measunc

February 8, 2026

1 Measurement and Uncertainty

1-8 and 11-13

Table 1: Probabilites for two fair dice

Sum	Unique Combinations	Number	Probabilty
1	none	none	0:36
2	(1, 1)	1	1:36
3	(2, 1), (1, 2)	2	2:36
4	(3, 1), (2, 2), (1, 3)	3	3:36
5	(4, 1), (3, 2), (2, 3), (1, 4)	4	4:36
6	(5, 1), (4, 2), (3, 3), (2, 4), (1, 5)	5	5:36
7	(6, 1), (5, 2), (4, 3), (3, 4), (2, 5), (1, 6)	6	6:36
8	(6, 2), (5, 3), (4, 4), (3, 5), (2, 6)	5	5:36
9	(6, 3), (5, 4), (4, 5), (3, 6)	4	4:36
10	(6, 4), (5, 5), (4, 6)	3	3:36
11	(5, 6), (6, 5)	2	2:36
12	(6, 6)	1	1:36
13	none	0	0:36

Scatter Plots

[96]:
```
#Import libraries
import numpy as np
import matplotlib.pyplot as plt
```

[97]:
```
tt = np.arange(10, 26, 1)
vv = np.array([5, 6, 1, 3, 0, 0.1,0, -2, -2, -2, -2,-9,-15,-5,-7,-7])
sv = np.array([9,5, 10,2,3,0.2,4,4,7,13,17,6,1,21,22,25])
```

[98]:
```
plt.figure(figsize=(6,4))

plt.errorbar(
    tt,
    vv,
    yerr=sv,
```

```
    fmt='o',
    color='black',
    ecolor='black',
    elinewidth=1
)

plt.title("v versus t")
plt.xlabel("time [s]")
plt.ylabel("velocity [m/s]")
txt="Figure 1: Scatterplot of velocity versus time. Error bars indicate 1␣
 ↪standard uncertainty in the measurement."
plt.figtext(0.5, -0.1, txt, wrap=True, horizontalalignment='center',␣
 ↪fontsize=12)
plt.grid()
plt.show()
```

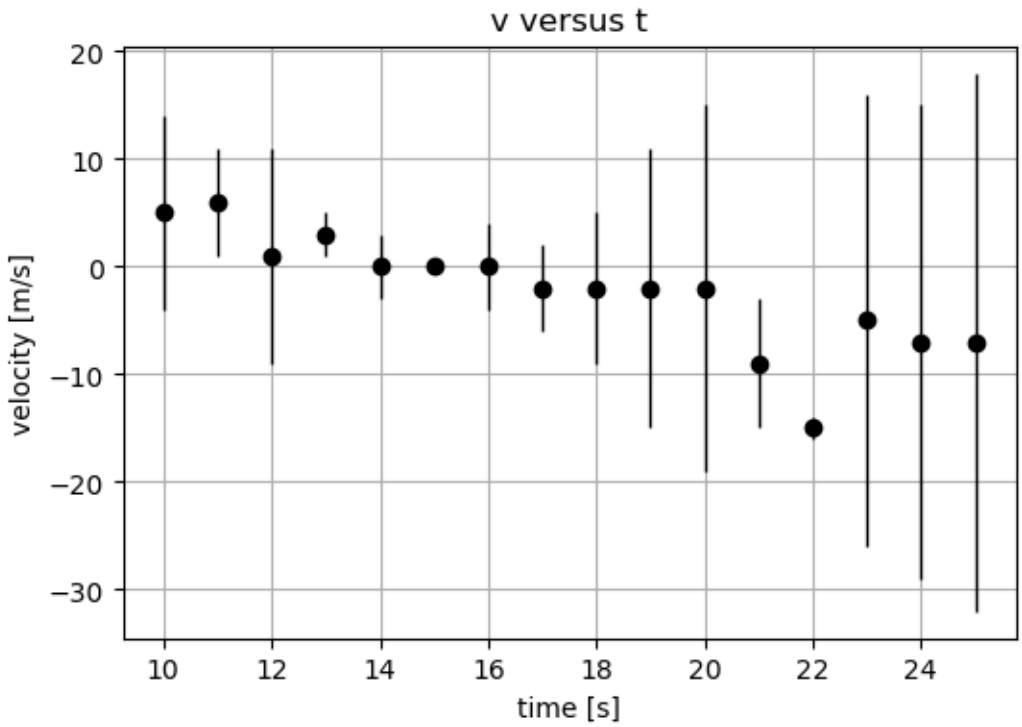


Figure 1: Scatterplot of velocity versus time. Error bars indicate 1 standard uncertainty in the measurement.

