IBM HR Analysis

Group 3 Python Final Project

Justin Ghazi
Nicholas Laudadio
Nien-Thing Chiang
Duke (Xinyu) Li

In [75]:
```
df = pd.read_csv('HR.csv')
df.head()
```

Out[75]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisf
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	

In [76]:
```
df.shape
```

Out[76]: (1470, 35)

Explore the missing value

In [78]:
```
df.isnull().sum().head(20)
```

Out[78]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
dtype: int64
```

In [77]:
```
df.isnull().sum().sum()
```

Out[77]:
```
0
```

Data Cleaning

- first replace Yes and No in Attrition with 1 and 0.

```
In [33]: df1.replace(to_replace='Yes', value=1.0, inplace=True)
         df1.replace(to_replace='Y', value=1.0, inplace=True)
         df1.replace(to_replace='No', value= 0.0 , inplace=True)
         df1.replace(to_replace='N', value= 0.0, inplace=True)
```

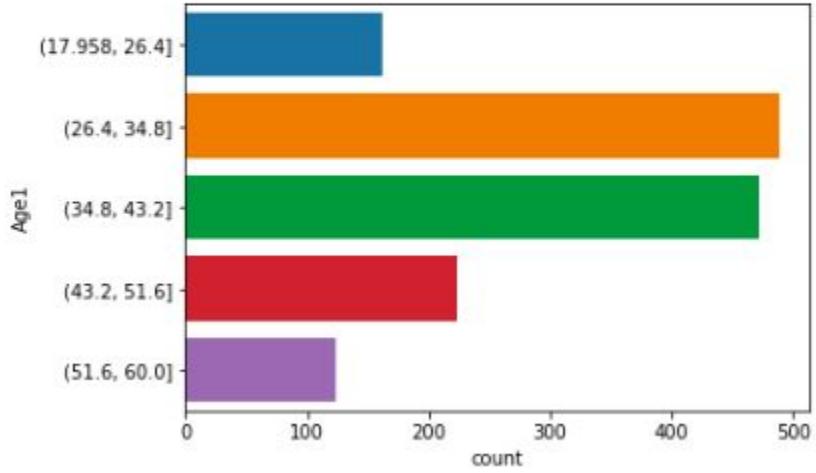
- For Gender column, transfer Female into 0, Male into 1

```
In [35]: df1.replace(to_replace='Male', value= 1.0, inplace=True)
         df1.replace(to_replace='Female', value= 0.0, inplace=True)
```

```
In [84]: df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
Age                         1470 non-null int64
Attrition                   1470 non-null float64
BusinessTravel              1470 non-null object
DailyRate                   1470 non-null int64
Department                  1470 non-null object
DistanceFromHome            1470 non-null int64
Education                   1470 non-null int64
EducationField              1470 non-null object
EmployeeCount               1470 non-null int64
EmployeeNumber              1470 non-null int64
EnvironmentSatisfaction     1470 non-null int64
Gender                      1470 non-null float64
HourlyRate                  1470 non-null int64
JobInvolvement              1470 non-null int64
JobLevel                    1470 non-null int64
JobRole                     1470 non-null object
JobSatisfaction             1470 non-null int64
MaritalStatus               1470 non-null object
MonthlyIncome               1470 non-null int64
MonthlyRate                 1470 non-null int64
```

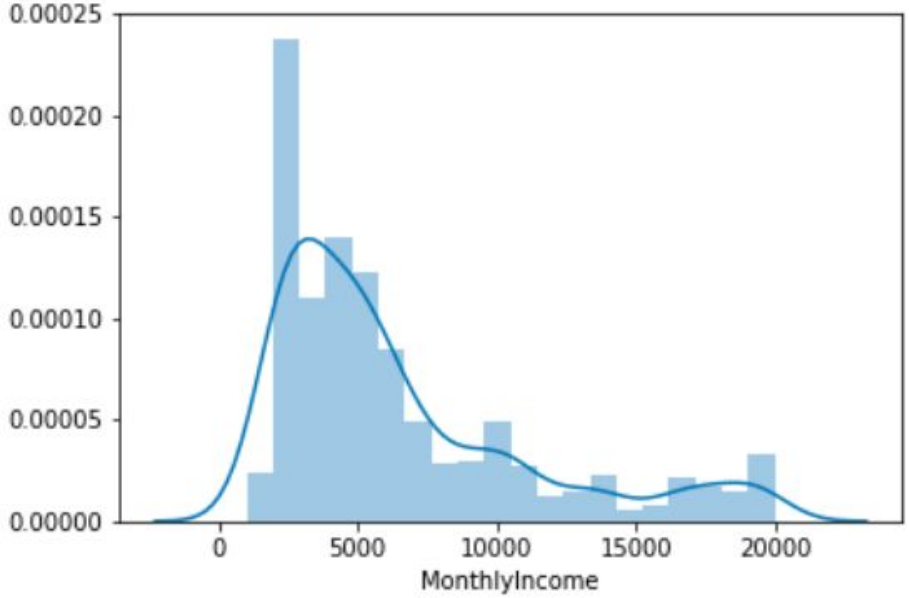
Data Exploration

```
In [39]: df2['Age1'] = pd.cut(df2.Age,5)

In [40]: sns.countplot(y='Age1', data=df2)

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1260a90b8>
```



```
In [28]: sns.distplot(df1.MonthlyIncome)
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x1184fbe80>
```



```
In [69]: df2.groupby(by=['Department'])['Department'].size()

Out[69]: Department
         Human Resources              63
         Research & Development      961
         Sales                       446
         Name: Department, dtype: int64
```

