assignment2

October 31, 2024

[ ]:
```python
import numpy as np
from scipy import stats

# 1. Test of Significance for Assembly Time

# Given data
population_mean = 35
sample_mean = 33
population_std_dev = 5
sample_size = 40

# Calculate the test statistic (z-score)
z = (sample_mean - population_mean) / (population_std_dev / np.
 ↪sqrt(sample_size))
print(f"Test Statistic (z): {z}")

# Find the critical value for a one-tailed test at alpha = 0.05
alpha = 0.05
critical_value = stats.norm.ppf(1 - alpha)
print(f"Critical Value: {critical_value}")

# Calculate the p-value
p_value = stats.norm.cdf(z)
print(f"P-value: {p_value}")

# Conclusion
if p_value < alpha:
    print("Reject the null hypothesis: There is significant evidence that the␣
 ↪average assembly time has decreased.")
else:
    print("Fail to reject the null hypothesis: No significant evidence of a␣
 ↪decrease in assembly time.")
```

```
Test Statistic (z): -2.5298221281347035
Critical Value: 1.6448536269514722
P-value: 0.005706018193000824
Reject the null hypothesis: There is significant evidence that the average
assembly time has decreased.
```

[ ]:
```python
# 2. Test of Significance for Graduate Students Study Hours
# Given data
population_mean_study = 25
sample_mean_study = 27
sample_std_dev = 4.5
sample_size_study = 15

# Calculate the test statistic (t-score)
t = (sample_mean_study - population_mean_study) / (sample_std_dev / np.
↪sqrt(sample_size_study))
print(f"Test Statistic (t): {t}")

# Find the critical value for a one-tailed test at alpha = 0.05 with df = n - 1
df = sample_size_study - 1
critical_value_study = stats.t.ppf(1 - alpha, df)
print(f"Critical Value (t): {critical_value_study}")

# Calculate the p-value
p_value_study = 1 - stats.t.cdf(t, df)
print(f"P-value: {p_value_study}")

# Conclusion
if p_value_study < alpha:
    print("Reject the null hypothesis: There is significant evidence that␣
↪graduate students study more than 25 hours per week.")
else:
    print("Fail to reject the null hypothesis: No significant evidence that␣
↪graduate students study more than 25 hours.")
```

```
Test Statistic (t): 1.7213259316477407
Critical Value (t): 1.7613101357748562
P-value: 0.05360191367469436
Fail to reject the null hypothesis: No significant evidence that graduate
students study more than 25 hours.
```

[ ]:
```python
import pandas as pd
import statsmodels.api as sm


# 3. Simple Linear Regression Analysis

# Corrected data input
data = {
    "Hours of Study": [5, 5, 7, 3, 0, 5, 2, 7, 4, 2, 1, 6, 8, 5, 7,
                       8, 8, 8, 7, 8, 8, 0, 2, 0, 2, 6, 7, 4, 4, 8],
    "Score": [52.1221, 52.1221, 72.1221, 32.1221, 2.122104, 52.1221,
              22.1221, 72.1221, 42.1221, 22.1221, 12.1221, 62.1221,
```

```
                 82.1221, 52.1221, 72.1221, 82.1221, 82.1221, 72.1221,
                 72.1221, 82.1221, 82.1221, 2.122104, 2.122104, 62.1221,
                 72.1221, 42.1221, 42.1221, 62.1221, 82.1221, 82.1221]
}

# Check lengths of data
print(f"Length of Hours of Study: {len(data['Hours of Study'])}")
print(f"Length of Score: {len(data['Score'])}")

# Create DataFrame
df = pd.DataFrame(data)

# Fit the model
X = sm.add_constant(df["Hours of Study"])  # Add a constant for the intercept
y = df["Score"]
model = sm.OLS(y, X).fit()

# Summary
print(model.summary())
```

```
Length of Hours of Study: 30
Length of Score: 30
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Score   R-squared:                       0.598
Model:                            OLS   Adj. R-squared:                  0.583
Method:                 Least Squares   F-statistic:                     41.58
Date:                Thu, 31 Oct 2024   Prob (F-statistic):           5.53e-07
Time:                        10:34:02   Log-Likelihood:                -126.80
No. Observations:                  30   AIC:                             257.6
Df Residuals:                      28   BIC:                             260.4
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
==
                     coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
--
const             17.4673      6.491      2.691      0.012       4.172
30.763
Hours of Study     7.4806      1.160      6.448      0.000       5.104
9.857
==============================================================================
Omnibus:                        8.132   Durbin-Watson:                   1.711
Prob(Omnibus):                  0.017   Jarque-Bera (JB):                6.470
Skew:                           0.934   Prob(JB):                       0.0394
```

```
Kurtosis:                       4.298   Cond. No.                      11.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]:

```python
from pulp import LpProblem, LpMaximize, LpVariable, lpSum, LpStatus, value

# 4. Linear Optimization Problem

# Create the linear programming problem
problem = LpProblem("Maximize_Production", LpMaximize)

# Define decision variables
x1 = LpVariable('x1', lowBound=0, cat='Continuous')  # Product A
x2 = LpVariable('x2', lowBound=0, cat='Continuous')  # Product B
x3 = LpVariable('x3', lowBound=0, cat='Continuous')  # Product C

# Objective function
problem += lpSum([x1 + x2 + x3]), "Total_Production"

# Constraints
problem += (2 * x1 + 1 * x2 + 3 * x3 <= 100, "Machine_X")
problem += (4 * x1 + 3 * x2 + 2 * x3 <= 85, "Machine_Y")

# Solve the problem
problem.solve()

# Output the results
print("Status:", LpStatus[problem.status])
print("Optimal Production of Product A (x1):", value(x1))
print("Optimal Production of Product B (x2):", value(x2))
print("Optimal Production of Product C (x3):", value(x3))
print("Maximum Total Production:", value(problem.objective))
```

```
Status: Optimal
Optimal Production of Product A (x1): 0.0
Optimal Production of Product B (x2): 7.8571429
Optimal Production of Product C (x3): 30.714286
Maximum Total Production: 38.5714289
```