Sleep Health and Lifestyle

👦💻 About the Dataset

- 👤 Person ID : Individual identification number. A unique identifier for each person.
- 🚻 Gender : Gender information. Categorized as Male or Female.
- 🎂 Age : Age information. The person's age is indicated.
- 👔 Occupation : Occupation information. The type of job or profession the person holds.
- 🛌 Sleep Duration : Sleep duration. The total amount of time a person sleeps during the night (usually in hours).
- 🌙 Quality of Sleep : Sleep quality. Indicates how efficient and restful a person's sleep process is.
- 🏃 Physical Activity Level : Physical activity level. Indicates the amount of physical activity a person does daily (for example, low, moderate, high).
- 😰 Stress Level : Stress level. An indicator of how stressed a person is in their daily lives.
- ⚖️ BMI Category : Body mass index (BMI) category. A health indicator that divides a person into categories such as overweight, normal, and obese.
- ❤️ Blood Pressure : Blood pressure. Indicates a person's blood pressure level (e.g. high, normal, and low).
- ❤️ Heart Rate : Heart rate. The number of times a person's heart beats per minute.
- 🚶 Daily Steps : Daily step count. Indicates the total number of steps a person takes in a day.

- 🛌 Sleep Disorder : Sleep disorder. Any problem or discomfort a person experiences with their sleep patterns.

1.1 Libraries and Utilities

In [17]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
from pywaffle import Waffle
import warnings
warnings.filterwarnings("ignore")
warnings.warn("this will not show")
```

Loading data

In [18]:
```
df = pd.read_csv("Sleep_health_and_lifestyle_dataset.csv")
df.head()
```

Out[18]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BM Catego
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweig
1	2	Male	28	Doctor	6.2	6	60	8	Norm
2	3	Male	28	Doctor	6.2	6	60	8	Norm
3	4	Male	28	Sales Representative	5.9	4	30	8	Obe
4	5	Male	28	Sales Representative	5.9	4	30	8	Obe

In [19]:
```
df1 = df.copy()      # We took a copy of our original data because we will be
```

Non-Null ValueCounts and Feature Datatypes

In [20]:
```
df.info()

# The `df.info()` method provides a quick overview of a pandas DataFrame's st
# and memory usage for each column, which is crucial for understanding the da
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Person ID                374 non-null    int64  
 1   Gender                   374 non-null    object 
 2   Age                      374 non-null    int64  
 3   Occupation               374 non-null    object 
 4   Sleep Duration           374 non-null    float64
 5   Quality of Sleep         374 non-null    int64  
 6   Physical Activity Level  374 non-null    int64  
 7   Stress Level             374 non-null    int64  
 8   BMI Category             374 non-null    object 
 9   Blood Pressure           374 non-null    object 
 10  Heart Rate               374 non-null    int64  
 11  Daily Steps              374 non-null    int64  
 12  Sleep Disorder           155 non-null    object 
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

Statistics of Categorical and Numerical Data

In [21]:

```
# stats of numerical data
round (df.describe(exclude = 'object'), 2).style.background_gradient(cmap='Bu
```

Out[21]:

	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.00
mean	187.500000	42.180000	7.130000	7.310000	59.170000	5.390000	70.17
std	108.110000	8.670000	0.800000	1.200000	20.830000	1.770000	4.14
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.00
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.00
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.00
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.00
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.00

In [22]:

```
# stats of categorical data
round (df.describe(exclude = ['float', 'int64']),2).style.set_properties(**{'
```

Out[22]:

	Gender	Occupation	BMI Category	Blood Pressure	Sleep Disorder
count	374	374	374	374	155
unique	2	11	4	25	2
top	Male	Nurse	Normal	130/85	Sleep Apnea

freq	189	73	195	99	78

1.Initial Insights About Dataset

1. Data from Sleep Disorder feature is missing.
2. Both Categorical and numerical features are available.

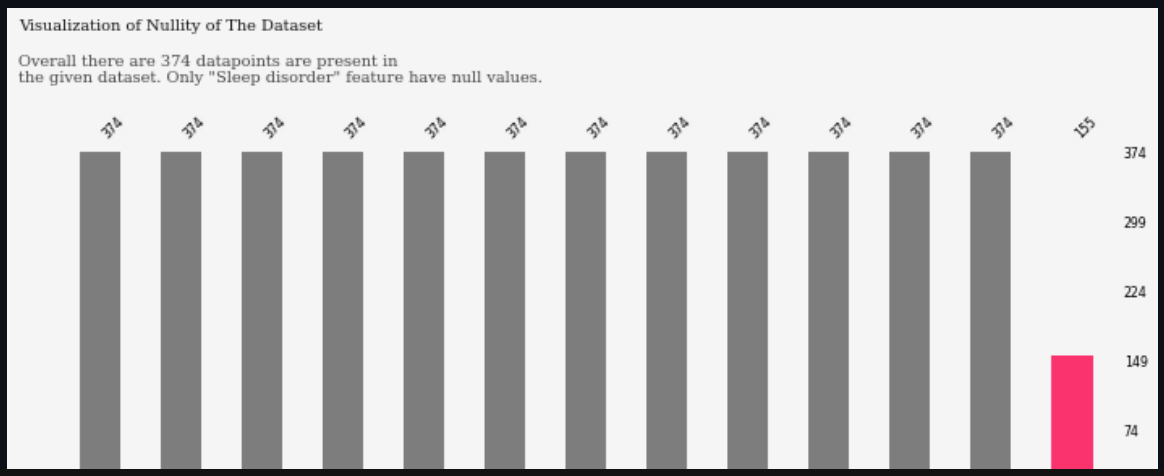
- **Categorical Features**: Gender, Occupation, BMI Category, Blood Pressure, Sleep Disorder
- **Binary Numerical Features**: Person ID, Age, Sleep Duration, Quality of Sleep, Physical Activity Level, Stress Level, Heart Rate, Daily Steps

