

In [1]: `#pip install pywaffle`

In [2]: `from pywaffle import Waffle`

In [3]: `import os  
import matplotlib.pyplot as plt  
import seaborn as sns`

Backing off send\_request(...) for 0.0s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: self signed certificate in certificate chain (\_ssl.c:1125)'))))  
Backing off send\_request(...) for 0.2s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: self signed certificate in certificate chain (\_ssl.c:1125)'))))

In [4]: `df=pd.read_csv(r'C:\Users\ckraft\Downloads\Employee.csv')  
df.head(10)`

Out[4]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDorr
0	Bachelors	2017	Bangalore	3	34	Male	No	
1	Bachelors	2013	Pune	1	28	Female	No	
2	Bachelors	2014	New Delhi	3	38	Female	No	
3	Masters	2016	Bangalore	3	27	Male	No	
4	Masters	2017	Pune	3	24	Male	Yes	
5	Bachelors	2016	Bangalore	3	22	Male	No	
6	Bachelors	2015	New Delhi	3	38	Male	No	
7	Bachelors	2016	Bangalore	3	34	Female	No	
8	Bachelors	2016	Pune	3	23	Male	No	
9	Masters	2017	New Delhi	2	37	Male	No	

Backing off send\_request(...) for 2.6s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: self signed certificate in certificate chain (\_ssl.c:1125)'))))

In [5]: `df.shape`

Out[5]: (4653, 9)

In [6]: `df.size`

Out[6]: 41877

In [7]: `df.info`

Out[7]: <bound method DataFrame.info of  
 e Gender EverBenched \ Education JoiningYear City PaymentTier Ag  
 0 Bachelors 2017 Bangalore 3 34 Male No  
 1 Bachelors 2013 Pune 1 28 Female No  
 2 Bachelors 2014 New Delhi 3 38 Female No  
 3 Masters 2016 Bangalore 3 27 Male No  
 4 Masters 2017 Pune 3 24 Male Yes  
 ... ..  
 4648 Bachelors 2013 Bangalore 3 26 Female No  
 4649 Masters 2013 Pune 2 37 Male No  
 4650 Masters 2018 New Delhi 3 27 Male No  
 4651 Bachelors 2012 Bangalore 3 30 Male Yes  
 4652 Bachelors 2015 Bangalore 3 33 Male Yes  
  
 ExperienceInCurrentDomain LeaveOrNot  
 0 0 0  
 1 3 1  
 2 2 0  
 3 5 1  
 4 2 1  
 ... ..  
 4648 4 0  
 4649 2 1  
 4650 5 1  
 4651 2 0  
 4652 4 0  
  
 [4653 rows x 9 columns]>

In [8]: `#stats on numerical data.....  
 round(df.describe(exclude = 'object'), 2)`

Out[8]:

	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
<b>count</b>	4653.00	4653.00	4653.00	4653.00	4653.00
<b>mean</b>	2015.06	2.70	29.39	2.91	0.34
<b>std</b>	1.86	0.56	4.83	1.56	0.48
<b>min</b>	2012.00	1.00	22.00	0.00	0.00
<b>25%</b>	2013.00	3.00	26.00	2.00	0.00
<b>50%</b>	2015.00	3.00	28.00	3.00	0.00
<b>75%</b>	2017.00	3.00	32.00	4.00	1.00
<b>max</b>	2018.00	3.00	41.00	7.00	1.00

In [9]: `#stats on categorical data  
 round(df.describe(exclude = ['float', 'int64']),2)`

Out[9]:

	Education	City	Gender	EverBenched
<b>count</b>	4653	4653	4653	4653
<b>unique</b>	3	3	2	2
<b>top</b>	Bachelors	Bangalore	Male	No
<b>freq</b>	3601	2228	2778	4175

In [10]: `df[df.duplicated()].shape`

Out[10]: (1889, 9)

In [11]: `df.drop_duplicates(inplace=True); #dropping dupes`

In [12]: `df.shape #cleaned up`

Out[12]: (2764, 9)

In [13]: `print(categorical_features)`

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4244\2475559238.py in <module>
----> 1 print(categorical_features)

NameError: name 'categorical_features' is not defined
```

In [ ]: `#What degress will we need in the future?
print(df["Education"].value_counts())
print('\n')`

In [ ]: `#distribution of edu
b = df["Education"].value_counts()
d = pd.DataFrame({'Education':b.index, 'Statistics':b.values})
sort_df = d.sort_values(by='Statistics', ascending=False)
plt.rcParams.update({'font.size': 15})
fig, ax = plt.subplots(figsize=(8,5))
colors = ['#d4210b', '#000000']
ax.plot(sort_df.Statistics, d.Education, "D", color='white', markersize=50)
ax.barh(sort_df.Education, width=sort_df.Statistics, color=colors)
plt.xlabel("Frequency of Education", fontsize = 15)
plt.ylabel("Education", fontsize = 15)
plt.title("Distribution of Education", fontsize = 20);`

In [ ]: `#battle of the sexes
print(df["Gender"].value_counts())
print('\n')`

```
In [ ]: #distribution of gender
b = df["Gender"].value_counts()
d = pd.DataFrame({'Gender':b.index, 'Statistics':b.values})
sort_df = d.sort_values(by='Statistics', ascending=False)
plt.rcParams.update({'font.size': 15})
fig, ax = plt.subplots(figsize=(8,5))
colors = ['#d4210b', '#000000']
ax.plot(sort_df.Statistics, d.Gender, "D", color='white', markersize=100)
ax.barh(sort_df.Gender, width=sort_df.Statistics, color=colors)
plt.xlabel("Frequency of Gender", fontsize = 15)
plt.ylabel("Gender", fontsize = 15)
plt.title("Distribution of Gender", fontsize = 20);
```

```
In [ ]: #what cities will haave highest concentration of brainy humans?
print(df["City"].value_counts())
print('\n')
```

```
In [ ]: #distribution of the cities
b = df["City"].value_counts()
d = pd.DataFrame({'City':b.index, 'Statistics':b.values})
sort_df = d.sort_values(by='Statistics', ascending=False)
plt.rcParams.update({'font.size': 15})
fig, ax = plt.subplots(figsize=(8,5))
colors = ['#d4210b', '#000000']
ax.plot(sort_df.Statistics, d.City, "D", color='white', markersize=48)
ax.barh(sort_df.City, width=sort_df.Statistics, color=colors)
plt.xlabel("Frequency of City", fontsize = 15)
plt.ylabel("City", fontsize = 15)
plt.title("Distribution of City", fontsize = 20);
```

```
In [ ]: #endurance- stay or Leave
print(df["LeaveOrNot"].value_counts())
print('\n')
```

```
In [ ]: b = df["LeaveOrNot"].value_counts()
d = pd.DataFrame({'LeaveOrNot':b.index, 'Statistics':b.values})
sort_df = d.sort_values(by='Statistics', ascending=False)
plt.rcParams.update({'font.size': 15})
fig, ax = plt.subplots(figsize=(8,5))
colors = ['#d4210b', '#000000']
ax.plot(sort_df.Statistics, d.LeaveOrNot, "D", color='white', markersize=75)
ax.barh(sort_df.LeaveOrNot, width=sort_df.Statistics, color=colors)
plt.xlabel("Frequency of Leave Or Not", fontsize = 15)
plt.ylabel("Leave Or Not", fontsize = 15)
plt.title("Distribution of Leave Or Not", fontsize = 20);
plt.ylim(-0.5,1.5)
```

