

ibm-attribution-3

June 23, 2025

```
[20]: import pandas as pd
df= pd.read_csv("ibm.csv")
```

```
[8]: ibm
```

```
[8]:      Age  Attrition  BusinessTravel  DailyRate  Department \
0      41        Yes    Travel_Rarely      1102         Sales
1      49         No  Travel_Frequently       279  Research & Development
2      37        Yes    Travel_Rarely     1373  Research & Development
3      33         No  Travel_Frequently     1392  Research & Development
4      27         No    Travel_Rarely       591  Research & Development
...  ...
1465   36         No  Travel_Frequently       884  Research & Development
1466   39         No    Travel_Rarely       613  Research & Development
1467   27         No    Travel_Rarely       155  Research & Development
1468   49         No  Travel_Frequently     1023         Sales
1469   34         No    Travel_Rarely       628  Research & Development
```

```
      DistanceFromHome  Education  EducationField  EmployeeCount \
0                    1          2  Life Sciences              1
1                    8          1  Life Sciences              1
2                    2          2         Other              1
3                    3          4  Life Sciences              1
4                    2          1         Medical              1
...
1465                23          2         Medical              1
1466                 6          1         Medical              1
1467                 4          3  Life Sciences              1
1468                 2          3         Medical              1
1469                 8          3         Medical              1
```

```
      EmployeeNumber  ...  RelationshipSatisfaction  StandardHours \
0                   1  ...                        1              80
1                   2  ...                        4              80
2                   4  ...                        2              80
3                   5  ...                        3              80
4                   7  ...                        4              80
```

| | | | | |
|------|------|-----|-----|-----|
| ... | ... | ... | ... | ... |
| 1465 | 2061 | ... | 3 | 80 |
| 1466 | 2062 | ... | 1 | 80 |
| 1467 | 2064 | ... | 2 | 80 |
| 1468 | 2065 | ... | 4 | 80 |
| 1469 | 2068 | ... | 1 | 80 |

| | StockOptionLevel | TotalWorkingYears | TrainingTimesLastYear | \ |
|------|------------------|-------------------|-----------------------|---|
| 0 | 0 | 8 | 0 | |
| 1 | 1 | 10 | 3 | |
| 2 | 0 | 7 | 3 | |
| 3 | 0 | 8 | 3 | |
| 4 | 1 | 6 | 3 | |
| ... | ... | ... | ... | |
| 1465 | 1 | 17 | 3 | |
| 1466 | 1 | 9 | 5 | |
| 1467 | 1 | 6 | 0 | |
| 1468 | 0 | 17 | 3 | |
| 1469 | 0 | 6 | 3 | |

| | WorkLifeBalance | YearsAtCompany | YearsInCurrentRole | \ |
|------|-----------------|----------------|--------------------|---|
| 0 | 1 | 6 | 4 | |
| 1 | 3 | 10 | 7 | |
| 2 | 3 | 0 | 0 | |
| 3 | 3 | 8 | 7 | |
| 4 | 3 | 2 | 2 | |
| ... | ... | ... | ... | |
| 1465 | 3 | 5 | 2 | |
| 1466 | 3 | 7 | 7 | |
| 1467 | 3 | 6 | 2 | |
| 1468 | 2 | 9 | 6 | |
| 1469 | 4 | 4 | 3 | |

| | YearsSinceLastPromotion | YearsWithCurrManager |
|------|-------------------------|----------------------|
| 0 | 0 | 5 |
| 1 | 1 | 7 |
| 2 | 0 | 0 |
| 3 | 3 | 0 |
| 4 | 2 | 2 |
| ... | ... | ... |
| 1465 | 0 | 3 |
| 1466 | 1 | 7 |
| 1467 | 0 | 3 |
| 1468 | 0 | 8 |
| 1469 | 1 | 2 |