3. Most of the data is numerical and needs special attention to visualize it

1.1 Visualization of Dataset

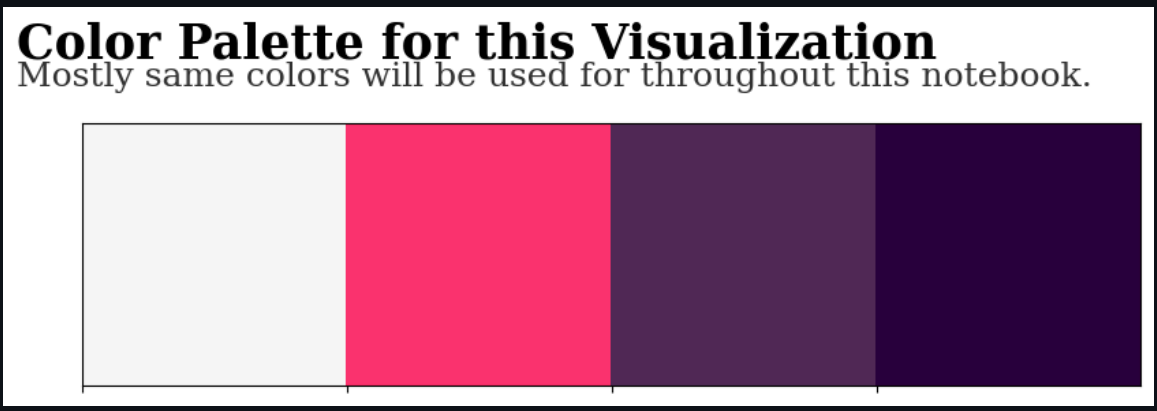
In [23]:

```
color = ['grey','grey','grey','grey','grey','grey','grey','grey','grey','grey
fig, ax = plt.subplots(figsize = (12,4), dpi = 70)
fig.patch.set_facecolor('#f6f5f5')
ax.set_facecolor('#f6f5f5')


msno.bar(df, sort = 'descending',
         color = color,
         ax = ax, fontsize =8,
         labels = 'off',filter = 'top')

ax.text(-1,1.35,'Visualization of Nullity of The Dataset',{'font': 'Serif',
ax.text(-1,1.2,'Overall there are 374 datapoints are present in \nthe given d

ax.set_xticklabels(ax.get_xticklabels(),rotation = 90,
                   ha = 'center', **{'font': 'Serif','weight':'normal','color
ax.set_yticklabels('')
ax.spines['bottom'].set_visible(True)

fig.show()
```



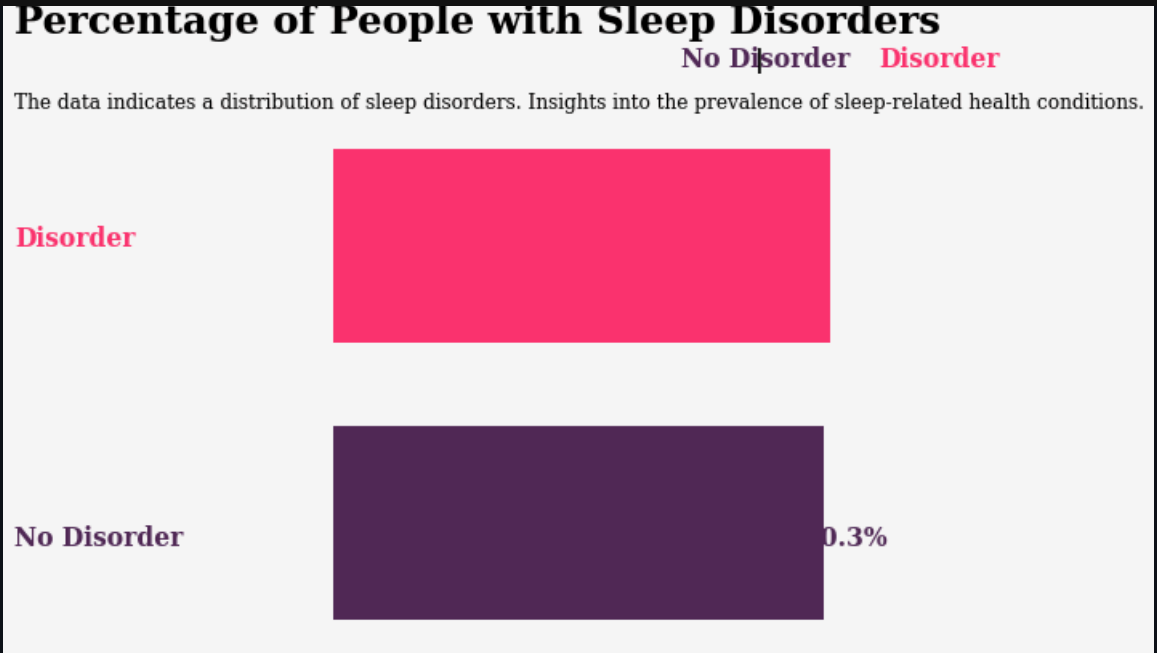
In [24]:

```
# color palette for visualizations
colors = ['#f6f5f5','#fe346e','#512b58','#2c003e']
palette = sns.color_palette( palette = colors)

sns.palplot(palette, size =2.5)
plt.text(-0.75,-0.75,'Color Palette for this Visualization', {'font':'serif',
plt.text(-0.75,-0.64,'Mostly same colors will be used for throughout this not
plt.show()
```



In [25]:

```
x = pd.DataFrame(df.groupby(['Sleep Disorder'])['Sleep Disorder'].count())

fig, ax = plt.subplots(figsize=(6, 6), dpi=70)
ax.barh([1], x['Sleep Disorder'][1], height=0.7, color='#fe346e')
plt.text(-50, -0.08, 'No Disorder', {'font': 'Serif', 'weight': 'bold', 'size
plt.text(x['Sleep Disorder'][1] * 0.95, -0.08, f'{(x["Sleep Disorder"][1] / x

ax.barh([0], x['Sleep Disorder'][0], height=0.7, color='#512b58')
plt.text(-50, 1, 'Disorder', {'font': 'Serif', 'weight': 'bold', 'size': '16'
plt.text(x['Sleep Disorder'][0] * 0.05, 1, f'{(x["Sleep Disorder"][0] / x["Sl

fig.patch.set_facecolor('#f6f5f5')
ax.set_facecolor('#f6f5f5')

plt.text(-50, 1.77, 'Percentage of People with Sleep Disorders', {'font': 'Se
plt.text(x['Sleep Disorder'][1] * 0.7, 1.65, 'No Disorder', {'font': 'Serif',
plt.text(x['Sleep Disorder'][1] * 0.85, 1.65, '|', {'color': 'black', 'size':
plt.text(x['Sleep Disorder'][1] * 1.1, 1.65, 'Disorder', {'font': 'Serif', 'w
plt.text(-50, 1.5, 'The data indicates a distribution of sleep disorders. Ins

# Removing axes and spines for cleaner look
ax.axes.get_xaxis().set_visible(False)
ax.axes.get_yaxis().set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)

plt.show()
```



