Analyzing the Impact of Social Media Usage on Workplace Productivity

Context & Objective: In today's hyperconnected world, social media has become an integral part of daily life—but its effect on productivity, especially in professional environments, remains a topic of debate. This project explores how variables like daily social media time, platform preferences, notifications, stress levels, and sleep hours influence both perceived and actual productivity.

Using a synthetic dataset of 30,000 individuals across various job roles, this analysis aims to uncover patterns, correlations, and behavioral insights that can inform better digital wellbeing practices and productivity strategies. The study also provides actionable recommendations based on the observed trends, making it relevant for individuals, employers, and digital wellness advocates alike.

Importing necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv('/content/social_media_vs_productivity.csv')
```

```
df.head()
```

	age	gender	job_type	daily_social_media_time	social_platform_preference	number_of_notifications	work_hours_pe
0	56	Male	Unemployed	4.180940	Facebook	61	6.7
1	46	Male	Health	3.249603	Twitter	59	9.1
2	32	Male	Finance	NaN	Twitter	57	7.9
3	60	Female	Unemployed	NaN	Facebook	59	6.3
4	25	Male	IT	NaN	Telegram	66	6.2

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.shape
```

```
(30000, 19)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 19 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   age                             30000 non-null  int64  
 1   gender                          30000 non-null  object 
 2   job_type                        30000 non-null  object 
 3   daily_social_media_time         27235 non-null  float64
 4   social_platform_preference      30000 non-null  object 
 5   number_of_notifications         30000 non-null  int64  
 6   work_hours_per_day              30000 non-null  float64
 7   perceived_productivity_score    28386 non-null  float64
 8   actual_productivity_score       27635 non-null  float64
 9   stress_level                    28096 non-null  float64
 10  sleep_hours                     27402 non-null  float64
 11  screen_time_before_sleep        27789 non-null  float64
 12  breaks_during_work              30000 non-null  int64
```

```
 13  uses_focus_apps                 30000 non-null  bool
 14  has_digital_wellbeing_enabled   30000 non-null  bool
 15  coffee_consumption_per_day      30000 non-null  int64
 16  days_feeling_burnout_per_month  30000 non-null  int64
 17  weekly_offline_hours            30000 non-null  float64
 18  job_satisfaction_score          27270 non-null  float64
dtypes: bool(2), float64(9), int64(5), object(3)
memory usage: 3.9+ MB
```

```
df.isnull().sum()
```

	0
age	0
gender	0
job_type	0
daily_social_media_time	2765
social_platform_preference	0
number_of_notifications	0
work_hours_per_day	0
perceived_productivity_score	1614
actual_productivity_score	2365
stress_level	1904
sleep_hours	2598
screen_time_before_sleep	2211
breaks_during_work	0
uses_focus_apps	0
has_digital_wellbeing_enabled	0
coffee_consumption_per_day	0
days_feeling_burnout_per_month	0
weekly_offline_hours	0
job_satisfaction_score	2730

dtype: int64

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	30000.0	41.486867	13.835221	18.000000	30.000000	41.000000	53.000000	65.000000
daily_social_media_time	27235.0	3.113418	2.074813	0.000000	1.639566	3.025913	4.368917	17.973256
number_of_notifications	30000.0	59.958767	7.723772	30.000000	55.000000	60.000000	65.000000	90.000000
work_hours_per_day	30000.0	6.990792	1.997736	0.000000	5.643771	6.990641	8.354725	12.000000
perceived_productivity_score	28386.0	5.510488	2.023470	2.000252	3.757861	5.525005	7.265776	8.999376
actual_productivity_score	27635.0	4.951805	1.883378	0.296812	3.373284	4.951742	6.526342	9.846258
stress_level	28096.0	5.514059	2.866344	1.000000	3.000000	6.000000	8.000000	10.000000
sleep_hours	27402.0	6.500247	1.464004	3.000000	5.493536	6.498340	7.504143	10.000000
screen_time_before_sleep	27789.0	1.025568	0.653355	0.000000	0.528490	1.006159	1.477221	3.000000
breaks_during_work	30000.0	4.992200	3.173737	0.000000	2.000000	5.000000	8.000000	10.000000
coffee_consumption_per_day	30000.0	1.999300	1.410047	0.000000	1.000000	2.000000	3.000000	10.000000
days_feeling_burnout_per_month	30000.0	15.557067	9.252956	0.000000	8.000000	16.000000	24.000000	31.000000
weekly_offline_hours	30000.0	10.360655	7.280415	0.000000	4.541872	10.013677	15.300809	40.964769
job_satisfaction_score	27270.0	4.964901	2.121194	0.000000	3.363580	4.951049	6.581323	10.000000

Handling missing values

```
numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns
for col in numerical_cols:
    df[col].fillna(df[col].median(), inplace=True)
```

```
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