```
df2['satisfaction'] = df2.RelationshipSatisfaction + df2.EnvironmentSatisfaction + df2.JobSatisfaction
```

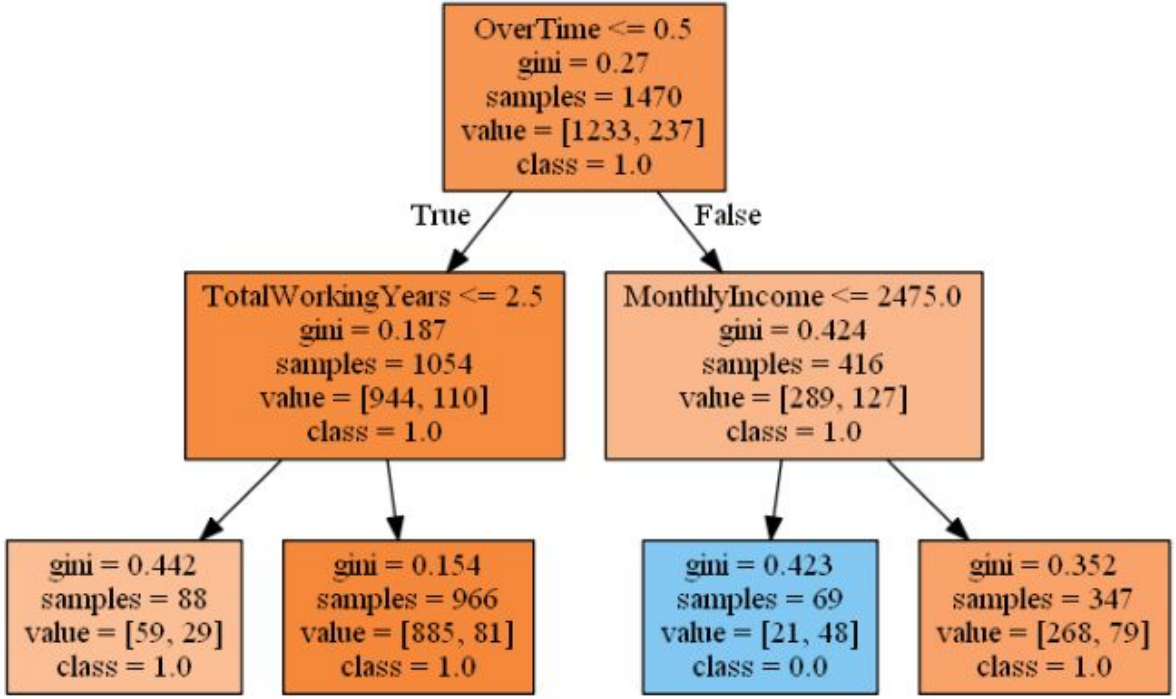
```
df2.groupby(by=['Department'])[['OverTime','WorkLifeBalance','satisfaction','MonthlyIncome',\
                                'PercentSalaryHike','StockOptionLevel','Attrition']].mean()
```

Department	OverTime	WorkLifeBalance	satisfaction	MonthlyIncome	PercentSalaryHike	StockOptionLevel	Attrition
Human Resources	0.269841	2.920635	8.174603	6654.507937	14.761905	0.777778	0.190476
Research & Development	0.281998	2.725286	8.178980	6281.252862	15.291363	0.804370	0.138398
Sales	0.286996	2.816143	8.125561	6959.172646	15.096413	0.773543	0.206278

One-hot encoding

```
l = ['BusinessTravel','Department','EducationField','JobRole','MaritalStatus']

def one_hot(df,l):
    for n in l:
        df = pd.get_dummies(df,columns = [n])
    return df

df2 = one_hot(df1,l)
```

Out[53]:


OverTime <= 0.5
gini = 0.27
samples = 1470
value = [1233, 237]
class = 1.0
True
False
TotalWorkingYears <= 2.5
gini = 0.187
samples = 1054
value = [944, 110]
class = 1.0
MonthlyIncome <= 2475.0
gini = 0.424
samples = 416
value = [289, 127]
class = 1.0
gini = 0.442
samples = 88
value = [59, 29]
class = 1.0
gini = 0.154
samples = 966
value = [885, 81]
class = 1.0
gini = 0.423
samples = 69
value = [21, 48]
class = 0.0
gini = 0.352
samples = 347
value = [268, 79]
class = 1.0

In [104]:
```
dt2.feature_importances_
```

Out[104]:
```
array([0.        , 0.08079671, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.30299613, 0.        , 0.        , 0.        , 0.28944502,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.11702951, 0.        , 0.0351275 , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.06576693, 0.        , 0.        , 0.        , 0.        ,
       0.1088382 ])
```

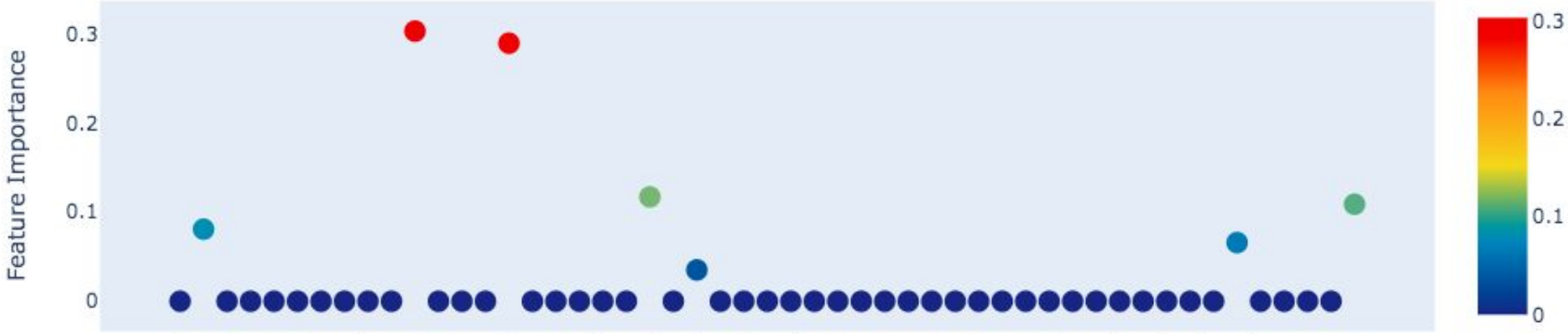
In [105]:
```
d = {X.columns[i] : dt2.feature_importances_[i] for i in range(0,len(X.columns))}
d
```

...

In [109]:
```
s = pd.Series(d)
s.nlargest(5)
```

Out[109]:
```
MonthlyIncome           0.302996
OverTime                0.289445
TotalWorkingYears       0.117030
MaritalStatus_Single    0.108838
DailyRate               0.080797
dtype: float64
```

In [97]:
```
import plotly.graph_objs as go
import plotly.offline as py
```



Point 1: How to attract more young talents?

K-means: Cluster all the employees to 3 groups

K-Means Model

```
In [27]: from sklearn.cluster import KMeans

In [29]: clu = KMeans(n_clusters=3, random_state=0)

In [30]: clu

Out[30]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                random_state=0, tol=0.0001, verbose=0)
```

Normalization

```
In [31]: from sklearn.preprocessing import StandardScaler
         ss = StandardScaler()
         df2nol =ss.fit_transform(df2)

In [33]: clu.fit(df2nol)

Out[33]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                random_state=0, tol=0.0001, verbose=0)