```
In [ ]: #start
print(df["JoiningYear"].value_counts())
print('\n')
```

```
In [ ]: b = df["JoiningYear"].value_counts()
d = pd.DataFrame({'JoiningYear':b.index, 'Statistics':b.values})
```

```

sort_df = d.sort_values(by='Statistics', ascending=False)
plt.rcParams.update({'font.size': 15})
fig, ax = plt.subplots(figsize=(8,5))
colors = ['#d4210b', '#000000']
ax.plot(sort_df.Statistics, d.JoiningYear, "D", color='white', markersize=22)
ax.barh(sort_df.JoiningYear, width=sort_df.Statistics, color=colors)
plt.xlabel("Frequency of Joining Year", fontsize = 15)
plt.ylabel("Joining Year", fontsize = 15)
plt.title("Distribution of Joining Year", fontsize = 20);

```

```

In [ ]: #when did adulting begin ?
print(df["Age"].value_counts().sort_values(ascending = True))
print('\n')

```

```

In [ ]: #adult-ers
b = df["Age"].value_counts()
d = pd.DataFrame({'Age':b.index, 'Statistics':b.values})
sort_df = d.sort_values(by='Statistics', ascending=False)
plt.rcParams.update({'font.size': 15})
fig, ax = plt.subplots(figsize=(8,5))
colors = ['#d4210b', '#000000']
ax.plot(sort_df.Statistics, d.Age, "D", color='white', markersize=7)
ax.barh(sort_df.Age, width=sort_df.Statistics, color=colors)
plt.xlabel("Frequency of Age", fontsize = 15)
plt.ylabel("Age", fontsize = 15)
plt.title("Distribution of Age", fontsize = 20);
ax.set_ylim(bottom=21, top=43, emit=True)

```

```

In [14]: pd.crosstab(df['Education'],df['LeaveOrNot'],margins=True)

```

```

Out[14]:
LeaveOrNot    0    1   All
Education
Bachelors  1232  739  1971
Masters    328  309   637
PHD        116   40   156
All       1676 1088  2764

```

```

In [15]:
leave_gen = df[df['LeaveOrNot'] == 1]['Gender'].value_counts()
notleave_gen = df[df['LeaveOrNot'] == 0]['Gender'].value_counts()

female = df['Gender'].value_counts().values[0]
male = df['Gender'].value_counts().values[1]

leave_female = int(round(leave_gen.values[0] / female * 100, 0))
leave_male = int(round(leave_gen.values[1] / male * 100, 0))
notleave_female = int(round(notleave_gen.values[0] / female * 100, 0))
notleave_male = int(round(notleave_gen.values[1] / male * 100, 0))

female_per = int(round(female / (female + male) * 100, 0))
male_per = int(round(male / (female + male) * 100, 0))

```

In [16]:

```

#Loyalty
fig = plt.figure(FigureClass = Waffle,
                 constrained_layout = True,
                 figsize = (6,6),
                 facecolor = '#ffffff',dpi = 108,

                 plots = {'121':
                        {
                            'rows':7,
                            'columns': 7,
                            'values' : [notleave_male,leave_male],
                            'colors' : ['#000000','#d4210b'],
                            'vertical' : True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'male',
                            'icon_legend': False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '122' :
                        {
                            'rows': 7,
                            'columns':7,
                            'values':[notleave_female,leave_female],
                            'colors' : ['#000000','#d4210b'],
                            'vertical': True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'female',
                            'icon_legend' :False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        }
                    },

                )

fig.text(0.03, 0.8, 'How is the distribution of LeaveOrNot by Gender?', {'font':'Georgia'
fig.text(0.21, 0.21, '{}%'.format(notleave_male), {'font':'Georgia', 'size':20,'weight'
fig.text(0.68, 0.21, '{}%'.format(notleave_female), {'font':'Georgia', 'size':20,'weigh
fig.text(0.16, 0.74, 'Male ({}%)'.format(male_per), {'font':'Georgia', 'size':14,'weigh
fig.text(0.62, 0.74, 'Female({}% )'.format(female_per), {'font':'Georgia', 'size':14,'we
fig.text(0.915,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '12','weight':
fig.text(1.01,0.73, '|', {'color':'black' , 'size':'12', 'weight': 'bold'})
fig.text(1.02,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '12','style
fig.show()

```

C:\Users\ckraft\Anaconda3\lib\site-packages\pywaffle\waffle.py:394: MatplotlibDeprecatio  
nWarning: Passing non-integers as three-element position specification is deprecated sin  
ce 3.3 and will be removed two minor releases later.

self.ax = self.add\_subplot(loc, aspect="equal")

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\3659303970.py:49: MatplotlibDeprecatio  
nWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed

two minor releases later

```
fig.text(0.915,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '12','weight':'bold','style':'normal', 'color':'#d4210b'})
```

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\3659303970.py:51: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later

```
fig.text(1.02,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '12','style':'normal', 'weight':'bold','color':'#000000'},alpha = 1)
```

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\3659303970.py:52: UserWarning: Matplotlib is currently using module://matplotlib\_inline.backend\_inline, which is a non-GUI backend, so cannot show the figure.

```
fig.show()
```

C:\Users\ckraft\Anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning: This figure was using constrained\_layout, but that is incompatible with subplots\_adjust and/or tight\_layout; disabling constrained\_layout.

```
fig.canvas.print_figure(bytes_io, **kw)
```

## How is the distribution of LeaveOrNot by Gender?

Male (45%)

Female(55%)

Leave |Notleave



50%

69%

```
In [17]: pd.crosstab(df['EverBenched'],df['LeaveOrNot'],margins=True)
```

```
Out[17]:
```

	LeaveOrNot	0	1	All
EverBenched				
No		1474	929	2403
Yes		202	159	361
All		1676	1088	2764

```
In [18]:
leave_gen = df[df['LeaveOrNot'] == 1]['EverBenched'].value_counts()
notleave_gen = df[df['LeaveOrNot'] == 0]['EverBenched'].value_counts()

no = df['EverBenched'].value_counts().values[0]
yes = df['EverBenched'].value_counts().values[1]

leave_no = int(round(leave_gen.values[0] / no * 100, 0))
leave_yes = int(round(leave_gen.values[1] / yes * 100, 0))
notleave_no = int(round(notleave_gen.values[0] / no * 100, 0))
notleave_yes = int(round(notleave_gen.values[1] / yes * 100, 0))
```

```
no_per = int(round(no/(no+yes) * 100, 0))
yes_per = int(round(yes/(no+yes)* 100, 0))
```

In [19]:

```
fig = plt.figure(FigureClass = Waffle,
                 constrained_layout = True,
                 figsize = (7,7),
                 facecolor = '#ffffff',dpi = 100,

                 plots = {'121':
                        {
                            'rows':7,
                            'columns': 7,
                            'values' : [notleave_yes,leave_yes],
                            'colors' : ['#000000','#d4210b'],
                            'vertical' : True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'thumbs-up',
                            'icon_legend': False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '122' :
                        {
                            'rows': 7,
                            'columns':7,
                            'values':[notleave_no,leave_no],
                            'colors' : ['#000000','#d4210b'],
                            'vertical': True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'thumbs-down',
                            'icon_legend' :False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        }
                    },

                 )

fig.text(0., 0.8, 'How is the distribution of LeaveOrNot by EverBenched?', {'font':'Geo
fig.text(0.21, 0.21, '{}%'.format(notleave_yes), {'font':'Georgia', 'size':20,'weight':
fig.text(0.68, 0.21, '{}%'.format(notleave_no), {'font':'Georgia', 'size':20,'weight':
fig.text(0.08, 0.735, 'EverBenched_Yes ({}%)'.format(yes_per), {'font':'Georgia', 'size
fig.text(0.55, 0.735, 'EverBenched_No({}%)'.format(no_per), {'font':'Georgia', 'size':1
fig.text(0.945,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '14','weight':
fig.text(1.05,0.73, '|', {'color':'black' , 'size':'14', 'weight': 'bold'})
fig.text(1.07,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '14','style
fig.show()
```

C:\Users\ckraft\Anaconda3\lib\site-packages\pywaffle\waffle.py:394: MatplotlibDeprecatio  
nWarning: Passing non-integers as three-element position specification is deprecated sin  
ce 3.3 and will be removed two minor releases later.

```
self.ax = self.add_subplot(loc, aspect="equal")
```



11/11/21, 1:25 AM

Employee

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\1510933394.py:48: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later  
fig.text(0.945,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '14','weight':'bold','style':'normal', 'color':'#d4210b'})  
C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\1510933394.py:50: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later  
fig.text(1.07,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '14','style':'normal', 'weight':'bold','color':'#000000'},alpha = 1)  
C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\1510933394.py:51: UserWarning: Matplotlib is currently using module://matplotlib\_inline.backend\_inline, which is a non-GUI backend, so cannot show the figure.  
fig.show()  
C:\Users\ckraft\Anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning: This figure was using constrained\_layout, but that is incompatible with subplots\_adjust and/or tight\_layout; disabling constrained\_layout.  
fig.canvas.print\_figure(bytes\_io, \*\*kw)  
Backing off send\_request(...) for 4.9s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verification failed: self signed certificate in certificate chain (\_ssl.c:1125))))))

## How is the distribution of LeaveOrNot by EverBenchd?

EverBenchd\_Yes (13%)

EverBenchd\_No(87%)

Leave | Notleave

EverBenchd	LeaveOrNot	Count	Percentage
Yes	Leave	17	13%
No	Notleave	224	87%

56%

61%

In [20]:

pd.crosstab(df['Age'],df['LeaveOrNot'],margins=True)

Out[20]:

LeaveOrNot	0	1	All
Age			
22	17	14	31
23	28	13	41
24	80	81	161
25	79	88	167
26	85	94	179
27	116	102	218
28	224	141	365

localhost:8888/nbconvert/html/Desktop/My Experiments/Employee.ipynb?download=false

9/23

LeaveOrNot	0	1	All
Age			
29	116	64	180
30	112	74	186
31	80	35	115
32	65	48	113
33	77	37	114
34	80	38	118
35	68	42	110
36	80	41	121
37	77	42	119
38	82	35	117
39	79	36	115
40	79	40	119
41	52	23	75
All	1676	1088	2764

```
In [21]: pd.crosstab(df['JoiningYear'],df['LeaveOrNot'],margins=True)
```

Out[21]:

LeaveOrNot	0	1	All
JoiningYear			
2012	225	83	308
2013	225	171	396
2014	266	119	385
2015	272	192	464
2016	222	88	310
2017	461	201	662
2018	5	234	239
All	1676	1088	2764

```
In [22]: df['JoiningYear'].value_counts().values[6]
```

Out[22]: 239

```
In [23]: leave_gen = df[df['LeaveOrNot'] == 1]['JoiningYear'].value_counts()
notleave_gen = df[df['LeaveOrNot'] == 0]['JoiningYear'].value_counts()
```

```

Year_2017 = df['JoiningYear'].value_counts().values[0]
Year_2015 = df['JoiningYear'].value_counts().values[1]
Year_2013 = df['JoiningYear'].value_counts().values[2]
Year_2014 = df['JoiningYear'].value_counts().values[3]
Year_2016 = df['JoiningYear'].value_counts().values[4]
Year_2012 = df['JoiningYear'].value_counts().values[5]
Year_2018 = df['JoiningYear'].value_counts().values[6]

leave_Year_2017 = int(round(leave_gen.values[0] / Year_2017 * 100, 0))
leave_Year_2015 = int(round(leave_gen.values[1] / Year_2015 * 100, 0))
leave_Year_2013 = int(round(leave_gen.values[2] / Year_2013 * 100, 0))
leave_Year_2014 = int(round(leave_gen.values[3] / Year_2014 * 100, 0))
leave_Year_2016 = int(round(leave_gen.values[4] / Year_2016 * 100, 0))
leave_Year_2012 = int(round(leave_gen.values[5] / Year_2012 * 100, 0))
leave_Year_2018 = int(round(leave_gen.values[6] / Year_2018 * 100, 0))

notleave_Year_2017 = int(round(notleave_gen.values[0] / Year_2017 * 100, 0))
notleave_Year_2015 = int(round(notleave_gen.values[1] / Year_2015 * 100, 0))
notleave_Year_2013 = int(round(notleave_gen.values[2] / Year_2013 * 100, 0))
notleave_Year_2014 = int(round(notleave_gen.values[3] / Year_2014 * 100, 0))
notleave_Year_2016 = int(round(notleave_gen.values[4] / Year_2016 * 100, 0))
notleave_Year_2012 = int(round(notleave_gen.values[5] / Year_2012 * 100, 0))
notleave_Year_2018 = int(round(notleave_gen.values[6] / Year_2018 * 100, 0))

Year_2017_per = int(round(Year_2017/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+
Year_2015_per = int(round(Year_2015/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+
Year_2013_per = int(round(Year_2013/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+
Year_2014_per = int(round(Year_2014/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+
Year_2016_per = int(round(Year_2016/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+
Year_2012_per = int(round(Year_2012/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+
Year_2018_per = int(round(Year_2018/(Year_2017+Year_2015+Year_2013+Year_2014+Year_2016+

```

In [24]:

```

#turnover rate by seniority
fig = plt.figure(FigureClass = Waffle,
                  constrained_layout = True,
                  figsize = (24,15),
                  facecolor = '#ffffff',dpi = 100,

                  plots = {'171':
                          {
                              'rows':7,
                              'columns': 7,
                              'values' : [notleave_Year_2017,leave_Year_2017],
                              'colors' : ['#000000','#d4210b'],
                              'vertical' : True,
                              'interval_ratio_y': 3.95,
                              'interval_ratio_x': 3.95,
                              'icons' : 'calendar-alt',
                              'icon_legend': False,
                              'icon_size':25,
                              'plot_anchor':'C',
                              'alpha':0.1
                          },

