

Pairs Project – Finding the Truth in TEQS Code

Bias-correction

```
import numpy as np

# Attendance categories and observed counts
attendance = np.array([0.25, 0.5, 0.75, 1.0])
observed = np.array([7, 19, 42, 29])

# Correct for selection bias
relative_true = observed / attendance
true_counts = 150 * relative_true / relative_true.sum()

# Mean and standard deviation of attendance
mean_attendance = np.sum(attendance * true_counts)
variance = np.sum(attendance * (1 - attendance) * true_counts)
std_attendance = np.sqrt(variance)

print("Estimated true counts:", true_counts)
print("Expected attendance:", mean_attendance)
print("Variance:", variance)
print("Standard deviation:", std_attendance)
```

Graph 1

```
import numpy as np
import matplotlib.pyplot as plt

attendance = np.array([0.25, 0.5, 0.75, 1.0])
observed = np.array([7, 19, 42, 29])

# Bias correction
relative_true = observed / attendance
true_counts = 150 * relative_true / relative_true.sum()

attendance_labels = ['25%', '50%', '75%', '100%']
x = np.arange(len(attendance_labels))
width = 0.35

plt.bar(x - width/2, observed, width, label='Observed')
plt.bar(x + width/2, true_counts, width, label='Estimated true')

plt.xticks(x, attendance_labels, rotation=45, fontsize=15)
plt.xlabel('Percentage of Lectures Attended', fontsize=15)
plt.ylabel('Number of students', fontsize=15)
plt.yticks(fontsize=15)

plt.legend(fontsize=15)
plt.title('Attendance and Estimated True Counts', fontsize=20)
```

```
plt.tight_layout()
```

```
plt.show()
```

Graph 2

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
np.random.seed(57)
```

```
attendance = np.array([0.25, 0.5, 0.75, 1.0])
```

```
observed = np.array([7, 19, 42, 29])
```

```
relative_true = observed / attendance
```

```
true_counts = 150 * relative_true / relative_true.sum()
```

```
n_trials = 50000
```

```
results = []
```

```
for _ in range(n_trials):
```

```
    total = 0
```

```
    for a, n in zip(attendance, true_counts.astype(int)):
```

```
        total += np.sum(np.random.rand(n) < a)
```

```
    results.append(total)
```

```
mean_attendance = np.mean(results)
std_attendance = np.std(results)
print("Estimated mean attendance:", mean_attendance)
print("Estimated standard deviation:", std_attendance)

plt.hist(results, bins=40, density=True)
plt.axvline(97, linestyle='--', label='Observed (97)', color = 'r')
plt.xlabel('Number of Students Attending', fontsize=15)
plt.ylabel('Probability density (PDF)', fontsize=15)
plt.title('Monte Carlo Simulation of Attendance', fontsize=20)
plt.legend(fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

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