[99]:
```
sums = np.arange(1, 14, 1)
prob = np.array([0,1/36,2/36,3/36,4/36,5/36,6/36,5/36,4/36,3/36,2/36,1/36,0])
plt.plot(sums, prob, marker='o', linestyle='-')
```

```
plt.axvline(x=7, ymin=0.05, ymax=0.95, color='black', linestyle='--',␣
 ↪label='a=6')
plt.annotate('a = 6',
             xy=(7, 0.0),
             xytext=(13, 0.0),
             ha='center',
             arrowprops=dict(arrowstyle='<->', linewidth=1.5))
txt="Figure 2: Probability distribution for two fair dice. Note the symmetry of␣
 ↪the distribution (an isosceles triangle). The connected dots define a␣
 ↪triangular PDF for the roll of two fair dice. The symmetry implies that the␣
 ↪center of the distribution is also the mean. Also shown is the half-width, a.
 ↪"
plt.figtext(0.5, -0.1, txt, wrap=True, horizontalalignment='center',␣
 ↪fontsize=12)
plt.show()
```

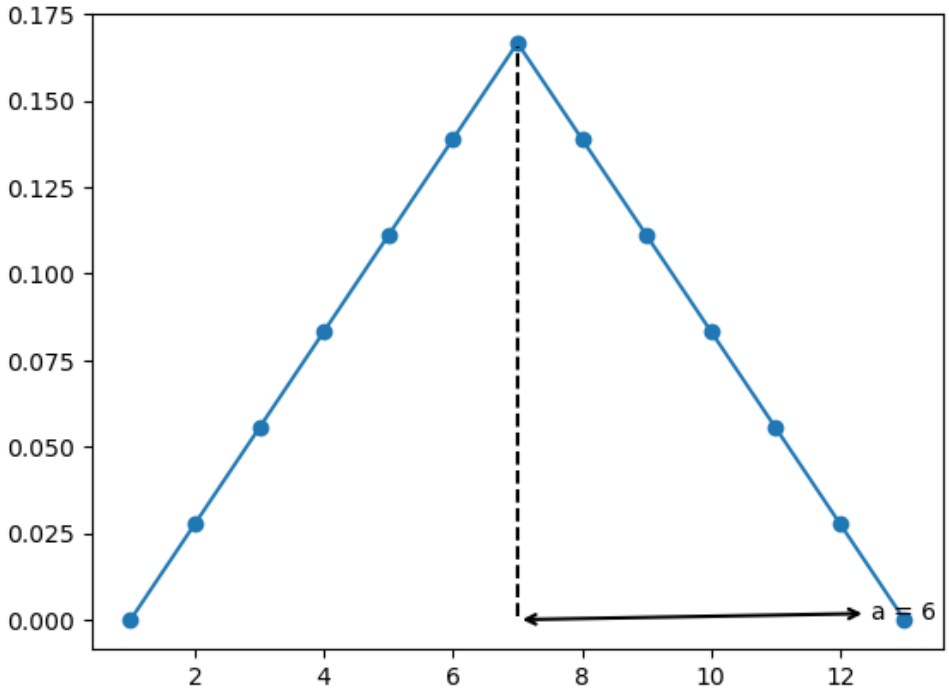
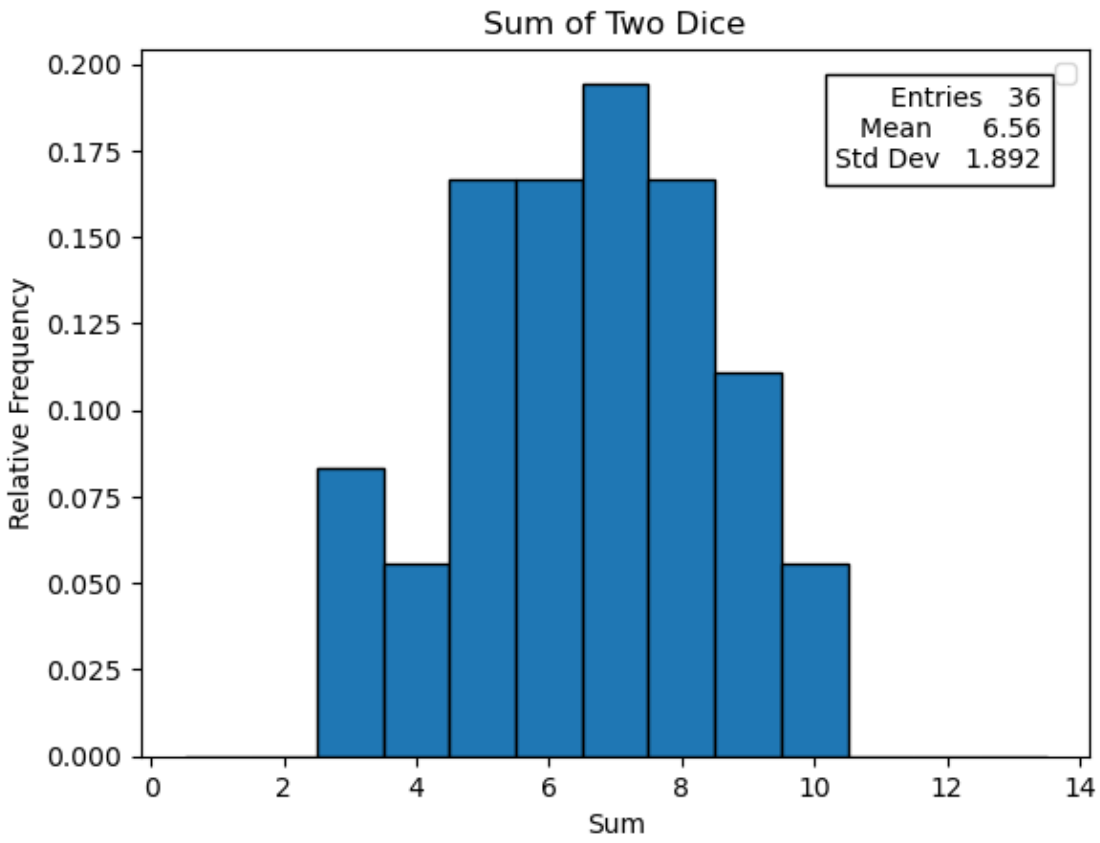


Figure 2: Probability distribution for two fair dice. Note the symmetry of the distribution (an isosceles triangle). The connected dots define a triangular PDF for the roll of two fair dice. The symmetry implies that the center of the distribution is also the mean. Also shown is the half-width, a.

[100]:
```
import random

sums = np.arange(1,15, 1)
size = 36
```

```
entries = []
for i in range(size):
    x = random.randrange(1, 7)
    y = random.randrange(1, 7)
    s = x + y
    entries.append(s)

relative_freq = [np.count_nonzero(entries == i)/size for i in range(1, 15)]
plt.hist(entries, bins=np.arange(0.5, 14.5, 1), density=True, edgecolor='black')

mean = np.mean(entries)
std = np.std(entries, ddof=0)   # population std


plt.title("Sum of Two Dice")
plt.xlabel("Sum")
plt.ylabel("Relative Frequency")
plt.legend()
info_text = (
    f"Entries   {size}\n"
    f"Mean      {mean:.2f}\n"
    f"Std Dev   {std:.3f}"
)

plt.text(0.95, 0.95, info_text,
         transform=plt.gca().transAxes,
         fontsize=10,
         verticalalignment='top',
         horizontalalignment='right',
         bbox=dict(facecolor='white', edgecolor='black'))


plt.show()
```

```
C:\Users\haru\AppData\Local\Temp\ipykernel_28240\1666612121.py:22: UserWarning:
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
  plt.legend()
```



11. Confidence Levels

[101]:
```
from scipy.stats import norm

# (a) Confidence level for ±2
conf_2 = norm.cdf(2) - norm.cdf(-2)

# (b) Confidence level for ±3
conf_3 = norm.cdf(3) - norm.cdf(-3)

# (c) Coverage factor n for 90% confidence
# 90% symmetric → upper tail at 0.95
n_90 = norm.ppf(0.95)

print("Confidence for ±2 :", conf_2)
print("Confidence for ±3 :", conf_3)
print("Coverage factor for 90%:", n_90)
```

```
Confidence for ±2 : 0.9544997361036416
Confidence for ±3 : 0.9973002039367398
Coverage factor for 90%: 1.6448536269514722
```