ESA - Statistical Methods for Decision Making - Answer Sheet

Section A

1. a) Mean: 20.0, Standard Deviation: 7.91

```
import numpy as np

data = np.array([10, 15, 20, 25, 30])
mean = np.mean(data)
std_dev = np.std(data, ddof=1)

mean, std_dev
```

b) Ordinal data: Categorizes data with a meaningful order but the intervals between the categories are not necessarily equal (e.g., rankings).

Interval data: Similar to ordinal data but with equal intervals between values, and no true zero point (e.g., temperature in Celsius).

c) Probability of drawing a red card: 0.5

```
# No code needed; basic probability calculation
prob_red_card = 26 / 52
prob_red_card
```

d) Probability of answering exactly 3 out of 5 questions correctly: 0.088

```
from scipy.stats import binom

prob_correct_3_out_of_5 = binom.pmf(3, 5, 0.25)
prob_correct_3_out_of_5
```

e) If p-value < ?, reject the null hypothesis.

2. a) Type I Error (False Positive)

b) Minimum sample size recommended: 30

c) Suitable statistical test: Paired t-test (Dependent samples t-test)

d) P(A) = 0.02, P(A') = 0.98, P(B|A) = 0.98, P(B|A') = 0.01

```
P_A = 0.02
P_A_prime = 1 - P_A
P_B_given_A = 0.98
P_B_given_A_prime = 0.01
```

ESA - Statistical Methods for Decision Making - Answer Sheet

```
P_A, P_A_prime, P_B_given_A, P_B_given_A_prime
```

e) Mean increases by 5 years (new mean = 25 years), Standard Deviation remains the same (3 years)

Section B

3. t-statistic: -0.480, p-value: 0.642. No significant difference between the average weight of the product samples and the target weight of 20 units.

```
import numpy as np
from scipy.stats import ttest_1samp

samples = np.array([19, 21, 20, 18, 22, 19, 20, 21, 18, 20])
target_weight = 20

t_stat, p_value = ttest_1samp(samples, target_weight)
t_stat, p_value
```

4. t-statistic: 0.396, p-value: 0.699. No significant difference in the average daily sales between Store A and Store B.

```
Store_A_sales = np.array([500, 520, 480, 510, 490, 530, 550])

Store_B_sales = np.array([480, 510, 520, 525, 495, 505, 515])


t_stat, p_value = ttest_ind(Store_A_sales, Store_B_sales)

t_stat, p_value
```

5. f-statistic: 16.337, p-value: 0.00038. Significant difference in the average test scores among the three different schools.

```
school1_scores = np.array([85, 87, 90, 95, 88])

school2_scores = np.array([78, 80, 82, 85, 83])

school3_scores = np.array([92, 94, 89, 91, 93])


f_stat, p_value = stats.f_oneway(school1_scores, school2_scores, school3_scores)

f_stat, p_value
```

6. Pearson correlation coefficient: 0.959, p-value: 0.000011. Strong positive correlation between the number of study hours and the test scores of students.

```
study_hours = np.array([3, 4, 6, 5, 7, 8, 5, 9, 10, 7])
```

```
test_scores = np.array([60, 65, 70, 75, 80, 85, 75, 90, 95, 85])


correlation, p_value = stats.pearsonr(study_hours, test_scores)

correlation, p_value
```

7. Probability they agree: 0.6, Probability they contradict: 0.4

```
p_A_truth = 2/3

p_B_truth = 4/5


p_agree = (p_A_truth * p_B_truth) + ((1 - p_A_truth) * (1 - p_B_truth))

p_contradict = (p_A_truth * (1 - p_B_truth)) + ((1 - p_A_truth) * p_B_truth)

p_agree, p_contradict
```

Section C

8. a) Chi-square statistic: 12.754, p-value: 0.047. Significant difference in the preferences for different ice cream flavors among people of different age groups.

```
data = np.array([[45, 35, 20, 25],

                 [30, 40, 25, 25],

                 [20, 25, 30, 25]])
```

```
chi2_stat, p_value, dof, expected = stats.chi2_contingency(data)

chi2_stat, p_value, dof, expected
```

8. b) Probability of receiving exactly 8 calls in a given hour: 0.113

```
lambda_ = 10

k = 8


poisson_prob = stats.poisson.pmf(k, lambda_)

poisson_prob
```

8. c) Required sample size: 1068

```
confidence_level = 0.95

z_score = stats.norm.ppf(0.5 + confidence_level / 2)

margin_of_error = 0.03

p = 0.5


n = (z_score**2 * p * (1 - p)) / margin_of_error**2

sample_size = np.ceil(n)
```

```
z_score, sample_size
```

9. a) ANOVA statistic: 1.393, p-value: 0.310. No significant difference in the students' final course grades based on ethnicity.

```
data = {
    'ethnicity': ['Group A', 'Group A', 'Group B', 'Group B', 'Group C', 'Group C', 'Group A', 'Group B', 'Group C', 'Group A'],
    'total_score': [78, 85, 80, 70, 65, 90, 88, 75, 85, 92],
    'institute': ['Institute 1', 'Institute 2', 'Institute 1', 'Institute 2', 'Institute 1', 'Institute 2', 'Institute 1', 'Institute 2', 'Institute 1', 'Institute 2']
}

df = pd.DataFrame(data)

anova_result = stats.f_oneway(
    df[df['ethnicity'] == 'Group A']['total_score'],
    df[df['ethnicity'] == 'Group B']['total_score'],
    df[df['ethnicity'] == 'Group C']['total_score']
)
anova_result
```

ESA - Statistical Methods for Decision Making - Answer Sheet

9. b) t-statistic: -0.549, p-value: 0.598. No significant impact of the type of institute on the final scores.

```
t_stat, p_value = stats.ttest_ind(
    df[df['institute'] == 'Institute 1']['total_score'],
    df[df['institute'] == 'Institute 2']['total_score']
)
t_stat, p_value
```