                          '172' :
                          {
                              'rows': 7,
                              'columns':7,
                              'values': [notleave_Year_2015,leave_Year_2015],

```

```

        'colors' : ['#000000','#d4210b'],
        'vertical': True,
        'interval_ratio_y': 3.95,
        'interval_ratio_x': 3.95,
        'icons' : 'calendar-alt',
        'icon_legend' :False,
        'icon_size':25,
        'plot_anchor':'C',
        'alpha':0.1

    },

    '173' :
    {
        'rows': 7,
        'columns':7,
        'values':[notleave_Year_2013,leave_Year_2013],
        'colors' : ['#000000','#d4210b'],
        'vertical': True,
        'interval_ratio_y': 3.95,
        'interval_ratio_x': 3.95,
        'icons' : 'calendar-alt',
        'icon_legend' :False,
        'icon_size':25,
        'plot_anchor':'C',
        'alpha':0.1

    },

    '174' :
    {
        'rows': 7,
        'columns':7,
        'values':[notleave_Year_2014,leave_Year_2014],
        'colors' : ['#000000','#d4210b'],
        'vertical': True,
        'interval_ratio_y': 3.95,
        'interval_ratio_x': 3.95,
        'icons' : 'calendar-alt',
        'icon_legend' :False,
        'icon_size':25,
        'plot_anchor':'C',
        'alpha':0.1

    },

    '175' :
    {
        'rows': 7,
        'columns':7,
        'values':[notleave_Year_2016,leave_Year_2016],
        'colors' : ['#000000','#d4210b'],
        'vertical': True,
        'interval_ratio_y': 3.95,
        'interval_ratio_x': 3.95,
        'icons' : 'calendar-alt',
        'icon_legend' :False,
        'icon_size':25,
        'plot_anchor':'C',
        'alpha':0.1

    },

    '176' :

```

```

{
    'rows': 7,
    'columns': 7,
    'values': [notleave_Year_2012, leave_Year_2012],
    'colors': ['#000000', '#d4210b'],
    'vertical': True,
    'interval_ratio_y': 3.95,
    'interval_ratio_x': 3.95,
    'icons': 'calendar-alt',
    'icon_legend': False,
    'icon_size': 25,
    'plot_anchor': 'C',
    'alpha': 0.1
},
'177' :
{
    'rows': 7,
    'columns': 7,
    'values': [notleave_Year_2018, leave_Year_2018],
    'colors': ['#000000', '#d4210b'],
    'vertical': True,
    'interval_ratio_y': 3.95,
    'interval_ratio_x': 3.95,
    'icons': 'calendar-alt',
    'icon_legend': False,
    'icon_size': 25,
    'plot_anchor': 'C',
    'alpha': 0.1
},
}

)

fig.text(0.001, 0.8, 'How is the distribution of Leave Or Not by Joining Year?', {'font': 'Georgia', 'size': 22, 'weight': 'bold'})
fig.text(0.06, 0.37, '{}%'.format(notleave_Year_2017), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.205, 0.37, '{}%'.format(notleave_Year_2015), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.34, 0.37, '{}%'.format(notleave_Year_2013), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.485, 0.37, '{}%'.format(notleave_Year_2014), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.635, 0.37, '{}%'.format(notleave_Year_2016), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.77, 0.37, '{}%'.format(notleave_Year_2012), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.92, 0.37, '{}%'.format(notleave_Year_2018), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})

fig.text(0.015, 0.63, 'Year_2017({})'.format(Year_2017_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.16, 0.63, 'Year_2015({})'.format(Year_2015_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.305, 0.63, 'Year_2013({})'.format(Year_2013_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.44, 0.63, 'Year_2014({})'.format(Year_2014_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.58, 0.63, 'Year_2016({})'.format(Year_2016_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.72, 0.63, 'Year_2012({})'.format(Year_2012_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})
fig.text(0.86, 0.63, 'Year_2018({})'.format(Year_2018_per), {'font': 'Georgia', 'size': 22, 'weight': 'normal'})

fig.text(0.76, 0.73, 'Leave', {'font': 'Georgia', 'weight': 'bold', 'Size': '26', 'weight': 'bold'})
fig.text(0.81, 0.73, '|', {'color': 'black', 'size': '26', 'weight': 'bold'})
fig.text(0.815, 0.73, 'Notleave', {'font': 'Georgia', 'weight': 'bold', 'Size': '26', 'style': 'italic'})
fig.show()

```

C:\Users\ckraft\Anaconda3\lib\site-packages\pywaffle\waffle.py:394: MatplotlibDeprecationWarning: Passing non-integers as three-element position specification is deprecated since

ce 3.3 and will be removed two minor releases later.

```
self.ax = self.add_subplot(loc, aspect="equal")
```

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\3964328810.py:142: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later

```
fig.text(0.76,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '26','weight':'bold','style':'normal', 'color':'#d4210b'})
```

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\3964328810.py:144: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later

```
fig.text(0.815,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '26','style':'normal', 'weight':'bold','color':'#000000'},alpha = 1)
```

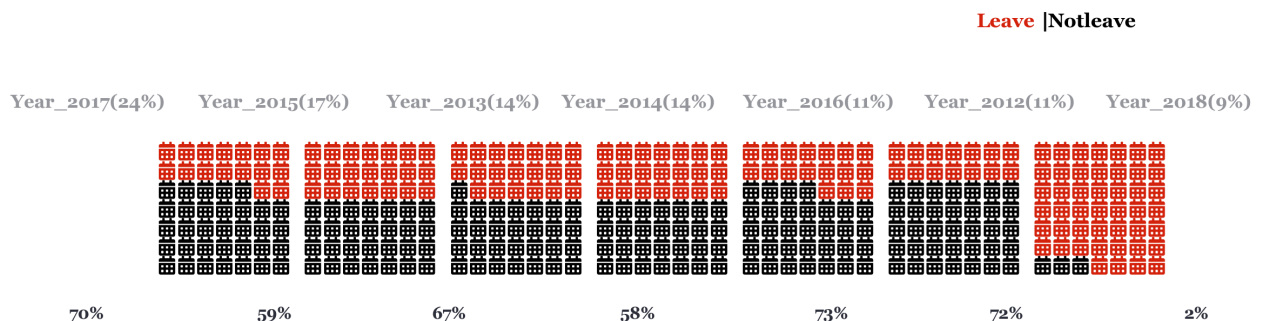
C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\3964328810.py:145: UserWarning: Matplotlib is currently using module://matplotlib\_inline.backend\_inline, which is a non-GUI backend, so cannot show the figure.

```
fig.show()
```

C:\Users\ckraft\Anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning: This figure was using constrained\_layout, but that is incompatible with subplots\_adjust and/or tight\_layout; disabling constrained\_layout.

```
fig.canvas.print_figure(bytes_io, **kw)
```

## How is the distribution of Leave Or Not by Joining Year?