Sleep Disorders Analysis 🛌❌

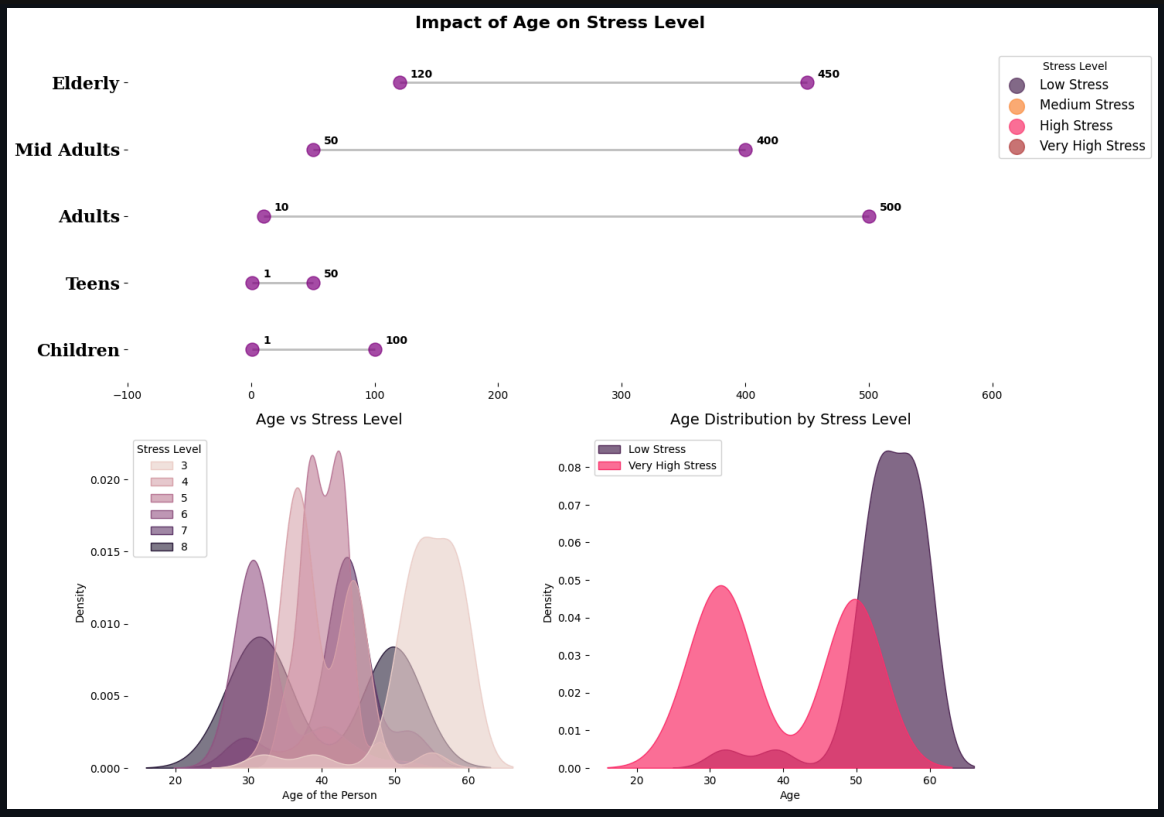
The data suggests that the vast majority of individuals do not experience **sleep disorders**. The low or near-zero rate observed in the **'Disorder'** category indicates that such health problems may be relatively rare or possibly poorly defined in this dataset.

2.1Dataset Analysis

In [26]:

```
conditions = [
    (df['Stress Level'] == 3),  # Low Stress
    (df['Stress Level'] == 4) | (df['Stress Level'] == 5),  # Medium Stress
    (df['Stress Level'] == 6) | (df['Stress Level'] == 7),  # High Stress
    (df['Stress Level'] == 8)   # Very High Stress
]

choices = ['Low Stress', 'Medium Stress', 'High Stress', 'Very High Stress']

df['Stress Category'] = np.select(conditions, choices, default='Unknown')

plt.style.use('default')
fig = plt.figure(figsize=(15, 15), facecolor='white')


ax1 = plt.subplot2grid((3, 2), (0, 0), colspan=2)
ax2 = plt.subplot2grid((3, 2), (1, 0), colspan=1)
ax3 = plt.subplot2grid((3, 2), (1, 1), colspan=1)

for ax in [ax1, ax2, ax3]:
    ax.set_facecolor('white')
    ax.grid(False)
    for spine in ax.spines.values():
        spine.set_visible(False)

low_stress = df[df['Stress Category'] == 'Low Stress'].Age.value_counts()
medium_stress = df[df['Stress Category'] == 'Medium Stress '].Age.value_count
```

```
high_stress = df[df['Stress Category'] == 'High Stress '].Age.value_counts()
very_high_stress = df[df['Stress Category'] == 'Very High Stress'].Age.value_


y_coords = ['Children', 'Teens', 'Adults', 'Mid Adults', 'Elderly']
xmin_values = [100, 50, 500, 400, 450]
xmax_values = [1, 1, 10, 50, 120]


ax1.hlines(y=y_coords, xmin=xmin_values, xmax=xmax_values,
           color='grey', alpha=0.5, linewidth=2)

for i, (y, xmin, xmax) in enumerate(zip(y_coords, xmin_values, xmax_values)):

    ax1.scatter(xmin, y, color='purple', s=150, alpha=0.7, zorder=5)
    ax1.annotate(f'{xmin}', (xmin, y),
                 xytext=(10, 5), textcoords='offset points',
                 fontsize=10, fontweight='bold')

    # Bitiş noktası
    ax1.scatter(xmax, y, color='purple', s=150, alpha=0.7, zorder=5)
    ax1.annotate(f'{xmax}', (xmax, y),
                 xytext=(10, 5), textcoords='offset points',
                 fontsize=10, fontweight='bold')

scatter_size = 200
ax1.scatter(low_stress.values, low_stress.index, s=scatter_size, color='#512b
ax1.scatter(medium_stress.values, medium_stress.index, s=scatter_size, color=
ax1.scatter(high_stress.values, high_stress.index, s=scatter_size, color='#fe
ax1.scatter(very_high_stress.values, very_high_stress.index, s=scatter_size,


ax1.set_xlim(-100, 600)
ax1.set_ylim(-0.5, 4.5)
ax1.set_yticklabels(labels=['Children', 'Teens', 'Adults', 'Mid Adults', 'Eld
                    fontdict={'font':'Serif', 'fontsize':16, 'fontweight':'bo


ax1.set_title('Impact of Age on Stress Level', pad=20, fontsize=16, fontweigh


ax1.legend(loc='upper left', bbox_to_anchor=(1, 1), fontsize=12, title='Stres

sns.kdeplot(data=df, x='Age', hue='Stress Level', fill=True, alpha=0.6, ax=ax
ax2.set_title('Age vs Stress Level', fontsize=14, pad=10)
ax2.set_xlabel('Age of the Person', fontsize=10)


sns.kdeplot(data=df[df['Stress Category'] == 'Low Stress'], x='Age', ax=ax3,
            shade=True, alpha=0.7, color='#512b58', label='Low Stress')
sns.kdeplot(data=df[df['Stress Category'] == 'Medium Stress '], x='Age', ax=a
            shade=True, alpha=0.7, color='#fe346e', label='Medium Stress')
sns.kdeplot(data=df[df['Stress Category'] == 'High Stress '], x='Age', ax=ax3
            shade=True, alpha=0.7, color='#fe346e', label='High Stress')
sns.kdeplot(data=df[df['Stress Category'] == 'Very High Stress'], x='Age', ax
            shade=True, alpha=0.7, color='#fe346e', label='Very High Stress')

ax3.set_title('Age Distribution by Stress Level', fontsize=14, pad=10)
ax3.set_xlabel('Age', fontsize=10)
ax3.legend()

plt.tight_layout()
plt.show()
```



