```
# Import libraries
import matplotlib.pyplot as plt
import numpy as np

# ------------------------------
# Background:
# This grouped bar chart compares average Likert-scale scores for five
well-being variables under two environmental conditions:
# high heat and fair weather. These variables—fatigue, irritability,
social withdrawal, motivation, and recovery time—were
# assessed across two survey periods, and scores reflect participants'
self-assessments on a 1–5 scale.
#
# Survey Questions Mapped:
# - Fatigue: "How physically fatigued do you feel after work on a hot
day?"
# - Irritability: "How emotionally irritable or reactive do you feel
after work?"
# - Social Withdrawal: "To what extent do you avoid communication or
engagement after work?"
# - Motivation: "How motivated do you feel to be productive or social
after work?"
# - Recovery Time: "How long does it take to recover after a typical
workday?"
#
# Purpose:
# The chart provides a holistic view of the psychological and behavioral
toll of heat stress,
# and illustrates improvement across all domains under fair weather.
# Color-coded bars reinforce visual comparison between the two
conditions.
# ------------------------------

# Data
categories = ['Fatigue', 'Irritability', 'Social Withdrawal',
'Motivation', 'Recovery Time']
high_heat_scores = [4.3, 4.1, 3.9, 3.2, 4.0]
fair_weather_scores = [2.1, 2.0, 1.9, 3.5, 2.2]

x = np.arange(len(categories))
width = 0.35

# Plot
fig, ax = plt.subplots(figsize=(12, 6))
bars1 = ax.bar(x - width/2, high_heat_scores, width, label='High Heat
(Darker)', color='#3B0F70', edgecolor='black')
bars2 = ax.bar(x + width/2, fair_weather_scores, width, label='Fair
Weather (Lighter)', color='#B7E1B0', edgecolor='black')

# Titles and labels
ax.set_title('Impact of Environmental Conditions on Worker Well-Being',
fontsize=14, fontweight='bold')
ax.set_ylabel('Mean Score (1-5 Likert Scale)', fontsize=12)
```

```
ax.set_xticks(x)
ax.set_xticklabels(categories)
ax.set_ylim(0, 5)
ax.legend(title='Condition')

# Annotate values
for bar in bars1 + bars2:
    height = bar.get_height()
    ax.annotate(f'{height:.1f}',
                xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),
                textcoords="offset points",
                ha='center', va='bottom', fontsize=10)

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