[1470 rows x 35 columns]

```
[38]: import matplotlib.pyplot as plt
import seaborn as sns

# Map education levels to their meanings
education_mapping = {
    1: 'Below College',
    2: 'College',
    3: 'Bachelor',
    4: 'Master',
    5: 'Doctor'
}
df['EducationLevel'] = df['Education'].map(education_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['EducationLevel', 'Attrition'])['MonthlyIncome'].
    .mean().reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

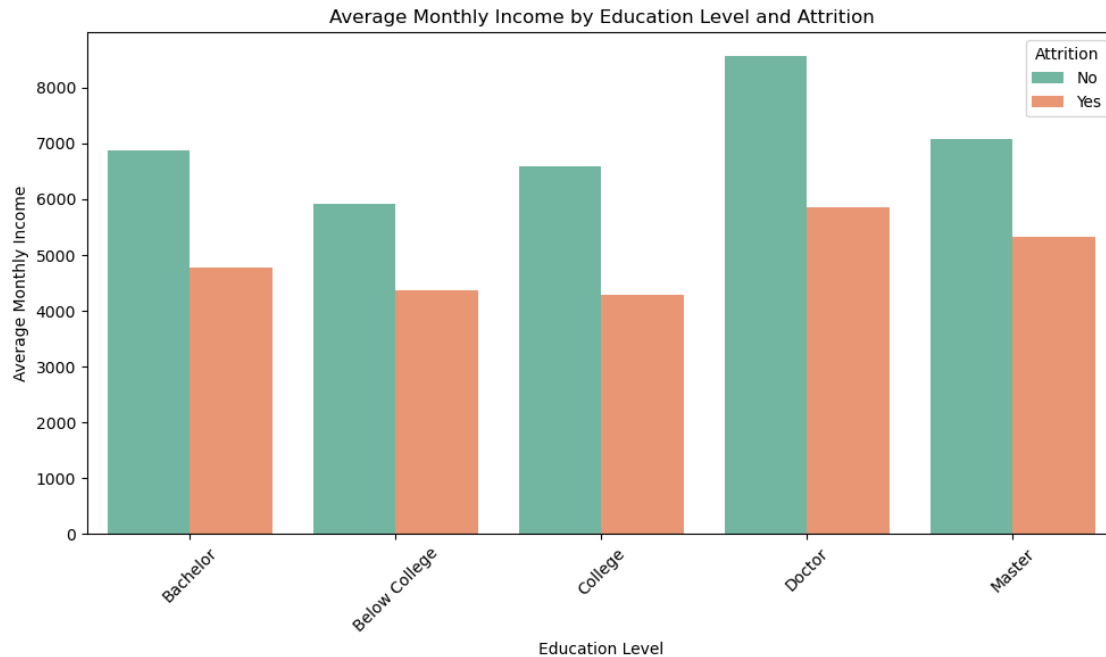
print(avg_income)

plt.figure(figsize=(10, 6))
sns.barplot(data=avg_income, x='EducationLevel', y='MonthlyIncome',
    hue='Attrition', palette='Set2')

# Add labels and title
plt.title('Average Monthly Income by Education Level and Attrition')
plt.xlabel('Education Level')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=45)
plt.tight_layout()

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

| | EducationLevel | Attrition | MonthlyIncome |
|---|----------------|-----------|---------------|
| 0 | Bachelor | No | 6882.92 |
| 1 | Bachelor | Yes | 4770.24 |
| 2 | Below College | No | 5926.13 |
| 3 | Below College | Yes | 4360.16 |
| 4 | College | No | 6586.06 |
| 5 | College | Yes | 4282.55 |
| 6 | Doctor | No | 8559.91 |
| 7 | Doctor | Yes | 5850.20 |
| 8 | Master | No | 7087.81 |
| 9 | Master | Yes | 5335.16 |



```
[40]: import pandas as pd

# Map education levels to their meanings
enviromental_mapping = {
1: 'Low',
2: 'Medium',
3: 'High',
4: 'Very High'
}
df['SatisfactionLevel'] = df['EnvironmentSatisfaction'].
    ↪map(enviromental_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['SatisfactionLevel', 'Attrition'])['MonthlyIncome'].
    ↪mean().reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

print(avg_income)

plt.figure(figsize=(10, 6))
```

```

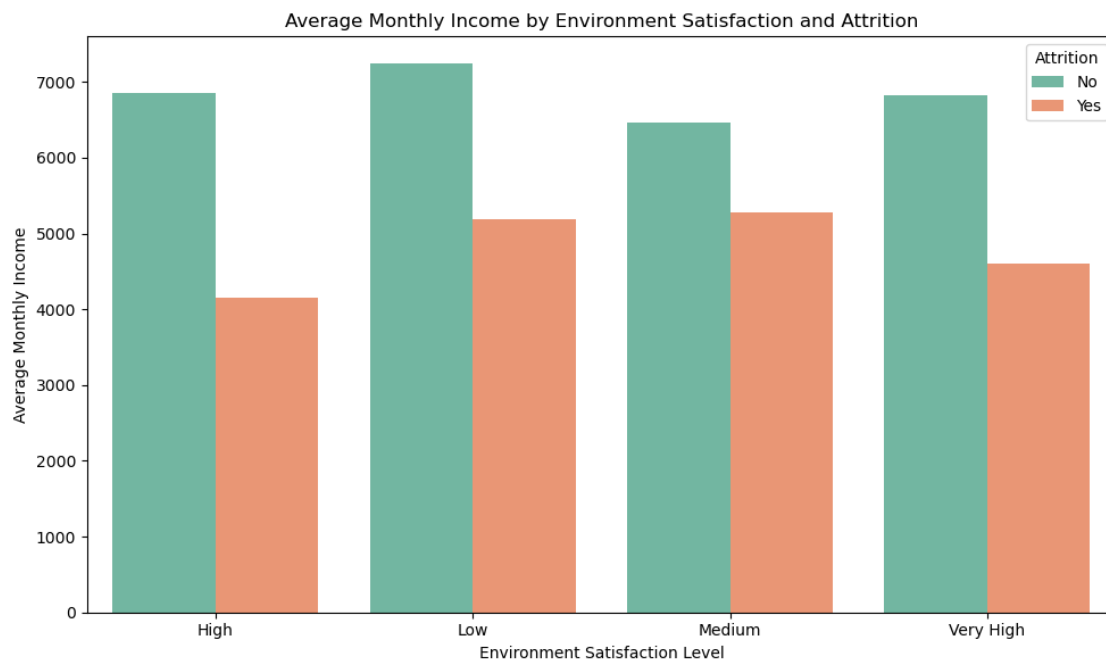
sns.barplot(data=avg_income, x='SatisfactionLevel', y='MonthlyIncome',
            hue='Attrition', palette='Set2')

# Add title and labels
plt.title('Average Monthly Income by Environment Satisfaction and Attrition')
plt.xlabel('Environment Satisfaction Level')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=0)
plt.tight_layout()

# Show plot
plt.show()

```

| | SatisfactionLevel | Attrition | MonthlyIncome |
|---|-------------------|-----------|---------------|
| 0 | High | No | 6851.10 |
| 1 | High | Yes | 4156.94 |
| 2 | Low | No | 7236.89 |
| 3 | Low | Yes | 5186.53 |
| 4 | Medium | No | 6460.00 |
| 5 | Medium | Yes | 5283.51 |
| 6 | Very High | No | 6827.79 |
| 7 | Very High | Yes | 4603.17 |



```

[42]: # Map education levels to their meanings
JobInvolvement_mapping = {

```

```

1: 'Low',
2: 'Medium',
3: 'High',
4: 'Very High'
}
df['jobInvolve'] = df['JobInvolvement'].map(JobInvolvement_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['jobInvolve', 'Attrition'])['MonthlyIncome'].mean().
    ↪reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

print(avg_income)
sns.set_style("whitegrid")