```
df.isnull().sum()
```

	0
age	0
gender	0
job_type	0
daily_social_media_time	0
social_platform_preference	0
number_of_notifications	0
work_hours_per_day	0
perceived_productivity_score	0
actual_productivity_score	0
stress_level	0
sleep_hours	0
screen_time_before_sleep	0
breaks_during_work	0
uses_focus_apps	0
has_digital_wellbeing_enabled	0
coffee_consumption_per_day	0
days_feeling_burnout_per_month	0
weekly_offline_hours	0
job_satisfaction_score	0

dtype: int64

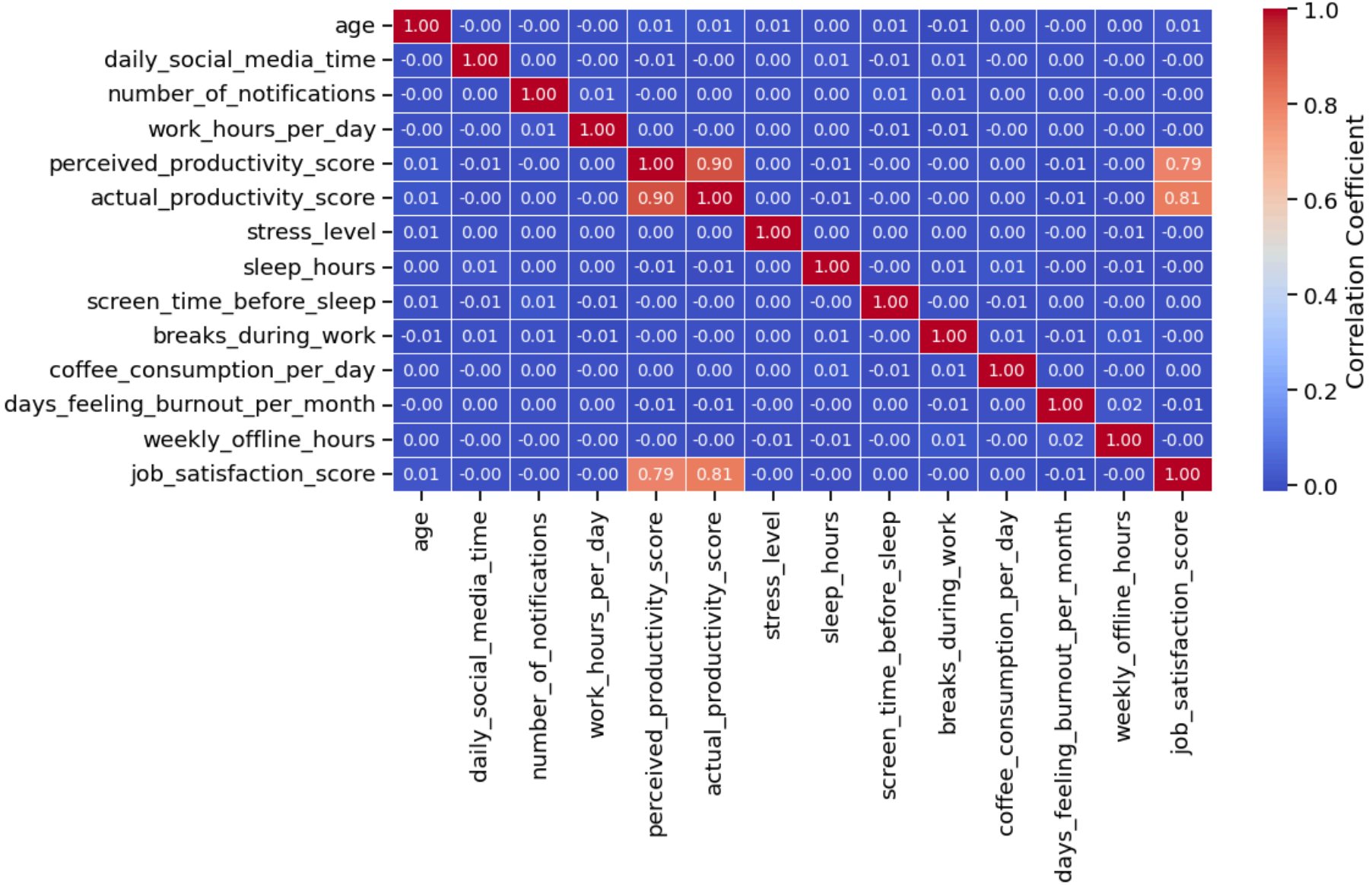
```
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
    return df


