E-news Express Project

Problem Statement

An online news portal aims to expand its business by acquiring new subscribers. Every visitor to the website takes certain actions based on their interest. The company plans to analyze these interests and wants to determine whether a new feature will be effective or not. Companies often analyze users' responses to two variants of a product to decide which of the two variants is more effective. This experimental technique is known as a/b testing that is used to determine whether a new feature attracts users based on a chosen metric.

Suppose you are hired as a Data Scientist in E-news Express. The design team of the company has created a new landing page. You have been assigned the task to decide whether the new landing page is more effective to gather new subscribers. Suppose you randomly selected 100 users and divided them equally into two groups. The old landing page is served to the first group (control group) and the new landing page is served to the second group (treatment group). Various data about the customers in both groups are collected in 'abtest.csv'. Perform the statistical analysis to answer the following questions using the collected data.

1. Explore the dataset and extract insights using Exploratory Data Analysis.
2. Do the users spend more time on the new landing page than the existing landing page?
3. 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?
4. Does the converted status depend on the preferred language? [Hint: Create a contingency table using the pandas.crosstab() function]
5. Is the time spent on the new page same for the different language users?

*Consider a significance level of 0.05 for all tests.

The idea behind answering these questions is to decide whether the new page is effective enough to gather new subscribers for the news portal. We will perform the statistical analysis on the collected data to make the business decision.

Data Dictionary

1. user_id - This represents the user ID of the person visiting the website.
2. group - This represents whether the user belongs to the first group (control) or the second group (treatment).
3. landing_page - This represents whether the landing page is new or old.
4. time_spent_on_the_page - This represents the time (in minutes) spent by the user on the landing page.

5. converted - This represents whether the user gets converted to a subscriber of the news portal or not.

6. language_preferred - This represents the language chosen by the user to view the landing page.

In [3]:
```
#import libraries needed for data manipulation

import numpy as np
import pandas as pd

#import libraries needed for data visualization

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# this library contains a large number of probability distributions as well as a growing library of
# statistical functions.
import scipy.stats as stats
```

In [4]:
```
#import possible required functions from scipy.stats

from scipy import stats
from scipy.stats import levene
from statsmodels.stats.multicomp import pairwise_tukeyhsd

from scipy.stats import ttest_ind           # two sample independent t test
from scipy.stats import chi2_contingency    # chi squared test for independence
from scipy.stats import f_oneway            # one way ANOVA test

# 2 proportion z test
from statsmodels.stats.proportion import proportions_ztest
```

In [5]:
```
#import dataset named foodhub_order.csv

df = pd.read_csv('abtest.csv')

#read first five rows of dataset

df.head()
```

Out[5]:

	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

	user_id	group	landing_page	time_spent_on_the_page	converted	language_preferred
4	546459	treatment	new	4.75	yes	Spanish

Question 1: Explore the dataset and extract insights using Exploratory Data Analysis.

In [6]:
```
print('The dataset has dimensions of',  df.shape[0], 'by', df.shape[1], 'meaning 100 entries, i.e. users, and 6 variables.')
```

```
The dataset has dimensions of 100 by 6 meaning 100 entries, i.e. users, and 6 variables.
```

In [7]:
```
# define a function to plot a boxplot and a histogram along the same scale

def histbox(data, feature, figsize=(12, 7), kde=False, bins=None):
    """
    Boxplot and histogram combined

    data: dataframe
    feature: dataframe column
    figsize: size of figure (default (12,7))
    kde: whether to show the density curve (default False)
    bins: number of bins for histogram (default None)
    """
    f2, (box, hist) = plt.subplots(
        nrows=2,                                            # Number of rows of the subplot grid = 2
                                                                # boxplot first then histogram created below
        sharex=True,                                        # x-axis same among all subplots
        gridspec_kw={"height_ratios": (0.25, 0.75)},        # boxplot 1/3 height of histogram
        figsize=figsize,                                    # figsize defined above as (12, 7)
    )
    # defining boxplot inside function, so when using it say histbox(df, 'cost'), df: data and cost: feature

    sns.boxplot(
        data=data, x=feature, ax=box, showmeans=True, color="chocolate"
    )  # showmeans makes mean val on boxplot have star, ax =
    sns.histplot(
        data=data, x=feature, kde=kde, ax=hist, bins=bins, color = "darkgreen"
    ) if bins else sns.histplot(
        data=data, x=feature, kde=kde, ax=hist, color = "darkgreen"
    )  # For histogram if there are bins in potential graph

    # add vertical line in histogram for mean and median
    hist.axvline(
        data[feature].mean(), color="purple", linestyle="--"
    )  # Add mean to the histogram
    hist.axvline(
        data[feature].median(), color="black", linestyle="-"
    )  # Add median to the histogram
```

In [8]:
```
# define a function to create labeled barplots
```

```
def bar(data, feature, perc=False, n=None):
    """
    Barplot with percentage at the top

    data: dataframe
    feature: dataframe column
    perc: whether to display percentages instead of count (default is False)
    n: displays the top n category levels (default is None, i.e., display all levels)
    """

    total = len(data[feature])  # length of the column
    count = data[feature].nunique()
    if n is None:
        plt.figure(figsize=(count + 1, 5))
    else:
        plt.figure(figsize=(n + 1, 5))

    plt.xticks(rotation=90, fontsize=15)
    ax = sns.countplot(
        data=data,
        x=feature,
        palette="Paired",
        order=data[feature].value_counts().index[:n].sort_values(),
    )

    for p in ax.patches:
        if perc == True:
            label = "{:.1f}%".format(
                100 * p.get_height() / total
            )  # percentage of each class of the category
        else:
            label = p.get_height()  # count of each level of the category

        x = p.get_x() + p.get_width() / 2  # width of the plot
        y = p.get_height()  # height of the plot

        ax.annotate(
            label,
            (x, y),
            ha="center",
            va="center",
            size=12,
            xytext=(0, 5),
            textcoords="offset points",
        )  # annotate the percentage

    plt.show()  # show the plot
```

In [56]:

```
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
```

In [23]:
```
df['user_id'].value_counts().shape
```

Out[23]: (100,)

Observations:

- There are total 100 non-null observations in each of the columns. No missing values.
- User_ID seems to be just an identifier variable.
- There are 6 columns named **'user_id', 'group', 'landing_page', 'time_spent_on_the_page', 'converted', 'language_preferred'** whose data types are **int64, object, object, float64, object, object** respectively.
- 'group', 'landing_page', 'converted', and 'language_preferred' are objects, we can change them to categories.

In [12]:
```
# Convert group, landing_page, converted, and language_preferred variables to category to reduce memory usage.

df['group'] = df['group'].astype('category')
df['landing_page'] = df['landing_page'].astype('category')
df['converted'] = df['converted'].astype('category')
df['language_preferred'] = df['language_preferred'].astype('category')
```

In [13]:
```
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    category
 2   landing_page            100 non-null    category
 3   time_spent_on_the_page  100 non-null    float64 
 4   converted               100 non-null    category
 5   language_preferred      100 non-null    category
dtypes: category(4), float64(1), int64(1)
memory usage: 2.6 KB
```

In [18]:
```
# Analyze numeric variable

df.describe(include=['float64']).T
```

Out[18]:

	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

Observations:

- The time spent on the landing page is less than 11 minutes
- The mean time spent on the landing page is approximately 5 minutes.
- The median time spent on the landing page is approximately 5 minutes.

In [19]:
```
# Analyze categorical variables

df.describe(include = ['category']).T
```

Out[19]:

	count	unique	top	freq
group	100	2	control	50
landing_page	100	2	new	50
converted	100	2	yes	54
language_preferred	100	3	French	34

In [15]:
```
df.groupby(by = ["group"])['landing_page'].value_counts()
```

Out[15]:
```
group      landing_page
control    old             50
treatment  new             50
Name: landing_page, dtype: int64
```

In [59]:
```
df.groupby(by = ["group"])['time_spent_on_the_page'].mean()
```

Out[59]:
```
group
control      4.5324
treatment    6.2232
Name: time_spent_on_the_page, dtype: float64
```

In [60]:
```
df.groupby(by = ["group"])['time_spent_on_the_page'].sum()
```

Out[60]:
```
group
control      226.62
treatment    311.16
Name: time_spent_on_the_page, dtype: float64
```

In [62]:
```
df.groupby(by = ["group"])['converted'].value_counts()
```

Out[62]:
```
group      converted
control    no           29
           yes          21
treatment  yes          33
           no           17
Name: converted, dtype: int64
```

In [63]:
```
df.groupby(by = ["landing_page"])['converted'].value_counts()
```

Out[63]:
```
landing_page  converted
new           yes          33
              no           17
old           no           29
              yes          21
Name: converted, dtype: int64
```

In [66]:
```
df['language_preferred'].value_counts()
```

Out[66]:
```
French     34
Spanish    34
English    32
Name: language_preferred, dtype: int64
```

Observations:

- There are 2 unique groups - control and treatment. Each group consists of 50 users.
- **There are 2 landing_pages - old (corresponding to the control group) and new (corresponding to the treatment group).**
- The control group has a lower mean and cumulative time spent on the page compared to the treatment group.
- After surfing the old landing page, 42.0 % converted to a subscriber, versus 66.0 % for those who experienced the new landing page.
- There are 3 unique preferred languages - English, French, and Spanish.

Univariate Analysis

In [24]:
```
# Split dataset into two groups: old landing page (control group) and new landing page (treatment group)

old = df[df["landing_page"] == "old" ]
new = df[df["landing_page"] == "new" ]
```

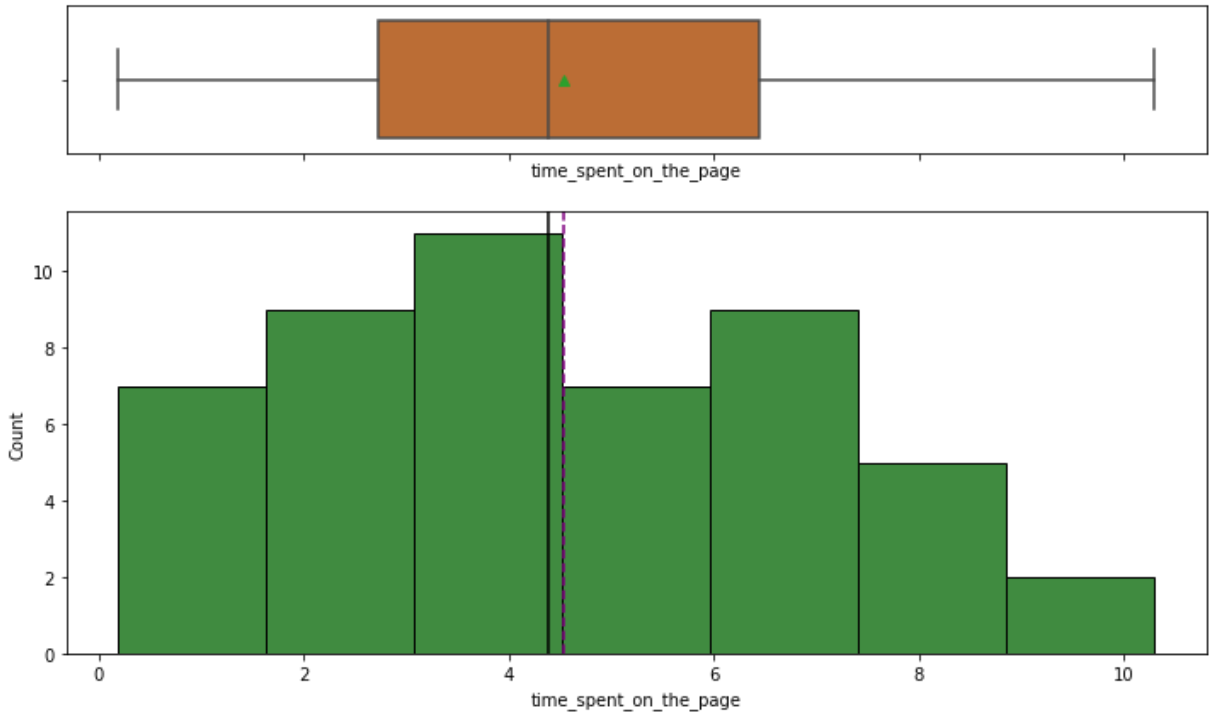
In [69]:
```
# Only numeric variable is time. Summary statistics for old landing page "time spent":

old.describe(include=['float64'])
```

Out[69]:

	time_spent_on_the_page
count	50.000000

	time_spent_on_the_page
mean	4.532400
std	2.581975
min	0.190000
25%	2.720000
50%	4.380000
75%	6.442500
max	10.300000

In [70]:
```
histbox(old, 'time_spent_on_the_page')
```



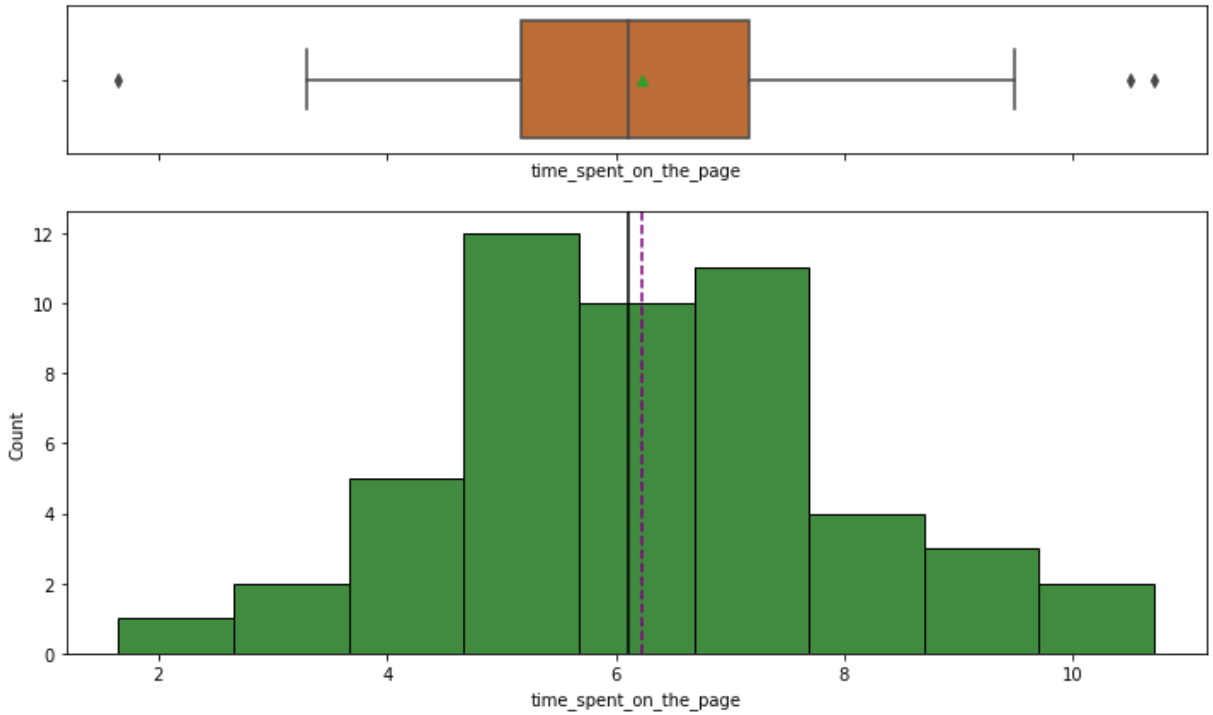
In [71]:
```
# Summary statistics for new landing page "time spent":

new.describe(include=['float64'])
```

Out[71]:

	time_spent_on_the_page
count	50.000000

	time_spent_on_the_page
mean	6.223200
std	1.817031
min	1.650000
25%	5.175000
50%	6.105000
75%	7.160000
max	10.710000

In [72]:
```
histbox(new, 'time_spent_on_the_page')
```

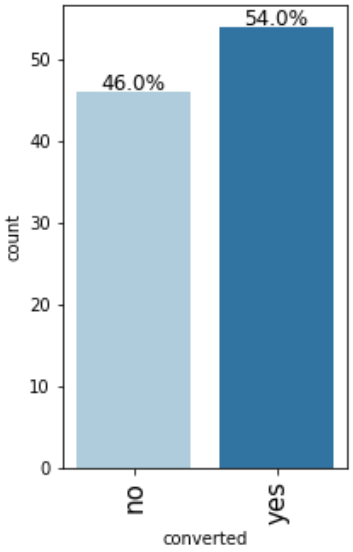


Observations:

- Old landing page: The mean is greater than the median, indicating a slight right-skew. The majority of the users have time spent in the 4-5 minute mark.
- New landing page: The mean is greater than the median, indicating a slight right-skew. The majority of the users have time spent in the 5-7 minute mark. There are a few outliers.

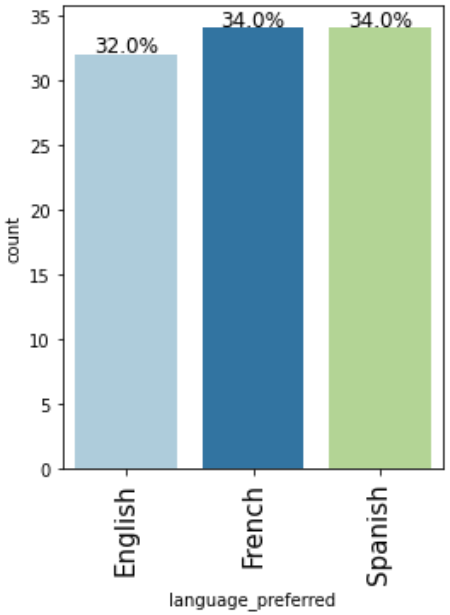
- Both the old and new landing pages have a close to normal distribution for time spent on the pages.
- Overall, the time spent on the new page seems to be greater than the time spent on the new page.

In [25]:
```
bar(df,'converted',perc=True)
```



- Overall, 54% of the users get converted after visiting the landing page.
- Overall, 46% of the users do not get converted after visiting the landing page.

In [26]:
```
bar(df,'language_preferred',perc=True)
```

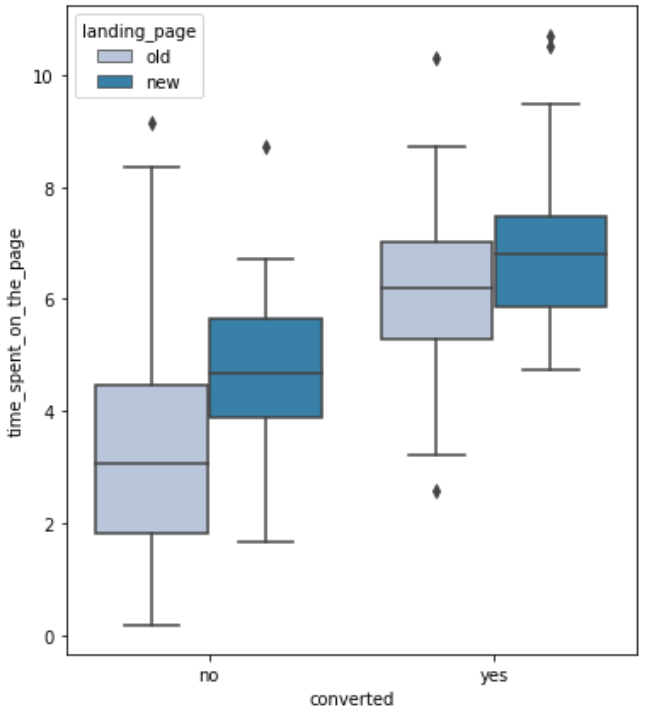


- The distribution of observations across various preferred languages is fairly uniform.

Bivariate Analysis

In [74]:
```
'''Relationship between time spent on page to whether the user converted to a subscriber or not, comparing and
contrasting between the new and old landing page.'''

plt.figure(figsize=(6,7))
sns.boxplot(x = "converted", y = "time_spent_on_the_page", data = df, palette = 'PuBu', hue="landing_page")
```

Out[74]:
```
<AxesSubplot:xlabel='converted', ylabel='time_spent_on_the_page'>
```

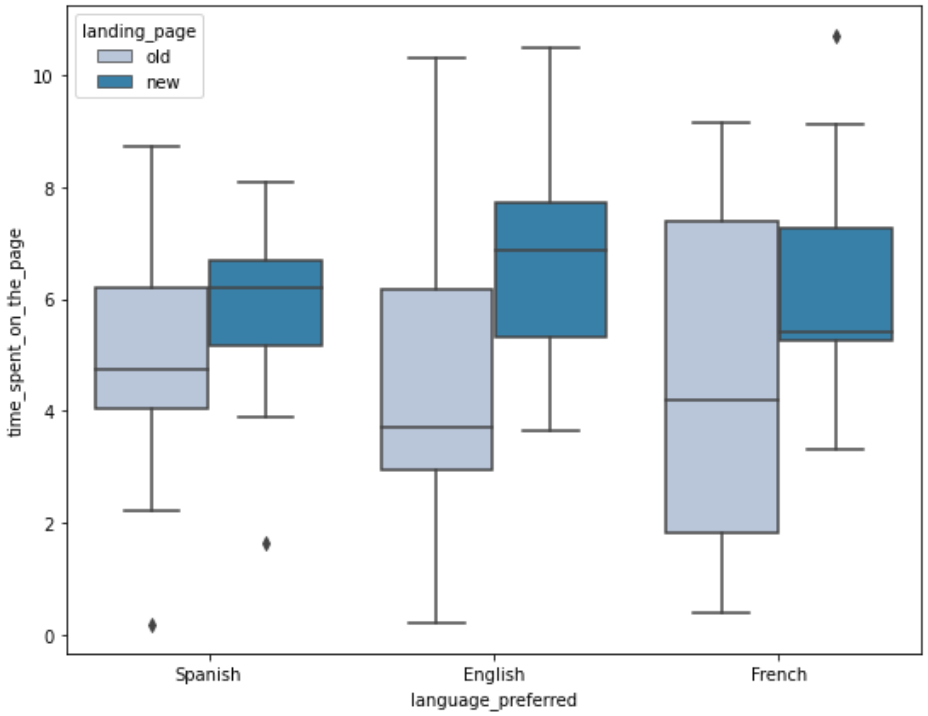


Observations:

- The users that were converted spent significantly more time on both versions of the pages than those who were not with smaller ranges.
- Even with those who were not converted, the new landing page had a larger mean of time spent than the old landing page.
- The new landing page also had more positive outliers.

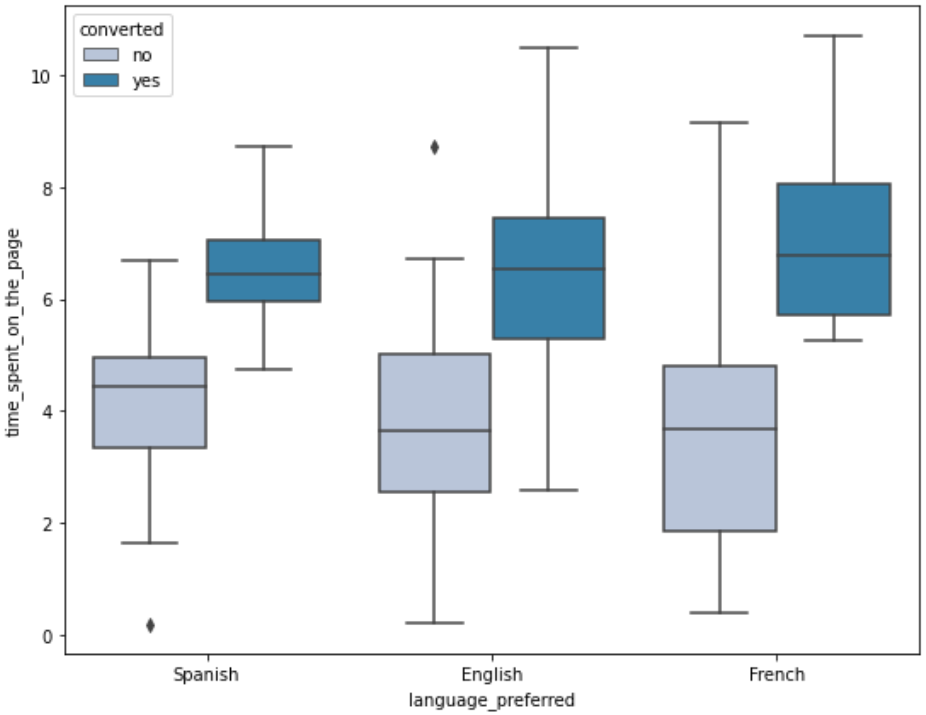
In [76]:
```
# Relationship between time spent on the page to language preferred, comparing/contrasting the new/old landing page.

plt.figure(figsize=(9,7))
sns.boxplot(x = "language_preferred", y = "time_spent_on_the_page", data = df, palette = 'PuBu', hue="landing_page")
```

Out[76]:
```
<AxesSubplot:xlabel='language_preferred', ylabel='time_spent_on_the_page'>
```



In [77]:
```
# Relationship between time spent on the page to language preferred, comparing/contrasting converted/not converted.

plt.figure(figsize=(9,7))
sns.boxplot(x = "language_preferred", y = "time_spent_on_the_page", data = df, palette = 'PuBu', hue="converted")
```

Out[77]:
```
<AxesSubplot:xlabel='language_preferred', ylabel='time_spent_on_the_page'>
```

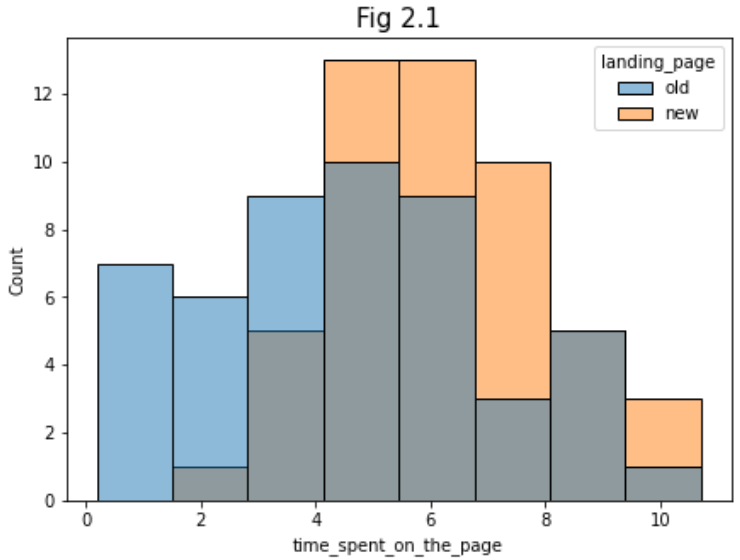


Observations:

- When comparing converted/not converted:
  - Looking at whether users converted, we can see for all 3 languages, the users spent significantly more time on the landing page if they did.
- When comparing new/old landing pages:
  - The "time spent" ranges and means for all three languages is significantly higher for the new landing page.
  - English users spent the most time on the new landing page, and French users spent the most on the old page.
  - French users had the most variation within their ranges for the old landing page, indicated by the IQR taking up a majority of the boxplot. There was a medium variation in the new landing page, with most users time spent recorded above the mean.

Question 2: Do the users spend more time on the new landing page than the old landing page?

In [79]:
```
plt.figure(figsize=(7,5))
a = sns.histplot(data = df, x = 'time_spent_on_the_page', hue="landing_page")
a.set_title("Fig 2.1", fontsize=15)
plt.show()

# Fig 2.1 Observations/Insights detailed after 2 sample T-test.
```



- By observing the above plot, we can say that overall people spent more times on the new page than the old age. Let's perform a hypothesis test to see if there are enough statistical evidence to support our observation.

Step 1: Define the null and alternate hypotheses:

H_0 : The mean time spent by the users on the new page is equal to the mean time spent by the users on the old page.

H_a : The mean time spent by the users on the new page is greater than the mean time spent by the users on the old page.

Let μ_1 and μ_2 be the mean time spent by the users on the new and old page respectively.

Mathematically, the above formulated hypotheses can be written as:

$H_0 : \mu_1 = \mu_2$

$H_a : \mu_1 > \mu_2$

Step 2: Select Appropriate Test:

This is a one-tailed test concerning two population means from two independent populations. As the population standard deviations are unknown, the two sample independent t-test will be the appropriate test for this problem.

Step 3: Decide the Significance Level:

As given in the problem statement, we select α = 0.05.

Step 4: Collect and prepare data:

In [28]:
```
print('The mean time spent on the page for control group is', round(old['time_spent_on_the_page'].mean(),2))
print('The mean time spent on the page for treatment group is', round(old['time_spent_on_the_page'].std(),2))

print('The standard deviation of time spent on the page for control group is', round(new['time_spent_on_the_page'].mean(),2))
print('The standard deviation of time spent on the page for treatment group is', round(new['time_spent_on_the_page'].std(),2))
```

```
The mean time spent on the page for control group is 4.53
The mean time spent on the page for treatment group is 2.58
The standard deviation of time spent on the page for control group is 6.22
The standard deviation of time spent on the page for treatment group is 1.82
```

As the sample standard deviations are different, the population standard deviations may be assumed to be different.

Let's test whether the T-test assumptions are satisfied or not:

- Continuous data - Yes, the time spent on the page is measured on a continuous scale.
- Normally distributed populations - Yes, from univariate analysis the distributions are close to normal.
- Independent populations - Our samples are from two different groups from two independent populations - the new landing page was the treatment group and the old landing page was the control group.
- Unequal population standard deviations - As the sample standard deviations are different, the population standard deviations may be assumed to be different.
- Random sampling from the population - Yes, it is given to us that the collected sample is random.

All conditions satisfied! We can use two sample T-test (one tailed) for this problem.

Step 5: Find the p-value:

In [34]:
```
# find the p-value
test_stat, p_value = ttest_ind(new['time_spent_on_the_page'], old['time_spent_on_the_page'], equal_var = False, alternative = 'greate
print('The p-value is', p_value)
```

```
The p-value is 0.0001392381225166549
```

Step 6: Compare the p-value with α:

In [35]:
```
# print the conclusion based on p-value
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject the null hypothesis.')
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail to reject the null hypothesis.')
```

```
As the p-value 0.0001392381225166549 is less than the level of significance, we reject the null hypothesis.
```

Step 7: Draw inference:

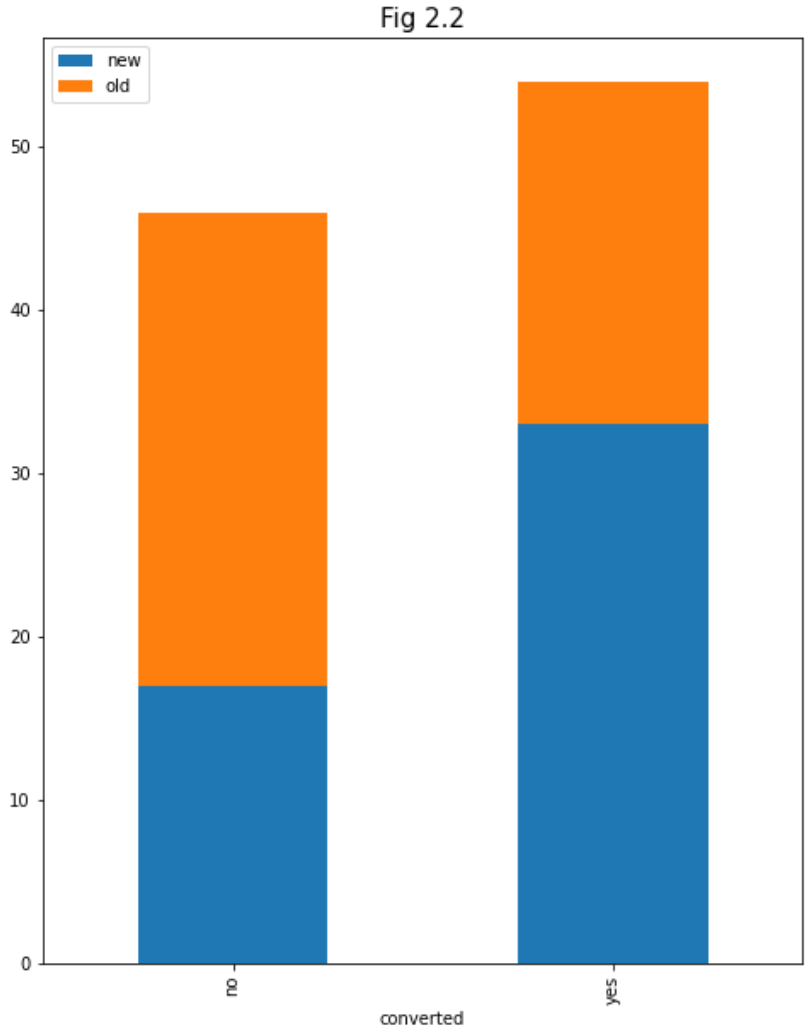
- From Fig 2.1:
  - Initial impression is that the mean time spent on the new landing page is higher.
  - There are very few users that are in the 9-10 minute bracket for both versions of the landing page.
  - The old landing page has relatively more users spending between 0-4 minutes on it.
- From 2 sample T-test:

Since the p-value is less than the 5% significance level, we reject the null hypothesis. Hence, we have enough statistical evidence to say that the mean time spent by the users on the new page is greater than the mean time spent by the users on the old page.

Question 3: Is the conversion rate (proportion of users who visit the landing page and get converted) for the new page greater than the conversion rate for the old page?

In [37]:
```
plt.figure(figsize=(4,6))
a= pd.crosstab(df['converted'],df['landing_page']).plot(kind="bar", figsize=(8,10),
                 stacked=True)
a.set_title("Fig 2.2", fontsize=15)
plt.legend()
plt.show()

# Fig 2.2 Observations/Insights detailed after 2 proportion Z-test.
```

```
<Figure size 288x432 with 0 Axes>
```



In [38]:
```
df.groupby(by = ["converted"])['landing_page'].value_counts()
```

Out[38]:
```
converted  landing_page
no         old             29
           new             17
yes        new             33
           old             21
Name: landing_page, dtype: int64
```

By observing the above plot, we can say that overall the number of users who get converted is more for the new page than the old page. Let's perform a hypothesis test to see if there is enough statistical evidence to say that the conversion rate for the new page is greater than the old page.

Step 1: Define the null and alternate hypotheses:

H_0 : The conversion rate for the new page is equal to the conversion rate for the old page.

H_a : The conversion rate for the new page is greater than the conversion rate for the old page.

Let p_1 and p_2 be the conversion rate for the new and old page respectively.

Mathematically, the above formulated hypotheses can be written as:

$H_0 : p_1 = p_2$

$H_a : p_1 > p_2$

Step 2: Select Appropriate Test:

This is a one-tailed test concerning two population proportions from two independent populations. Hence, the two proportion z-test will be the appropriate test for this problem.

Step 3: Decide the Significance Level:

As given in the problem statement, we select α = 0.05.

Step 4: Collect and prepare data:

In [41]:
```
new_converted = df[df['group'] == 'treatment']['converted'].value_counts()['yes']
old_converted = df[df['group'] == 'control']['converted'].value_counts()['yes']
print('The numbers of converted users for the new and old pages are {0} and {1} respectively'.format(new_converted, old_converted))
n_control = df.group.value_counts()['control'] # number of users in the control group
n_treatment = df.group.value_counts()['treatment'] #number of users in the treatment group
print('The numbers of users served the new and old pages are {0} and {1} respectively'.format(n_control, n_treatment ))
```

```
The numbers of converted users for the new and old pages are 33 and 21 respectively
The numbers of users served the new and old pages are 50 and 50 respectively
```

Let's test whether the Z-test assumptions are satisfied or not:

- Binomally distributed population - Yes, a user is either converted or not converted.
- Random sampling from the population - Yes, it is given to us that the collected sample is random.
- Can this binmonial be approximated to normal distribution? - Yes. For binary, CLT works slower. Standard is to check if n*p and n*(1-p) are greater than or equal to 10, where n is sample size and p is sample proportion.
  - n *p_old = 50* (21/50) = 21 > 10 ✓
  - n *p_new = 50* (33/50) = 33 > 10 ✓
  - n *(1-p_old) = 50* ((50-21)/50) = 29 > 10 ✓
  - n *(1-p_old) = 50* ((50-33)/50) = 17 > 10 ✓

All conditions satisfied! We can use two proportion Z-test (one tailed) for this problem.

Step 5: Calculate the p-value:

In [44]:
```
# set the counts of converted users for new and old landing pages respectively
converted = np.array([33, 21])

# set the sample sizes
sample_sizes = np.array([50, 50])

# find the p-value
test_stat, p_value = proportions_ztest(converted, sample_sizes, alternative = 'larger')
print('The p-value is', p_value)
```

```
The p-value is 0.008026308204056278
```

Step 6: Compare the p-value with α:

In [45]:
```
# print the conclusion based on p-value
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject the null hypothesis.')
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail to reject the null hypothesis.')
```

```
As the p-value 0.008026308204056278 is less than the level of significance, we reject the null hypothesis.
```