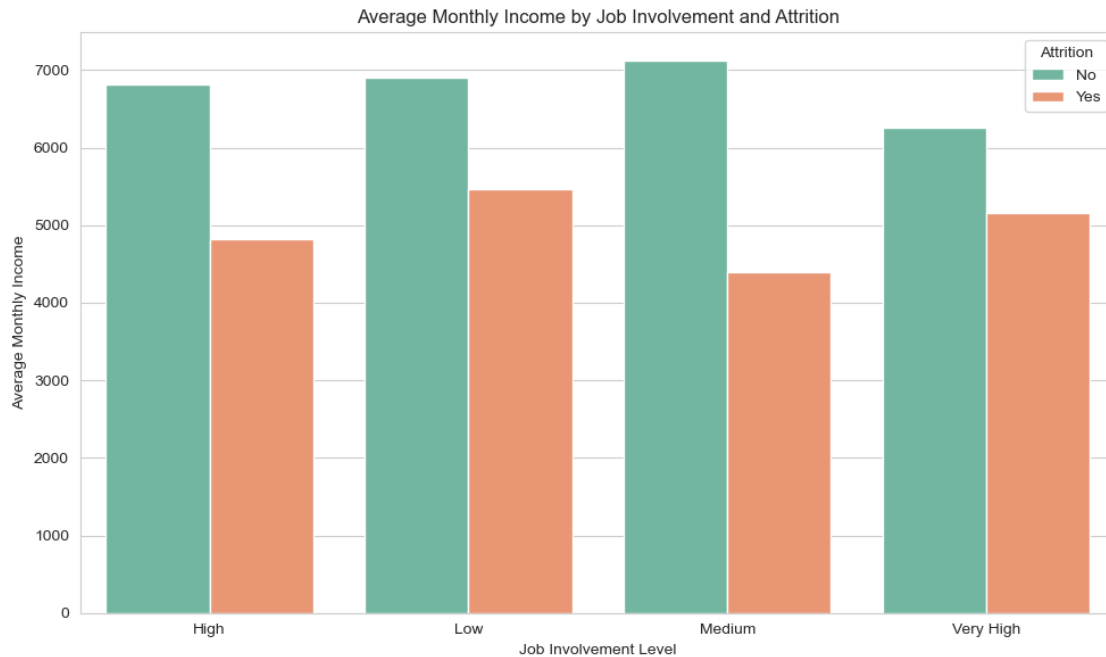
# Create the grouped bar chart
plt.figure(figsize=(10, 6))
sns.barplot(data=avg_income, x='jobInvolve', y='MonthlyIncome',
    ↪hue='Attrition', palette='Set2')

# Add labels and title
plt.title('Average Monthly Income by Job Involvement and Attrition')
plt.xlabel('Job Involvement Level')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=0)
plt.tight_layout()

# Show the plot
plt.show()

```

| | jobInvolve | Attrition | MonthlyIncome |
|---|------------|-----------|---------------|
| 0 | High | No | 6808.26 |
| 1 | High | Yes | 4817.93 |
| 2 | Low | No | 6902.98 |
| 3 | Low | Yes | 5465.68 |
| 4 | Medium | No | 7126.31 |
| 5 | Medium | Yes | 4397.46 |
| 6 | Very High | No | 6260.81 |
| 7 | Very High | Yes | 5157.00 |



```
[44]: # Map education levels to their meanings
JobSatisfaction_mapping = {
    1: 'Low',
    2: 'Medium',
    3: 'High',
    4: 'Very High'
}
df['jobSatisfy'] = df['JobSatisfaction'].map(JobSatisfaction_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['jobSatisfy', 'Attrition'])['MonthlyIncome'].mean().
    ↪reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

print(avg_income)