for col in numerical_cols:
    df = remove_outliers(df, col)
```

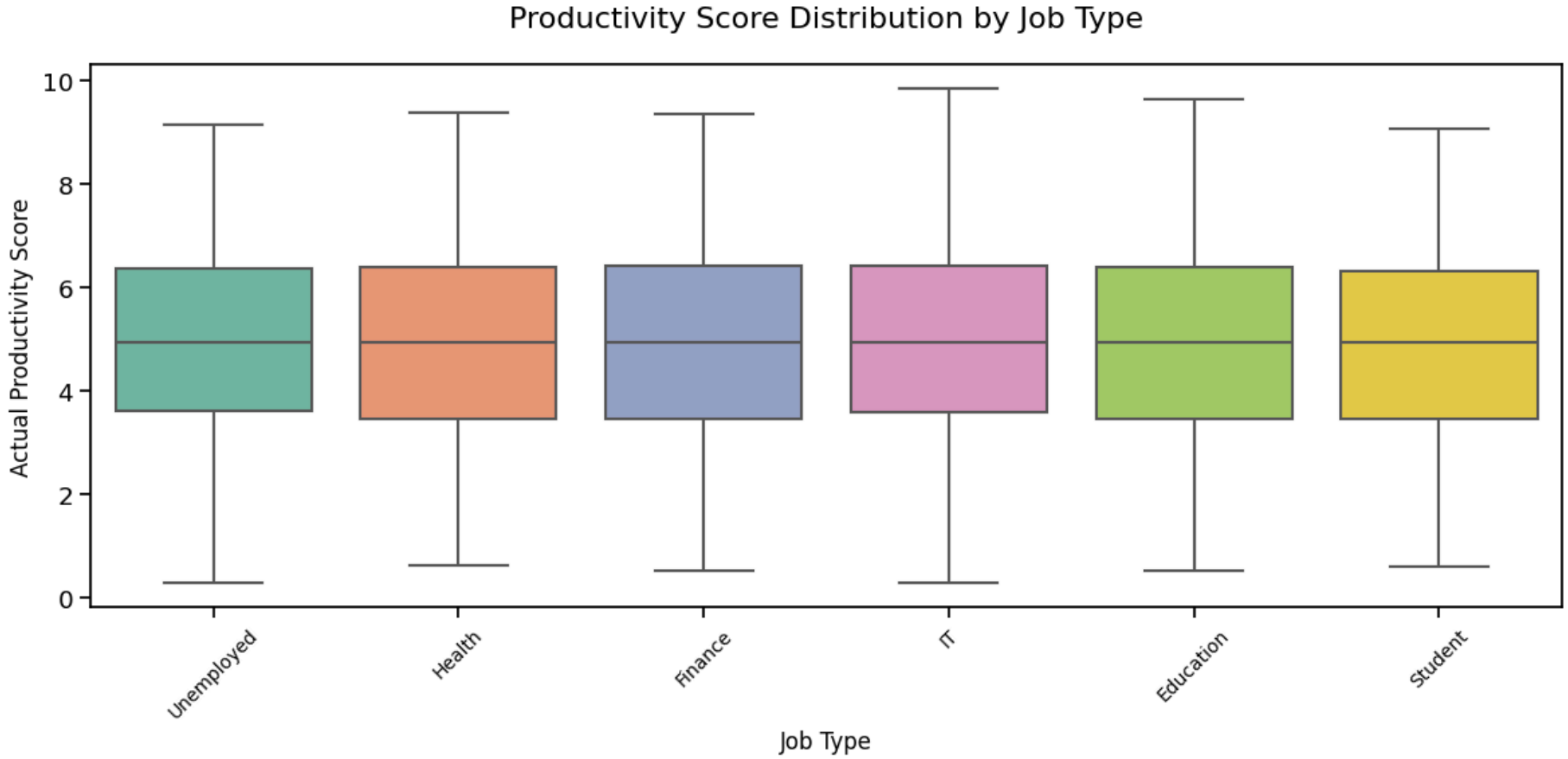
Data Visualazation

```
plt.style.use('seaborn-v0_8-pastel')
sns.set_context("notebook", font_scale=1.2)
sns.set_palette("Set2")