Impact of Age and Stress Level 🧠💪

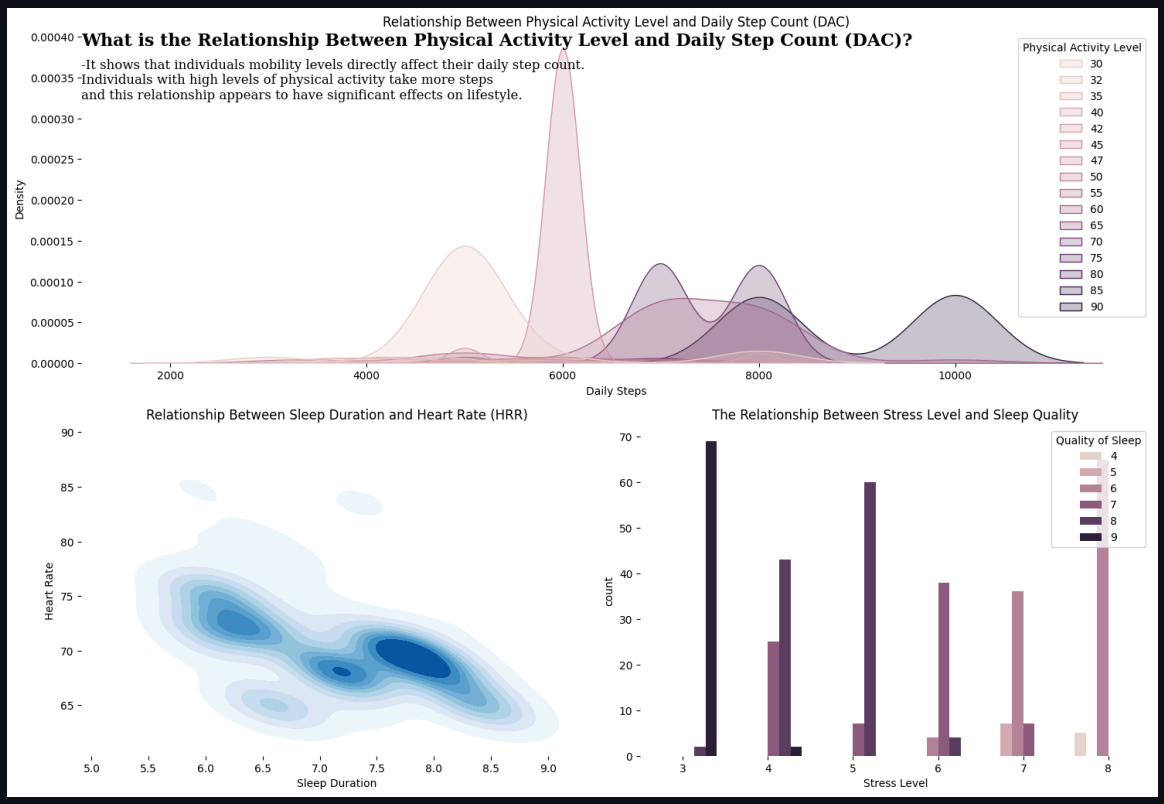
In this analysis, we explored the relationship between **age** 👴👶 and **stress level** 😰. We examined how different age groups correlate with varying stress levels, providing insights into the patterns of stress across different age categories. Additionally, we analyzed the **age distribution** by **stress level**, highlighting trends in how stress varies with age. 📊

2.2 Dataset Analysis

In [27]:

```
fig = plt.figure(figsize=(15, 15), facecolor='white')

ax1 = plt.subplot2grid((3, 2), (0, 0), colspan=2)
ax2 = plt.subplot2grid((3, 2), (1, 0), colspan=1)
ax3 = plt.subplot2grid((3, 2), (1, 1), colspan=1)

# Grafik özelliklerini ayarlama
for ax in [ax1, ax2, ax3]:
    ax.set_facecolor('white')
    ax.grid(False)
    for spine in ax.spines.values():
        spine.set_visible(False)


# Fiziksel Aktivite Seviyesi ve Günlük Adım Sayısı (KDE Plot)
sns.kdeplot(data=df, x='Daily Steps', hue='Physical Activity Level', fill=Tru
ax1.set_title('Relationship Between Physical Activity Level and Daily Step Co
ax1.text(0, 1,
         'What is the Relationship Between Physical Activity Level and Daily
         ha='left', va='top',
```

```
        fontweight='bold', fontsize=16,
        fontname='Serif', color='black', transform=ax1.transAxes)
ax1.text(0, 0.8,
        '-It shows that individuals mobility levels directly affect their da
        fontsize=12,
        fontname='Serif', color='black', transform=ax1.transAxes)

# Uyku Süresi ve Kalp Hızı (KDE Plot)
sns.kdeplot(data=df, x='Sleep Duration', y='Heart Rate', fill=True, cmap='Blu
ax2.set_title('Relationship Between Sleep Duration and Heart Rate (HRR)')

scatter_size = 200
# Stres Seviyesi ve Uyku Kalitesi
sns.countplot(x='Stress Level', hue='Quality of Sleep', data=df, ax=ax3)
ax3.set_title('The Relationship Between Stress Level and Sleep Quality')



plt.tight_layout()
plt.show()
```



Relationship Between Physical Activity Level and Daily Step Count (DAC) 🚶‍♂️🏃‍♀️

The analysis shows that an individual's **mobility levels** directly impact their **daily step count**. Those with higher levels of physical activity tend to take more steps, and this relationship has a significant effect on their overall **lifestyle** and well-being.

Relationship Between Sleep Duration and Heart Rate (HRR) 🛌❤️‍🩹

The relationship between **sleep duration** and **heart rate** reveals that longer sleep duration is generally associated with a lower heart rate, indicating better cardiovascular

health. This connection suggests that improved sleep may play a crucial role in regulating heart rate and enhancing overall health.