# Set style
sns.set_style("whitegrid")

# Create clustered bar plot
plt.figure(figsize=(10, 6))
```

```

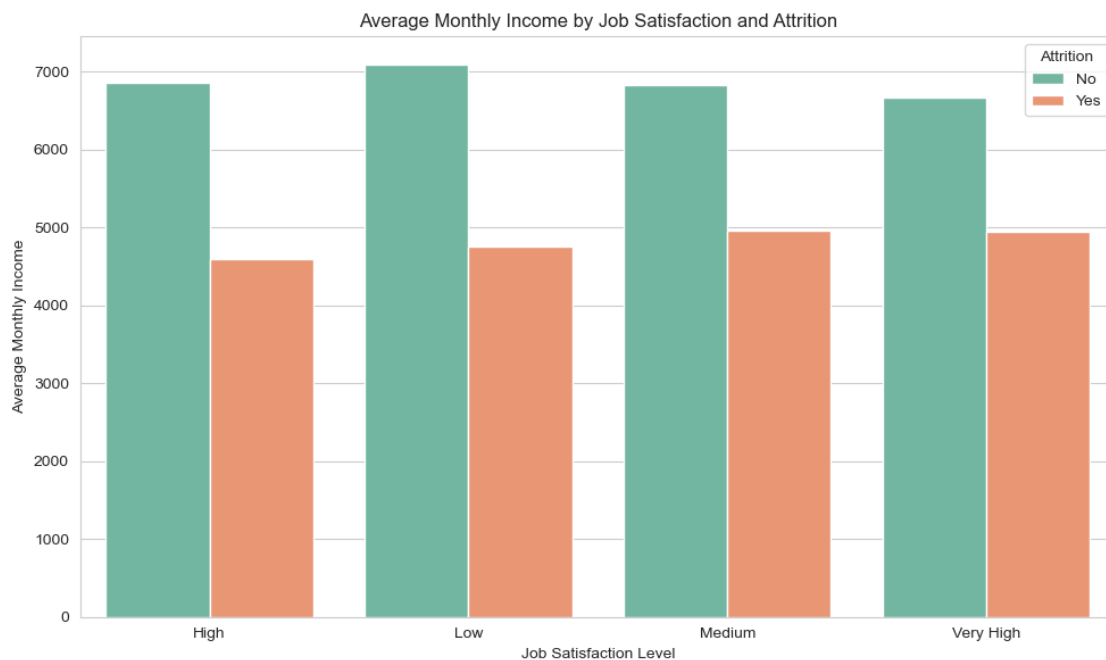
sns.barplot(data=avg_income, x='jobSatisfy', y='MonthlyIncome', hue='Attrition',
            palette='Set2')

# Add labels and title
plt.title('Average Monthly Income by Job Satisfaction and Attrition')
plt.xlabel('Job Satisfaction Level')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=0)
plt.tight_layout()

# Show the plot
plt.show()

```

| | jobSatisfy | Attrition | MonthlyIncome |
|---|------------|-----------|---------------|
| 0 | High | No | 6853.31 |
| 1 | High | Yes | 4595.99 |
| 2 | Low | No | 7096.90 |
| 3 | Low | Yes | 4752.80 |
| 4 | Medium | No | 6834.89 |
| 5 | Medium | Yes | 4962.76 |
| 6 | Very High | No | 6668.11 |
| 7 | Very High | Yes | 4943.50 |



```

[46]: # Map education levels to their meaning
RelationshipSatisfaction_mapping = {

```



```

1: 'Low',
2: 'Medium',
3: 'High',
4: 'Very High'
}
df['RelationSatisfy'] = df['RelationshipSatisfaction'].
    ↪map(RelationshipSatisfaction_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['RelationSatisfy', 'Attrition'])['MonthlyIncome'].
    ↪mean().reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

print(avg_income)

sns.set_style("whitegrid")