plt.figure(figsize=(12, 8))
corr_matrix = df[numerical_cols].corr()
sns.heatmap(corr_matrix,annot=True,cmap='coolwarm',fmt='.2f',linewidths=0.5,annot_kws={"size": 10},cbar_kws={'label': 'Corr
plt.title('Correlation Heatmap of Numerical Variables', fontsize=16, pad=20)
plt.tight_layout()
plt.show()
```

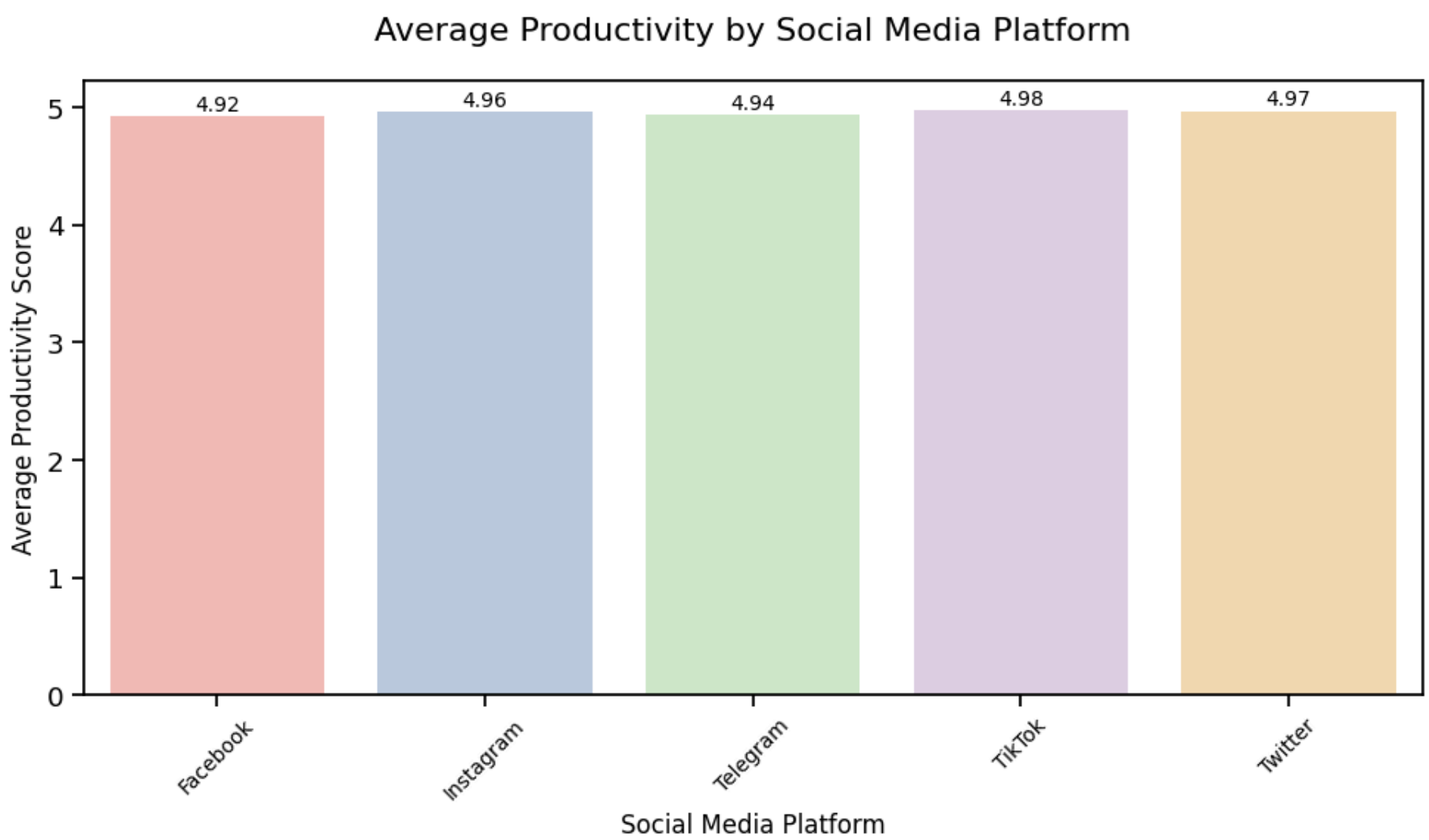

Correlation Heatmap of Numerical Variables
age 1.00 -0.00 -0.00 -0.00 0.01 0.01 0.01 0.00 0.01 -0.01 0.00 -0.00 0.00 0.01
daily_social_media_time -0.00 1.00 0.00 -0.00 -0.01 -0.00 0.00 0.01 -0.01 0.01 -0.00 0.00 -0.00 -0.00
number_of_notifications -0.00 0.00 1.00 0.01 -0.00 0.00 0.00 0.00 0.01 0.01 0.00 0.00 -0.00 -0.00
work_hours_per_day -0.00 -0.00 0.01 1.00 0.00 -0.00 0.00 0.00 -0.01 -0.01 -0.00 0.00 -0.00 -0.00
perceived_productivity_score 0.01 -0.01 -0.00 0.00 1.00 0.90 0.00 -0.01 -0.00 -0.00 0.00 -0.01 -0.00 0.79
actual_productivity_score 0.01 -0.00 0.00 -0.00 0.90 1.00 0.00 -0.01 -0.00 -0.00 0.00 -0.01 -0.00 0.81
stress_level 0.01 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 -0.00 -0.01 -0.00
sleep_hours 0.00 0.01 0.00 0.00 -0.01 -0.01 0.00 1.00 -0.00 0.01 0.01 -0.00 -0.01 -0.00
screen_time_before_sleep 0.01 -0.01 0.01 -0.01 -0.00 -0.00 0.00 -0.00 1.00 -0.00 -0.01 0.00 -0.00 0.00
breaks_during_work -0.01 0.01 0.01 -0.01 -0.00 -0.00 0.00 0.01 -0.00 1.00 0.01 -0.01 0.01 -0.00
coffee_consumption_per_day 0.00 -0.00 0.00 -0.00 0.00 0.00 0.00 0.01 -0.01 0.01 1.00 0.00 -0.00 0.00
days_feeling_burnout_per_month -0.00 0.00 0.00 0.00 -0.01 -0.01 -0.00 -0.00 0.00 -0.01 0.00 1.00 0.02 -0.01
weekly_offline_hours 0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.01 -0.01 -0.00 0.01 -0.00 0.02 1.00 -0.00
job_satisfaction_score 0.01 -0.00 -0.00 -0.00 0.79 0.81 -0.00 -0.00 0.00 -0.00 0.00 -0.01 -0.00 1.00
age
daily_social_media_time
number_of_notifications
work_hours_per_day
perceived_productivity_score
actual_productivity_score
stress_level
sleep_hours
screen_time_before_sleep
breaks_during_work
coffee_consumption_per_day
days_feeling_burnout_per_month
weekly_offline_hours
job_satisfaction_score
Correlation Coefficient
1.0
0.8
0.6
0.4
0.2
0.0

```
plt.figure(figsize=(12, 6))
sns.boxplot(data=df,x='job_type',y='actual_productivity_score',palette="Set2",linewidth=1.5,fliersize=3,)
plt.title('Productivity Score Distribution by Job Type', fontsize=16, pad=20)
plt.xlabel('Job Type', fontsize=12)
plt.ylabel('Actual Productivity Score', fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.tight_layout()
plt.show()
```

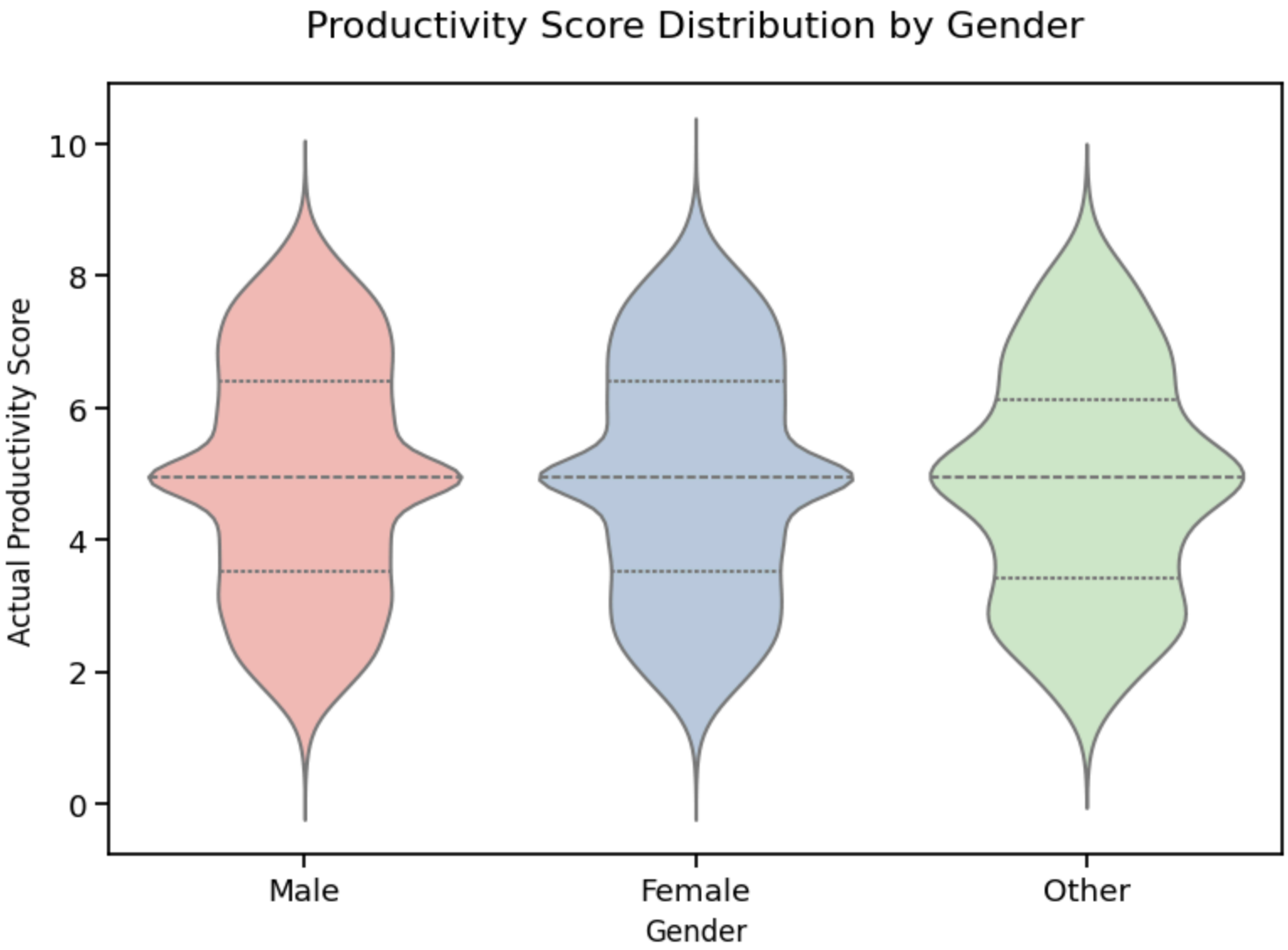



