hypothesis-project

September 6, 2023

[2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
```

[3]:
```
df= pd.read_csv('bike_sharing.csv')
df
```

[3]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0000	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0000	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0000	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0000	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0000	0	1	1
…	…	…	…	…	…	…	…	…	…	…	…	…
10881	2012-12-19 19:00:00	4	0	1	1	15.58	19.695	50	26.0027	7	329	336
10882	2012-12-19 20:00:00	4	0	1	1	14.76	17.425	57	15.0013	10	231	241
10883	2012-12-19 21:00:00	4	0	1	1	13.94	15.910	61	15.0013	4	164	168
10884	2012-12-19 22:00:00	4	0	1	1	13.94	17.425	61	6.0032	12	117	129
10885	2012-12-19 23:00:00	4	0	1	1	13.12	16.665	66	8.9981	4	84	88

[10886 rows x 12 columns]

[4]:
```
df.head()
```

[4]:

	datetime	season	holiday	workingday	weather	temp	atemp	\
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	

	humidity	windspeed	casual	registered	count
0	81	0.0	3	13	16
1	80	0.0	8	32	40
2	80	0.0	5	27	32
3	75	0.0	3	10	13
4	75	0.0	0	1	1

[5]:
```
df.shape
```

[5]:
```
(10886, 12)
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  object
 1   season      10886 non-null  int64
 2   holiday     10886 non-null  int64
 3   workingday  10886 non-null  int64
 4   weather     10886 non-null  int64
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64
 10  registered  10886 non-null  int64
 11  count       10886 non-null  int64
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

[7]:
```
df.nunique()
```

[7]:
```
datetime      10886
season            4
holiday           2
```

```
workingday        2
weather           4
temp             49
atemp            60
humidity         89
windspeed        28
casual          309
registered      731
count           822
dtype: int64
```

[8]:
```
df.describe()
```

[8]:

	season	holiday	workingday	weather	temp
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000
mean	2.506614	0.028569	0.680875	1.418427	20.23086
std	1.116174	0.166599	0.466159	0.633839	7.79159
min	1.000000	0.000000	0.000000	1.000000	0.82000
25%	2.000000	0.000000	0.000000	1.000000	13.94000
50%	3.000000	0.000000	1.000000	1.000000	20.50000
75%	4.000000	0.000000	1.000000	2.000000	26.24000
max	4.000000	1.000000	1.000000	4.000000	41.00000

	atemp	humidity	windspeed	casual	registered
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
mean	23.655084	61.886460	12.799395	36.021955	155.552177
std	8.474601	19.245033	8.164537	49.960477	151.039033
min	0.760000	0.000000	0.000000	0.000000	0.000000
25%	16.665000	47.000000	7.001500	4.000000	36.000000
50%	24.240000	62.000000	12.998000	17.000000	118.000000
75%	31.060000	77.000000	16.997900	49.000000	222.000000
max	45.455000	100.000000	56.996900	367.000000	886.000000

	count
count	10886.000000
mean	191.574132
std	181.144454
min	1.000000
25%	42.000000
50%	145.000000
75%	284.000000
max	977.000000

- temp: max 41 degree celsius, avg 20.5 degree celsius
- atemp: max 45 degree celsius, avg 23.65 degree celsius
- humidity: max 100 degree celsius, avg 61.88 degree celisus

[9]:
```
df.isnull().sum()
```

[9]:
```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

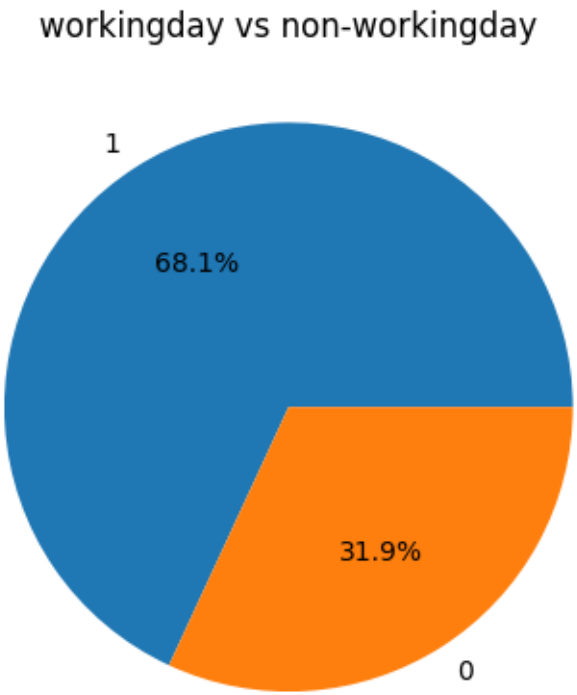
hence their is no null value in the data.

[10]:
```
df['season'].value_counts()
```

[10]:
```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

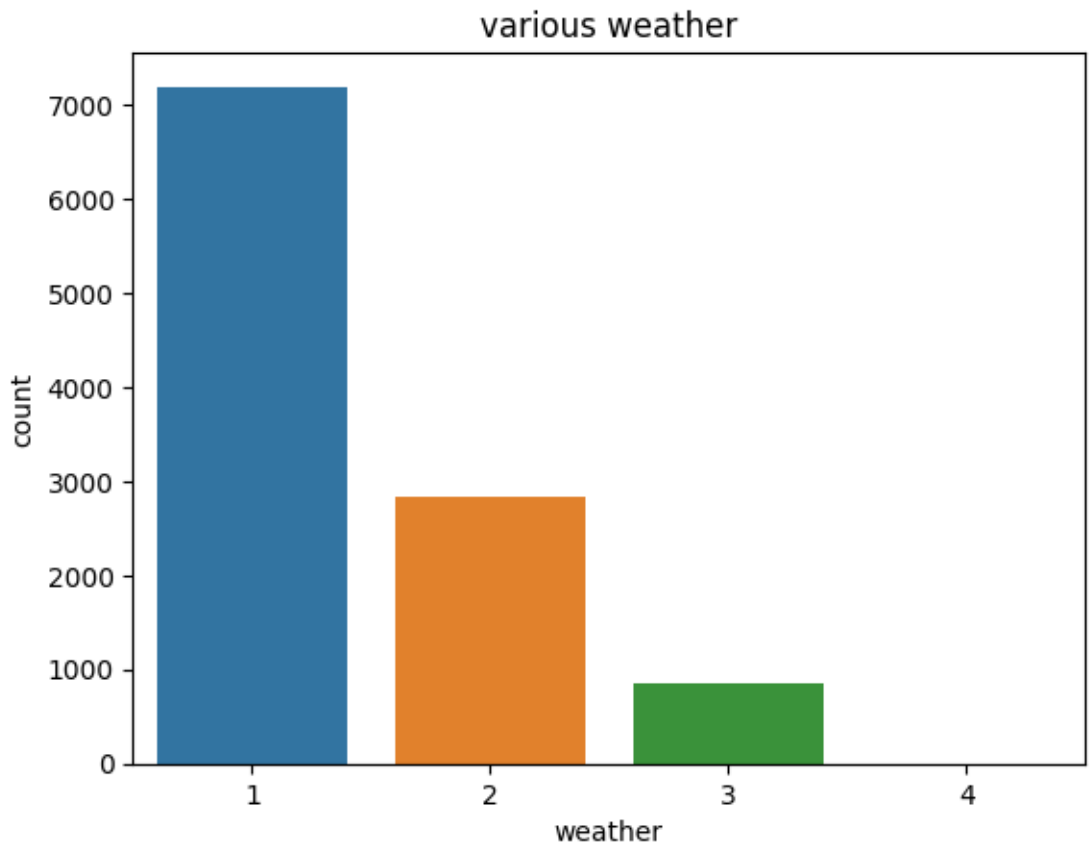
[11]:
```
plt.title('workingday vs non-workingday')
plt.pie(
df['workingday'].value_counts(),
labels = df['workingday'].value_counts().index, autopct = '%1.1f%%')
plt.show()

print(df['workingday'].value_counts())
```



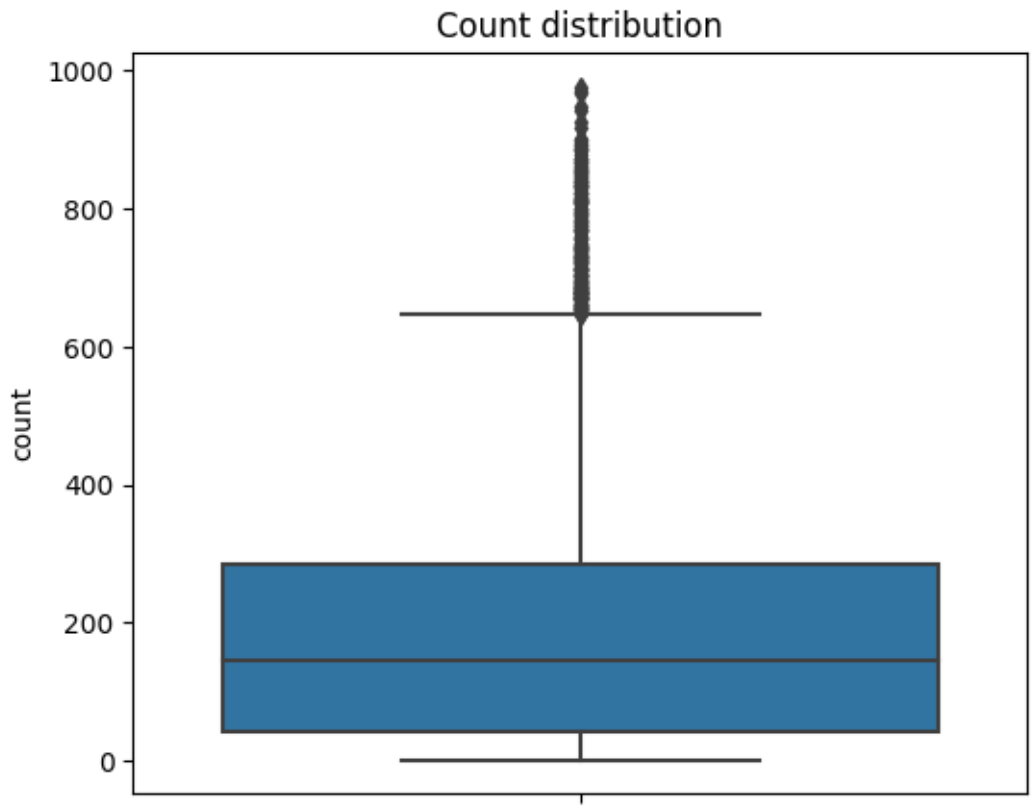

```
1    7412
0    3474
Name: workingday, dtype: int64
```

[12]:
```
plt.title('various weather')
p = sns.countplot(x=df['weather'])
plt.show()
print(df['weather'].value_counts())
```




```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

[13]:
```
plt.figure(figsize=(6,5))
plt.title('Count distribution')
sns.boxplot(data=df, y='count')
plt.show()
```

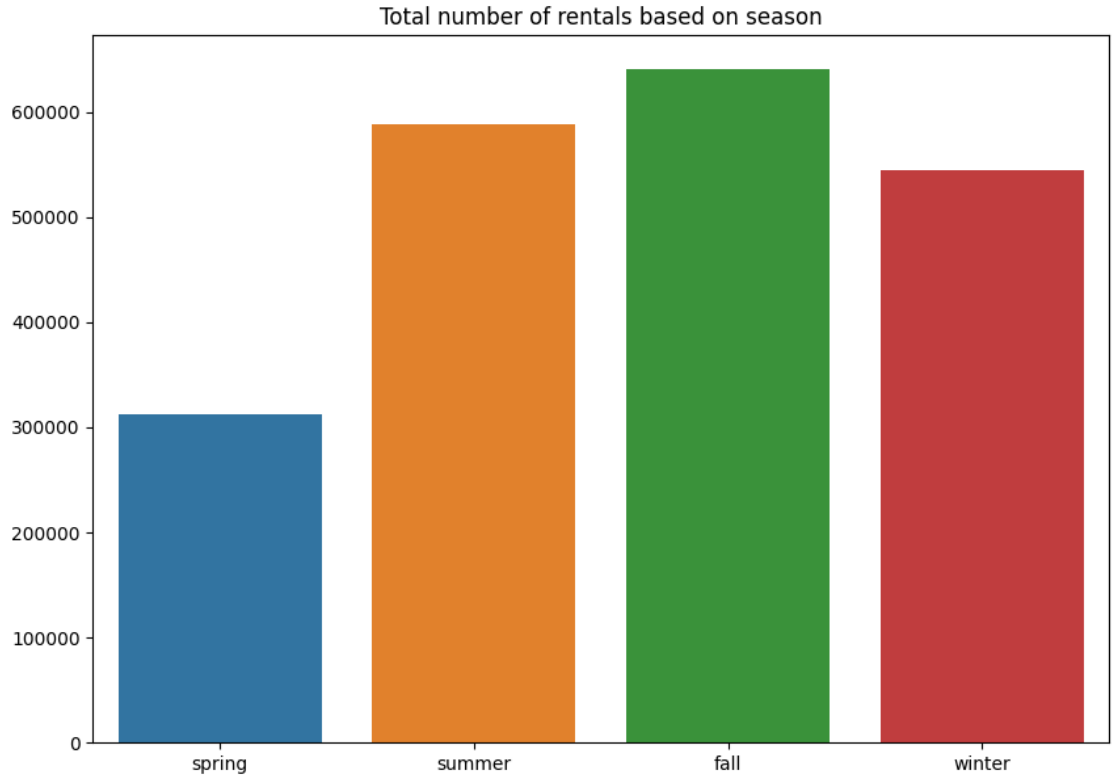


0.1 Number of rentals based on season

[14]:
```
no_of_total_rentals = []
for i in (df['season'].unique()):
  no_of_total_rentals.append(df[df['season']==i]['count'].sum())

no_of_total_rentals = np.array(no_of_total_rentals)
plt.figure(figsize=(10,7))
plt.title('Total number of rentals based on season')
sns.barplot(x=['spring','summer','fall','winter'], y=no_of_total_rentals)
```

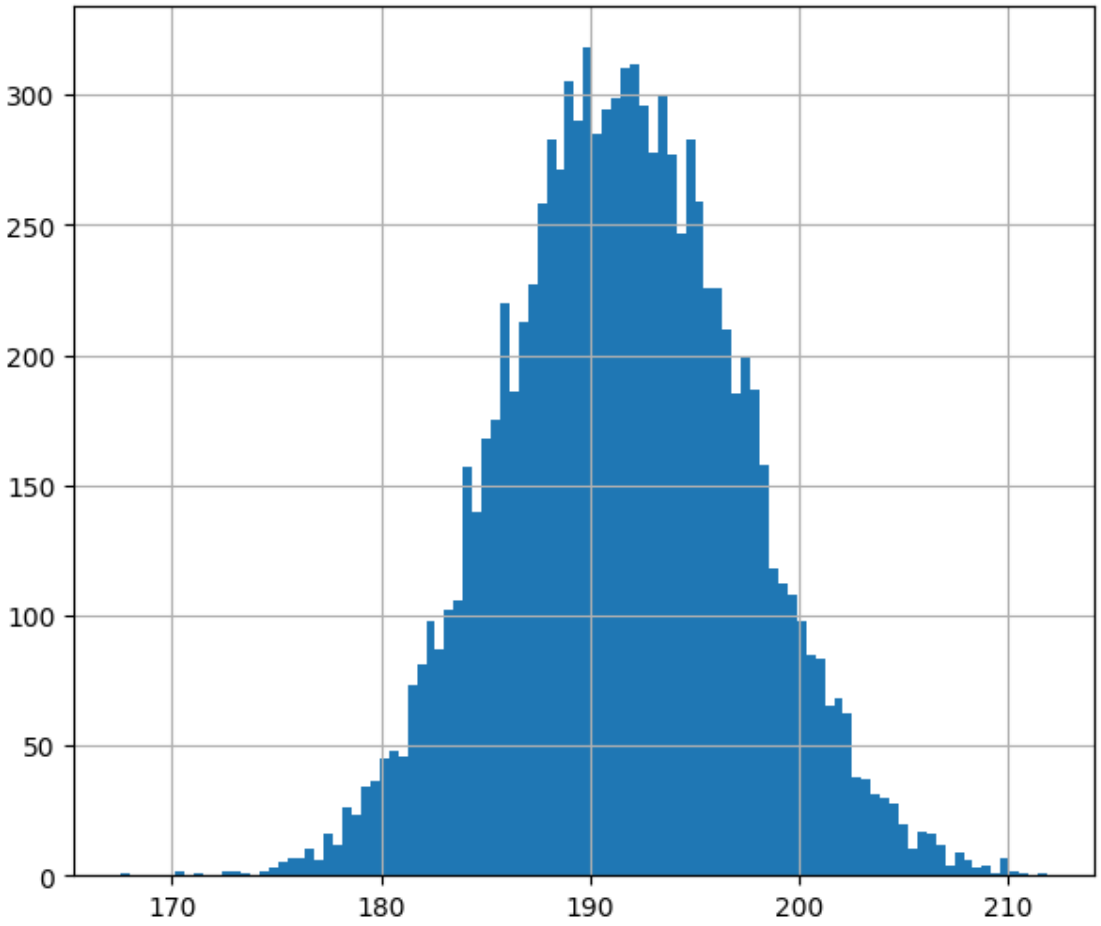
[14]:
```
<Axes: title={'center': 'Total number of rentals based on season'}>
```



- fall has the highest rentals with count 640662 and 30.72%
- followed by summer with count 588282 and 28.2%
- winter with count 544034 and 26%
- spring has the least rental with 14.98% with a count of 312498

0.2 T-TEST

[15]:
```
sample_means = [np.mean(np.random.choice(df['count'], size=1000)) for _ in␣
↪range(10000)]
plt.figure(figsize=(7,6))
plt.hist(sample_means, bins=100)
plt.grid()
plt.show()
```



##T-test on working data vs rental count

1. H0: working day effect on rentals.
2. H1: working day does not effect rentals.

[16]:
```
alpha_value = 0.05

working = df[df['workingday']==1]['count']
non_working = df[df['workingday']==0]['count']

tvalue, pvalue = stats.ttest_ind(non_working, working, equal_var=False)
print(f'tvalue={tvalue}, pvalue={pvalue}')
if alpha_value > pvalue:
  print('Reject Null hypothesis')
else:
  print('Fail to reject Null hypothesis')
```

```
tvalue=-1.2362580418223226, pvalue=0.21640312280695098
Fail to reject Null hypothesis
```

T-Test test Inference: 1. Since the P-Value of the test is 0.216, hence the P-Value is greater than the alpha value. Therefore we fail to reject Null hypothesis.

##ANOVA test on rental count, weather, season

[17]:
```
w1 = df[df['weather'] == 1]['count']
w2 = df[df['weather'] == 2]['count']
w3 = df[df['weather'] == 3]['count']
w4 = df[df['weather'] == 4]['count']
```

[18]:
```
print('''
h0: No. of cycle rented is similar in different weather
h1: No. of cycle rented is different in different weather
''')

alpha = 0.05

F, P_value = stats.f_oneway(w1, w2, w3, w4)
print(f'fstatics:{F}, pvalue:{P_value}')
print()
if alpha > P_value:
  print('Reject The null hypothesis')
else:
  print('Accept the null hypothesis')
```

```
h0: No. of cycle rented is similar in different weather
h1: No. of cycle rented is different in different weather

fstatics:65.53024112793271, pvalue:5.482069475935669e-42

Reject The null hypothesis
```

ANOVA test Inference: 1. The P-Value of the test is 5.292326890121564e-137(approx 0), hence reject the Null Hypothesis.

[19]:
```
s1 = df[df['season'] == 1]['count']
s2 = df[df['season'] == 2]['count']
s3 = df[df['season'] == 3]['count']
s4 = df[df['season'] == 4]['count']
```

[20]:
```
print('''
h0: No. of cycle rented is similar in different season
h1: No. of cycle rented is different in different season
''')

alpha = 0.05

F, P_value = stats.f_oneway(s1, s2, s3, s4)
```

```
print(f'fstatics:{F}, pvalue:{P_value}')
print()

if alpha > P_value:
  print('Reject The null hypothesis')
else:
  print('Accept the null hypothesis')
```

```
h0: No. of cycle rented is similar in different season
h1: No. of cycle rented is different in different season

fstatics:236.94671081032106, pvalue:6.164843386499654e-149

Reject The null hypothesis
```

ANOVA test Inference: 1. The p-value of the test is 6.164843386499654e-149(almost 0), hence reject the Null Hypothesis.

0.3 CHI-SQUARE TEST ON BASIS OF WEATHER AND MONTH

CHI-SQUARE TEST occures when there is categorical to categorical relationship.

[21]:
```
print('''
h0: weather is independent on season
h1: weather is dependent on season
''')

alpha = 0.05

chi_square_val, p_value,*a = stats.chi2_contingency(pd.
 ↪crosstab(df['weather'],df['season']))
print(chi_square_val, p_value)
print()

if p_value > alpha:
  print('weather is independent of season hence fail to reject the null␣
 ↪hypothesis')
else:
  print('weather is dependent on season hence reject the null hypothesis')
```

```
h0: weather is independent on season
h1: weather is dependent on season

49.15865559689363 1.5499250736864862e-07

weather is dependent on season hence reject the null hypothesis
```