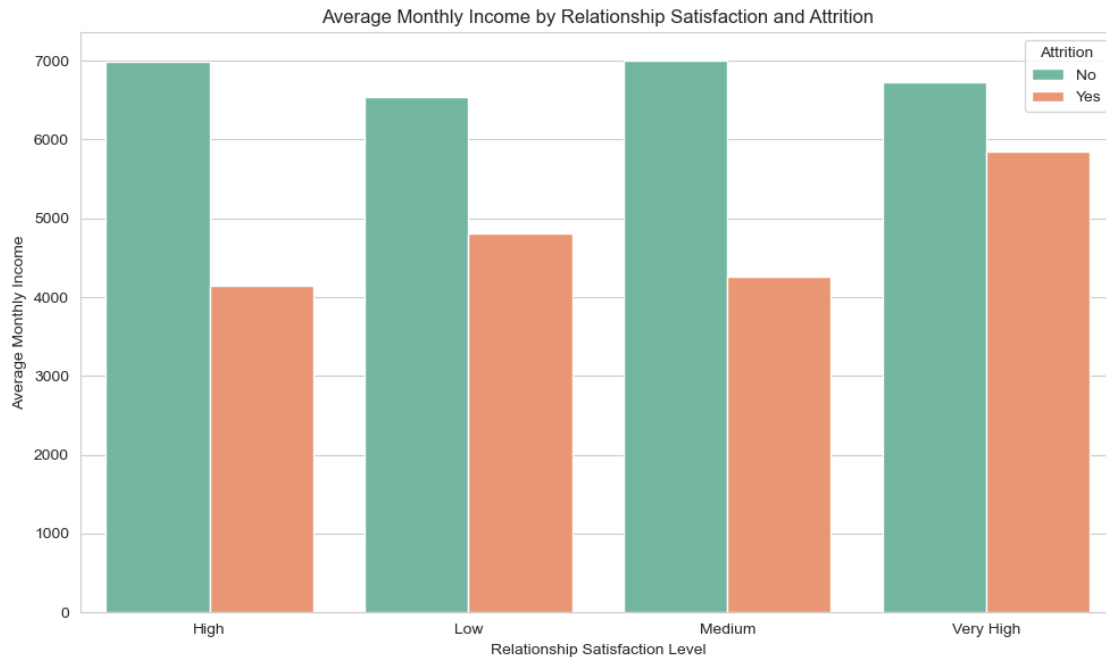
# Create clustered bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=avg_income, x='RelationSatisfy', y='MonthlyIncome',
    ↪hue='Attrition', palette='Set2')

# Add labels and title
plt.title('Average Monthly Income by Relationship Satisfaction and Attrition')
plt.xlabel('Relationship Satisfaction Level')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=0)
plt.tight_layout()

# Show the plot
plt.show()

```

| | RelationSatisfy | Attrition | MonthlyIncome |
|---|-----------------|-----------|---------------|
| 0 | High | No | 6981.96 |
| 1 | High | Yes | 4149.17 |
| 2 | Low | No | 6534.10 |
| 3 | Low | Yes | 4808.82 |
| 4 | Medium | No | 7005.07 |
| 5 | Medium | Yes | 4257.27 |
| 6 | Very High | No | 6732.31 |
| 7 | Very High | Yes | 5847.97 |



```
[48]: # Map education levels to their meanin
PerformanceRating_mapping = {
1: 'Low',
2: 'Good',
3: 'Excellent',
4: 'Outstanding',
}
df['Rating'] = df['JobSatisfaction'].map(PerformanceRating_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['Rating', 'Attrition'])['MonthlyIncome'].mean().
    ↪reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

print(avg_income)

sns.set_style("whitegrid")

# Clustered bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=avg_income, x='Rating', y='MonthlyIncome', hue='Attrition',
    ↪palette='Set2')
```

```

# Add title and labels
plt.title('Average Monthly Income by Performance Rating and Attrition')
plt.xlabel('Performance Rating')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=0)
plt.tight_layout()

# Show plot
plt.show()

```

| | Rating | Attrition | MonthlyIncome |
|---|-------------|-----------|---------------|
| 0 | Excellent | No | 6853.31 |
| 1 | Excellent | Yes | 4595.99 |
| 2 | Good | No | 6834.89 |
| 3 | Good | Yes | 4962.76 |
| 4 | Low | No | 7096.90 |
| 5 | Low | Yes | 4752.80 |
| 6 | Outstanding | No | 6668.11 |
| 7 | Outstanding | Yes | 4943.50 |