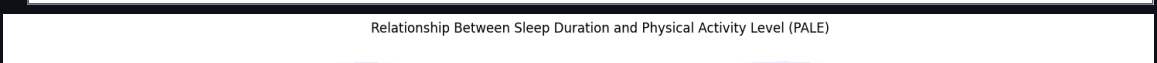
Relationship Between Stress Level and Sleep Quality 😓💤

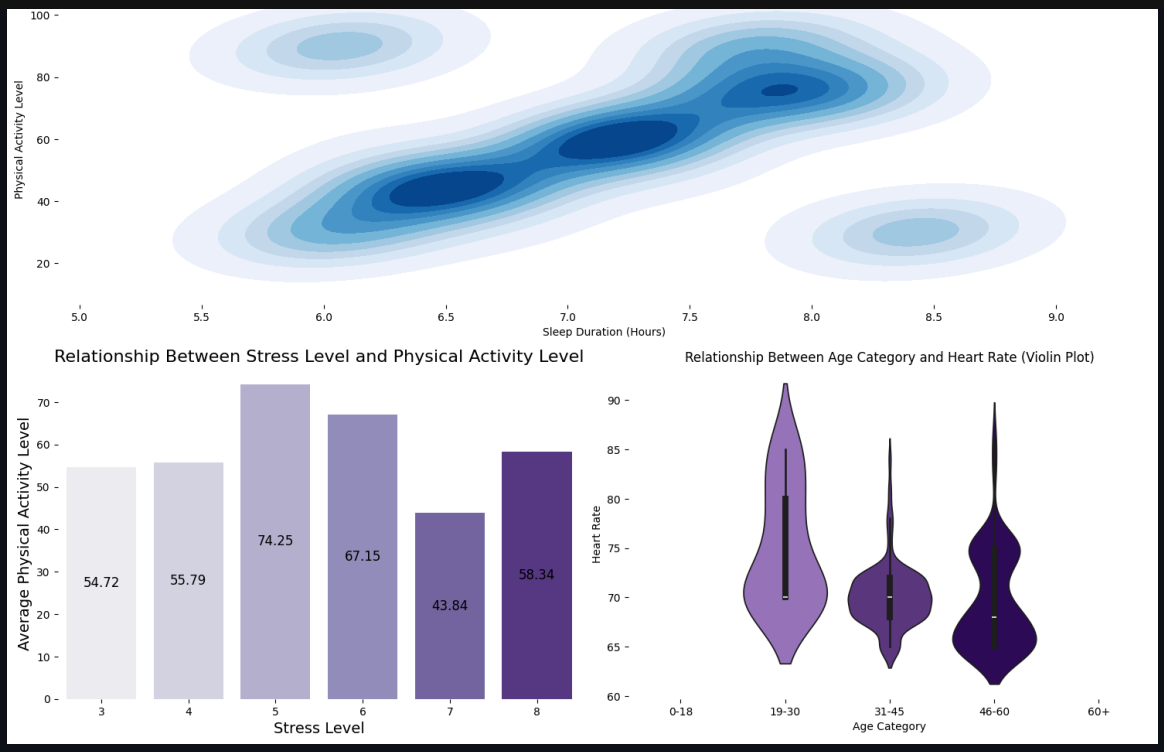
The analysis of **stress levels** and **sleep quality** highlights that individuals experiencing higher stress tend to report poorer sleep quality. This suggests a direct link between **stress** and the ability to achieve restful, restorative sleep.

2.3 Dataset Analysis

In [28]:

```
fig = plt.figure(figsize=(15, 15), facecolor='white')

ax1 = plt.subplot2grid((3, 2), (0, 0), colspan=2)
ax2 = plt.subplot2grid((3, 2), (1, 0), colspan=1)
ax3 = plt.subplot2grid((3, 2), (1, 1), colspan=1)

# Grafik özelliklerini ayarlama
for ax in [ax1, ax2, ax3]:
    ax.set_facecolor('white')
    ax.grid(False)
    for spine in ax.spines.values():
        spine.set_visible(False)

sns.kdeplot(x='Sleep Duration', y='Physical Activity Level', data=df, fill=Tr
ax1.set_title('Relationship Between Sleep Duration and Physical Activity Leve
ax1.set_xlabel('Sleep Duration (Hours)')
ax1.set_ylabel('Physical Activity Level')


sns.barplot(x='Stress Level', y='Physical Activity Level', data=df, estimator
ax2.set_title('Relationship Between Stress Level and Physical Activity Level'
ax2.set_xlabel('Stress Level', fontsize=14)
ax2.set_ylabel('Average Physical Activity Level', fontsize=14)

for p in ax2.patches:
    ax2.text(p.get_x() + p.get_width() / 2., p.get_height() / 2,
             f'{p.get_height():.2f}', ha='center', va='center', fontsize=12,

bins = [0, 18, 30, 45, 60, 100]
labels = ['0-18', '19-30', '31-45', '46-60', '60+']
df['Age Category'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

sns.violinplot(x='Age Category', y='Heart Rate', data=df, ax=ax3, palette=['#
ax3.set_title('Relationship Between Age Category and Heart Rate (Violin Plot)
ax3.set_xlabel('Age Category')
ax3.set_ylabel('Heart Rate')


plt.tight_layout()
plt.show()
```





Relationship Between Sleep Duration and Physical Activity Level (PALE)

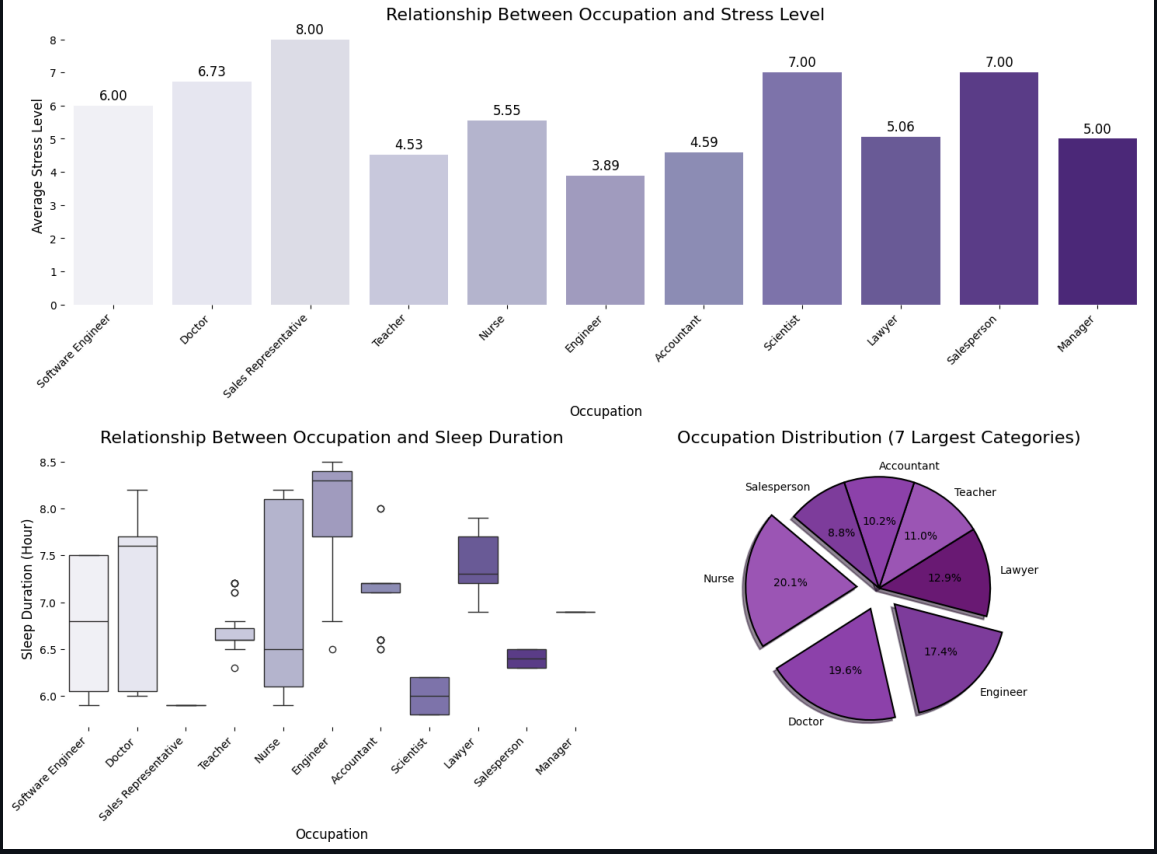
We analyzed the relationship between **sleep duration** and **physical activity level (PALE)** using a **bar chart** to visualize the patterns. Later, a **violin plot** was employed to provide a deeper insight into the distribution and variation of sleep duration across different levels of physical activity, revealing key trends in how sleep duration correlates with physical activity levels.

2.4 Dataset Analysis

In [29]:

```
fig = plt.figure(figsize=(15, 15), facecolor='white')

ax1 = plt.subplot2grid((3, 2), (0, 0), colspan=2)
ax2 = plt.subplot2grid((3, 2), (1, 0), colspan=1)
ax3 = plt.subplot2grid((3, 2), (1, 1), colspan=1)


for ax in [ax1, ax2, ax3]:
    ax.set_facecolor('white')
    ax.grid(False)
    for spine in ax.spines.values():
        spine.set_visible(False)


sns.barplot(x='Occupation', y='Stress Level', data=df, estimator=np.mean, ci=
            palette='Purples', ax=ax1)
ax1.set_title('Relationship Between Occupation and Stress Level', fontsize=16
ax1.set_xlabel('Occupation', fontsize=12)
ax1.set_ylabel('Average Stress Level', fontsize=12)
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=45, ha='right')
for p in ax1.patches:
```

```
    ax1.annotate(f'{p.get_height():.2f}',
                 (p.get_x() + p.get_width() / 2., p.get_height()),
                 ha='center', va='center', fontsize=12, color='black',
                 xytext=(0, 9), textcoords='offset points')


sns.boxplot(x='Occupation', y='Sleep Duration', data=df, palette='Purples', a
ax2.set_title('Relationship Between Occupation and Sleep Duration', fontsize=
ax2.set_xlabel('Occupation', fontsize=12)
ax2.set_ylabel('Sleep Duration (Hour)', fontsize=12)
ax2.set_xticklabels(ax2.get_xticklabels(), rotation=45, ha='right')


occupation_counts = df['Occupation'].value_counts().head(7)
colors = ['#9b59b6', '#8e44ad', '#7f3d9b', '#6a1d77', '#9b59b6', '#8e44ad', '
explode = [0.2 if i < 3 else 0 for i in range(len(occupation_counts))]

ax3.pie(occupation_counts, labels=occupation_counts.index, autopct='%1.1f%%',
        startangle=140, explode=explode, shadow=True, wedgeprops={'edgecolor':


ax3.set_title('Occupation Distribution (7 Largest Categories)', fontsize=16)
ax3.set_aspect('equal')

plt.tight_layout()
plt.show()
```



Relationship Between Occupation and Stress Level 💼😰

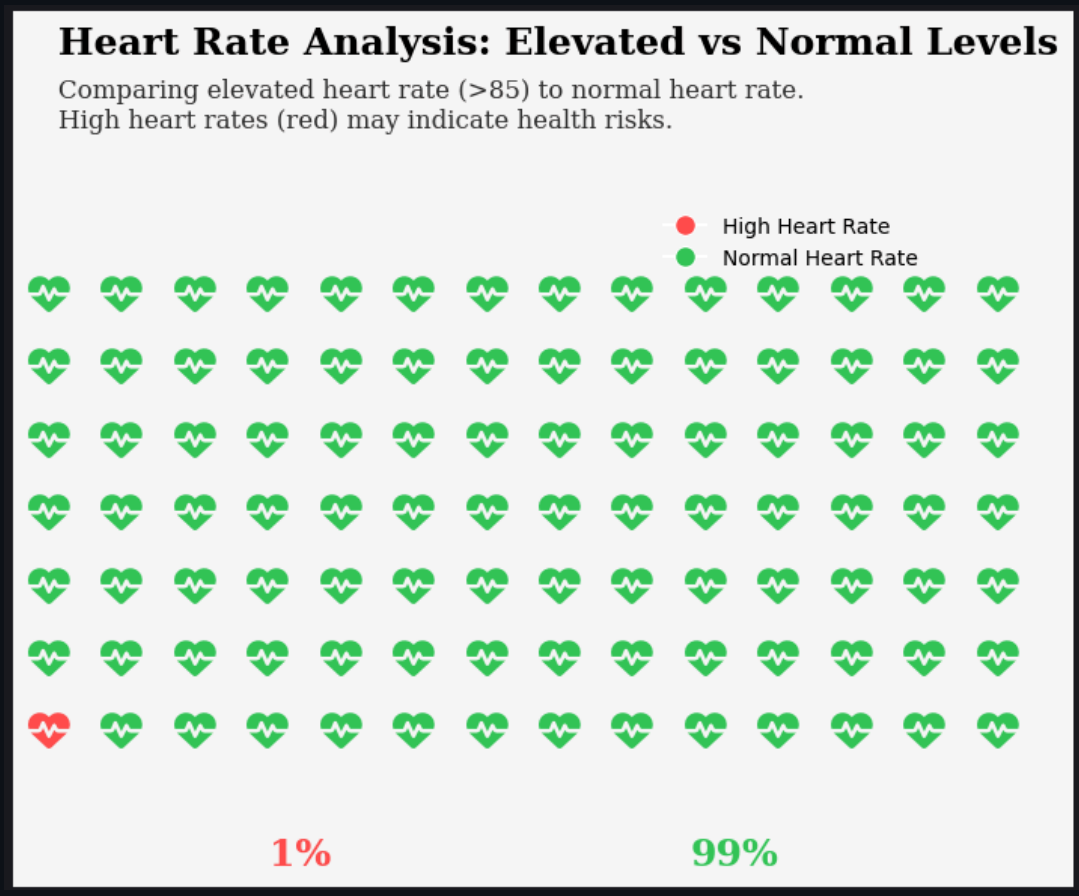
We explored the relationship between **occupation** and **stress level**, followed by an analysis of **sleep duration** for each occupation. The results, visualized with a **pie chart**, show that **sales representatives** experience the highest stress levels, while **engineers** have the lowest stress levels, highlighting significant occupation-based differences in stress and sleep patterns.