```
In [25]: pd.crosstab(df['PaymentTier'],df['LeaveOrNot'])
```

```
Out[25]:
```

	LeaveOrNot	0	1
PaymentTier			
1	141	77	
2	227	343	
3	1308	668	

```
In [26]: leave_gen = df[df['LeaveOrNot'] == 1]['PaymentTier'].value_counts()
notleave_gen = df[df['LeaveOrNot'] == 0]['PaymentTier'].value_counts()

PaymentTier_3 = df['PaymentTier'].value_counts().values[0]
PaymentTier_2 = df['PaymentTier'].value_counts().values[1]
PaymentTier_1 = df['PaymentTier'].value_counts().values[2]

leave_PaymentTier_3 = int(round(leave_gen.values[0] / PaymentTier_3 * 100, 0))
leave_PaymentTier_2 = int(round(leave_gen.values[1] / PaymentTier_2 * 100, 0))
leave_PaymentTier_1 = int(round(leave_gen.values[2] / PaymentTier_1 * 100, 0))
notleave_PaymentTier_3 = int(round(notleave_gen.values[0] / PaymentTier_3 * 100, 0))
notleave_PaymentTier_2 = int(round(notleave_gen.values[1] / PaymentTier_2 * 100, 0))
notleave_PaymentTier_1 = int(round(notleave_gen.values[2] / PaymentTier_1 * 100, 0))
```

```

PaymentTier3_per = int(round(PaymentTier_3/(PaymentTier_3+PaymentTier_2+PaymentTier_1)
PaymentTier2_per = int(round(PaymentTier_2/(PaymentTier_3+PaymentTier_2+PaymentTier_1)*
PaymentTier1_per = int(round(PaymentTier_1/(PaymentTier_3+PaymentTier_2+PaymentTier_1)*

```

In [27]:

```

#does pay tier matter to retention?
fig = plt.figure(FigureClass = Waffle,
                 constrained_layout = True,
                 figsize = (10,7),
                 facecolor = '#ffffff',dpi = 100,

                 plots = {'131':
                        {
                            'rows':7,
                            'columns': 7,
                            'values' : [notleave_PaymentTier_3,leave_PaymentTier_3],
                            'colors' : ['#000000','#d4210b'],
                            'vertical' : True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'money-bill-alt',
                            'icon_legend': False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '132' :
                        {
                            'rows': 7,
                            'columns':7,
                            'values':[notleave_PaymentTier_2,leave_PaymentTier_2],
                            'colors' : ['#000000','#d4210b'],
                            'vertical': True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'money-bill-alt',
                            'icon_legend' :False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '133' :
                        {
                            'rows': 7,
                            'columns':7,
                            'values':[notleave_PaymentTier_1,leave_PaymentTier_1],
                            'colors' : ['#000000','#d4210b'],
                            'vertical': True,
                            'interval_ratio_y': 0.1,
                            'interval_ratio_x': 0.1,
                            'icons' : 'money-bill-alt',
                            'icon_legend' :False,
                            'icon_size':20,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },
                    },})

```

)

```
fig.text(0.02, 0.8, 'How is the distribution of Leave Or Not by Payment Tier?', {'font': 'Georgia', 'size': 20})
fig.text(0.12, 0.23, '{}%'.format(notleave_PaymentTier_3), {'font': 'Georgia', 'size': 20})
fig.text(0.46, 0.23, '{}%'.format(notleave_PaymentTier_2), {'font': 'Georgia', 'size': 20})
fig.text(0.78, 0.23, '{}%'.format(notleave_PaymentTier_1), {'font': 'Georgia', 'size': 20})
fig.text(0.08, 0.72, 'PaymentTier3 ({}%)'.format(PaymentTier3_per), {'font': 'Georgia', 'size': 12})
fig.text(0.40, 0.72, 'PaymentTier2({}% )'.format(PaymentTier2_per), {'font': 'Georgia', 'size': 12})
fig.text(0.725, 0.72, 'PaymentTier1({}% )'.format(PaymentTier1_per), {'font': 'Georgia', 'size': 12})
fig.text(0.955, 0.73, 'Leave', {'font': 'Georgia', 'weight': 'bold', 'Size': '12', 'weight': 'bold'})
fig.text(1.01, 0.73, '|', {'color': 'black', 'size': '12', 'weight': 'bold'})
fig.text(1.02, 0.73, 'Notleave', {'font': 'Georgia', 'weight': 'bold', 'Size': '12', 'style': 'italic'})
fig.show()
```

```
C:\Users\ckraft\Anaconda3\lib\site-packages\pywaffle\waffle.py:394: MatplotlibDeprecationWarning: Passing non-integers as three-element position specification is deprecated since 3.3 and will be removed two minor releases later.
```

```
self.ax = self.add_subplot(loc, aspect="equal")
```

```
C:\Users\ckraft\AppData\Local\Temp\ipykernel_4244\1163396892.py:66: MatplotlibDeprecatio
nWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed
two minor releases later
```

```
fig.text(0.955,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '12','weight':'bold','style':'normal', 'color':'#d4210b'})
```

```
C:\Users\ckraft\AppData\Local\Temp\ipykernel_4244\1163396892.py:68: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later
```

```
fig.text(1.02,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '12','style':'normal', 'weight':'bold','color':'#000000'},alpha = 1)
```

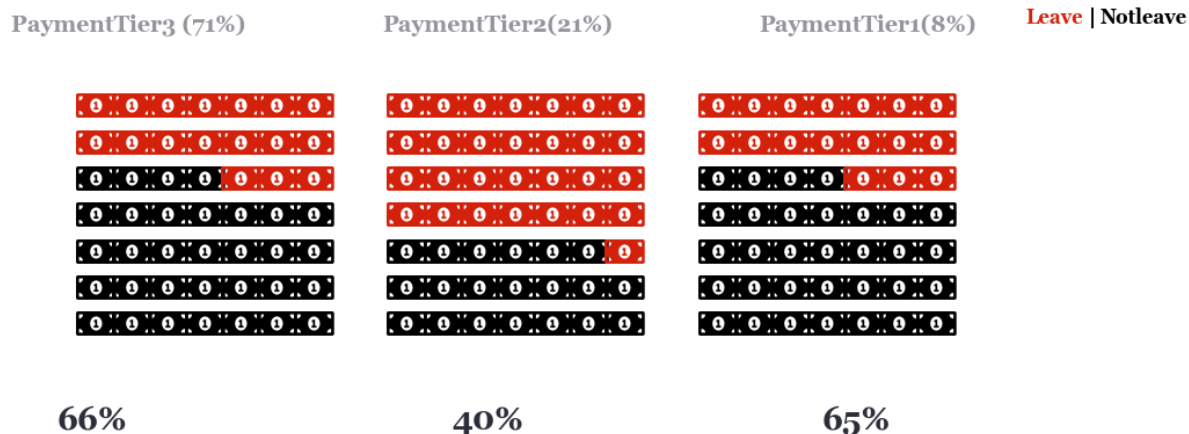
```
C:\Users\ckraft\AppData\Local\Temp\ipykernel_4244\1163396892.py:69: UserWarning: Matplotlib is currently using module://matplotlib_inline.backend_inline, which is a non-GUI backend, so cannot show the figure.
```

```
fig.show()
```

C:\Users\ckraft\Anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning: This figure was using constrained\_layout, but that is incompatible with subplots\_adjust and/or tight layout; disabling constrained layout.