```

[50]: # Map education levels to their meanin
WorkLifeBalance_mapping = {
1: 'Bad',
2: 'Good',
3: 'Better',

```

```

4: 'Best',
}
df['WorkBalance'] = df['WorkLifeBalance'].map(WorkLifeBalance_mapping)

# Group by EducationLevel and Attrition, then calculate average MonthlyIncome
avg_income = df.groupby(['WorkBalance', 'Attrition'])['MonthlyIncome'].mean().
    ↪reset_index()

# Optional: Format MonthlyIncome for better readability
avg_income['MonthlyIncome'] = avg_income['MonthlyIncome'].round(2)

print(avg_income)

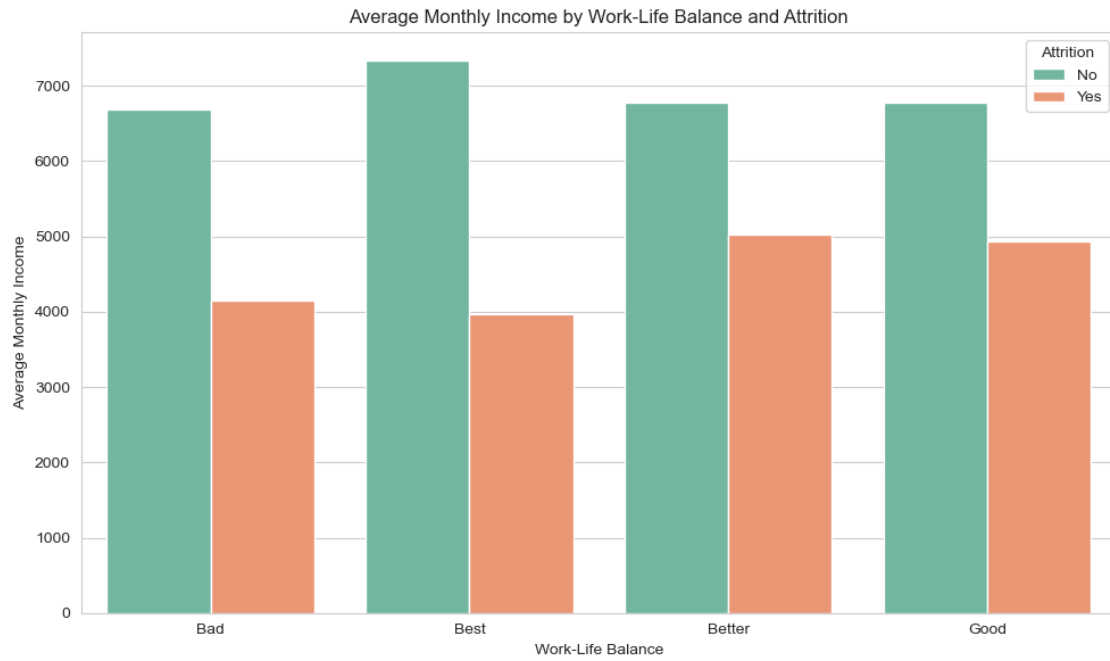
sns.set_style("whitegrid")

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(data=avg_income, x='WorkBalance', y='MonthlyIncome',
    ↪hue='Attrition', palette='Set2')

plt.title('Average Monthly Income by Work-Life Balance and Attrition')
plt.xlabel('Work-Life Balance')
plt.ylabel('Average Monthly Income')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()

```

| | WorkBalance | Attrition | MonthlyIncome |
|---|-------------|-----------|---------------|
| 0 | Bad | No | 6679.67 |
| 1 | Bad | Yes | 4143.56 |
| 2 | Best | No | 7340.17 |
| 3 | Best | Yes | 3975.19 |
| 4 | Better | No | 6782.62 |
| 5 | Better | Yes | 5022.01 |
| 6 | Good | No | 6772.85 |
| 7 | Good | Yes | 4928.05 |



[]: