

## Python Script: tips\_charts.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import os

# =====
# Build correct paths
# =====

BASE_DIR = os.path.dirname(os.path.abspath(__file__)) # .../Tips/src

DATA_PATH = os.path.join(BASE_DIR, "..", "data", "tips.csv")
DATA_PATH = os.path.normpath(DATA_PATH)

CHARTS_DIR = os.path.join(BASE_DIR, "..", "charts")
PNG_DIR = os.path.join(CHARTS_DIR, "png")
PDF_DIR = os.path.join(CHARTS_DIR, "pdf")
SVG_DIR = os.path.join(CHARTS_DIR, "svg")

# Create folder structure
for folder in [CHARTS_DIR, PNG_DIR, PDF_DIR, SVG_DIR]:
    if not os.path.exists(folder):
        os.makedirs(folder)

# =====
# Load dataset
# =====
df = pd.read_csv(DATA_PATH)
df["tip_percentage"] = (df["tip"] / df["total_bill"]) * 100

# Modern bright color palette
palette = ["#1f77b4", "#ff7f0e", "#2ca02c"] # Blue, Orange, Teal

def save_chart(name):
    """Save each chart in PNG, PDF, and SVG formats."""
    plt.tight_layout(pad=2.0)

    plt.savefig(os.path.join(PNG_DIR, f"{name}.png"), dpi=300)
    plt.savefig(os.path.join(PDF_DIR, f"{name}.pdf"))
    plt.savefig(os.path.join(SVG_DIR, f"{name}.svg"))

plt.close()

# =====
```

```

# 1. Total Bill vs Tip (Scatter Plot)
# =====
plt.figure(figsize=(12, 7))
plt.scatter(df["total_bill"], df["tip"], color=palette[0])
plt.title("Total Bill vs Tip", fontsize=18, fontweight="bold")
plt.xlabel("Total Bill", fontsize=14)
plt.ylabel("Tip", fontsize=14)
save_chart("01_total_bill_vs_tip")

# =====
# 2. Tip Percentage Distribution (Histogram)
# =====
plt.figure(figsize=(12, 7))
plt.hist(df["tip_percentage"], bins=20, color=palette[1])
plt.title("Tip Percentage Distribution", fontsize=18, fontweight="bold")
plt.xlabel("Tip Percentage", fontsize=14)
plt.ylabel("Frequency", fontsize=14)
save_chart("02_tip_percentage_distribution")

# =====
# 3. Average Tip Percentage by Day (Bar Chart)
# =====
avg_tip_percent_day = df.groupby("day")["tip_percentage"].mean()
plt.figure(figsize=(12, 7))
avg_tip_percent_day.plot(kind="bar", color=palette[0])
plt.title("Average Tip Percentage by Day", fontsize=18, fontweight="bold")
plt.xlabel("Day", fontsize=14)
plt.ylabel("Average Tip Percentage", fontsize=14)
save_chart("03_avg_tip_percentage_by_day")

# =====
# 4. Customer Count per Day (Bar Chart)
# =====
cust_count = df["day"].value_counts().sort_index()
plt.figure(figsize=(12, 7))
cust_count.plot(kind="bar", color=palette[1])
plt.title("Customer Count per Day", fontsize=18, fontweight="bold")
plt.xlabel("Day", fontsize=14)
plt.ylabel("Customer Count", fontsize=14)
save_chart("04_customer_count_per_day")

# =====
# 5. Average Tip by Group Size (Bar Chart)
# =====
avg_tip_size = df.groupby("size")["tip"].mean()
plt.figure(figsize=(12, 7))
avg_tip_size.plot(kind="bar", color=palette[2])
plt.title("Average Tip by Group Size", fontsize=18, fontweight="bold")

```

```

plt.xlabel("Group Size", fontsize=14)
plt.ylabel("Average Tip", fontsize=14)
save_chart("05_avg_tip_by_group_size")

# =====
# 6. Total Revenue per Day (Bar Chart)
# =====
total_rev = df.groupby("day")["total_bill"].sum()
plt.figure(figsize=(12, 7))
total_rev.plot(kind="bar", color=palette[0])
plt.title("Total Revenue per Day", fontsize=18, fontweight="bold")
plt.xlabel("Day", fontsize=14)
plt.ylabel("Total Revenue", fontsize=14)
save_chart("06_total_revenue_per_day")

# =====
# 7. Gender Count by Time (Clustered Bar Chart)
# =====
gender_time = df.groupby(["time", "sex"]).size().unstack()
plt.figure(figsize=(12, 7))
gender_time.plot(kind="bar", color=palette[:2])
plt.title("Gender Count by Time", fontsize=18, fontweight="bold")
plt.xlabel("Time", fontsize=14)
plt.ylabel("Count", fontsize=14)
plt.legend(title="Sex", fontsize=12)
save_chart("07_gender_count_by_time")

# =====
# 8. Gender Count by Day (Clustered Bar Chart)
# =====
gender_day = df.groupby(["day", "sex"]).size().unstack()
plt.figure(figsize=(12, 7))
gender_day.plot(kind="bar", color=palette[:2])
plt.title("Gender Count by Day", fontsize=18, fontweight="bold")
plt.xlabel("Day", fontsize=14)
plt.ylabel("Count", fontsize=14)
plt.legend(title="Sex", fontsize=12)
save_chart("08_gender_count_by_day")

# =====
# 9. Tip Percentage by Smoker Status
# =====
smoker_avg = df.groupby("smoker")["tip_percentage"].mean()
plt.figure(figsize=(12, 7))
smoker_avg.plot(kind="bar", color=palette[1])
plt.title("Tip Percentage by Smoker Status", fontsize=18, fontweight="bold")
plt.xlabel("Smoker Status", fontsize=14)
plt.ylabel("Average Tip Percentage", fontsize=14)

```

```
save_chart("09_tip_percentage_by_smoker_status")

# =====
# 10. Correlation Heatmap
# =====
plt.figure(figsize=(12, 7))
sns.heatmap(
df[["total_bill", "tip", "tip_percentage", "size"]].corr(),
annot=True,
cmap="coolwarm",
fmt=".2f"
)
plt.title("Correlation Heatmap", fontsize=18, fontweight="bold")
save_chart("10_correlation_heatmap")

print("All charts generated successfully!")
```