```
fig.canvas.print_figure(bytes_io, **kw)
```

## How is the distribution of Leave Or Not by Payment Tier?



```
In [28]: pd.crosstab(df['ExperienceInCurrentDomain'],df['LeaveOrNot'],margins=True)
```

```
Out[28]:
```



ExperienceInCurrentDomain	0	1	All
0	178	109	287
1	273	160	433
2	390	291	681
3	255	196	451
4	258	167	425
5	310	160	470
6	6	2	8
7	6	3	9
All	1676	1088	2764

In [29]:

```

leave_gen = df[df['LeaveOrNot'] == 1]['ExperienceInCurrentDomain'].value_counts()
notleave_gen = df[df['LeaveOrNot'] == 0]['ExperienceInCurrentDomain'].value_counts()

Exp_2 = df['ExperienceInCurrentDomain'].value_counts().values[0]
Exp_5 = df['ExperienceInCurrentDomain'].value_counts().values[1]
Exp_3 = df['ExperienceInCurrentDomain'].value_counts().values[2]
Exp_1 = df['ExperienceInCurrentDomain'].value_counts().values[3]
Exp_4 = df['ExperienceInCurrentDomain'].value_counts().values[4]
Exp_0 = df['ExperienceInCurrentDomain'].value_counts().values[5]
Exp_7 = df['ExperienceInCurrentDomain'].value_counts().values[6]
Exp_6 = df['ExperienceInCurrentDomain'].value_counts().values[7]

leave_Exp_2 = int(round(leave_gen.values[0] / Exp_2 * 100, 0))
leave_Exp_5 = int(round(leave_gen.values[1] / Exp_5 * 100, 0))
leave_Exp_3 = int(round(leave_gen.values[2] / Exp_3 * 100, 0))
leave_Exp_1 = int(round(leave_gen.values[3] / Exp_1 * 100, 0))
leave_Exp_4 = int(round(leave_gen.values[4] / Exp_4 * 100, 0))
leave_Exp_0 = int(round(leave_gen.values[5] / Exp_0 * 100, 0))
leave_Exp_7 = int(round(leave_gen.values[6] / Exp_7 * 100, 0))
leave_Exp_6 = int(round(leave_gen.values[7] / Exp_6 * 100, 0))

notleave_Exp_2 = int(round(notleave_gen.values[0] / Exp_2 * 100, 0))
notleave_Exp_5 = int(round(notleave_gen.values[1] / Exp_5 * 100, 0))
notleave_Exp_3 = int(round(notleave_gen.values[2] / Exp_3 * 100, 0))
notleave_Exp_1 = int(round(notleave_gen.values[3] / Exp_1 * 100, 0))
notleave_Exp_4 = int(round(notleave_gen.values[4] / Exp_4 * 100, 0))
notleave_Exp_0 = int(round(notleave_gen.values[5] / Exp_0 * 100, 0))
notleave_Exp_7 = int(round(notleave_gen.values[6] / Exp_7 * 100, 0))
notleave_Exp_6 = int(round(notleave_gen.values[7] / Exp_6 * 100, 0))

Exp_2_per = int(round(Exp_2 / (Exp_0 + Exp_1 + Exp_2 + Exp_3 + Exp_4 + Exp_5 + Exp_6 + Exp_7) * 100, 0))
Exp_5_per = int(round(Exp_5 / (Exp_0 + Exp_1 + Exp_2 + Exp_3 + Exp_4 + Exp_5 + Exp_6 + Exp_7) * 100, 0))
Exp_3_per = int(round(Exp_3 / (Exp_0 + Exp_1 + Exp_2 + Exp_3 + Exp_4 + Exp_5 + Exp_6 + Exp_7) * 100, 0))
Exp_1_per = int(round(Exp_1 / (Exp_0 + Exp_1 + Exp_2 + Exp_3 + Exp_4 + Exp_5 + Exp_6 + Exp_7) * 100, 0))
Exp_4_per = int(round(Exp_4 / (Exp_0 + Exp_1 + Exp_2 + Exp_3 + Exp_4 + Exp_5 + Exp_6 + Exp_7) * 100, 0))
Exp_0_per = int(round(Exp_0 / (Exp_0 + Exp_1 + Exp_2 + Exp_3 + Exp_4 + Exp_5 + Exp_6 + Exp_7) * 100, 0))

```

```
Exp_7_per = int(round(Exp_7/(Exp_0+Exp_1+Exp_2+Exp_3+Exp_4+Exp_5+Exp_6+Exp_7) * 100, 0))
Exp_6_per = int(round(Exp_6/(Exp_0+Exp_1+Exp_2+Exp_3+Exp_4+Exp_5+Exp_6+Exp_7) * 100, 0))
```

In [30]:

```
#does experience matter?
fig = plt.figure(FigureClass = Waffle,
                 constrained_layout = True,
                 figsize = (30,12),
                 facecolor = '#ffffff',dpi = 120,

                 plots = {'181':
                        {
                            'rows':7,
                            'columns': 7,
                            'values' : [notleave_Exp_2,leave_Exp_2],
                            'colors' : ['#000000','#d4210b'],
                            'vertical' : True,
                            'interval_ratio_y': 0.95,
                            'interval_ratio_x': 0.95,
                            'icons' : 'briefcase',
                            'icon_legend': False,
                            'icon_size':25,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '182' :
                        {
                            'rows': 7,
                            'columns':7,
                            'values':[notleave_Exp_5,leave_Exp_5],
                            'colors' : ['#000000','#d4210b'],
                            'vertical': True,
                            'interval_ratio_y': 0.95,
                            'interval_ratio_x': 0.95,
                            'icons' : 'briefcase',
                            'icon_legend' :False,
                            'icon_size':25,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '183' :
                        {
                            'rows': 7,
                            'columns':7,
                            'values':[notleave_Exp_3,leave_Exp_3],
                            'colors' : ['#000000','#d4210b'],
                            'vertical': True,
                            'interval_ratio_y': 0.95,
                            'interval_ratio_x': 0.95,
                            'icons' : 'briefcase',
                            'icon_legend' :False,
                            'icon_size':25,
                            'plot_anchor':'C',
                            'alpha':0.1
                        },

                        '184' :
```

```

{
  'rows': 7,
  'columns': 7,
  'values': [notleave_Exp_1, leave_Exp_1],
  'colors' : ['#000000', '#d4210b'],
  'vertical': True,
  'interval_ratio_y': 0.95,
  'interval_ratio_x': 0.95,
  'icons' : 'briefcase',
  'icon_legend' : False,
  'icon_size': 25,
  'plot_anchor': 'C',
  'alpha': 0.1
},
'185' :
{
  'rows': 7,
  'columns': 7,
  'values': [notleave_Exp_4, leave_Exp_4],
  'colors' : ['#000000', '#d4210b'],
  'vertical': True,
  'interval_ratio_y': 0.95,
  'interval_ratio_x': 0.95,
  'icons' : 'briefcase',
  'icon_legend' : False,
  'icon_size': 25,
  'plot_anchor': 'C',
  'alpha': 0.1
},
'186' :
{
  'rows': 7,
  'columns': 7,
  'values': [notleave_Exp_0, leave_Exp_0],
  'colors' : ['#000000', '#d4210b'],
  'vertical': True,
  'interval_ratio_y': 0.95,
  'interval_ratio_x': 0.95,
  'icons' : 'briefcase',
  'icon_legend' : False,
  'icon_size': 25,
  'plot_anchor': 'C',
  'alpha': 0.1
},
'187' :
{
  'rows': 7,
  'columns': 7,
  'values': [notleave_Exp_7, leave_Exp_7],
  'colors' : ['#000000', '#d4210b'],
  'vertical': True,
  'interval_ratio_y': 0.95,
  'interval_ratio_x': 0.95,
  'icons' : 'briefcase',
  'icon_legend' : False,
  'icon_size': 25,
  'plot_anchor': 'C',
  'alpha': 0.1
}

```

```

    },
    '188' :
    {
        'rows': 7,
        'columns':7,
        'values':[notleave_Exp_6,leave_Exp_6],
        'colors' : ['#000000','#d4210b'],
        'vertical': True,
        'interval_ratio_y': 0.95,
        'interval_ratio_x': 0.95,
        'icons' : 'briefcase',
        'icon_legend' :False,
        'icon_size':25,
        'plot_anchor': 'C',
        'alpha':0.1

    },

}

)

fig.text(0.001, 0.8, 'How is the distribution of Leave Or Not by Experience In Current
fig.text(0.05, 0.32, '{}%'.format(notleave_Exp_2), {'font':'Georgia', 'size':22,'weight
fig.text(0.18, 0.32, '{}%'.format(notleave_Exp_5), {'font':'Georgia', 'size':22,'weight
fig.text(0.31, 0.32, '{}%'.format(notleave_Exp_3), {'font':'Georgia', 'size':22,'weight
fig.text(0.42, 0.32, '{}%'.format(notleave_Exp_1), {'font':'Georgia', 'size':22,'weight
fig.text(0.55, 0.32, '{}%'.format(notleave_Exp_4), {'font':'Georgia', 'size':22,'weight
fig.text(0.67, 0.32, '{}%'.format(notleave_Exp_0), {'font':'Georgia', 'size':22,'weight
fig.text(0.795, 0.32, '{}%'.format(notleave_Exp_7), {'font':'Georgia', 'size':22,'weigh
fig.text(0.92, 0.32, '{}%'.format(notleave_Exp_6), {'font':'Georgia', 'size':22,'weight

fig.text(0.04, 0.67, 'Exp_2({}% )'.format(Exp_2_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.16, 0.67, 'Exp_5({}% )'.format(Exp_5_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.28, 0.67, 'Exp_3({}% )'.format(Exp_3_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.40, 0.67, 'Exp_1({}% )'.format(Exp_1_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.53, 0.67, 'Exp_4({}% )'.format(Exp_4_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.66, 0.67, 'Exp_0({}% )'.format(Exp_0_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.78, 0.67, 'Exp_7({}% )'.format(Exp_7_per), {'font':'Georgia', 'size':24,'weig
fig.text(0.905, 0.67, 'Exp_6({}% )'.format(Exp_6_per), {'font':'Georgia', 'size':24,'wei

fig.text(0.80,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '26','weight':
fig.text(0.84,0.73, '|', {'color':'black' , 'size':'26', 'weight': 'bold'})
fig.text(0.845,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '26','styl
fig.show()

```

C:\Users\ckraft\Anaconda3\lib\site-packages\pywaffle\waffle.py:394: MatplotlibDeprecationWarning: Passing non-integers as three-element position specification is deprecated since 3.3 and will be removed two minor releases later.

self.ax = self.add\_subplot(loc, aspect="equal")

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\2568247967.py:162: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later

fig.text(0.80,0.73, 'Leave', {'font': 'Georgia','weight':'bold','Size': '26','weight':'bold','style':'normal', 'color':'#d4210b'})

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\2568247967.py:164: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed

d two minor releases later

```
fig.text(0.845,0.73, 'Notleave', {'font': 'Georgia','weight':'bold', 'Size': '26','style':'normal', 'weight':'bold','color':'#000000'},alpha = 1)
```

C:\Users\ckraft\AppData\Local\Temp\ipykernel\_4244\2568247967.py:165: UserWarning: Matplotlib is currently using module://matplotlib\_inline.backend\_inline, which is a non-GUI backend, so cannot show the figure.

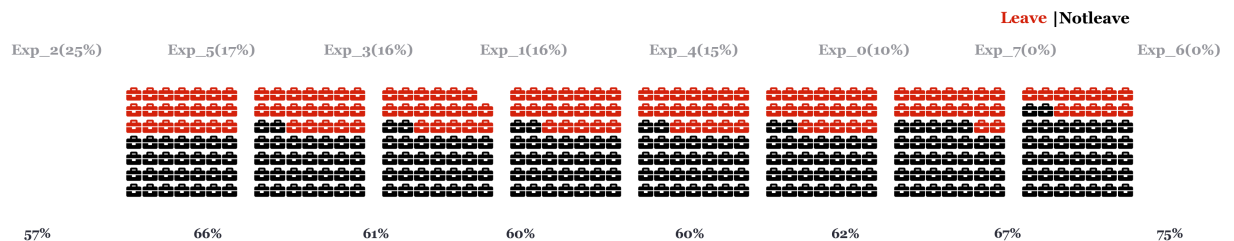
```
fig.show()
```

C:\Users\ckraft\Anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning: This figure was using constrained\_layout, but that is incompatible with subplots\_adjust and/or tight\_layout; disabling constrained\_layout.

```
fig.canvas.print_figure(bytes_io, **kw)
```

Backing off send\_request(...) for 4.5s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verification failed: self signed certificate in certificate chain (\_ssl.c:1125))))))

### How is the distribution of Leave Or Not by Experience In Current Domain ?

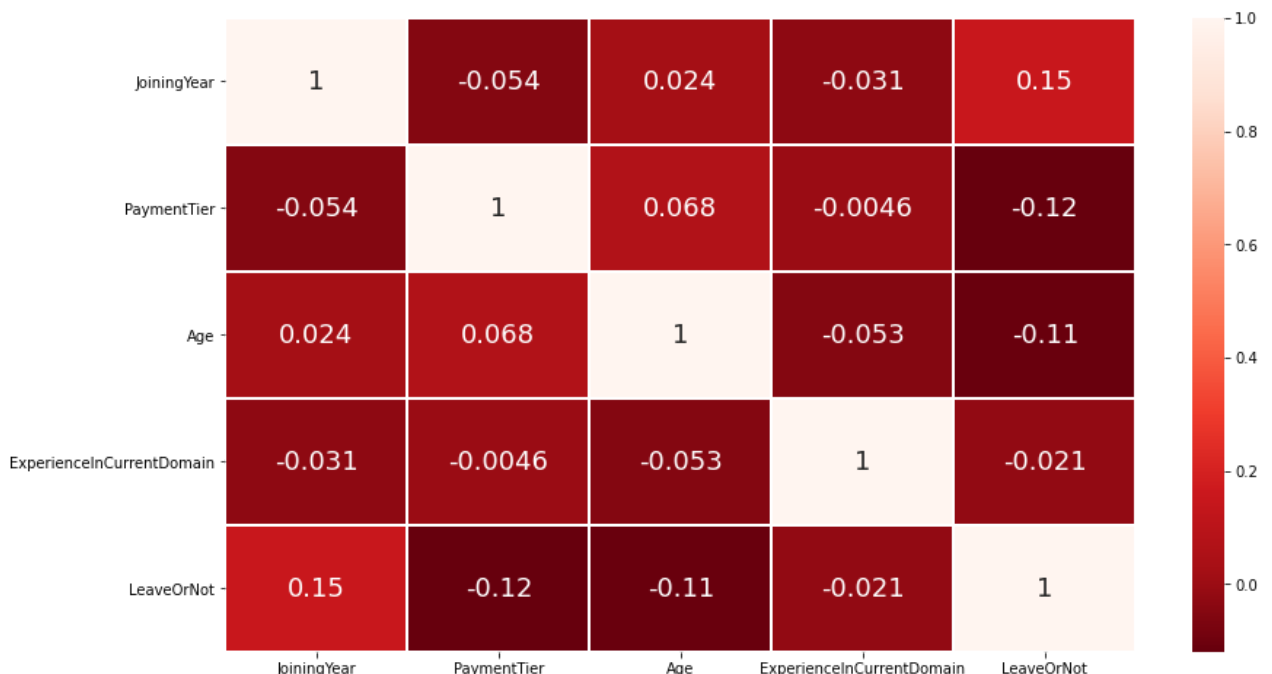


In [31]:

```
#correlations
plt.figure(figsize=(14,8))
sns.heatmap(df.corr(), annot=True,cmap = "Reds_r",linewidths=2,linecolor = "white",cbar
```

Out[31]:

<AxesSubplot:>



In [32]:

```
Num_features = df.iloc[:, [1,3,4,7,8]]
print(type(Num_features))
print('\n')
print(Num_features.head())
```

<class 'pandas.core.frame.DataFrame'>

	JoiningYear	Age	ExperienceInCurrentDomain	LeaveOrNot
0	2017	34	0	0
1	2013	28	3	1
2	2014	38	2	0
3	2016	27	5	1
4	2017	24	2	1

In [33]:

```
import matplotlib
background_color = "#f6f6f6"

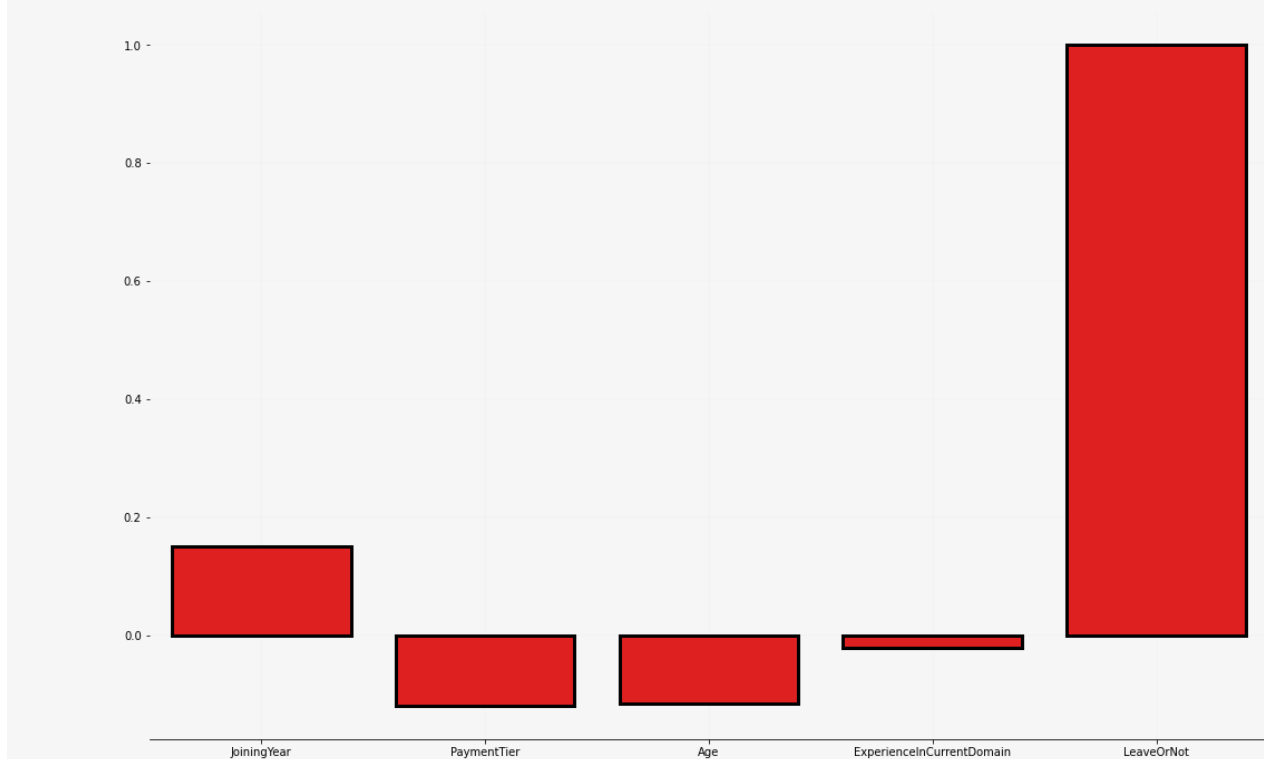
fig = plt.figure(figsize=(18,12), facecolor=background_color)
gs = fig.add_gridspec(1, 1)
ax0 = fig.add_subplot(gs[0, 0])
colors = ["#2f5586", "#f6f5f5", "#2f5586"]
colormap = matplotlib.colors.LinearSegmentedColormap.from_list("", colors)

ax0.set_facecolor(background_color)
ax0.text(-1.1, 1.25, 'Correlation of Numerical Features with Target', fontsize=20, font

chart_df = pd.DataFrame(df.corrwith(df['LeaveOrNot']))
chart_df.columns = ['corr']
sns.barplot(x=chart_df.index, y=chart_df['corr'], ax=ax0, color='red', zorder=3, edgeco
ax0.grid(which='major', axis='x', zorder=0, color='#EEEEEE', linewidth=0.4)
ax0.grid(which='major', axis='y', zorder=0, color='#EEEEEE', linewidth=0.4)
ax0.set_ylabel('')

for s in ["top", "right", 'left']:
    ax0.spines[s].set_visible(False)

plt.show()
```

**Correlation of Numerical Features with Target**

In [34]:

```
#senority is highly correlated with retention,
#the earlier you're in the game the longer you're likely to not leave

#is that due to comfortability or early investments into the social fabric of the employ
#for further study
```

```
Backing off send_request(...) for 13.0s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1125)'))))
Backing off send_request(...) for 38.0s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1125)'))))
Backing off send_request(...) for 26.1s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1125)'))))
```