder Inc	4221582137197481	02/24
9992	885 Allen Mountains Apt. 230\nWallhaven, LA 16995	40 vH	PM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_6_5) A...	Wells Ltd	4664825258997302	10/20
9993	7555 Larson Locks Suite 229\nEllisburgh,	72	PM	Mozilla/5.0 (Macintosh; U;	Colon	30025560104631	10/25

In [7]:
```
data1.tail(-5)
```

Out[7]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	C Ex Dat
5	7502 Powell Mission Apt. 768\nTravisland, VA 3...	21 XT	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_5...	Silva-Anderson	30246185196287	07/2
6	93971 Conway Causeway\nAndersonburgh, AZ 75107	96 Xt	AM	Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...	Gibson and Sons	6011398782655569	07/2
7	260 Rachel Plains Suite 366\nCastroberg, WV 24...	96 pG	PM	Mozilla/5.0 (X11; Linux i686) AppleWebKit/5350...	Marshall-Collins	561252141909	06/2
8	2129 Dylan Burg\nNew Michelle, ME 28650	45 JN	PM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...	Galloway and Sons	180041795790001	04/2
9	3795 Dawson Extensions\nLake Tinafort, ID 88739	15 Ug	AM	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gec...	Rivera, Buchanan and Ramirez	4396283918371	01/1
...	...	...	...	...	...	...	.
9995	966 Castaneda Locks\nWest Juliafurt, CO 96415	92 XI	PM	Mozilla/5.0 (Windows NT 5.1) AppleWebKit/5352 ...	Randall-Sloan	342945015358701	03/2
9996	832 Curtis Dam Suite 785\nNorth Edwardburgh, T...	41 JY	AM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Hale, Collins and Wilson	210033169205009	07/2
9997	Unit 4434 Box 6343\nDPO AE 28026-0283	74 Zh	AM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...	Anderson Ltd	6011539787356311	05/2
9998	0096 English Rest\nRoystad, IA 12457	74 cL	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_8;...	Cook Inc	180003348082930	11/1
9999	40674 Barrett Stravenue\nGrimesville, WI 79682	64 Hr	AM	Mozilla/5.0 (X11; Linux i686; rv:1.9.5.20) Gec...	Greene Inc	4139972901927273	02/1

9995 rows × 14 columns

check datatype in each column

In [8]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Address           10000 non-null  object 
 1   Lot               10000 non-null  object 
 2   AM or PM          10000 non-null  object 
 3   Browser Info      10000 non-null  object 
 4   Company           10000 non-null  object 
 5   Credit Card       10000 non-null  int64  
 6   CC Exp Date       10000 non-null  object 
 7   CC Security Code  10000 non-null  int64  
 8   CC Provider       10000 non-null  object 
 9   Email             10000 non-null  object 
 10  Job               10000 non-null  object 
 11  IP Address        10000 non-null  object 
 12  Language          10000 non-null  object 
 13  Purchase Price    10000 non-null  float64
dtypes: float64(1), int64(2), object(11)
memory usage: 1.1+ MB
```

check describe in dataset

In [9]:
```
data1.describe()
```

Out[9]:

	Credit Card	CC Security Code	Purchase Price
count	1.000000e+04	10000.000000	10000.000000
mean	2.341374e+15	907.217800	50.347302
std	2.256103e+15	1589.693035	29.015836
min	6.040186e+10	0.000000	0.000000
25%	3.056322e+13	280.000000	25.150000
50%	8.699942e+14	548.000000	50.505000
75%	4.492298e+15	816.000000	75.770000
max	6.012000e+15	9993.000000	99.990000

check null value in dataset

In [10]:
```
data1.isnull().values.any()
```

Out[10]:
```
False
```

how many row and column are there in our dataset

In [11]:
```
data1.shape
```

Out[11]:
```
(10000, 14)
```

In [12]:
```
print("the number of row is",data1.shape[0])
print("the number of columns is",data1.shape[1])
```

```
the number of row is 10000
the number of columns is 14
```

In [13]:
```
data1.columns
```

Out[13]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [14]:
```
len(data1.columns)
```

Out[14]:
```
14
```

In [15]:
```
len(data1)
```

Out[15]:
```
10000
```

In [16]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Address           10000 non-null  object
 1   Lot               10000 non-null  object
 2   AM or PM          10000 non-null  object
 3   Browser Info      10000 non-null  object
 4   Company           10000 non-null  object
 5   Credit Card       10000 non-null  int64
 6   CC Exp Date       10000 non-null  object
 7   CC Security Code  10000 non-null  int64
 8   CC Provider       10000 non-null  object
 9   Email             10000 non-null  object
 10  Job               10000 non-null  object
 11  IP Address        10000 non-null  object
 12  Language          10000 non-null  object
 13  Purchase Price    10000 non-null  float64
```

In [17]:
```
for col in data1.columns:
        print(col)
```

```
Address
Lot
AM or PM
Browser Info
Company
Credit Card
CC Exp Date
CC Security Code
CC Provider
Email
Job
IP Address
Language
Purchase Price
```

In [18]:
```
dff3 = [col for col in data1.columns if data1[col].dtypes!="O"]
```

In [19]:
```
dff3
```

Out[19]:
```
['Credit Card', 'CC Security Code', 'Purchase Price']
```

In [20]:
```
[col for col in data1.columns if data1[col].dtypes=="O"]
```

Out[20]:
```
['Address',
 'Lot',
 'AM or PM',
 'Browser Info',
 'Company',
 'CC Exp Date',
 'CC Provider',
 'Email',
 'Job',
 'IP Address',
 'Language']
```

Highest and lowest purchase price

In [21]:
```
data1.columns
```

Out[21]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [22]:
```
data1["Purchase Price"].max()
```

Out[22]:
```
99.99
```

In [23]:
```
data1["Purchase Price"].min()
```

Out[23]:
```
0.0
```

Avarge purchase price

In [24]:
```
data1["Purchase Price"].mean()
```

Out[24]:
```
50.34730200000025
```

How many people have French "fr" as their language

In [25]:
```
data1["Language"] == "fr"
```

Out[25]:
```
0       False
1        True
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: Language, Length: 10000, dtype: bool
```

In [26]:
```
data1[data1["Language"] == "fr"]
```

Out[26]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	561	I
19	125 Hall Summit\nBoothton, IL 41721	99 CU	PM	Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...	Turner-Mckinney	676343504830	02/20	440	
53	PSC 9431, Box 7059\nAPO AA 29285-1363	14 qD	AM	Opera/9.34. (X11; Linux x86_64; it-IT) Presto/2...	Higgins, Cardenas and Kennedy	869972604798355	08/17	157	
76	49206 Campbell Port\nNorth Cliffordshire, HI 3...	71 iu	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_9...	Jacobs-Tucker	6011343518820988	01/17	806	
82	493 Smith Valleys Suite 004\nNew Madelineville...	35 ls	PM	Mozilla/5.0 (iPod; U; CPU iPhone OS 4_0 like M...	Robinson, Johnston and Valdez	4351359627548412	06/17	937	
...	...	...	...	...	...	...	...	...	
9941	43757 Brown Lodge\nValerieberg, PR 88518	68 DG	AM	Mozilla/5.0 (compatible; MSIE 5.0; Windows NT ...	Jones, Williams and Dunn	6011508474487291	04/26	906	I
9947	32827 Carl Inlet\nSouth Nicole, NY 38081-6636	40 Dp	PM	Mozilla/5.0 (Windows; U; Windows 95) AppleWebK...	Pruitt-Flores	3158564208754951	03/17	386	I
9951	518 Brown Dam\nGarciaside, IN 33057	71 wq	AM	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT ...	Crosby, Vasquez and Ballard	5256758441931287	12/17	146	
9977	02182 Keith Expressway\nEast Shannon, CT 20578...	34 RL	AM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Deleon, Jacobson and Benton	4186094003664688	06/21	397	
9980	6061 Dave Lights\nPhillipsview, UT 40500	25 Mv	AM	Opera/8.58. (Windows NT 5.0; it-IT) Presto/2.9....	Cortez-Frye	6011944199756993	12/24	755	

1097 rows × 14 columns

In [27]:
```
len(data1[data1["Language"] == "fr"])
```

Out[27]:
```
1097
```

job title contains engineer

In [28]:
```
data1["Job"]
```

Out[28]:
```
0       Scientist, product/process development
1                            Drilling engineer
2                    Customer service manager
3                            Drilling engineer
4                                  Fine artist
                          ...
9995                                Printmaker
9996                          Energy engineer
9997                        Veterinary surgeon
9998                  Local government officer
9999                    Embryologist, clinical
Name: Job, Length: 10000, dtype: object
```

In [29]:
```
data1["Job"].str.contains("engineer")
```

Out[29]:
```
0       False
1        True
2       False
3        True
4       False
        ...
9995    False
9996     True
9997    False
9998    False
9999    False
Name: Job, Length: 10000, dtype: bool
```

In [30]:
```
data1[data1["Job"].str.contains("engineer")]
```

Out[30]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Securit Cod
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	56
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/24	38
50	41159 Michael Centers\nAdamsfort, RI 37108-6674	46 Ce	PM	Mozilla/5.0 (Windows 98; Win 9x 4.90; sl-SI; r...	Wright, Williams and Mendez	4008586485908075	05/19	94

In [31]:
```
len(data1[data1["Job"].str.contains("engineer")])
```

Out[31]: 531

In [32]:
```
len(data1[data1["Job"].str.contains("engineer", case = False)])
```

Out[32]: 984

Find Email of the person with following ip address:# 132.207.160.22

In [33]:
```
data1["IP Address"]
```

Out[33]:
```
0       149.146.147.205
1          15.160.41.51
2        132.207.160.22
3          30.250.74.19
4          24.140.33.94
              ...
9995      29.73.197.114
9996     121.133.168.51
9997      156.210.0.254
9998       55.78.26.143
9999    176.119.198.199
Name: IP Address, Length: 10000, dtype: object
```

In [34]:
```
data1["IP Address"]== "132.207.160.22"
```

Out[34]:
```
0       False
1       False
2        True
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: IP Address, Length: 10000, dtype: bool
```

In [35]:
```
data1[data1["IP Address"]== "132.207.160.22"]
```

Out[35]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider	
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/19	699	JCB 16 digit	amymill

In [36]:
```
data1[data1["IP Address"]== "132.207.160.22"]["Email"]
```

Out[36]:
```
2    amymiller@morales-harrison.com
Name: Email, dtype: object
```

In [37]:
```
data1[data1["IP Address"]== "121.133.168.51"]["Email"]
```

Out[37]:
```
9996    mary85@hotmail.com
Name: Email, dtype: object
```

In [38]:
```
data1[data1["IP Address"]== "121.133.168.51"]["Job"]
```

Out[38]:
```
9996    Energy engineer
Name: Job, dtype: object
```

How many people have mastercard as their credit card provider and made a purchase above 50?

In [39]:
```
data1["CC Provider"]
```

Out[39]:
```
0                    JCB 16 digit
1                      Mastercard
2                    JCB 16 digit
3                        Discover
4       Diners Club / Carte Blanche
                   ...
9995                 JCB 15 digit
9996                 JCB 16 digit
9997                VISA 16 digit
9998              American Express
9999                 JCB 15 digit
Name: CC Provider, Length: 10000, dtype: object
```

In [40]:
```
data1["CC Provider"]=="Mastercard"
```

Out[40]:
```
0       False
1        True
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: CC Provider, Length: 10000, dtype: bool
```

In [41]:
```
data1[data1["CC Provider"]=="Mastercard"]
```

Out[41]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	Se
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	
18	461 Christopher Square\nWest Michaelchester, C...	17 SB	PM	Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gec...	Beard, Abbott and Pena	6011350184276270	12/22	
31	USNS Alvarado\nFPO AA 27052-1231	26 Lh	PM	Opera/8.84. (X11; Linux i686; sl-SI) Presto/2.9...	Nicholson Group	4614997834548	03/22	
	93392 Webb Gardens			Mozilla/5.0				

In [42]:
```
data1[(data1["CC Provider"]=="Mastercard") & (data1["Purchase Price"]>50)]
```

Out[42]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	Se
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	
18	461 Christopher Square\nWest Michaelchester, C...	17 SB	PM	Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gec...	Beard, Abbott and Pena	6011350184276270	12/22	
31	USNS Alvarado\nFPO AA 27052-1231	26 Lh	PM	Opera/8.84. (X11; Linux i686; sl-SI) Presto/2.9...	Nicholson Group	4614997834548	03/22	
35	93392 Webb Gardens Apt. 220\nLaurabury, AR 999...	37 om	AM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...	Mora Ltd	6011049630969815	09/16	
90	431 Bowen Lights\nFergusonborough, MH 01362	31 tG	AM	Mozilla/5.0 (Windows NT 5.2; it-IT; rv:1.9.1.2...	Copeland-Lee	639032576097	06/18	
...	...	...	...	...	...	...	...	...
9941	43757 Brown Lodge\nValerieberg, PR 88518	68 DG	AM	Mozilla/5.0 (compatible; MSIE 5.0; Windows NT ...	Jones, Williams and Dunn	6011508474487291	04/26	
9948	95544 Johnson Isle Suite 939\nMichaelberg, RI ...	91 bW	AM	Opera/8.36. (X11; Linux x86_64; sl-SI) Presto/2...	Fox-Peterson	4762924304307	03/17	
9954	051 Samantha Wells Apt. 328\nScottborough, OR ...	11 vG	AM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_2...	Mendoza, Zimmerman and Reilly	30232295374907	05/21	
9981	087 Larson Roads Apt. 587\nRicestad, WI 98077-...	95 Kw	AM	Opera/9.63. (Windows NT 5.01; en-US) Presto/2.9...	Ortiz, Schultz and Adams	4690863684605	01/19	
9987	2754 Klein Mission\nJohnsonview, PA 64925-9624	65 Fg	PM	Opera/9.26. (Windows CE; sl-SI) Presto/2.9.188 ...	Curtis LLC	180027305702263	12/24	

405 rows × 14 columns

In [43]:
```
len(data1[(data1["CC Provider"]=="Mastercard") & (data1["Purchase Price"]>50)])
```

Out[43]:
```
405
```

Find Email of the person with following Credit Card number:# 4664825258997302

In [44]:
```
data1["Credit Card"]
```

Out[44]:
```
0       6011929061123406
1       3337758169645356
2          675957666125
3       6011578504430710
4       6011456623207998
              ...
9995     342945015358701
9996     210033169205009
9997    6011539787356311
9998     180003348082930
9999    4139972901927273
Name: Credit Card, Length: 10000, dtype: int64
```

In [45]:
```
data1["Credit Card"]== "46648225258997302"
```

Out[45]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: Credit Card, Length: 10000, dtype: bool
```

In [46]:
```
data1[data1["Credit Card"]== 4664825258997302]
```

Out[46]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	Provi
9992	885 Allen Mountains Apt. 230\nWallhaven, LA 16995	40 vH	PM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_6_5) A...	Wells Ltd	4664825258997302	10/20	431	Disco

In [47]:
```
data1[data1["Credit Card"]==4664825258997302]["Email"]
```

Out[47]:
```
9992    bberry@wright.net
Name: Email, dtype: object
```

How Many people purchase During The Am and How Many people purchase during Pm?

In [48]:
```
data1["AM or PM"].value_counts()
```

Out[48]:
```
PM    5068
AM    4932
Name: AM or PM, dtype: int64
```

How Many people Have A credit Card That expire In 2020

Using Lamada Function

In [49]:
```
data1["CC Exp Date"]
```

Out[49]:
```
0       02/20
1       11/18
2       08/19
3       02/24
4       10/25
        ...
9995    03/22
9996    07/25
9997    05/21
9998    11/17
9999    02/19
Name: CC Exp Date, Length: 10000, dtype: object
```

In [50]:
```
data1["CC Exp Date"]== 2020
```

Out[50]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: CC Exp Date, Length: 10000, dtype: bool
```

In [51]:
```
data1["CC Exp Date"].apply(lambda x:x.split("/")[1] == "20")
```

Out[51]:
```
0        True
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: CC Exp Date, Length: 10000, dtype: bool
```

In [52]:
```
data1[data1["CC Exp Date"].apply(lambda x:x.split("/")[1] == "20")]
```

Out[52]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	C Securit Cod
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56. (X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	90
19	125 Hall Summit\nBoothton, IL 41721	99 CU	PM	Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...	Turner-Mckinney	676343504830	02/20	44
32	Unit 3628 Box 6778\nDPO AE 72362	39 Qm	PM	Mozilla/5.0 (Windows 98; Win 9x 4.90) AppleWeb...	Martinez-Wilson	4942281854569455	01/20	836
36	9374 Skinner Common Apt. 254\nChristopherfort,...	80 Fq	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Hanna-Grant	180042289507877	09/20	91
38	9671 Riley Drives Apt. 746\nPort Davidtown, TN...	15 vj	AM	Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gec...	Bryant, Hubbard and Gonzales	210094965373094	12/20	24
...	...	...	...	...	...	...	...	.
9883	751 Petersen Groves\nLake Rebecca, NM 55875	61 tE	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_7...	Perez and Sons	4939814860063025	05/20	23
9901	908 Norman Burgs\nPhillipsberg, VT 92023-9956	90 Ng	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Hughes Inc	30148833294789	09/20	48
9911	44320 Gray Centers Suite 325\nJenniferfurt, OK...	19 Bf	PM	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gec...	Jones-Gray	4058074716188	12/20	92
9957	73556 Chase Pine\nBrucemouth, MP 73473	25 rU	PM	Opera/9.49. (Windows 95; en-US) Presto/2.9.189 ...	Martin, Clayton and Jarvis	3337737293232646	05/20	79
9992	885 Allen Mountains Apt. 230\nWallhaven, LA 16995	40 vH	PM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_6_5) A...	Wells Ltd	4664825258997302	10/20	43

988 rows × 14 columns

In [53]:
```
len(data1[data1["CC Exp Date"].apply(lambda x:x.split("/")[1] == "20")])
```

Out[53]: 988

In [54]:
```
list = []
for email in data1["CC Exp Date"]:
    list.append(email.split("/")[1])
```

In [55]:
```
list
```

Out[55]:
```
['20',
 '18',
 '19',
 '24',
 '25',
 '25',
 '24',
 '25',
 '24',
 '17',
 '17',
 '26',
 '26',
 '25',
 '22',
 '25',
 '22',
 '26',
 '22',
```

In [56]:
```
uni_list = []
for item in list:
    if item not in uni_list:
        uni_list.append(item)
```

In [57]:
```
uni_list
```

Out[57]: ['20', '18', '19', '24', '25', '17', '26', '22', '23', '16', '21']

In [58]:
```
data1["year"]=list
```

In [59]:
```
data1.head()
```

Out[59]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	Pro
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56. (X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	900	J
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	561	Maste
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/19	699	J
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/24	384	Dis
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20 IE	AM	Opera/9.58. (X11; Linux x86_64; it-IT) Presto/2...	Brown, Watson and Andrews	6011456623207998	10/25	678	I Bl

In [60]:
```
data1["year"].value_counts()
```

Out[60]:
```
25    1033
23    1010
21    1006
22     996
18     995
19     995
24     992
20     988
17     955
26     654
16     376
Name: year, dtype: int64
```

top 5 most popular email provider(e.g.gmail.com, yahoo.com,etc....)

using def function

In [61]:
```
list = []
for email in data1["Email"]:
    list.append(email.split("@")[1])
```

In [62]:
```
list
```

Out[62]:
```
['yahoo.com',
 'reed.com',
 'morales-harrison.com',
 'olson-robinson.info',
 'gmail.com',
 'gmail.com',
 'yahoo.com',
 'parks.info',
 'rasmussen.com',
 'hunt-huerta.com',
 'hotmail.com',
 'coleman.com',
 'hotmail.com',
 'yahoo.com',
 'yahoo.com',
 'yahoo.com',
 'wilson.com',
 'hall.com',
 'yahoo.com',
```

In [63]:
```
len(list)
```

Out[63]:
```
10000
```

In [64]:
```
len(list)
```

Out[64]:
```
10000
```

In [65]:
```
uni_list = []
for value in list:
     if value not in uni_list:
        uni_list.append(value)
```

In [66]:
```
uni_list
```

Out[66]:
```
['yahoo.com',
 'reed.com',
 'morales-harrison.com',
 'olson-robinson.info',
 'gmail.com',
 'parks.info',
 'rasmussen.com',
 'hunt-huerta.com',
 'hotmail.com',
 'coleman.com',
 'wilson.com',
 'hall.com',
 'smith.info',
 'gardner-meadows.com',
 'mooney.com',
 'atkinson.com',
 'carey-davis.com',
 'mcdowell.com',
 'walsh.com',
```

In [67]:
```
len(uni_list)
```

Out[67]:
```
3416
```

In [68]:
```
len(uni_list)
```

Out[68]:
```
3416
```

In [69]:
```
data1["com"] = list
```

In [70]:
```
data1.head()
```

Out[70]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	Pro
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56. (X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	900	J
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93. (Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	561	Maste
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/19	699	J
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/24	384	Dis
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20 IE	AM	Opera/9.58. (X11; Linux x86_64; it-IT) Presto/2...	Brown, Watson and Andrews	6011456623207998	10/25	678	I Bl

In [71]:
```
data1["com"].value_counts()
```

Out[71]:
```
hotmail.com           1638
yahoo.com             1616
gmail.com             1605
smith.com               42
williams.com            37
                      ...
booker.com               1
woods-allen.biz          1
richards-wilson.com      1
morris-thomas.com        1
wade-garner.com          1
Name: com, Length: 3416, dtype: int64
```

In [72]:
```
data1["com"].value_counts().head(5)
```

Out[72]:
```
hotmail.com    1638
yahoo.com      1616
gmail.com      1605
smith.com        42
williams.com     37
Name: com, dtype: int64
```

Using Lamada Function

In [73]:
```
data1["Email"]
```

Out[73]:
```
0                    pdunlap@yahoo.com
1                  anthony41@reed.com
2       amymiller@morales-harrison.com
3          brent16@olson-robinson.info
4          christopherwright@gmail.com
                      ...
9995            iscott@wade-garner.com
9996                mary85@hotmail.com
9997                 tyler16@gmail.com
9998           elizabethmoore@reid.net
9999             rachelford@vaughn.com
Name: Email, Length: 10000, dtype: object
```

In [74]:
```
data1["Email"].apply(lambda x : x.split("@")[1])
```

Out[74]:
```
0                  yahoo.com
1                   reed.com
2       morales-harrison.com
3        olson-robinson.info
4                  gmail.com
                   ...
9995         wade-garner.com
9996             hotmail.com
9997               gmail.com
9998                reid.net
9999              vaughn.com
Name: Email, Length: 10000, dtype: object
```

In [75]:
```
data1["Email"].apply(lambda x : x.split("@")[1]).value_counts()
```

Out[75]:
```
hotmail.com            1638
yahoo.com              1616
gmail.com              1605
smith.com                42
williams.com             37
                       ...
booker.com                1
woods-allen.biz           1
richards-wilson.com       1
morris-thomas.com         1
wade-garner.com           1
Name: Email, Length: 3416, dtype: int64
```

In [76]:
```
data1["Email"].apply(lambda x : x.split("@")[1]).value_counts().head(5)
```

Out[76]:
```
hotmail.com     1638
yahoo.com       1616
gmail.com       1605
smith.com         42
williams.com      37
Name: Email, dtype: int64
```

without lamabda function

In [77]:
```
data1.columns
```

Out[77]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price', 'year', 'com'],
      dtype='object')
```

In [78]:
```
data1['Email'].str.split('@').str[1].value_counts().head(5)
```

Out[78]:
```
hotmail.com     1638
yahoo.com       1616
gmail.com       1605
smith.com         42
williams.com      37
Name: Email, dtype: int64
```

In [79]:
```
df1 = pd.DataFrame( data1['Email'].str.split('@').str[1].value_counts().head(5
df1
```

Out[79]:

	Email
hotmail.com	1638
yahoo.com	1616
gmail.com	1605
smith.com	42
williams.com	37

Check Unique Value in Dataset

In [80]:
```
data1.columns
```

Out[80]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price', 'year', 'com'],
      dtype='object')
```

In [81]:
```
data1["AM or PM"].nunique()
```

Out[81]:
```
2
```

In [82]:
```
data1["AM or PM"].unique()
```

Out[82]:
```
array(['PM', 'AM'], dtype=object)
```

In [83]:
```
data1['CC Provider'].nunique()
```

Out[83]:
```
10
```

In [84]:
```
data1['CC Provider'].unique()
```

Out[84]:
```
array(['JCB 16 digit', 'Mastercard', 'Discover',
       'Diners Club / Carte Blanche', 'VISA 16 digit', 'VISA 13 digit',
       'American Express', 'Voyager', 'JCB 15 digit', 'Maestro'],
      dtype=object)
```

In [85]:
```
data1['Language'].nunique()
```

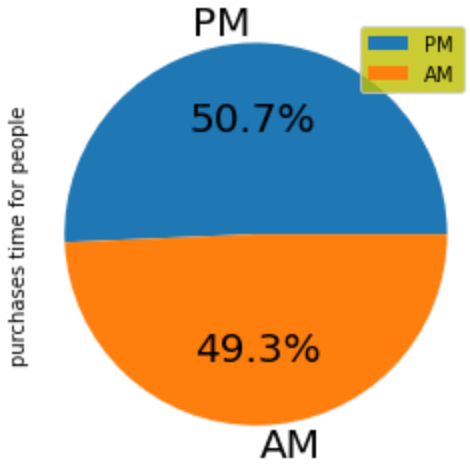
Out[85]:
```
9
```

In [86]:
```
data1['Language'].unique()
```

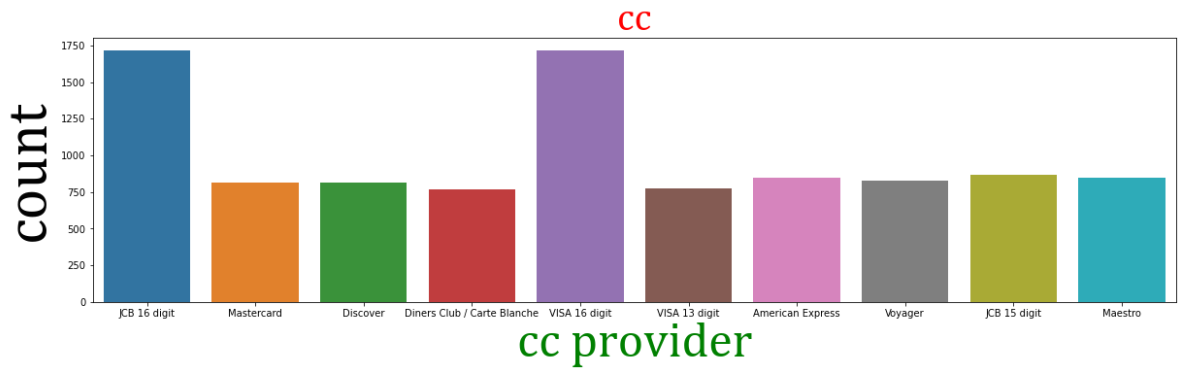
Out[86]:
```
array(['el', 'fr', 'de', 'es', 'ru', 'pt', 'zh', 'en', 'it'], dtype=object)
```

Data Visualization

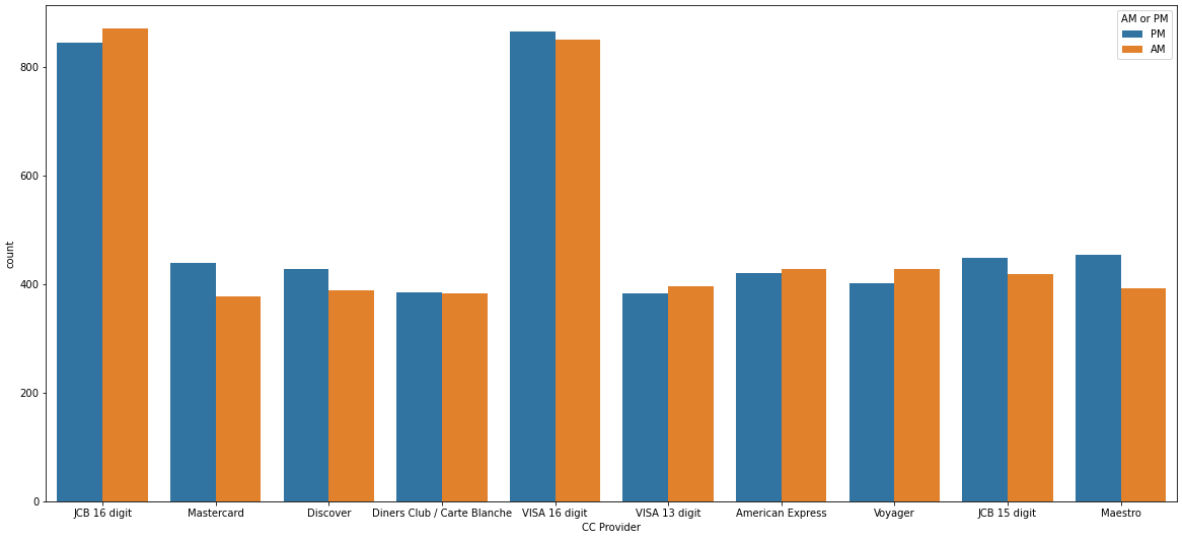
In [87]:
```
data1['AM or PM'].value_counts().plot(kind = "pie",ylabel = "purchases time fo
plt.legend(facecolor = "y")
plt.show()
```



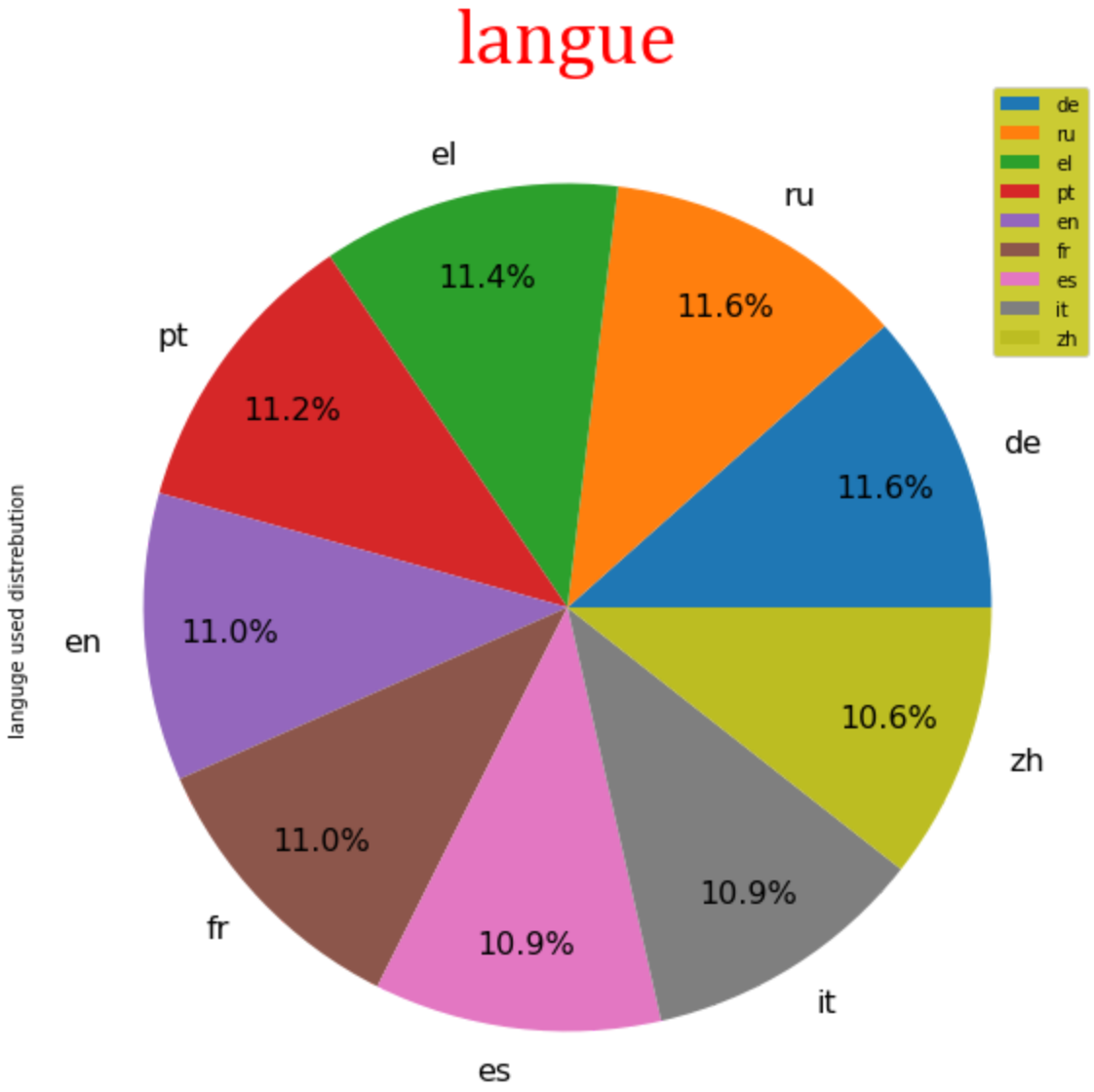
In [88]:
```
plt.figure(figsize = (20,5))
f1= {"family":"cambria","color":"r","size":40}
f2 = {"family":"cambria","color":"g","size":50}
f3  = {"family":"cambria","color":"k","size":60}
sns.countplot(x='CC Provider',data = data1)
plt.xlabel("cc provider",fontdict = f2)
plt.ylabel("count",fontdict = f3)
plt.title("cc",  fontdict = f1)
plt.show()
```



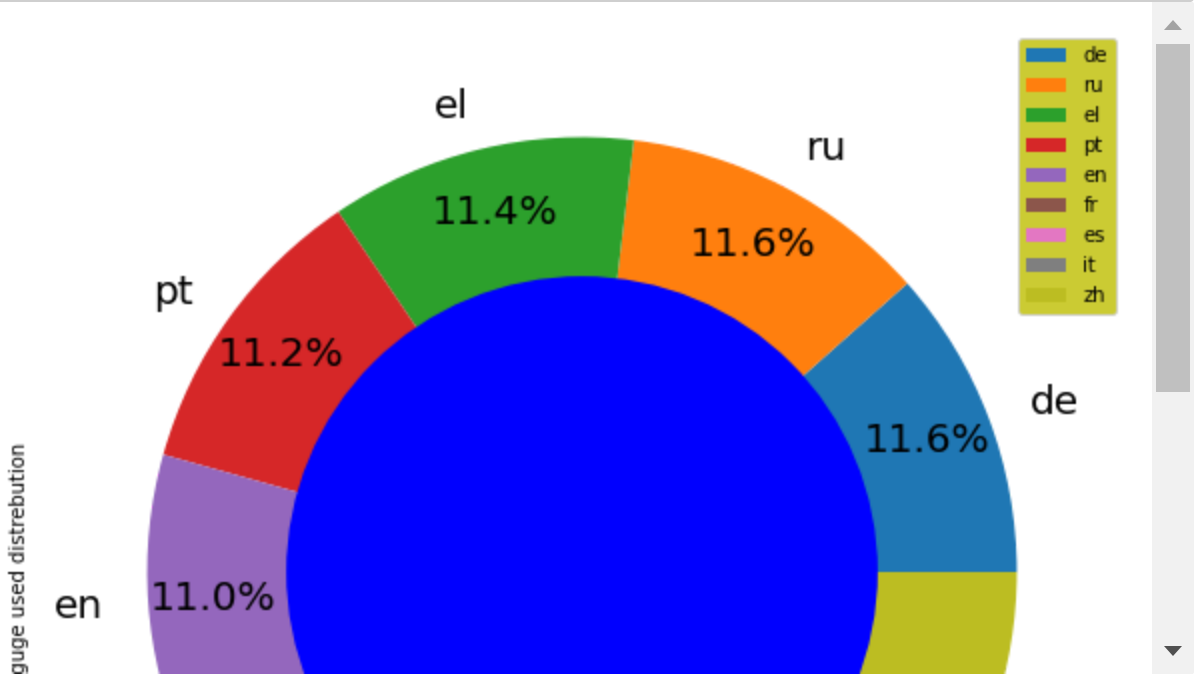
In [89]:
```
plt.figure(figsize = (20,9))
sns.countplot(x='CC Provider',data = data1,hue='AM or PM',saturation=0.75)
plt.show()
```



In [90]:
```
plt.figure(figsize = (10,10))
f1= {"family":"cambria","color":"r","size":40}
data1["Language"].value_counts().plot(kind = "pie",ylabel="languge used distre
plt.legend(facecolor = "y")
plt.title("langue",  fontdict = f1)
plt.show()
```



In [91]:
```
plt.figure(figsize = (10,10))
data1["Language"].value_counts().plot.pie(ylabel="languge used distrebution",
centre_circle = plt.Circle((0, 0), radius = 0.68, fc ='b')
plt.gca().add_artist(centre_circle)
plt.legend(facecolor = "y")
plt.show()
```



In [92]:
```
plt.figure(figsize = (10,10))
data1["Language"].value_counts().plot(kind = "pie", ylabel="languge used distr
centre_circle = plt.Circle((0, 0), radius = 0.75, fc ='white')
plt.gca().add_artist(centre_circle)
plt.legend(facecolor = "c")
plt.show()
```