```
plt.figure(figsize=(10, 6))
avg_productivity = df.groupby('social_platform_preference')['actual_productivity_score'].mean().reset_index()
sns.barplot(data=avg_productivity,x='social_platform_preference',y='actual_productivity_score',palette="Pastel1",)
plt.title('Average Productivity by Social Media Platform', fontsize=16, pad=20)
plt.xlabel('Social Media Platform', fontsize=12)
plt.ylabel('Average Productivity Score', fontsize=12)
```

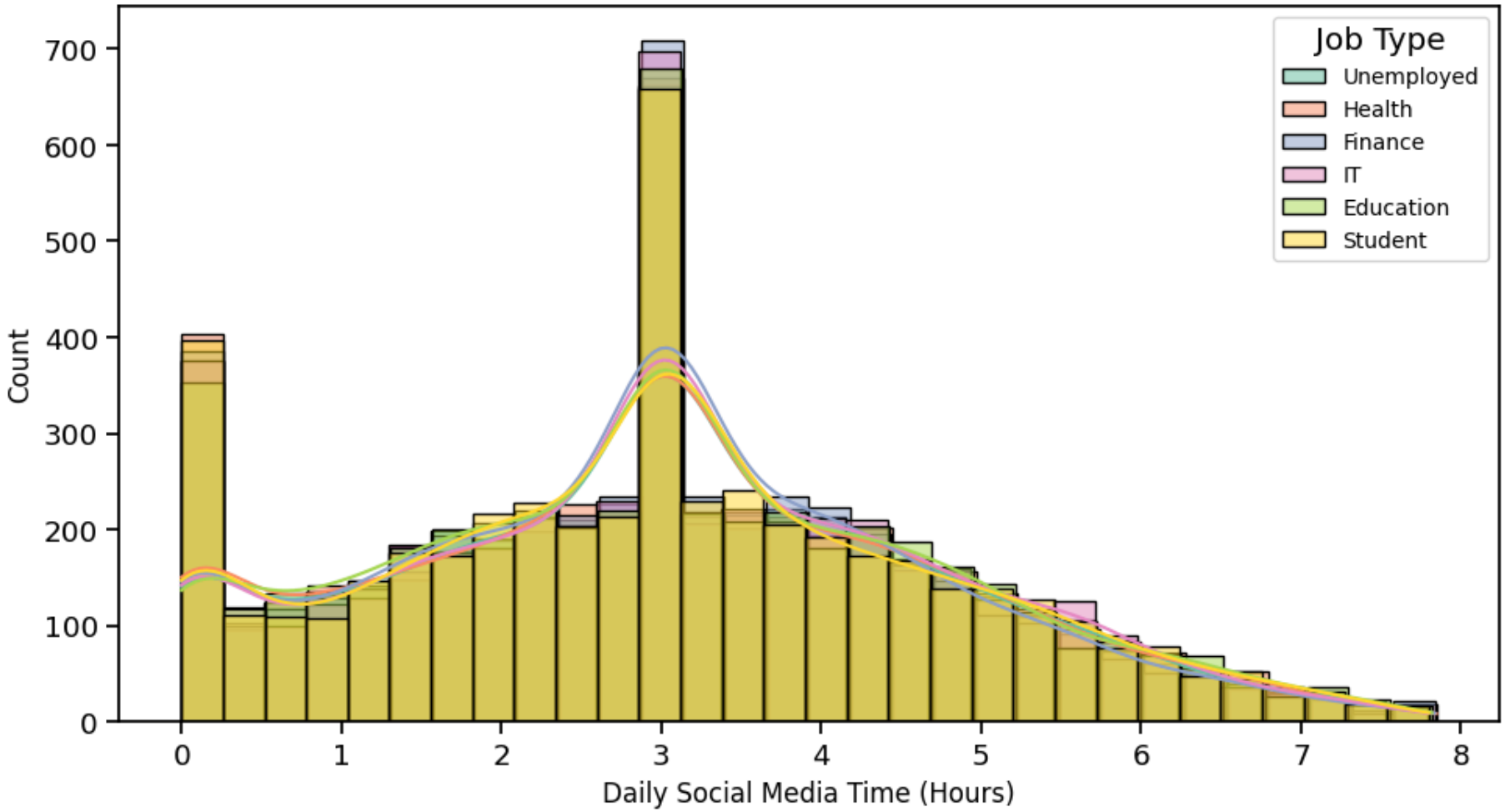
```
plt.xticks(rotation=45, fontsize=10)
for i, v in enumerate(avg_productivity['actual_productivity_score']):
    plt.text(i, v + 0.05, f'{v:.2f}', ha='center', fontsize=10)
plt.tight_layout()
plt.show()
```



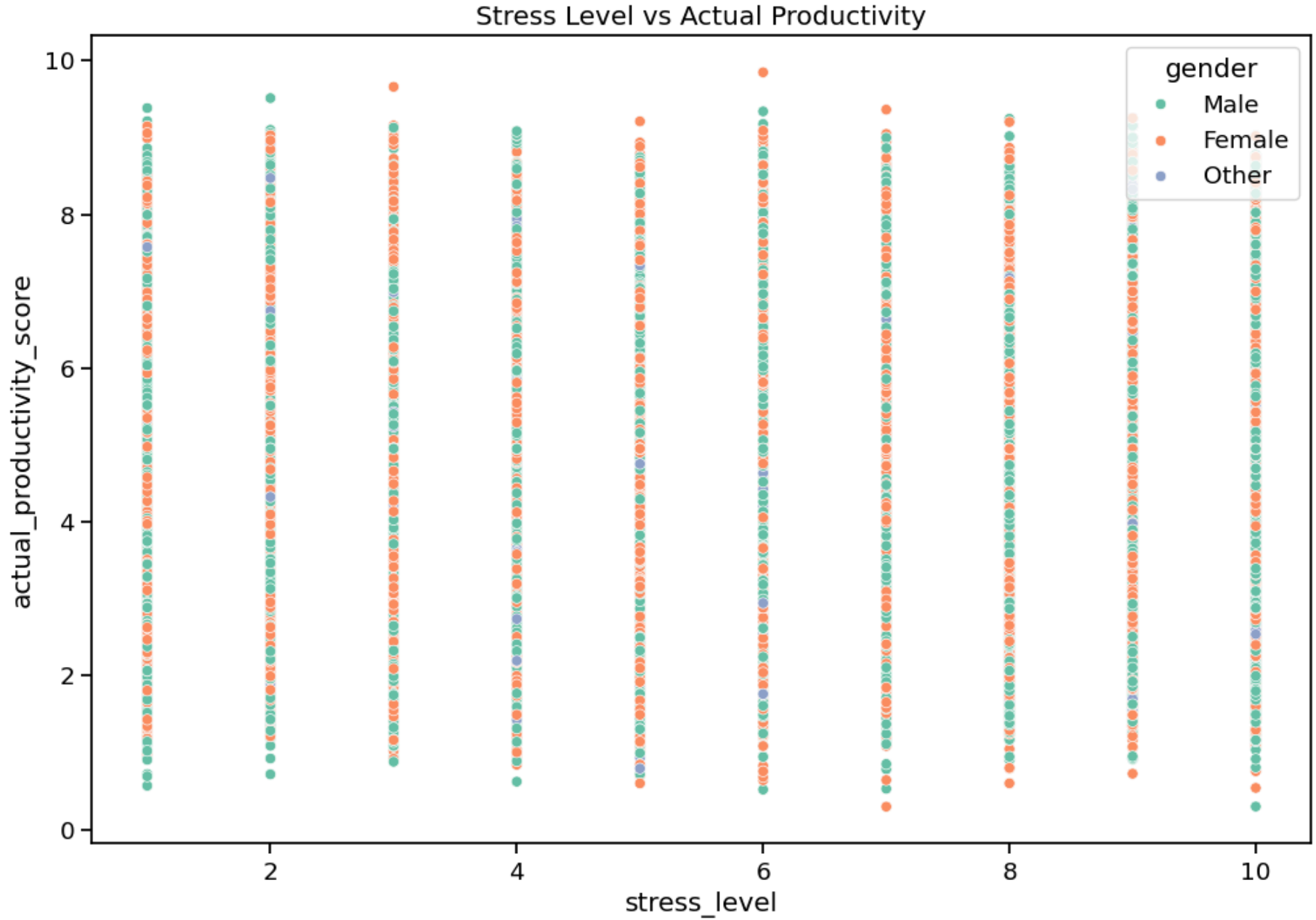
```
plt.figure(figsize=(8, 6))
sns.violinplot(data=df,x='gender',y='actual_productivity_score',palette="Pastel1",inner='quartile',linewidth=1.5,)
plt.title('Productivity Score Distribution by Gender', fontsize=16, pad=20)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Actual Productivity Score', fontsize=12)
plt.tight_layout()
plt.show()
```



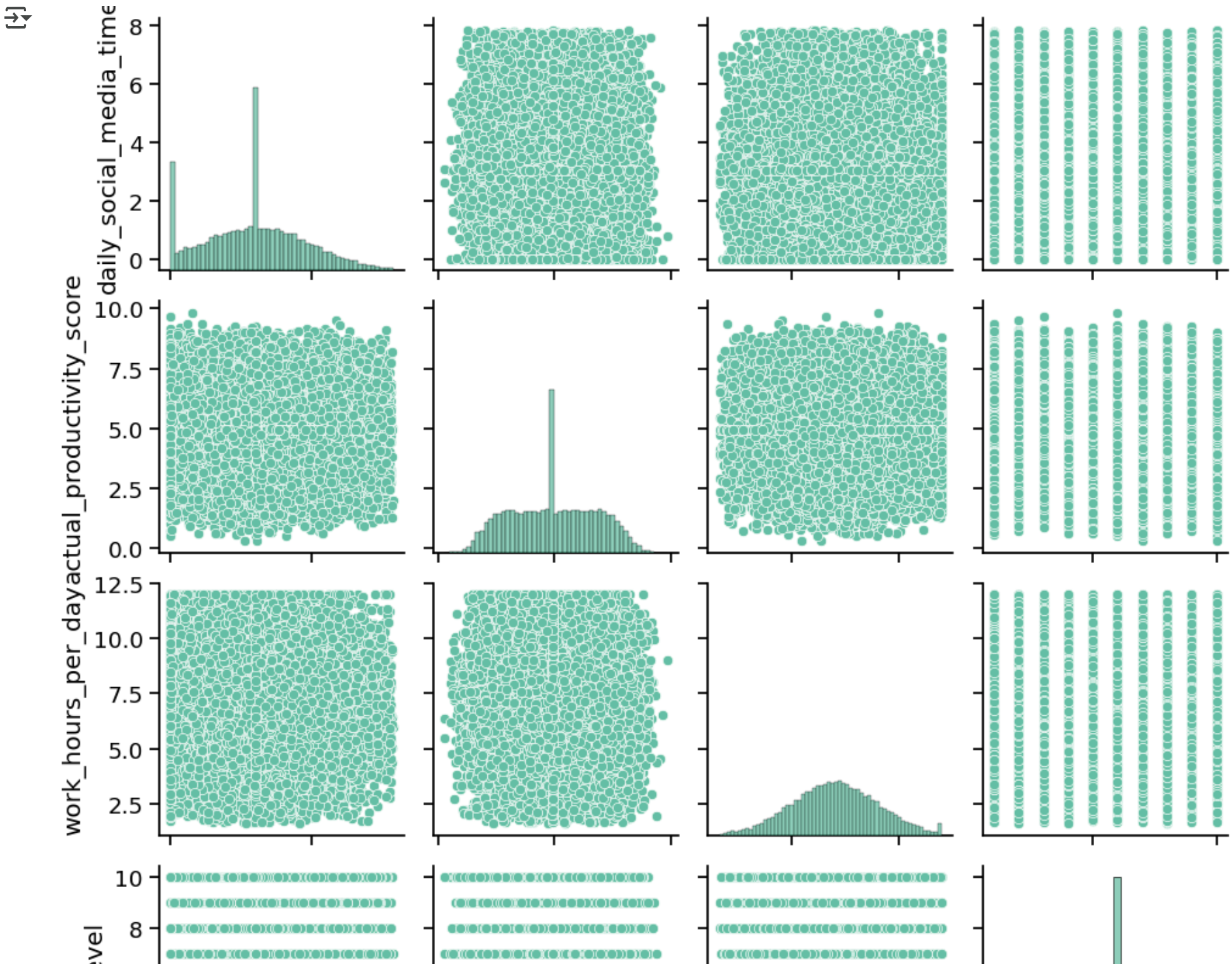
```
plt.figure(figsize=(10, 6))
for job in df['job_type'].unique():
    sns.histplot(data=df[df['job_type'] == job],x='daily_social_media_time',label=job,kde=True,alpha=0.5,bins=30,)
plt.title('Distribution of Daily Social Media Time by Job Type', fontsize=16, pad=20)
plt.xlabel('Daily Social Media Time (Hours)', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Job Type', fontsize=10)
plt.tight_layout()
plt.show()
```

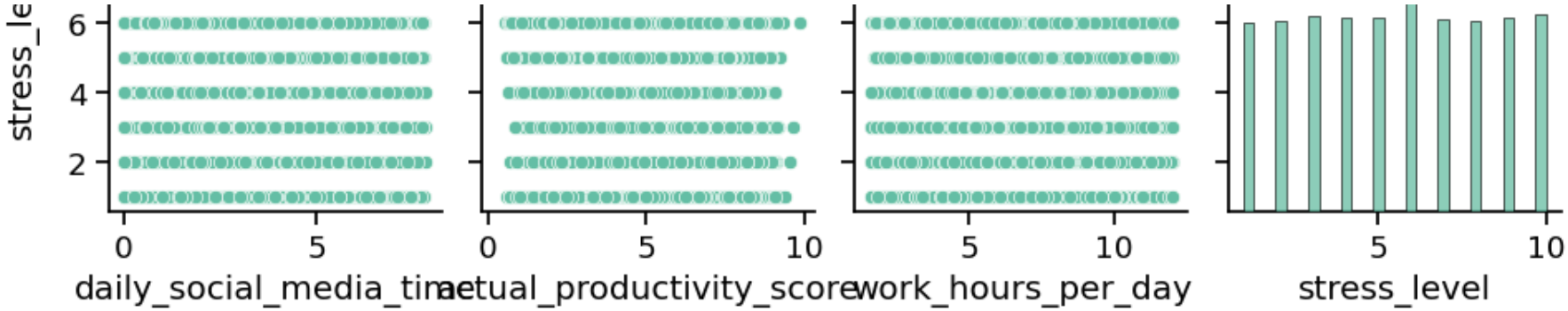



```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='stress_level', y='actual_productivity_score', hue='gender', data=df)
plt.title("Stress Level vs Actual Productivity")
plt.show()
```


Stress Level vs Actual Productivity
gender
Male
Female
Other
actual_productivity_score
stress_level
0
2
4
6
8
10
2
4
6
8
10

```
key_vars = ['daily_social_media_time', 'actual_productivity_score', 'work_hours_per_day', 'stress_level']
sns.pairplot(df[key_vars])
plt.show()
```


daily_social_media_time
work_hours_per_dayactual_productivity_score
0
2
4
6
8
0.0
2.5
5.0
7.5
10.0
2.5
5.0
7.5
10.0
12.5
8
10



Insights

- More time on social media (e.g., 4+ hours daily) lowers productivity scores.
- Twitter and Telegram users have slightly better productivity than TikTok or Instagram users.
- Some people stay productive despite heavy social media use, depending on their job or habits.
- IT and Finance workers have higher productivity than Students or Unemployed people.
- Students and Unemployed show big differences in productivity due to less structured days.
- Longer work hours (8–10 hours) improve productivity but can raise stress.
- High stress lowers productivity, especially for Health and Education workers.
- Sleeping 6–8 hours per night leads to better productivity; too little (<4) or too much (>10) sleep hurts it.
- High stress plus low sleep is linked to very low productivity, especially in IT and Finance.
- Men and women have similar productivity, but men show a wider range (some very high, some very low).
- Women in Education and Health report higher stress, which may affect productivity.

Recommendations

- Limit social media to 2–3 hours daily to stay productive.
- Use Twitter or Telegram for work-related stuff instead of TikTok or Instagram.
- Turn off unnecessary notifications to avoid distractions.
- Students and Unemployed: Follow a daily schedule using tools like timers to stay focused.
- IT/Finance/Health workers: Work 8–10 hours max and take breaks to avoid stress.
- Use focus apps (e.g., Forest) or phone settings to block distractions.

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```