In [34]: clu.labels_

Out[34]: array([2, 1, 1, ..., 1, 2, 1], dtype=int32)
```

1. Build the Model
2. Normalization

K-means: Cluster all the employees to 3 groups

cluster	Age	Attrition	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement
0	46.039841	0.055777	809.717131	8.928287	3.087649	2.713147	0.513944	66.123506	2.717131
1	34.996341	0.164634	801.456098	9.178049	2.859756	2.745122	0.629268	66.159756	2.750000
2	35.150376	0.220551	800.052632	9.388471	2.912281	2.679198	0.593985	65.192982	2.696742

Attrition rate is different, but other part are very similar.

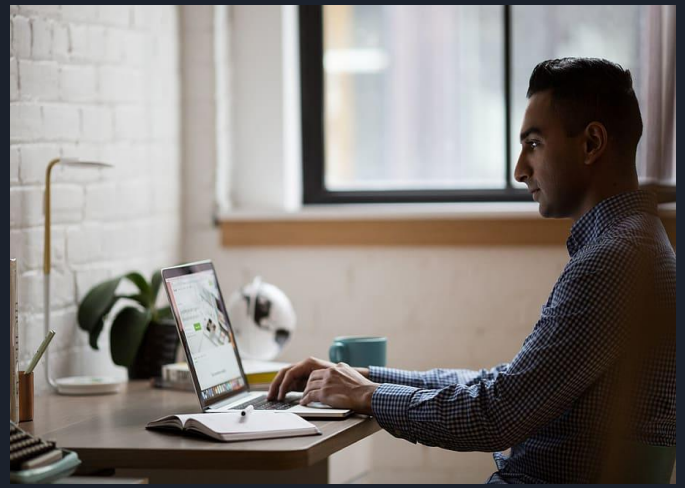
Group Demography:

MonthlyIncome	TotalWorkingYears	Department_Human Resources	Department_Research & Development	Department_Sales
15035.039841	23.561753	0.051793	0.760956	0.187251
4178.826829	8.476829	0.060976	0.939024	0.000000
5911.969925	9.313283	0.000000	0.000000	1.000000

Group 0 - Senior Manager
Group 1 - Young Talent (R&D)
Group 2 - Young Talent (Sales)

Group Demography:

0



1

2

If you are the boss, which group should you look at?

Of course Young Talent!

1. Subset Young Talent

```
dfyoungtalent = df1[df1.cluster != 0]
dfyoungtalent.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	JobInv
0	41	1.0	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	0.0	94	
1	49	0.0	Travel_Frequently	279	Research & Devel...	8	1	Life Sciences	3	1.0	61	
2	37	1.0	Travel_Rarely	1373	Research & Devel...	2	2	Other	4	1.0	92	
3	33	0.0	Travel_Frequently	1392	Research & Devel...	3	4	Life Sciences	4	0.0	56	
4	27	0.0	Travel_Rarely	591	Research & Devel...	2	1	Medical	1	1.0	40	

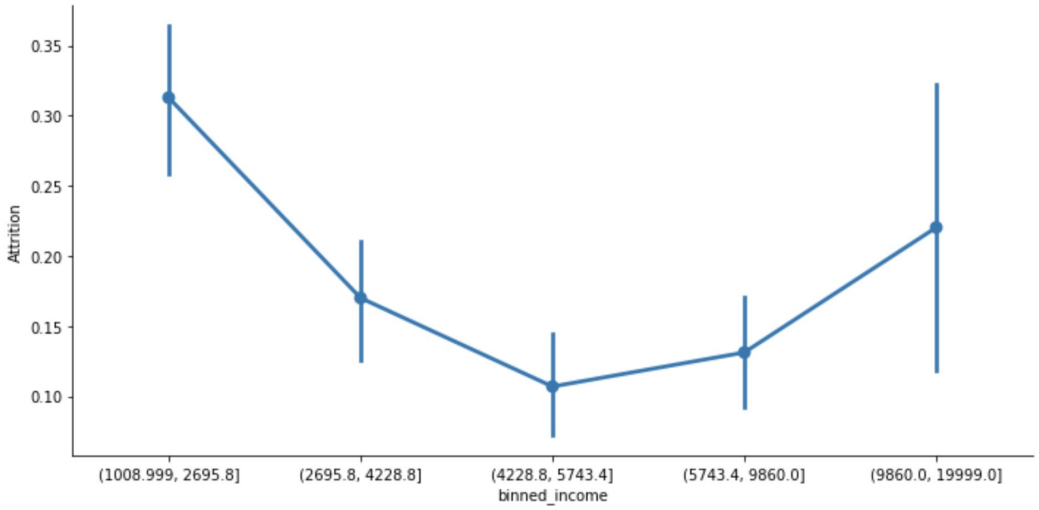
2. Pcut Monthly Income

```
dfyoungtalent['binned_income'] = pd.qcut(dfyoungtalent.MonthlyIncome,5)
```

U shape:

```
sns.catplot(x='binned_income',y='Attrition',data=dfjunior, kind='point',aspect = 2)
```

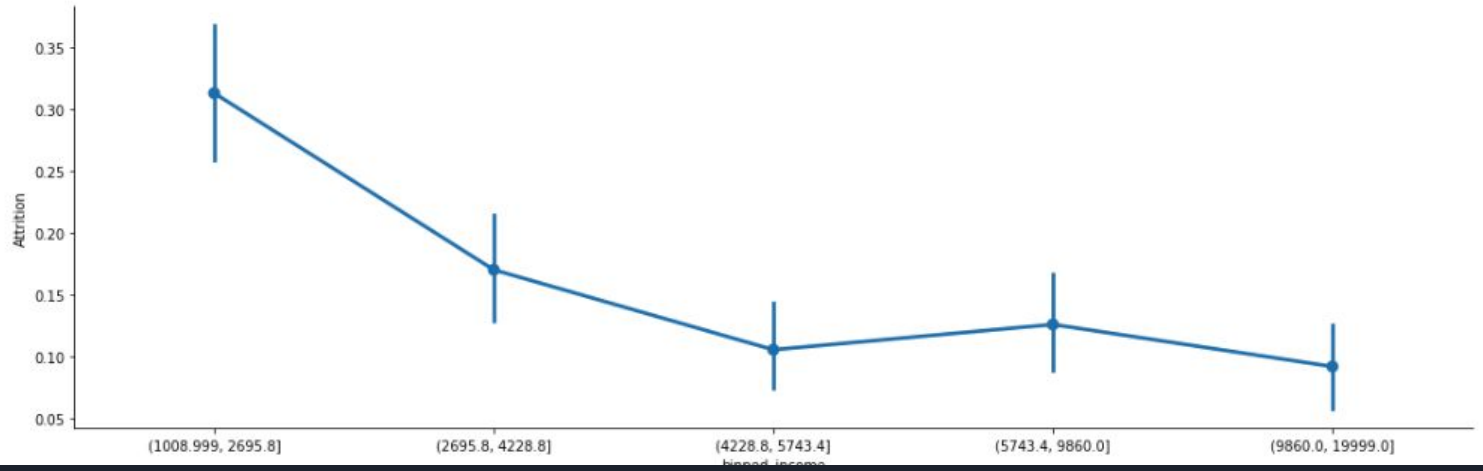
```
<seaborn.axisgrid.FacetGrid at 0x1c3ba64c90>
```



If we plot for everyone:

```
sns.catplot(x='binned_income',y='Attrition',data=df1, kind='point',aspect = 3)
```

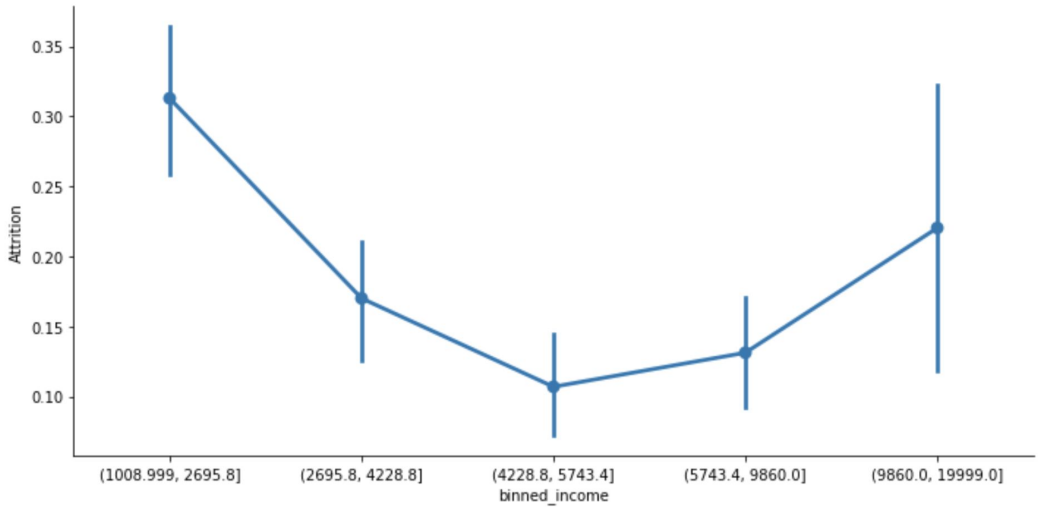
```
<seaborn.axisgrid.FacetGrid at 0x1a22efddd0>
```



U shape:

```
sns.catplot(x='binned_income',y='Attrition',data=dfjunior, kind='point',aspect = 2)
```

```
<seaborn.axisgrid.FacetGrid at 0x1c3ba64c90>
```



Other companies have higher salary level!!!!

Suggestion for boss

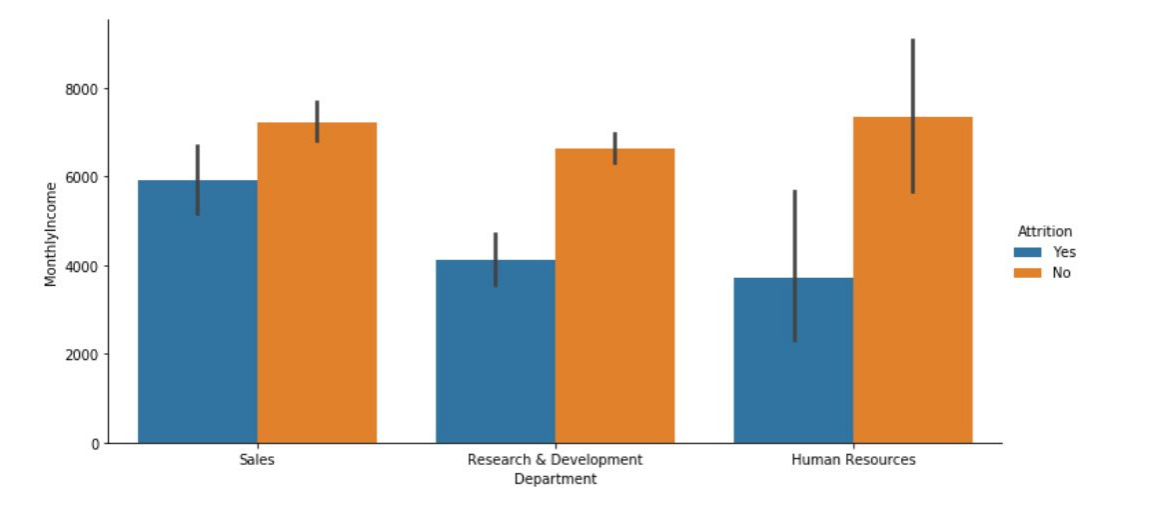
We should increase monthly income for the young talent whose monthly income between 5000 - 20000

Point 2: Money is The Root of All Decisions...Right?

```
(df3.groupby('Department').agg({'Attrition':'sum'})\
        .rename(columns={'Attrition':'Attrition_Proportion'}))\
        /len(df3[df3.Attrition == 1])
```

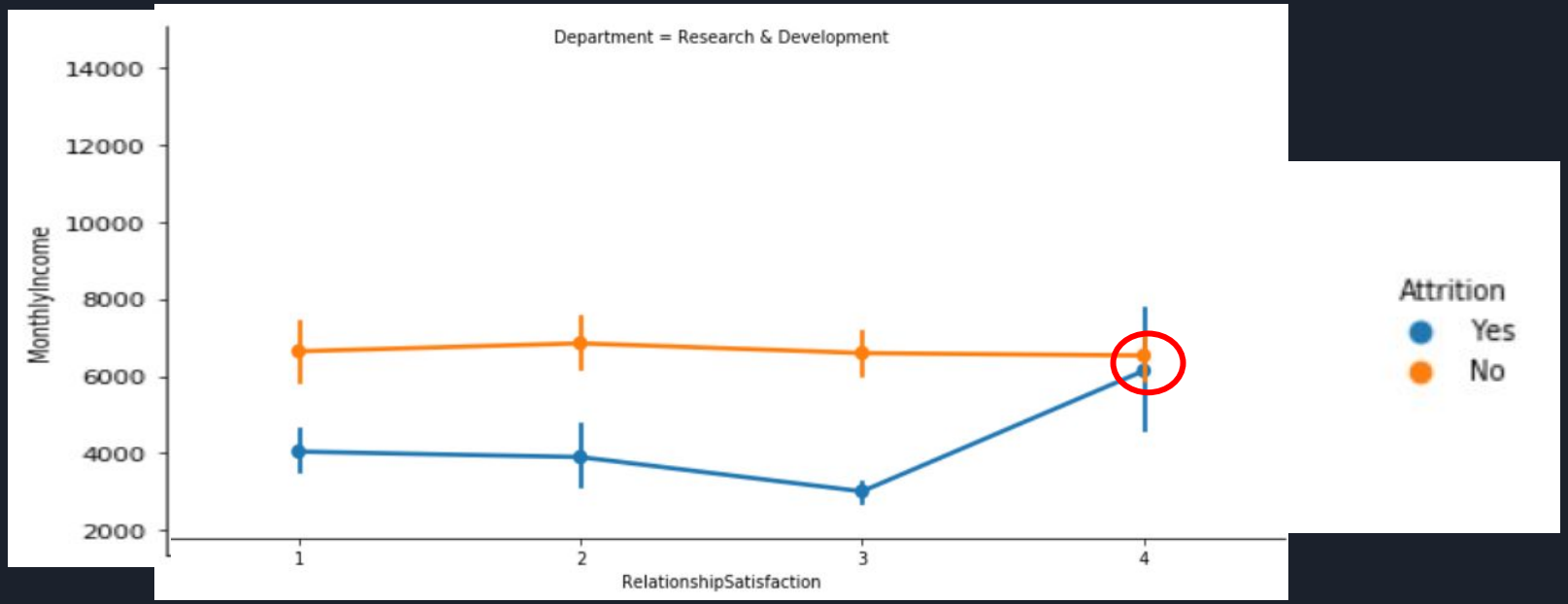
Department	Attrition_Proportion
Human Resources	0.050633
Research & Development	0.561181
Sales	0.388186

```
sns.catplot(x='Department', y='MonthlyIncome',hue = 'Attrition', data=df, kind ='bar', aspect = 2)
```



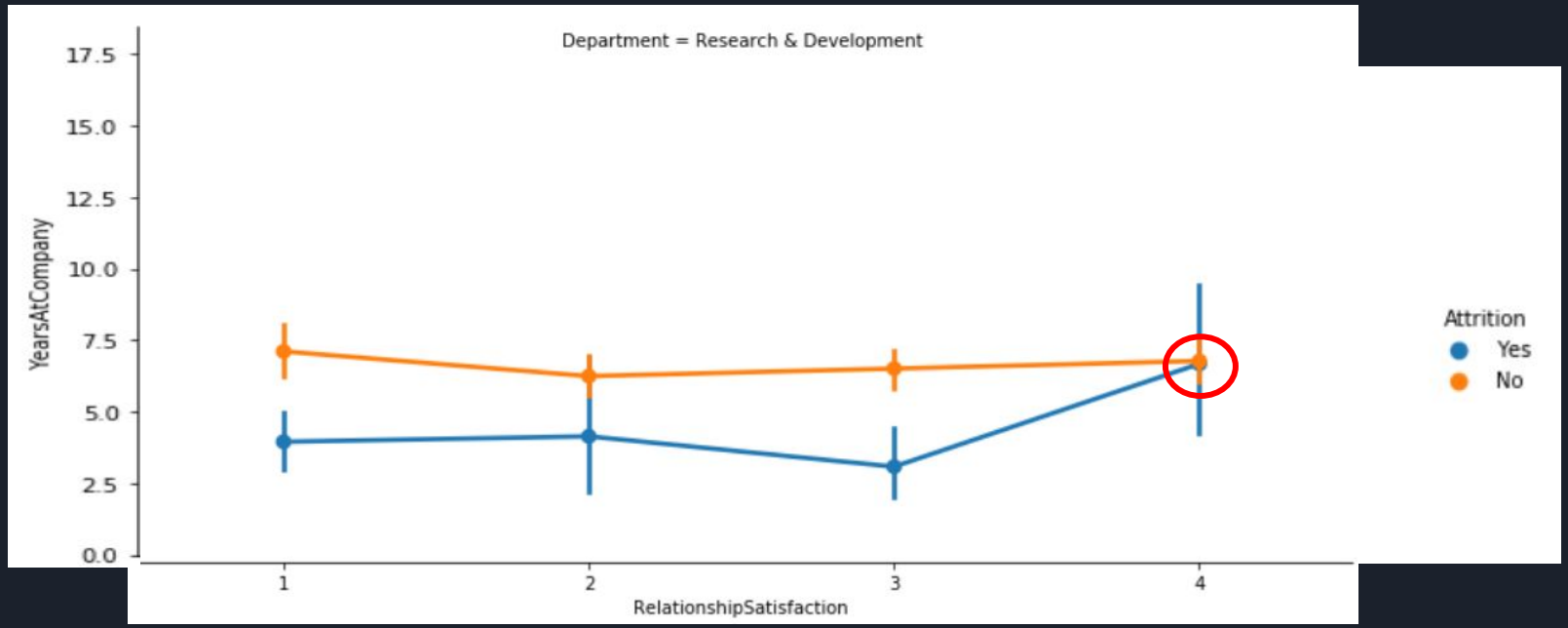
R&D attrition is high in comparison with total attrition

```
sns.catplot(x='RelationshipSatisfaction',y='MonthlyIncome', hue='Attrition', \
            col='Department', kind='point', data=df, aspect=2)
```



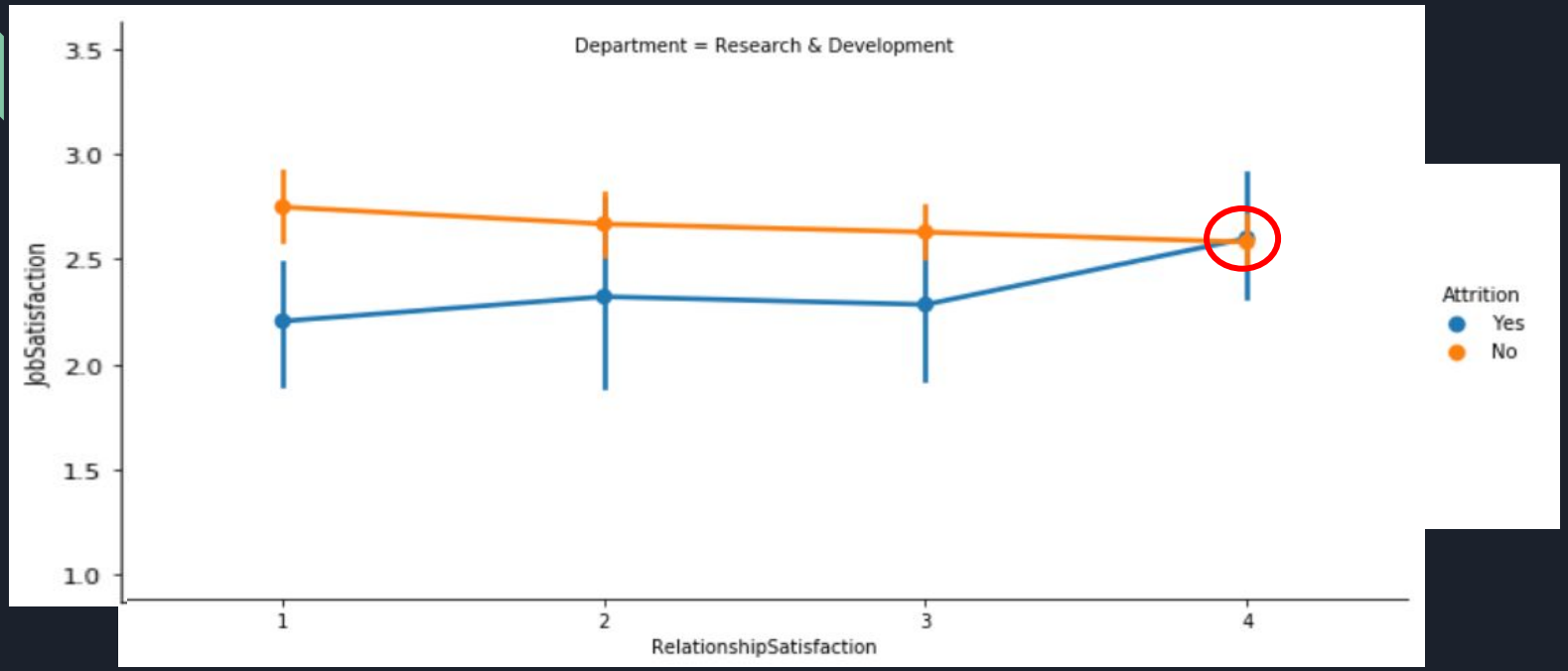
Relationship Satisfaction is high, monthly income is similar. But still attrition?

```
sns.catplot(x='RelationshipSatisfaction',y='YearsAtCompany', hue='Attrition', \
            col='Department', kind='point', data=df, aspect=2)
```



Working for moderate amount of time, yet attrition still?

```
sns.catplot(x='RelationshipSatisfaction',y='JobSatisfaction', hue='Attrition', \
            col='Department', kind='point', data=df, aspect=2)
```



Job satisfaction is fairly low....Interesting

Suggestions for Decision Makers

- ***Scale the R&D Department:***
  - ***More strategic projects***
  - ***Increase budget***
    - ***Create more higher-tier roles***
    - ***Create more promotion opportunities***
    - ***Create more workshops***

- ***Important: Ensure changes do not compromise high relationship satisfaction***

Point 3: Why are Research Directors Quitting?

Who is quitting when making higher Monthly Income?

```
df2.groupby(['JobRole','Attrition'])['MonthlyIncome'].mean()
```

JobRole	Attrition	
Healthcare Representative	0	7453.557377
	1	8548.222222
Human Resources	0	4391.750000
	1	3715.750000
Laboratory Technician	0	3337.223350
	1	2919.258065
Manager	0	17201.484536
	1	16797.400000
Manufacturing Director	0	7289.925926
	1	7365.500000
Research Director	0	15947.346154
	1	19395.500000
Research Scientist	0	3328.122449
	1	2780.468085
Sales Executive	0	6804.617100
	1	7489.000000
Sales Representative	0	2798.440000
	1	2364.727273

Name: MonthlyIncome, dtype: float64

Employees Who Quit made, on average, ___% more in Monthly Income than those who didn't quit

Healthcare Representative - 15% more

Manufacturing Director - 1% more

Research Director - 22% more

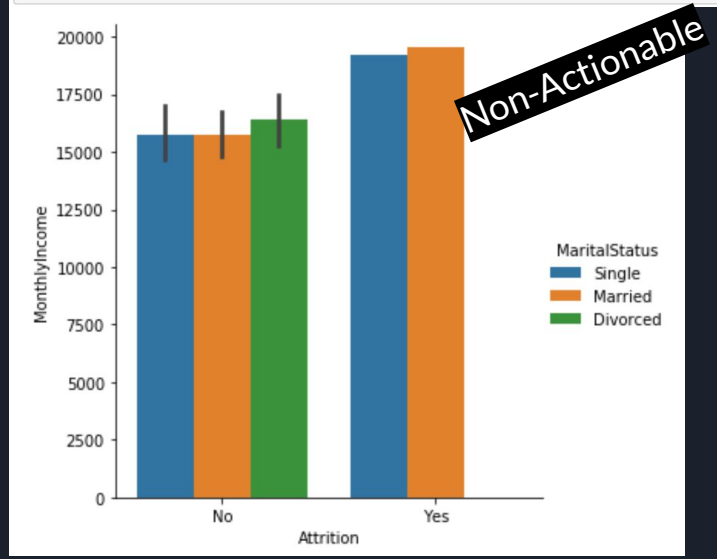
Sales Executive - 10% more

```
df4 = df[df2.JobRole == 'Research Director']
```

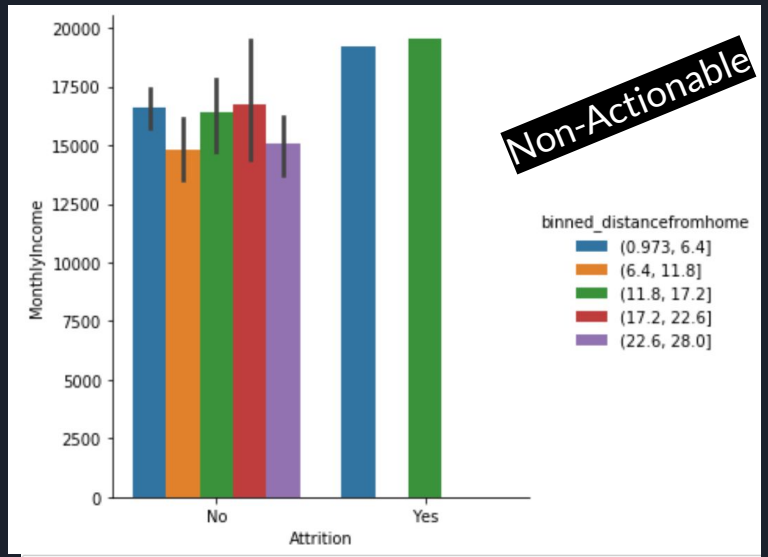
Possible reasons attrition is occuring

Marital Status

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='MaritalStatus', data=df4, kind='bar')
```



Distance From Home



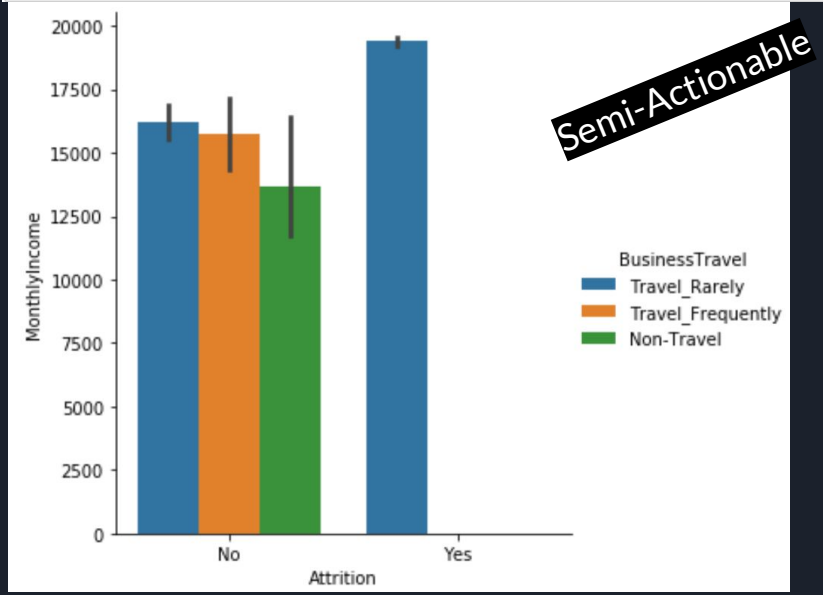

```
df4['binned_distancefromhome'] = pd.cut(df4.DistanceFromHome,5)
```

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='binned_distancefromhome', data=df4, kind='bar')
```

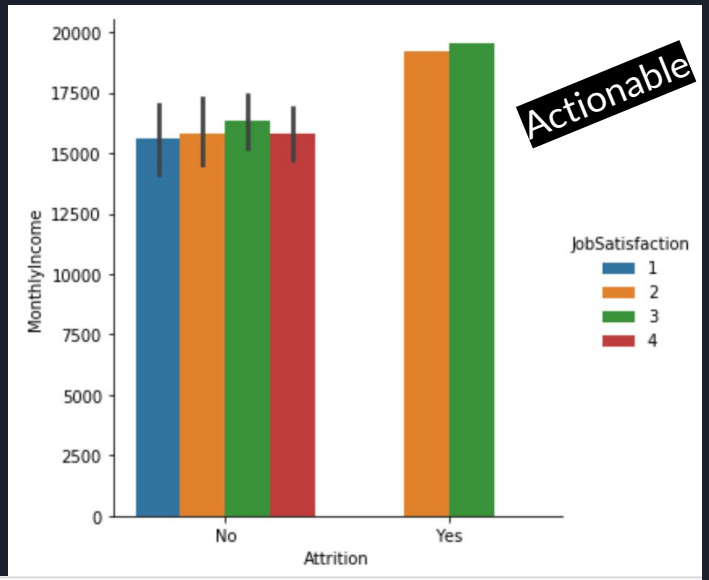
Possible reasons attrition is occuring continued

Business Travel

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='BusinessTravel', data=df4, kind='bar')
```



Job Satisfaction

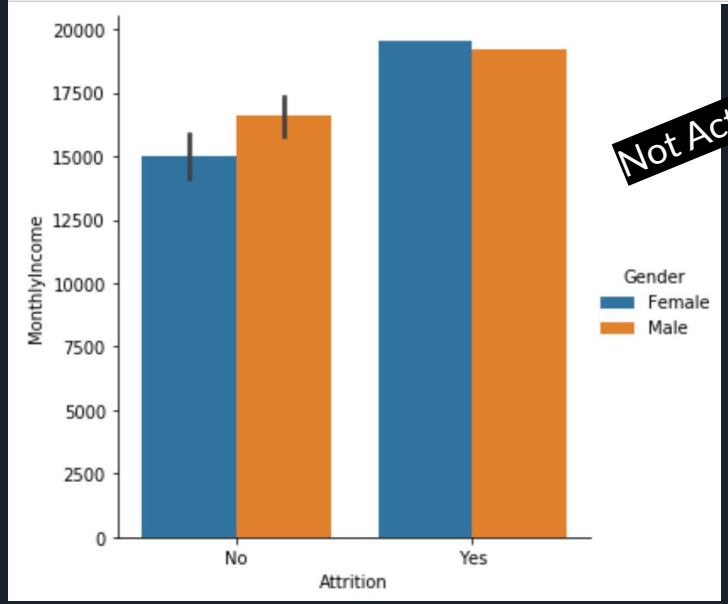



```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='JobSatisfaction', data=df4, kind='bar')
```

Possible reasons attrition is occuring continued

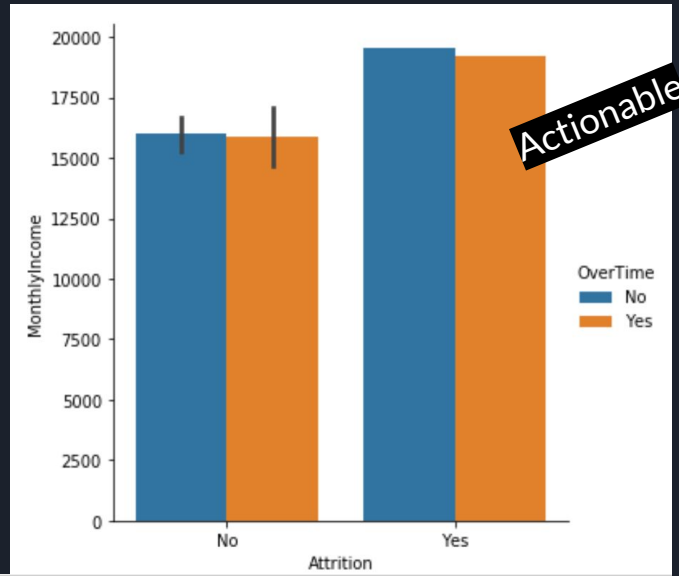
Gender

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='Gender', data=df4, kind='bar')
```



Not Actionable

Overtime



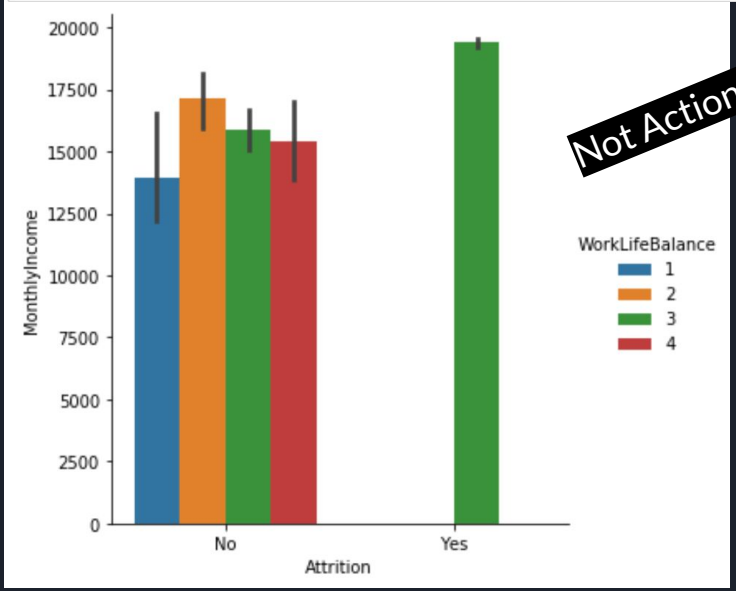
Actionable

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='OverTime', data=df4, kind='bar')
```

Possible reasons attrition is occuring continued

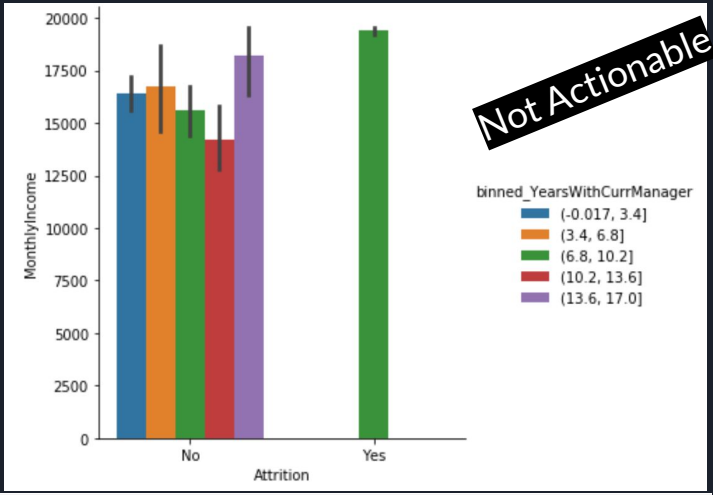
Work Life Balance

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='WorkLifeBalance', data=df4, kind='bar')
```



Not Actionable

Years with Current Manager



Not Actionable

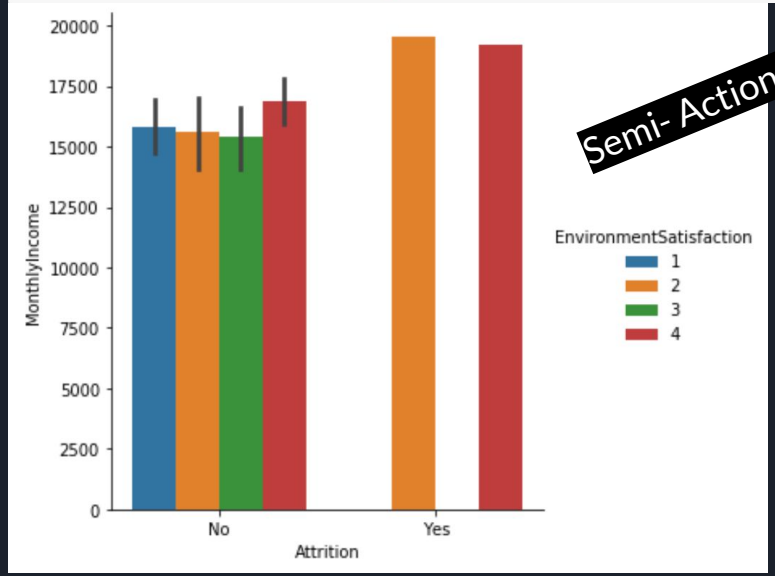
```
df4['binned_YearsWithCurrManager'] = pd.cut(df4.YearsWithCurrManager,5)
```

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='binned_YearsWithCurrManager', data=df4, kind='bar')
```

Possible reasons attrition is occuring continued

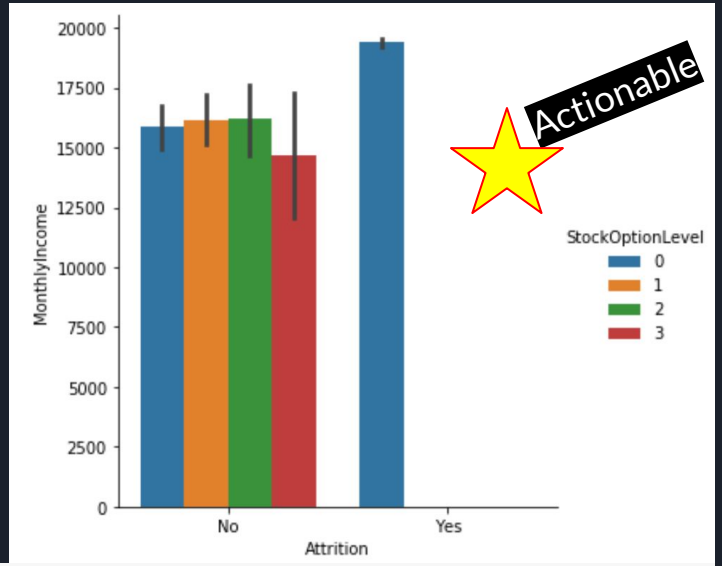
Environment Satisfaction

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='EnvironmentSatisfaction', data=df4, kind='bar')
```



Semi- Actionable

Stock Option Level



Actionable

```
sns.catplot(x='Attrition', y='MonthlyIncome', hue='StockOptionLevel', data=df4, kind='bar')
```

Executive Recommendation

Recommendations to help reduce Attrition from Research Director Employees

1) Gather additional information regarding “Job Satisfaction” & “Environment Satisfaction” from Research Director employees to understand what could be improved to help raise these scores, to help reduce attrition
   a) Use insights gathered to identify actionable steps to reduce attrition

2) Look into making revisions to the business travel policies
   a) Offer higher salary or additional perks for business travelers
   b) Look for opportunities to eliminate all business travel if possible

3) Re-Evaluate overtime requirements and compensation
   a) Hire additional employees to help reduce total overtime needed
   b) Adjust compensation/perks for Research Directors who work overtime

4) Provide Stock Options to all Research Director Employees