stress and sleep patterns.

2.5 Dataset Analysis

```python
import matplotlib.pyplot as plt
from pywaffle import Waffle

high_heart_rate = df[df['Heart Rate'] > 85]['Heart Rate'].count()
normal_heart_rate = df[df['Heart Rate'] <= 85]['Heart Rate'].count()

total = high_heart_rate + normal_heart_rate
high_percentage = int(round(high_heart_rate / total * 100, 0))
normal_percentage = int(round(normal_heart_rate / total * 100, 0))

fig = plt.figure(
    FigureClass=Waffle,
    rows=7,
    columns=14,
    values={"High Heart Rate": high_heart_rate, "Normal Heart Rate": normal_heart_rate},
    colors=['#ff4c4c', '#34c759'],
    icons='heartbeat',
    icon_size=20,
    icon_legend=True,
    figsize=(7, 7),
    facecolor='#f6f5f5',
    dpi=100
)

fig.text(0.05, 0.92, 'Heart Rate Analysis: Elevated vs Normal Levels', {'font': 'Serif', 'size': 18, 'color': 'black', 'weight': 'bold'})
fig.text(0.05, 0.85, 'Comparing elevated heart rate (>85) to normal heart rate.\nHigh heart rates (red) may indicate health risks.',
         {'font': 'Serif', 'size': 12, 'color': 'black'}, alpha=0.8)


fig.text(0.25, 0.15, '{}%'.format(high_percentage), {'font': 'Serif', 'size': 18, 'weight': 'bold', 'color': '#ff4c4c'})
fig.text(0.65, 0.15, '{}%'.format(normal_percentage), {'font': 'Serif', 'size': 18, 'weight': 'bold', 'color': '#34c759'})


plt.legend(
    handles=[
        plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='#ff4c4c', markersize=10, label='High Heart Rate'),
        plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='#34c759', markersize=10, label='Normal Heart Rate')
    ],
    loc='upper right', bbox_to_anchor=(0.9, 1.1), frameon=False
)

fig.show()
```

[39] ✓ 22.0s



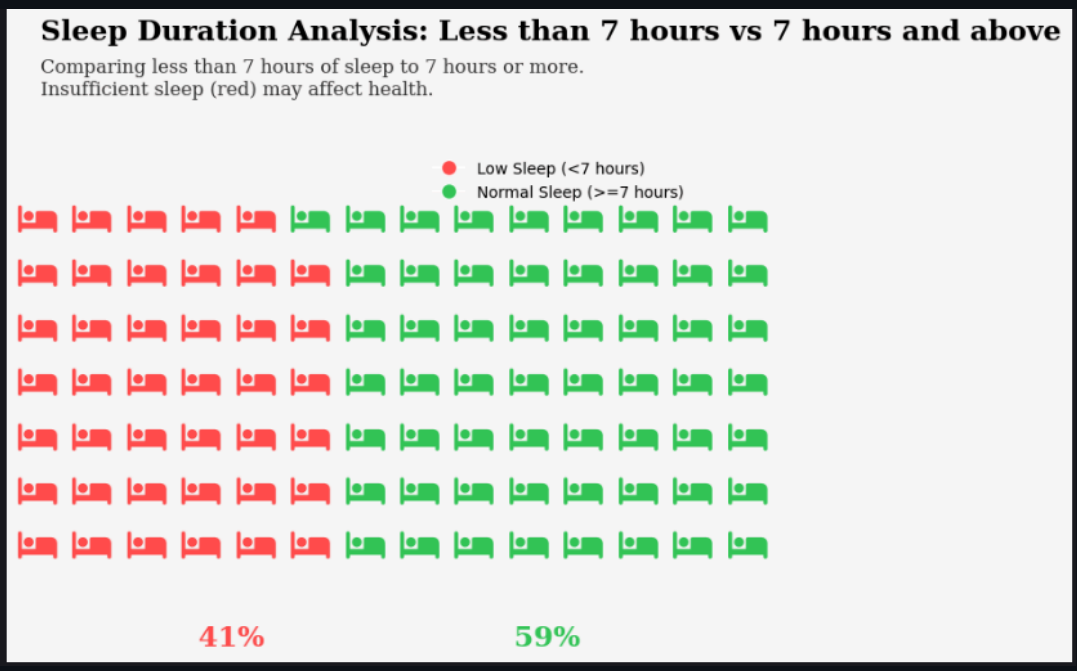
Heart Rate Analysis ❤️‍🩹

This analysis shows that there are **1%** whose heart rate is above **85** ⚠️. The remaining **99%** have a heart rate below **85**, and this group has a healthier heart rate 👨🛐💚

and generally has lower health risks ✅.

