Project Business Statistics: E-news Express

Problem Statement

```
E-news Express suspects that its current webpage design is less effective at retaining visitors
and converting them into subscribers. To address this, the design team has developed a new
landing page.
To evaluate the effectiveness of the new design, my Data Science team has been brought in to run
an A/B test.
```

Our Approach:

- Randomly assign 100 users into two groups: the control group (existing landing page) and the treatment group (new landing page).
- Collect data on user interactions, including time spent on the page, conversion status, and preferred language, for both groups.
- Perform statistical testing at a 5% significance level, using appropriate visualizations, hypothesis formulation, and p-value calculations.
- Use insights derived from the analysis to provide recommendations on whether to adopt the new landing page to enhance user engagement and increase subscriptions.

Objectives

```
The primary objective is to assess whether the new landing page improves user engagement and
subscription conversion rates. Specifically, the analysis will address the following questions:
```

1. **Time Spent Comparison:**
   - Do users spend more time on the new landing page compared to the existing landing page?
2. **Conversion Rate Comparison:**
   - Is the conversion rate (the proportion of users who visit the landing page and get converted) for the new page greater than the conversion rate for the old page?
3. **Language Dependence:**
   - Does the converted status depend on the preferred language?
4. **Language and Engagement:**
   - Is the time spent on the new page the same for the different language users?

Import all the necessary libraries

In [4]:

```
# Using !pip to install needed packages. --upgrade to ensure the latest compatible & I removed -q (quiet) to displa
!pip install numpy pandas matplotlib seaborn scipy --upgrade --user
```

```
Requirement already satisfied: numpy in /Applications/anaconda3/lib/python3.12/site-packages (1.26.4)
Collecting numpy
  Using cached numpy-2.2.0-cp312-cp312-macosx_14_0_arm64.whl.metadata (62 kB)
Requirement already satisfied: pandas in ./.local/lib/python3.12/site-packages (2.2.3)
Requirement already satisfied: matplotlib in ./.local/lib/python3.12/site-packages (3.9.4)
Requirement already satisfied: seaborn in /Applications/anaconda3/lib/python3.12/site-packages (0.13.2)
Requirement already satisfied: scipy in ./.local/lib/python3.12/site-packages (1.14.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /Applications/anaconda3/lib/python3.12/site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /Applications/anaconda3/lib/python3.12/site-packages (from pandas) (2
024.1)
Requirement already satisfied: tzdata>=2022.7 in /Applications/anaconda3/lib/python3.12/site-packages (from pandas)
(2023.3)
Requirement already satisfied: contourpy>=1.0.1 in /Applications/anaconda3/lib/python3.12/site-packages (from matplo
tlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /Applications/anaconda3/lib/python3.12/site-packages (from matplotli
b) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /Applications/anaconda3/lib/python3.12/site-packages (from matpl
otlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /Applications/anaconda3/lib/python3.12/site-packages (from matpl
otlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /Applications/anaconda3/lib/python3.12/site-packages (from matplot
lib) (23.2)
Requirement already satisfied: pillow>=8 in /Applications/anaconda3/lib/python3.12/site-packages (from matplotlib)
(10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /Applications/anaconda3/lib/python3.12/site-packages (from matplo
tlib) (3.0.9)
Requirement already satisfied: six>=1.5 in /Applications/anaconda3/lib/python3.12/site-packages (from python-dateuti
l>=2.8.2->pandas) (1.16.0)
```

In [5]:

```
# Importing the needed packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```python
import seaborn as sns
from scipy import stats
from scipy.stats import ttest_ind
from statsmodels.stats.proportion import proportions_ztest
from scipy.stats import chi2_contingency
from scipy.stats import f_oneway
```

Reading the Data into a DataFrame

In [7]:

```python
# Defining the file path to the CSV file on my local computer
file_path = '/Users/estarconsulting/Downloads/abtest.csv'

# Loading the CSV file into a DataFrame
abtest_df = pd.read_csv(file_path)

# Displaying info about the dataset
print(abtest_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   user_id                100 non-null    int64  
 1   group                  100 non-null    object 
 2   landing_page           100 non-null    object 
 3   time_spent_on_the_page 100 non-null    float64
 4   converted              100 non-null    object 
 5   language_preferred     100 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 4.8+ KB
None
```

Explore the dataset and extract insights using Exploratory Data Analysis

In [9]:

```python
# Displaying the first and last 5 rows of the dataset
print("First five rows of the dataset:")
display(abtest_df.head())

print("\n", "Last five rows of the dataset:")
display(abtest_df.tail())
```

```
First five rows of the dataset:
```

	user_id	group	landing_page	time_spent_on_the_page	converted	language_preferred
0	546592	control	old	3.48	no	Spanish
1	546468	treatment	new	7.13	yes	English
2	546462	treatment	new	4.40	no	Spanish
3	546567	control	old	3.02	no	French
4	546459	treatment	new	4.75	yes	Spanish

```
 Last five rows of the dataset:
```

	user_id	group	landing_page	time_spent_on_the_page	converted	language_preferred
95	546446	treatment	new	5.15	no	Spanish
96	546544	control	old	6.52	yes	English
97	546472	treatment	new	7.07	yes	Spanish
98	546481	treatment	new	6.20	yes	Spanish
99	546483	treatment	new	5.86	yes	English

In [10]:

```python
# Checking the shape of the dataset
print("Dataset shape:", abtest_df.shape)
```

```
Dataset shape: (100, 6)
```

In [11]:

```python
# Generating a statistical summary for relevant numeric variables (excluding user_id since it has no info to offer)
print("\nStatistical summary of the dataset (excluding user_id):")
display(abtest_df.drop('user_id', axis=1).describe().T)
```

```
Statistical summary of the dataset (excluding user_id):
```

	count	mean	std	min	25%	50%	75%	max
time_spent_on_the_page	100.0	5.3778	2.378166	0.19	3.88	5.415	7.0225	10.71

In [12]:

```python
# Checking for missing values in the dataset
missing_values = abtest_df.isnull().sum()
print("Missing values per column:","\n",missing_values)
```

```
Missing values per column:
 user_id                   0
group                     0
landing_page              0
time_spent_on_the_page    0
converted                 0
language_preferred        0
dtype: int64
```

In [13]:
```
# Checking for duplicate rows in the dataset
duplicate_rows = abtest_df.duplicated().sum()
print("Number of duplicate rows in the dataset", duplicate_rows)
```

```
Number of duplicate rows in the dataset 0
```

Summarizing the Data Exploration Section:

1. **Head and Tail Check:**

   - The `head()` and `tail()` outputs show exactly what we'd expect: all six columns are populated with the right type of data, like `user_id`, `group`, `landing_page`, and `converted`. Everything matches up with the experiment details.

2. **Data Shape:**

   - The dataset shape confirms there are **100 rows** (100 random customers) and **6 columns**, which is exactly what the experiment setup mentioned.

3. **Stat Summary:**

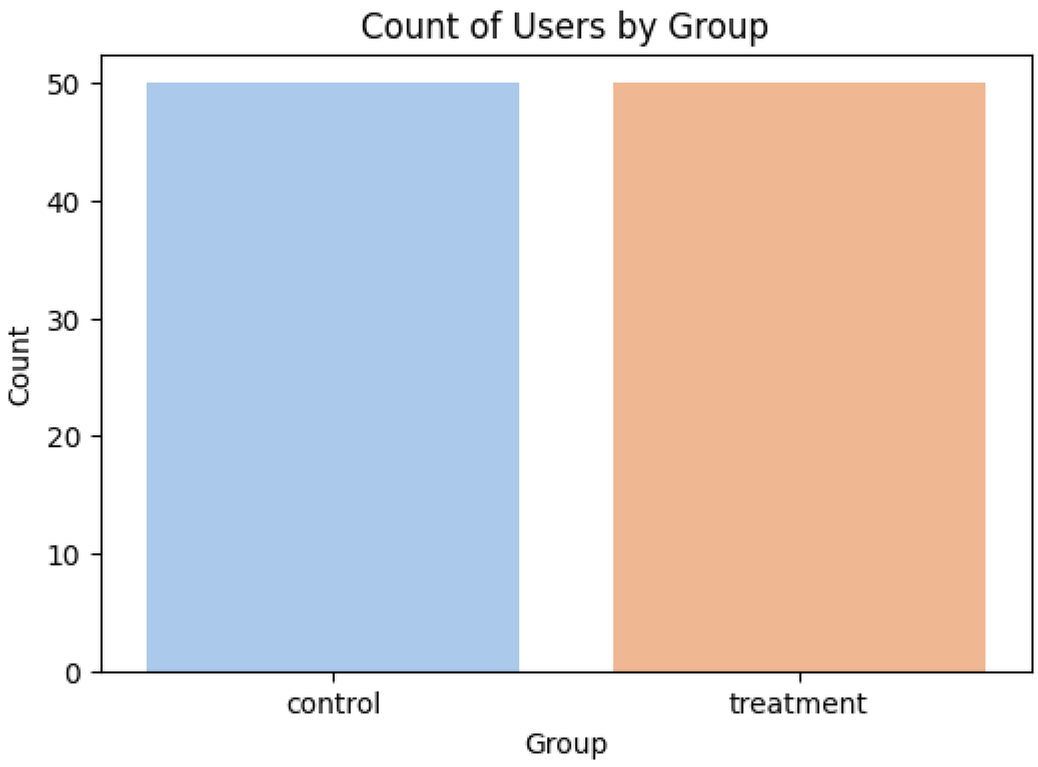
   - Since only numerical columns can be analyzed, **time_spent_on_the_page** was the focus here. Key takeaways:
     - No missing values (count = 100).
     - Average time spent: around **5.38 minutes**.
     - Shortest time: **0.19 minutes** (about 11 seconds), and the longest: around **10.71 minutes**.
     - This shows a range from very little time spent to quite a lot. A boxplot will help us see how this breaks down—whether users tend to spend too little time, just enough, or a lot on the page.

4. **Cleanliness Check:**

   - **No missing values** across any columns.
   - **No duplicate rows**, so the dataset is clean and ready for deeper analysis.

Univariate Analysis

In [16]:
```
# Univariate Analysis for 'group' (Categorical)
# Using a bar plot to count the number of users in each group (control/treatment), as it's a clear way to summarize
plt.figure(figsize=(6, 4))
sns.countplot(data=abtest_df, x="group", hue="group", palette="pastel")
plt.title("Count of Users by Group")
plt.xlabel("Group")
plt.ylabel("Count")
plt.show()
```

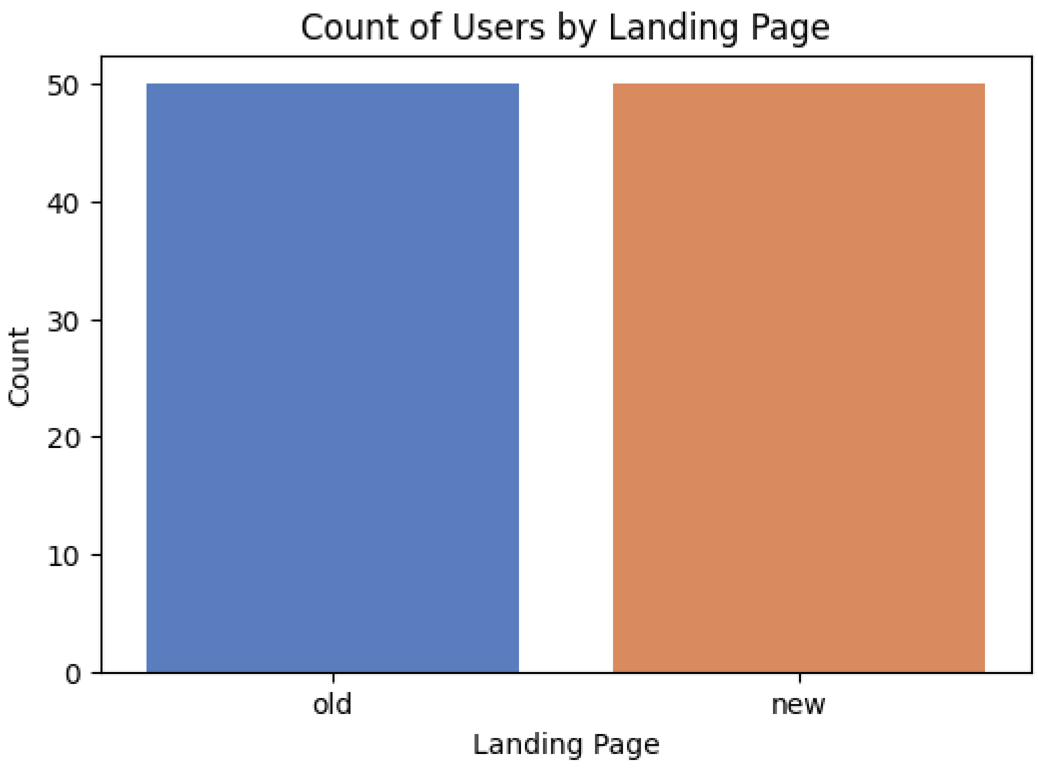


Observation: Univariate Analysis for 'group' (Categorical)

- Equal Group Sizes: The plot shows that both the control and treatment groups have an equal number of users (50 each).
- Required for A/B Test: This balance is essential for a valid A/B test comparison, as it minimizes the influence of pre-existing group differences on the results.
- Equal Distribution Across Landing Pages & Languages: When we move to multivariate analysis, we should expect and verify that users are also equally distributed across landing pages and preferred languages within each group.

In [18]:

```
# Univariate Analysis for 'landing_page' (Categorical)
# Bar plot since it’s straightforward for category counts.
plt.figure(figsize=(6, 4))
sns.countplot(data=abtest_df, x="landing_page", hue="landing_page", palette="muted")
plt.title("Count of Users by Landing Page")
plt.xlabel("Landing Page")
plt.ylabel("Count")
plt.show()
```

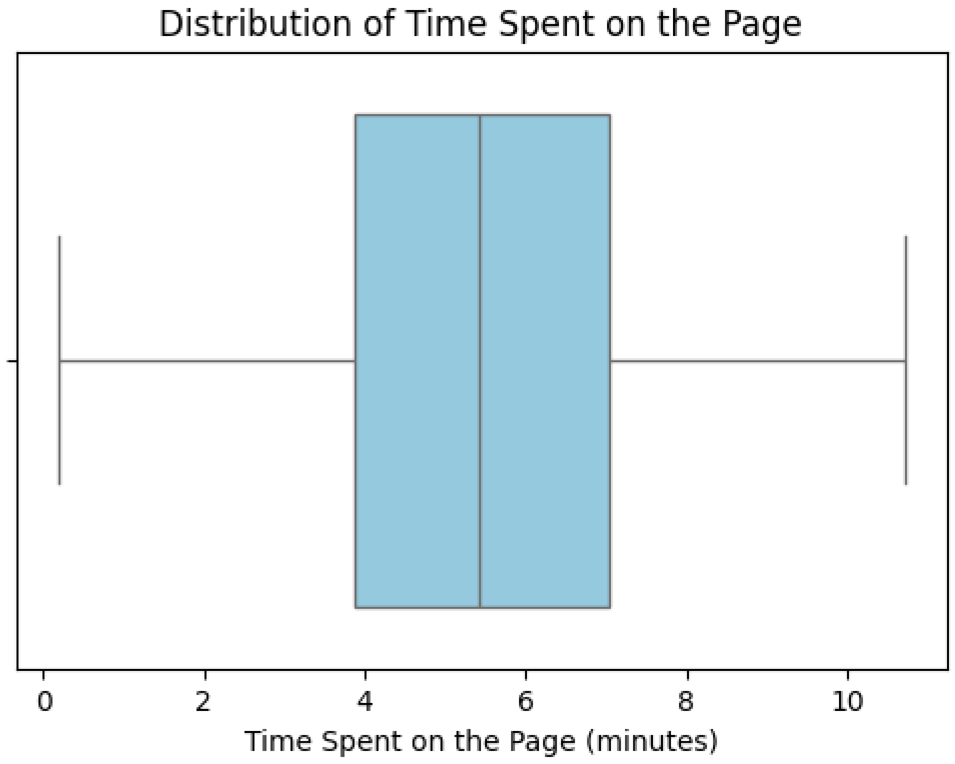


Observation: Univariate Analysis for 'landing_page' (Categorical)

- Similar to the 'group' category, the landing page has an equal distribution among the old and new versions as exepected. This equal distribution ensures a fair comparison between the two landing page designs.
- And in multivariate analysis comparing user behavior of time spent, conversions between the landing page groups will give us more eintereting insights.

In [20]:

```
# Univariate Analysis for 'time_spent_on_the_page' (Numerical)
# Using a box plot to visualize the distribution of time spent, as it highlights spread and potential outliers.
plt.figure(figsize=(6, 4))
sns.boxplot(data=abtest_df, x="time_spent_on_the_page", color="skyblue")
plt.title("Distribution of Time Spent on the Page")
plt.xlabel("Time Spent on the Page (minutes)")
plt.show()
```



Observation: Univariate Analysis for 'time_spent_on_the_page' (Numerical)

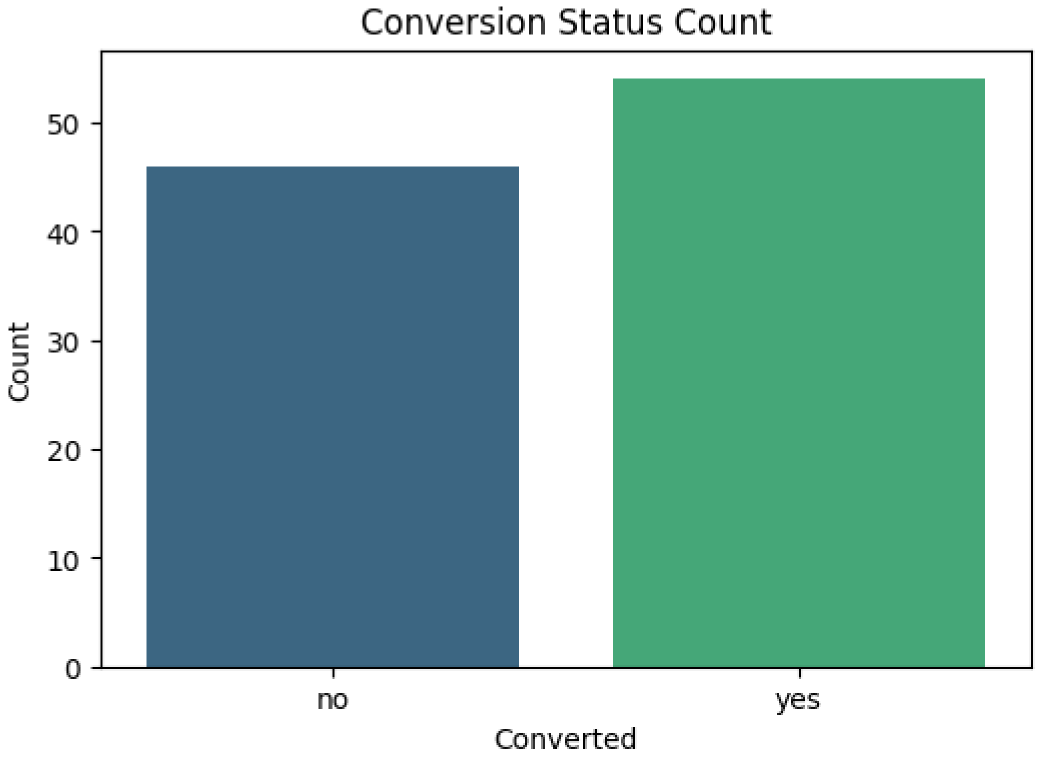
Connecting observation with the statistical summary for relevant numeric variables.

- Possible Symmetry: The boxplot suggests a potentially symmetrical distribution, supported by the close values of the mean (5.38) and median (5.41).
- Moderate Spread: The IQR (75th percentile - 25th percentile = 5.71 - 3.88 = 1.83 minutes) indicates a moderate spread in the time spent on the page.
- No Extreme Outliers: The whiskers extend to a reasonable range, suggesting no extreme outliers in the data.

- Time Range: The minimum time spent is 0.19 minutes, and the maximum is 10.71 minutes, indicating a considerable range in user engagement.

In [22]:

```
# Univariate Analysis for 'converted' (Categorical)
# A bar plot to show the number of users converted (yes/no), as it visually breaks down the proportions.
plt.figure(figsize=(6, 4))
sns.countplot(data=abtest_df, x="converted", hue="converted", palette="viridis")
plt.title("Conversion Status Count")
plt.xlabel("Converted")
plt.ylabel("Count")
plt.show()
```



In [23]:

```
# Calculating % difference in conversion for better observational analysis

# Counting the number of users in each category
converted_counts = abtest_df['converted'].value_counts()

# Calculating the percentage difference
percentage_difference = ((converted_counts['yes'] - converted_counts['no']) / ((converted_counts['yes'] + converted

print("Percentage Difference:", percentage_difference)
```

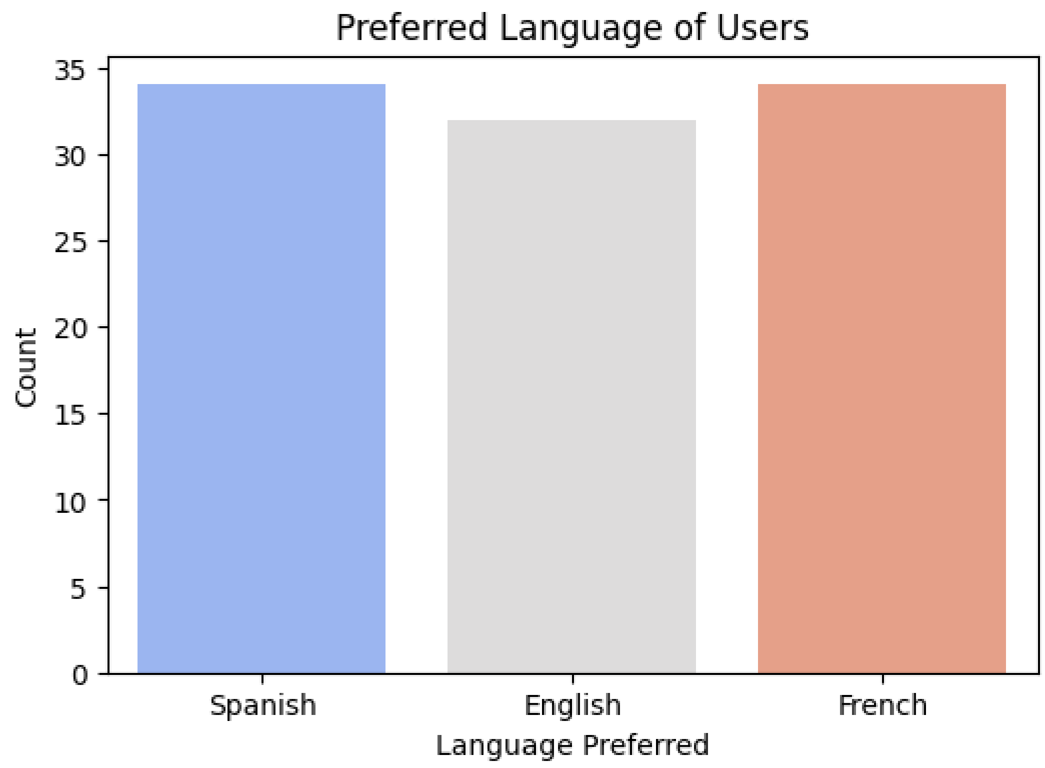
```
Percentage Difference: 16.0
```

Observation: Univariate Analysis for 'converted' (Categorical)

- The bar plot shows that slightly more users converted (yes) compared to those who did not (no). This represents a small but potentially significant edge in conversions for the "yes" category.
- The calculated percentage difference between "yes" and "no" conversions is 16%. From a percentage standpoint and based on personal expertise in sales and growth, this difference would be considered significant.
- In multivariate analysis, this observation will be further broken down by groups (control vs. treatment) and landing pages to identify specific patterns in conversions. This will help determine if the new page design significantly improves conversions.

In [25]:

```
# Univariate Analysis for 'language_preferred' (Categorical)
# Bar plot again to show counts of users by their preferred language, as it’s clear for categorical data.
plt.figure(figsize=(6, 4))
sns.countplot(data=abtest_df, x="language_preferred", hue="language_preferred", palette="coolwarm")
plt.title("Preferred Language of Users")
plt.xlabel("Language Preferred")
plt.ylabel("Count")
plt.show()
```



Observation: Univariate Analysis for 'language_preferred' (Categorical)

- Users are nearly evenly distributed across Spanish, English, and French, with 32-34 users per language.
- This nearly balanced distribution minimizes bias and ensures fair comparisons across languages.
- While language is unlikely to introduce bias, further analysis is needed to determine if it significantly affects user behavior.

Bivariate Analysis

Explaining my approach to Bivariate Analysis.

A. Group vs. Landing Page: - Validation Step: To confirm that the group and landing_page columns have the same information (i.e., control corresponds to old, and treatment corresponds to new). - Action: If confirmed, I'll use landing_page for comparisons since it directly relates to the business problem.

B.

- Landing Page vs. Time Spent: - Objective: Assess whether the new landing page leads to increased time spent compared to the old page. - Analysis: Use visualizations and statistical tests to compare the distributions.

- Landing Page vs. Conversion: - Objective: Determine whether the new landing page improves conversion rates compared to the old page. - Analysis: Use bar plots for visual comparison and a hypothesis test to evaluate statistical significance.

- Language vs. Conversion: - Objective: Investigate if preferred language influences the likelihood of conversion. - Analysis: Use grouped bar plots and a chi-square test to check for significant differences in conversion rates across languages.

- Language vs. Time Spent: - Objective: Explore whether time spent on the page varies significantly across preferred languages. - Analysis: Use box plots for visual comparisons and an ANOVA test to assess statistical differences.

- Importance of Language Insights: - Language preferences can help identify key markets or demographics to target for future optimization. This analysis might not directly affect the A/B test outcome but adds value to business recommendations.

After deciding on this plan, I realized after reviewing the questions below that my intended approach for bivariate analysis directly addresses them. So, I'll proceed with it!

A. Group vs. Landing Page

In [30]:
```
# Validation Step: Checking if group and landing_page have the same information
mapping_consistent = all(
    (abtest_df['group'] == 'control') == (abtest_df['landing_page'] == 'old')
)

# Output result
if mapping_consistent:
    print("Validation Passed: 'group' and 'landing_page' are consistent.")
    print("Action: Using 'landing_page' for further analysis.")
else:
    print("Validation Failed: 'group' and 'landing_page' are not consistent.")
```

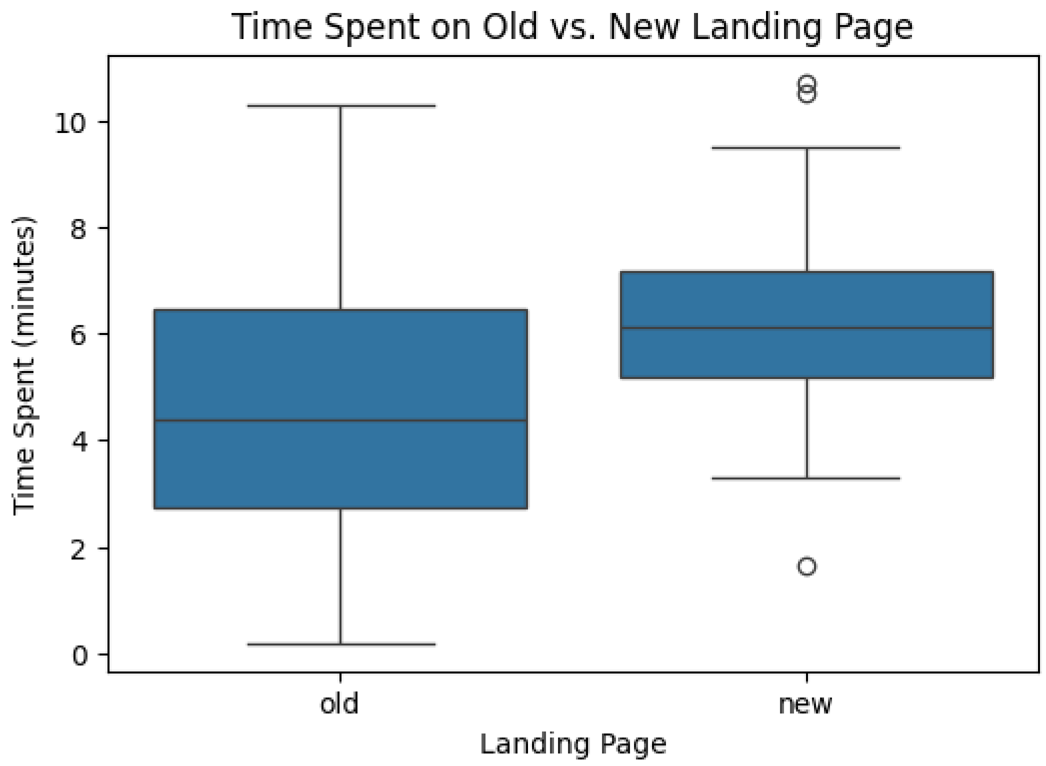
```
Validation Passed: 'group' and 'landing_page' are consistent.
Action: Using 'landing_page' for further analysis.
```

1. Do the users spend more time on the new landing page than the existing landing page?

Perform Visual Analysis

In [33]:

```
# Visual Analysis: Comparing time spent on old vs. new landing pages
plt.figure(figsize=(6, 4))
sns.boxplot(data=abtest_df, x="landing_page", y="time_spent_on_the_page")
plt.title("Time Spent on Old vs. New Landing Page")
plt.xlabel("Landing Page")
plt.ylabel("Time Spent (minutes)")
plt.show()
```



Observations:

- **Median Time:** The median time spent on the new landing page appears slightly higher than that on the old landing page.
- **Spread:** The interquartile range (IQR) for the new landing page is narrower, suggesting more consistent time spent compared to the old page, which has a wider spread.
- **Outliers:**
    - New landing page: 2 users spent significantly more time than average, and 1 user spent significantly less time than average.
- **Maximum and Minimum:** The maximum time spent is similar for both pages, but the minimum is higher for the new landing page.

Interpretation:

- Users tend to spend slightly more time on the new landing page, with more consistent engagement.
- The presence of outliers suggests some users behave differently, but overall, the new design seems to encourage higher engagement.

Step 1: Define the null and alternate hypotheses

- Null Hypothesis (H0): There is no difference in the mean time spent on the old and new landing pages. H0: μ_old = μ_new

- Alternate Hypothesis (Ha): The mean time spent on the new landing page is greater than the mean time spent on the old landing page. Ha: μ_new > μ_old

Step 2: Select Appropriate test

A one-tailed independent two-sample t-test is the rigth choice because:

- The data involves two independent groups (users on the old vs. new landing page).
- The goal is to test if the mean time spent on the new page is greater than the old page.
- The test assumes the data is normally distributed and has similar variances.

Step 3: Decide the significance level

Significance Level (α): alpha = 0.05
I'll use this significance level of 5%, which is standard in hypothesis testing.

Interpretation:

- If the p-value from the test is less than 0.05, i'll will reject the null hypothesis (H0) and conclude that the new landing page leads to significantly more time spent.

- If the p-value is greater than 0.05, i'll fail to reject H0, indicating insufficient evidence to support that the new page improves time spent.

Step 4: Collect and prepare data

In [42]:

```
# Collecting data for time spent on the old and new landing pages
old_page_time = abtest_df[abtest_df['landing_page'] == 'old']['time_spent_on_the_page']
new_page_time = abtest_df[abtest_df['landing_page'] == 'new']['time_spent_on_the_page']

# Displaying summary statistics for validation
print("Summary of Time Spent on Old Landing Page:")
print(old_page_time.describe())

print("\nSummary of Time Spent on New Landing Page:")
print(new_page_time.describe())
```

```
Summary of Time Spent on Old Landing Page:
count    50.000000
mean      4.532400
std       2.581975
min       0.190000
25%       2.720000
50%       4.380000
75%       6.442500
max      10.300000
Name: time_spent_on_the_page, dtype: float64

Summary of Time Spent on New Landing Page:
count    50.000000
mean      6.223200
std       1.817031
min       1.650000
25%       5.175000
50%       6.105000
75%       7.160000
max      10.710000
Name: time_spent_on_the_page, dtype: float64
```

Step 5: Calculate the p-value

In [44]:

```
# Performing two-sample t-test
t_stat, p_value_two_tailed = ttest_ind(new_page_time, old_page_time, equal_var=True)

# Display the results
print("T-statistic:", t_stat)
print("Two-tailed p-value:", p_value_two_tailed)
```

```
T-statistic: 3.7867702694199856
Two-tailed p-value: 0.0002632247056190011
```

Step 6: Compare the p-value with α

- Significance Level (α): 0.05
- Two-tailed p-value: 0.000263

Comparison:

- The p-value (0.000263) is much smaller than the significance level (0.05).

Step 7: Draw inference

Based on the results of the hypothesis test:

- The **p-value (0.000263)** is significantly smaller than the significance level (0.05), allowing me to confidently **reject the null hypothesis (H_0)**.
- This indicates that the **new landing page** results in a **statistically significant difference** in the mean time spent compared to the old landing page.

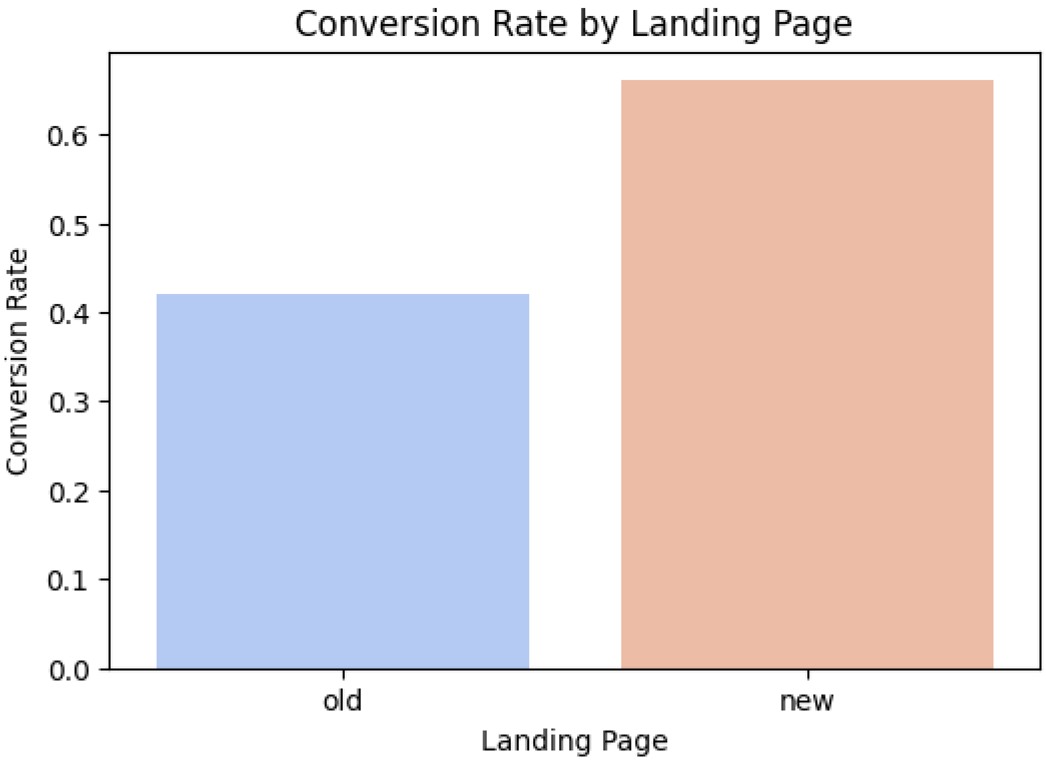
Final Inference:

- Users spend **more time** on the new landing page compared to the old one, as supported by both the visual analysis and statistical testing.
- This suggests that the new landing page design is more effective at engaging users, which could positively impact business outcomes like conversions.

2. Is the conversion rate (the proportion of users who visit the landing page and get converted) for the new page greater than the conversion rate for the old page?

Perform Visual Analysis

In [51]:
```
# Visual Analysis: Comparing conversion rates VS the old and new landing pages
plt.figure(figsize=(6, 4))
sns.barplot(data=abtest_df, x="landing_page", y=(abtest_df["converted"] == "yes").astype(int), errorbar=None, hue="
plt.title("Conversion Rate by Landing Page")
plt.xlabel("Landing Page")
plt.ylabel("Conversion Rate")
plt.show()
```



Observations:

- **Higher Conversion Rate:**
  The conversion rate for the **new landing page** is visibly higher compared to the old landing page.
- **Old Landing Page:**
  The conversion rate appears to be approximately **40%**.
- **New Landing Page:**
  The conversion rate is approximately **60%**, indicating a substantial improvement.

Interpretation:

- **Initial Evidence:**
  The visual comparison suggests that the new landing page is more effective at driving user conversions.
- **Next Steps:**
  A hypothesis test will help determine if this observed difference is statistically significant or could have occurred by chance.

Step 1: Define the null and alternate hypotheses

- Null Hypothesis (H0): The conversion rate for the new landing page is equal to or less than the conversion rate for the old landing page. H0: p_new <= p_old

- Alternate Hypothesis (Ha): The conversion rate for the new landing page is greater than the conversion rate for the old landing page. Ha: p_new > p_old

Step 2: Select Appropriate test

A Two-Proportion Z-Test is the rigth choice because:

- Is used to compare two proportions (in this case, the proportion of users who convert on the new landing page and the proportion of users who convert on the old landing page)

Step 3: Decide the significance level

Significance Level (α): alpha = 0.05
I'll use this significance level of 5%, which is standard/common in hypothesis testing.

Interpretation:

- If the p-value from the test is less than 0.05, we will reject the null hypothesis (H_0) and conclude that the new landing page leads to a significantly higher conversion rate compared to the old landing page.

- If the p-value is greater than 0.05, we will fail to reject the null hypothesis (H_0), indicating insufficient evidence to support that the new page improves conversion rates.

Step 4: Collect and prepare data

In [60]:

```
# Filtering data for conversion counts on the old and new landing pages
old_page_conversions = abtest_df[abtest_df['landing_page'] == 'old']['converted'].value_counts()
new_page_conversions = abtest_df[abtest_df['landing_page'] == 'new']['converted'].value_counts()

# Calculating the number of successes
success_old = old_page_conversions.get('yes', 0)  # Get the number of 'yes' for old page, default to 0 if none
success_new = new_page_conversions.get('yes', 0)  # Get the number of 'yes' for new page, default to 0 if none

n_old = old_page_conversions.sum()  # Total observations on old page
n_new = new_page_conversions.sum()  # Total observations on new page

# Displaying the results
print("Old Landing Page - Conversions: ", success_old, "Total: ", n_old)
print("New Landing Page - Conversions: ", success_new, "Total: ", n_new)
```

```
Old Landing Page - Conversions:  21 Total:  50
New Landing Page - Conversions:  33 Total:  50
```

Step 5: Calculate the p-value

In [62]:

```
# Array of the number of successful conversions for each group
successes = np.array([success_new, success_old])

# Array of the number of observations (total visitors) for each group
nobs = np.array([n_new, n_old])

# Performing the two-proportion z-test
z_stat, p_value = proportions_ztest(successes, nobs, alternative='larger')  # 'larger' indicates one-tailed test

# Displaying the test results
print("Z-statistic:", z_stat)
print("P-value:", p_value)
```

```
Z-statistic: 2.4077170617153842
P-value: 0.008026308204056278
```

Step 6: Compare the p-value with α

- Significance Level (α): 0.05
- p-value: 0.008026308204056278

Comparison:

- The p-value (0.008026308204056278) is much smaller than the significance level (0.05).

Step 7: Draw inference

Based on the results of the hypothesis test:

- We reject the null hypothesis (H_0).
- This provides strong statistical evidence that the new landing page has a significantly higher conversion rate than the old landing page.

Business Insight:

- Implementing the new landing page could potentially increase the number of subscribers based on the observed improvement in conversion rates.

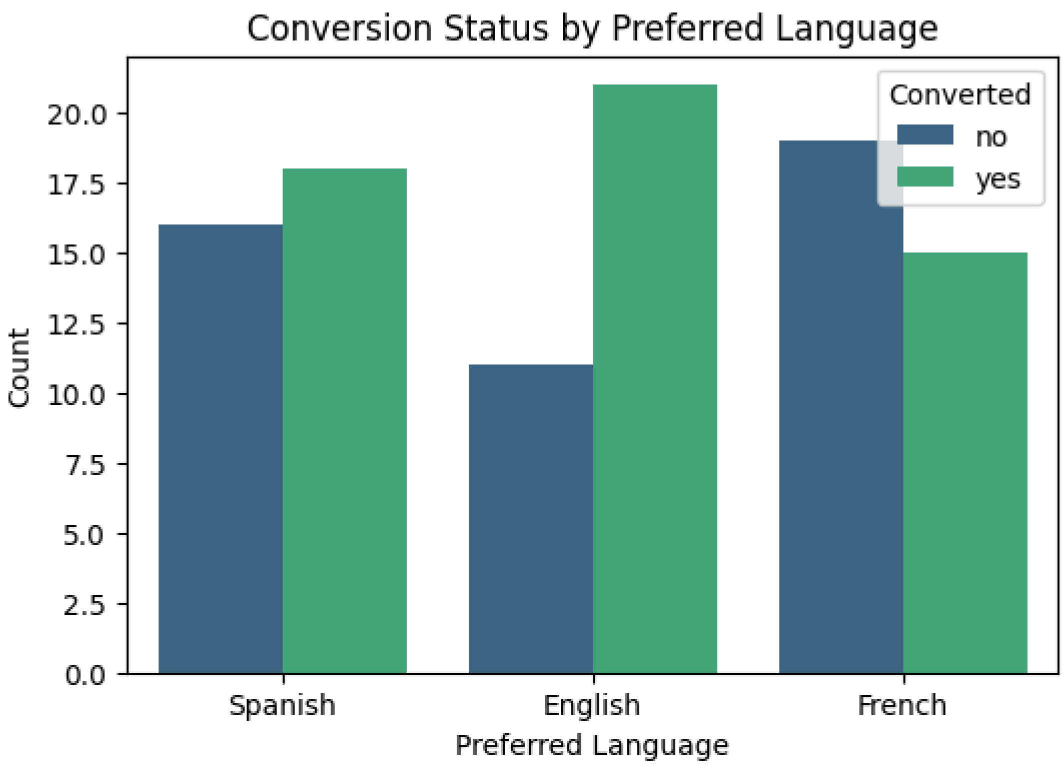
A similar approach can be followed to answer the other questions.

3. Is the conversion and preferred language are independent or related?

Perform Visual Analysis

In [70]:

```
# Visual Analysis: Conversion vs. Preferred Language
plt.figure(figsize=(6, 4))
sns.countplot(data=abtest_df, x="language_preferred", hue="converted", palette="viridis")
plt.title("Conversion Status by Preferred Language")
plt.xlabel("Preferred Language")
plt.ylabel("Count")
plt.legend(title="Converted")
plt.show()
```



Observations:

- **Spanish Users:**
  The conversion count for "yes" and "no" is almost equal, suggesting balanced behavior for this group.
- **English Users:**
  The majority of English-speaking users converted, showing a higher conversion tendency compared to others.
- **French Users:**
  Similar to Spanish users, but slightly fewer conversions ("yes") compared to non-conversions ("no").

Interpretation:

- The visual suggests potential differences in conversion rates across languages, with **English-speaking users** showing higher conversions.

Step 1: Define the null and alternate hypotheses

- Null Hypothesis (H_0): Conversion status and preferred language are independent. P(C ∩ L) = P(C) × P(L)

- Alternate Hypothesis (H_a): Conversion status and preferred language are not independent. P(C ∩ L) ≠ P(C) × P(L)

Step 2: Select Appropriate test

A Chi-Square Test of Independence is the rigth choice because:

- it assesses whether the observed frequencies in each category differ significantly from what would be expected if the two variables were indeed independent.

Step 3: Decide the significance level

Significance Level (α): alpha = 0.05
I'll use this significance level of 5%, which is standard/common in hypothesis testing.

Interpretation:

- If the p-value from the Chi-Square test is less than 0.05, we reject the null hypothesis (H_0) and conclude that conversion status and preferred language are not independent. In other words, there is a significant association between the two variables.

- If the p-value is greater than 0.05, we fail to reject the null hypothesis (H_0), indicating there is insufficient evidence to conclude that conversion status and preferred language are related.

Step 4: Collect and prepare data

In [80]:

```
# Creating a crosstab for conversion status and preferred language
crosstab_table = pd.crosstab(abtest_df['language_preferred'], abtest_df['converted'])

# Display the contingency table
print("Crosstab:")
print(crosstab_table)
```

```
Crosstab:
converted           no  yes
language_preferred
English             11   21
French              19   15
Spanish             16   18
```

Step 5: Calculate the p-value

In [83]:

```python
# Performing the Chi-Square Test of Independence
chi2_stat, p_value, dof, expected = chi2_contingency(crosstab_table)

# Displaying the test results
print(f"Chi-Square Statistic: {chi2_stat}")
print(f"P-value: {p_value}")
print(f"Degrees of Freedom: {dof}")
print("Expected Frequencies:")
print(expected)
```

```
Chi-Square Statistic: 3.0930306905370832
P-value: 0.21298887487543453
Degrees of Freedom: 2
Expected Frequencies:
[[14.72 17.28]
 [15.64 18.36]
 [15.64 18.36]]
```

Step 6: Compare the p-value with α

- Chi-Square Statistic (χ^2): 3.09
- P-value: 0.213
- Degrees of Freedom (dof): 2
- Significance Level (α): 0.05

Comparison:

- The p-value (0.213) is greater than the significance level (α = 0.05).

Step 7: Draw inference

Based on the results of the Chi-Square Test of Independence:

- Since the p-value (0.213) is greater than the significance level (α = 0.05), we fail to reject the null hypothesis (H_0).

This indicates that there is no statistically significant relationship between conversion status and preferred language.

Business Insight:

- Users' preferred language does not appear to influence whether they convert.

This suggests that marketing efforts and landing page design can be language-neutral, with no specific need to target languages differently based on conversion behavior.

A similar approach can be followed to answer the other questions.

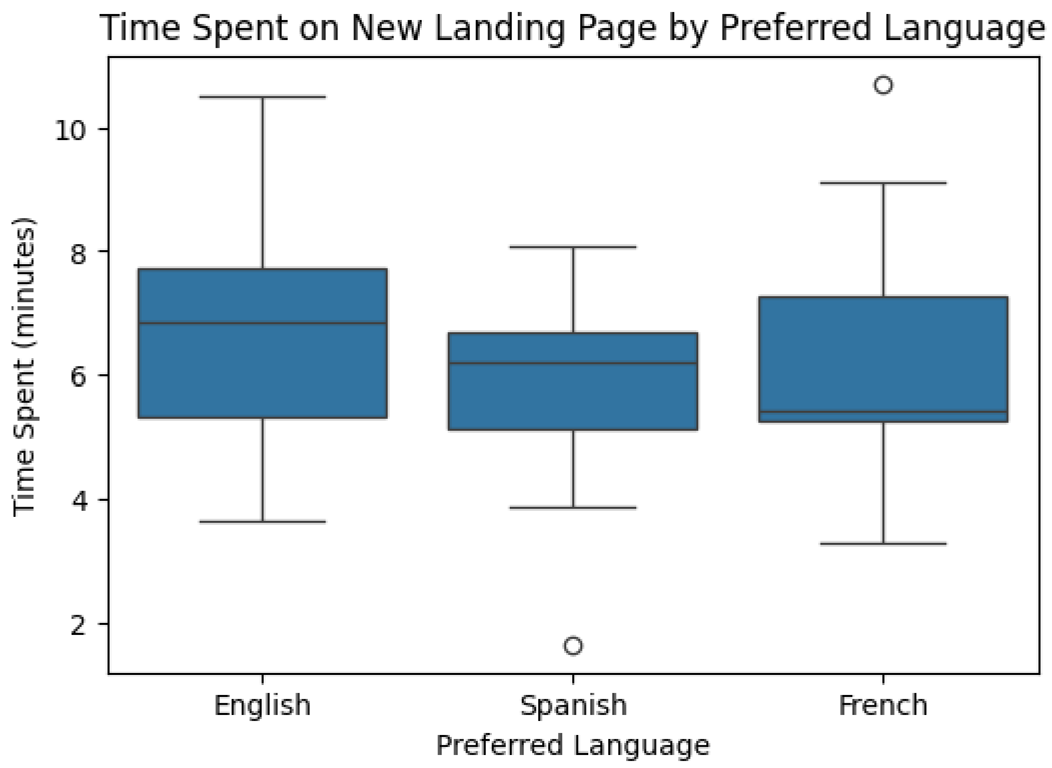
4. Is the time spent on the new page same for the different language users?

Perform Visual Analysis

In [92]:

```python
# Filtering data for users on the new landing page
new_page_data = abtest_df[abtest_df['landing_page'] == 'new']

# Visual Analysis: Comparing time spent on new landing page by preferred language
plt.figure(figsize=(6, 4))
sns.boxplot(data=new_page_data, x="language_preferred", y="time_spent_on_the_page")
plt.title("Time Spent on New Landing Page by Preferred Language")
plt.xlabel("Preferred Language")
plt.ylabel("Time Spent (minutes)")
plt.show()
```



Observations:

- **English Users:**
  The median time spent for English speakers is slightly higher compared to the other languages, with a moderate spread and no extreme outliers.
- **Spanish Users:**
  The time spent is slightly lower, with one visible outlier below 3 minutes.
- **French Users:**
  French speakers show a slightly wider spread than spanish, with an outlier spending more than 10 minutes.

Interpretation:

- There appears to be some variation in the time spent on the new landing page across different preferred languages.
- To confirm whether these differences are statistically significant, I'll will proceed with a hypothesis test.

Step 1: Define the null and alternate hypotheses

- Null Hypothesis (H_0): The mean time spent on the new landing page is the same across all preferred languages. H_0: $\mu_{Spanish} = \mu_{English} = \mu_{French}$

- Alternate Hypothesis (H_a): At least one language group has a different mean time spent on the new landing page. H_a: $\mu_{Spanish} \neq \mu_{English} \neq \mu_{French}$ (at least one differs)

Step 2: Select Appropriate test

One-Way Analysis of Variance (ANOVA) is the rigth choice because:

- since we are comparing the mean time spent across multiple language groups (more than two groups) and the data is continuous.

Step 3: Decide the significance level

Significance Level (α): alpha = 0.05
I'll use this significance level of 5%, which is standard/common in hypothesis testing.

Interpretation:

- If the p-value from the ANOVA test is less than 0.05, we will reject the null hypothesis (H_0) and conclude that there is a significant difference in the mean time spent across the language groups.

- If the p-value is greater than 0.05, we will fail to reject H_0, indicating insufficient evidence to suggest differences in mean time spent among the language groups.

Step 4: Collect and prepare data

In [102…

```
# Filtering tje data for users on the new landing page
new_page_data = abtest_df[abtest_df['landing_page'] == 'new']

# Creating separate groups for each language
time_english = new_page_data[new_page_data['language_preferred'] == 'English']['time_spent_on_the_page']
```

```
time_spanish = new_page_data[new_page_data['language_preferred'] == 'Spanish']['time_spent_on_the_page']
time_french = new_page_data[new_page_data['language_preferred'] == 'French']['time_spent_on_the_page']

# Displaying summary statistics for each group
print("Summary of Time Spent for English Users:")
print(time_english.describe())

print("\nSummary of Time Spent for Spanish Users:")
print(time_spanish.describe())

print("\nSummary of Time Spent for French Users:")
print(time_french.describe())
```

```
Summary of Time Spent for English Users:
count    16.00000
mean      6.66375
std       1.98415
min       3.65000
25%       5.32750
50%       6.86500
75%       7.71250
max      10.50000
Name: time_spent_on_the_page, dtype: float64

Summary of Time Spent for Spanish Users:
count    17.000000
mean      5.835294
std       1.525656
min       1.650000
25%       5.150000
50%       6.200000
75%       6.700000
max       8.080000
Name: time_spent_on_the_page, dtype: float64

Summary of Time Spent for French Users:
count    17.000000
mean      6.196471
std       1.933394
min       3.300000
25%       5.250000
50%       5.420000
75%       7.270000
max      10.710000
Name: time_spent_on_the_page, dtype: float64
```

Step 5: Calculate the p-value

In [105…

```
# Performing One-Way ANOVA
f_stat, p_value = f_oneway(time_english, time_spanish, time_french)

# Displaying the results
print("F-statistic:", f_stat)
print("P-value:", p_value)
```

```
F-statistic: 0.854399277000682
P-value: 0.43204138694325955
```

Step 6: Compare the p-value with α

- F-statistic: 0.854
- P-value: 0.432
- Significance Level (α): 0.05

Comparison: The p-value (0.432) is greater than the significance level (α = 0.05)

Step 7: Draw inference

Based on the results of the One-Way ANOVA test:

- Since the p-value (0.432) is greater than the significance level ((\alpha = 0.05)), we fail to reject the null hypothesis ((H_0)).
- Meaning there is no statistically significant difference in the mean time spent on the new landing page across the preferred language groups.

Business Insight:

- The time spent on the new landing page is consistent across **Spanish**, **English**, and **French** users.
- This suggests that the new landing page design engages users equally well, regardless of their preferred language.

A similar approach can be followed to answer the other questions.

Conclusion and Business Recommendations

Based on the A/B test analysis, here's the conclusion and recommendations I'll have for the design team of E-news Express:

Conclusion:

The new landing page design is statistically more effective than the old design in achieving the following:

- **Increased User Engagement:** Users spend significantly more time on the new landing page compared to the old one. This suggests the new design is more engaging and keeps visitors interested in the content.
- **Improved Conversion Rate:** The new landing page leads to a significantly higher conversion rate compared to the old one. This translates to a greater number of visitors subscribing to E-news Express.

Business Recommendations:

Given the positive results of the A/B test, E-news Express should strongly consider:

- **Adopting the New Landing Page:** The new design demonstrably improves user engagement and conversion rates, leading to a potentially significant increase in subscriber acquisition.
- **Further Optimization:** While the new landing page performs well, there might be room for further improvement. Consider conducting additional A/B tests with variations of the new design to optimize specific elements or calls to action.
- **Language Analysis:** Although this analysis didn't reveal a significant impact of language on conversion rates, explore if specific languages require adjustments or targeted content within the new landing page for better performance.

Additional Insights:

- The boxplot analysis of time spent on the landing page suggests a wider spread for the old page compared to the new one. This indicates the new design might be more effective at keeping users engaged for a consistent period, potentially leading to a better understanding of the content and a higher conversion rate.
- The observed 16% difference in conversions (calculated during the univariate analysis) aligns with the statistically significant result from the two-proportion z-test. This further strengthens the evidence that the new landing page is driving positive business outcomes.

Overall, the A/B test results are encouraging and suggest that the new landing page design is a significant improvement for E-news Express. Implementing these recommendations will likely lead to increased user engagement, subscriber acquisition, and overall business growth.