Employee_Productivity_Satisfaction

April 1, 2024

1 Employee Productivity and Satisfaction HR Data

1.0.1 Import Python Libraries

[1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import datetime
```

1.0.2 Read and extract the file

[2]:

```
# Dataset from https://www.kaggle.com/datasets/adityaab1407/
 ↪employee-productivity-and-satisfaction-hr-data?
 ↪resource=download&select=hr_dashboard_data.csv
hr_data = pd.read_csv('hr_dashboard_data.csv')
hr_data.head(10)
```

[2]:

```
                Name  Age  Gender  Projects Completed  Productivity (%)  \
0    Douglas Lindsey   25    Male                  11                57
1  Anthony Roberson   59  Female                  19                55
2      Thomas Miller   30    Male                   8                87
3       Joshua Lewis   26  Female                   1                53
4   Stephanie Bailey   43    Male                  14                 3
5      Jonathan King   24    Male                   5                63
6      Kyle Figueroa   33  Female                  13                41
7      Shannon Allen   23  Female                   4                92
8        Daryl Noble   30  Female                   7                32
9     Tracy Figueroa   39  Female                  17                10

   Satisfaction Rate (%)  Feedback Score Department          Position  \
0                     25             4.7  Marketing           Analyst
1                     76             2.8         IT           Manager
2                     10             2.4         IT           Analyst
3                      4             1.4  Marketing            Intern
4                      9             4.5         IT         Team Lead
5                     33             4.2      Sales  Junior Developer
6                     39             2.3      Sales           Analyst
```

```
7                     68             2.8         HR            Intern
8                     43             2.3  Marketing  Junior Developer
9                     15             1.1         HR         Team Lead

  Joining Date  Salary
0       Jan-20   63596
1       Jan-99  112540
2       Jan-17   66292
3       Jan-22   38303
4       Jan-05  101133
5       Jan-21   48740
6       Jan-18   73502
7       Jan-21   39670
8       Jan-19   49323
9       Jan-04   92915
```

[3]:

```
# Check its datatype and value count
hr_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Name                   200 non-null    object
 1   Age                    200 non-null    int64
 2   Gender                 200 non-null    object
 3   Projects Completed     200 non-null    int64
 4   Productivity (%)       200 non-null    int64
 5   Satisfaction Rate (%)  200 non-null    int64
 6   Feedback Score         200 non-null    float64
 7   Department             200 non-null    object
 8   Position               200 non-null    object
 9   Joining Date           200 non-null    object
 10  Salary                 200 non-null    int64
dtypes: float64(1), int64(5), object(5)
memory usage: 17.3+ KB
```

[4]:

```
hr_data.describe(include='all')
```

[4]:

```
                   Name         Age Gender  Projects Completed  \
count               200  200.000000    200          200.000000
unique              200         NaN      2                 NaN
top     Douglas Lindsey         NaN   Male                 NaN
freq                  1         NaN    100                 NaN
mean                NaN   34.650000    NaN           11.455000
std                 NaN    9.797318    NaN            6.408849
min                 NaN   22.000000    NaN            0.000000
```

```
25%                 NaN   26.000000    NaN            6.000000
50%                 NaN   32.000000    NaN           11.000000
75%                 NaN   41.000000    NaN           17.000000
max                 NaN   60.000000    NaN           25.000000

        Productivity (%)  Satisfaction Rate (%)  Feedback Score Department  \
count         200.000000             200.000000      200.000000        200
unique               NaN                    NaN             NaN          5
top                  NaN                    NaN             NaN      Sales
freq                 NaN                    NaN             NaN         47
mean           46.755000              49.935000        2.883000        NaN
std            28.530068              28.934353        1.123263        NaN
min             0.000000               0.000000        1.000000        NaN
25%            23.000000              25.750000        1.900000        NaN
50%            45.000000              50.500000        2.800000        NaN
75%            70.000000              75.250000        3.900000        NaN
max            98.000000             100.000000        4.900000        NaN

       Position Joining Date         Salary
count       200          200     200.000000
unique        6           25            NaN
top     Manager       Jan-18            NaN
freq         40           23            NaN
mean        NaN          NaN   76619.245000
std         NaN          NaN   27082.299202
min         NaN          NaN   30231.000000
25%         NaN          NaN   53080.500000
50%         NaN          NaN   80540.000000
75%         NaN          NaN  101108.250000
max         NaN          NaN  119895.000000
```

[5]:

```
hr_data.shape
```

[5]:

```
(200, 11)
```

1.0.3 Data Preparation

[6]:

```
# Check if there is a column with NaN value/s
hr_data.isnull().sum()
```

[6]:

```
Name                     0
Age                      0
Gender                   0
Projects Completed       0
Productivity (%)         0
Satisfaction Rate (%)    0
Feedback Score           0
```

```
Department             0
Position               0
Joining Date           0
Salary                 0
dtype: int64
```

```
[7]: # Check if there is a column with duplicated value/s
     hr_data.duplicated().sum()
```

```
[7]: 0
```

```
[8]: # Drop the Name column
     hr_data = hr_data.drop(['Name'],axis=1)
```

```
[9]: # Convert Joining Date into Date Time
     hr_data['Joining Date'] = pd.to_datetime(hr_data['Joining Date'],␣
      ↪format='%b-%y')
```

```
[10]: #Insert new column
      hr_data['Year'] = hr_data['Joining Date'].dt.year

      # Get the current date
      current_date = datetime.datetime.now()
      hr_data['Experience'] = current_date.year - hr_data['Year']
```

```
[11]: # Create a range
      bins = [0, 5, 10, 15, 20, 30]
      labels = ['1-5', '6-10', '11-15', '16-20', '21-25']
      hr_data['Experience_Range'] = pd.cut(hr_data['Experience'], bins=bins,␣
       ↪labels=labels, right=False)
      hr_data.head(5)
```

```
[11]:    Age  Gender  Projects Completed  Productivity (%)  Satisfaction Rate (%)  \
      0   25    Male                  11                57                     25
      1   59  Female                  19                55                     76
      2   30    Male                   8                87                     10
      3   26  Female                   1                53                      4
      4   43    Male                  14                 3                      9

         Feedback Score Department   Position Joining Date  Salary  Year  \
      0             4.7  Marketing    Analyst   2020-01-01   63596  2020
      1             2.8         IT    Manager   1999-01-01  112540  1999
      2             2.4         IT    Analyst   2017-01-01   66292  2017
      3             1.4  Marketing     Intern   2022-01-01   38303  2022
      4             4.5         IT  Team Lead   2005-01-01  101133  2005

         Experience Experience_Range
```

```
0           3            1-5
1          24          21-25
2           6           6-10
3           1            1-5
4          18          16-20
```

[12]:

```python
# Drop the some columns
hr_data = hr_data.drop(['Joining Date', 'Year', 'Experience'],axis=1)
hr_data
```

[12]:

```
     Age  Gender  Projects Completed  Productivity (%)  Satisfaction Rate (%)  \
0     25    Male                  11                57                     25
1     59  Female                  19                55                     76
2     30    Male                   8                87                     10
3     26  Female                   1                53                      4
4     43    Male                  14                 3                      9
..   …     …                   …                 …                     …
195   29  Female                   9                32                     87
196   26    Male                   7                45                     28
197   22    Male                   3                36                     77
198   36  Female                  23                96                     50
199   43    Male                  10                86                     71

     Feedback Score Department          Position  Salary Experience_Range
0               4.7  Marketing           Analyst   63596              1-5
1               2.8         IT           Manager  112540            21-25
2               2.4         IT           Analyst   66292             6-10
3               1.4  Marketing            Intern   38303              1-5
4               4.5         IT         Team Lead  101133            16-20
..              …        …               …     …              …
195             3.5         HR  Junior Developer   50051              1-5
196             2.8         IT  Junior Developer   46612              1-5
197             1.6    Finance            Intern   32992              1-5
198             3.4  Marketing           Manager  104216            16-20
199             2.0         IT         Team Lead  104341             6-10

[200 rows x 10 columns]
```

[13]:

```python
# Save a copy of the preprocessed data in a csv format
hr_data.to_csv('hr_data_processed.csv', index = True)
hr_data = pd.read_csv('hr_data_processed.csv')
hr_data.head(5)
```

[13]:

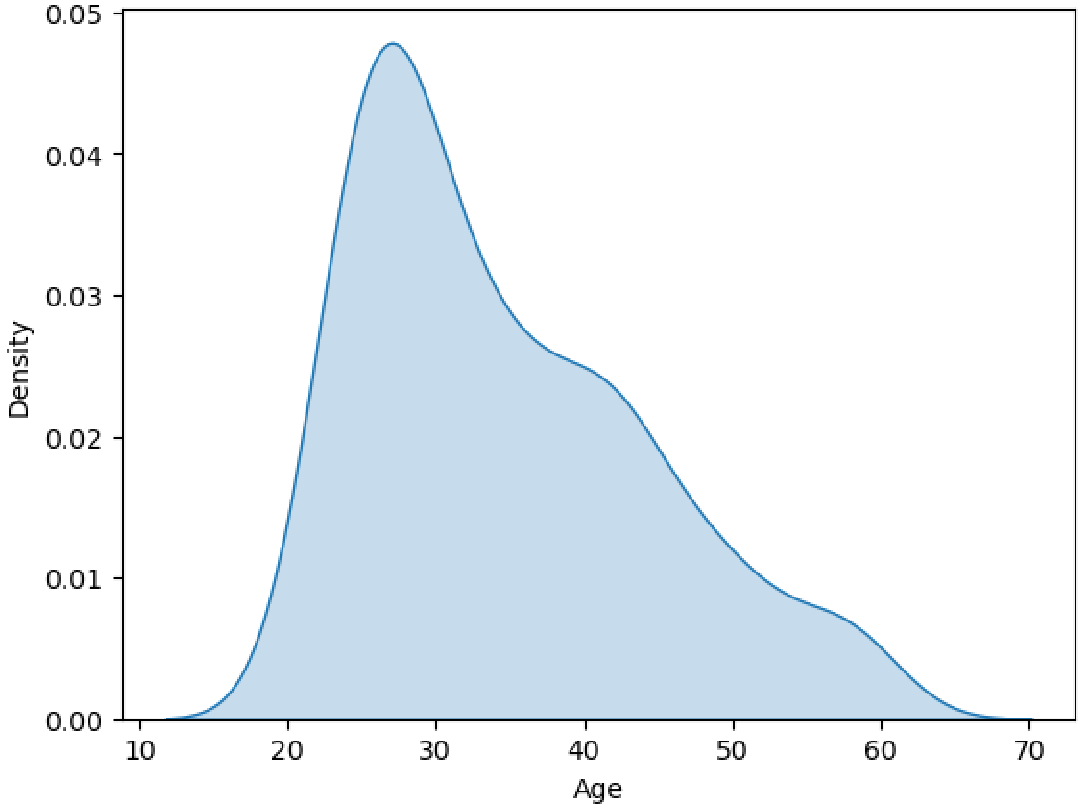
```
   Unnamed: 0  Age  Gender  Projects Completed  Productivity (%)  \
0           0   25    Male                  11                57
1           1   59  Female                  19                55
2           2   30    Male                   8                87
```

```
3           3   26  Female                        1                53
4           4   43    Male                       14                 3

   Satisfaction Rate (%)  Feedback Score Department   Position  Salary  \
0                     25             4.7  Marketing    Analyst   63596
1                     76             2.8         IT    Manager  112540
2                     10             2.4         IT    Analyst   66292
3                      4             1.4  Marketing     Intern   38303
4                      9             4.5         IT  Team Lead  101133

  Experience_Range
0              1-5
1            21-25
2             6-10
3              1-5
4            16-20
```

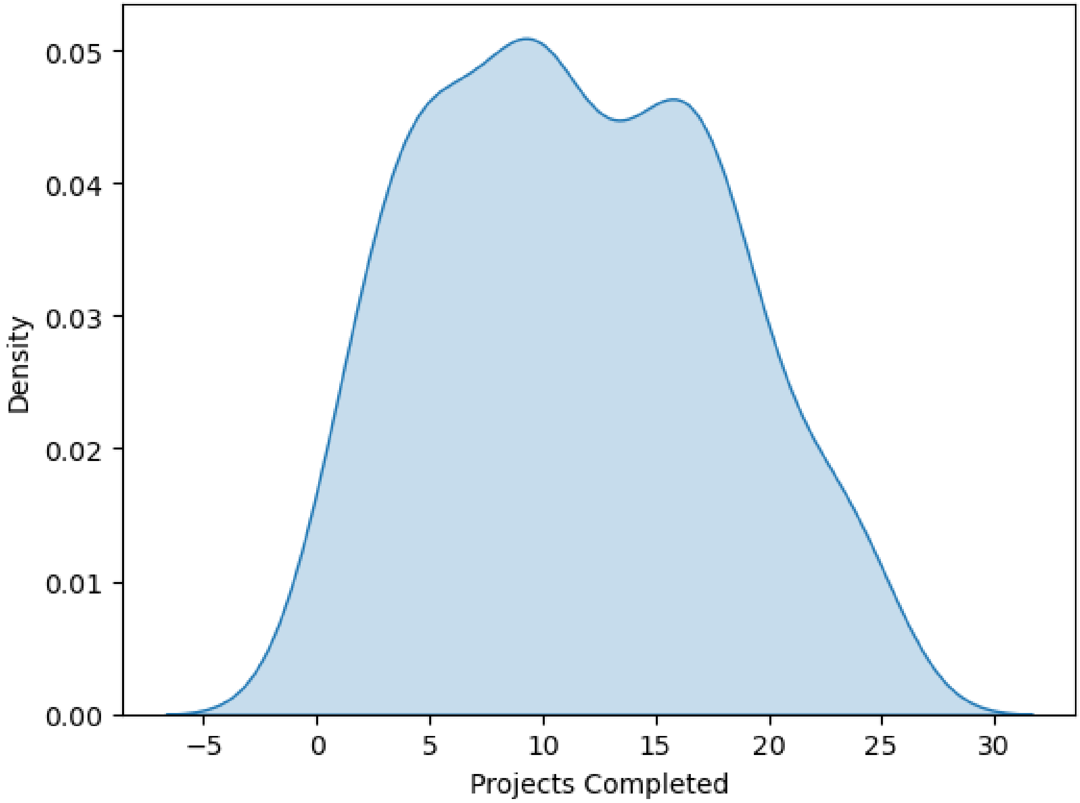
1.0.4 Exploratory Data Analysis

[14]:

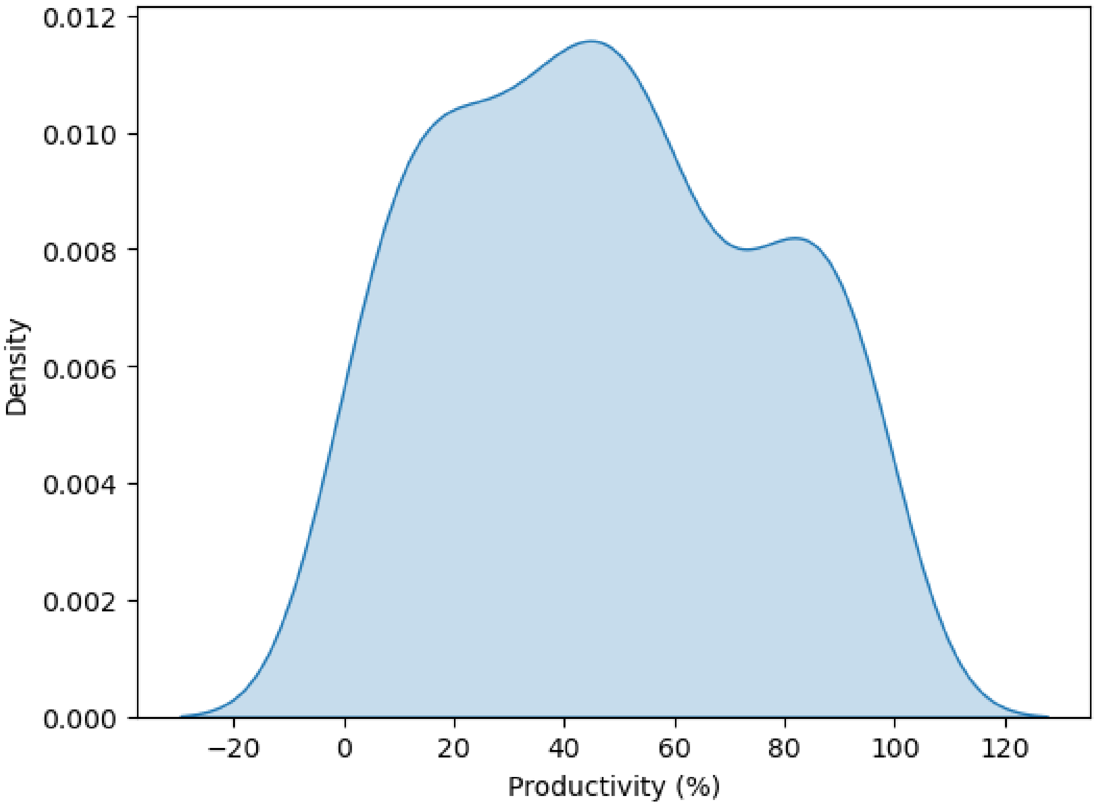
```python
# Use kdeplot to show the density of Age
sns.kdeplot(data=hr_data['Age'], fill = True)
plt.show()
```



[15]:
```
# Use kdeplot to show the density of Projects Completed
sns.kdeplot(data=hr_data['Projects Completed'], fill = True)
plt.show()
```



[16]:
```
# Use kdeplot to show the density of Productivity (%)
sns.kdeplot(data=hr_data['Productivity (%)'], fill = True)
plt.show()
```



1.0.5 Composition of Employee Based on their Gender

[17]:
```
# Count the male and female employee
male_count = (hr_data['Gender'] == 'Male').sum()
female_count = (hr_data['Gender'] == 'Female').sum()
print(f'Total number of male employees are {male_count}')
print(f'Total number of female employees are {female_count}')
```

```
Total number of male employees are 100
Total number of female employees are 100
```

[18]:
```
# Data
labels = ['Male', 'Female']
sizes = [male_count, female_count]
colors = ['#43C6DB', '#E3F9A6']

# Create a pie chart with hover labels
fig = px.pie(names=labels, values=sizes, color_discrete_sequence=colors,␣
 ↪template='plotly_dark',
             title="Employee Gender Composition",
             labels={'names': 'Gender', 'values': 'Counts'})
```

```
# Show the plot
fig.show()
```

1.0.6 Total number of Projects Completed based on their Gender

[19]:
```
gender_data = hr_data.groupby('Gender').sum(numeric_only = True)['Projects␣
 ↪Completed'].reset_index()
gender_data.columns = ['Gender','Project_Completed']
gender_data
```

[19]:
```
   Gender  Project_Completed
0  Female               1194
1    Male               1097
```

[20]:
```
# Data
custom_colors = ['#43C6DB', '#E3F9A6']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(gender_data, x = 'Gender', y = 'Project_Completed', color =␣
 ↪'Gender', template='plotly_dark',
              color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Gender')
fig.update_yaxes(title_text='Projects Completed')
fig.update_layout(title='Total Projects Completed by Gender')

# Show the plot
fig.show()
```

1.0.7 Average Productivity Rate of the Employees by Gender

[21]:
```
gender_data1 = hr_data.groupby('Gender').mean(numeric_only =␣
 ↪True)['Productivity (%)'].reset_index()
gender_data1.columns = ['Gender','Productivity_Rate']
gender_data1
```

[21]:
```
   Gender  Productivity_Rate
0  Female              42.97
1    Male              50.54
```

[22]:
```
# Data
custom_colors = ['#43C6DB', '#E3F9A6']

# Create a bar plot with the 'plotly_dark' template
```

```
fig = px.bar(gender_data1, x = 'Gender', y = 'Productivity_Rate', color =␣
↪'Gender', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Gender')
fig.update_yaxes(title_text='Productivity (%)')
fig.update_layout(title='Average Productivity by Gender')

# Show the plot
fig.show()
```

1.0.8 Average Satisfaction Rate By Gender

[23]:

```
gender_data2 = hr_data.groupby('Gender').mean(numeric_only =␣
↪True)['Satisfaction Rate (%)'].reset_index()
gender_data2.columns = ['Gender','Satisfaction_Rate']
gender_data2
```

[23]:

```
   Gender  Satisfaction_Rate
0  Female              51.25
1    Male              48.62
```

[24]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(gender_data2, x = 'Gender', y = 'Satisfaction_Rate', color =␣
↪'Gender', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Gender')
fig.update_yaxes(title_text='Satisfaction Rate (%)')
fig.update_layout(title='Average Satisfaction Rate by Gender')

# Show the plot
fig.show()
```

1.0.9 Feedback Scores of Employees by Gender

[25]:

```
gender_data3 = hr_data.groupby('Gender').mean(numeric_only = True)['Feedback␣
↪Score'].reset_index()
gender_data3.columns = ['Gender','Feedback_Score']
gender_data3
```

[25]:

```
   Gender  Feedback_Score
0  Female           3.010
1    Male           2.756
```

[26]:

```
# Create a bar plot with the 'plotly_dark' template
custom_colors = ['#43C6DB', '#E3F9A6']

fig = px.bar(gender_data3, x= 'Gender', y = 'Feedback_Score', color= 'Gender',␣
 ↪template='plotly_dark',
             color_discrete_sequence = custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Gender')
fig.update_yaxes(title_text='Feedback Scores')
fig.update_layout(title='Average Feedback Scores by Gender')

# Show the plot
fig.show()
```

1.0.10 Composition of Employees based on their Position

[27]:

```
position_count = hr_data['Position'].value_counts().reset_index()
position_count.columns = ['Position','Count']
position_count
```

[27]:

```
           Position  Count
0           Manager     40
1  Junior Developer     35
2           Analyst     33
3         Team Lead     32
4            Intern     30
5  Senior Developer     30
```

[28]:

```
# Data
labels = position_count['Position']
sizes = position_count['Count']
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a pie chart with hover labels
fig = px.pie(names=labels, values=sizes, color_discrete_sequence=colors,␣
 ↪template='plotly_dark',
             title="Employee Position Composition",
             labels={'names': 'Postion', 'values': 'Counts'})

# Show the plot
fig.show()
```

1.0.11 Total Number of Projects Completed by Position

```
[29]: position_count1 = hr_data.groupby('Position').sum(numeric_only =␣
        ↪True)['Projects Completed'].reset_index()
      position_count1.columns = ['Position','Projects_Completed']
      position_count1
```

```
[29]:             Position  Projects_Completed
      0            Analyst                 310
      1             Intern                  81
      2   Junior Developer                 237
      3            Manager                 799
      4   Senior Developer                 385
      5          Team Lead                 479
```

```
[30]: # Data
      custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

      # Create a bar plot with the 'plotly_dark' template
      fig = px.bar(position_count1, x = 'Position', y = 'Projects_Completed', color =␣
        ↪'Position', template='plotly_dark',
                   color_discrete_sequence=custom_colors)

      # Update the axis labels and title
      fig.update_xaxes(title_text='Position')
      fig.update_yaxes(title_text='Projects Completed')
      fig.update_layout(title='Total Projects Completed by Position')

      # Show the plot
      fig.show()
```

1.0.12 Average Productivity Rate of the Employees by Position

```
[31]: position_count2 = hr_data.groupby('Position').mean(numeric_only =␣
        ↪True)['Productivity (%)'].reset_index().round(2)
      position_count2.columns = ['Position','Productivity']
      position_count2
```

```
[31]:             Position  Productivity
      0            Analyst         44.55
      1             Intern         41.33
      2   Junior Developer         52.31
      3            Manager         48.68
      4   Senior Developer         50.23
      5          Team Lead         42.38
```

[32]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(position_count2, x = 'Position', y = 'Productivity', color =␣
  ↪'Position', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Position')
fig.update_yaxes(title_text='Productivity (%)')
fig.update_layout(title='Total Projects Completed by Position')

# Show the plot
fig.show()
```

1.0.13 Average Satisfaction Rate By Position

[33]:

```
position_count3 = hr_data.groupby('Position').mean(numeric_only =␣
  ↪True)['Satisfaction Rate (%)'].reset_index().round(2)
position_count3.columns = ['Position','Satisfaction_Rate']
position_count3
```

[33]:

```
           Position  Satisfaction_Rate
0           Analyst              39.79
1            Intern              55.27
2  Junior Developer              54.06
3           Manager              55.58
4  Senior Developer              50.40
5         Team Lead              43.41
```

[34]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(position_count3, x = 'Position', y = 'Satisfaction_Rate', color =␣
  ↪'Position', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Position')
fig.update_yaxes(title_text='Satisfaction Rate (%)')
fig.update_layout(title='Average Satisfaction of Employees by Position')

# Show the plot
fig.show()
```

1.0.14 Average Feedback Scores of Employee by Position

[35]:

```
position_count4 = hr_data.groupby('Position').mean(numeric_only =␣
 ↪True)['Feedback Score'].reset_index().round(2)
position_count4.columns = ['Position','Feedback_Score']
position_count4
```

[35]:

```
           Position  Feedback_Score
0           Analyst            2.77
1            Intern            2.95
2  Junior Developer            2.85
3           Manager            3.02
4  Senior Developer            2.63
5         Team Lead            3.03
```

[36]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(position_count4, x = 'Position', y = 'Feedback_Score', color =␣
 ↪'Position', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Position')
fig.update_yaxes(title_text='Satisfaction Rate (%)')
fig.update_layout(title='Average Satisfaction of Employees by Position')

# Show the plot
fig.show()
```

1.0.15 Composition of Employees based on their Department

[37]:

```
dep_count = hr_data['Department'].value_counts().reset_index()
dep_count.columns = ['Department','Count']
dep_count
```

[37]:

```
  Department  Count
0      Sales     47
1  Marketing     42
2    Finance     41
3         IT     38
4         HR     32
```

[38]:

```
# Data
labels = dep_count['Department']
sizes = dep_count['Count']
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']
```

```
# Create a pie chart with hover labels
fig = px.pie(names=labels, values=sizes, color_discrete_sequence=colors,␣
 ↪template='plotly_dark',
             title="Employee Department Composition",
             labels={'names': 'Department', 'values': 'Counts'})

# Show the plot
fig.show()
```

1.0.16 Total number of Projects Completed based on their Department

[39]:
```
dep_count1 = hr_data.groupby('Department').sum(numeric_only = True)['Projects␣
 ↪Completed'].reset_index()
dep_count1.columns = ['Department','Projects_Completed']
dep_count1
```

[39]:
```
  Department  Projects_Completed
0    Finance                 505
1         HR                 337
2         IT                 443
3  Marketing                 493
4      Sales                 513
```

[40]:
```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(dep_count1, x = 'Department', y = 'Projects_Completed', color =␣
 ↪'Department', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Department')
fig.update_yaxes(title_text='Projects Completed')
fig.update_layout(title='Total Projects Completed by Department')

# Show the plot
fig.show()
```

1.0.17 Average Productivity Rate of Employees by Department

[41]:
```
dep_count2 = hr_data.groupby('Department').mean(numeric_only =␣
 ↪True)['Productivity (%)'].reset_index().round(2)
dep_count2.columns = ['Department','Productivity']
dep_count2
```

[41]:

```
  Department  Productivity
0    Finance         42.27
1         HR         48.12
2         IT         56.34
3  Marketing         44.26
4      Sales         44.21
```

[42]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(dep_count2, x = 'Department', y = 'Productivity', color =␣
 ↪'Department', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Department')
fig.update_yaxes(title_text='Productivity (%)')
fig.update_layout(title='Average Productivity by Department')

# Show the plot
fig.show()
```

1.0.18 Average Satisfaction Rate of Employees by Department

[43]:

```
dep_count3 = hr_data.groupby('Department').mean(numeric_only =␣
 ↪True)['Satisfaction Rate (%)'].reset_index().round(2)
dep_count3.columns = ['Department','Satisfaction_Rate']
dep_count3
```

[43]:

```
  Department  Satisfaction_Rate
0    Finance              50.05
1         HR              51.62
2         IT              54.34
3  Marketing              46.02
4      Sales              48.62
```

[44]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(dep_count3, x = 'Department', y = 'Satisfaction_Rate', color =␣
 ↪'Department', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Department')
fig.update_yaxes(title_text='Satisfaction_Rate (%)')
fig.update_layout(title='Average Satisfaction Rate by Department')

# Show the plot
fig.show()
```

1.0.19 Average Feedback Scores of Employees by Department

[45]:

```
dep_count4 = hr_data.groupby('Department').mean(numeric_only = True)['Feedback␣
 ↪Score'].reset_index().round(2)
dep_count4.columns = ['Department','Feedback_Score']
dep_count4
```

[45]:

```
  Department  Feedback_Score
0    Finance            2.71
1         HR            2.62
2         IT            3.01
3  Marketing            3.14
4      Sales            2.88
```

[46]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(dep_count4, x = 'Department', y = 'Feedback_Score', color =␣
 ↪'Department', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Department')
fig.update_yaxes(title_text='Feedback Score')
fig.update_layout(title='Average Feedback Score by Department')

# Show the plot
fig.show()
```

1.0.20 Project Completion and Salary Relationship

[47]:

```
# Custom colors
custom_colors = ['#43C6DB', '#E3F9A6']

# Create a scatter plot with the 'plotly_dark' template and custom colors
fig = px.scatter(hr_data, x='Projects Completed', y='Salary', color='Gender',␣
 ↪template='plotly_dark', color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_layout(title='Project Completion and Salary Relationship')
```

```

# Show the plot
fig.show()
```

1.0.21 Feedback Scores and Salary Relationship

[48]:
```
# Custom colors
custom_colors = ['#43C6DB', '#E3F9A6']

# Create a scatter plot with the 'plotly_dark' template and custom colors
fig = px.scatter(hr_data, x='Salary', y='Feedback Score', color='Gender',␣
 ↪template='plotly_dark', color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_layout(title='Feedback Scores and Salary Relationship')

# Show the plot
fig.show()
```

1.0.22 Salary and Productivity Rate Relationship

[49]:
```
# Custom colors
custom_colors = ['#43C6DB', '#E3F9A6']

# Create a scatter plot with the 'plotly_dark' template and custom colors
fig = px.scatter(hr_data, x='Salary', y='Productivity (%)', color='Gender',␣
 ↪template='plotly_dark', color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_layout(title='Feedback Scores and Salary Relationship')

# Show the plot
fig.show()
```

1.0.23 Composition of Employees based on their Department

[50]:
```
exp_count = hr_data['Experience_Range'].value_counts().reset_index()
exp_count.columns = ['Experience_Range','Count']
exp_count
```

[50]:
```
  Experience_Range  Count
0              1-5     72
1             6-10     46
2            16-20     36
3            11-15     30
4            21-25     16
```

[51]:

```
# Data
labels = exp_count['Experience_Range']
sizes = exp_count['Count']
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a pie chart with hover labels
fig = px.pie(names=labels, values=sizes, color_discrete_sequence=colors,␣
 ↪template='plotly_dark',
             title="Employee Working Experience Composition",
             labels={'names': 'Experience_Range', 'values': 'Counts'})

# Show the plot
fig.show()
```

1.0.24 Total number of Projects Completed based on their Working Experience

[52]:

```
# Define the custom sorting order
custom_order = ['1-5', '6-10', '11-15', '16-20', '21-25']

# Create a Categorical data type based on the custom order
hr_data['Experience_Range'] = pd.Categorical(hr_data['Experience_Range'],␣
 ↪categories=custom_order, ordered=True)

# Sort the DataFrame based on the custom order
hr_data_sorted = hr_data.sort_values(by='Experience_Range')
hr_data_sorted.head(5)
```

[52]:

```
     Unnamed: 0  Age  Gender  Projects Completed  Productivity (%)  \
0             0   25    Male                  11                57
62           62   28  Female                   5                13
134         134   26  Female                   5                53
59           59   23  Female                   8                23
137         137   25    Male                   4                31

     Satisfaction Rate (%)  Feedback Score Department          Position  \
0                       25             4.7  Marketing           Analyst
62                      40             4.1         IT           Analyst
134                     56             3.9         HR            Intern
59                      62             2.0      Sales  Junior Developer
137                     90             2.8         IT            Intern

     Salary Experience_Range
0     63596              1-5
62    70439              1-5
134   38714              1-5
59    59877              1-5
137   32010              1-5
```

[53]:
```
hr_data_sorted = hr_data.groupby('Experience_Range').sum(numeric_only =␣
 ↪True)['Projects Completed'].reset_index()
hr_data_sorted.columns = ['Experience_Range','Projects_Completed']
hr_data_sorted
```

[53]:
```
  Experience_Range  Projects_Completed
0              1-5                 417
1             6-10                 502
2            11-15                 493
3            16-20                 590
4            21-25                 289
```

[54]:
```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(hr_data_sorted, x = 'Experience_Range', y = 'Projects_Completed',␣
 ↪color = 'Experience_Range', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Work Experience')
fig.update_yaxes(title_text='Projects Completed')
fig.update_layout(title='Total Projects Completed by Working Experience')

# Show the plot
fig.show()
```

1.0.25 Average Productivity Rate of Employees by Working Experience

[55]:
```
hr_data_sorted1 = hr_data.groupby('Experience_Range').mean(numeric_only =␣
 ↪True)['Productivity (%)'].reset_index().round(2)
hr_data_sorted1.columns = ['Experience_Range','Productivity']
hr_data_sorted1
```

[55]:
```
  Experience_Range  Productivity
0              1-5         43.14
1             6-10         52.00
2            11-15         48.50
3            16-20         41.31
4            21-25         56.94
```

[56]:
```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
```

```
fig = px.bar(hr_data_sorted1, x = 'Experience_Range', y = 'Productivity', color␣
 ↪= 'Experience_Range', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Work Experience')
fig.update_yaxes(title_text='Productivity (%)')
fig.update_layout(title='Average Productivity by Working Experience')

# Show the plot
fig.show()
```

1.0.26 Average Satisfaction Rate of Employees By Working Experience

[57]:

```
hr_data_sorted2 = hr_data.groupby('Experience_Range').mean(numeric_only =␣
 ↪True)['Satisfaction Rate (%)'].reset_index().round(2)
hr_data_sorted2.columns = ['Experience_Range','Satisfaction_Rate']
hr_data_sorted2
```

[57]:

```
  Experience_Range  Satisfaction_Rate
0              1-5              49.25
1             6-10              48.15
2            11-15              56.30
3            16-20              43.81
4            21-25              60.00
```

[58]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(hr_data_sorted2, x = 'Experience_Range', y = 'Satisfaction_Rate',␣
 ↪color = 'Experience_Range', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Work Experience')
fig.update_yaxes(title_text='Satisfaction Rate (%)')
fig.update_layout(title='Average Satisfaction Rate by Working Experience')

# Show the plot
fig.show()
```

1.0.27 Average Feedback Scores of Employees by Working Experience

[59]:

```
hr_data_sorted3 = hr_data.groupby('Experience_Range').mean(numeric_only =␣
  ↪True)['Feedback Score'].reset_index().round(2)
hr_data_sorted3.columns = ['Experience_Range','Feedback_Score']
hr_data_sorted3
```

[59]:

```
  Experience_Range  Feedback_Score
0              1-5            2.85
1             6-10            2.98
2            11-15            2.84
3            16-20            2.91
4            21-25            2.77
```

[60]:

```
# Data
custom_colors = ['#43C6DB', '#E3F9A6','#99E0EB','#FFFFFF','#1E899A','#C5F346']

# Create a bar plot with the 'plotly_dark' template
fig = px.bar(hr_data_sorted3, x = 'Experience_Range', y = 'Feedback_Score',␣
  ↪color = 'Experience_Range', template='plotly_dark',
             color_discrete_sequence=custom_colors)

# Update the axis labels and title
fig.update_xaxes(title_text='Work Experience')
fig.update_yaxes(title_text='Feedback Scores')
fig.update_layout(title='Average Feedback Scores by Working Experience')

# Show the plot
fig.show()
```