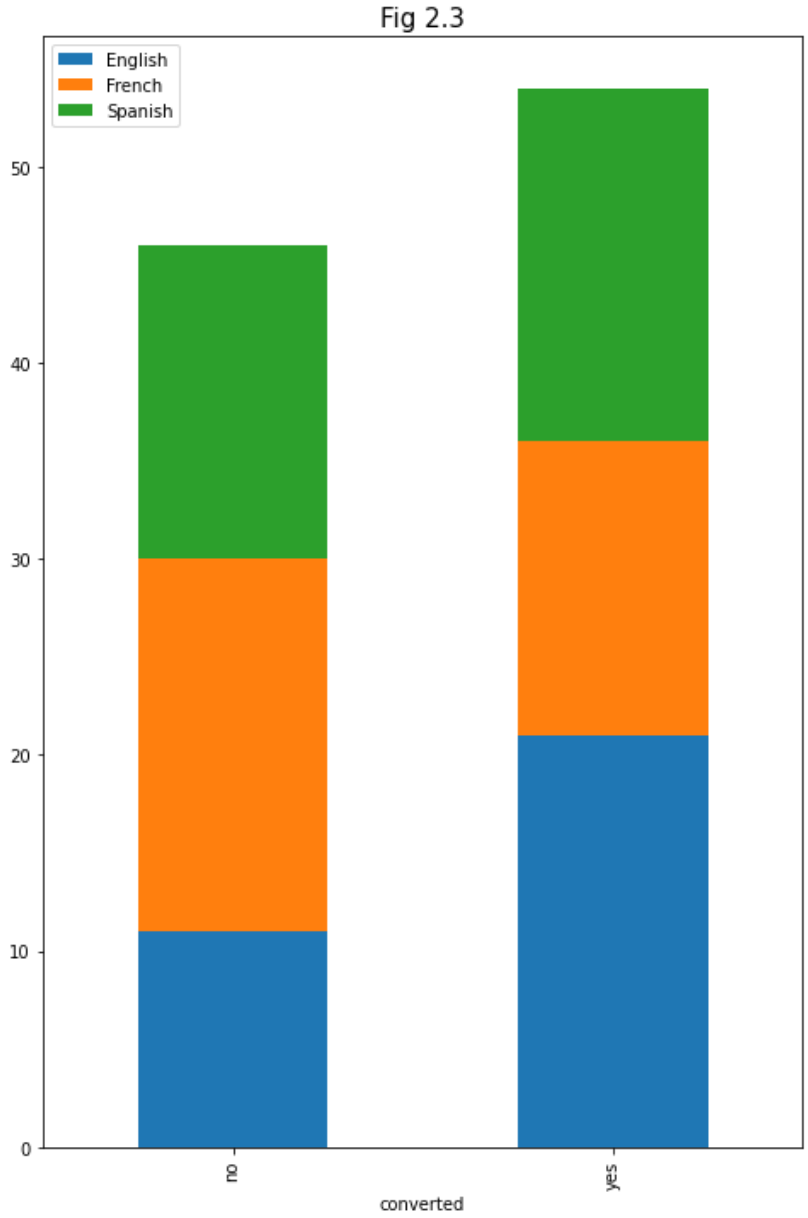
Step 7: Draw inference:

- From Fig 2.2:
  - Initial impression is that the proportion of users who converted after viewing the landing page is higher for the new landing page versus the old one.
- From 2 proportion Z-Test:
  - Since the p-value is less than the 5% significance level, we reject the null hypothesis. Hence, we have enough statistical evidence to say that the conversion rate for the new page is greater than the conversion rate for the old page.

Question 4: Does the converted status depend on the preferred language?

In [46]:
```
# visual analysis of the dependency between conversion status and preferred langauge
plt.figure(figsize=(6,7))
a = pd.crosstab(df['converted'],df['language_preferred']).plot(kind="bar", figsize=(8,12),
                 stacked=True)
a.set_title("Fig 2.3", fontsize=15)
plt.legend()
plt.show()

# Fig 2.3 Observations/Insights detailed after Chi-Square Test for Independence.
```

```
<Figure size 432x504 with 0 Axes>
```



The distribution of conversion status for English and French language users is not uniformly distributed. Let's perform the hypothesis test to check whether we have enough statistical evidence to say that the conversion status and preferred language are independent or not.

Step 1: Define the null and alternate hypotheses:

H_0 : The converted status is independent of the preferred language.

H_a : The converted status is not independent of the preferred language.

Step 2: Select Appropriate test:

This is a problem of Chi-square test of independence, concerning the two independent categorical variables, converted status and preferred language.

Step 3: Decide the Significance Level:

As given in the problem statement, we select α = 0.05.

Step 4: Collect and prepare data:

In [50]:
```
# create the contingency table showing the distribution of language against conversion status

Q4 = pd.crosstab(df['converted'], df['language_preferred'])
Q4
```

Out[50]:

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

Let's test whether Chi-Square-test for independence assumptions are satisfied:

- Categorical variables - Yes
- Expected value of the number of sample observations in each level of the variable is at least 5 - Yes, the number of observations in each level is greater than 5.
- Random sampling from the population - Yes, it is given to us that the collected sample is random.

All conditions satisfied! We can use a chi-squared test of independence for this problem.

Step 5: Calculate the p-value:

In [52]:
```
# find the p-value

chi2, p_value, dof, exp_freq = chi2_contingency(Q4)
print('The p-value is', p_value)
```

```
The p-value is 0.21298887487543447
```

Step 6: Compare the p-value with α:

In [53]:
```
# print the conclusion based on p-value
```

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject the null hypothesis.')
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail to reject the null hypothesis.')
```

```
As the p-value 0.21298887487543447 is greater than the level of significance, we fail to reject the null hypothesis.
```

Step 7: Draw inference:

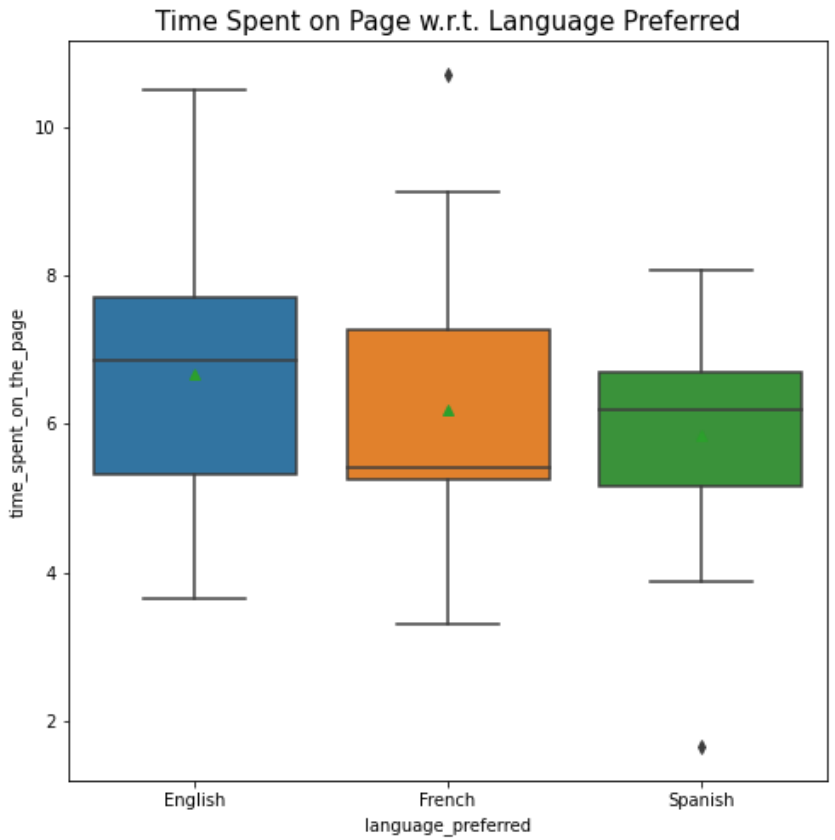
- From Fig 2.3:
  - Initial impression is that English has the highest proportion of users converted, and French has the lowest.
- From Chi-Square Test for Independence:
  - Since the p-value is greater than the 5% significance level, we fail to reject the null hypothesis. Hence, we do not have enough statistical evidence to say that the converted status depends on the preferred language.

Question 5: Is the mean time spent on the new page the same for the different language users?

In [54]:
```
df.groupby('language_preferred')['time_spent_on_the_page'].mean()
```

Out[54]:
```
language_preferred
English    5.559063
French     5.253235
Spanish    5.331765
Name: time_spent_on_the_page, dtype: float64
```

In [56]:
```
# Use dataframe we created above only for new landing page

plt.figure(figsize=(8,8))
a = sns.boxplot(x = 'language_preferred', y = 'time_spent_on_the_page', showmeans = True, data = new)
a.set_title("Time Spent on Page w.r.t. Language Preferred", fontsize=15)
plt.show()
```



The mean time spent on the new page by English users is a bit higher than the mean time spent by French and Spanish users, but we need to test if this difference is statistically significant or not.

Step 1: Define the null and alternate hypotheses:

H_0 : The mean times spent on the new page by English, French, and Spanish users are equal (μ_English = μ_Spanish = μ_French)

H_a : At least one of the mean times spent on the new page by English, French, and Spanish users is unequal.

Step 2: Select Appropriate test:

This is a problem, concerning three population means. One-way ANOVA could be the appropriate test here provided normality and equality of variance assumptions are verified.

- For testing of normality, Shapiro-Wilk's test is applied to the response variable.
- For equality of variance, Levene test is applied to the response variable.

Shapiro-Wilk's test

We will test the null hypothesis

> H_0 : Time spent on the new page follows a normal distribution

against the alternative hypothesis

> H_a : Time spent on the new page does not follow a normal distribution

In [57]:
```
# Assumption 1: Normality (Shapiro-Wilk's Test applied to response variable)

# find the p-value
w, p_value = stats.shapiro(new['time_spent_on_the_page'])
print('The p-value is', p_value)
```

```
The p-value is 0.8040016293525696
```

Since p-value of the test is very large than the 5% significance level, we fail to reject the null hypothesis that the response follows the normal distribution. Thus, we do not have enough statistical evidence to say that the time spent on the page doesn't follow a normal distribution.

Levene's test

We will test the null hypothesis

> H_0: All the population variances are equal

against the alternative hypothesis

> H_a: At least one variance is different from the rest

In [58]:
```
# Assumption 2: Homogeneity of Variance (Levene's Test applied to response variable)

# find the p-value
statistic, p_value = levene( new['time_spent_on_the_page'][new['language_preferred']=="English"],
                             new['time_spent_on_the_page'][new['language_preferred']=="Spanish"],
                             new['time_spent_on_the_page'][new['language_preferred']=="French"])
print('The p-value is', p_value)
```

```
The p-value is 0.46711357711340173
```

Since the p-value is large than the 5% significance level, we fail to reject the null hypothesis of homogeneity of variances. Thus, we do not have enough statistical evidence to say that the variances are not all equal.

Check if assumptions are satisfied:

- Normal distribution for populations - Yes, the normality assumption is verified using the Shapiro-Wilk's test.
- Samples independent and random - Yes, it is given to us that the collected sample is a simple random sample.

- Population variances are equal - Yes, the homogeneity of variance assumption is verified using the Levene's test.

Step 3: Decide the Significance Level:

As given in the problem statement, we select α = 0.05.

Step 4: Collect and prepare data:

In [59]:
```
English = new[new['language_preferred']=="English"]['time_spent_on_the_page']
French = new[new['language_preferred']=="French"]['time_spent_on_the_page']
Spanish = new[new['language_preferred']=="Spanish"]['time_spent_on_the_page']
```

Step 5: Calculate the p-value:

In [60]:
```
# perform one-way anova test

test_stat, p_value = f_oneway(English, French, Spanish)
print('The p-value is ', p_value)
```

```
The p-value is  0.43204138694325955
```

Step 6: Compare the p-value with α:

In [61]:
```
# print the conclusion based on p-value
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject the null hypothesis.')
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail to reject the null hypothesis.')
```

```
As the p-value 0.43204138694325955 is greater than the level of significance, we fail to reject the null hypothesis.
```

Step 7: Draw inference:

- From "Time Spent on Page w.r.t. Language Preferred" boxplot:
  - Spanish users seem to spend the least time on the page, and French users seem to spend the most time.
- From Chi-squared test for independence:
  - Since the p-value is greater than the 5% significance level, we fail to reject the null hypothesis. Hence, we do not have enough statistical evidence to say that the mean times spent on the new page by English, French, and Spanish users differ to any meaningful degree.

For a cross check, a multiple pairwise comparison tests which (if any) means are different from the rest.

In [62]:
```
# perform multiple pairwise comparison (Tukey HSD)
m_comp = pairwise_tukeyhsd(endog = new['time_spent_on_the_page'], groups = new['language_preferred'], alpha = 0.05)
print(m_comp)
```

```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
====================================================
 group1  group2 meandiff p-adj   lower  upper  reject
----------------------------------------------------
English  French  -0.4673 0.7259 -2.0035  1.069  False
English Spanish  -0.8285  0.401 -2.3647 0.7078  False
 French Spanish  -0.3612  0.816  -1.874 1.1516  False
----------------------------------------------------
```

Observations:

- The p-adj column refers to the p-value. In this case, they are all above the significance level, meaning we must fail to reject the null hypothesis that all the means for time spent for different languages are the same.
- Thus, we can say that the mean time spent on the page for different languages are all similar.

Conclusions/Key Insights:

- The users spend more time on the new page.
- The conversion rate for the new page is greater than the conversion rate of the old page.
- The conversion status is independent of the preferred language.
- Based on the conclusions of the hypothesis tests, you can recommend that the news company should use the new landing page to gather more subscribers.
- The longer a visitor spends on a site, the more likely they are to convert.

Business Recommendations

- Overall, Enews Express made a good decision updating the landing page for their online news portal.
- Our sample of users and further hypotheses we conducted leads us to the conclusion that users will be more likely to convert to subscribers of the portal with the new landing page.
- Further recommendations to expand their business include:
  - Expand the languages offered to view the online news portal.
  - Make the Subscribe button easy to find on the home page.
  - Once subscribed, to keep users subscribed, provide incentives like newsletters, discounts for holidays, etc.
  - For the smartphone/mobile version of the news portal, be sure to adjust the design for the screen size.
  - Include hyperlinks within a certain story if it references other articles/newsletters.