

# Project Business Statistics: E-news Express

## Define Problem Statement and Objectives

### Import all the necessary libraries

In [ ]:

```
# Installing the libraries with the specified version.
!pip install numpy==1.25.2 pandas==1.5.3 matplotlib==3.7.1 seaborn==0.13.1 scipy==1.11.4
```

Requirement already satisfied: numpy==1.25.2 in /root/.local/lib/python3.11/site-packages (1.25.2)  
Requirement already satisfied: pandas==1.5.3 in /root/.local/lib/python3.11/site-packages (1.5.3)  
Requirement already satisfied: matplotlib==3.7.1 in /root/.local/lib/python3.11/site-packages (3.7.1)  
Requirement already satisfied: seaborn==0.13.1 in /root/.local/lib/python3.11/site-packages (0.13.1)  
Requirement already satisfied: scipy==1.11.4 in /root/.local/lib/python3.11/site-packages (1.11.4)  
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.11/dist-packages (from pandas==1.5.3) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas==1.5.3) (2025.1)  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (1.3.1)  
Requirement already satisfied: cyclor>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (0.12.1)  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (4.56.0)  
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (1.4.8)  
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (24.2)  
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (11.1.0)  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.7.1) (3.2.1)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.1->pandas==1.5.3) (1.17.0)

**Note:** After running the above cell, kindly restart the notebook kernel and run all cells sequentially from the start again.

In [ ]:

```
#importing essential libraries
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [ ]:

```
# Mounting the file path
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

## Reading the Data into a DataFrame

In [ ]:

```
#Reading the dataset
df=pd.read_csv('/content/drive/My Drive/Google_colab/abtest.csv')
```

## Explore the dataset and extract insights using Exploratory Data Analysis

- Data Overview
  - Viewing the first and last few rows of the dataset
  - Checking the shape of the dataset
  - Getting the statistical summary for the variables
- Check for missing values
- Check for duplicates

In [ ]:

```
#Viewing the first and last few rows of the dataset
df.head()
```

Out[ ]:

	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

The User\_id the unique id given for the users viewing the news portal There are two groups named 'control' and 'treatment'. There are two landing pages named 'old' and 'new'. The time\_Spent\_on\_the\_page is the time in minutes spent by the users. The converted column has two values 'no' and 'yes'. This states whether the user got converted as subscriber or not. The language preferred shows the languages prepared by the users.

In [ ]:

```
df.tail()
```

Out[ ]:

	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

This shows there are 100 rows in the dataset.

In [ ]:

```
#To find the number of rows and columns in the dataset
df.shape
```

Out[ ]:

```
(100, 6)
```

There are 100 rows and 6 columns in the dataset

In [ ]:

```
# To find the statistical information about the dataset
df.describe(include='all')
```

Out[ ]:

	user_id	group	landing_page	time_spent_on_the_page	converted	language_preferred
<b>count</b>	100.000000	100	100	100.000000	100	100
<b>unique</b>	NaN	2	2	NaN	2	3
<b>top</b>	NaN	control	old	NaN	yes	Spanish
<b>freq</b>	NaN	50	50	NaN	54	34
<b>mean</b>	546517.000000	NaN	NaN	5.377800	NaN	NaN
<b>std</b>	52.295779	NaN	NaN	2.378166	NaN	NaN
<b>min</b>	546443.000000	NaN	NaN	0.190000	NaN	NaN
<b>25%</b>	546467.750000	NaN	NaN	3.880000	NaN	NaN
<b>50%</b>	546492.500000	NaN	NaN	5.415000	NaN	NaN
<b>75%</b>	546567.250000	NaN	NaN	7.022500	NaN	NaN
<b>max</b>	546592.000000	NaN	NaN	10.710000	NaN	NaN

This shows there are no missing values in the dataset as total rows and the count of all rows in the columns are 100. There are only two unique values in group, landing\_page and converted column. We have 50 rows that has 'control' as group and 'old' as landing\_page. 54 Users got converted to subscribers. 34 users gave 'Spanish' as the preferred language. 'Spanish' is the most preferred language. Average time spent by the users in the website is found to be approximately 5 minutes 37 seconds. Maximum time spent in the portal is approximately around 10 minutes 71 seconds.

In [ ]:

```
#To find the datatype of the columns  
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
```

This shows there are no missing values in the dataset. User\_id is of integer datatype and time\_spent\_on\_the\_page is in float datatype. Other columns belong to string datatype

```
In [ ]:
```

```
#To find duplicate rows  
df.duplicated().sum()
```

```
Out[ ]:
```

```
0
```

There are no duplicate rows in the dataset

```
In [ ]:
```

```
#Finding unique value count of user_id column  
df['user_id'].nunique()
```

```
Out[ ]:
```

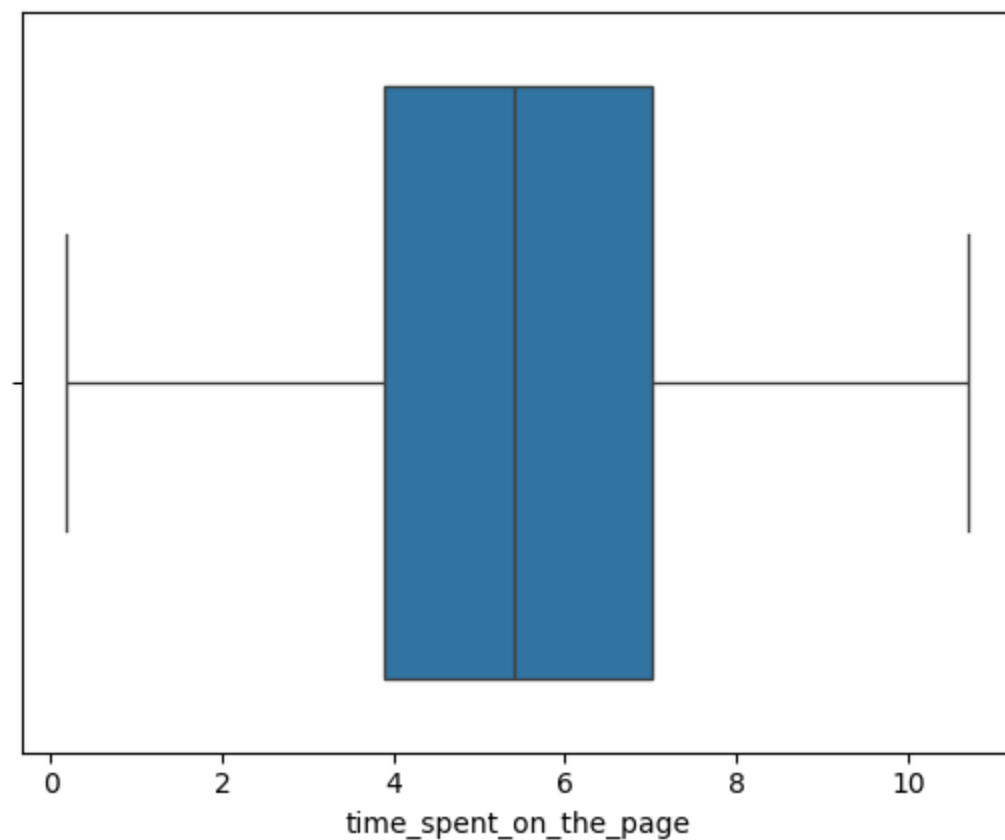
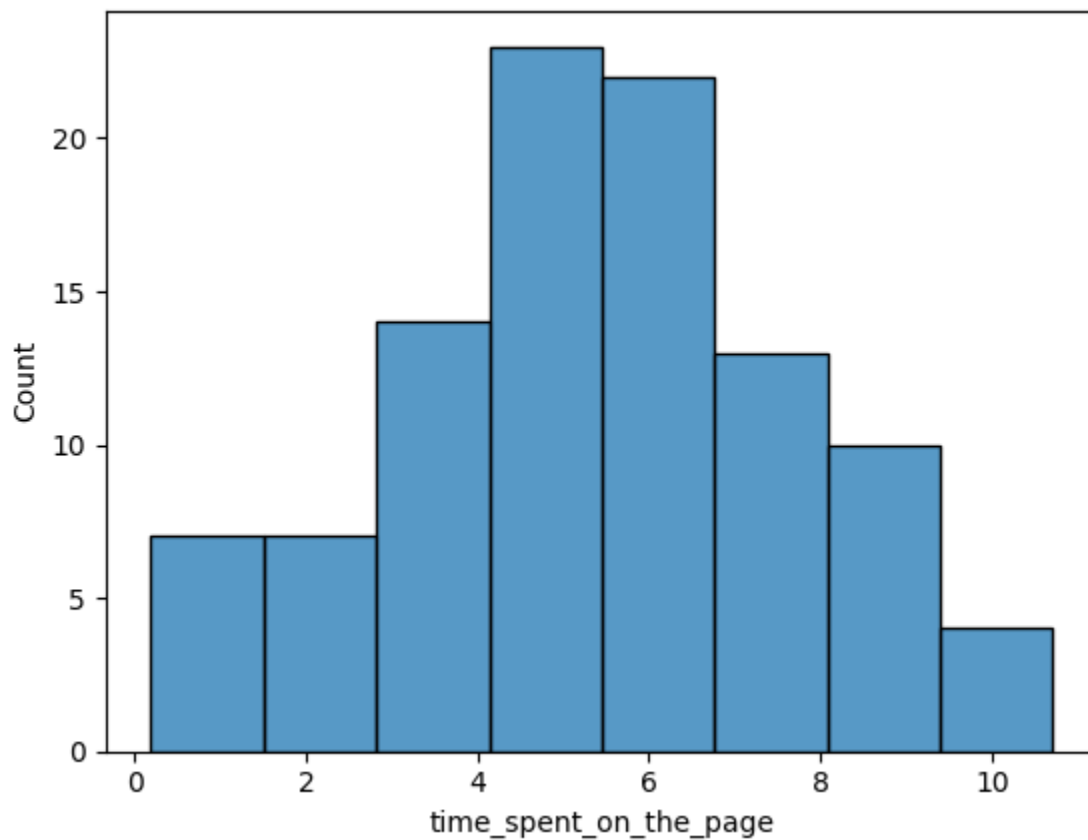
```
100
```

There are no duplicates in the user\_id column

## Univariate Analysis

```
In [ ]:
```

```
#Visualizing time_spent_on_the_page  
sns.histplot(data=df, x='time_spent_on_the_page')  
plt.show()  
sns.boxplot(data=df, x='time_spent_on_the_page')  
plt.show()
```



There are no outliers in the dataset .The median falls around 5 and 6 minutes .More then 20 users spend approximately around 5 minutes to 6 minutes in the portal.

In [ ]:

```
#To find the value_counts of column 'group'  
df['group'].value_counts()
```

Out[ ]:

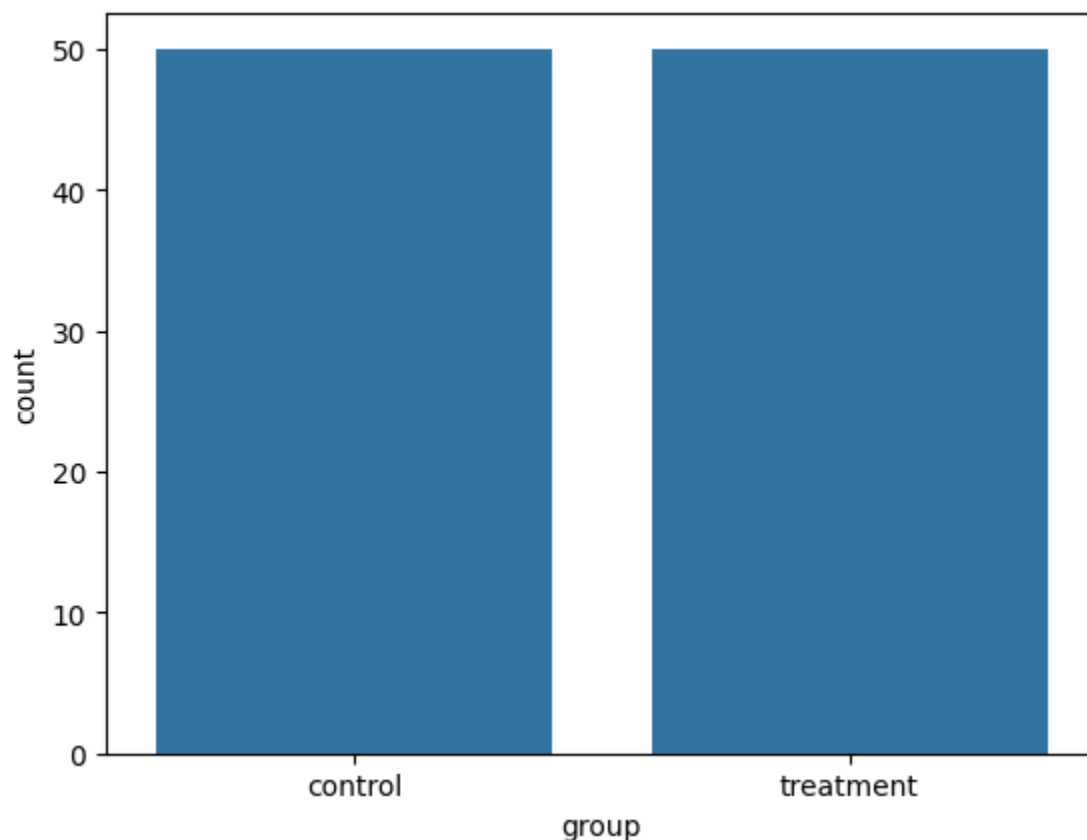
group	
control	50
treatment	50

**dtype:** int64

There are two groups 'control' and 'treatment' which has equal counts 50

In [ ]:

```
#Visualising the column 'group' with countplot  
sns.countplot(data=df,x='group')  
plt.show()
```



This shows that both values have equal count 50 in the dataset

In [ ]:

```
#Finding counts of landing_page  
df['landing_page'].value_counts()
```

Out[ ]:

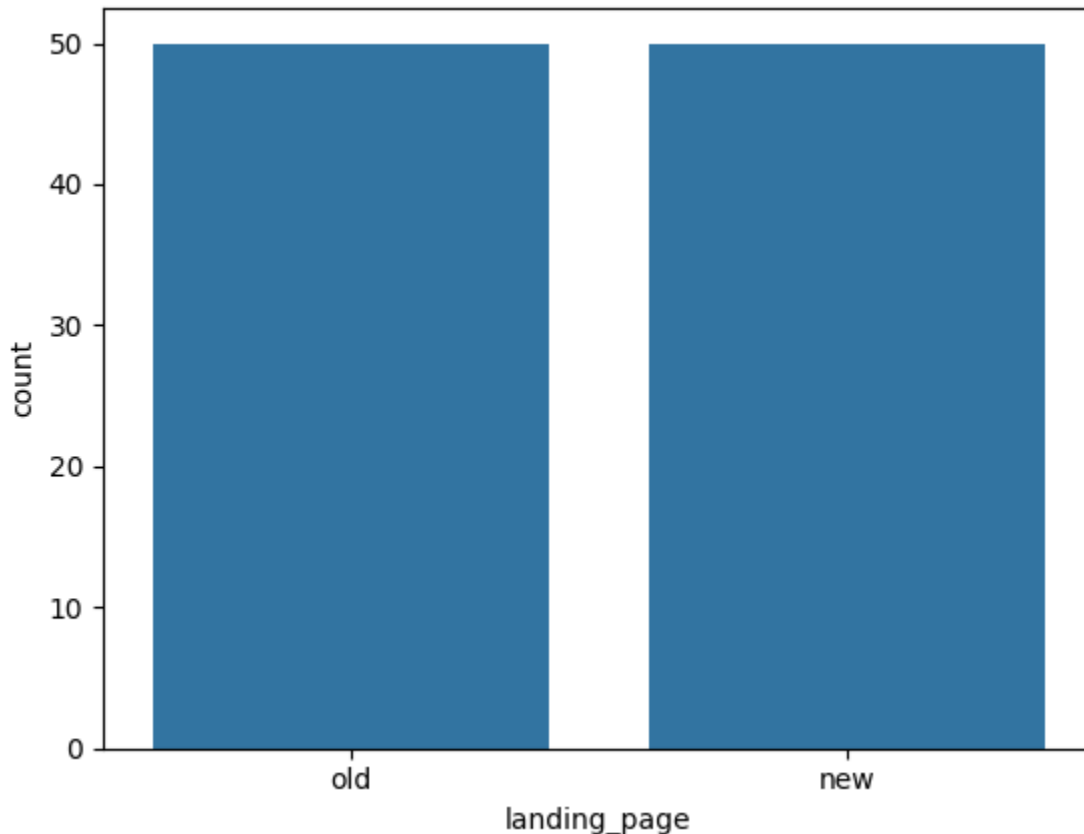
landing_page	
old	50
new	50

**dtype:** int64

There are two types of landing\_page 'old' and 'new' and they are equally spread in the dataset with count as 50

In [ ]:

```
# Visualising the categorical variable 'landing_page' with count plot
sns.countplot(data=df,x='landing_page')
plt.show()
```



This shows that both values have equal count 50 in the dataset

In [ ]:

```
#To find the counts of 'converted' column
df['converted'].value_counts()
```

Out[ ]:

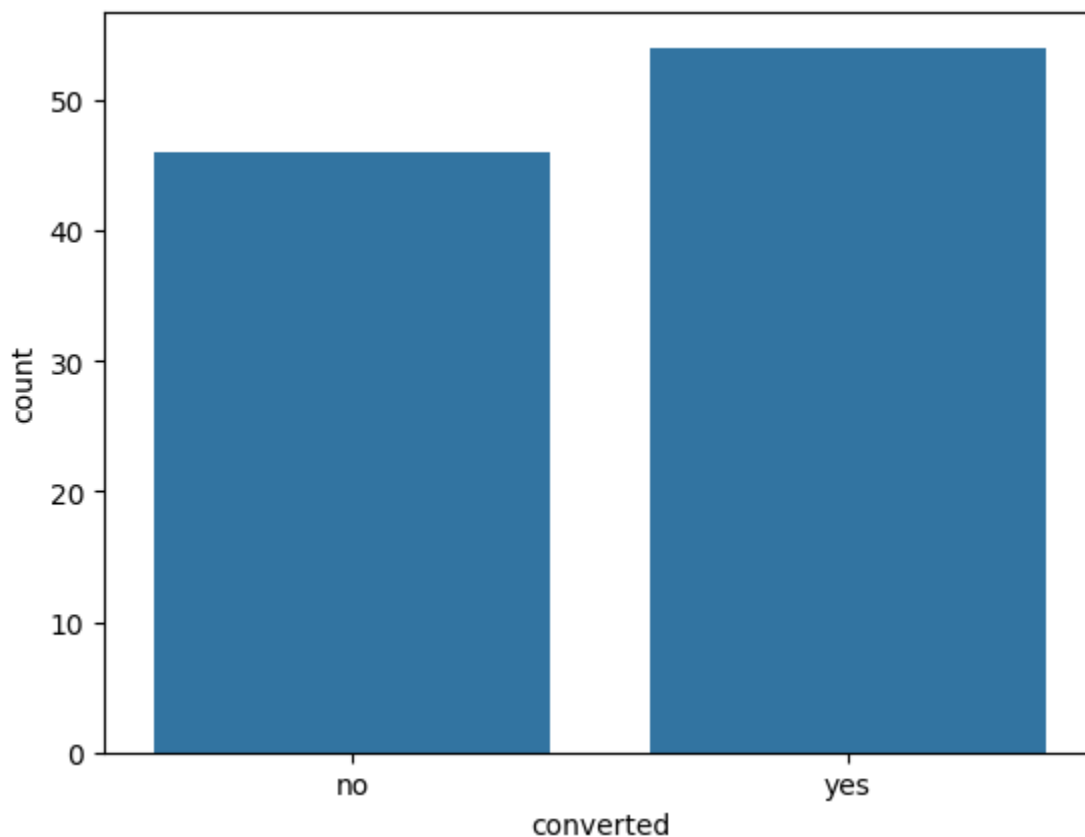
converted	
yes	54
no	46

**dtype:** int64

There are 54 users who got converted to subscribers and 46 users are still non subscribers

In [ ]:

```
# Visualising 'converted' column with countplot
sns.countplot(data=df,x='converted')
plt.show()
```



There are 54 users who got converted to subscribers and 46 users are still non subscribers

In [ ]:

```
#To find the counts of the column 'language_preferred'  
df['language_preferred'].value_counts()
```

Out[ ]:

language_preferred	
Spanish	34
French	34
English	32

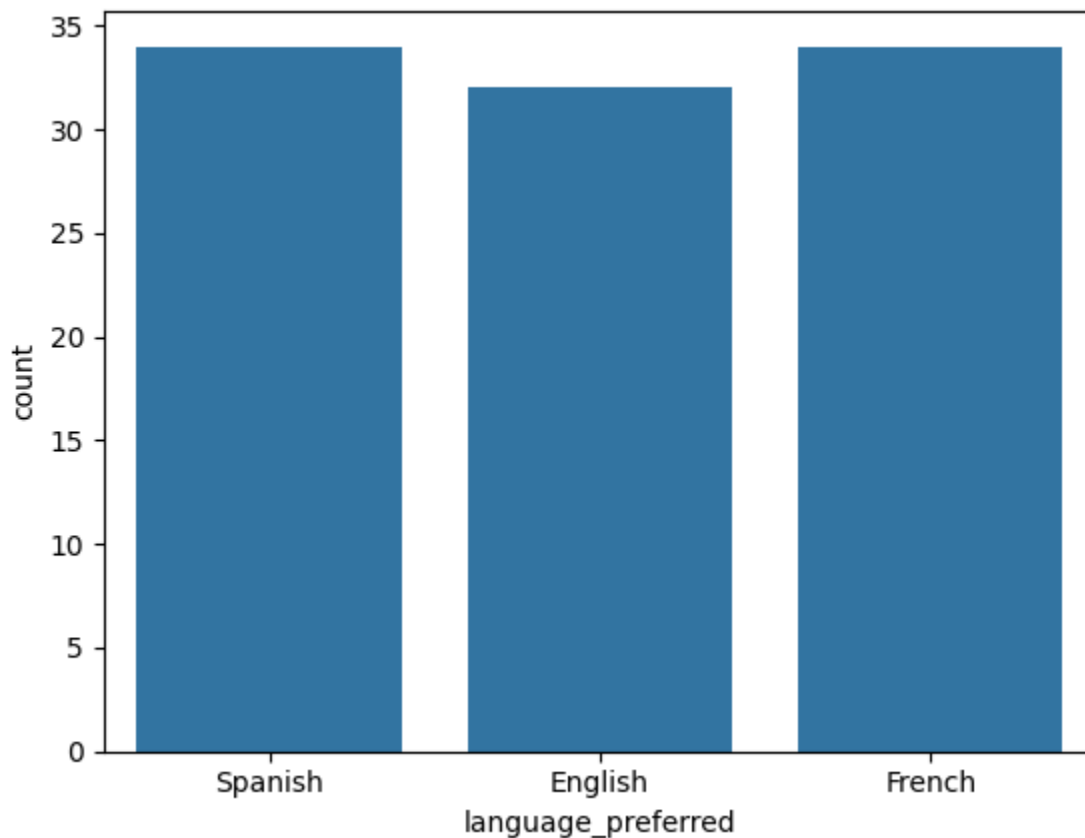
**dtype:** int64

34 users gave 'spanish' and 'French' as their preferred language.32 users gave 'English' as their preferred language.

In [ ]:

```
# Visualising 'language_preferred' column with countplot  
sns.countplot(data=df,x='language_preferred')  
plt.show()
```



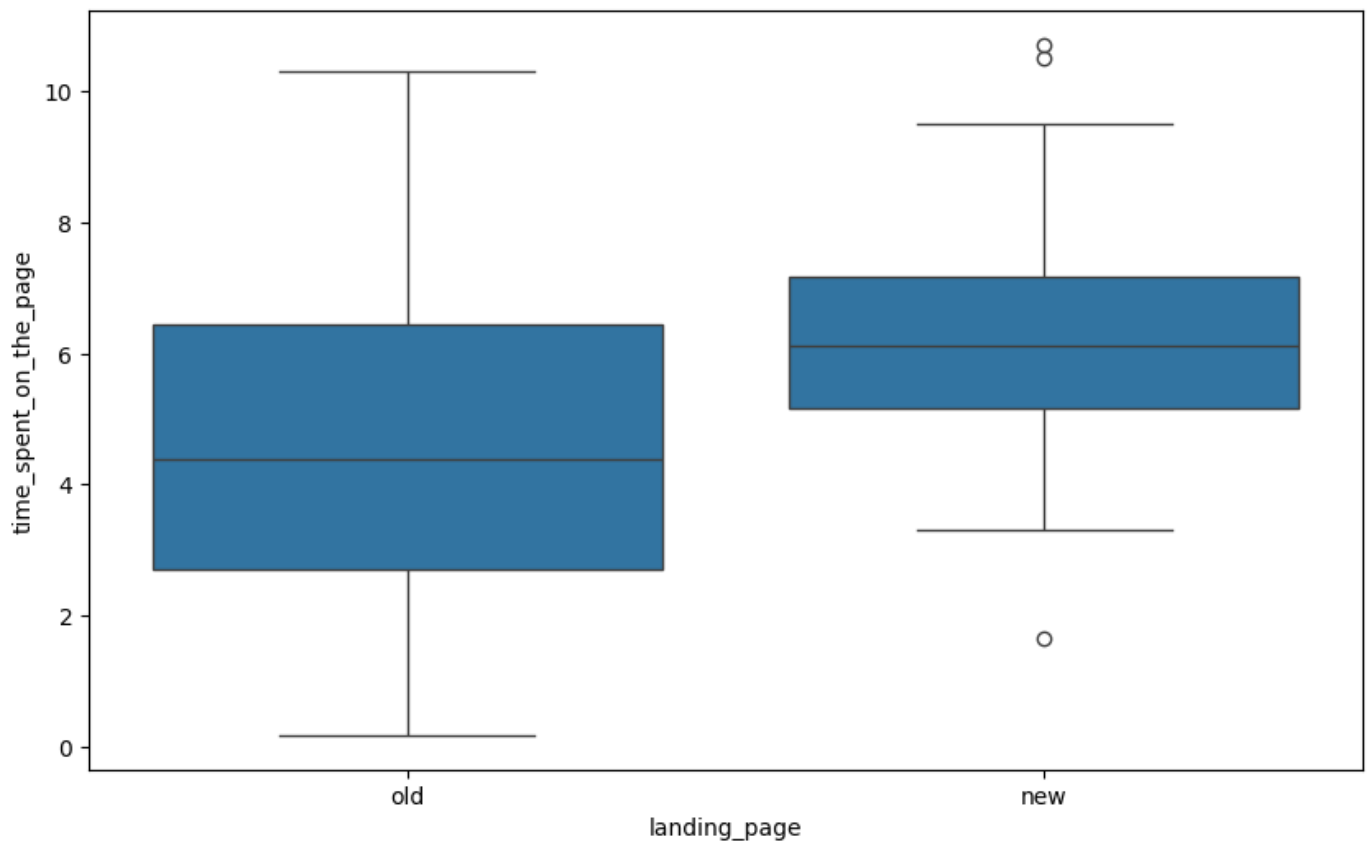


34 users gave 'spanish' and 'French' as their preferred language. 32 users gave 'English' as their preferred language.

## Bivariate Analysis

In [ ]:

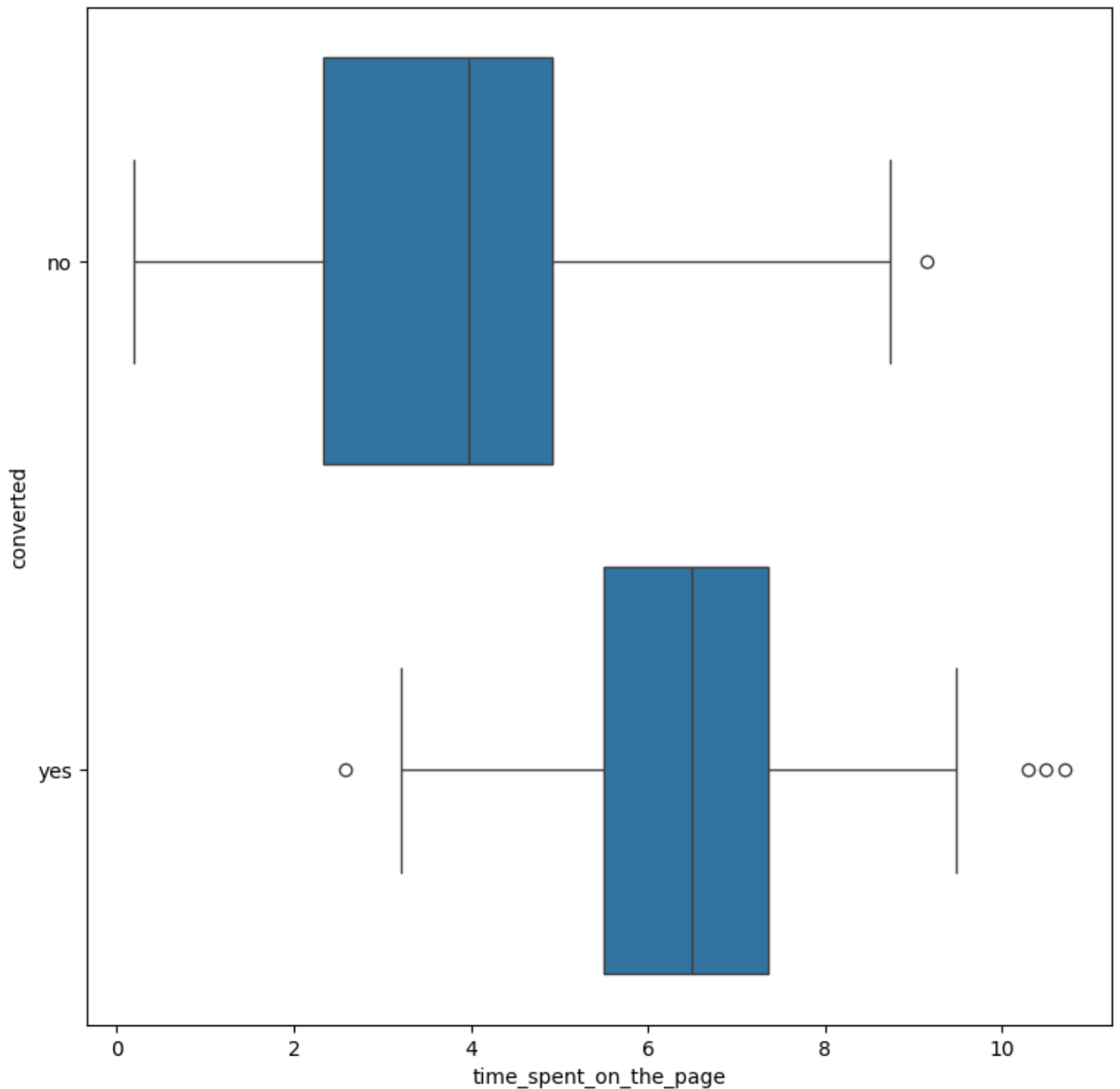
```
#comparing landing_page and time_Spent_on_the_page with boxplot
plt.figure(figsize=(10,6))
sns.boxplot(data=df,x='landing_page',y='time_spent_on_the_page')
plt.show()
```



The median falls around 4 to 5 minutes in old landing page and 6 to 7 minutes in the new landing page .The minimum time spent on the new page (around 3 minutes)is greater than the old landing page (around 0 to 1 minute).The maximum time spent is around 9 minutes in new page and (around 10.5) in old page.

In [ ]:

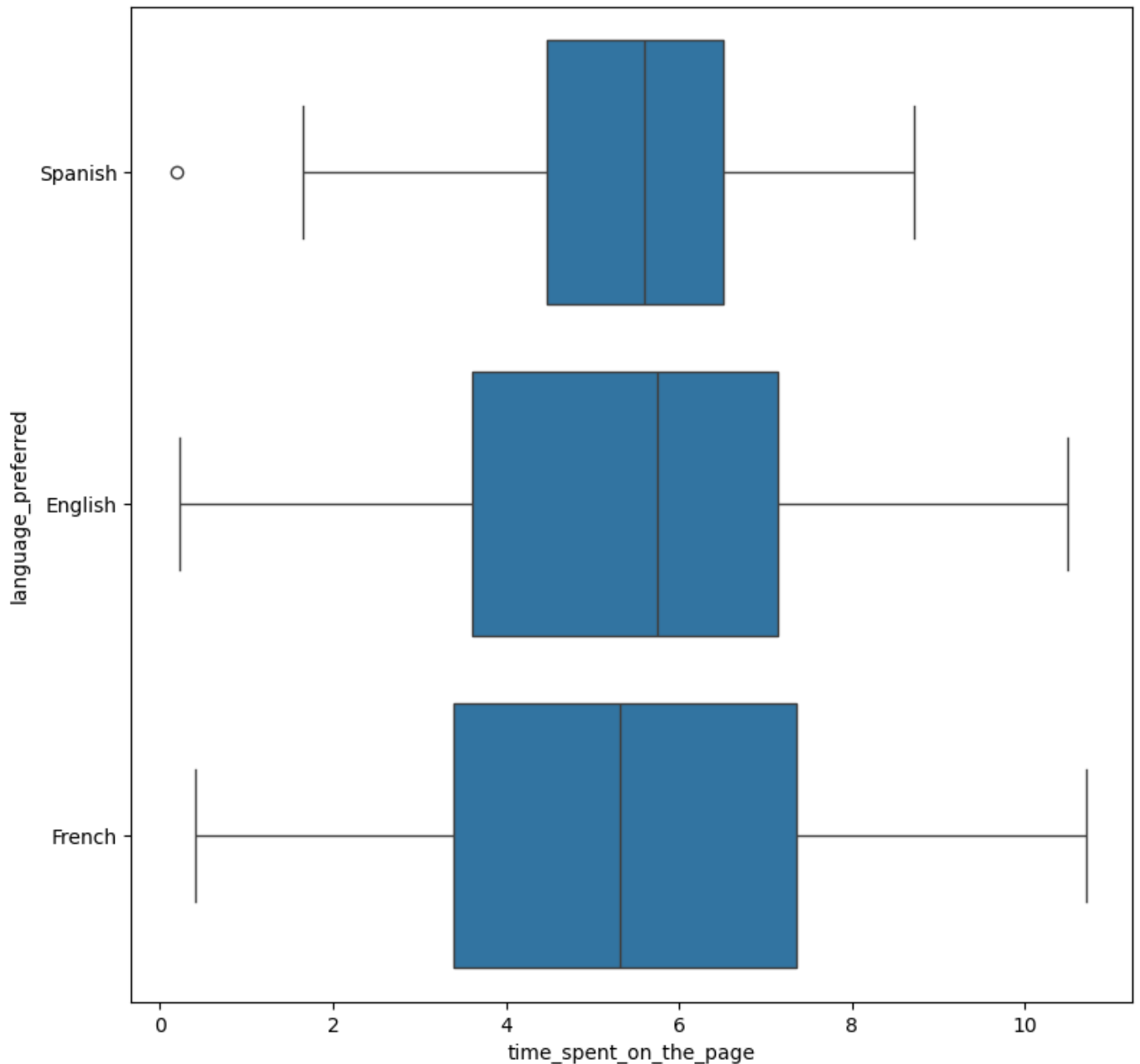
```
# To understand the relationship between 'time_spent_on_the_page' and 'converted' column
plt.figure(figsize=(9, 9))
sns.boxplot(data = df, x = 'time_spent_on_the_page', y = 'converted')
plt.show()
```



This shows that the users who got converted to subscribers spent more time on the page compared to the non subscribers. The subscribers spent around 7 minutes on an average and the non subscribers spent around 4 minutes on an average in the portal page

In [ ]:

```
# To understand the distribution of 'time_spent_on_the_page' among the 'language_preferred'
plt.figure(figsize=(9, 9))
sns.boxplot(data = df, x = 'time_spent_on_the_page', y = 'language_preferred')
plt.show()
```



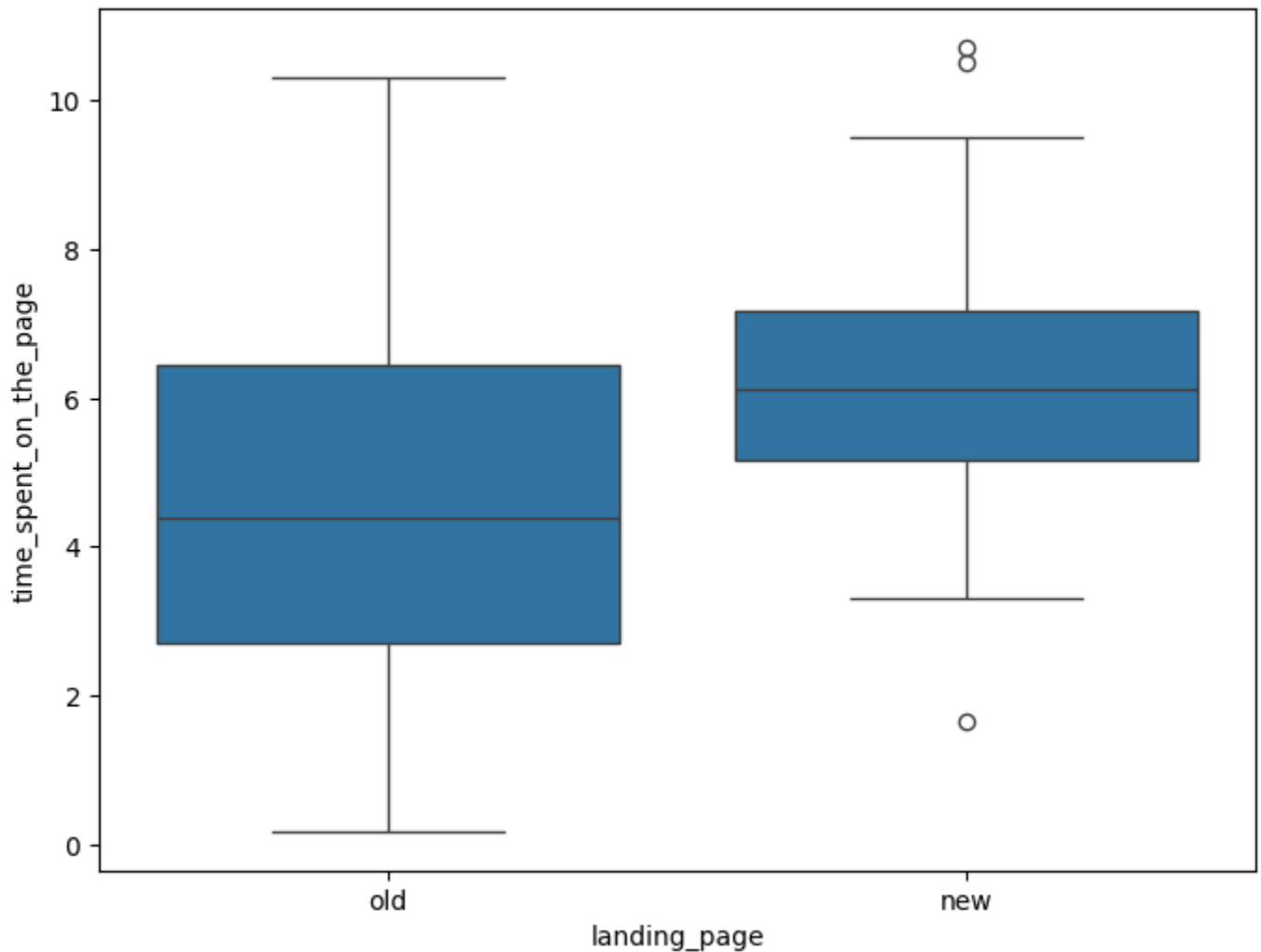
The median in spanish and english is around 6 minutes and french is around 5 minutes .The minimum time spent by the spanish users is approximately around 2 minutes and is greater than english and french .

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

### Perform Visual Analysis

In [ ]:

```
# visual analysis of the time spent on the new page and the time spent on the old page
plt.figure(figsize=(8,6))
sns.boxplot(x = 'landing_page', y = 'time_spent_on_the_page', data = df)
plt.show()
```



## Step 1: Define the null and alternate hypotheses

The null hypothesis is:

$$H_0 : \mu_{\text{new}} \leq \mu_{\text{existing}}$$

The alternate hypothesis is:

$$H_a : \mu_{\text{new}} > \mu_{\text{existing}}$$

## Step 2: Select Appropriate test

We need to a one tailed test and the standard deviation is unknown and we have two independent samples to compare ,so we will select Two Independent sample t test

## Step 3: Decide the significance level

$\alpha=0.05$

## Step 4: Collect and prepare data

In [ ]:

```
# create subsetted data frame for new landing page users
time_spent_new = df[df['landing_page'] == 'new']['time_spent_on_the_page']

# create subsetted data frame for old landing page users
time_spent_old = df[df['landing_page'] == 'old']['time_spent_on_the_page']
```

In [ ]:

```
print('The sample standard deviation of the time spent on the new page is:', round(time_
print('The sample standard deviation of the time spent on the new page is:', round(time_
```

The sample standard deviation of the time spent on the new page is: 1.82

The sample standard deviation of the time spent on the new page is: 2.58

The two standard deviations are assumed to be unequal

## Step 5: Calculate the p-value

In [ ]:

```
from scipy.stats import ttest_ind

# code to calculate the p-value
test_stat, p_value = ttest_ind(time_spent_new, time_spent_old, equal_var = False, alter
print('The p-value is', p_value)
```

The p-value is 0.0001392381225166549

## Step 6: Compare the p-value with $\alpha$

In [ ]:

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject t
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail
```

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

## Step 7: Draw inference

This test shows that the users spend more time on the new landing page than the existing landing page

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

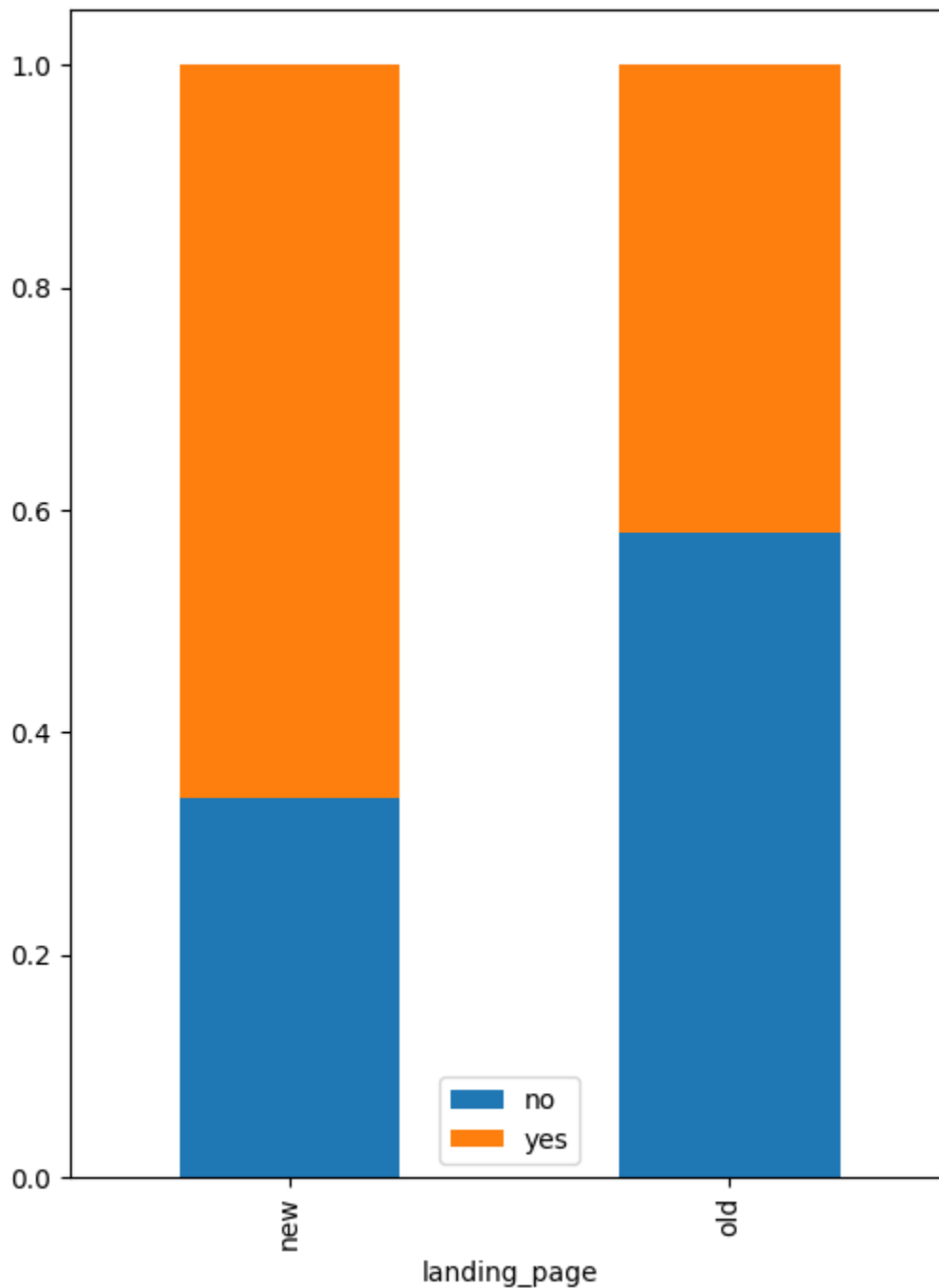
## 2. Is the conversion rate (the proportion of users who visit the landing page and get converted) for the new page greater than the conversion rate for the old page?

### Performing Visual Analysis

In [ ]:

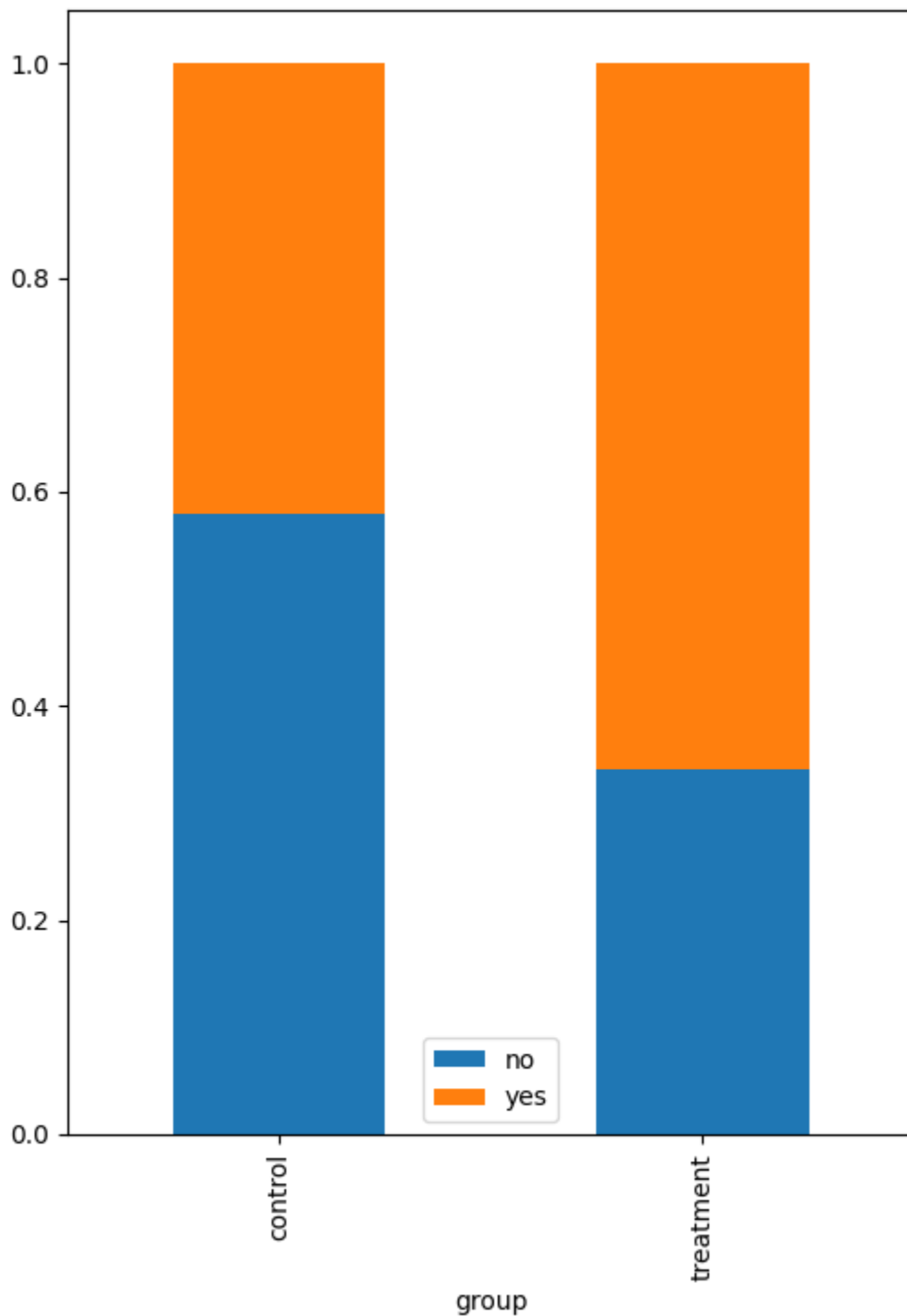
```
# the code to visually compare the conversion rate for the new page and the conversion r
pd.crosstab(df['landing_page'],df['converted'],normalize='index').plot(kind="bar", figsi
```

```
plt.legend()  
plt.show()
```



In [ ]:

```
# the code to visually compare the conversion rate for the new page and the conversion r  
pd.crosstab(df['group'],df['converted'],normalize='index').plot(kind="bar", figsize=(6,8  
plt.legend()  
plt.show()
```



The above graph shows that the users who got converted to subscribers are more in new landing page (group :treatment) compared to the old landing page(group:control)

### Step 1: Define the null and alternate hypotheses

The null hypothesis is:

$$H_0 : p_{\text{new}} \leq p_{\text{existing}}$$

The alternate hypothesis is:

$$H_a : p_{\text{new}} > p_{\text{existing}}$$



## Step 2: Select Appropriate test

We need to compare proportions of two independent samples and hence we take two proportion z test

## Step 3: Decide the significance level

$\alpha = 0.05$

Step 4: Collect and prepare data

In [ ]:

```
# calculate the number of converted users in the treatment group
new_converted = df[df['group'] == 'treatment']['converted'].value_counts()['yes']
# calculate the number of converted users in the control group
old_converted = df[df['group'] == 'control']['converted'].value_counts()['yes']

n_control = df.group.value_counts()['control'] # total number of users in the control gr
n_treatment = df.group.value_counts()['treatment'] # total number of users in the treatm

print('The numbers of users served the new and old pages are {0} and {1} respectively'.f
```

The numbers of users served the new and old pages are 50 and 50 respectively

In [ ]:

```
print('The numbers of users converted to subscribers from the new and old pages are {0}
```

The numbers of users converted to subscribers from the new and old pages are 33 and 21 respectively

## Step 5: Calculate the p value

In [ ]:

```
# the code to import the required function
from statsmodels.stats.proportion import proportions_ztest

# write the code to calculate the p-value
test_stat, p_value = proportions_ztest([new_converted, old_converted], [n_treatment, n_

print('The p-value is', p_value)
```

The p-value is 0.008026308204056278

## Step 6 : Comparing p value with $\alpha$

In [ ]:

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject t
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail
```

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

## Step 7: Draw Inference

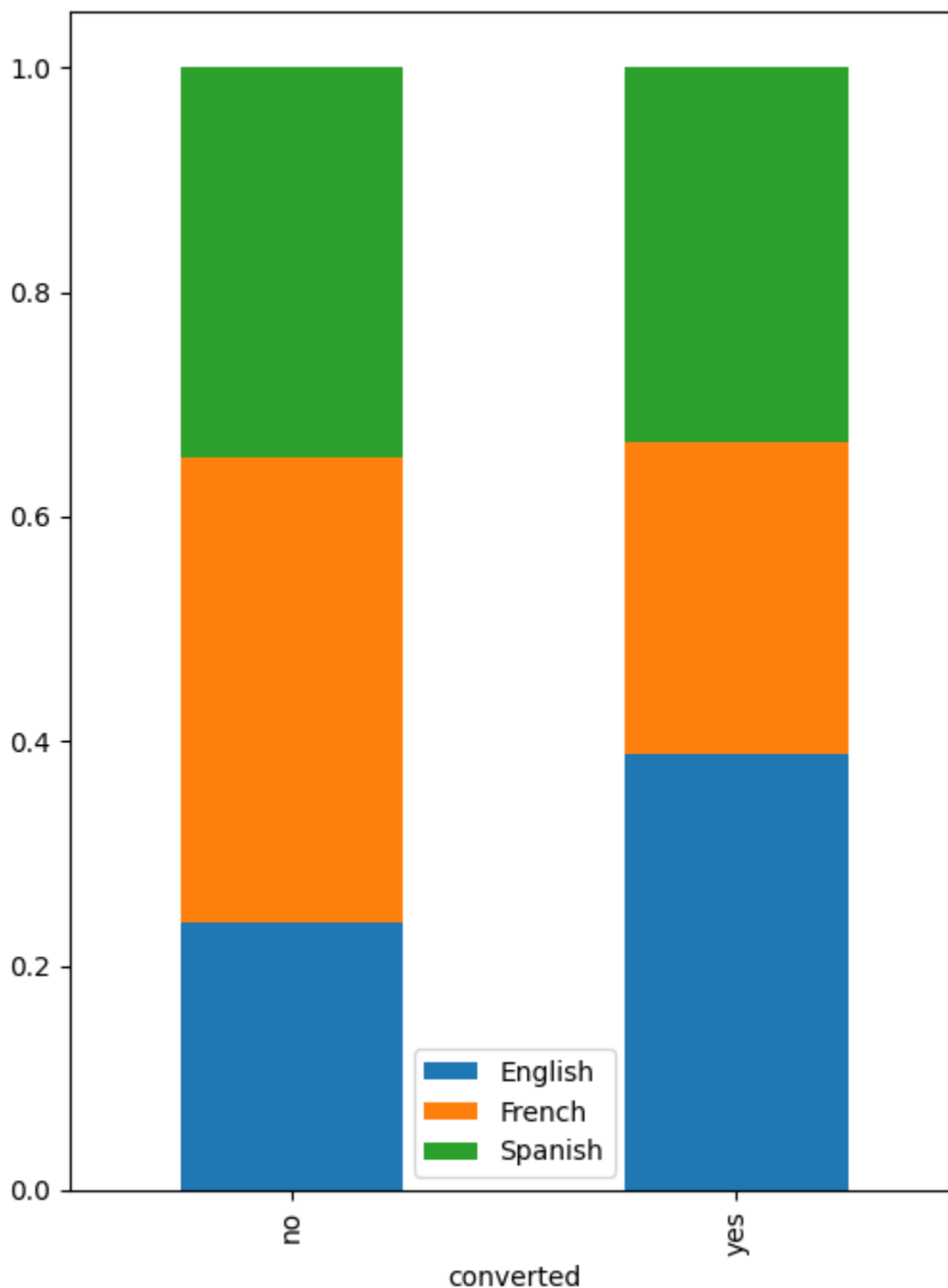
The conversion rate (the proportion of users who visit the landing page and get converted) for the new page is greater than the conversion rate for the old page

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

#### Perform Visual Analysis

In [ ]:

```
# the code to visually plot the dependency between conversion status and preferred language
pd.crosstab(df['converted'],df['language_preferred'],normalize='index').plot(kind="bar",
plt.legend()
plt.show()
```



From the graph, visually we can infer that the people who chose french as the preferred language seems to be mostly non subscriber compared to the subscribers. In the list of subscribers (converted='yes'), most

users are the ones who gave their preferred language as english. It seems there is dependency between both attributes. Let's dive deeply and infer the result

### Step 1: Define the null and alternate hypotheses

Null Hypothesis ( $H_0$ ): Conversion and preferred language are independent.

Alternative Hypothesis ( $H_1$ ): Conversion and preferred language are related.

### Step 2: Select Appropriate test

We are comparing two categorical variables and we are finding their dependency. So we can choose chi-square test also known as Test of Independence

### Step 3: Define Significance level

$\alpha = 0.05$

### Step 4: Collect and prepare data

In [ ]:

```
#Create the contingency table
contingency_table=pd.crosstab(df['converted'],df['language_preferred'],margins=True)
```

In [ ]:

```
contingency_table
```

Out[ ]:

language_preferred	English	French	Spanish	All
converted				
no	11	19	16	46
yes	21	15	18	54
All	32	34	34	100

### Step 5: calculate p value

In [ ]:

```
from scipy.stats import chi2_contingency
chi,p_value,dof,expected=chi2_contingency(contingency_table)
p_value
```

Out[ ]:

```
0.7970832186581233
```

### Step 6: Compare p value with level of significance

In [ ]:

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject t
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail
```

As the p-value 0.7970832186581233 is greater than the level of significance, we fail to reject the null hypothesis.

### Step 7 :Draw Inference

Conversion and preferred language are related.

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

### Perform Visual Analysis

In [ ]:

```
# create a new DataFrame for users who got served the new page
df_new = df[df['landing_page'] == 'new']
```

In [ ]:

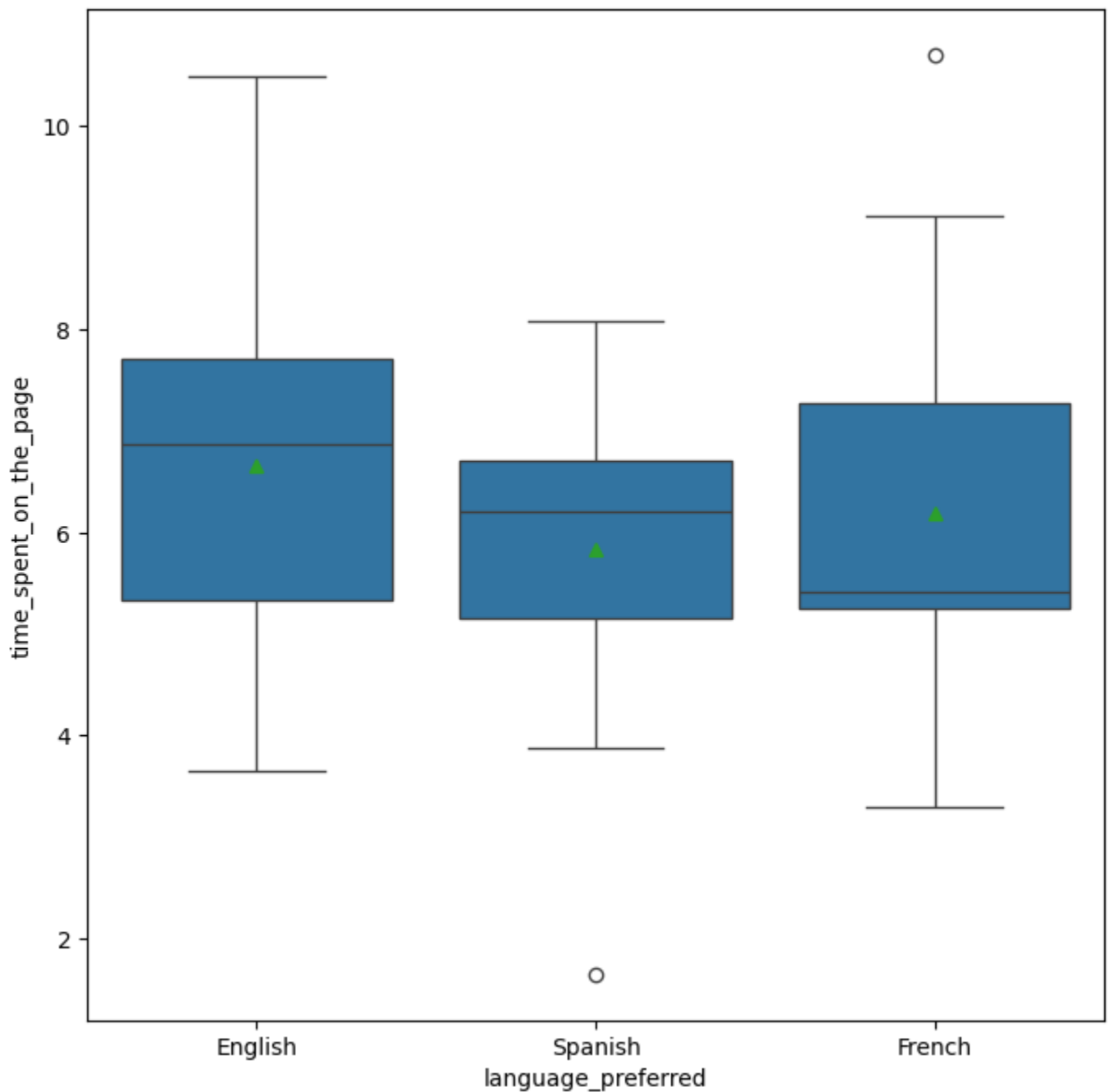
```
df_new.head()
```

Out[ ]:

	user_id	group	landing_page	time_spent_on_the_page	converted	language_preferred
1	546468	treatment	new	7.13	yes	English
2	546462	treatment	new	4.40	no	Spanish
4	546459	treatment	new	4.75	yes	Spanish
6	546448	treatment	new	5.25	yes	French
8	546461	treatment	new	10.71	yes	French

In [ ]:

```
# the code to visually plot the time spent on the new page for different language users
plt.figure(figsize=(8,8))
sns.boxplot(x = df_new.language_preferred, y = df_new.time_spent_on_the_page, showmeans)
plt.show()
```



From the above graph, it seems like the users on new landing page with English as preferred language spent more time than the other language preferred users

In [ ]:

```
# the code to calculate the mean time spent on the new page for different language users
df_new.groupby(['language_preferred'])['time_spent_on_the_page'].mean()
```

Out[ ]:

time_spent_on_the_page	
language_preferred	
English	6.663750
French	6.196471
Spanish	5.835294

**dtype:** float64

The above result shows that users who spent more time on the page gave their preferred language as 'english'

### Step 1: Define Null hypothesis and Alternate hypothesis

The Null hypothesis is: The time spent on the new page is different for the different language users

The Alternate hypothesis is: The time spent on the new page is same for the different language users

### Step 2: Select Appropriate test

We are comparing the time spent on the page with the language\_preferred column for the new page users. We are comparing one categorical and one numerical variables. and we are comparing more than two population means, so we can use ANOVA test

### Step 3: Define level of significance

$\alpha = 0.05$

### Step 4: collect and prepare data

In [ ]:

```
# create a subsetting data frame of the time spent on the new page by English language users
time_spent_English = df_new[df_new['language_preferred']=="English"]['time_spent_on_the_
# create subsetting data frames of the time spent on the new page by French and Spanish l
time_spent_French = df_new[df_new['language_preferred']=="French"]['time_spent_on_the_pa
time_spent_Spanish = df_new[df_new['language_preferred']=="Spanish"]['time_spent_on_the_
```

For performing ANOVA test, we need to perform the following two tests:

- Shapiro Wilk's Test- To check for the normality
- Levene's test - To Test the equality of variance

### Step 5: Shapiro Wilk's test

Shapiro wilk test is conducted to check for normality

The null hypothesis is:

time\_Spent\_on\_the\_page follows normal distribution

The alternate hypothesis is:

time\_Spent\_on\_the\_page does not follow normal distribution

In [ ]:

```
from scipy import stats
w,p_value=stats.shapiro(df_new['time_spent_on_the_page'])
print('The p-value is', p_value)
```

The p-value is 0.8040016293525696

The p-value 0.80 is greater than alpha so we fail to reject the null hypothesis

Hence,

time\_Spent\_on\_the\_page follows normal distribution

Step 6: Levene's test

Levene's test to test for equality of variance

Null hypothesis ( $H_0$ ): The variances are equal across the groups.

Alternative hypothesis ( $H_1$ ): The variances are not equal across the groups

In [ ]:

```
from scipy.stats import levene
w,p_value=stats.levene(time_spent_English,time_spent_French,time_spent_Spanish)
print('The p-value is', p_value)
```

The p-value is 0.46711357711340173

The p value 0.46 is greater than the alpha which is 0.05

Hence,we fail to reject the null hypothesis

The variances are equal across the groups.

time\_Spent\_on\_the\_page follows normal distribution and the variances are equal across the groups.Now we proceed with the ANOVA test

### Step 7:ANOVA test

In [ ]:

```
from scipy.stats import f_oneway
f_value,p_value=f_oneway(time_spent_English,time_spent_French,time_spent_Spanish)
print('The p-value is', p_value)
```

The p-value is 0.43204138694325955

### Step 8 :Comparing the p value with level of significance

In [ ]:

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we reject t
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we fail
```

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

### Step 9:Draw Inference

The time spent on the new page is different for the different language users

---

# Conclusion and Business Recommendations

## Conclusions:

The dataset do not contain any missing values, as the total number of rows and the counts in all columns is 100.

There are only two unique values in the `group`, `landing_page`, and `converted` columns

There are 50 rows labeled as 'control' with 'old' as the landing page, and 54 users got converted to subscribers. The 'Spanish' language is the most preferred among users, with 34 users choosing it as their preferred language.

The average time spent by users on the website is approximately 5 minutes and 37 seconds, with the maximum time spent being around 10 minutes and 71 seconds.

There are no outliers in this data. The median time spent by users on the website falls around 5 to 6 minutes, and more than 20 users spend approximately 5 to 6 minutes in the portal.

There are 54 users (converted to subscribers) and 46 non-subscribers. Users prefer 'Spanish', 'French', and 'English' as their languages, with Spanish being the most chosen language.

The median time spent on the old landing page is around 4 to 5 minutes, while the median for the new landing page is around 6 to 7 minutes. The new landing page also has a higher minimum time spent (around 3 minutes) compared to the old page (around 0 to 1 minute). The maximum time spent is around 9 minutes for the new page and 10.5 minutes for the old page.

Subscribers spend more time on the page than non-subscribers, with an average of 7 minutes spent by subscribers and around 4 minutes by non-subscribers.

The median time spent for Spanish and English users is around 6 minutes, while French users have a median of around 5 minutes. The minimum time spent by Spanish users (around 2 minutes) is greater than that of English and French users.

Users on the new landing page spent more time compared to those on the old landing page. The users who got converted to subscribers were mostly from the new landing page (33 converted users) compared to the old landing page (21 converted users).

The conversion rate (the proportion of users who visit the landing page and get converted) is higher for the new page compared to the old page.

The visual graph suggests that French-speaking users are more likely to be non-subscribers compared to subscribers. There seems to be a relationship between preferred language and conversion status.

Users on the new landing page with English as their preferred language spent more time compared to users who preferred other languages.



The time spent on the page follows a normal distribution, and the variance in time spent is equal across groups.

There is a significant difference in the time spent on the new landing page between users of different preferred languages.

## **Recommendations:**

### **Business Recommendations:**

If the new landing page proves to engage users more effectively (by spending more time or converting at a higher rate), it is a wise choice to prioritize the new page across all users. This could potentially increase user retention and subscription rates.

Based on the above results, if the new landing page significantly increases time spent or conversion rates, the design of the new page should be expanded for higher engagement of users.

If there is a dependency between conversion status and preferred language (e.g., users who prefer English are more likely to convert), increase marketing efforts to specific language groups.

If the new landing page proves a higher conversion rate, promote the new page through advertisements and campaigns.

If users on the new landing page spend more time, consider which features of the new page (such as interactive features) are attracting users. Ensure these elements are developed to maximize user time on the page.

If certain language groups are showing better user conversion to subscribers, provide more localized content, such as region-specific news and features. This could help increase conversion rates among different language users.

If non-subscribers are spending less time on the old landing page, try to implement features to increase their time spent on the portal.

If some language groups or user segments cause decrease in ratio of users getting converted to subscribers, understand which elements are making them feel that way. Address these barriers and make improvements.

It would be nice idea to collect feedback and reviews from the users(e.g., surveys ) to understand what users like or dislike about the new page. This will make the portal achieve its business target.

---