2.6 Dataset Analysis

```
low_sleep = df[df['Sleep Duration'] < 7]['Sleep Duration'].count()
normal_sleep = df[df['Sleep Duration'] >= 7]['Sleep Duration'].count()


total_sleep = low_sleep + normal_sleep


low_sleep_percentage = int(round(low_sleep / total_sleep * 100, 0))
normal_sleep_percentage = int(round(normal_sleep / total_sleep * 100, 0))


fig = plt.figure(
    FigureClass=Waffle,
    rows=7,
    columns=14,
    values={"Low Sleep (<7 hours)": low_sleep, "Normal Sleep (>=7 hours)": normal_sleep},
    colors=['#ff4c4c', '#34c759'],  # Kırmızı ve yeşil renkler
    icons='bed',  # Yatak ikonu
    icon_size=20,
    icon_legend=True,
    figsize=(7, 7),
    facecolor='#f6f5f5',
    dpi=100
)


fig.text(0.05, 0.92, 'Sleep Duration Analysis: Less than 7 hours vs 7 hours and above',
         {'font': 'Serif', 'size': 18, 'color': 'black', 'weight': 'bold'})
fig.text(0.05, 0.85, 'Comparing less than 7 hours of sleep to 7 hours or more.\nInsufficient sleep (red) may affect health.',
         {'font': 'Serif', 'size': 12, 'color': 'black'}, alpha=0.8)


fig.text(0.25, 0.15, '{}%'.format(low_sleep_percentage), {'font': 'Serif', 'size': 18, 'weight': 'bold', 'color': '#ff4c4c'})
fig.text(0.65, 0.15, '{}%'.format(normal_sleep_percentage), {'font': 'Serif', 'size': 18, 'weight': 'bold', 'color': '#34c759'})


plt.legend(
    handles=[
        plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='#ff4c4c', markersize=10, label='Low Sleep (<7 hours)'),
        plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='#34c759', markersize=10, label='Normal Sleep (>=7 hours)')
    ],
    loc='upper right', bbox_to_anchor=(0.9, 1.1), frameon=False
)


fig.show()
```



Sleep Duration Analysis 🛌

In this analysis, **41%** sleep **7 hours** or less 😪. It was determined that **59%** sleep **more than 7 hours** 🌙.

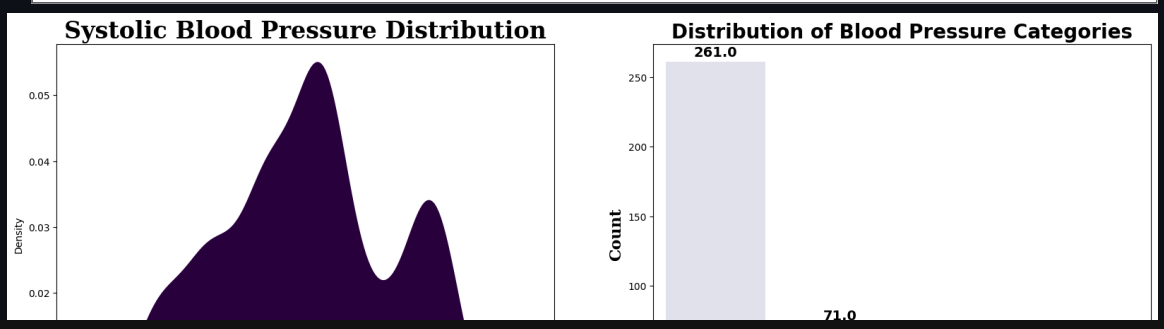
For a quality sleep, we should not forget that in addition to sleeping for a sufficient amount of time, sleep quality is also important. Improving your sleep pattern can help

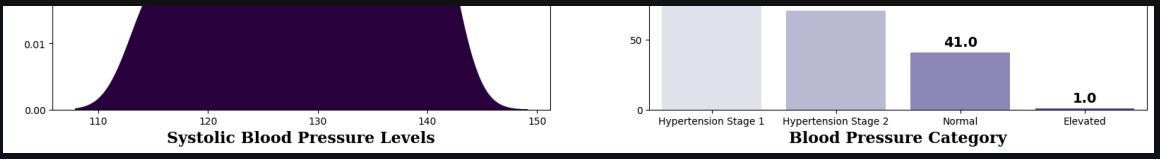
amount of time, sleep quality is also important. Improving your sleep pattern can help you feel more energetic and healthy throughout the day. The ideal sleep duration may vary from person to person, but adults are generally recommended to sleep between **7-9 hours** ⭐.

2.7 Dataset Analysis

In [32]:

```
df1['Systolic'] = df1['Blood Pressure'].str.split('/', expand=True)[0].astype
df1['Diastolic'] = df1['Blood Pressure'].str.split('/', expand=True)[1].astyp

def categorize_blood_pressure(systolic, diastolic):
    if systolic < 90 and diastolic < 60:
        return 'Low'
    elif 90 <= systolic < 120 and diastolic < 80:
        return 'Normal'
    elif 120 <= systolic < 130 and diastolic < 80:
        return 'Elevated'
    elif 130 <= systolic < 140 or 80 <= diastolic < 90:
        return 'Hypertension Stage 1'
    else:
        return 'Hypertension Stage 2'

df1['Blood Pressure Category'] = df1.apply(lambda row: categorize_blood_press

fig, (ax2, ax) = plt.subplots(1, 2, figsize=(20, 7))


sns.kdeplot(data=df1, x='Systolic', ax=ax2, shade=True, color='#2c003e', alph
ax2.set_xlabel('Systolic Blood Pressure Levels', fontdict={'font': 'Serif', '
ax2.set_title('Systolic Blood Pressure Distribution', fontdict={'font': 'Seri

sns.countplot(data=df1, x='Blood Pressure Category', ax=ax, palette='Purples'

ax.set_xlabel('Blood Pressure Category', fontdict={'font': 'Serif', 'color':
ax.set_ylabel('Count', fontdict={'font': 'Serif', 'color': 'black', 'fontsize

for p in ax.patches:
    ax.annotate(f'{p.get_height()}',
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center',
                fontsize=14, weight='bold', color='black',
                xytext=(0, 10), textcoords='offset points')

ax.set_title('Distribution of Blood Pressure Categories', fontsize=20, weight

plt.show()
```





💉 Blood Pressure Distribution 📊

In the analysis of blood pressure distribution, high blood pressure was observed at 3 different levels. The highest rate in the distribution is in the High category, while the Very High and Low categories are less common. According to the data, people with Normal blood pressure levels are quite rare. This situation indicates that the majority are at high blood pressure levels and a situation that needs to be controlled. 🩺⚠️

🔍 Blood Pressure Categories 📉

According to the distribution of blood pressure categories, the numbers of the categories are as follows:

High: 261 people 💪 Very High: 71 people 🚨 Low: 41 people 🤷 Normal: 1 person 😊
When you look at this data, it is understood that High and Very High levels are common, while the Normal category is observed quite rarely. High blood pressure levels stand out as an important risk factor for health. ⚠️

🎉 Thank you to everyone who reviewed this far! 🎉

🙏 **Thank you so much for your support and interest!** 🙏 I am grateful to each and every one of you for taking your valuable time to review this project. I hope the information I provided was useful and everything about the project was as you expected. 🚀

💡 **If you have any questions or feedback, please feel free to let me know.** 💡

🔗 **See you in the next project!** 🔗