Introduction

The dataset used for this projcet can be downloaded from the link: https://open.canada.ca/data/en/dataset/3fd71b16-34ce-2728-c6b5-5a865a6ee463/resource/6368b646-3c77-49bc-a4e2-d2f78a9acd9d

First, Importing Libraries

In [38]:
```
import pandas as pd
import math
```

In [39]:
```
import matplotlib.pyplot as plt
```

Loading the dataset into a dataframe

In [40]:
```
data = pd.read_csv('./CENSUS_2001_LBF_PARTICIPATION_0.CSV')
display(data)
```

	Region Code	Region	Data Note	Sex	Population 15 years and over by labour force activity - 20% Sample Data	In the labour force	Employed	Unemployed	Not in the labour force	Participation rate	...	Employmer rate
0	YK.WLR	Watson Lake Region (combined census subdivisio...	In some cases, the census subdivision boundary...	Total	900	685	555	130	215	76.1	...	50
1	YK.CRTG	Carcross/Tagish Region (combines the census su...	For smaller communities where the population d...	Total	340	210	165	45	130	60.9	...	33
2	YK.TA	Teslin Area (combined census subdivisions of T...	In some cases, the census subdivision boundary...	Total	215	145	125	25	70	67.4	...	50
3	YK.AKHWYN	Alaska Highway North (combined census subdivis...	For smaller communities where the population d...	Total	615	490	440	50	125	80.3	...	52
4	YK	Yukon	NaN	Total	22485	17945	15860	2085	4535	79.8	...	53
...	...	...	...	...	...	...	...	...	...	...	...	
85	YK.CC	Carcross	NaN	Female	60	35	30	0	20	58.3	...	66
86	YK.DB	Destruction Bay	NaN	Female	20	20	20	0	0	100.0	...	0
87	YK.SC	Stewart Crossing	NaN	Female	10	10	0	0	10	100.0	...	0
88	YK.IV	Ibex Valley	NaN	Female	110	95	80	10	20	86.4	...	60
89	YK.WHUO	Whitehorse Unorganized	NaN	Female	685	530	500	30	155	77.4	...	63

90 rows × 28 columns

Giving columns names

In [41]:
```
column_names = ['Rc', 'Rg', 'Dn', 'S', 'P_over15', 'Lf', 'E', 'U', 'notLf', 'Pr', 'Er', 'Ur', 'P_15to24',
```

In [42]:
```
data = pd.read_csv('./CENSUS_2001_LBF_PARTICIPATION_0.CSV', names = column_names, header = 1)
```

In [43]:
```
display(data)
```

	Rc	Rg	Dn	S	P_over15	Lf	E	U	notLf	Pr	...	Er1	Ur1	P_over25	Lf2	
0	YK.CRTG	Carcross/Tagish Region (combines the census su...	For smaller communities where the population d...	Total	340	210	165	45	130	60.9	...	33.3	40.0	300	185	
1	YK.TA	Teslin Area (combined census subdivisions of T...	In some cases, the census subdivision boundary...	Total	215	145	125	25	70	67.4	...	50.0	0.0	185	120	
2	YK.AKHWYN	Alaska Highway North (combined census subdivis...	For smaller communities where the population d...	Total	615	490	440	50	125	80.3	...	52.6	23.1	520	420	
3	YK	Yukon	NaN	Total	22485	17945	15860	2085	4535	79.8	...	53.1	21.5	18565	15295	1
4	YK.WL	Watson Lake	NaN	Total	670	535	465	70	140	79.9	...	55.0	16.7	570	475	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
84	YK.CC	Carcross	NaN	Female	60	35	30	0	20	58.3	...	66.7	100.0	45	30	
85	YK.DB	Destruction Bay	NaN	Female	20	20	20	0	0	100.0	...	0.0	0.0	15	20	
86	YK.SC	Stewart Crossing	NaN	Female	10	10	0	0	10	100.0	...	0.0	0.0	10	0	
87	YK.IV	Ibex Valley	NaN	Female	110	95	80	10	20	86.4	...	60.0	0.0	85	75	
88	YK.WHUO	Whitehorse Unorganized	NaN	Female	685	530	500	30	155	77.4	...	63.6	13.3	575	450	

89 rows × 28 columns

Calculate the shape of dataset

In [44]:
```
data.shape
```

Out[44]:
```
(89, 28)
```

In [45]:
```
num_rows = data.shape[0]
num_columns = data.shape[1]

print("Number of Rows: ", num_rows)
print("Number of Columns: ", num_columns)
```

```
Number of Rows:  89
Number of Columns:  28
```

List the column names

In [46]:
```
data.columns
```

Out[46]:
```
Index(['Rc', 'Rg', 'Dn', 'S', 'P_over15', 'Lf', 'E', 'U', 'notLf', 'Pr', 'Er',
       'Ur', 'P_15to24', 'Lf1', 'E1', 'U1', 'notLf1', 'Pr1', 'Er1', 'Ur1',
       'P_over25', 'Lf2', 'E2', 'U2', 'notLf2', 'Pr2', 'Er2', 'Ur2'],
      dtype='object')
```

Find the datatype of each column

In [47]:
```
data.dtypes
```

Out[47]:
```
Rc           object
Rg           object
Dn           object
S            object
P_over15      int64
Lf            int64
E             int64
U             int64
notLf         int64
Pr          float64
Er          float64
Ur          float64
P_15to24      int64
Lf1           int64
E1            int64
U1            int64
notLf1        int64
Pr1         float64
Er1         float64
Ur1         float64
P_over25      int64
Lf2           int64
E2            int64
U2            int64
notLf2        int64
Pr2         float64
Er2         float64
Ur2         float64
dtype: object
```

Explain what each column means (you may need to read documentation on the website from where you download the data.

Region Code (Rc): Comprises of a province and a region inside that province. For eg, a.b = in 'b' region within 'a' province. Region (Rg): The region inside a particular province. Data Note: Notes explaining special circumstance. Sex: Gender. Population 15 years and above/ 15 to 24 years/ over 25 years by labor force activity - 20% sample data: Population distribution of people in a certain age range by labor force activity. In the labor force: Number of people in the labor force. Employed: Number of people employed. Unemployed: Number of people unemployed. Not in any labor force: People not assiciated with labor force. Participation rate: Rate people who participated. Employement rate: Rate of people employed. Unemployement rate: Rate of people unemployed.

The columns: Sex, Regiona, Region Code, and Date Note are categorical because they represent a certain group (i.e., sex or region). The rest are quantitative.

For every column, compute the frequency distribution, relative frequencies, and percentage frequencies.

In [48]:

```
for column in data.columns:
    print("")
    print("----------------------------------")
    print("Value counts for column: ", column)
    print("----------------------------------")
    print("")
    if data[column].dtype == 'float64' or data[column].dtype == 'int64':
        fd = data[column].value_counts(bins = 10)
        print("FREQUENCY DISTRIBUTION FOR COLUMN: ", column)
        display(fd)
        rf = fd/num_rows
        print("----------------------------------------")
        print("RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN: ", column)
        print(rf)
        print("----------------------------------------")
        print("PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN: ", column)
        pf = rf*100
        print(pf)
        print("----------------------------------------")
   # elif data[column].dtype == 'float64':
    #    print("skipping value counts for column: ", column, "since floating point representations are app
     #   print(fd_)
    elif data[column].dtype == 'object':
        fd_Obj = data[column].value_counts()
        print("FREQUENCY DISTRIBUTION FOR COLUMN: ", column)
        print(fd_Obj)
        print("----------------------------------------")
        print("RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN: ", column)
        rf_Obj = fd_Obj/num_rows
        print(rf_Obj)
        print("----------------------------------------")
        print("PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN: ", column)
        pf_Obj = rf_Obj*100
        print(pf_Obj)
        print("----------------------------------------")
```

```
---------------------------------
Value counts for column:  Rc
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Rc
YK.CRTG      3
YK.TA        3
YK.WHUO      3
YK.IV        3
YK.SC        3
YK.DB        3
YK.CC        3
YK.YKUO      3
YK.MTL       3
YK.OC        3
YK.BC        3
YK.PC        3
YK.BL        3
YK.RR        3
YK.TG        3
YK.THMV      3
YK.UL        3
YK.DW        3
YK.MO        3
YK.HJ        3
YK.CM        3
YK.WH        3
YK.CC4       3
YK.TP        3
YK.TE        3
YK.FO        3
YK.WL        3
YK           3
YK.AKHWYN    3
YK.WLR       2
Name: Rc, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Rc
YK.CRTG      0.033708
YK.TA        0.033708
YK.WHUO      0.033708
YK.IV        0.033708
YK.SC        0.033708
YK.DB        0.033708
YK.CC        0.033708
YK.YKUO      0.033708
YK.MTL       0.033708
YK.OC        0.033708
YK.BC        0.033708
YK.PC        0.033708
YK.BL        0.033708
YK.RR        0.033708
YK.TG        0.033708
YK.THMV      0.033708
YK.UL        0.033708
YK.DW        0.033708
YK.MO        0.033708
YK.HJ        0.033708
YK.CM        0.033708
YK.WH        0.033708
YK.CC4       0.033708
YK.TP        0.033708
YK.TE        0.033708
YK.FO        0.033708
YK.WL        0.033708
YK           0.033708
YK.AKHWYN    0.033708
YK.WLR       0.022472
Name: Rc, dtype: float64
----------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Rc
YK.CRTG      3.370787
YK.TA        3.370787
YK.WHUO      3.370787
YK.IV        3.370787
YK.SC        3.370787
YK.DB        3.370787
YK.CC        3.370787
```

```
YK.YKUO      3.370787
YK.MTL       3.370787
YK.OC        3.370787
YK.BC        3.370787
YK.PC        3.370787
YK.BL        3.370787
YK.RR        3.370787
YK.TG        3.370787
YK.THMV      3.370787
YK.UL        3.370787
YK.DW        3.370787
YK.MO        3.370787
YK.HJ        3.370787
YK.CM        3.370787
YK.WH        3.370787
YK.CC4       3.370787
YK.TP        3.370787
YK.TE        3.370787
YK.FO        3.370787
YK.WL        3.370787
YK           3.370787
YK.AKHWYN    3.370787
YK.WLR       2.247191
Name: Rc, dtype: float64
----------------------------------------

---------------------------------
Value counts for column:  Rg
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Rg
Carcross/Tagish Region (combines the census subdivisions of: Carcross; Carcross 4; and Tagish)
3
Teslin Area (combined census subdivisions of Teslin and Teslin Post 13)
3
Whitehorse Unorganized
3
Ibex Valley
3
Stewart Crossing
3
Destruction Bay
3
Carcross
3
Yukon Unorganized
3
Mount Lorne
3
Old Crow
3
Beaver Creek
3
Pelly Crossing
3
Burwash Landing
3
Ross River
3
Tagish
3
Two and One Half Mile Village
3
Upper Liard
3
Dawson
3
Mayo
3
Haines Junction
3
Carmacks
3
Whitehorse
3
Carcross 4
3
Teslin Post
```

```
3
Teslin
3
Faro
3
Watson Lake
3
Yukon
3
Alaska Highway North (combined census subdivisions of Beaver Creek; Burwash Landing; Champagne Landing; De
struction Bay; and Haines Junction)    3
Watson Lake Region (combined census subdivisions of: Two and One-Half Mile Villag;, Two Mile Village; Uppe
r Liard; and Watson Lake)              2
Name: Rg, dtype: int64
----------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Rg
Carcross/Tagish Region (combines the census subdivisions of: Carcross; Carcross 4; and Tagish)
0.033708
Teslin Area (combined census subdivisions of Teslin and Teslin Post 13)
0.033708
Whitehorse Unorganized
0.033708
Ibex Valley
0.033708
Stewart Crossing
0.033708
Destruction Bay
0.033708
Carcross
0.033708
Yukon Unorganized
0.033708
Mount Lorne
0.033708
Old Crow
0.033708
Beaver Creek
0.033708
Pelly Crossing
0.033708
Burwash Landing
0.033708
Ross River
0.033708
Tagish
0.033708
Two and One Half Mile Village
0.033708
Upper Liard
0.033708
Dawson
0.033708
Mayo
0.033708
Haines Junction
0.033708
Carmacks
0.033708
Whitehorse
0.033708
Carcross 4
0.033708
Teslin Post
0.033708
Teslin
0.033708
Faro
0.033708
Watson Lake
0.033708
Yukon
0.033708
Alaska Highway North (combined census subdivisions of Beaver Creek; Burwash Landing; Champagne Landing; De
struction Bay; and Haines Junction)    0.033708
Watson Lake Region (combined census subdivisions of: Two and One-Half Mile Villag;, Two Mile Village; Uppe
r Liard; and Watson Lake)              0.022472
Name: Rg, dtype: float64
----------------------------------------
```

PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN: Rg
Carcross/Tagish Region (combines the census subdivisions of: Carcross; Carcross 4; and Tagish)
3.370787
Teslin Area (combined census subdivisions of Teslin and Teslin Post 13)
3.370787
Whitehorse Unorganized
3.370787
Ibex Valley
3.370787
Stewart Crossing
3.370787
Destruction Bay
3.370787
Carcross
3.370787
Yukon Unorganized
3.370787
Mount Lorne
3.370787
Old Crow
3.370787
Beaver Creek
3.370787
Pelly Crossing
3.370787
Burwash Landing
3.370787
Ross River
3.370787
Tagish
3.370787
Two and One Half Mile Village
3.370787
Upper Liard
3.370787
Dawson
3.370787
Mayo
3.370787
Haines Junction
3.370787
Carmacks
3.370787
Whitehorse
3.370787
Carcross 4
3.370787
Teslin Post
3.370787
Teslin
3.370787
Faro
3.370787
Watson Lake
3.370787
Yukon
3.370787
Alaska Highway North (combined census subdivisions of Beaver Creek; Burwash Landing; Champagne Landing; De
struction Bay; and Haines Junction)   3.370787
Watson Lake Region (combined census subdivisions of: Two and One-Half Mile Villag;, Two Mile Village; Uppe
r Liard; and Watson Lake)   2.247191
Name: Rg, dtype: float64

Value counts for column: Dn

FREQUENCY DISTRIBUTION FOR COLUMN: Dn
For smaller communities where the population does not exceed 250 people, income statistics are suppressed.
For this reason, the census subdivisions of Carcross (6001048), Carcross 4 (6001008) and Tagish (6001036)
were combined to provide income statistics for this geographic region.
1
In some cases, the census subdivision boundary does not match the municipal boundary. To provide a more re
presentative estimate of demography in this geographic area, the census subdivisions of Teslin (6001006) a
nd Teslin Post 13 (60010006) were combined.
1
For smaller communities where the population does not exceed 250 people, income statistics are suppressed.

```
For this reason, the census subdivisions of Beaver Creek (6001042), Burwash Landing (6001039), Champagne L
anding (6001038), Destruction Bay (6001048) and Haines Junction (6001018) were combined to provide income
statistics for this geographic region.    1
Name: Dn, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Dn
For smaller communities where the population does not exceed 250 people, income statistics are suppressed.
For this reason, the census subdivisions of Carcross (6001048), Carcross 4 (6001008) and Tagish (6001036)
were combined to provide income statistics for this geographic region.
0.011236
In some cases, the census subdivision boundary does not match the municipal boundary. To provide a more re
presentative estimate of demography in this geographic area, the census subdivisions of Teslin (6001006) a
nd Teslin Post 13 (60010006) were combined.
0.011236
For smaller communities where the population does not exceed 250 people, income statistics are suppressed.
For this reason, the census subdivisions of Beaver Creek (6001042), Burwash Landing (6001039), Champagne L
anding (6001038), Destruction Bay (6001048) and Haines Junction (6001018) were combined to provide income
statistics for this geographic region.    0.011236
Name: Dn, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Dn
For smaller communities where the population does not exceed 250 people, income statistics are suppressed.
For this reason, the census subdivisions of Carcross (6001048), Carcross 4 (6001008) and Tagish (6001036)
were combined to provide income statistics for this geographic region.
1.123596
In some cases, the census subdivision boundary does not match the municipal boundary. To provide a more re
presentative estimate of demography in this geographic area, the census subdivisions of Teslin (6001006) a
nd Teslin Post 13 (60010006) were combined.
1.123596
For smaller communities where the population does not exceed 250 people, income statistics are suppressed.
For this reason, the census subdivisions of Beaver Creek (6001042), Burwash Landing (6001039), Champagne L
anding (6001038), Destruction Bay (6001048) and Haines Junction (6001018) were combined to provide income
statistics for this geographic region.    1.123596
Name: Dn, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  S
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  S
Male      30
Female    30
Total     29
Name: S, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  S
Male      0.337079
Female    0.337079
Total     0.325843
Name: S, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  S
Male      33.707865
Female    33.707865
Total     32.584270
Name: S, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  P_over15
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  P_over15
(-12.475999999999999, 2257.5]    83
(6752.5, 9000.0]                  2
(9000.0, 11247.5]                 1
(11247.5, 13495.0]                1
(13495.0, 15742.5]                1
(20237.5, 22485.0]                1
(2257.5, 4505.0]                  0
(4505.0, 6752.5]                  0
(15742.5, 17990.0]                0
(17990.0, 20237.5]                0
Name: P_over15, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  P_over15
(-12.475999999999999, 2257.5]    0.932584
(6752.5, 9000.0]                 0.022472
(9000.0, 11247.5]                0.011236
(11247.5, 13495.0]               0.011236
(13495.0, 15742.5]               0.011236
(20237.5, 22485.0]               0.011236
(2257.5, 4505.0]                 0.000000
(4505.0, 6752.5]                 0.000000
(15742.5, 17990.0]               0.000000
(17990.0, 20237.5]               0.000000
Name: P_over15, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  P_over15
(-12.475999999999999, 2257.5]    93.258427
(6752.5, 9000.0]                  2.247191
(9000.0, 11247.5]                 1.123596
(11247.5, 13495.0]                1.123596
(13495.0, 15742.5]                1.123596
(20237.5, 22485.0]                1.123596
(2257.5, 4505.0]                  0.000000
(4505.0, 6752.5]                  0.000000
(15742.5, 17990.0]                0.000000
(17990.0, 20237.5]                0.000000
Name: P_over15, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Lf
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Lf
(-7.936, 1803.5]      83
(5390.5, 7184.0]       2
(7184.0, 8977.5]       1
(8977.5, 10771.0]      1
(10771.0, 12564.5]     1
(16151.5, 17945.0]     1
(1803.5, 3597.0]       0
(3597.0, 5390.5]       0
(12564.5, 14358.0]     0
(14358.0, 16151.5]     0
Name: Lf, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Lf
(-7.936, 1803.5]      0.932584
(5390.5, 7184.0]      0.022472
(7184.0, 8977.5]      0.011236
(8977.5, 10771.0]     0.011236
(10771.0, 12564.5]    0.011236
(16151.5, 17945.0]    0.011236
(1803.5, 3597.0]      0.000000
(3597.0, 5390.5]      0.000000
(12564.5, 14358.0]    0.000000
(14358.0, 16151.5]    0.000000
Name: Lf, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Lf
(-7.936, 1803.5]      93.258427
(5390.5, 7184.0]       2.247191
(7184.0, 8977.5]       1.123596
(8977.5, 10771.0]      1.123596
(10771.0, 12564.5]     1.123596
(16151.5, 17945.0]     1.123596
(1803.5, 3597.0]       0.000000
(3597.0, 5390.5]       0.000000
(12564.5, 14358.0]     0.000000
(14358.0, 16151.5]     0.000000
Name: Lf, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  E
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  E
```

```
(-15.860999999999999, 1586.0]    83
(4758.0, 6344.0]                  2
(6344.0, 7930.0]                  1
(7930.0, 9516.0]                  1
(9516.0, 11102.0]                 1
(14274.0, 15860.0]                1
(1586.0, 3172.0]                  0
(3172.0, 4758.0]                  0
(11102.0, 12688.0]                0
(12688.0, 14274.0]                0
Name: E, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  E
(-15.860999999999999, 1586.0]    0.932584
(4758.0, 6344.0]                 0.022472
(6344.0, 7930.0]                 0.011236
(7930.0, 9516.0]                 0.011236
(9516.0, 11102.0]                0.011236
(14274.0, 15860.0]               0.011236
(1586.0, 3172.0]                 0.000000
(3172.0, 4758.0]                 0.000000
(11102.0, 12688.0]               0.000000
(12688.0, 14274.0]               0.000000
Name: E, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  E
(-15.860999999999999, 1586.0]    93.258427
(4758.0, 6344.0]                  2.247191
(6344.0, 7930.0]                  1.123596
(7930.0, 9516.0]                  1.123596
(9516.0, 11102.0]                 1.123596
(14274.0, 15860.0]                1.123596
(1586.0, 3172.0]                  0.000000
(3172.0, 4758.0]                  0.000000
(11102.0, 12688.0]                0.000000
(12688.0, 14274.0]                0.000000
Name: E, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  U
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  U
(-2.086, 208.5]     83
(625.5, 834.0]       2
(417.0, 625.5]       1
(1042.5, 1251.0]     1
(1251.0, 1459.5]     1
(1876.5, 2085.0]     1
(208.5, 417.0]       0
(834.0, 1042.5]      0
(1459.5, 1668.0]     0
(1668.0, 1876.5]     0
Name: U, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  U
(-2.086, 208.5]     0.932584
(625.5, 834.0]      0.022472
(417.0, 625.5]      0.011236
(1042.5, 1251.0]    0.011236
(1251.0, 1459.5]    0.011236
(1876.5, 2085.0]    0.011236
(208.5, 417.0]      0.000000
(834.0, 1042.5]     0.000000
(1459.5, 1668.0]    0.000000
(1668.0, 1876.5]    0.000000
Name: U, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  U
(-2.086, 208.5]     93.258427
(625.5, 834.0]       2.247191
(417.0, 625.5]       1.123596
(1042.5, 1251.0]     1.123596
(1251.0, 1459.5]     1.123596
(1876.5, 2085.0]     1.123596
(208.5, 417.0]       0.000000
(834.0, 1042.5]      0.000000
(1459.5, 1668.0]     0.000000
(1668.0, 1876.5]     0.000000
Name: U, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  notLf
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  notLf
(-4.5360000000000005, 453.5]    83
(907.0, 1360.5]                  1
(1360.5, 1814.0]                 1
(1814.0, 2267.5]                 1
(2267.5, 2721.0]                 1
(2721.0, 3174.5]                 1
(4081.5, 4535.0]                 1
(453.5, 907.0]                   0
(3174.5, 3628.0]                 0
(3628.0, 4081.5]                 0
Name: notLf, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  notLf
(-4.5360000000000005, 453.5]    0.932584
(907.0, 1360.5]                 0.011236
(1360.5, 1814.0]                0.011236
(1814.0, 2267.5]                0.011236
(2267.5, 2721.0]                0.011236
(2721.0, 3174.5]                0.011236
(4081.5, 4535.0]                0.011236
(453.5, 907.0]                  0.000000
(3174.5, 3628.0]                0.000000
(3628.0, 4081.5]                0.000000
Name: notLf, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  notLf
(-4.5360000000000005, 453.5]    93.258427
(907.0, 1360.5]                  1.123596
(1360.5, 1814.0]                 1.123596
(1814.0, 2267.5]                 1.123596
(2267.5, 2721.0]                 1.123596
(2721.0, 3174.5]                 1.123596
(4081.5, 4535.0]                 1.123596
(453.5, 907.0]                   0.000000
(3174.5, 3628.0]                 0.000000
(3628.0, 4081.5]                 0.000000
Name: notLf, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Pr
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Pr
```

```
(75.0, 80.0]                  20
(80.0, 85.0]                  19
(70.0, 75.0]                  17
(65.0, 70.0]                  12
(55.0, 60.0]                   7
(60.0, 65.0]                   6
(85.0, 90.0]                   4
(95.0, 100.0]                  3
(49.949000000000005, 55.0]     1
(90.0, 95.0]                   0
Name: Pr, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Pr
(75.0, 80.0]                  0.224719
(80.0, 85.0]                  0.213483
(70.0, 75.0]                  0.191011
(65.0, 70.0]                  0.134831
(55.0, 60.0]                  0.078652
(60.0, 65.0]                  0.067416
(85.0, 90.0]                  0.044944
(95.0, 100.0]                 0.033708
(49.949000000000005, 55.0]    0.011236
(90.0, 95.0]                  0.000000
Name: Pr, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Pr
(75.0, 80.0]                  22.471910
(80.0, 85.0]                  21.348315
(70.0, 75.0]                  19.101124
(65.0, 70.0]                  13.483146
(55.0, 60.0]                   7.865169
(60.0, 65.0]                   6.741573
(85.0, 90.0]                   4.494382
(95.0, 100.0]                  3.370787
(49.949000000000005, 55.0]     1.123596
(90.0, 95.0]                   0.000000
Name: Pr, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  Er
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Er
(70.0, 80.0]      25
(40.0, 50.0]      19
(50.0, 60.0]      18
(60.0, 70.0]      18
(30.0, 40.0]       3
(80.0, 90.0]       3
(-0.101, 10.0]     2
(90.0, 100.0]      1
(10.0, 20.0]       0
(20.0, 30.0]       0
Name: Er, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Er
(70.0, 80.0]      0.280899
(40.0, 50.0]      0.213483
(50.0, 60.0]      0.202247
(60.0, 70.0]      0.202247
(30.0, 40.0]      0.033708
(80.0, 90.0]      0.033708
(-0.101, 10.0]    0.022472
(90.0, 100.0]     0.011236
(10.0, 20.0]      0.000000
(20.0, 30.0]      0.000000
Name: Er, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Er
(70.0, 80.0]      28.089888
(40.0, 50.0]      21.348315
(50.0, 60.0]      20.224719
(60.0, 70.0]      20.224719
(30.0, 40.0]       3.370787
(80.0, 90.0]       3.370787
(-0.101, 10.0]     2.247191
(90.0, 100.0]      1.123596
(10.0, 20.0]       0.000000
(20.0, 30.0]       0.000000
Name: Er, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Ur
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Ur
(6.67, 13.34]      23
(13.34, 20.01]     22
(-0.0677, 6.67]    15
(26.68, 33.35]     10
(20.01, 26.68]      8
(33.35, 40.02]      5
(46.69, 53.36]      4
(60.03, 66.7]       2
(40.02, 46.69]      0
(53.36, 60.03]      0
Name: Ur, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Ur
(6.67, 13.34]      0.258427
(13.34, 20.01]     0.247191
(-0.0677, 6.67]    0.168539
(26.68, 33.35]     0.112360
(20.01, 26.68]     0.089888
(33.35, 40.02]     0.056180
(46.69, 53.36]     0.044944
(60.03, 66.7]      0.022472
(40.02, 46.69]     0.000000
(53.36, 60.03]     0.000000
Name: Ur, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Ur
(6.67, 13.34]      25.842697
(13.34, 20.01]     24.719101
(-0.0677, 6.67]    16.853933
(26.68, 33.35]     11.235955
(20.01, 26.68]      8.988764
(33.35, 40.02]      5.617978
(46.69, 53.36]      4.494382
(60.03, 66.7]       2.247191
(40.02, 46.69]      0.000000
(53.36, 60.03]      0.000000
Name: Ur, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  P_15to24
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  P_15to24
```

```
(-3.921, 392.0]     83
(1176.0, 1568.0]     2
(1568.0, 1960.0]     1
(1960.0, 2352.0]     1
(2744.0, 3136.0]     1
(3528.0, 3920.0]     1
(392.0, 784.0]       0
(784.0, 1176.0]      0
(2352.0, 2744.0]     0
(3136.0, 3528.0]     0
Name: P_15to24, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  P_15to24
(-3.921, 392.0]     0.932584
(1176.0, 1568.0]    0.022472
(1568.0, 1960.0]    0.011236
(1960.0, 2352.0]    0.011236
(2744.0, 3136.0]    0.011236
(3528.0, 3920.0]    0.011236
(392.0, 784.0]      0.000000
(784.0, 1176.0]     0.000000
(2352.0, 2744.0]    0.000000
(3136.0, 3528.0]    0.000000
Name: P_15to24, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  P_15to24
(-3.921, 392.0]     93.258427
(1176.0, 1568.0]     2.247191
(1568.0, 1960.0]     1.123596
(1960.0, 2352.0]     1.123596
(2744.0, 3136.0]     1.123596
(3528.0, 3920.0]     1.123596
(392.0, 784.0]       0.000000
(784.0, 1176.0]      0.000000
(2352.0, 2744.0]     0.000000
(3136.0, 3528.0]     0.000000
Name: P_15to24, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  Lf1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Lf1
(-2.6559999999999997, 265.5]    83
(796.5, 1062.0]                  2
(1062.0, 1327.5]                 1
(1327.5, 1593.0]                 1
(1858.5, 2124.0]                 1
(2389.5, 2655.0]                 1
(265.5, 531.0]                   0
(531.0, 796.5]                   0
(1593.0, 1858.5]                 0
(2124.0, 2389.5]                 0
Name: Lf1, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Lf1
(-2.6559999999999997, 265.5]    0.932584
(796.5, 1062.0]                 0.022472
(1062.0, 1327.5]                0.011236
(1327.5, 1593.0]                0.011236
(1858.5, 2124.0]                0.011236
(2389.5, 2655.0]                0.011236
(265.5, 531.0]                  0.000000
(531.0, 796.5]                  0.000000
(1593.0, 1858.5]                0.000000
(2124.0, 2389.5]                0.000000
Name: Lf1, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Lf1
(-2.6559999999999997, 265.5]    93.258427
(796.5, 1062.0]                  2.247191
(1062.0, 1327.5]                 1.123596
(1327.5, 1593.0]                 1.123596
(1858.5, 2124.0]                 1.123596
(2389.5, 2655.0]                 1.123596
(265.5, 531.0]                   0.000000
(531.0, 796.5]                   0.000000
(1593.0, 1858.5]                 0.000000
(2124.0, 2389.5]                 0.000000
Name: Lf1, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  E1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  E1
(-2.081, 208.0]     83
(624.0, 832.0]       2
(832.0, 1040.0]      1
(1040.0, 1248.0]     1
(1456.0, 1664.0]     1
(1872.0, 2080.0]     1
(208.0, 416.0]       0
(416.0, 624.0]       0
(1248.0, 1456.0]     0
(1664.0, 1872.0]     0
Name: E1, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  E1
(-2.081, 208.0]     0.932584
(624.0, 832.0]      0.022472
(832.0, 1040.0]     0.011236
(1040.0, 1248.0]    0.011236
(1456.0, 1664.0]    0.011236
(1872.0, 2080.0]    0.011236
(208.0, 416.0]      0.000000
(416.0, 624.0]      0.000000
(1248.0, 1456.0]    0.000000
(1664.0, 1872.0]    0.000000
Name: E1, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  E1
(-2.081, 208.0]     93.258427
(624.0, 832.0]       2.247191
(832.0, 1040.0]      1.123596
(1040.0, 1248.0]     1.123596
(1456.0, 1664.0]     1.123596
(1872.0, 2080.0]     1.123596
(208.0, 416.0]       0.000000
(416.0, 624.0]       0.000000
(1248.0, 1456.0]     0.000000
(1664.0, 1872.0]     0.000000
Name: E1, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  U1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  U1
```

```
(-0.571, 57.0]    83
(114.0, 171.0]     1
(171.0, 228.0]     1
(228.0, 285.0]     1
(342.0, 399.0]     1
(399.0, 456.0]     1
(513.0, 570.0]     1
(57.0, 114.0]      0
(285.0, 342.0]     0
(456.0, 513.0]     0
Name: U1, dtype: int64
----------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  U1
(-0.571, 57.0]    0.932584
(114.0, 171.0]    0.011236
(171.0, 228.0]    0.011236
(228.0, 285.0]    0.011236
(342.0, 399.0]    0.011236
(399.0, 456.0]    0.011236
(513.0, 570.0]    0.011236
(57.0, 114.0]     0.000000
(285.0, 342.0]    0.000000
(456.0, 513.0]    0.000000
Name: U1, dtype: float64
----------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  U1
(-0.571, 57.0]    93.258427
(114.0, 171.0]     1.123596
(171.0, 228.0]     1.123596
(228.0, 285.0]     1.123596
(342.0, 399.0]     1.123596
(399.0, 456.0]     1.123596
(513.0, 570.0]     1.123596
(57.0, 114.0]      0.000000
(285.0, 342.0]     0.000000
(456.0, 513.0]     0.000000
Name: U1, dtype: float64
----------------------------------------

---------------------------------
Value counts for column:  notLf1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  notLf1
(-1.2659999999999998, 126.5]    83
(379.5, 506.0]                   2
(506.0, 632.5]                   1
(632.5, 759.0]                   1
(759.0, 885.5]                   1
(1138.5, 1265.0]                 1
(126.5, 253.0]                   0
(253.0, 379.5]                   0
(885.5, 1012.0]                  0
(1012.0, 1138.5]                 0
Name: notLf1, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  notLf1
(-1.2659999999999998, 126.5]    0.932584
(379.5, 506.0]                  0.022472
(506.0, 632.5]                  0.011236
(632.5, 759.0]                  0.011236
(759.0, 885.5]                  0.011236
(1138.5, 1265.0]                0.011236
(126.5, 253.0]                  0.000000
(253.0, 379.5]                  0.000000
(885.5, 1012.0]                 0.000000
(1012.0, 1138.5]                0.000000
Name: notLf1, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  notLf1
(-1.2659999999999998, 126.5]    93.258427
(379.5, 506.0]                   2.247191
(506.0, 632.5]                   1.123596
(632.5, 759.0]                   1.123596
(759.0, 885.5]                   1.123596
(1138.5, 1265.0]                 1.123596
(126.5, 253.0]                   0.000000
(253.0, 379.5]                   0.000000
(885.5, 1012.0]                  0.000000
(1012.0, 1138.5]                 0.000000
Name: notLf1, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Pr1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Pr1
(60.0, 70.0]      29
(-0.101, 10.0]    19
(50.0, 60.0]      11
(70.0, 80.0]      11
(90.0, 100.0]      8
(40.0, 50.0]       7
(30.0, 40.0]       2
(80.0, 90.0]       2
(10.0, 20.0]       0
(20.0, 30.0]       0
Name: Pr1, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Pr1
(60.0, 70.0]      0.325843
(-0.101, 10.0]    0.213483
(50.0, 60.0]      0.123596
(70.0, 80.0]      0.123596
(90.0, 100.0]     0.089888
(40.0, 50.0]      0.078652
(30.0, 40.0]      0.022472
(80.0, 90.0]      0.022472
(10.0, 20.0]      0.000000
(20.0, 30.0]      0.000000
Name: Pr1, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Pr1
(60.0, 70.0]      32.584270
(-0.101, 10.0]    21.348315
(50.0, 60.0]      12.359551
(70.0, 80.0]      12.359551
(90.0, 100.0]      8.988764
(40.0, 50.0]       7.865169
(30.0, 40.0]       2.247191
(80.0, 90.0]       2.247191
(10.0, 20.0]       0.000000
(20.0, 30.0]       0.000000
Name: Pr1, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Er1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Er1
```

```
(-0.101, 10.0]    27
(40.0, 50.0]      16
(30.0, 40.0]      14
(50.0, 60.0]      14
(60.0, 70.0]      11
(90.0, 100.0]      5
(20.0, 30.0]       1
(70.0, 80.0]       1
(10.0, 20.0]       0
(80.0, 90.0]       0
Name: Er1, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Er1
(-0.101, 10.0]    0.303371
(40.0, 50.0]      0.179775
(30.0, 40.0]      0.157303
(50.0, 60.0]      0.157303
(60.0, 70.0]      0.123596
(90.0, 100.0]     0.056180
(20.0, 30.0]      0.011236
(70.0, 80.0]      0.011236
(10.0, 20.0]      0.000000
(80.0, 90.0]      0.000000
Name: Er1, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Er1
(-0.101, 10.0]    30.337079
(40.0, 50.0]      17.977528
(30.0, 40.0]      15.730337
(50.0, 60.0]      15.730337
(60.0, 70.0]      12.359551
(90.0, 100.0]      5.617978
(20.0, 30.0]       1.123596
(70.0, 80.0]       1.123596
(10.0, 20.0]       0.000000
(80.0, 90.0]       0.000000
Name: Er1, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  Ur1
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Ur1
(-0.101, 10.0]    47
(10.0, 20.0]       9
(20.0, 30.0]       9
(90.0, 100.0]      7
(30.0, 40.0]       6
(60.0, 70.0]       6
(40.0, 50.0]       4
(50.0, 60.0]       1
(70.0, 80.0]       0
(80.0, 90.0]       0
Name: Ur1, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Ur1
(-0.101, 10.0]    0.528090
(10.0, 20.0]      0.101124
(20.0, 30.0]      0.101124
(90.0, 100.0]     0.078652
(30.0, 40.0]      0.067416
(60.0, 70.0]      0.067416
(40.0, 50.0]      0.044944
(50.0, 60.0]      0.011236
(70.0, 80.0]      0.000000
(80.0, 90.0]      0.000000
Name: Ur1, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Ur1
(-0.101, 10.0]    52.808989
(10.0, 20.0]      10.112360
(20.0, 30.0]      10.112360
(90.0, 100.0]      7.865169
(30.0, 40.0]       6.741573
(60.0, 70.0]       6.741573
(40.0, 50.0]       4.494382
(50.0, 60.0]       1.123596
(70.0, 80.0]       0.000000
(80.0, 90.0]       0.000000
Name: Ur1, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  P_over25
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  P_over25
(-8.556, 1865.5]      83
(5576.5, 7432.0]       2
(7432.0, 9287.5]       1
(9287.5, 11143.0]      1
(11143.0, 12998.5]     1
(16709.5, 18565.0]     1
(1865.5, 3721.0]       0
(3721.0, 5576.5]       0
(12998.5, 14854.0]     0
(14854.0, 16709.5]     0
Name: P_over25, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  P_over25
(-8.556, 1865.5]      0.932584
(5576.5, 7432.0]      0.022472
(7432.0, 9287.5]      0.011236
(9287.5, 11143.0]     0.011236
(11143.0, 12998.5]    0.011236
(16709.5, 18565.0]    0.011236
(1865.5, 3721.0]      0.000000
(3721.0, 5576.5]      0.000000
(12998.5, 14854.0]    0.000000
(14854.0, 16709.5]    0.000000
Name: P_over25, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  P_over25
(-8.556, 1865.5]      93.258427
(5576.5, 7432.0]       2.247191
(7432.0, 9287.5]       1.123596
(9287.5, 11143.0]      1.123596
(11143.0, 12998.5]     1.123596
(16709.5, 18565.0]     1.123596
(1865.5, 3721.0]       0.000000
(3721.0, 5576.5]       0.000000
(12998.5, 14854.0]     0.000000
(14854.0, 16709.5]     0.000000
Name: P_over25, dtype: float64
---------------------------------------

---------------------------------
Value counts for column:  Lf2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Lf2
```

```
(-15.296, 1529.5]     83
(4588.5, 6118.0]       2
(6118.0, 7647.5]       1
(7647.5, 9177.0]       1
(9177.0, 10706.5]      1
(13765.5, 15295.0]     1
(1529.5, 3059.0]       0
(3059.0, 4588.5]       0
(10706.5, 12236.0]     0
(12236.0, 13765.5]     0
Name: Lf2, dtype: int64
----------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Lf2
(-15.296, 1529.5]     0.932584
(4588.5, 6118.0]      0.022472
(6118.0, 7647.5]      0.011236
(7647.5, 9177.0]      0.011236
(9177.0, 10706.5]     0.011236
(13765.5, 15295.0]    0.011236
(1529.5, 3059.0]      0.000000
(3059.0, 4588.5]      0.000000
(10706.5, 12236.0]    0.000000
(12236.0, 13765.5]    0.000000
Name: Lf2, dtype: float64
----------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Lf2
(-15.296, 1529.5]     93.258427
(4588.5, 6118.0]       2.247191
(6118.0, 7647.5]       1.123596
(7647.5, 9177.0]       1.123596
(9177.0, 10706.5]      1.123596
(13765.5, 15295.0]     1.123596
(1529.5, 3059.0]       0.000000
(3059.0, 4588.5]       0.000000
(10706.5, 12236.0]     0.000000
(12236.0, 13765.5]     0.000000
Name: Lf2, dtype: float64
----------------------------------------

---------------------------------
Value counts for column:  E2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  E2
(-13.780999999999999, 1378.0]    83
(4134.0, 5512.0]                  2
(5512.0, 6890.0]                  1
(6890.0, 8268.0]                  1
(8268.0, 9646.0]                  1
(12402.0, 13780.0]                1
(1378.0, 2756.0]                  0
(2756.0, 4134.0]                  0
(9646.0, 11024.0]                 0
(11024.0, 12402.0]                0
Name: E2, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  E2
(-13.780999999999999, 1378.0]    0.932584
(4134.0, 5512.0]                 0.022472
(5512.0, 6890.0]                 0.011236
(6890.0, 8268.0]                 0.011236
(8268.0, 9646.0]                 0.011236
(12402.0, 13780.0]               0.011236
(1378.0, 2756.0]                 0.000000
(2756.0, 4134.0]                 0.000000
(9646.0, 11024.0]                0.000000
(11024.0, 12402.0]               0.000000
Name: E2, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  E2
(-13.780999999999999, 1378.0]    93.258427
(4134.0, 5512.0]                  2.247191
(5512.0, 6890.0]                  1.123596
(6890.0, 8268.0]                  1.123596
(8268.0, 9646.0]                  1.123596
(12402.0, 13780.0]                1.123596
(1378.0, 2756.0]                  0.000000
(2756.0, 4134.0]                  0.000000
(9646.0, 11024.0]                 0.000000
(11024.0, 12402.0]                0.000000
Name: E2, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  U2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  U2
(-1.5159999999999998, 151.5]    83
(454.5, 606.0]                   2
(303.0, 454.5]                   1
(757.5, 909.0]                   1
(909.0, 1060.5]                  1
(1363.5, 1515.0]                 1
(151.5, 303.0]                   0
(606.0, 757.5]                   0
(1060.5, 1212.0]                 0
(1212.0, 1363.5]                 0
Name: U2, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  U2
(-1.5159999999999998, 151.5]    0.932584
(454.5, 606.0]                  0.022472
(303.0, 454.5]                  0.011236
(757.5, 909.0]                  0.011236
(909.0, 1060.5]                 0.011236
(1363.5, 1515.0]                0.011236
(151.5, 303.0]                  0.000000
(606.0, 757.5]                  0.000000
(1060.5, 1212.0]                0.000000
(1212.0, 1363.5]                0.000000
Name: U2, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  U2
(-1.5159999999999998, 151.5]    93.258427
(454.5, 606.0]                   2.247191
(303.0, 454.5]                   1.123596
(757.5, 909.0]                   1.123596
(909.0, 1060.5]                  1.123596
(1363.5, 1515.0]                 1.123596
(151.5, 303.0]                   0.000000
(606.0, 757.5]                   0.000000
(1060.5, 1212.0]                 0.000000
(1212.0, 1363.5]                 0.000000
Name: U2, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  notLf2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  notLf2
```

```
(-3.271, 327.0]     83
(654.0, 981.0]       1
(981.0, 1308.0]      1
(1308.0, 1635.0]     1
(1635.0, 1962.0]     1
(1962.0, 2289.0]     1
(2943.0, 3270.0]     1
(327.0, 654.0]       0
(2289.0, 2616.0]     0
(2616.0, 2943.0]     0
Name: notLf2, dtype: int64
----------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  notLf2
(-3.271, 327.0]     0.932584
(654.0, 981.0]      0.011236
(981.0, 1308.0]     0.011236
(1308.0, 1635.0]    0.011236
(1635.0, 1962.0]    0.011236
(1962.0, 2289.0]    0.011236
(2943.0, 3270.0]    0.011236
(327.0, 654.0]      0.000000
(2289.0, 2616.0]    0.000000
(2616.0, 2943.0]    0.000000
Name: notLf2, dtype: float64
----------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  notLf2
(-3.271, 327.0]     93.258427
(654.0, 981.0]       1.123596
(981.0, 1308.0]      1.123596
(1308.0, 1635.0]     1.123596
(1635.0, 1962.0]     1.123596
(1962.0, 2289.0]     1.123596
(2943.0, 3270.0]     1.123596
(327.0, 654.0]       0.000000
(2289.0, 2616.0]     0.000000
(2616.0, 2943.0]     0.000000
Name: notLf2, dtype: float64
----------------------------------------

---------------------------------
Value counts for column:  Pr2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Pr2
(80.0, 90.0]      32
(70.0, 80.0]      28
(60.0, 70.0]      17
(90.0, 100.0]      5
(40.0, 50.0]       3
(-0.101, 10.0]     2
(50.0, 60.0]       2
(10.0, 20.0]       0
(20.0, 30.0]       0
(30.0, 40.0]       0
Name: Pr2, dtype: int64
```

```
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Pr2
(80.0, 90.0]      0.359551
(70.0, 80.0]      0.314607
(60.0, 70.0]      0.191011
(90.0, 100.0]     0.056180
(40.0, 50.0]      0.033708
(-0.101, 10.0]    0.022472
(50.0, 60.0]      0.022472
(10.0, 20.0]      0.000000
(20.0, 30.0]      0.000000
(30.0, 40.0]      0.000000
Name: Pr2, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Pr2
(80.0, 90.0]      35.955056
(70.0, 80.0]      31.460674
(60.0, 70.0]      19.101124
(90.0, 100.0]      5.617978
(40.0, 50.0]       3.370787
(-0.101, 10.0]     2.247191
(50.0, 60.0]       2.247191
(10.0, 20.0]       0.000000
(20.0, 30.0]       0.000000
(30.0, 40.0]       0.000000
Name: Pr2, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Er2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Er2
(70.0, 80.0]      34
(50.0, 60.0]      20
(60.0, 70.0]      17
(40.0, 50.0]       8
(80.0, 90.0]       4
(-0.101, 10.0]     2
(20.0, 30.0]       2
(90.0, 100.0]      2
(10.0, 20.0]       0
(30.0, 40.0]       0
Name: Er2, dtype: int64
--------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Er2
(70.0, 80.0]      0.382022
(50.0, 60.0]      0.224719
(60.0, 70.0]      0.191011
(40.0, 50.0]      0.089888
(80.0, 90.0]      0.044944
(-0.101, 10.0]    0.022472
(20.0, 30.0]      0.022472
(90.0, 100.0]     0.022472
(10.0, 20.0]      0.000000
(30.0, 40.0]      0.000000
Name: Er2, dtype: float64
--------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Er2
(70.0, 80.0]      38.202247
(50.0, 60.0]      22.471910
(60.0, 70.0]      19.101124
(40.0, 50.0]       8.988764
(80.0, 90.0]       4.494382
(-0.101, 10.0]     2.247191
(20.0, 30.0]       2.247191
(90.0, 100.0]      2.247191
(10.0, 20.0]       0.000000
(30.0, 40.0]       0.000000
Name: Er2, dtype: float64
--------------------------------------

---------------------------------
Value counts for column:  Ur2
---------------------------------

FREQUENCY DISTRIBUTION FOR COLUMN:  Ur2
```

```
(-0.101, 10.0]    31
(10.0, 20.0]      29
(20.0, 30.0]      19
(40.0, 50.0]       4
(30.0, 40.0]       3
(90.0, 100.0]      2
(50.0, 60.0]       1
(60.0, 70.0]       0
(70.0, 80.0]       0
(80.0, 90.0]       0
Name: Ur2, dtype: int64
---------------------------------------
RELATIVE FREQUENCY DISTRIBUTION FOR COLUMN:  Ur2
(-0.101, 10.0]    0.348315
(10.0, 20.0]      0.325843
(20.0, 30.0]      0.213483
(40.0, 50.0]      0.044944
(30.0, 40.0]      0.033708
(90.0, 100.0]     0.022472
(50.0, 60.0]      0.011236
(60.0, 70.0]      0.000000
(70.0, 80.0]      0.000000
(80.0, 90.0]      0.000000
Name: Ur2, dtype: float64
---------------------------------------
PERCENTAGE FREQUENCY DISTRIBUTION FOR COLUMN:  Ur2
(-0.101, 10.0]    34.831461
(10.0, 20.0]      32.584270
(20.0, 30.0]      21.348315
(40.0, 50.0]       4.494382
(30.0, 40.0]       3.370787
(90.0, 100.0]      2.247191
(50.0, 60.0]       1.123596
(60.0, 70.0]       0.000000
(70.0, 80.0]       0.000000
(80.0, 90.0]       0.000000
Name: Ur2, dtype: float64
---------------------------------------
```

For every column, explain which of the following plots make sense, and generate those plots in your notebook.

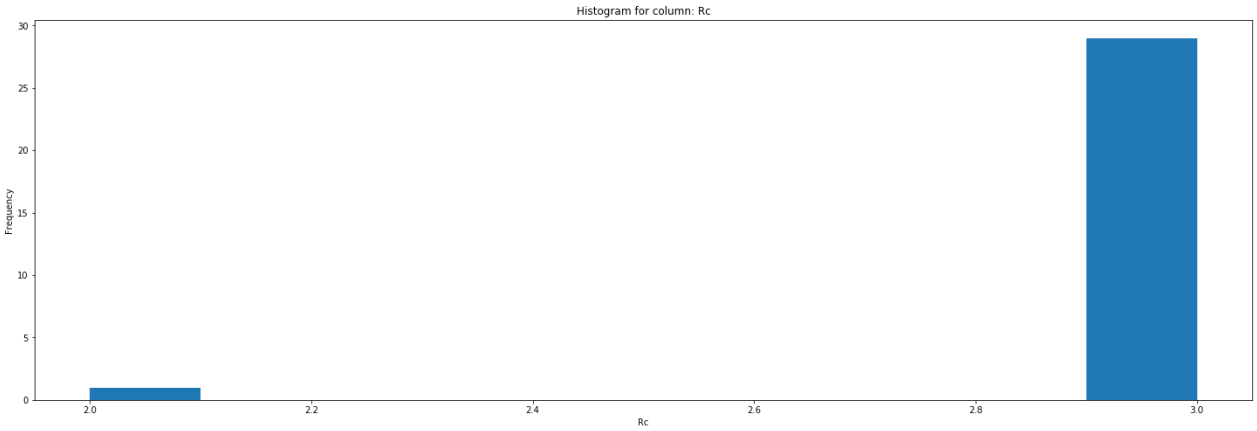
SInce the dataset I chose has very large number of x components for columns with quantitative data, it is not practical to plot the bar graph, horizontal bar graph. Thus, only pie chart and histograms are suitable for columns with quantitative data. For columns with categorical data, all the plot attributes are suitable.

Plotting histogram, cumulative histograms, bar graph, horizontal bar graph and pie chart for columns with categorical data.

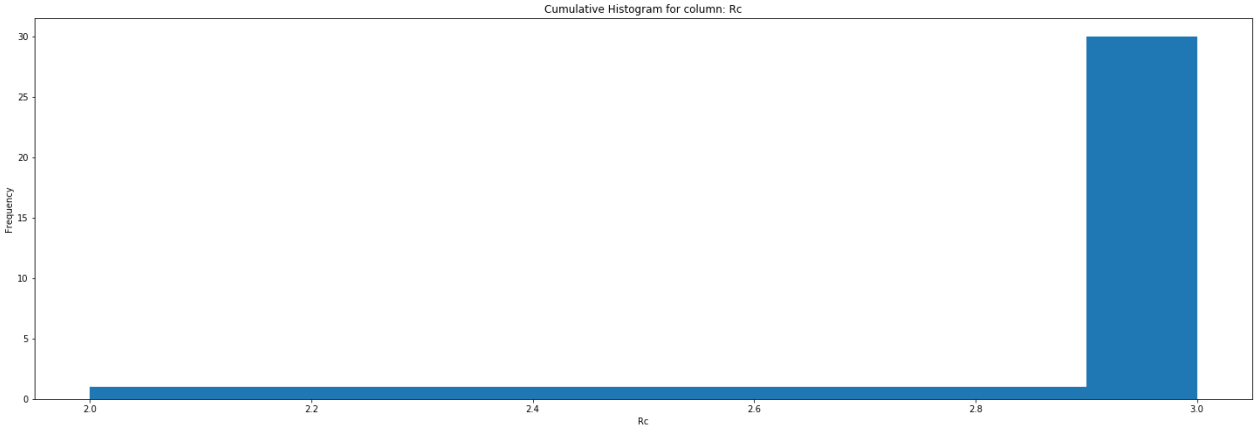
In [49]:

```
print("Plotting histogram, cumulative histograms, bar graph, horizontal bar graph, and pie chart for colum
for column in data.columns:
    if data[column].dtype == "object":

        print("")
        print("HISTOGRAM FOR COLUMN: ", column)
        plt.hist(data[column].value_counts())
        plt.rcParams["figure.figsize"] = (25, 8)
        plt.title("Histogram for column: " + column)
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.show()

        print("CUMULATIVE HISTOGRAM FOR COLUMN: ", column)
        plt.hist(data[column].value_counts(), cumulative = True)
        plt.title("Cumulative Histogram for column: " + column)
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.show()

        print("BAR GRAPH FOR COLUMN: ", column)
        plt.grid(zorder = 0)
        plt.bar(
            data[column].value_counts().keys(),
            data[column].value_counts(),
            color = ['C1', 'C2', 'C3', 'C4', 'C5'],
            zorder = 1)
        plt.title('Bar graph for column: '+ column)
        plt.xlabel(column)
        plt.ylabel('Frequency')
        plt.show()

        print("HORIZONTAL BAR GRAPH FOR COLUMN: ", column)
        plt.grid(zorder = 0)
        plt.barh(
            data[column].value_counts().keys(),
            data[column].value_counts(),
            color = ['C5', 'C6', 'C7', 'C1'],
            zorder = 1)
        plt.title('Horizontal bar graph for column: '+ column)
        plt.xlabel('Frequency')
        plt.ylabel(column)
        plt.show()

        print("PIE CHART FOR COLUMN: ", column)
        plt.pie(data[column].value_counts())
        plt.title("Pie-chart for column: "+ column)
        plt.show()
```

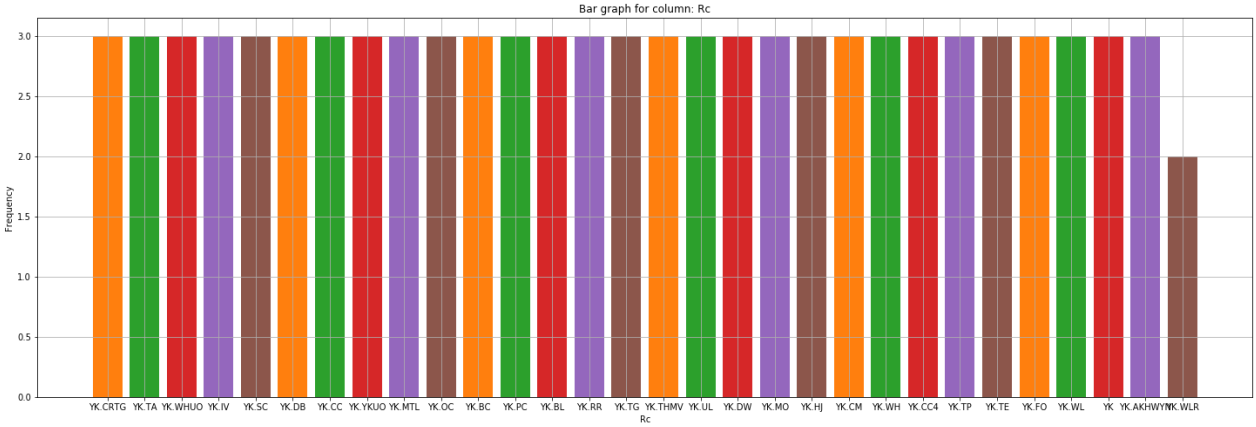
```
Plotting histogram, cumulative histograms, bar graph, horizontal bar graph, and pie chart for columns with
categorical data.

HISTOGRAM FOR COLUMN:  Rc
```



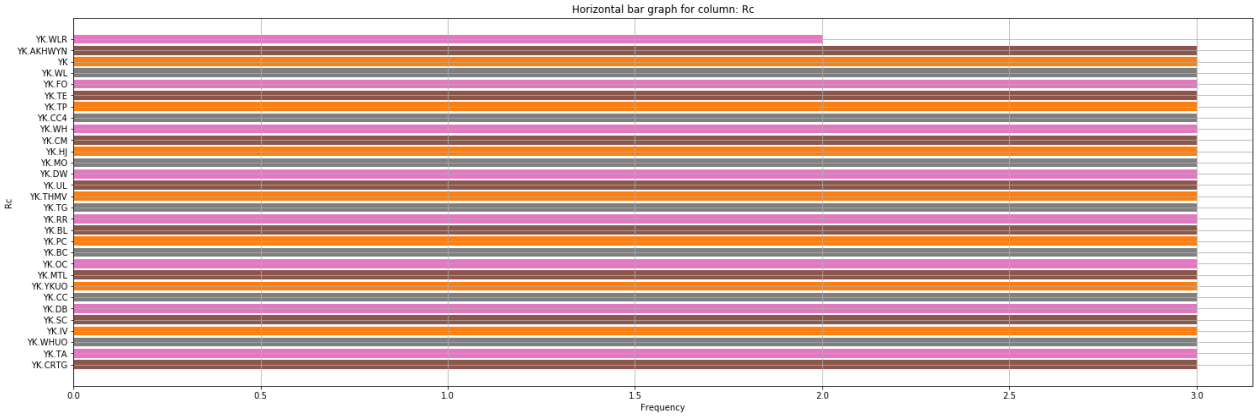

```
CUMULATIVE HISTOGRAM FOR COLUMN:  Rc
```



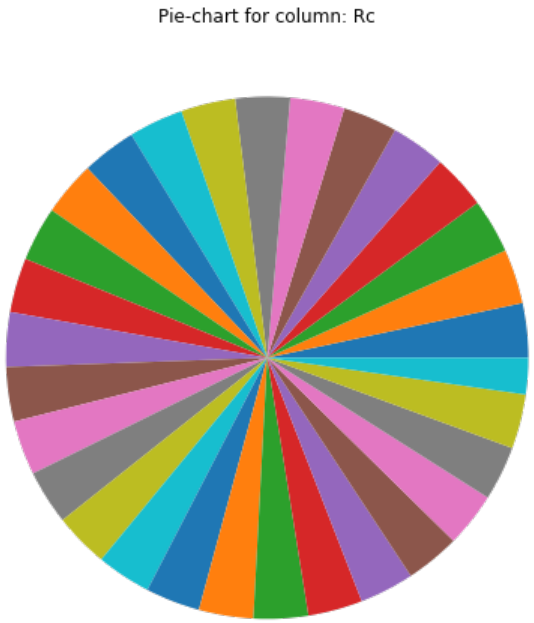
BAR GRAPH FOR COLUMN: Rc



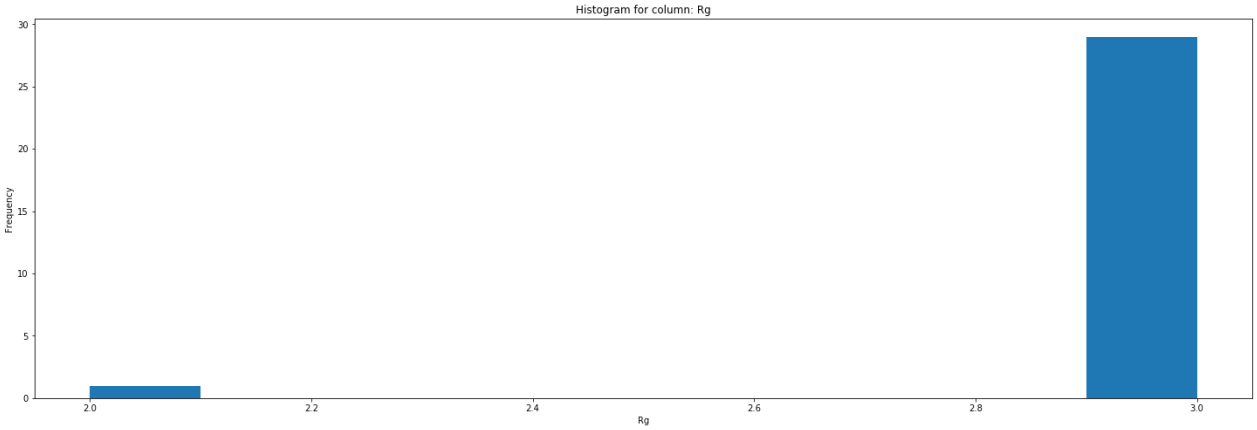
HORIZONTAL BAR GRAPH FOR COLUMN: Rc



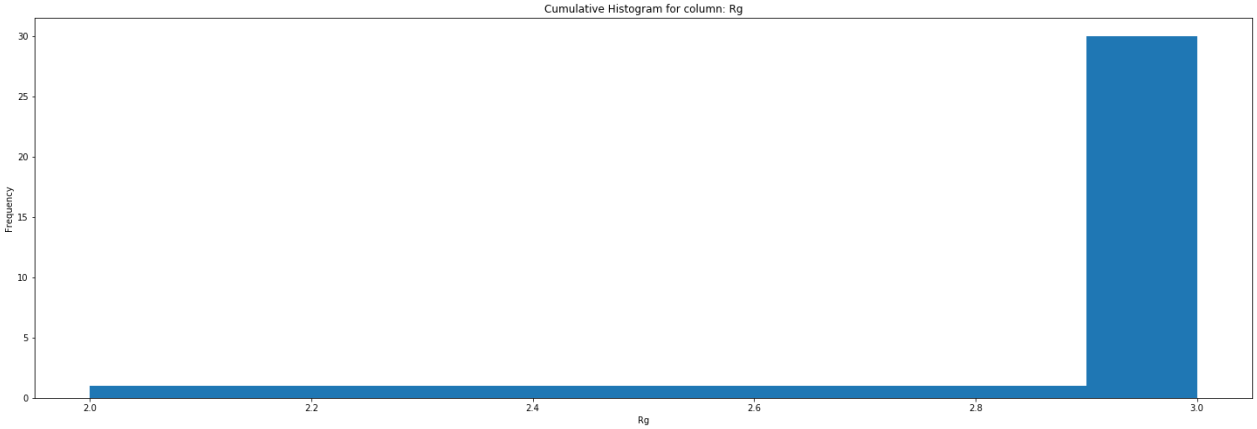
PIE CHART FOR COLUMN: Rc



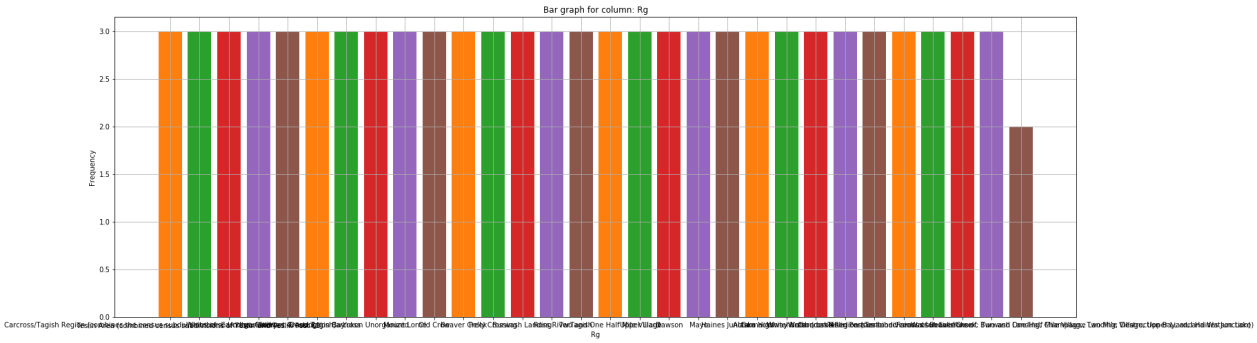
HISTOGRAM FOR COLUMN: Rg



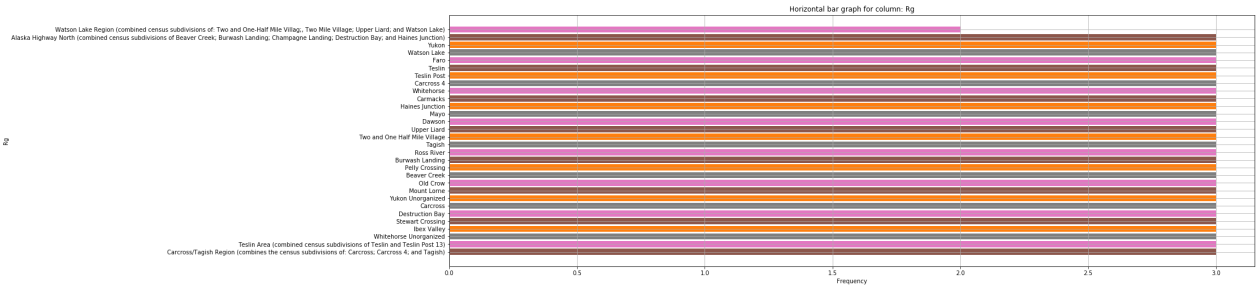
CUMULATIVE HISTOGRAM FOR COLUMN: Rg



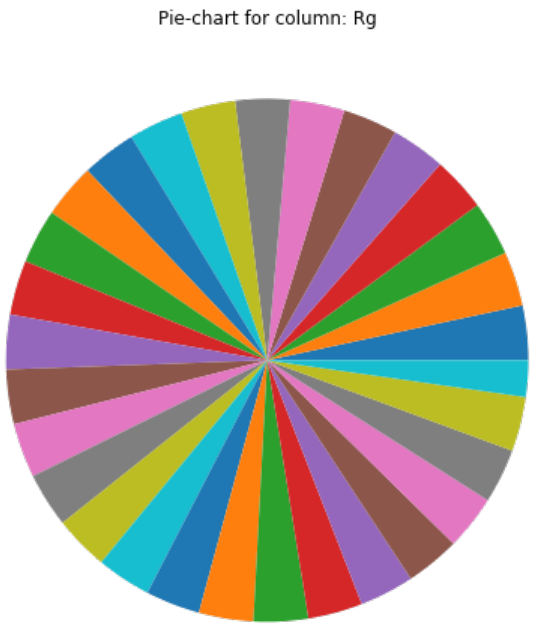
BAR GRAPH FOR COLUMN: Rg



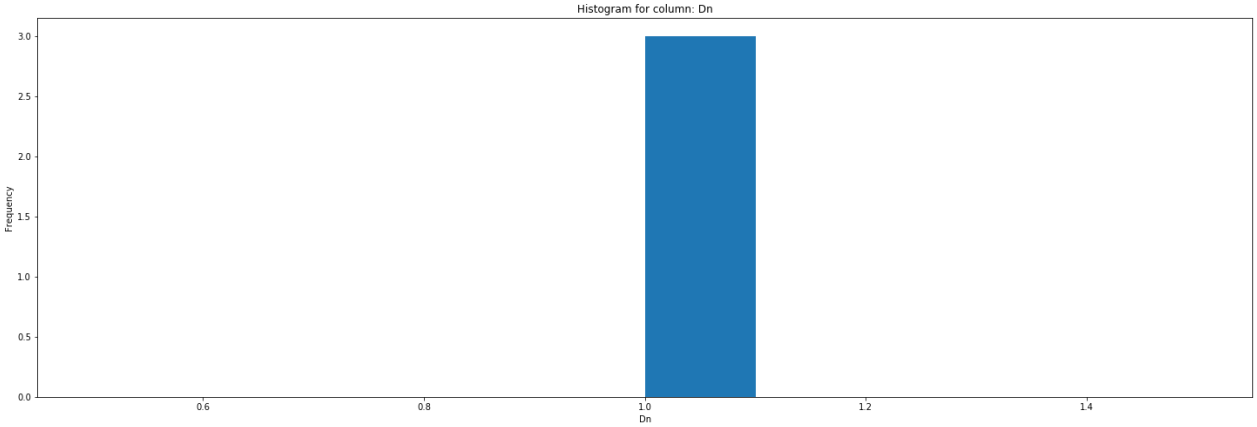
HORIZONTAL BAR GRAPH FOR COLUMN: Rg



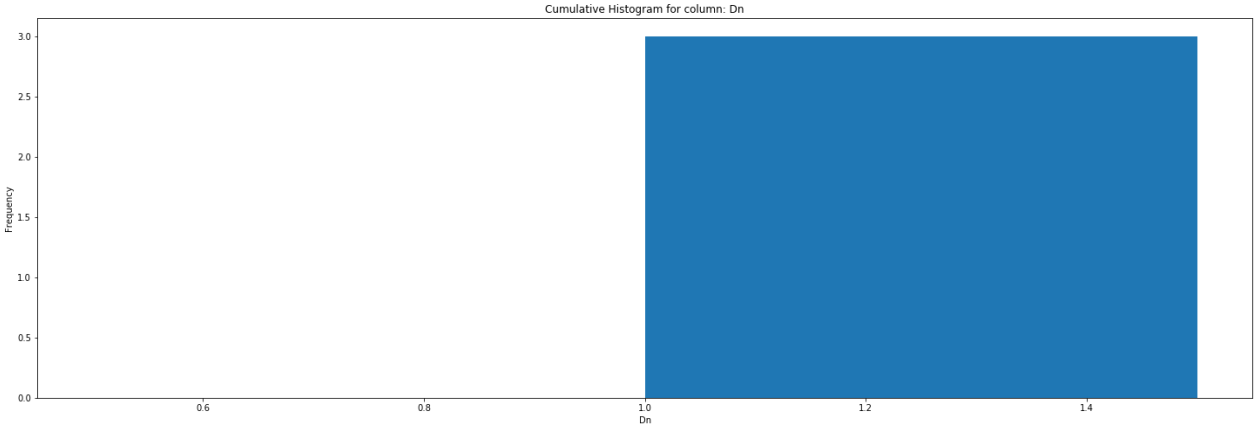
PIE CHART FOR COLUMN: Rg



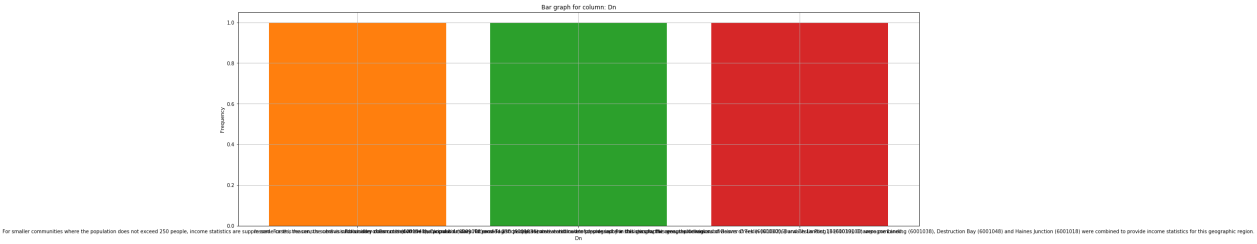
HISTOGRAM FOR COLUMN: Dn



CUMULATIVE HISTOGRAM FOR COLUMN: Dn



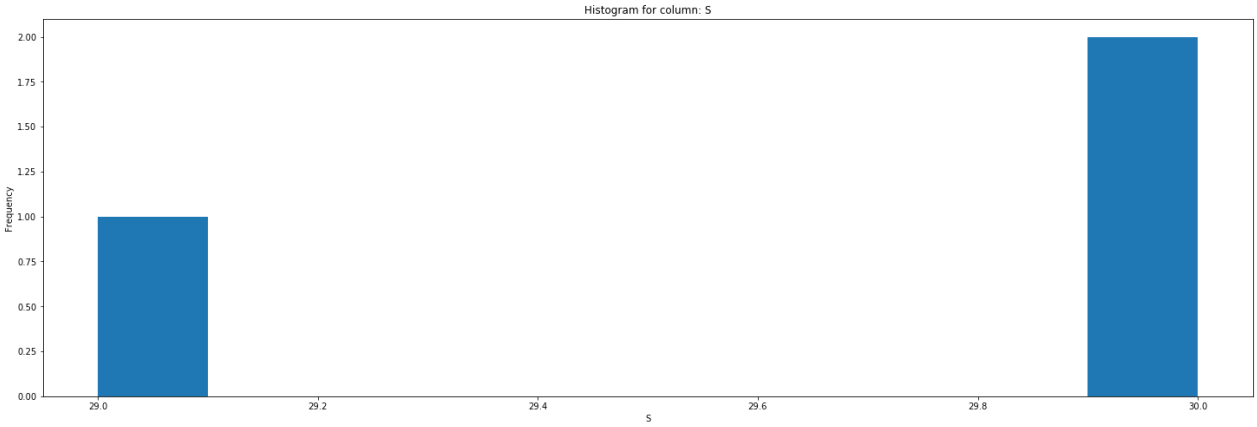
BAR GRAPH FOR COLUMN: Dn



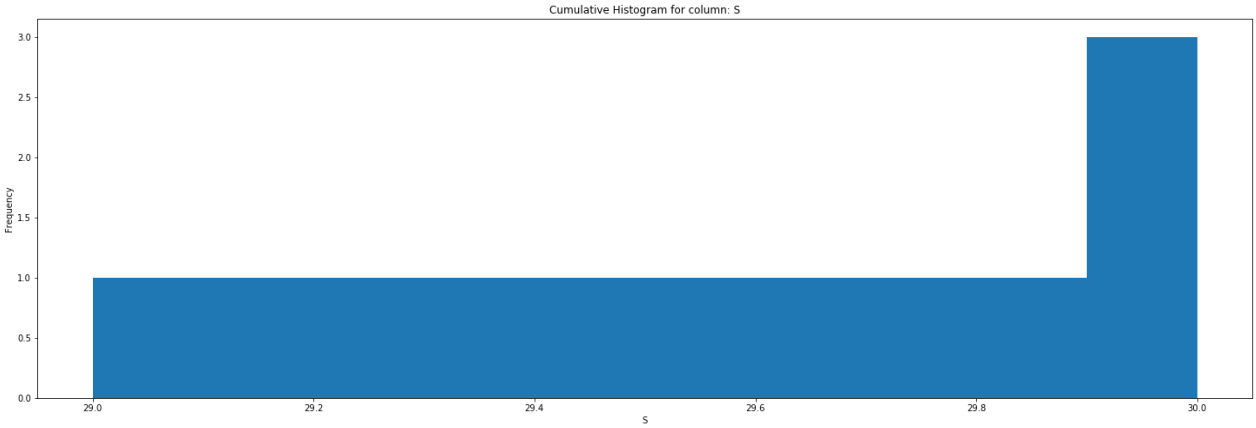
HORIZONTAL BAR GRAPH FOR COLUMN: Dn

PIE CHART FOR COLUMN: Dn

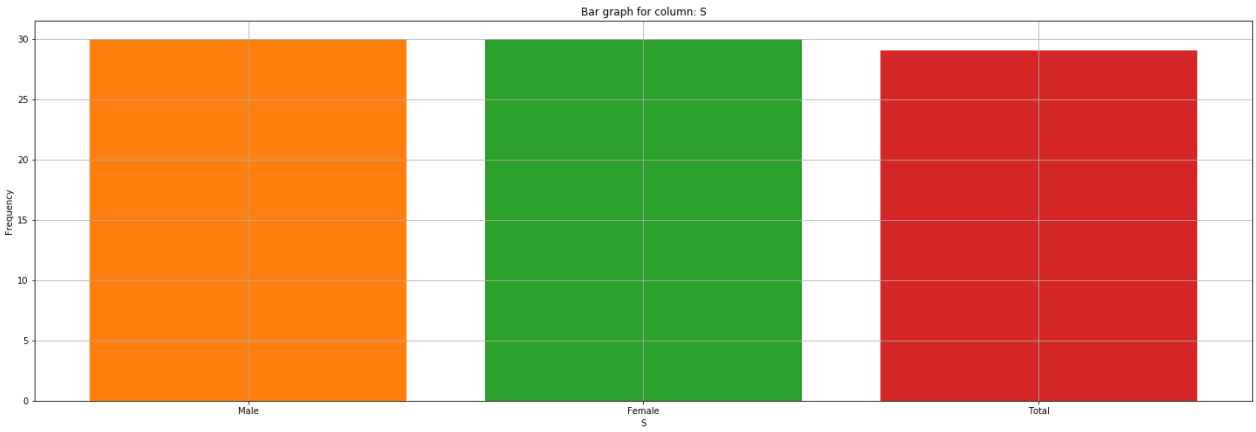
HISTOGRAM FOR COLUMN: S



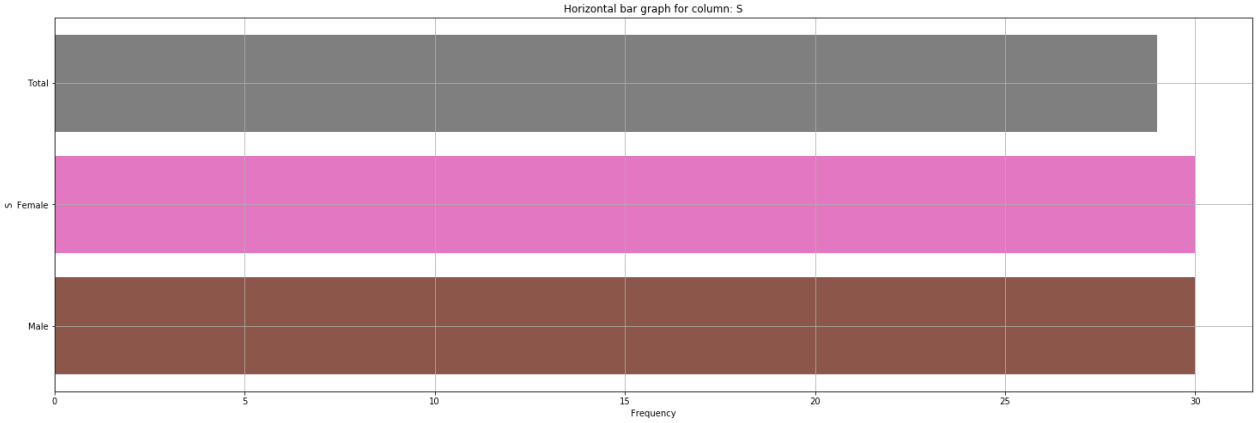
CUMULATIVE HISTOGRAM FOR COLUMN: S



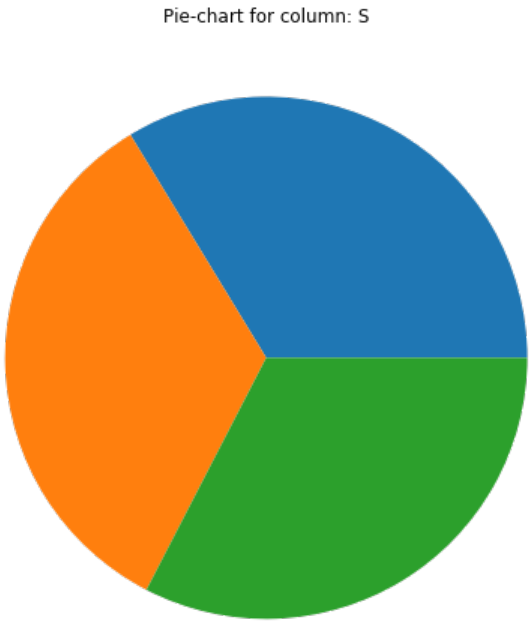
BAR GRAPH FOR COLUMN: S



HORIZONTAL BAR GRAPH FOR COLUMN: S



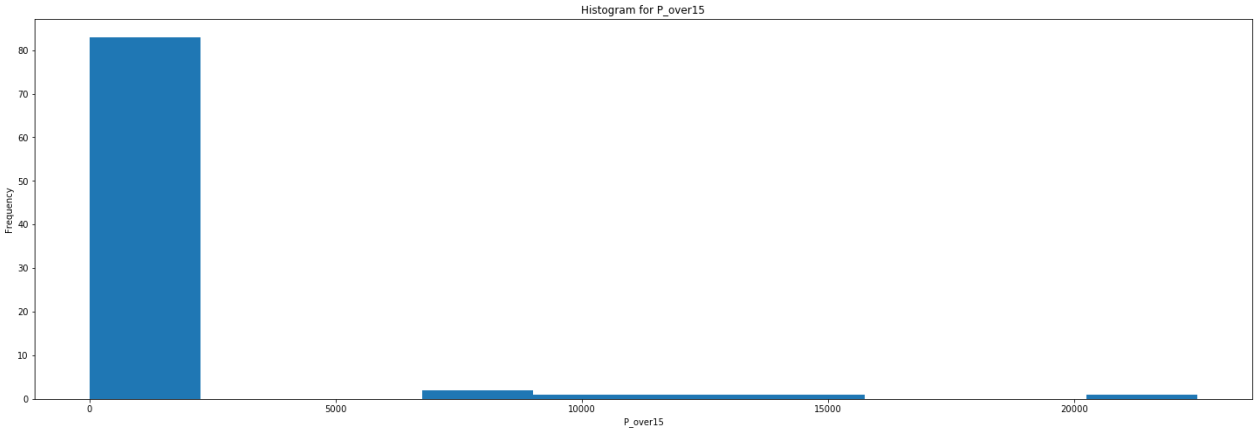
PIE CHART FOR COLUMN: S



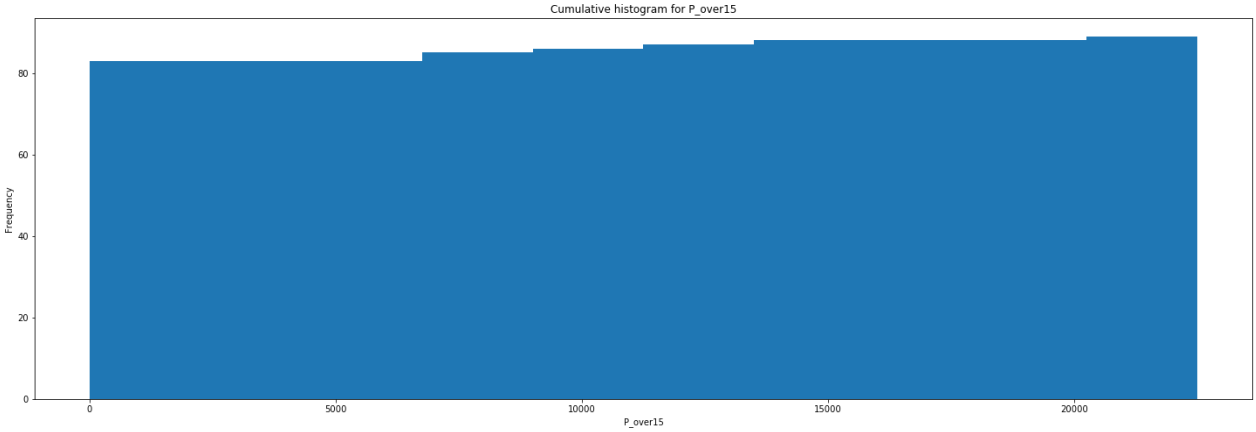
In [50]:

```
print("PLotting histogram, cumulative histogram, and pie chart for columns with quantitavtive data.")
for column in data.columns:
    if data[column].dtype != "object":

        print("")
        print("HISTOGRAM FOR COLUMN: ", column)
        plt.hist(data[column])
        plt.title("Histogram for " + column)
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.show()

        print("CUMULATIVE HISTOGRAM FOR COLUMN: ", column)
        plt.hist(data[column], cumulative = True)
        plt.title("Cumulative histogram for "+ column)
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.show()

        print("PIE CHART FOR COLUMN: ", column)
        plt.pie(data[column].value_counts(bins = 10))
        plt.title("Pie-chart for "+ column)
        plt.show()
```

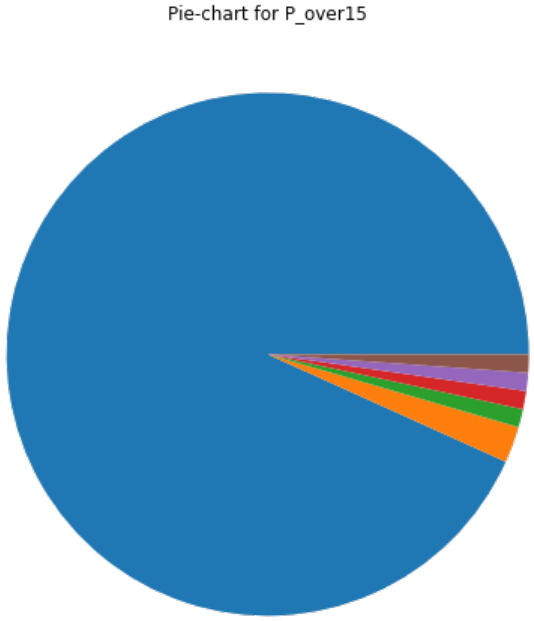
```
PLotting histogram, cumulative histogram, and pie chart for columns with quantitavtive data.

HISTOGRAM FOR COLUMN:  P_over15
```



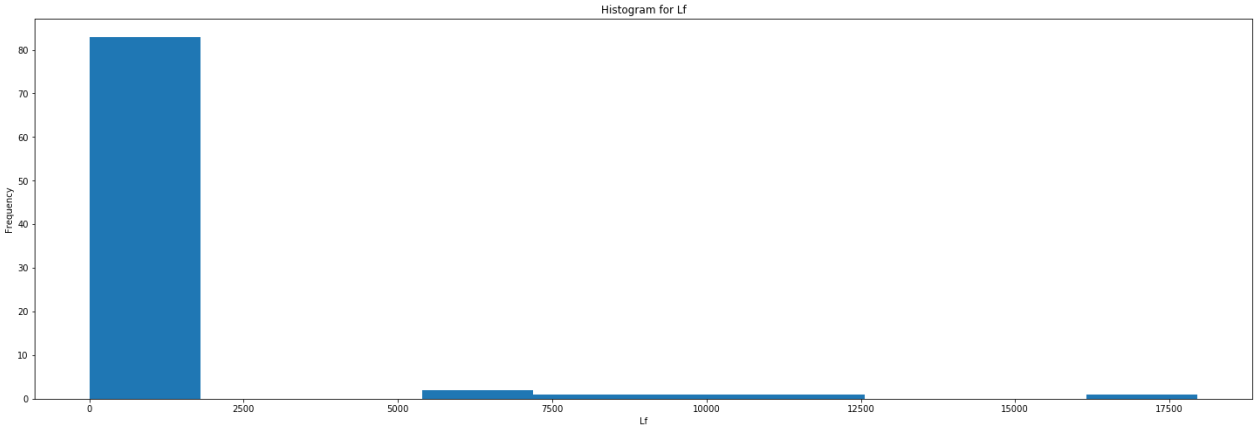

```
CUMULATIVE HISTOGRAM FOR COLUMN:  P_over15
```



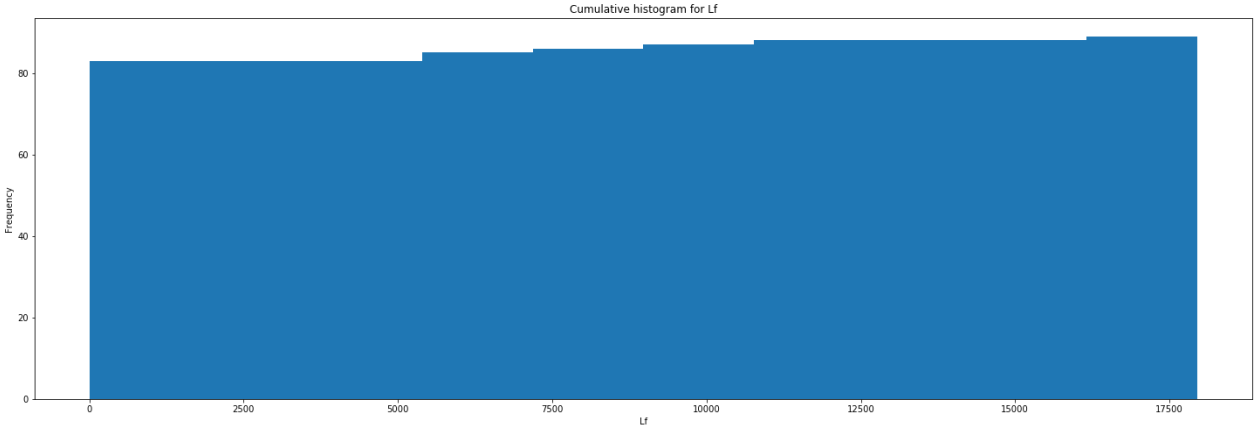
PIE CHART FOR COLUMN: P_over15



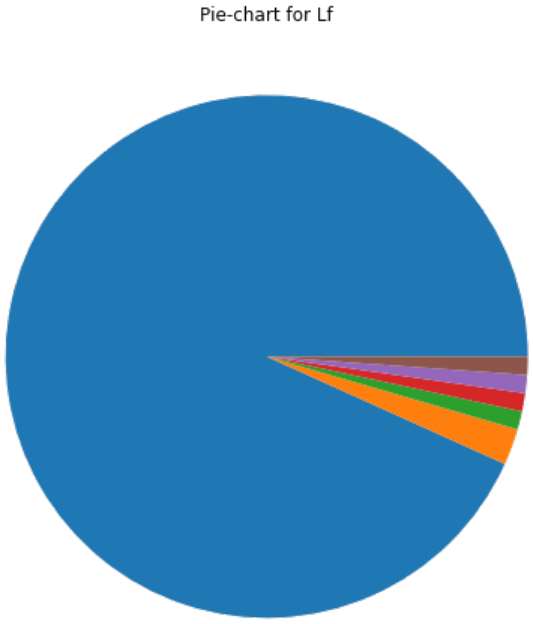
HISTOGRAM FOR COLUMN: Lf



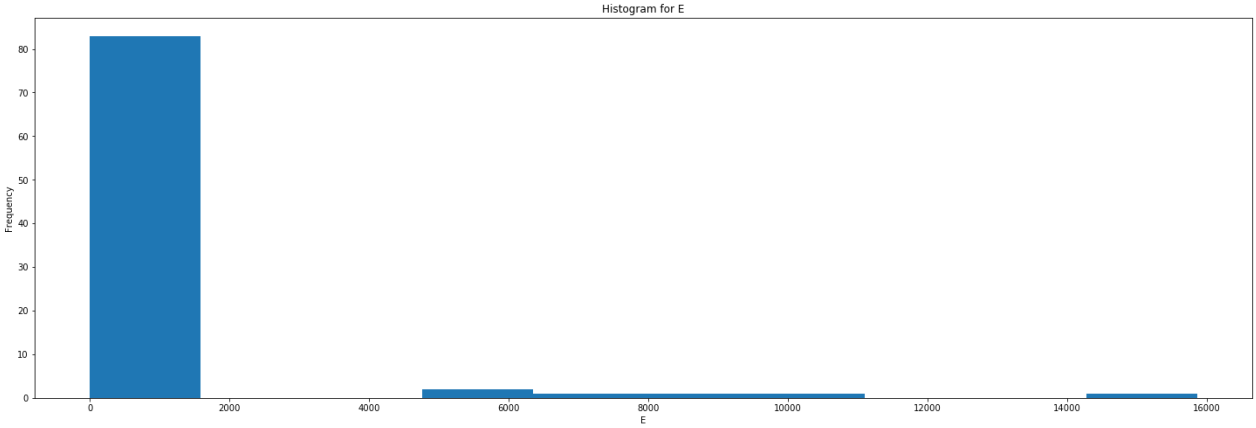
CUMULATIVE HISTOGRAM FOR COLUMN: Lf



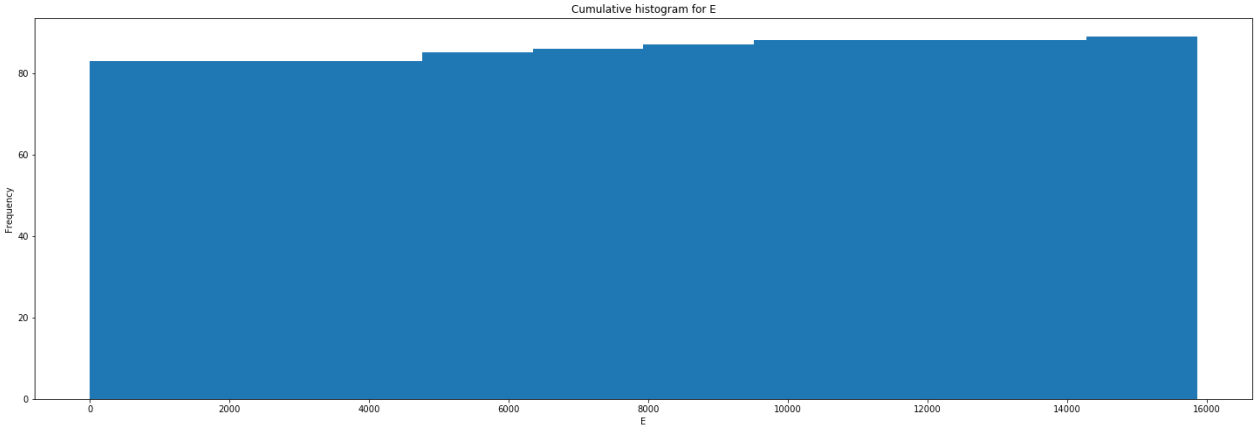
PIE CHART FOR COLUMN: Lf



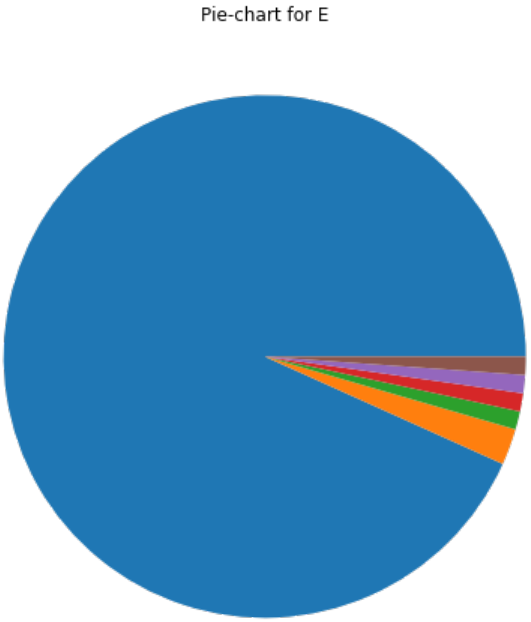
HISTOGRAM FOR COLUMN: E



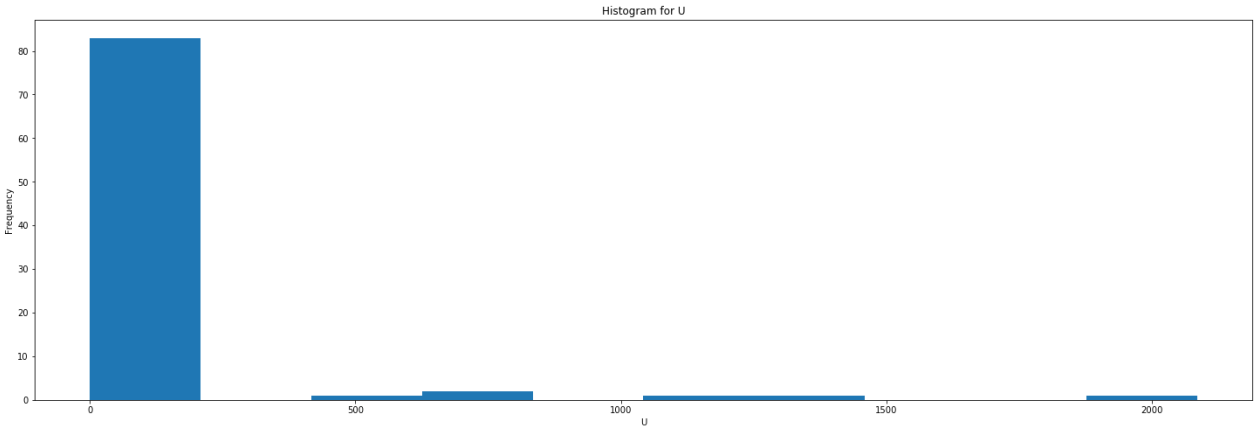
CUMULATIVE HISTOGRAM FOR COLUMN: E



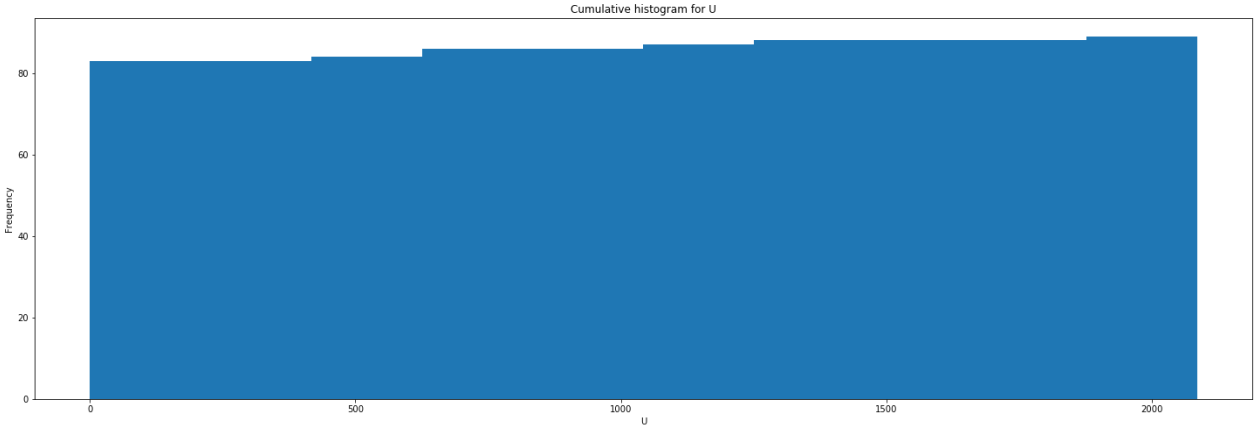
PIE CHART FOR COLUMN: E



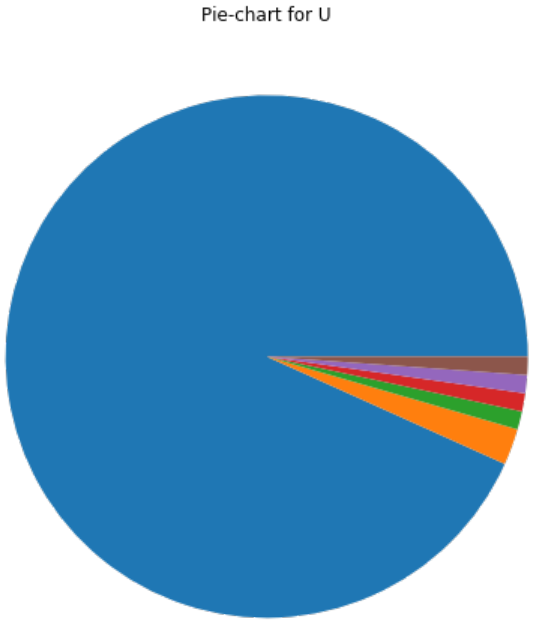
HISTOGRAM FOR COLUMN: U



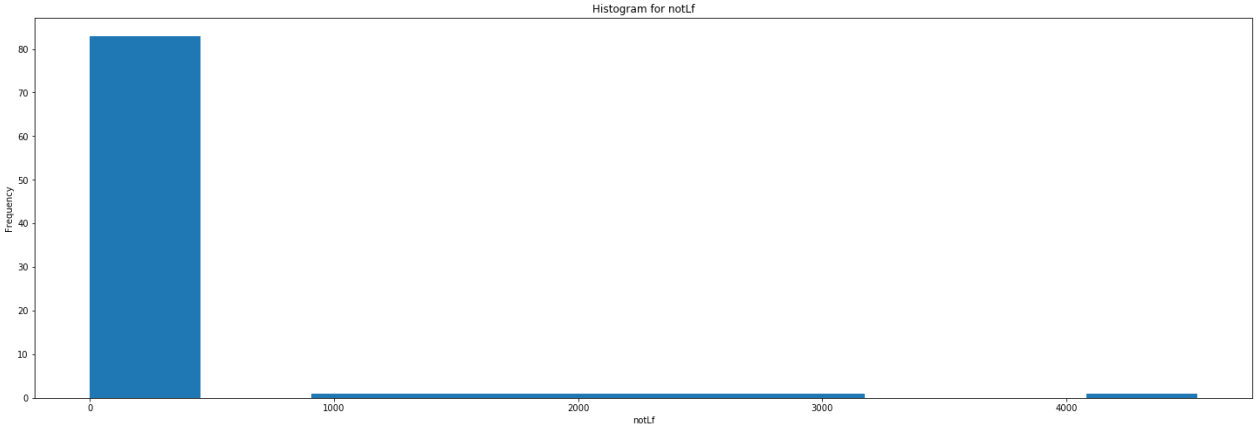
CUMULATIVE HISTOGRAM FOR COLUMN: U



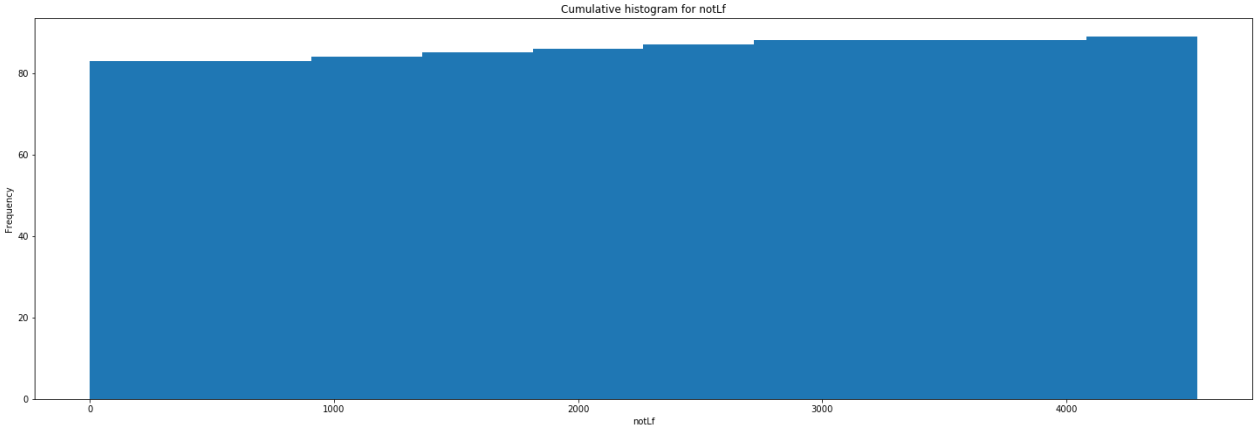
PIE CHART FOR COLUMN: U



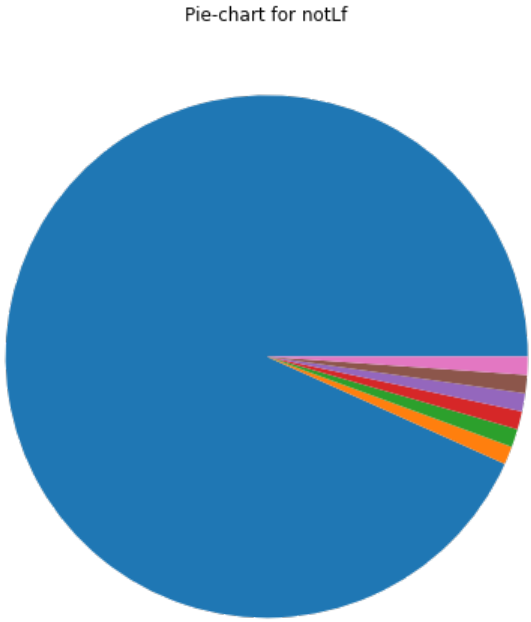
HISTOGRAM FOR COLUMN: notLf



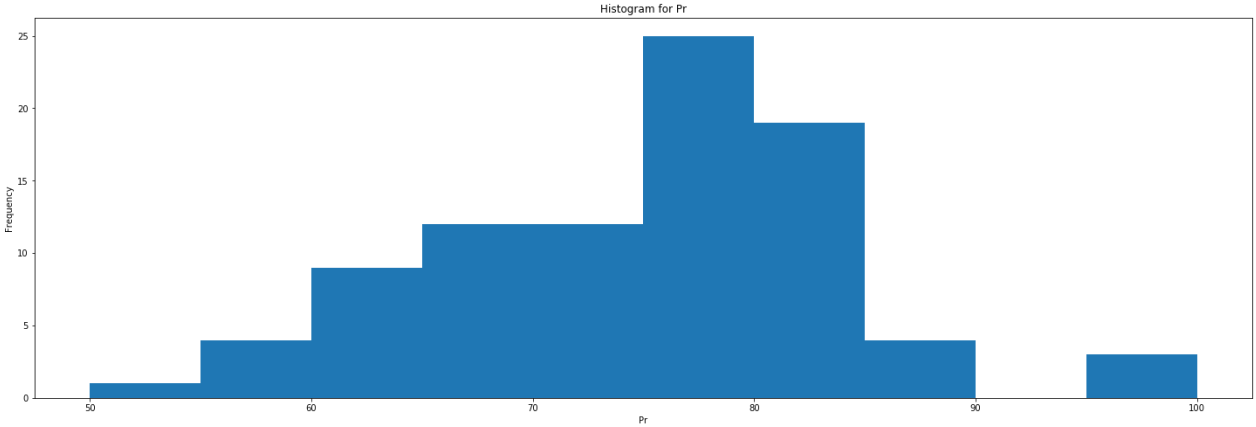
CUMULATIVE HISTOGRAM FOR COLUMN: notLf



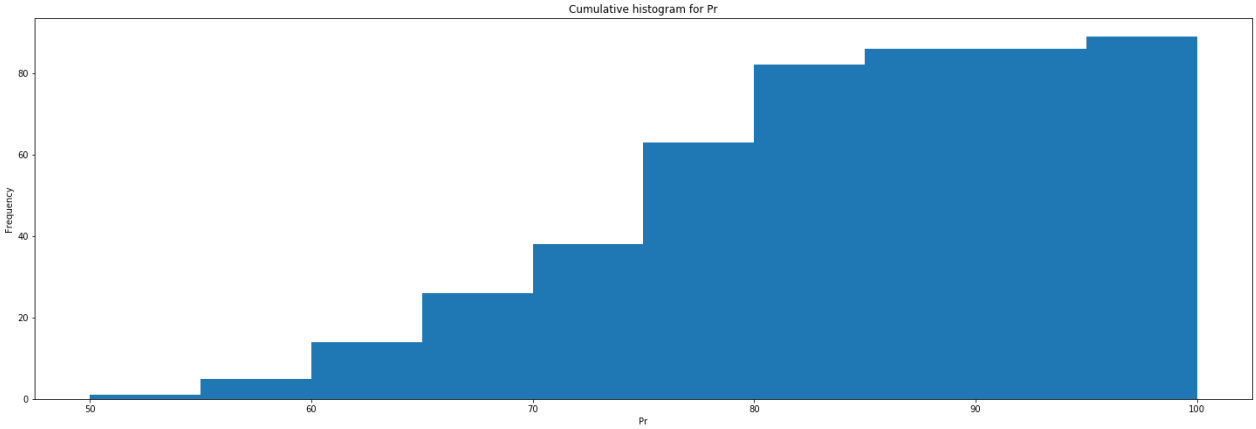
PIE CHART FOR COLUMN: notLf



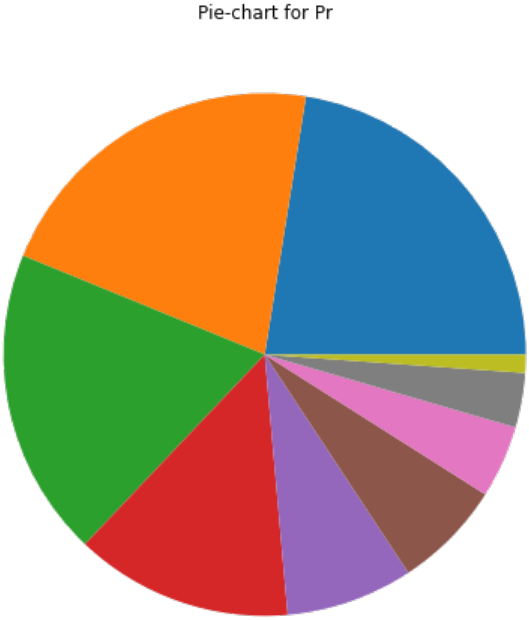
HISTOGRAM FOR COLUMN: Pr



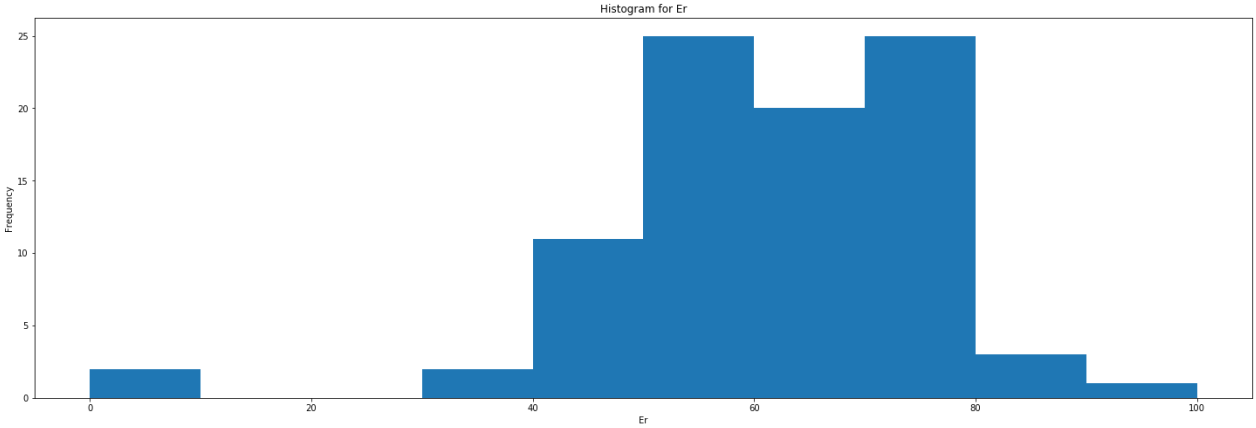
CUMULATIVE HISTOGRAM FOR COLUMN: Pr



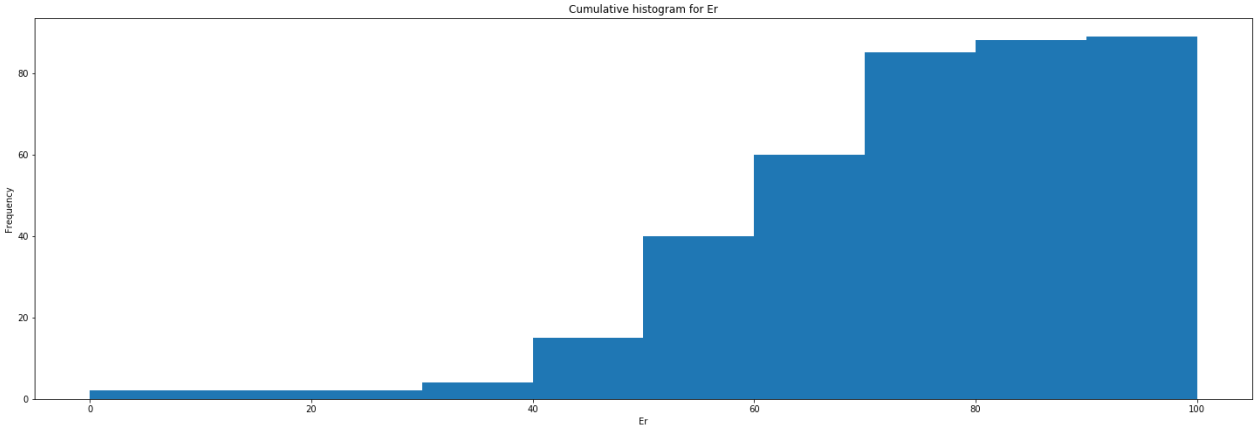
PIE CHART FOR COLUMN: Pr



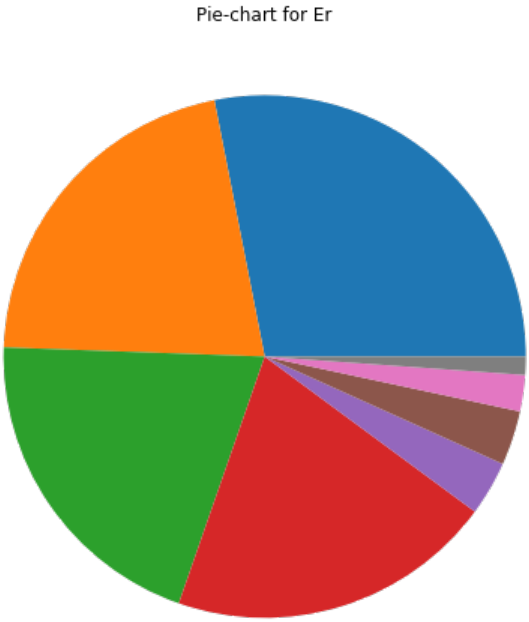
HISTOGRAM FOR COLUMN: Er



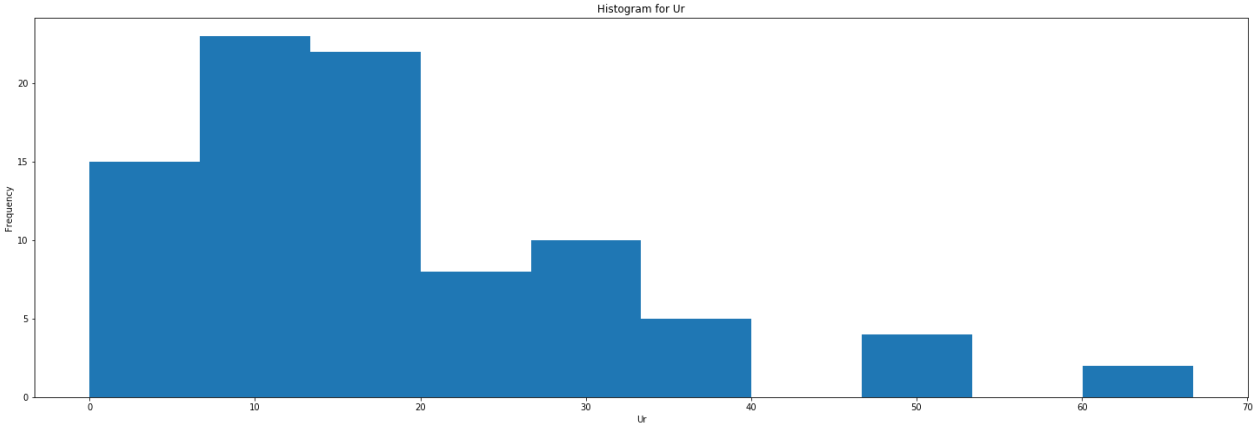
CUMULATIVE HISTOGRAM FOR COLUMN: Er



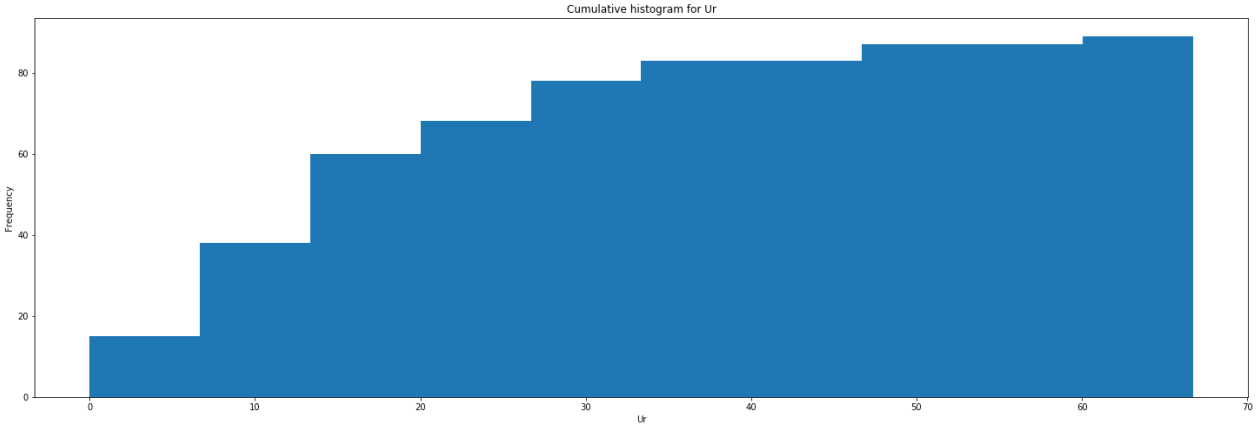
PIE CHART FOR COLUMN: Er



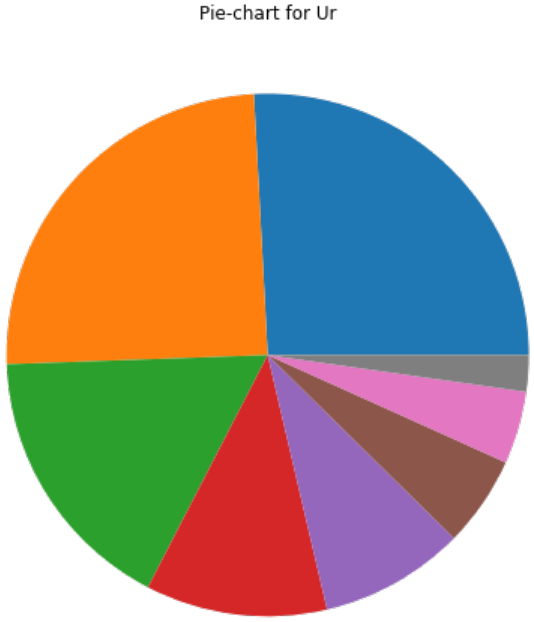
HISTOGRAM FOR COLUMN: Ur



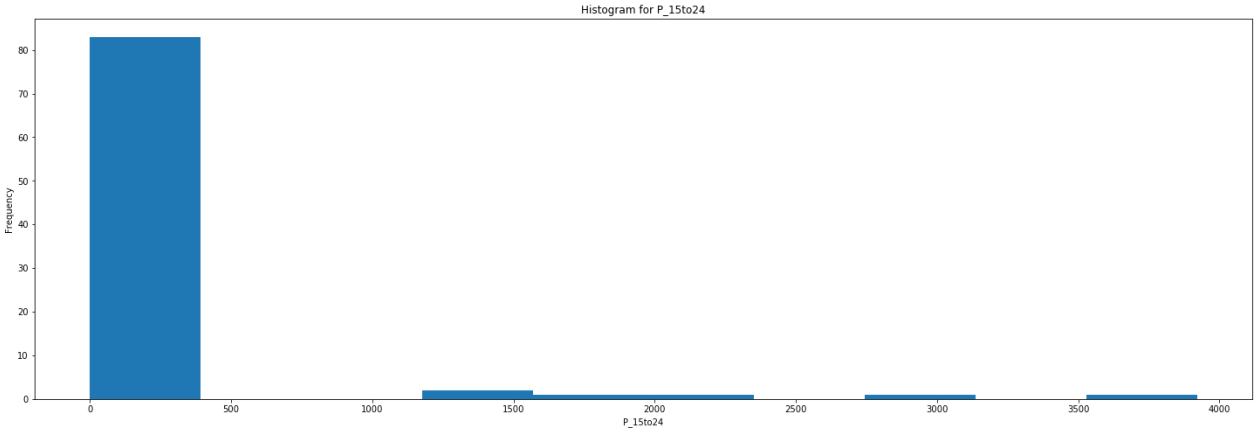
CUMULATIVE HISTOGRAM FOR COLUMN: Ur



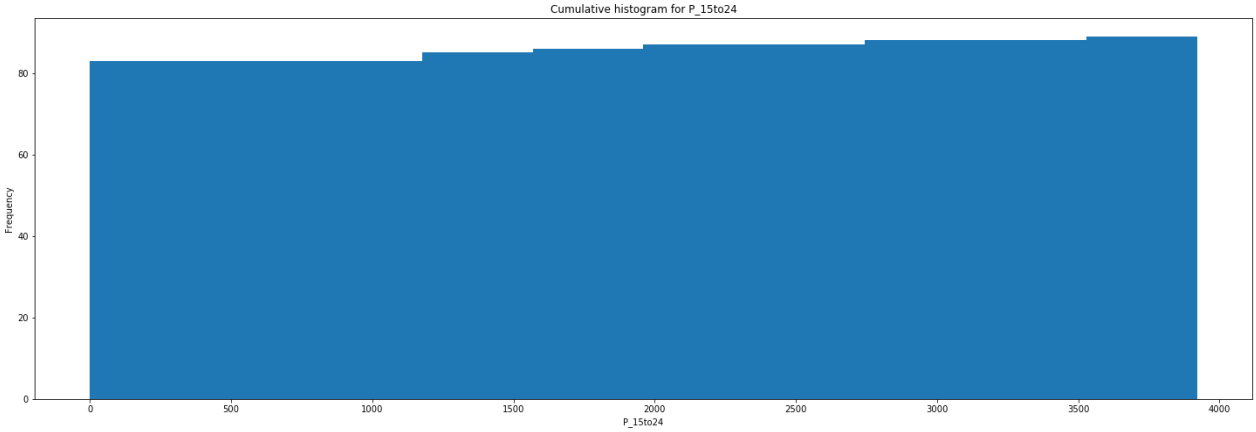
PIE CHART FOR COLUMN: Ur



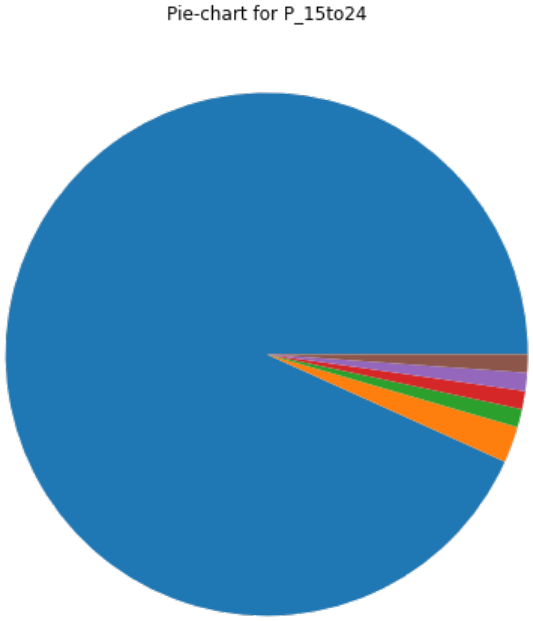
HISTOGRAM FOR COLUMN: P_15to24



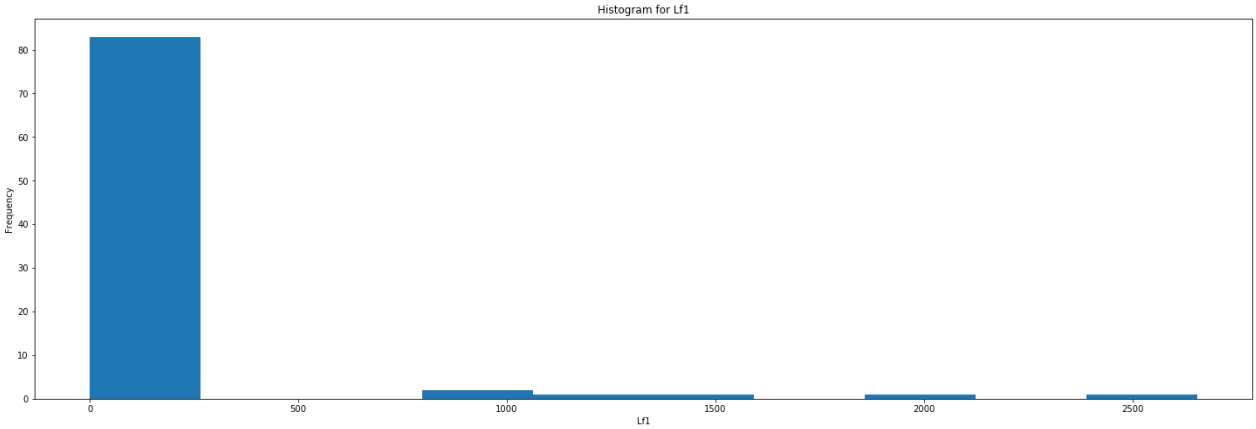
CUMULATIVE HISTOGRAM FOR COLUMN: P_15to24



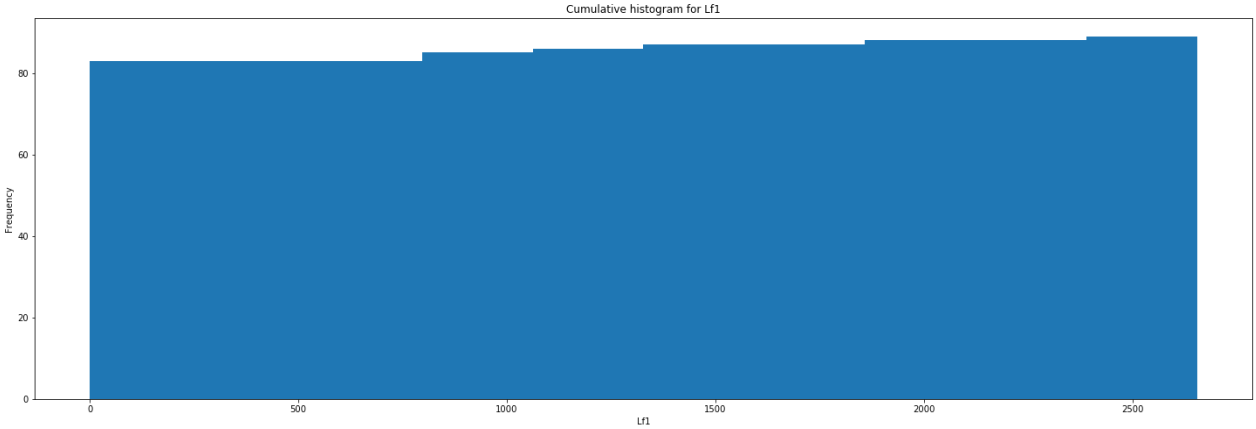
PIE CHART FOR COLUMN: P_15to24



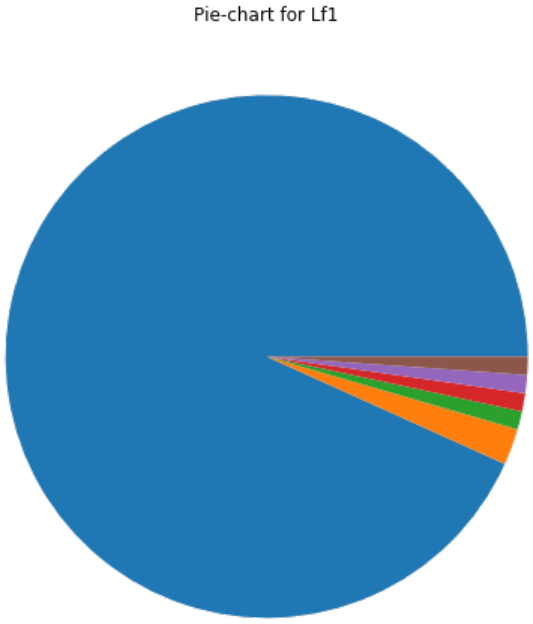
HISTOGRAM FOR COLUMN: Lf1



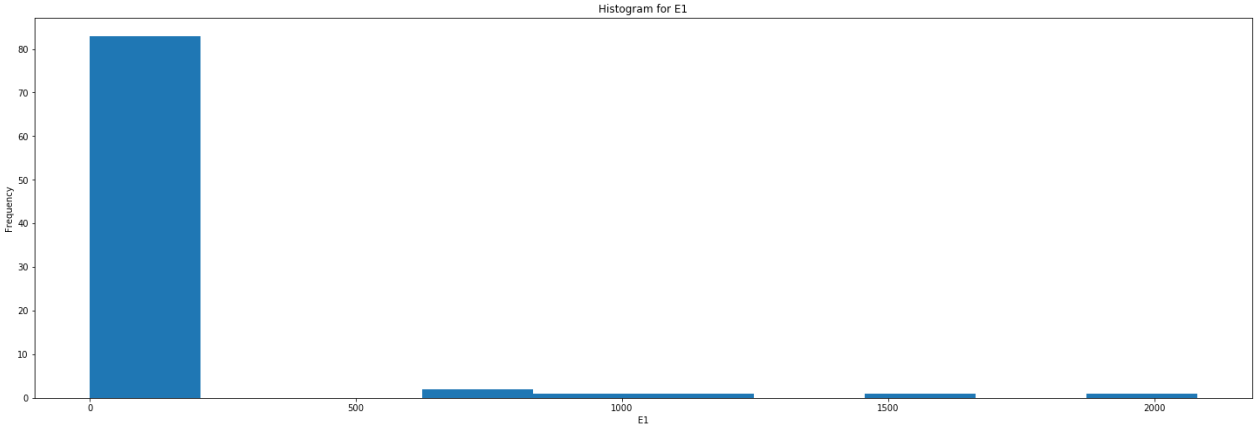
CUMULATIVE HISTOGRAM FOR COLUMN: Lf1



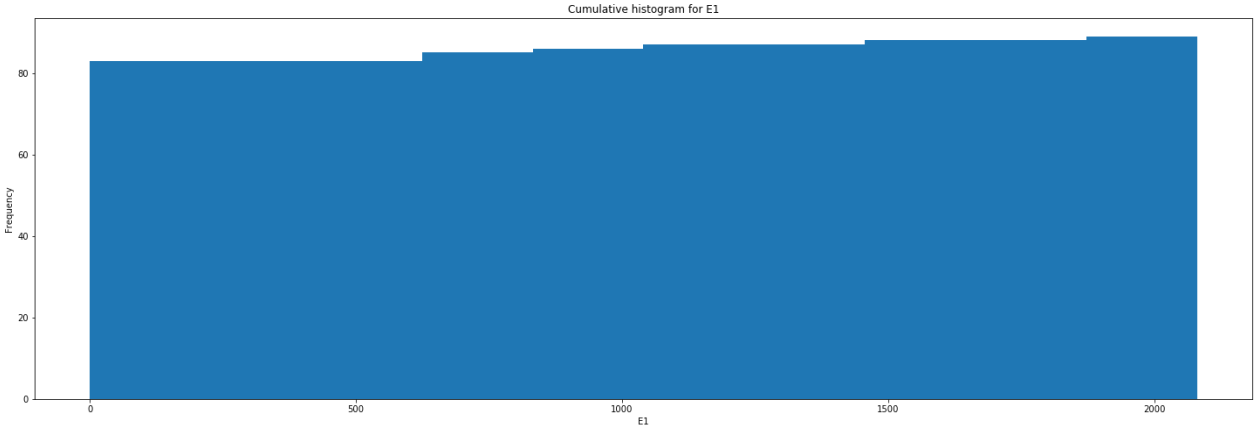
PIE CHART FOR COLUMN: Lf1



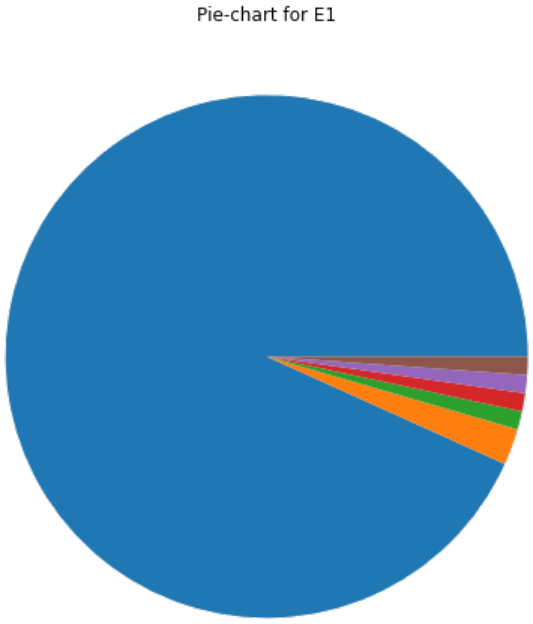
HISTOGRAM FOR COLUMN: E1



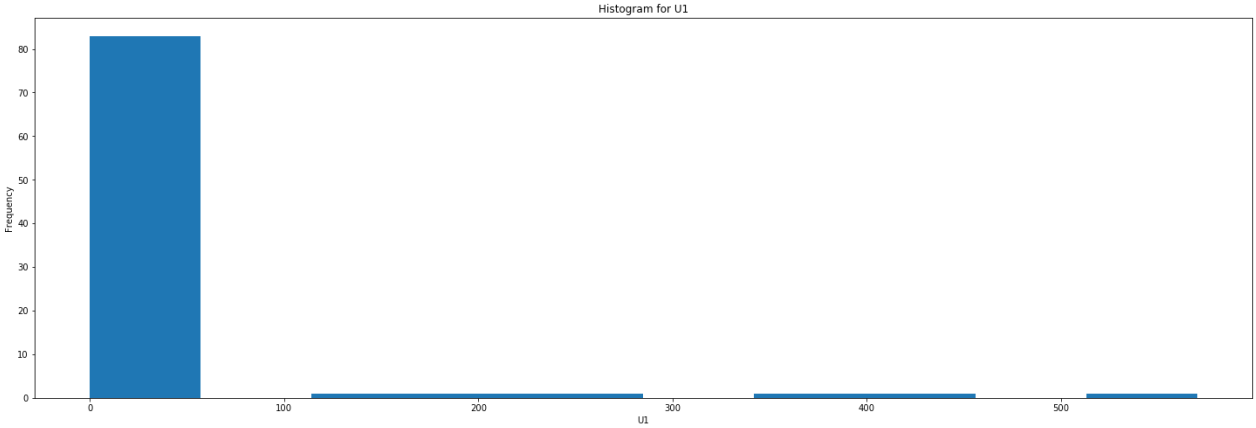
CUMULATIVE HISTOGRAM FOR COLUMN: E1



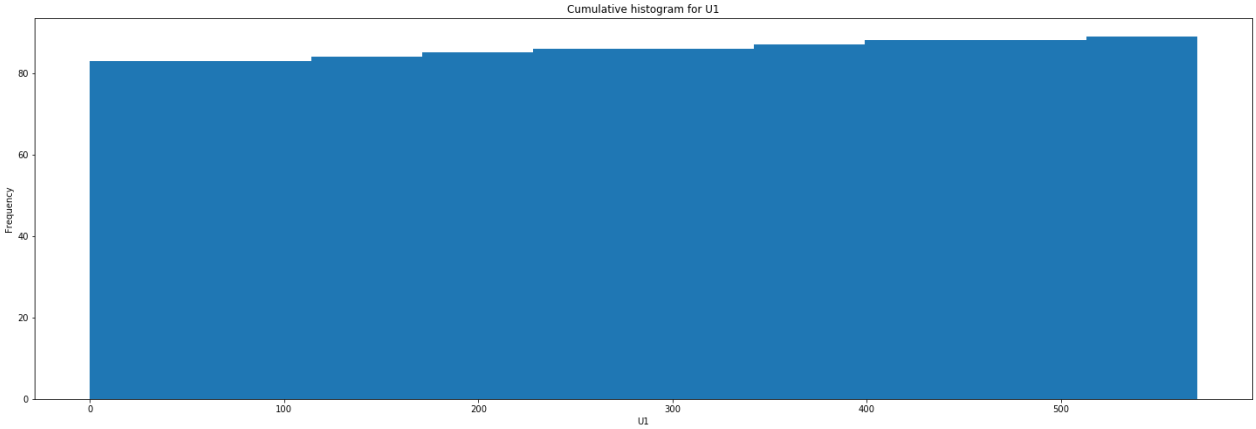
PIE CHART FOR COLUMN: E1



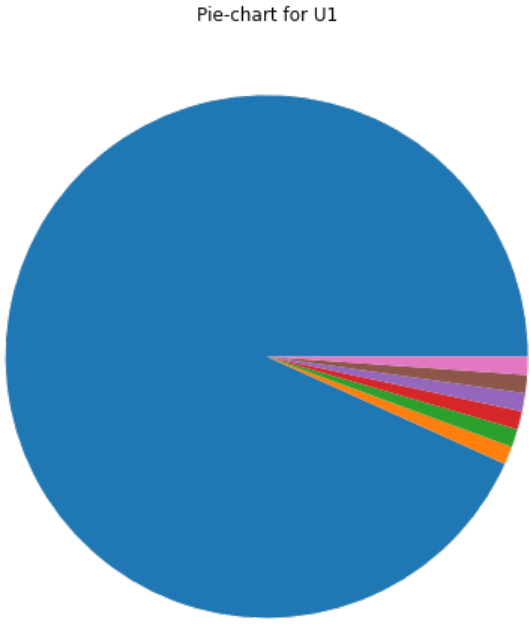
HISTOGRAM FOR COLUMN: U1



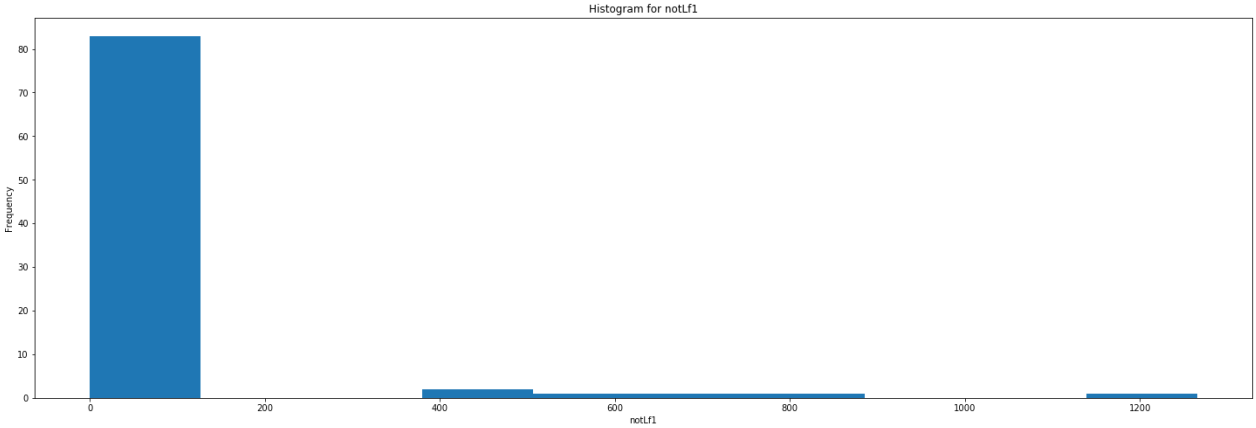
CUMULATIVE HISTOGRAM FOR COLUMN: U1



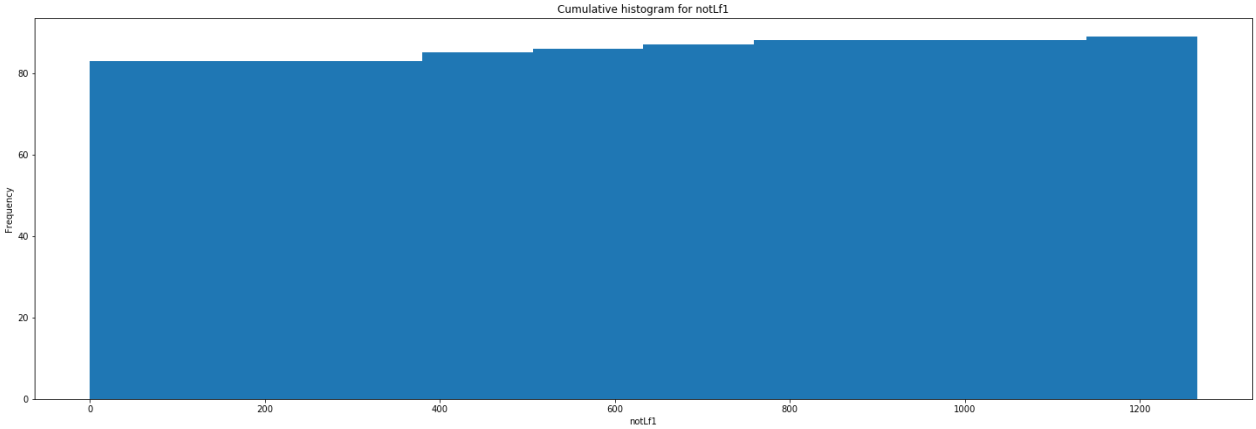
PIE CHART FOR COLUMN: U1



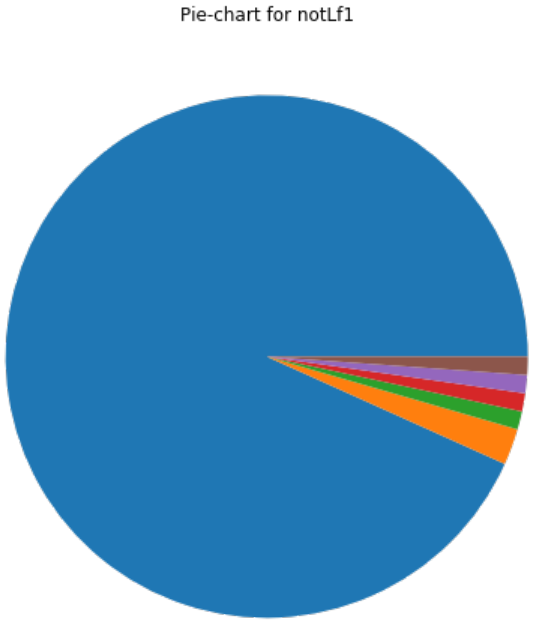
HISTOGRAM FOR COLUMN: notLf1



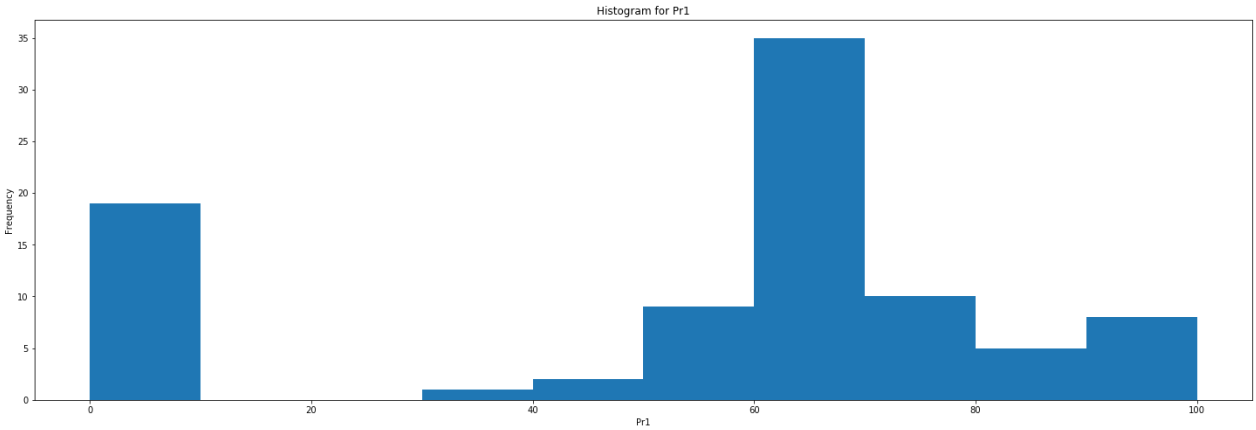
CUMULATIVE HISTOGRAM FOR COLUMN: notLf1



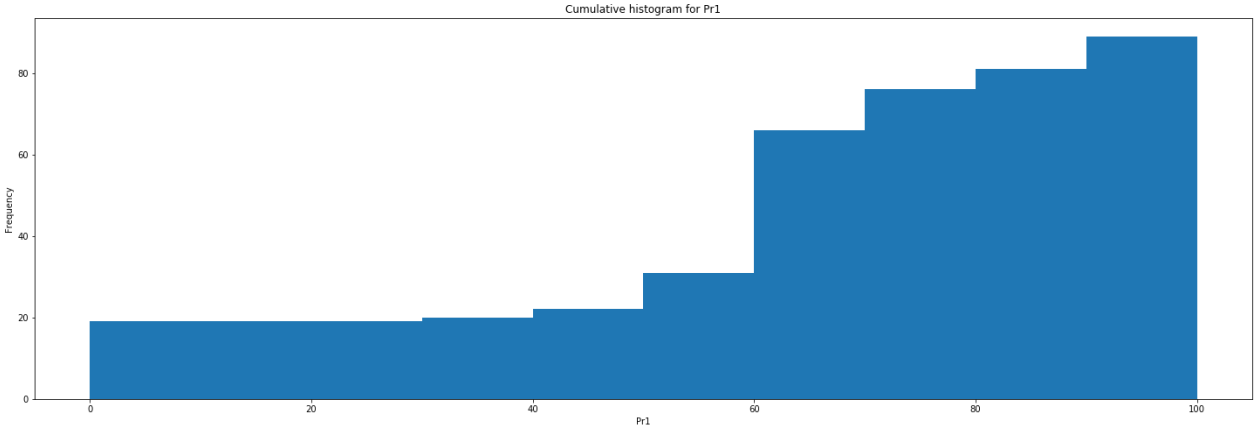
PIE CHART FOR COLUMN: notLf1



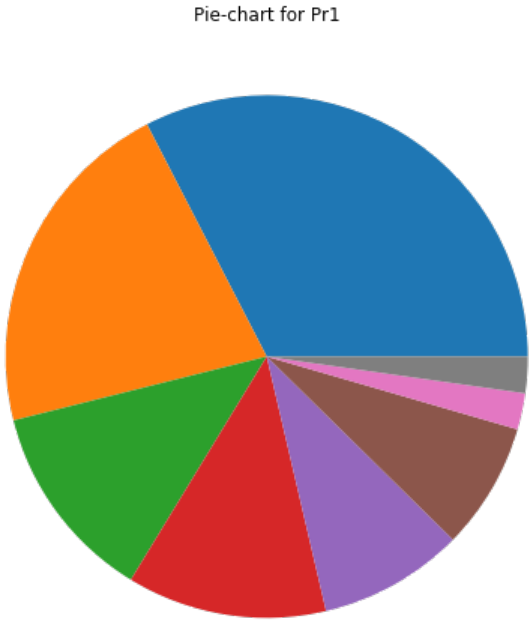
HISTOGRAM FOR COLUMN: Pr1



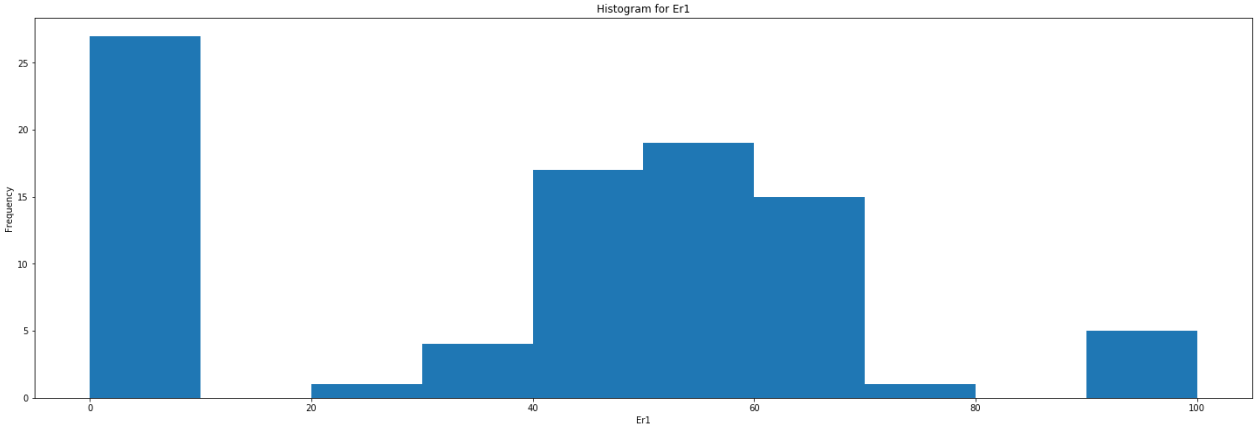
CUMULATIVE HISTOGRAM FOR COLUMN: Pr1



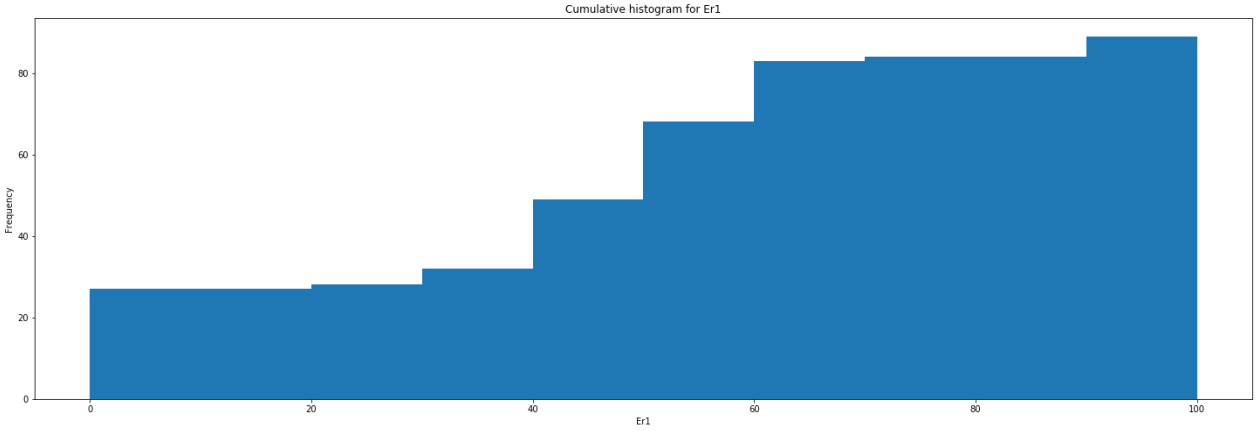
PIE CHART FOR COLUMN: Pr1



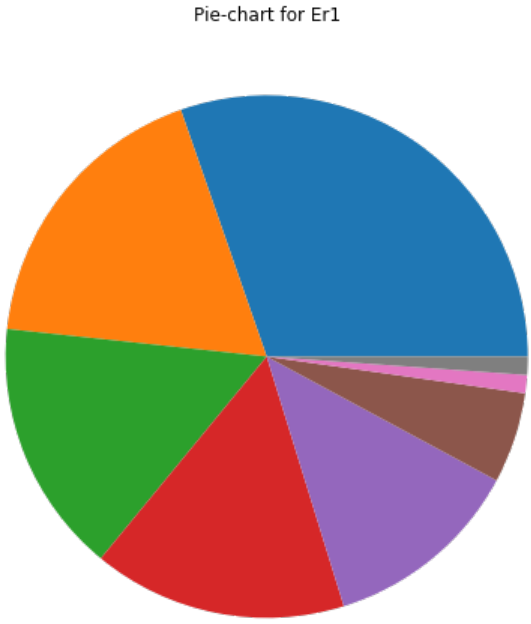
HISTOGRAM FOR COLUMN: Er1



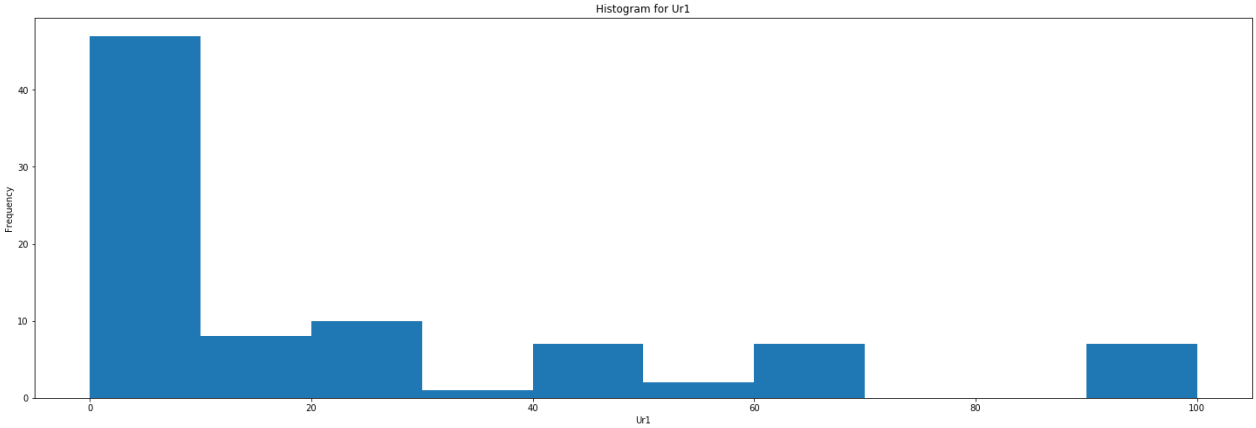
CUMULATIVE HISTOGRAM FOR COLUMN: Er1



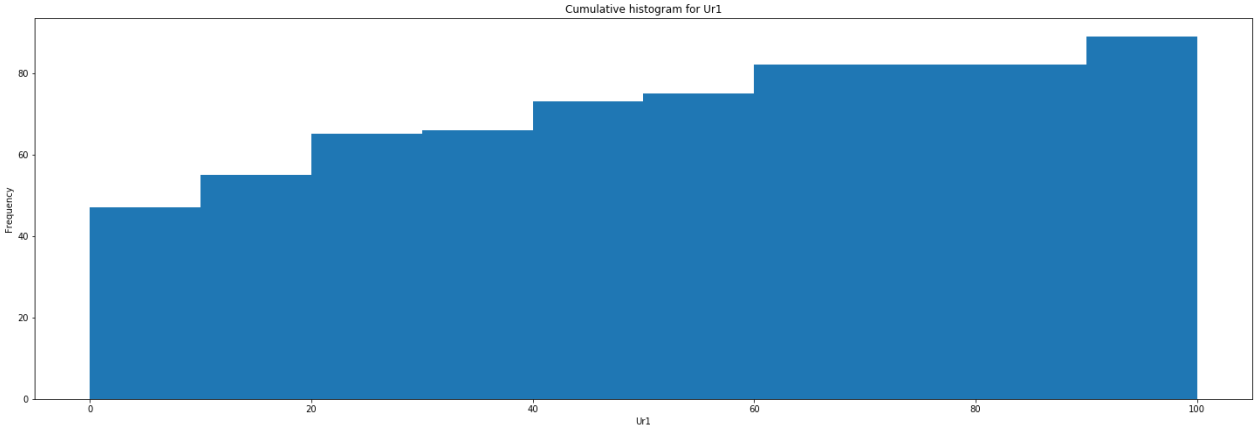
PIE CHART FOR COLUMN: Er1



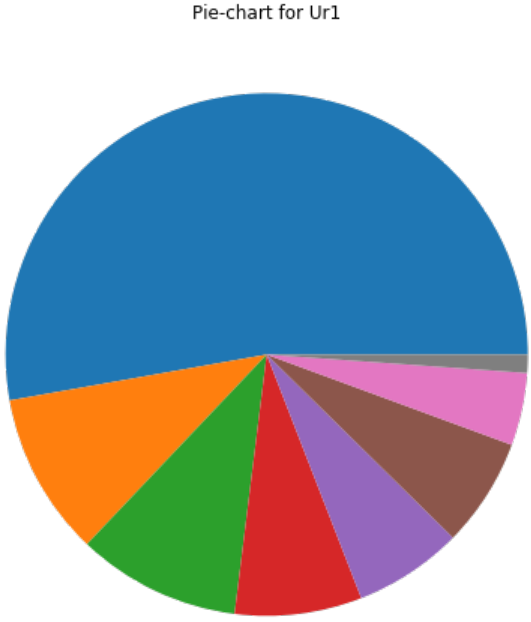
HISTOGRAM FOR COLUMN: Ur1



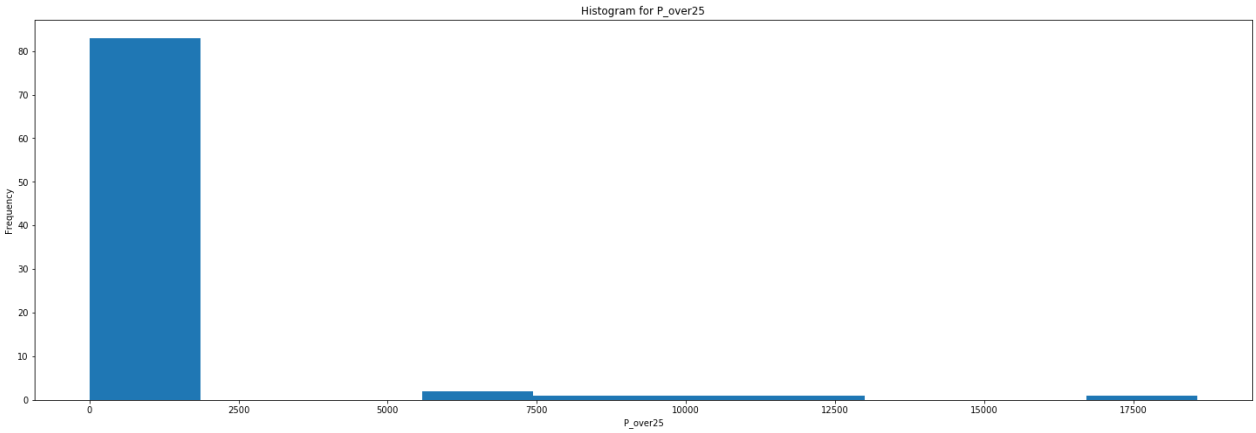
CUMULATIVE HISTOGRAM FOR COLUMN: Ur1



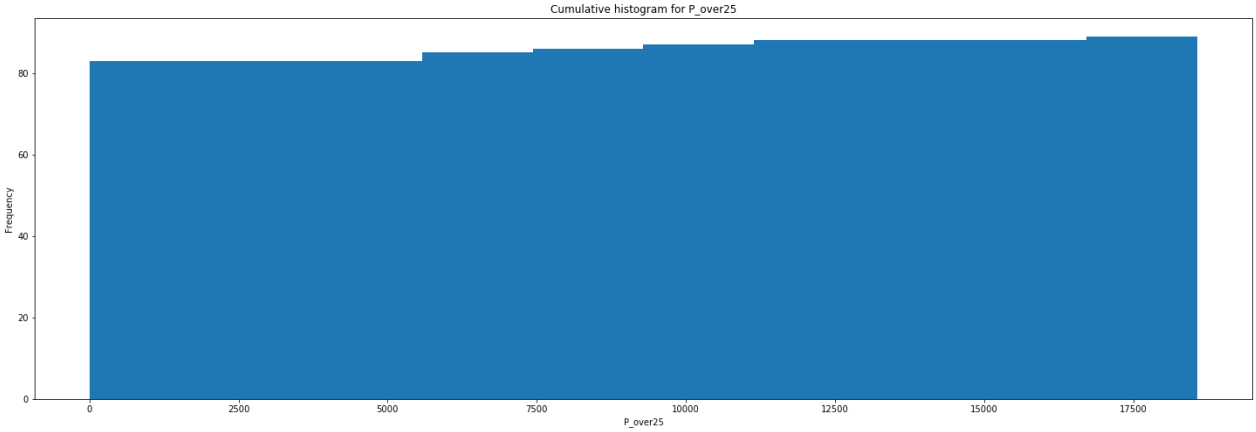
PIE CHART FOR COLUMN: Ur1



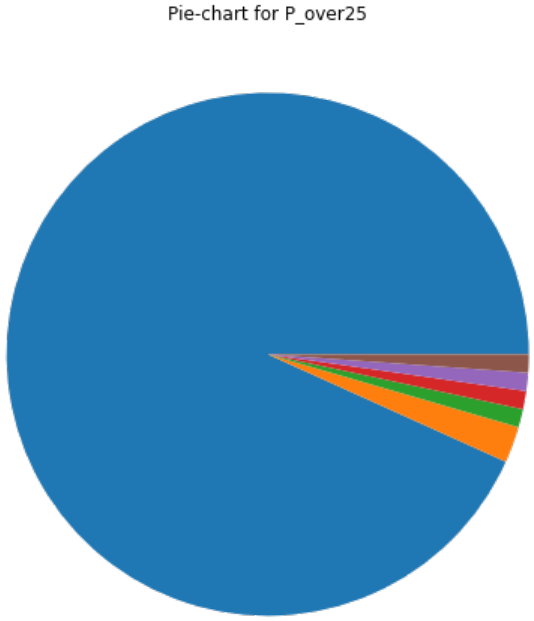
HISTOGRAM FOR COLUMN: P_over25



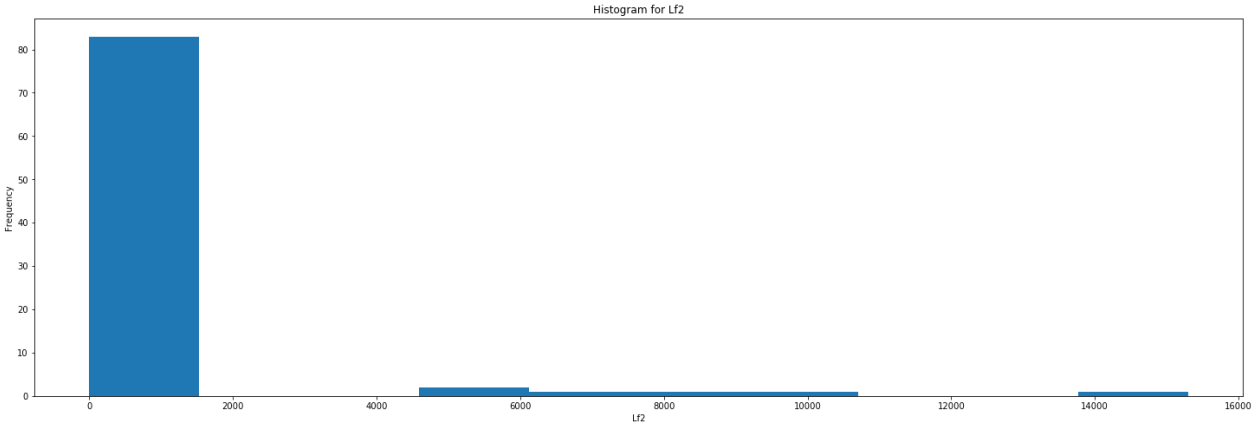
CUMULATIVE HISTOGRAM FOR COLUMN: P_over25



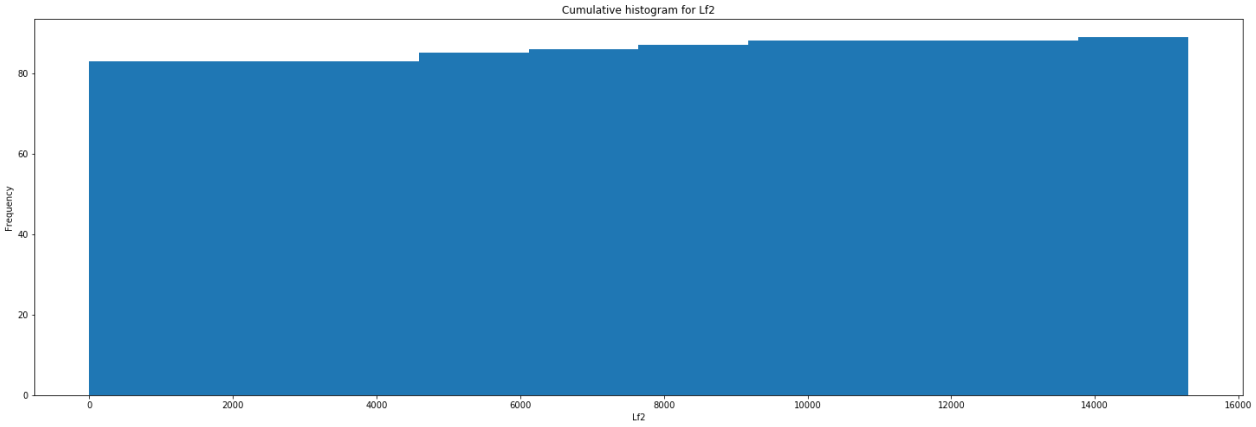
PIE CHART FOR COLUMN: P_over25



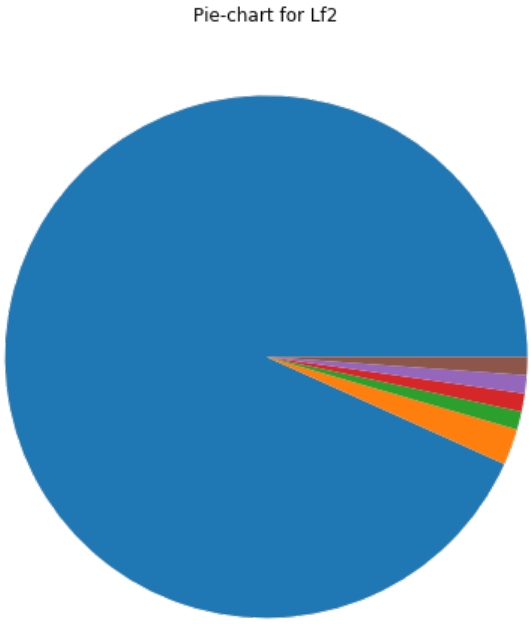
HISTOGRAM FOR COLUMN: Lf2



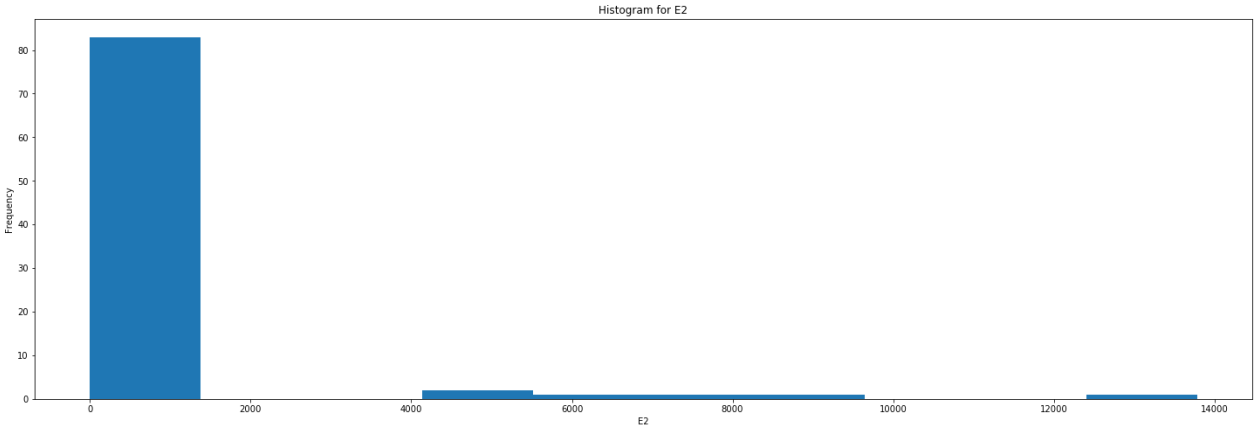
CUMULATIVE HISTOGRAM FOR COLUMN: Lf2



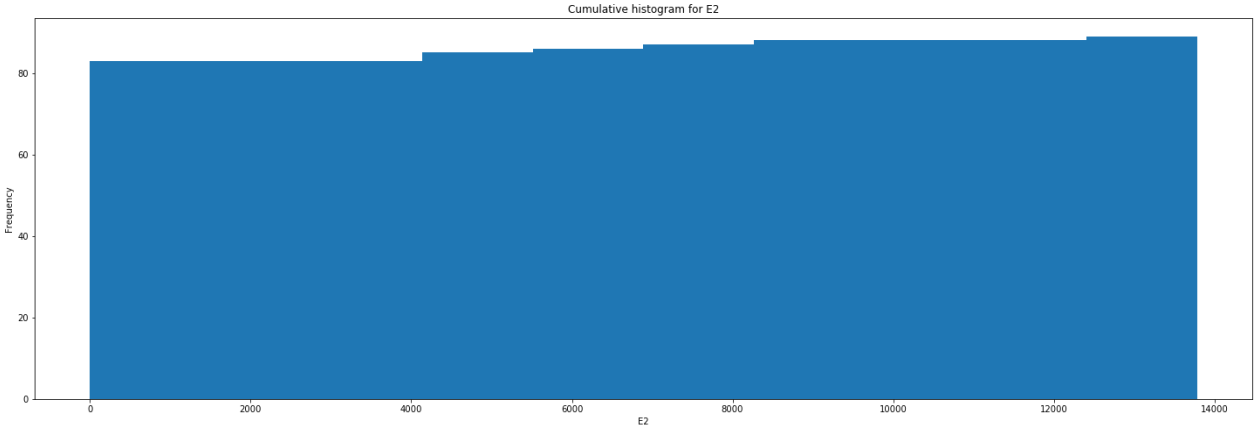
PIE CHART FOR COLUMN: Lf2



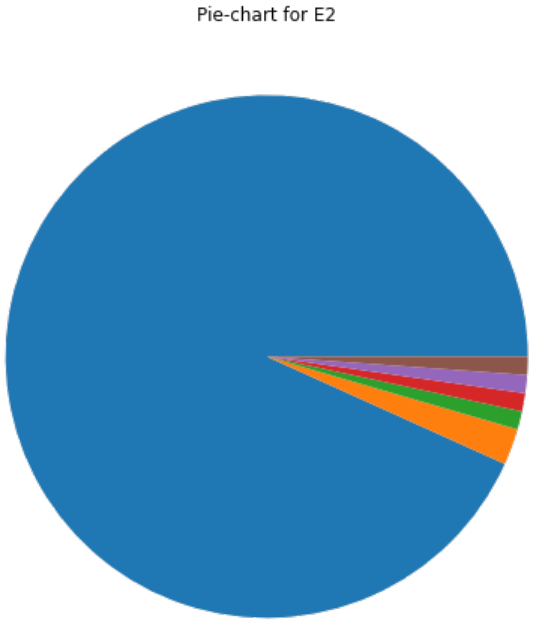
HISTOGRAM FOR COLUMN: E2



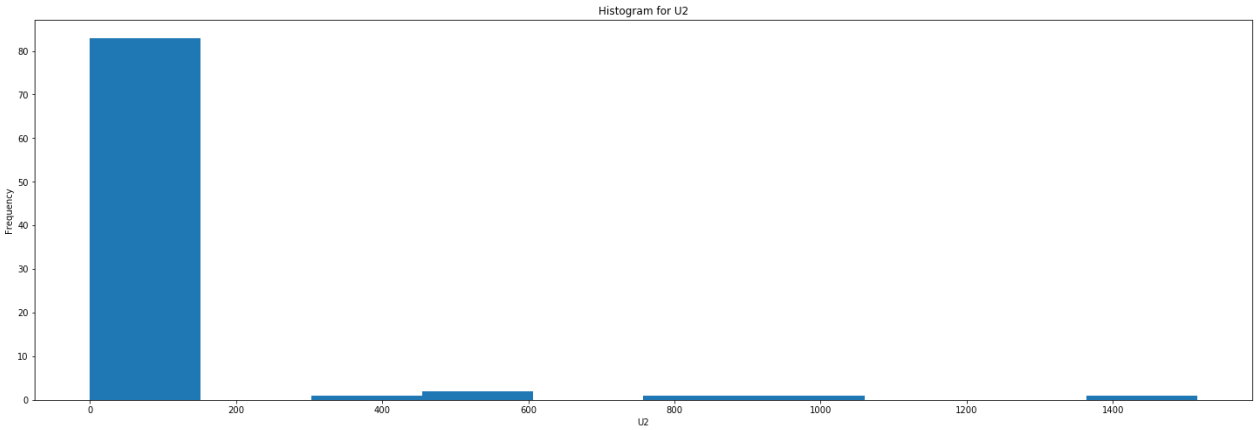
CUMULATIVE HISTOGRAM FOR COLUMN: E2



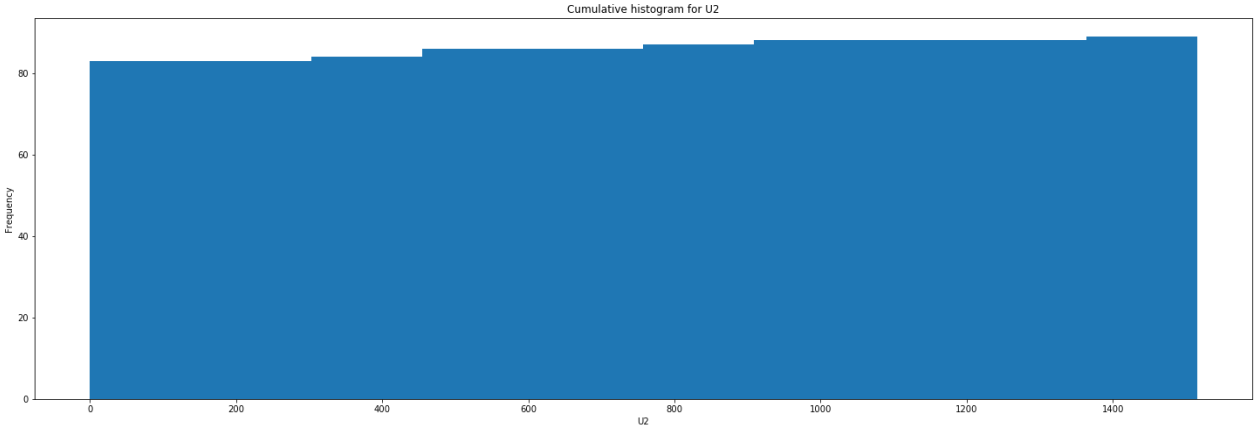
PIE CHART FOR COLUMN: E2



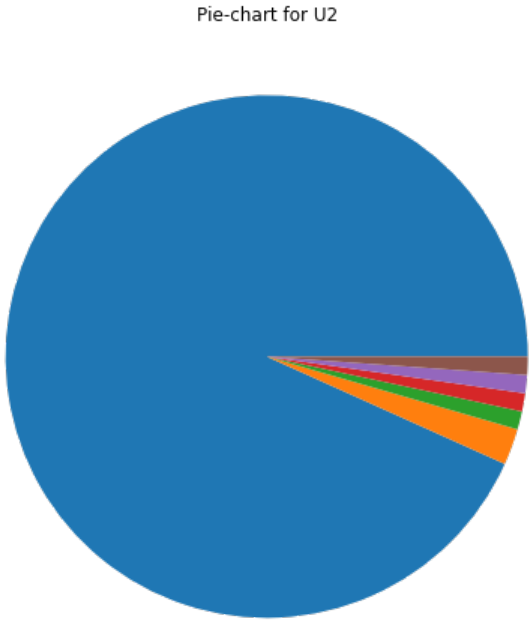
HISTOGRAM FOR COLUMN: U2



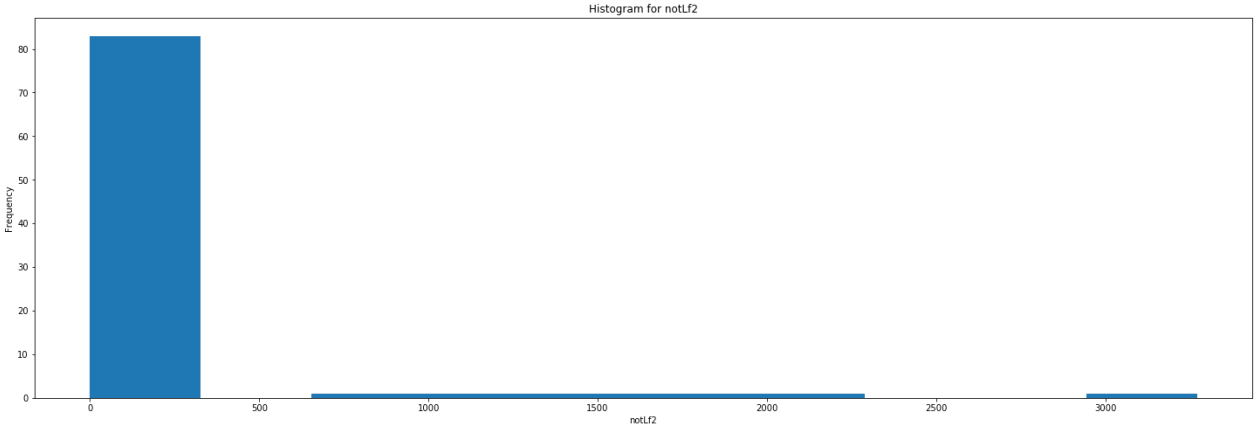
CUMULATIVE HISTOGRAM FOR COLUMN: U2



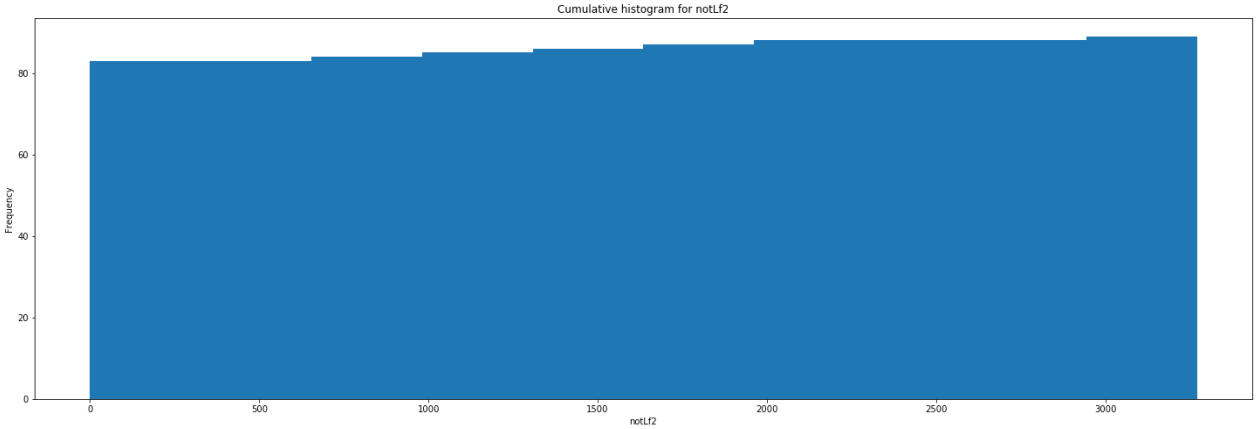
PIE CHART FOR COLUMN: U2



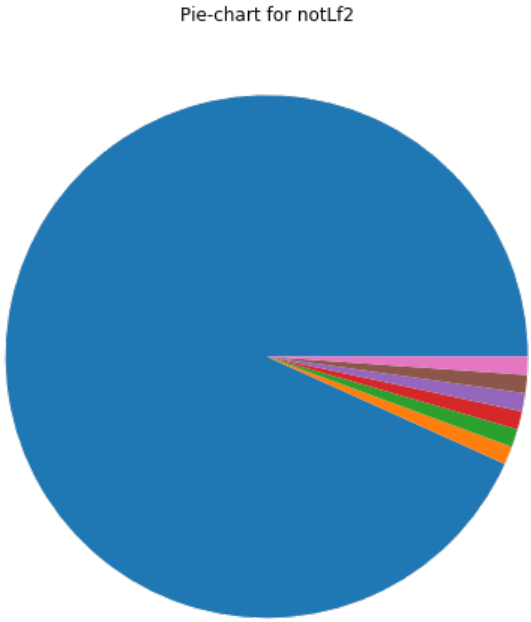
HISTOGRAM FOR COLUMN: notLf2



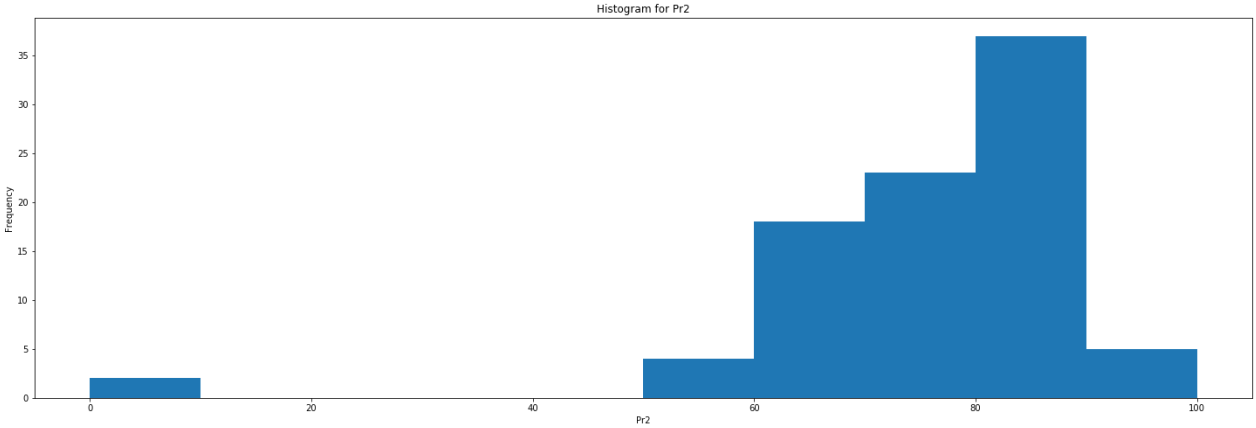
CUMULATIVE HISTOGRAM FOR COLUMN: notLf2



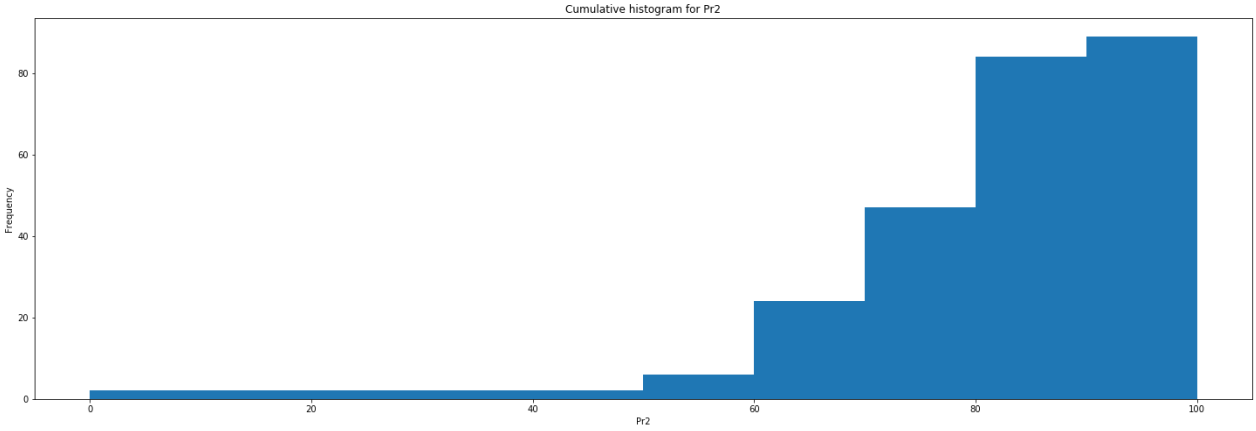
PIE CHART FOR COLUMN: notLf2



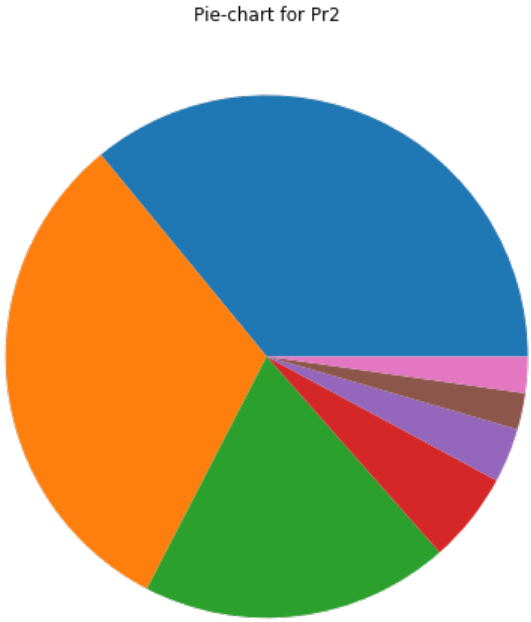
HISTOGRAM FOR COLUMN: Pr2



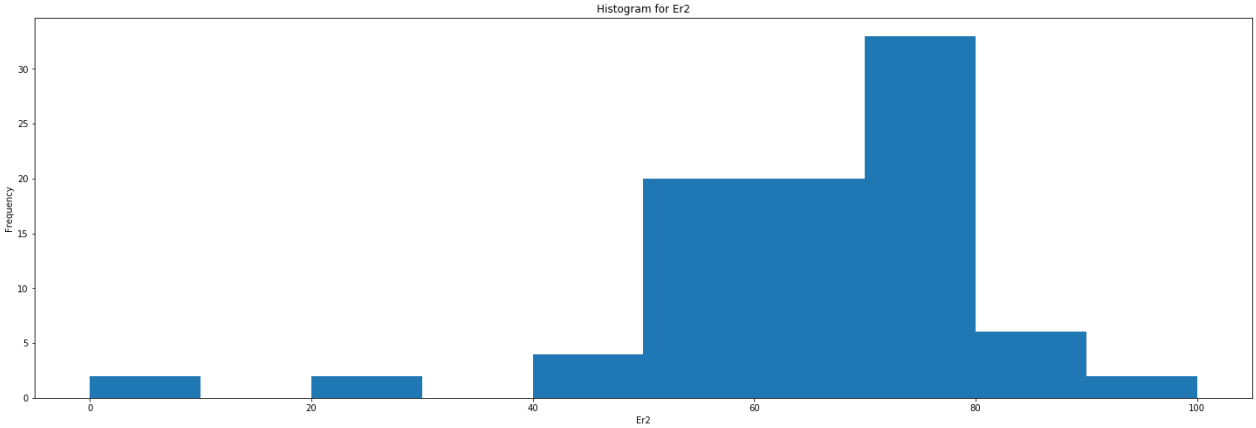
CUMULATIVE HISTOGRAM FOR COLUMN: Pr2



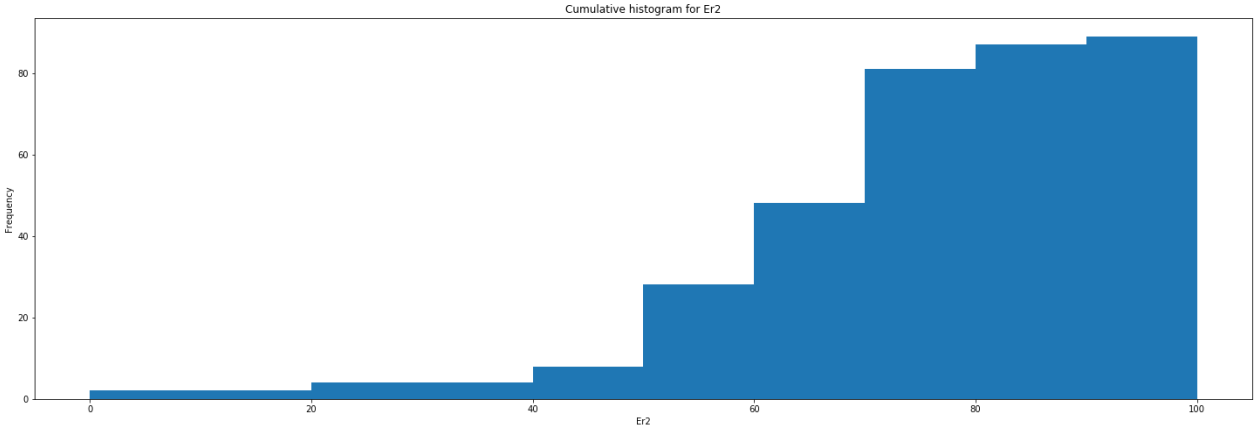
PIE CHART FOR COLUMN: Pr2



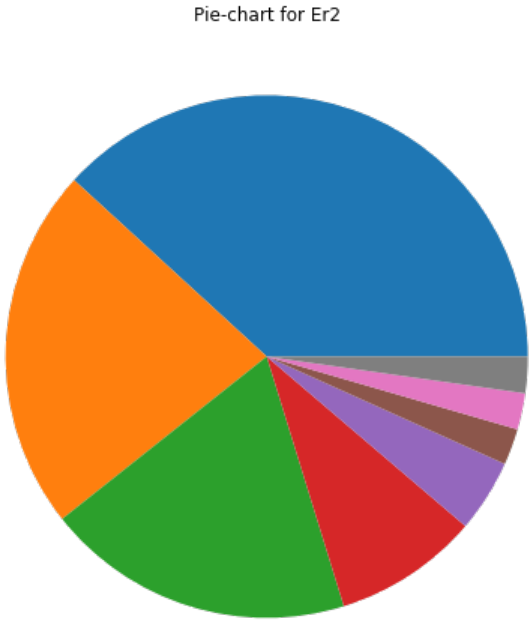
HISTOGRAM FOR COLUMN: Er2



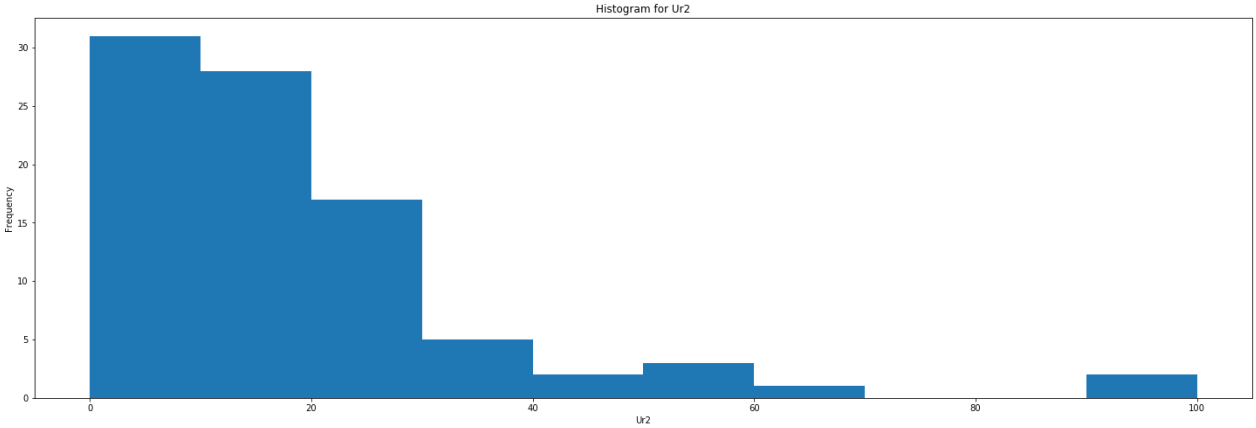
CUMULATIVE HISTOGRAM FOR COLUMN: Er2



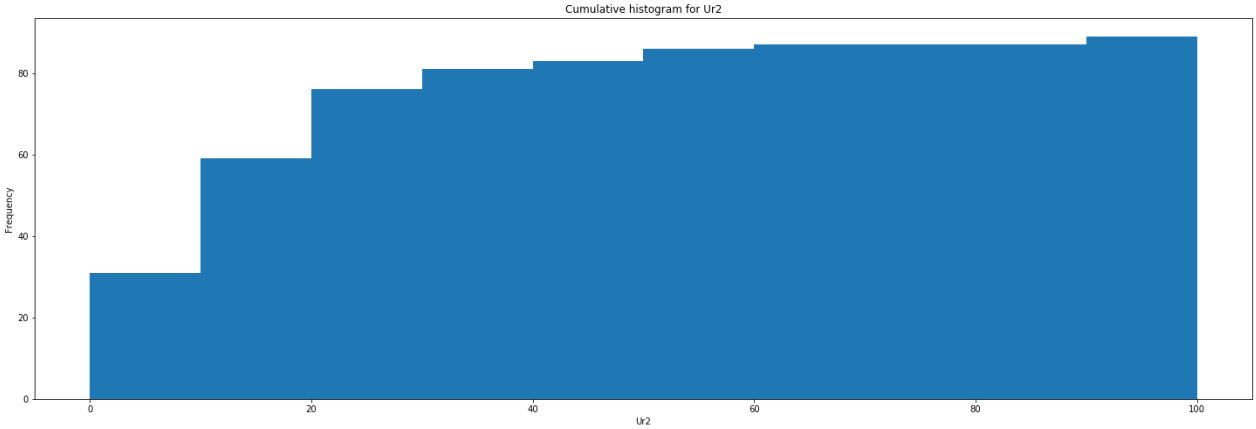
PIE CHART FOR COLUMN: Er2



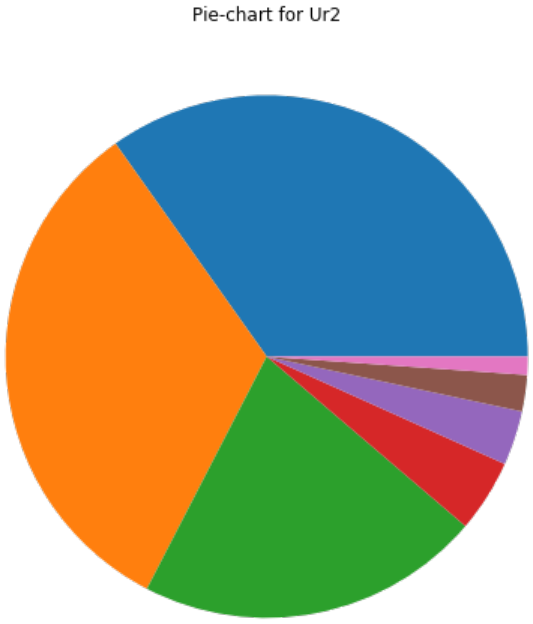
HISTOGRAM FOR COLUMN: Ur2



CUMULATIVE HISTOGRAM FOR COLUMN: Ur2



PIE CHART FOR COLUMN: Ur2



For every quantitative attribute, compute the mean, median, mode(s), variance, and standard deviation.

In [51]:
```
data.describe()
```

Out[51]:

	P_over15	Lf	E	U	notLf	Pr	Er	Ur	P_15to24	L
count	89.000000	89.000000	89.000000	89.000000	89.000000	89.000000	89.000000	89.000000	89.000000	89.0000
mean	1043.876404	830.955056	733.820225	97.640449	212.471910	74.925843	60.849438	18.529213	181.348315	122.8651
std	3371.332583	2703.188971	2403.570855	304.028903	670.247632	9.532177	15.359459	14.424941	600.978730	410.2015
min	10.000000	10.000000	0.000000	0.000000	0.000000	50.000000	0.000000	0.000000	0.000000	0.0000
25%	65.000000	45.000000	35.000000	10.000000	20.000000	69.200000	50.000000	10.000000	10.000000	10.0000
50%	135.000000	105.000000	80.000000	20.000000	35.000000	76.000000	62.000000	15.000000	25.000000	15.0000
75%	335.000000	255.000000	225.000000	40.000000	70.000000	80.800000	71.400000	24.400000	55.000000	35.0000
max	22485.000000	17945.000000	15860.000000	2085.000000	4535.000000	100.000000	100.000000	66.700000	3920.000000	2655.0000

8 rows × 24 columns

Mode for quantitative data.

In [52]:

```
for column in data.columns:
    if data[column].dtype != "object":
        df = pd.DataFrame(data[column])
        display(df.mode())
```

	P_over15
0	20
1	45

	Lf
0	15
1	30
2	105

	E
0	30

	U
0	10

	notLf
0	10

	Pr
0	83.3

	Er
0	50.0

	Ur
0	0.0

	P_15to24
0	0
1	10

	Lf1
0	10

	E1
0	10

	U1
0	0

	notLf1
0	0

	Pr1
0	0.0

	Er1
0	0.0

	Ur1
0	0.0

	P_over25
0	35

	Lf2
0	20

	E2
0	10
1	15
2	30
3	55

	U2
0	10

	notLf2
0	10

	Pr2
0	75.0

	Er2
0	50.0
1	60.0
2	66.7

	Ur2
0	0.0

Variance for quantitave data.

In [53]:
```
for column in data.columns:
    if data[column].dtype != "object":
        df = pd.DataFrame(data[column])
        display(df.var())
```

```
P_over15    1.136588e+07
dtype: float64
Lf    7.307231e+06
dtype: float64
E    5.777153e+06
dtype: float64
U    92433.5738
dtype: float64
notLf    449231.888407
dtype: float64
Pr    90.862393
dtype: float64
Er    235.912983
dtype: float64
Ur    208.07891
dtype: float64
P_15to24    361175.434116
dtype: float64
Lf1    168265.277068
dtype: float64
E1    103486.791369
dtype: float64
U1    7971.361083
dtype: float64
notLf1    36623.410368
dtype: float64
Pr1    965.663179
dtype: float64
```

```
Er1    846.529574
dtype: float64
Ur1    945.76105
dtype: float64
P_over25    7.678947e+06
dtype: float64
Lf2    5.263450e+06
dtype: float64
E2    4.337545e+06
dtype: float64
U2    46497.018641
dtype: float64
notLf2    229849.533963
dtype: float64
Pr2    242.178404
dtype: float64
Er2    261.866821
dtype: float64
Ur2    323.804155
dtype: float64
```

Median for quantitative data.

In [54]:

```
for column in data.columns:
    if data[column].dtype != "object":
        df = pd.DataFrame(data[column])
        display(df.median())
```

```
P_over15    135.0
dtype: float64
Lf    105.0
dtype: float64
E    80.0
dtype: float64
U    20.0
dtype: float64
notLf    35.0
dtype: float64
Pr    76.0
dtype: float64
Er    62.0
dtype: float64
Ur    15.0
dtype: float64
P_15to24    25.0
dtype: float64
Lf1    15.0
dtype: float64
E1    10.0
dtype: float64
U1    0.0
dtype: float64
notLf1    10.0
dtype: float64
Pr1    64.3
dtype: float64
Er1    42.9
dtype: float64
Ur1    0.0
dtype: float64
P_over25    115.0
dtype: float64
Lf2    90.0
dtype: float64
E2    70.0
dtype: float64
U2    15.0
dtype: float64
notLf2    30.0
dtype: float64
Pr2    78.6
dtype: float64
Er2    67.1
dtype: float64
Ur2    13.8
dtype: float64
```

For every quantitative attribute, compute exactly what percentage of instances are within one standard deviation, two standard deviations, and three standard deviations of the mean. DO NOT use Chebychev's Theorem of the Empirical Rule.

In [55]:

```
def mean(data):
    m = sum(data) / len(data)
    return m
def variance(data):
    n = len(data)
    mean = sum(data) / n
    deviations = [(x - mean) ** 2 for x in data]
    variance = sum(deviations) / n
    return variance
def std(data):
    s = math.sqrt(variance(data))
    return s
for column in data.columns:
    if data[column].dtype == 'float64':
        print("For column: ", column)
        s = std(data[column])
        m = mean(data[column])
        std3 = m + s * 3
        std_3 = m - s * 3
        std2 = m + s * 2
        std_2 = m - s * 2
        std1 = m + s
        std_1 = m - s
        c1 = 0
        c2 = 0
        c3 = 0
        for i in data[column]:
            if i >= std_3 and i <= std3:
                c3 += 1
            if i >= std_2 and i <= std2:
                c2 += 1
            if i >= std_1 and i <= std1:
                c1 += 1
        p1 = c1 / len(data[column]) * 100
        p2 = c2 / len(data[column]) * 100
        p3 = c3 / len(data[column]) * 100
        print("% of values within 1 standard deviation: ", p1)
        print("% of values within 2 standard deviations: ", p2)
        print("% of values within 3 standard deviations: ", p3)
        print("================================================")
```

```
For column:  Pr
% of values within 1 standard deviation:  75.28089887640449
% of values within 2 standard deviations:  94.3820224719101
% of values within 3 standard deviations:  100.0
================================================
For column:  Er
% of values within 1 standard deviation:  82.02247191011236
% of values within 2 standard deviations:  96.62921348314607
% of values within 3 standard deviations:  97.75280898876404
================================================
For column:  Ur
% of values within 1 standard deviation:  68.53932584269663
% of values within 2 standard deviations:  93.25842696629213
% of values within 3 standard deviations:  97.75280898876404
================================================
For column:  Pr1
% of values within 1 standard deviation:  68.53932584269663
% of values within 2 standard deviations:  100.0
% of values within 3 standard deviations:  100.0
================================================
For column:  Er1
% of values within 1 standard deviation:  61.79775280898876
% of values within 2 standard deviations:  94.3820224719101
% of values within 3 standard deviations:  100.0
================================================
For column:  Ur1
% of values within 1 standard deviation:  84.26966292134831
% of values within 2 standard deviations:  92.13483146067416
% of values within 3 standard deviations:  100.0
================================================
For column:  Pr2
% of values within 1 standard deviation:  87.64044943820225
% of values within 2 standard deviations:  97.75280898876404
% of values within 3 standard deviations:  97.75280898876404
================================================
For column:  Er2
% of values within 1 standard deviation:  84.26966292134831
% of values within 2 standard deviations:  93.25842696629213
% of values within 3 standard deviations:  97.75280898876404
================================================
For column:  Ur2
% of values within 1 standard deviation:  91.01123595505618
% of values within 2 standard deviations:  96.62921348314607
% of values within 3 standard deviations:  97.75280898876404
================================================
```

For every categorical attribute, compute the mode(s).

In [56]:

```
for column in data.columns:
    if data[column].dtype == "object":
        df = pd.DataFrame(data[column].value_counts())
        display(df.mode())
```

	Rc
0	3

	Rg
0	3

	Dn
0	1

	S
0	30

Explain what insights about the data you get from your analysis.

In general, majority of people over the age of 15 in the province of yukon tend to participate in labour force. The data also shows that they have high employement rate as well, with very few unemployed. It also seems that people over the age of 15 are more employed in the labour force in the province of yukon than people of age 15 ro 24 and over the age of 25 years.

In [ ]:

In [ ]:

In [ ]: