2024

SLPs' Level of Preparedness When Working with Interpreters

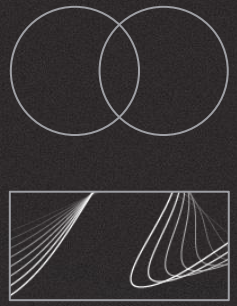
Fermina Yat

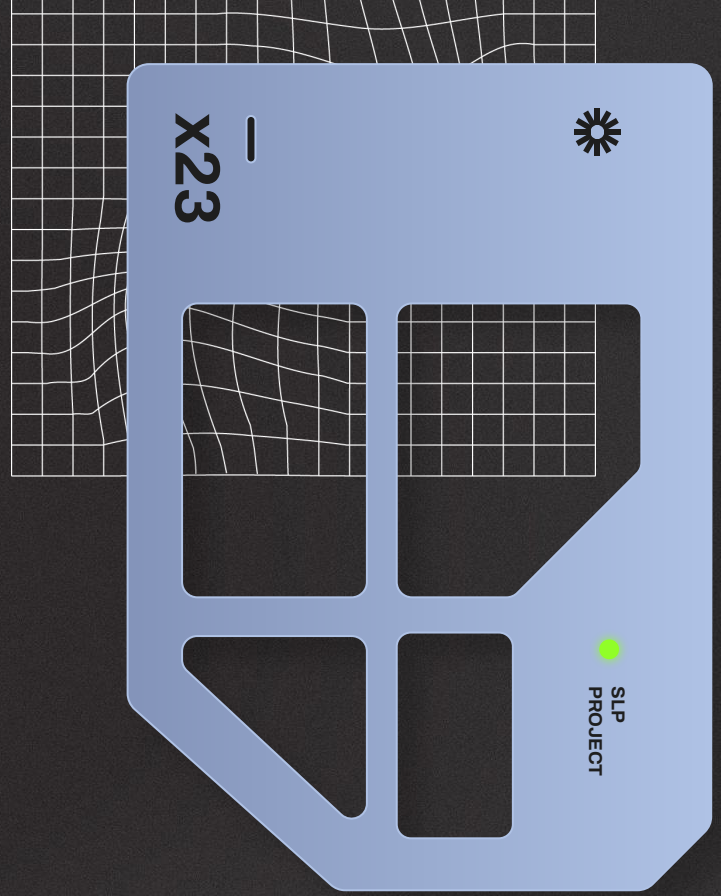
x23

Table of Contents

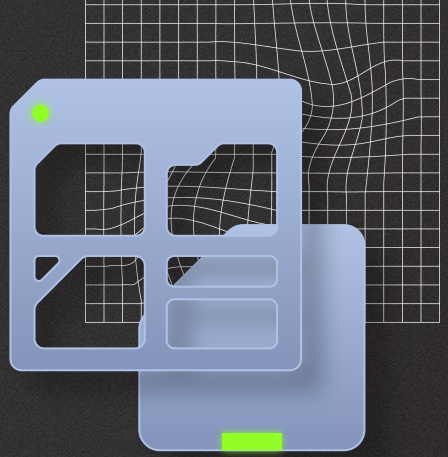
01

Background





Section 1



System review loaded

Background

The study evaluates how prepared speech-language pathologists (SLPs) are to collaborate with interpreters in treating multilingual children with communication disorders in school settings. A survey was conducted in hopes of assessing this.

- Survey was divided into four sections
  - Demographics
  - Experience
  - Working With Interpreters
  - Preparedness

Assessing medical condition...

Loading data...

91%

Additional Background Info

Research Question

- How do school-based speech-language pathologists perceive their level of preparedness to work with interpreters when working with multilingual children with possible communication disorders in elementary schools on Long Island, New York?

Problem

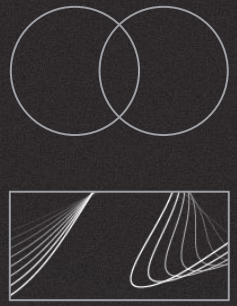
- Studies have shown that SLPs often find themselves unprepared to handle multilingual students.

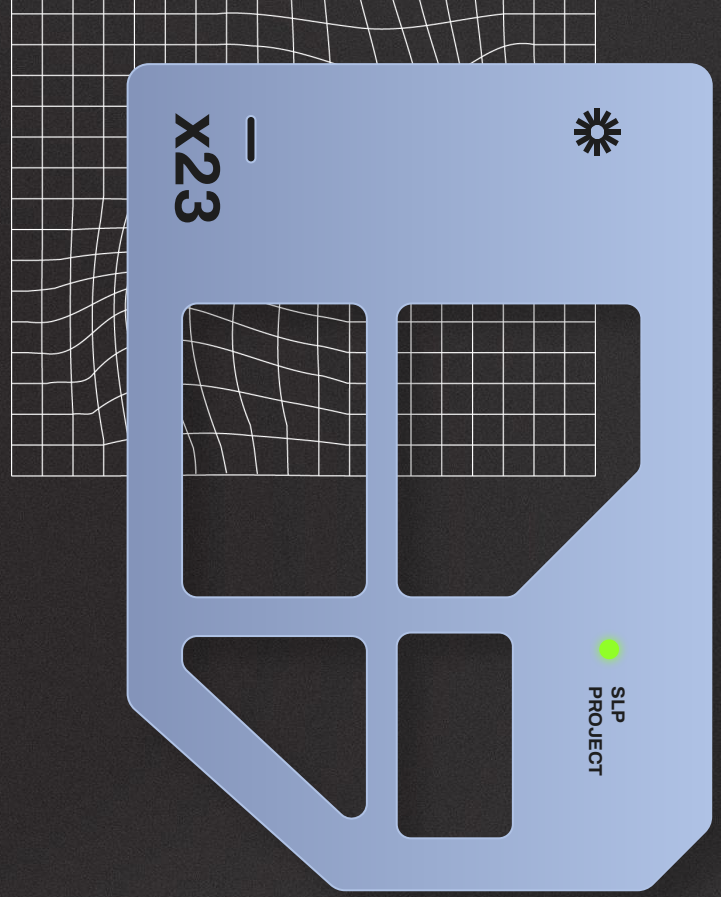
Purpose

- The study aims to enhance educational and therapeutic outcomes for multilingual children, ensuring they receive effective support.

02

Data Cleaning





Section 2

Data Cleaning

```python
#IMPORT PACKAGES
import pandas as pd
import numpy as py
```

Python

```python
#UPLOADING DATA
df = pd.read_csv('/Users/ferminayat/Downloads/SLPFiles/SLPrep.csv', header=0, skiprows=[1,2])
df
```

Python

```python
#CLEAN COLUMNS
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '_').str.replace(')', '_').str.replace('-', '_').str.replace('__', '_').str.replace('.', '_').str.replace(r
df
```

Python

```python
# SPECIFY COLUMNS TO FOCUS ON
focus_columns = ['q11', 'q29', 'q30', 'q31', 'q32', 'q34', 'q36', 'q45', 'q54']

# FILTER DATAFRAME
df_focus = df[focus_columns]
```

Python

Data Cleaning

```python
# MAP RESPONSES FOR Q11

experience_mapping = {
    'Less than one year': 1,  # Less than one year
    '1-2': 1,  # 1-2 years
    '3-5': 1,  # 3-5 years
    '6-10 ': 2,  # 6-10 years
    '11-15 \t\t': 2,  # 11-15 years
    '16-20\t\t\t': 3,  # 16-20 years
    '21+': 3,   # 21+ years
}
df['q11'] = df['q11'].replace(experience_mapping)

# MAP RESPONSES FOR Q29, Q30, Q31, Q54

frequency_mapping = {
    'Never': 0,
    'Rarely ': 1,
    'Rarely': 1,
    'Sometimes': 2,
    'Often ': 3,
    'Often': 3,
    'Always': 4
}
df['q29'] = df['q29'].replace(frequency_mapping)
df['q30'] = df['q30'].replace(frequency_mapping)
df['q31'] = df['q31'].replace(frequency_mapping)
df['q54'] = df['q54'].replace(frequency_mapping)

# MAP RESPONSES FOR Q32, Q34, Q36, Q45

agreement_mapping = {
    'Strongly Disagree': 1,
    'Disagree': 2,
    'Neutral': 3,
    'Agree': 4,
    'Strongly Agree': 5
}
df['q32'] = df['q32'].replace(agreement_mapping)
df['q34'] = df['q34'].replace(agreement_mapping)
df['q36'] = df['q36'].replace(agreement_mapping)
df['q45'] = df['q45'].replace(agreement_mapping)
```

Python

Data Cleaning

```python
# PRINTING SELECTED COLUMNS

print("Filtered DataFrame with Selected Columns:")
print(df_focus.head(41))
```

Python

```python
# APPLY MAPPING
# MAP RESPONSES Q11
df_focus['q11'] = df_focus['q11'].replace(experience_mapping)

# MAP RESPONSES Q29, Q30, Q31, Q54
df_focus['q29'] = df_focus['q29'].replace(frequency_mapping)
df_focus['q30'] = df_focus['q30'].replace(frequency_mapping)
df_focus['q31'] = df_focus['q31'].replace(frequency_mapping)
df_focus['q54'] = df_focus['q54'].replace(frequency_mapping)

# MAP RESPONSES Q32, Q34, Q36, Q45
df_focus['q32'] = df_focus['q32'].replace(agreement_mapping)
df_focus['q34'] = df_focus['q34'].replace(agreement_mapping)
df_focus['q36'] = df_focus['q36'].replace(agreement_mapping)
df_focus['q45'] = df_focus['q45'].replace(agreement_mapping)

# VERIFY MAPPED DATAFRAME
print("Mapped DataFrame:")
print(df_focus.head(41))
```

Python

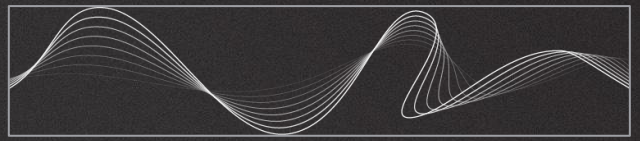
```python
# DROP ROWS WITH NaN VALUES IN SPECIFIED COLUMNS
columns_to_check = ['q11', 'q29', 'q30', 'q31', 'q32', 'q34', 'q36', 'q45', 'q54']
df_cleaned = df_focus.dropna(subset=columns_to_check)

# VERIFY RESULT
print(f"Number of rows after dropping NaN in specific columns: {len(df_cleaned)}")
print(df_cleaned)
```

Python

Data Cleaning Output

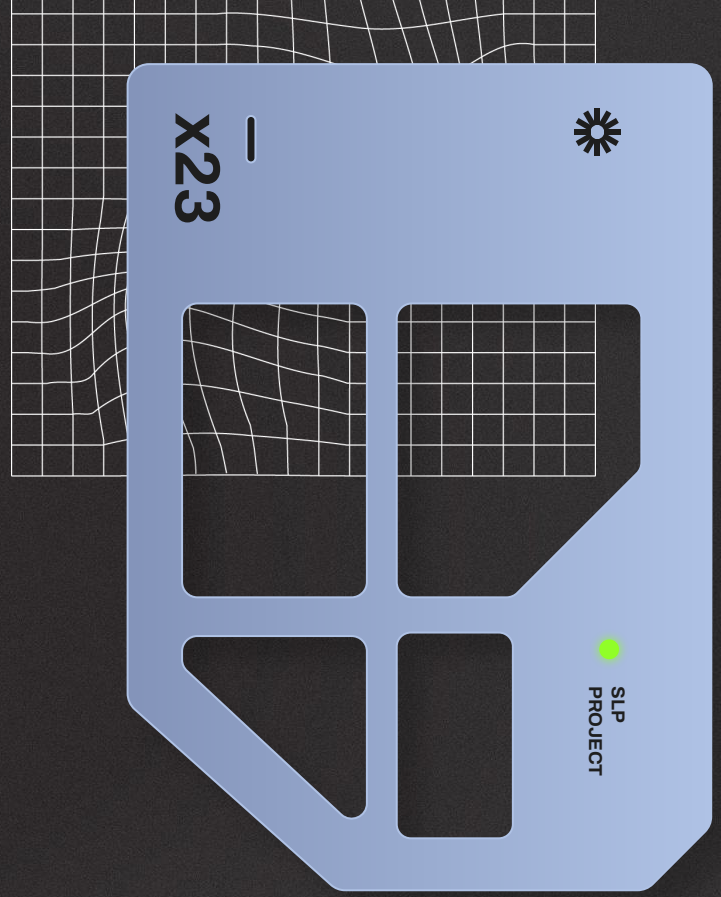
SLP PROJECT



Number of rows after dropping NaN in specific columns: 31

	q11	q29	q30	q31	q32	q34	q36	q45	q54
2	1.0	3.0	3.0	3.0	3.0	2.0	2.0	3.0	2.0
3	2.0	3.0	4.0	3.0	4.0	4.0	2.0	4.0	2.0
7	2.0	1.0	1.0	1.0	4.0	4.0	2.0	4.0	2.0
8	1.0	3.0	3.0	3.0	2.0	2.0	4.0	2.0	1.0
9	2.0	4.0	4.0	3.0	4.0	4.0	2.0	2.0	2.0
11	1.0	3.0	4.0	4.0	3.0	2.0	3.0	2.0	2.0
12	1.0	3.0	4.0	4.0	3.0	2.0	2.0	4.0	1.0
13	3.0	3.0	3.0	4.0	4.0	4.0	2.0	4.0	1.0
14	1.0	4.0	4.0	4.0	4.0	5.0	2.0	4.0	3.0
15	2.0	3.0	3.0	4.0	4.0	3.0	3.0	3.0	2.0
16	1.0	2.0	3.0	2.0	4.0	2.0	2.0	4.0	2.0
19	2.0	3.0	3.0	4.0	5.0	5.0	1.0	5.0	4.0
20	2.0	3.0	3.0	4.0	5.0	4.0	2.0	5.0	3.0
21	3.0	1.0	1.0	1.0	4.0	4.0	2.0	3.0	3.0
22	1.0	3.0	3.0	3.0	4.0	4.0	2.0	4.0	4.0
23	2.0	4.0	4.0	4.0	5.0	5.0	1.0	4.0	3.0
24	1.0	4.0	4.0	3.0	5.0	5.0	1.0	4.0	3.0
25	2.0	4.0	4.0	4.0	5.0	4.0	2.0	2.0	3.0
26	2.0	3.0	3.0	3.0	4.0	4.0	2.0	4.0	2.0
27	1.0	3.0	3.0	3.0	3.0	4.0	2.0	3.0	2.0
30	2.0	4.0	4.0	2.0	4.0	4.0	2.0	4.0	2.0
31	2.0	3.0	3.0	3.0	5.0	5.0	1.0	4.0	3.0
32	2.0	3.0	3.0	3.0	5.0	5.0	1.0	5.0	4.0
33	1.0	4.0	4.0	3.0	3.0	3.0	2.0	3.0	2.0
34	2.0	4.0	3.0	4.0	5.0	5.0	2.0	5.0	3.0
35	2.0	3.0	3.0	4.0	5.0	5.0	1.0	5.0	3.0
36	2.0	4.0	4.0	4.0	5.0	4.0	2.0	5.0	3.0
37	1.0	3.0	3.0	3.0	3.0	2.0	4.0	2.0	1.0

03 Results



Chi-Square

```
pip install scipy
```

Python

```
import pandas as pd
from scipy.stats import chi2_contingency
```

Python

```
# COMBINE Q29, Q30, Q31 INTO FREQUENCY SCORE
df_cleaned['FrequencyScore'] = df_cleaned[['q29', 'q30', 'q31']].sum(axis=1)

# CATEGORIZE INTO FREQUENCY LEVELS
def map_frequency(score):
    if score <= 6:  # LOW FREQUENCY
        return "Low"
    elif score <= 9:  # MEDIUM FREQUENCY
        return "Medium"
    else:
        return "High"  # HIGH FREQUENCY

df_cleaned['FrequencyLevel'] = df_cleaned['FrequencyScore'].apply(map_frequency)

# MAPPING YEARS OF EXPERIENCE
experience_mapping = {
    1: "Low Experience",  # CORRESPONDS a, b, c
    2: "Moderate Experience",  # CORRESPONDS  d, e
    3: "High Experience"  # CORRESPONDS f, g
}

df_cleaned['ExperienceGroup'] = df_cleaned['q11'].map(experience_mapping)

# CONTINGENCY TABLE
contingency_table = pd.crosstab(df_cleaned['ExperienceGroup'], df_cleaned['FrequencyLevel'])
print("Contingency Table:")
print(contingency_table)

# CHI-SQAURE TEST
from scipy.stats import chi2_contingency
chi2, p, dof, expected = chi2_contingency(contingency_table)

# OUTPUT RESULTS
print("\nChi-Square Test Results:")
print(f"Chi-Square Statistic: {chi2}")
print(f"P-Value: {p}")
print(f"Degrees of Freedom: {dof}")
print("Expected Frequencies:")
print(expected)
```

Python

Chi-Square Results

Key Findings

- P-Value = 0.073
- Indicates no significant association between years of experience and frequency of interpreter use.
- Closeness to 0.05 indicates that there may be other factors influencing interpreter use.

Other Factors

Further exploration of factors could yield insights on improving interpreter use among speech-language pathologists.

- Education
- Training Resources
- Cultural Competency Workplace Environment

One-Way ANOVA

```python
import pandas as pd
from scipy.stats import f_oneway

# Step 1: CALCULATE PREPAREDNESS SCORE - SUM OF Q32, Q34, and Q36
df_cleaned['PreparednessScore'] = df_cleaned[['q32', 'q34', 'q36']].sum(axis=1)

# FREQUENCY OF WORKSHOPS (Q54)
groups = [df_cleaned[df_cleaned['q54'] == value]['PreparednessScore'] for value in df_cleaned['q54'].unique()]

# ONE WAY ANOVA
f_stat, p_value = f_oneway(*groups)

# OUTPUT RESULTS
print("One-Way ANOVA Results:")
print(f"F-Statistic: {f_stat}")
print(f"P-Value: {p_value}")

# VISUALIZE RESULTS
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x='q54', y='PreparednessScore', data=df_cleaned)
plt.xlabel('Frequency of Workshops (Q54)')
plt.ylabel('Preparedness Score')
plt.title('Preparedness Score by Frequency of Workshops')
plt.show()
```

Python

One-Way ANOVA Results

Key Findings

- F = 10.31, P Value = .000107, p < .05)
- The one-way ANOVA reveals a significant relationship between workshop frequency during graduate programs and participants' perceived preparedness.

Meaning

- Participants who attended workshops "Often" or "Always" reported higher preparedness levels then those with lower workshop attendance.
- Frequent workshop opportunities during graduate programs play a critical role in readiness to work with interpreters.

Two-Way ANOVA

```
pip install statsmodels
```

Python

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
import matplotlib.pyplot as plt
import seaborn as sns
```

Python

```


# COMPUTE PREPAREDNESS SCORE
df_cleaned['PreparednessScore'] = df_cleaned[['q32', 'q34', 'q36']].sum(axis=1)

# TWO-WAY ANOVA
model = ols('PreparednessScore ~ C(q45) * C(q54)', data=df_cleaned).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

# DISPLAY RESULTS
print("Two-Way ANOVA Results:")
print(anova_table)

# VISUALIZE INTERACTION EFFECT
sns.boxplot(
    x='q54', y='PreparednessScore', hue='q45', data=df_cleaned, palette='Set3'
)
plt.title("Preparedness Score by Workshop Frequency and Quality")
plt.xlabel("Frequency of Workshops (Q54)")
plt.ylabel("Preparedness Score")
plt.legend(title="Quality of Workshops (Q45)", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Python

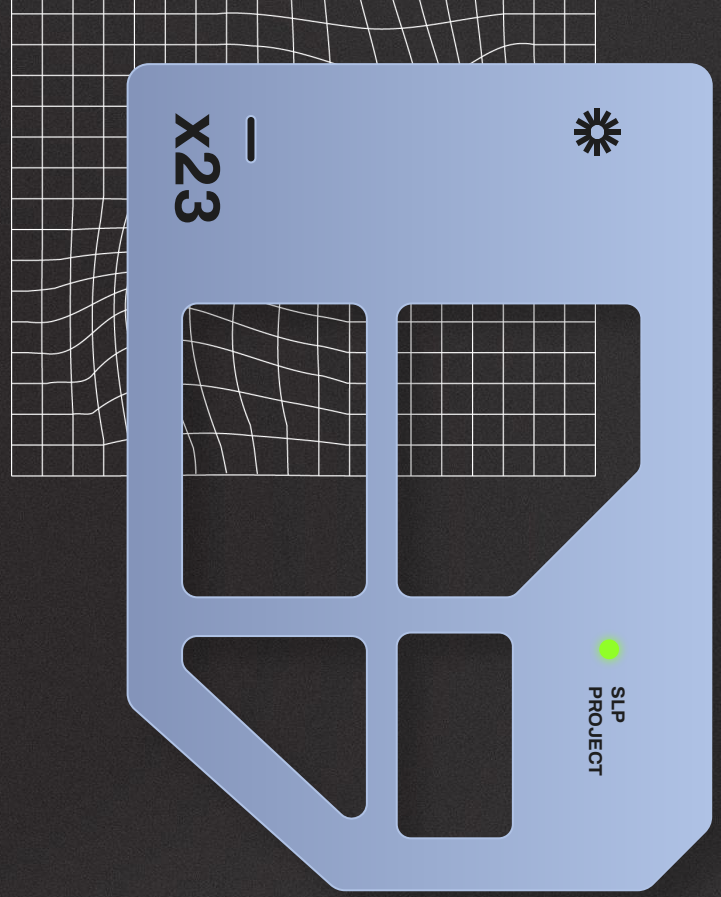
Two-Way ANOVA Results

Key Findings

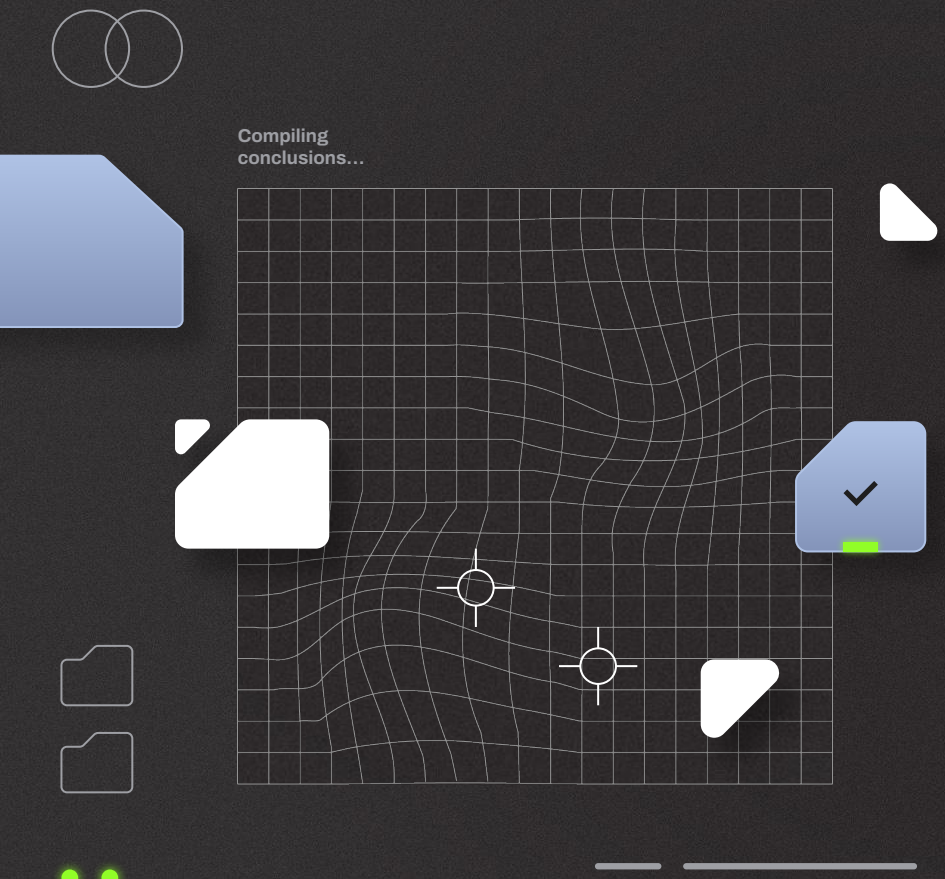
- When investigating the main effect of workshop frequency on preparedness scores, we found a significant p-value (.0016< 0.05), indicating that frequent workshops are associated with higher levels of preparedness.
- When examining the main effect of workshop quality on preparedness scores, we found no significant impact, indicated by p-value of 1.
- The interaction effect between workshop frequency and quality is not statistically significant, but the p-value of 0.064 suggests a trend toward significance, warranting further investigation with a larger dataset.

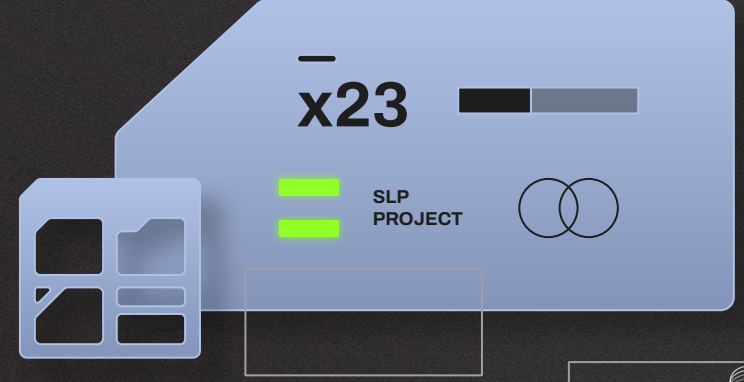
04

Reflections



Section 4





Conclusions

- Time spent data cleaning vs. statistical analysis.
- Found that preparation beforehand made cleaning and analysis easier.
- Many different ways to group questions.
- Weeks of preparation before any coding even took place.

Thank You!

CREDITS: This presentation template was created by Slidesgo, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution