In [1]:
```
# Our goal here is to analyize and compare the old landing page to the new landing pag
```

In [2]:
```
# I have to import all packages i will use in this project.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import scipy.stats as stats
```

In [3]:
```
# I am Loading in the dataset i will be using under the name df.
df = pd.read_csv('/Users/conne/Downloads/abtest.csv')
```

In [4]:
```
# I'm checking to see that I loaded the data properly.
df
```

Out[4]:

	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
...	...	...	...	...	...	...
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

100 rows × 6 columns

In [5]:
```
# I am looking at the number of rows and columns, as well as the first and last 5 rows
print(df.shape)
print(df.head)
print(df.tail)
```

```
(100, 6)
<bound method NDFrame.head of     user_id      group landing_page  time_spent_on_the_
page converted  \
0    546592    control          old                    3.48        no
1    546468  treatment          new                    7.13       yes
2    546462  treatment          new                    4.40        no
3    546567    control          old                    3.02        no
4    546459  treatment          new                    4.75       yes
..      ...        ...          ...                     ...       ...
95   546446  treatment          new                    5.15        no
96   546544    control          old                    6.52       yes
97   546472  treatment          new                    7.07       yes
98   546481  treatment          new                    6.20       yes
99   546483  treatment          new                    5.86       yes

   language_preferred
0             Spanish
1             English
2             Spanish
3              French
4             Spanish
..                ...
95            Spanish
96            English
97            Spanish
98            Spanish
99            English

[100 rows x 6 columns]>
<bound method NDFrame.tail of     user_id      group landing_page  time_spent_on_the_
page converted  \
0    546592    control          old                    3.48        no
1    546468  treatment          new                    7.13       yes
2    546462  treatment          new                    4.40        no
3    546567    control          old                    3.02        no
4    546459  treatment          new                    4.75       yes
..      ...        ...          ...                     ...       ...
95   546446  treatment          new                    5.15        no
96   546544    control          old                    6.52       yes
97   546472  treatment          new                    7.07       yes
98   546481  treatment          new                    6.20       yes
99   546483  treatment          new                    5.86       yes

   language_preferred
0             Spanish
1             English
2             Spanish
3              French
4             Spanish
..                ...
95            Spanish
96            English
97            Spanish
98            Spanish
99            English

[100 rows x 6 columns]>
```

In [6]:
```
# I am checking my 5 number suummary and my data types per column.
print(df.describe())
```

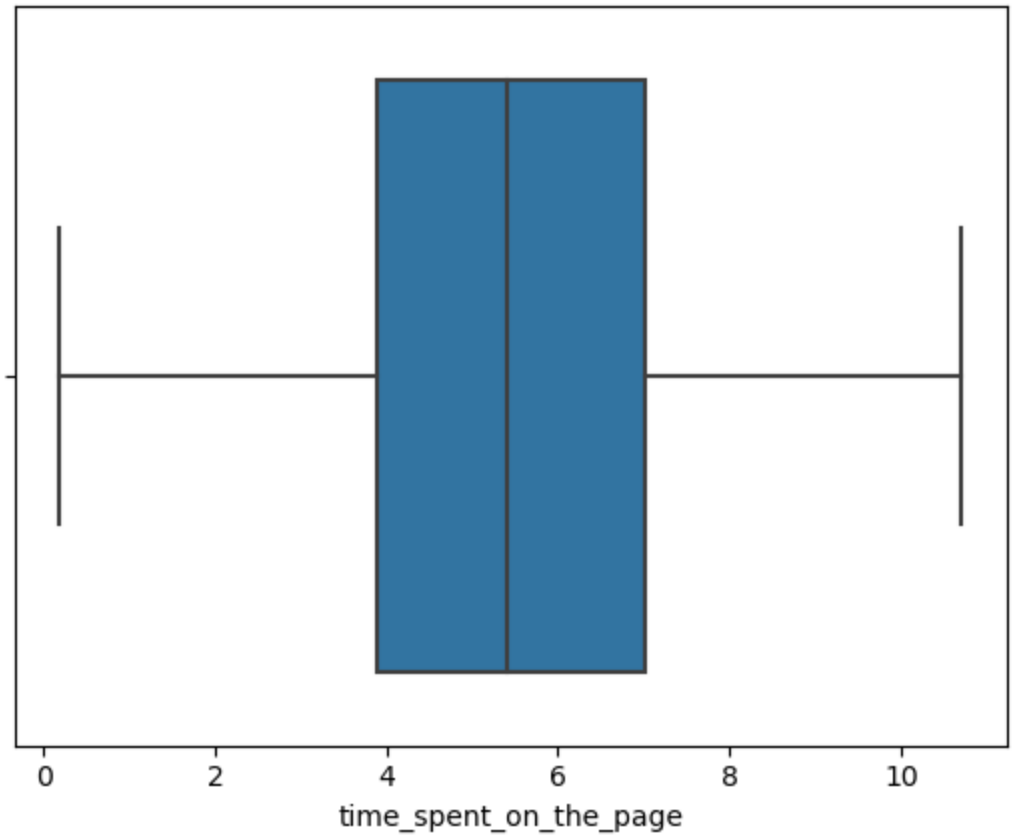
```
print(df.info())
```

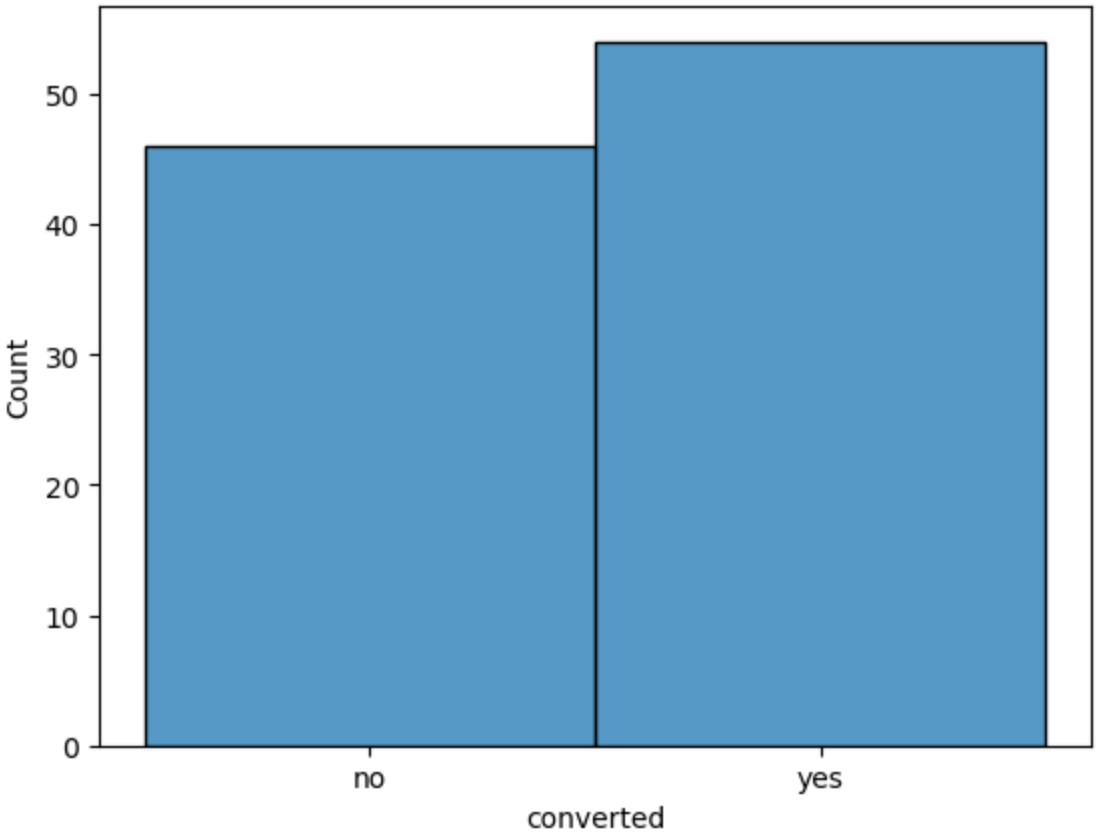
```
             user_id  time_spent_on_the_page
count     100.000000              100.000000
mean   546517.000000                5.377800
std        52.295779                2.378166
min    546443.000000                0.190000
25%    546467.750000                3.880000
50%    546492.500000                5.415000
75%    546567.250000                7.022500
max    546592.000000               10.710000
<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
```

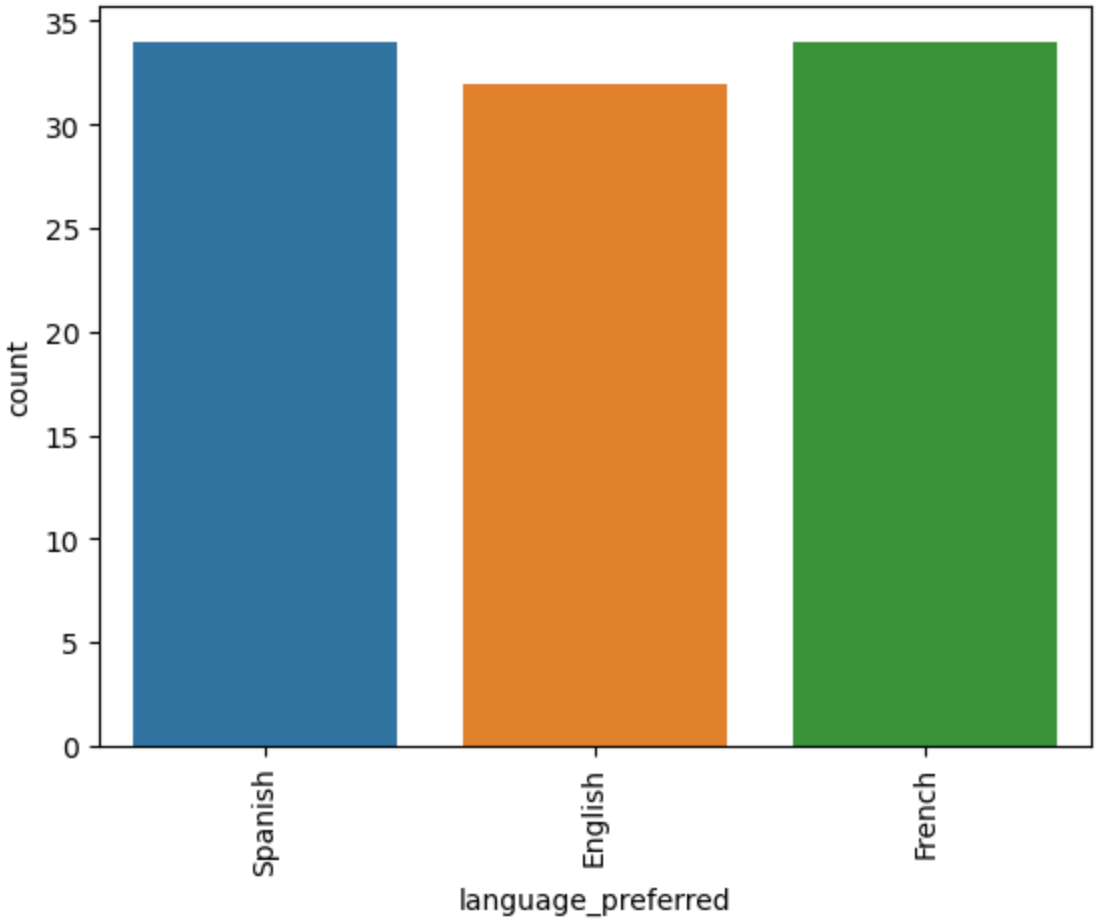
In [7]:
```
# I am checking to see if i have an duplicated rows
df.duplicated().sum()
```

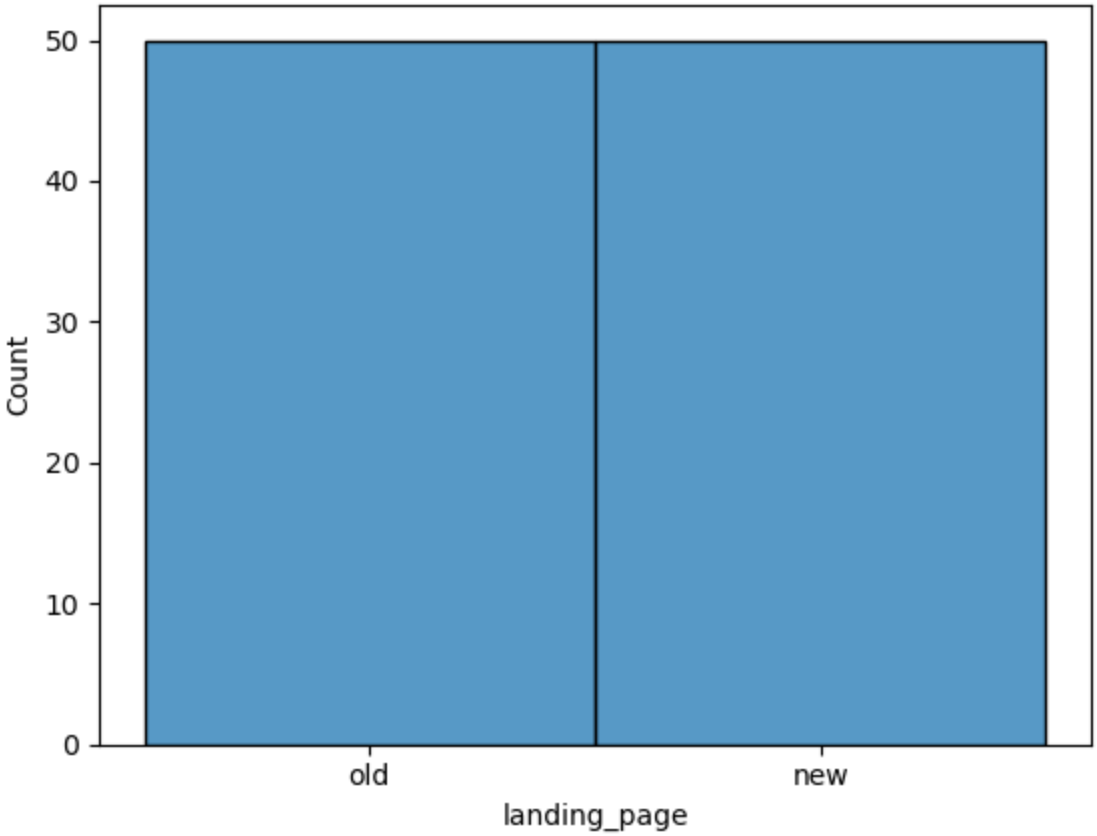
Out[7]:
```
0
```

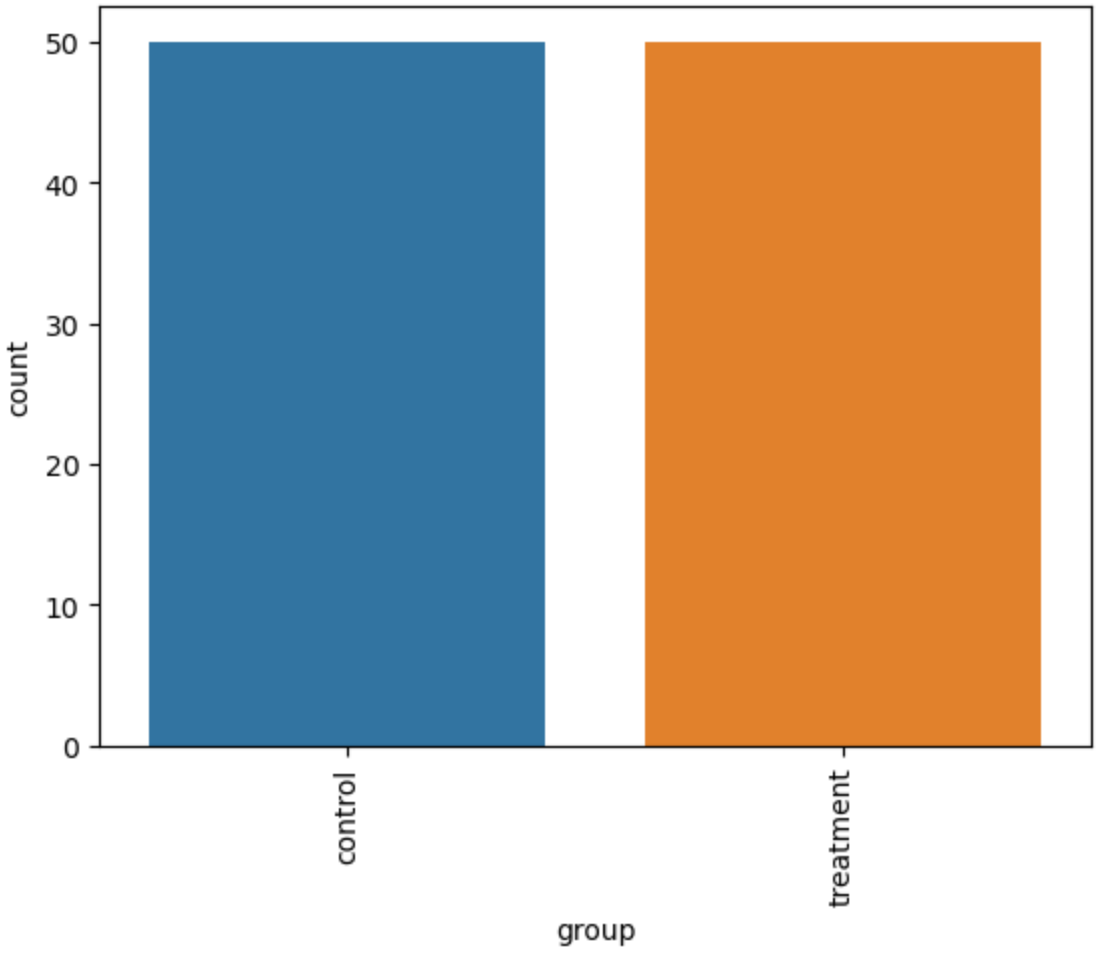
In [8]:
```
# Here I am preforming univariate analysis.
sns.boxplot(data=df,x='time_spent_on_the_page')
plt.show()
sns.histplot(data=df, x='converted')
plt.show()
sns.countplot(data=df, x='language_preferred')
plt.xticks(rotation=90)
plt.show()
sns.histplot(data=df, x='landing_page')
plt.show()
sns.countplot(data=df, x='group')
plt.xticks(rotation=90)
plt.show()
```


0
2
4
6
8
10
time_spent_on_the_page


50
40
30
20
10
0
Count
no
yes
converted

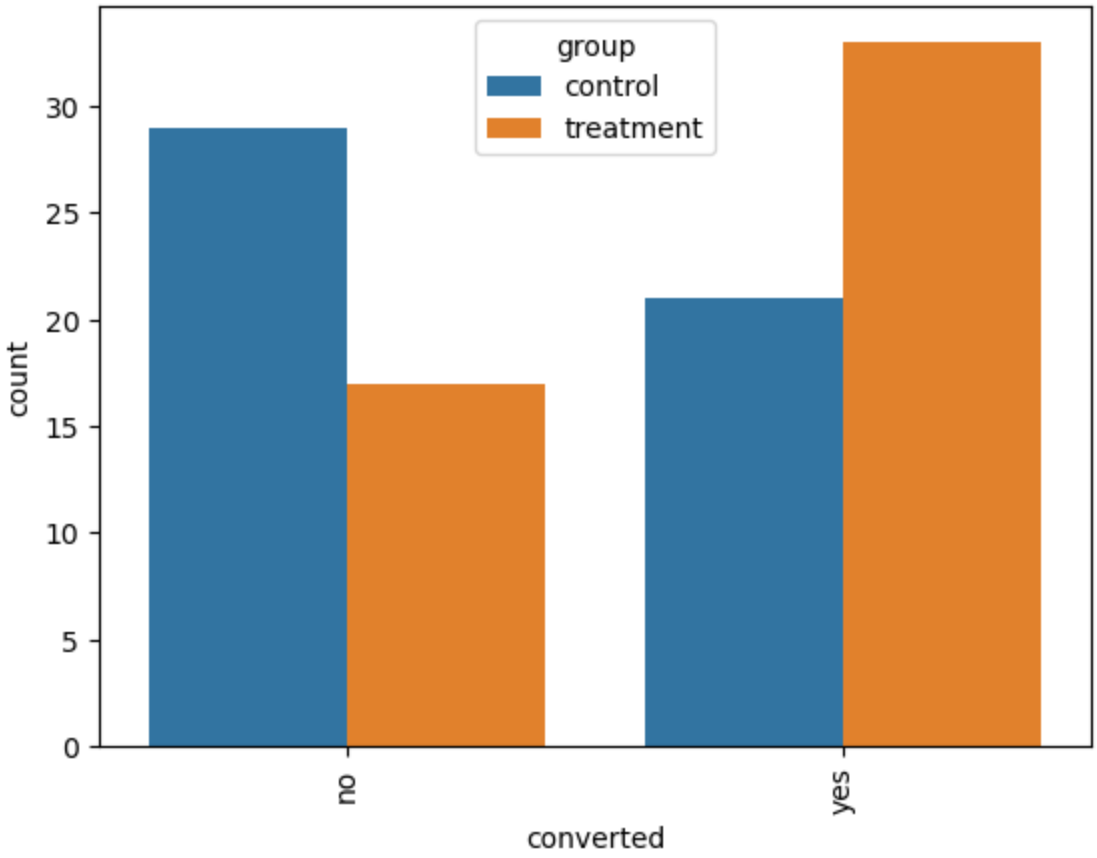

35
30
25
20
15
10
5
0
count
Spanish
English
French
language_preferred

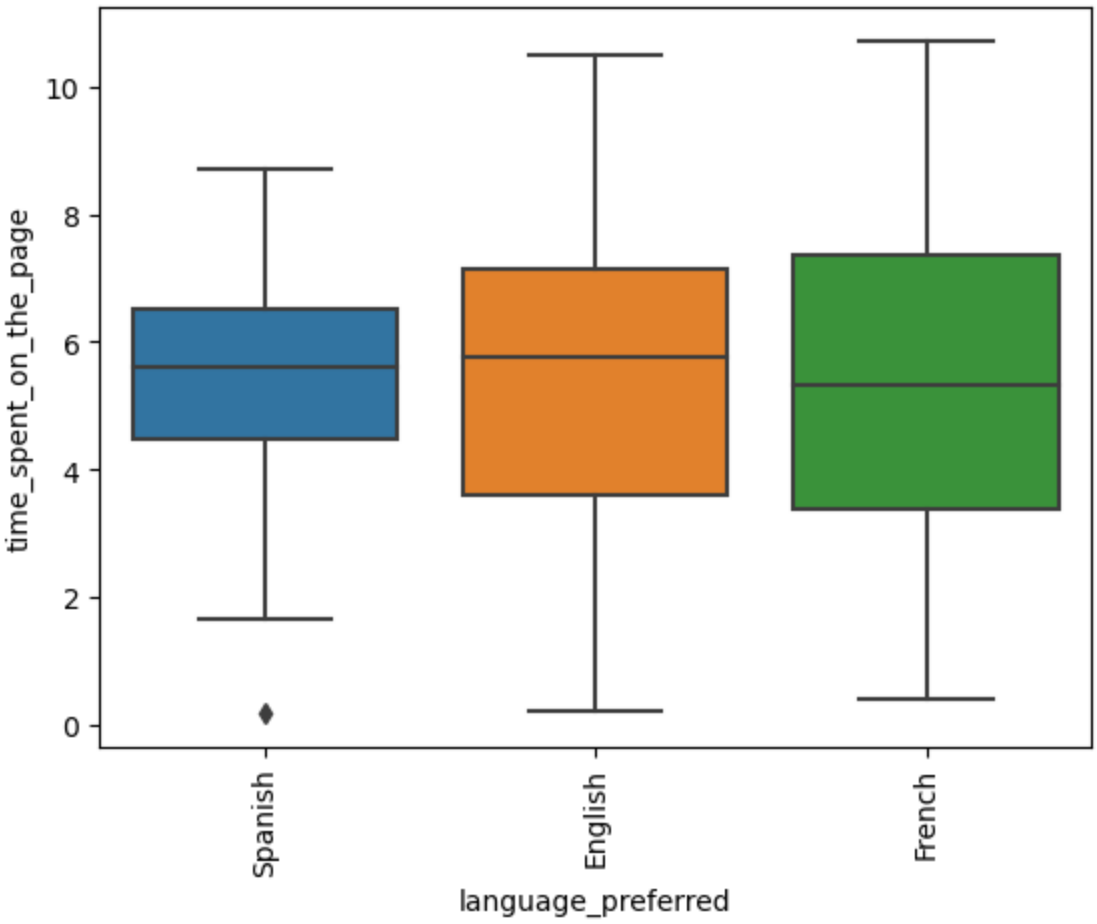

50
40
30
20
10
0
Count
old
new
landing_page

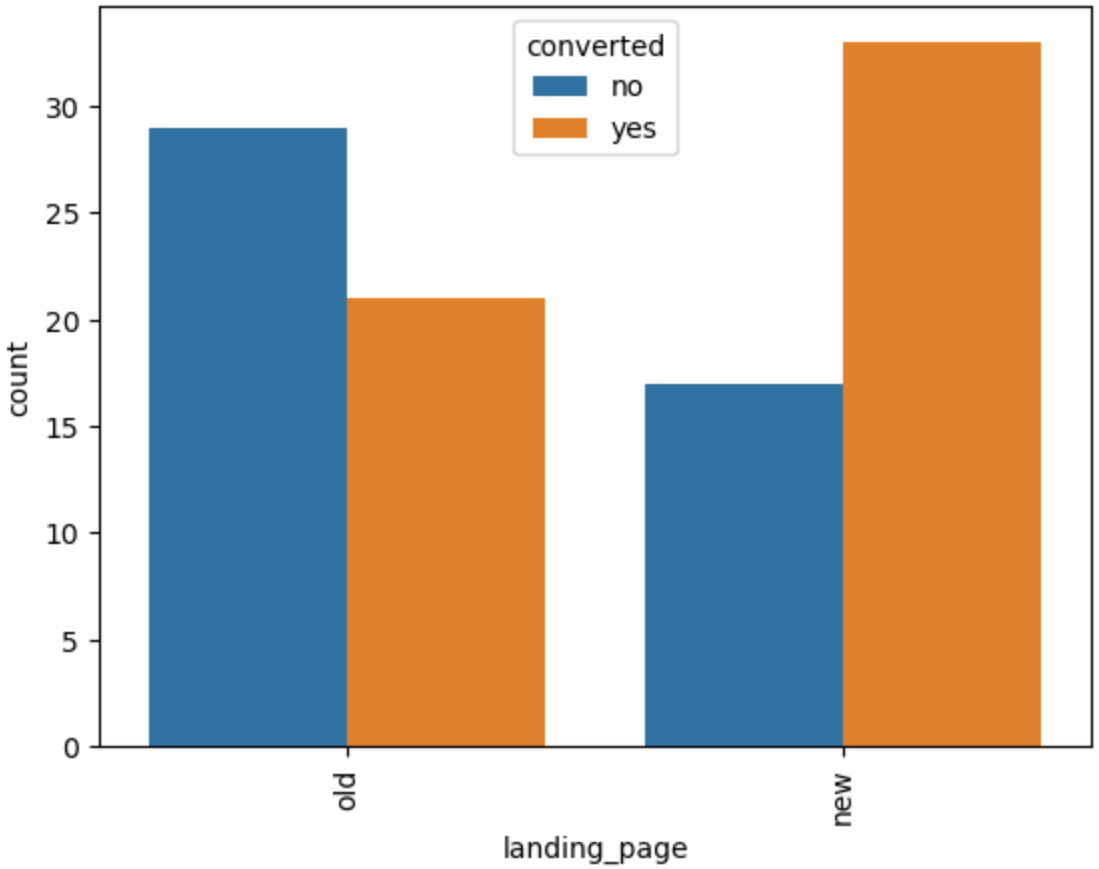


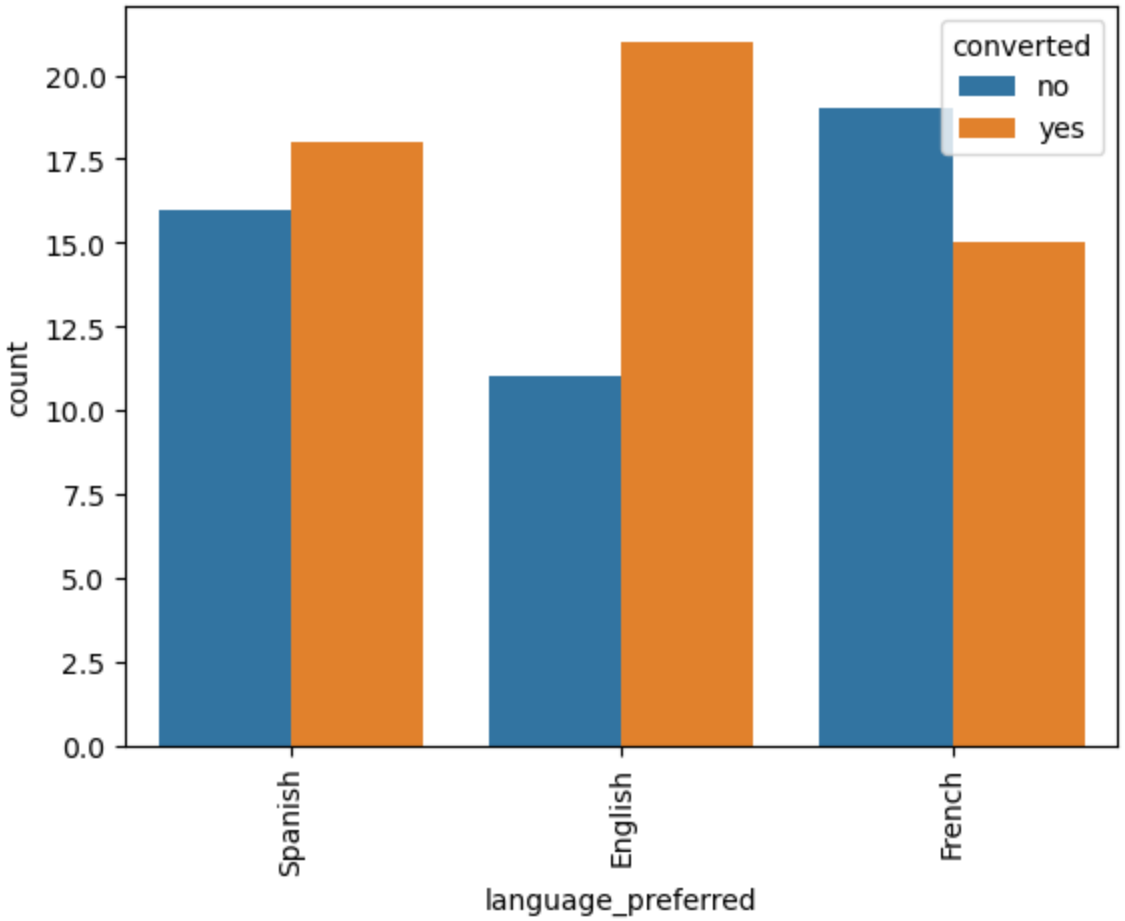
In [9]:
```
# Observations from univariate analysis.
# Control and treatment are equal
# There are more converted users than non converted users
# The average time spent on either page was around five to five and a half minutes
```

In [10]:
```
# Here I am preforming bivariate analysis.
sns.countplot(data=df, x='converted', hue='group' )
plt.xticks(rotation=90)
plt.show()
sns.boxplot(data=df, x="language_preferred", y='time_spent_on_the_page')
plt.xticks(rotation=90)
plt.show()
sns.countplot(data=df, x='landing_page', hue='converted' )
plt.xticks(rotation=90)
plt.show()
sns.countplot(data=df, x='language_preferred', hue='converted' )
plt.xticks(rotation=90)
plt.show()
```


group
control
treatment
count
converted
no
yes
0
5
10
15
20
25
30


time_spent_on_the_page
language_preferred
Spanish
English
French
0
2
4
6
8
10



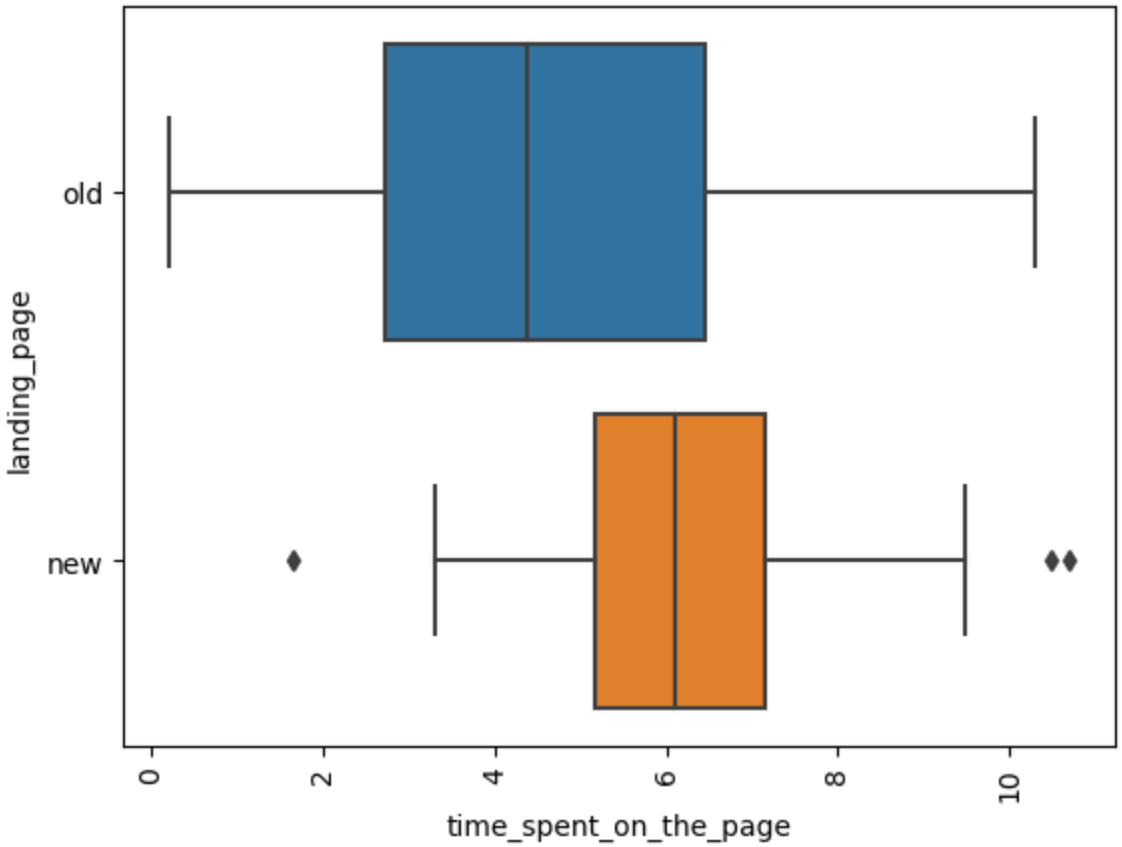


In [11]:
```
# Observations from bivariate analysis.
# English has the highest conversion rate while French has the worst
```

```
# The conversion rate is much higher on the new landing page
# Language does not seem to affect time spent on the page
# The number of treatment users on the new landing page is much higher than the old la
```

In [12]:
```
# Question 1 below
```

In [13]:
```
# I'm preforming a visaul analysis
sns.boxplot(data=df, x='time_spent_on_the_page', y='landing_page' )
plt.xticks(rotation=90)
plt.show()
```



In [14]:
```
# Mu1 is the sample mean of the users time spent on the new page
# Mu2 is the sample mean of the users time spent on the old page
# the null hypothesis is mu1 greater than mu2
# the alternitave hypothesis is mu1 less than or equal to mu2

df_blank = df.set_index("landing_page")
df_new_landing=df_blank.drop('old')
df_old_landing=df_blank.drop('new')
print(df_old_landing.head())
print(df_new_landing.head())
print(df_old_landing.mean(numeric_only=True))
print(df_new_landing.mean(numeric_only=True))
```

```
              user_id    group  time_spent_on_the_page converted  \
landing_page
old            546592  control                    3.48        no
old            546567  control                    3.02        no
old            546558  control                    5.28       yes
old            546581  control                    6.53       yes
old            546548  control                    2.08        no

             language_preferred
landing_page
old                     Spanish
old                      French
old                     English
old                     Spanish
old                     English
              user_id      group  time_spent_on_the_page converted  \
landing_page
new            546468  treatment                    7.13       yes
new            546462  treatment                    4.40        no
new            546459  treatment                    4.75       yes
new            546448  treatment                    5.25       yes
new            546461  treatment                   10.71       yes

             language_preferred
landing_page
new                     English
new                     Spanish
new                     Spanish
new                      French
new                      French
user_id                  546566.5000
time_spent_on_the_page        4.5324
dtype: float64
user_id                  546467.5000
time_spent_on_the_page        6.2232
dtype: float64
```

In [15]:
```
print(df_old_landing.mean(numeric_only=True))
print(df_new_landing.mean(numeric_only=True))
print('The old landings mean is 4.53')
print('The new landings mean is 6.22')
```

```
user_id                  546566.5000
time_spent_on_the_page        4.5324
dtype: float64
user_id                  546467.5000
time_spent_on_the_page        6.2232
dtype: float64
The old landings mean is 4.53
The new landings mean is 6.22
```

In [16]:
```
print(df_old_landing.std(numeric_only=True))
print(df_new_landing.std(numeric_only=True))
print('The old landings standard deviation is 2.58')
print('The new landings standard deviation is 1.82')
```

```
user_id                   17.677670
time_spent_on_the_page     2.581975
dtype: float64
user_id                   14.577380
time_spent_on_the_page     1.817031
dtype: float64
The old landings standard deviation is 2.58
The new landings standard deviation is 1.82
```

In [17]:
```
from scipy.stats import ttest_ind
p_value = ttest_ind(df_new_landing['time_spent_on_the_page'], df_old_landing['time_spe
print(p_value)
```

```
Ttest_indResult(statistic=3.7867702694199856, pvalue=0.9998683876471904)
```

In [18]:
```
# The p value is large so we fail to reject the null hypothesis
# Answer to Question 1
# Since we fail to reject the null hypothesis we know that users do spend more time on
```

In [19]:
```
# Question 2 below
```

In [20]:
```
# Here I am comparing the new converted users to the total number of new users to find
print(df_new_landing.count())
new_converted = df_new_landing[df_new_landing["converted"].str.contains("no") ==False]
print(new_converted.count())
print('The conversion rate for the new landing page is ' + str((33/50)*100)+ "%")
```

```
user_id                   50
group                     50
time_spent_on_the_page    50
converted                 50
language_preferred        50
dtype: int64
user_id                   33
group                     33
time_spent_on_the_page    33
converted                 33
language_preferred        33
dtype: int64
The conversion rate for the new landing page is 66.0%
```

In [21]:
```
# Here I am comparing the old converted users to the total number of old users to find
print(df_old_landing.count())
old_converted = df_old_landing[df_old_landing["converted"].str.contains("no") ==False]
print(old_converted.count())
print('The conversion rate for the old landing page is ' + str((21/50)*100)+ "%")
```

```
user_id                   50
group                     50
time_spent_on_the_page    50
converted                 50
language_preferred        50
dtype: int64
user_id                   21
group                     21
time_spent_on_the_page    21
converted                 21
language_preferred        21
dtype: int64
The conversion rate for the old landing page is 42.0%
```

In [22]:
```
# The conversion rate of the old landing page is 42% and the conversion rate for the n
# Answer to Question 2
# The new landing page has a 14% higher conversion rate.
# Yes, the conversion rateis higher on the new landing page than the old landing page.
```

In [23]:
```
# Question 3
# The best method to test the relationship of 2 catagorical variables is the chi squar
# The null hypothesis is conversion rate is independent of preferred language
# The alternative hypothesis is conversion rate is related to preferred language
```

In [24]:
```
# Here I am creating a numpy array to use in the chi squared test
from scipy.stats import chi2_contingency
df2 = {'yes': [18,21,15], 'no': [16,11,19]}
chi_df = pd.DataFrame(df2, index=['Spanish','English','French'])
print(chi_df)
array = np.array([[18,21,15], [16,11,19]])
chi2 = chi2_contingency(array)
print( 'The p value is ' + str(chi2.pvalue))
```

```
         yes  no
Spanish   18  16
English   21  11
French    15  19
The p value is 0.21298887487543447
```

In [25]:
```
# Since the the p value is greater than our .05 significance level we fail to reject t
# Answer to Question 3
# Since we fail to reject the null hypothesis it means conversion rate and language pr
```

In [26]:
```
# Question 4
# We can use the f_oneway ANOVA test to see if the time spent on the new landing page
# let Mu1,Mu2,Mu3 be Spanish,English, french respectively.
# The null hypothesis is Mu1=Mu2=Mu3
# The alternative hypothesis is that one or more means are unequal.
# Here I am displaying the mean value of time spent on the page by language preferred.
df_new_landing.groupby('language_preferred')['time_spent_on_the_page'].mean()
```

Out[26]:
```
language_preferred
English    6.663750
French     6.196471
Spanish    5.835294
Name: time_spent_on_the_page, dtype: float64
```

In [27]:
```
# Now we must preform the Shapiro-wilks test and lenvene test
```

In [28]:
```
# Shapiro Wilks test
# The null hypothesis is that the distribution is normal
# The alternative hypothesis is that the distribution is not normal
# Here i am preforming the shapiro wilks test to test for normality
sw_p_value = stats.shapiro(df['time_spent_on_the_page'])
print('The p-value is', sw_p_value)
```

```
The p-value is ShapiroResult(statistic=0.9887408018112183, pvalue=0.5643193125724792)
```

In [29]:
```
# Since the p value is higher than the significance level we fail to reject the null h
# The distribution is normal
```

In [30]:
```
# Levene test
# The null hypothesis is that the population variance is equal
# The alternative hypothesis is that the population variance is not equal
# Here I am preforming the levene test to test for equality of variance.
from scipy.stats import levene
statistic, l_p_value = levene( df['time_spent_on_the_page'][df['language_preferred']==
                                df['time_spent_on_the_page'][df['language_preferred
                                df['time_spent_on_the_page'][df['language_preferred
print('The p-value is', l_p_value)
```

```
The p-value is 0.06515086840327314
```

In [31]:
```
# Since the p value is higher than the significance level we fail to reject the null h
# The population variance is equal
```

In [32]:
```
# Since all assumptions were satisfied we can use the f_oneway ANOVA test to see if th
from scipy.stats import f_oneway
test_stat, f_p_value = f_oneway(df.loc[df['language_preferred'] == 'Spanish', 'time_sp
                                df.loc[df['language_preferred'] == 'English', 'time_sper
                                df.loc[df['language_preferred'] == 'French', 'time_spent
print('The p-value is ' + str(f_p_value))
```

```
The p-value is 0.8665610536012648
```

In [33]:
```
# Since our f oneway p value was larger than the significance level of .05, we fail to
# Answer to Question 4
# We can conclude the is not a significant differance in the time spent on the page be
```

In [34]:
```
# My conclusion and business recommendation would be to use the new landing page. The
# Users also spent roughly 25% more time on the new landing page compared to the old p
```