```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm, chi2, f
from scipy.stats import ttest_ind, f_oneway, kruskal
from scipy.stats import chisquare, chi2_contingency
from scipy.stats import kstest
from statsmodels.distributions.empirical_distribution import ECDF

from IPython.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
```

```
<IPython.core.display.HTML object>
```

```
import matplotlib_inline
matplotlib_inline.backend_inline.set_matplotlib_formats('svg')

df_yulu =
pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/asset
s/000/001/428/original/bike_sharing.csv?1642089089")

df_yulu.head()
```

```
              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

df_yulu.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
```

```
df_yulu.describe()
```

	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
```

```
df_yulu.nunique()
```

```
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
```

```
df_yulu.columns
```

```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather',
'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered',
'count'],
      dtype='object')
```

```
df_yulu.shape
```

```
(10886, 12)
```

```
df_yulu.isnull().sum()
```

```
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
```

```
df_yulu.isna().sum()
```

```
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
```

```
df_yulu.head()
```

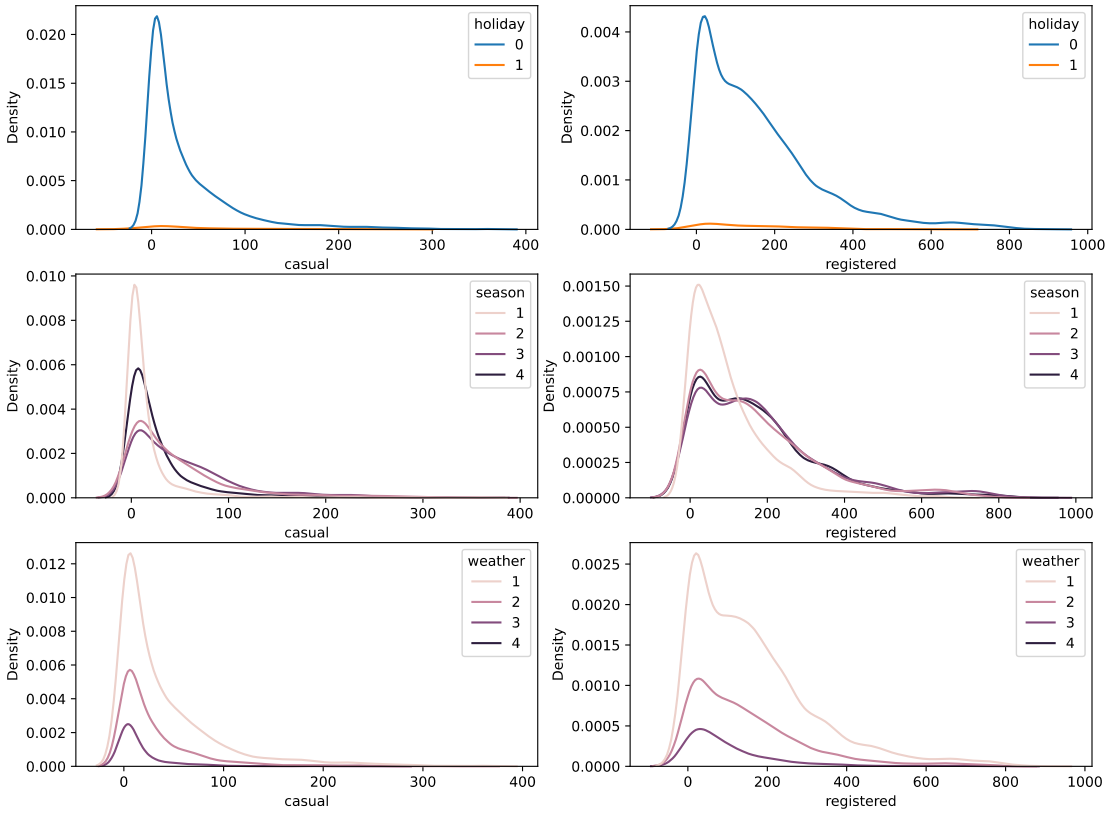
```
              datetime  season  holiday  workingday  weather  temp
atemp  \
0  2011-01-01 00:00:00       1        0           0        1  9.84
14.395
1  2011-01-01 01:00:00       1        0           0        1  9.02
13.635
2  2011-01-01 02:00:00       1        0           0        1  9.02
13.635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

```
plt.figure(figsize=(13,10))
plt.subplot(321)
sns.kdeplot(x=df_yulu["casual"],hue=df_yulu["holiday"])
plt.subplot(322)
sns.kdeplot(x=df_yulu["registered"],hue=df_yulu["holiday"])
plt.subplot(323)
sns.kdeplot(x=df_yulu["casual"],hue=df_yulu["season"])
```

```
plt.subplot(324)
sns.kdeplot(x=df_yulu["registered"],hue=df_yulu["season"])
plt.subplot(325)
sns.kdeplot(x=df_yulu["casual"],hue=df_yulu["weather"])
plt.subplot(326)
sns.kdeplot(x=df_yulu["registered"],hue=df_yulu["weather"])
```

```
/Users/mrunmay/opt/anaconda3/lib/python3.9/site-packages/seaborn/
distributions.py:316: UserWarning: Dataset has 0 variance; skipping
density estimate. Pass `warn_singular=False` to disable this warning.
  warnings.warn(msg, UserWarning)
/Users/mrunmay/opt/anaconda3/lib/python3.9/site-packages/seaborn/distr
ibutions.py:316: UserWarning: Dataset has 0 variance; skipping density
estimate. Pass `warn_singular=False` to disable this warning.
  warnings.warn(msg, UserWarning)
```

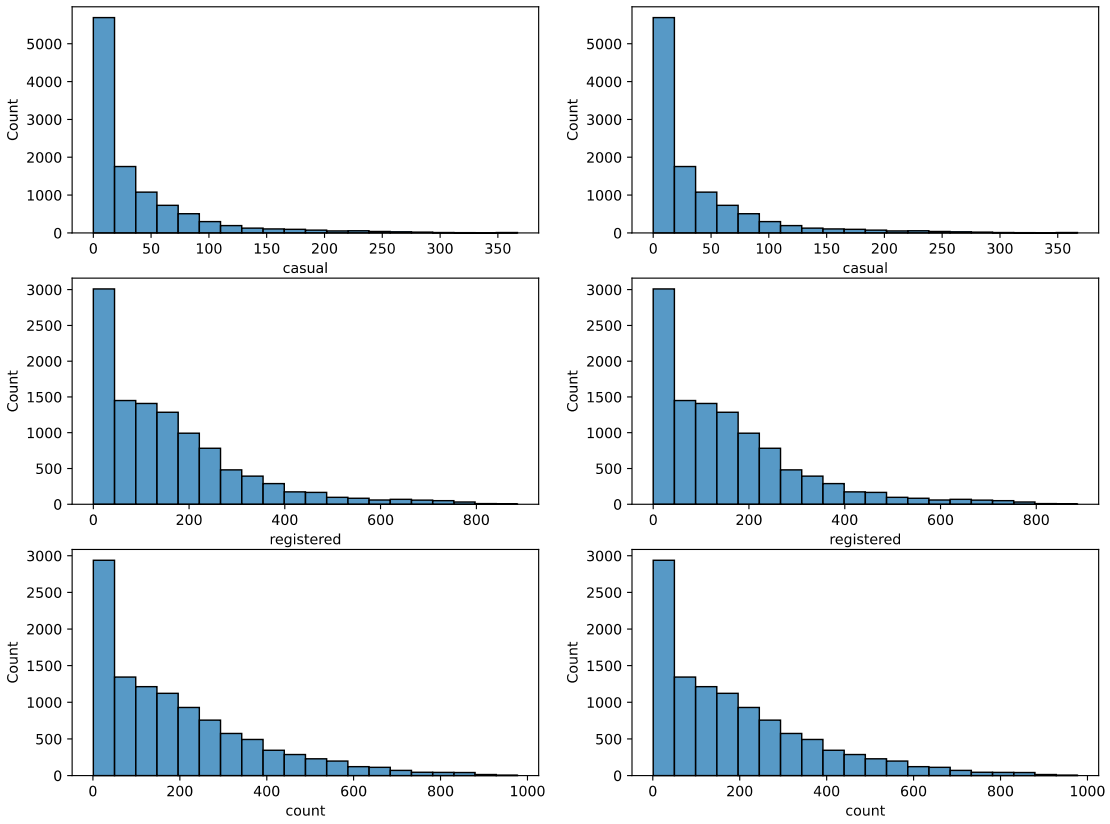
```
<AxesSubplot:xlabel='registered', ylabel='Density'>
```




```
plt.figure(figsize=(13,10))
plt.subplot(321)
sns.histplot(df_yulu["casual"],bins=20)
plt.subplot(322)
sns.histplot(df_yulu["casual"],bins=20)
plt.subplot(323)
sns.histplot(df_yulu["registered"],bins=20)
plt.subplot(324)
```

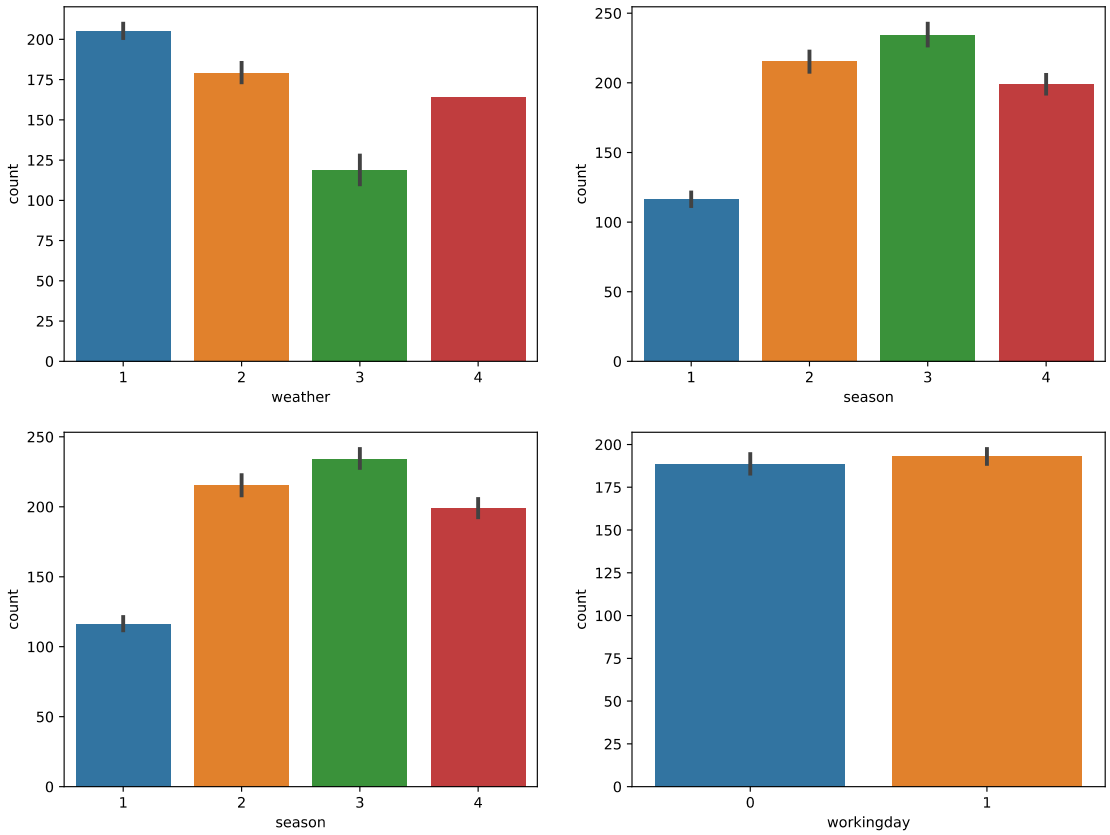
```
sns.histplot(df_yulu["registered"],bins=20)
plt.subplot(325)
sns.histplot(df_yulu["count"],bins=20)
plt.subplot(326)
sns.histplot(df_yulu["count"],bins=20)
```

```
<AxesSubplot:xlabel='count', ylabel='Count'>
```



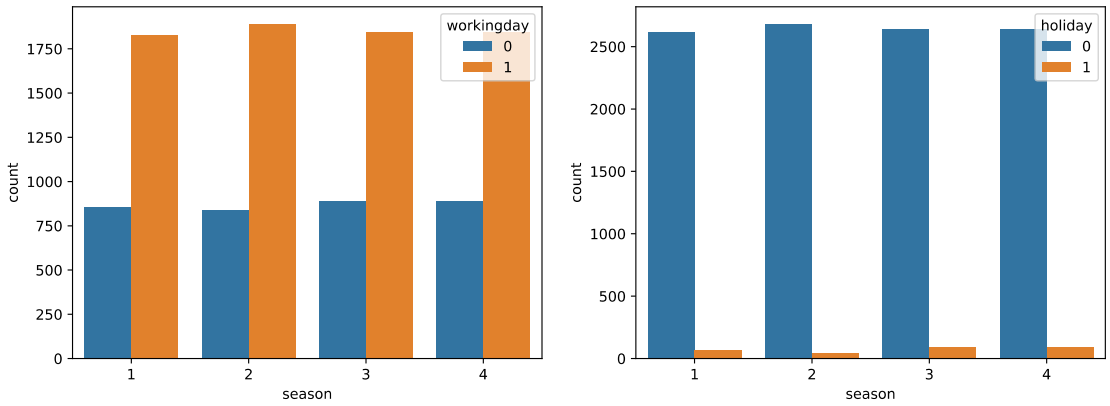

```
plt.figure(figsize=(13,10))
plt.subplot(221)
sns.barplot(x="weather", y="count", data=df_yulu, estimator=np.mean)
plt.subplot(222)
sns.barplot(x="season", y="count", data=df_yulu, estimator=np.mean)
plt.subplot(223)
sns.barplot(x="season", y="count", data=df_yulu, estimator=np.mean)
plt.subplot(224)
sns.barplot(x="workingday", y="count", data=df_yulu,
estimator=np.mean)
```

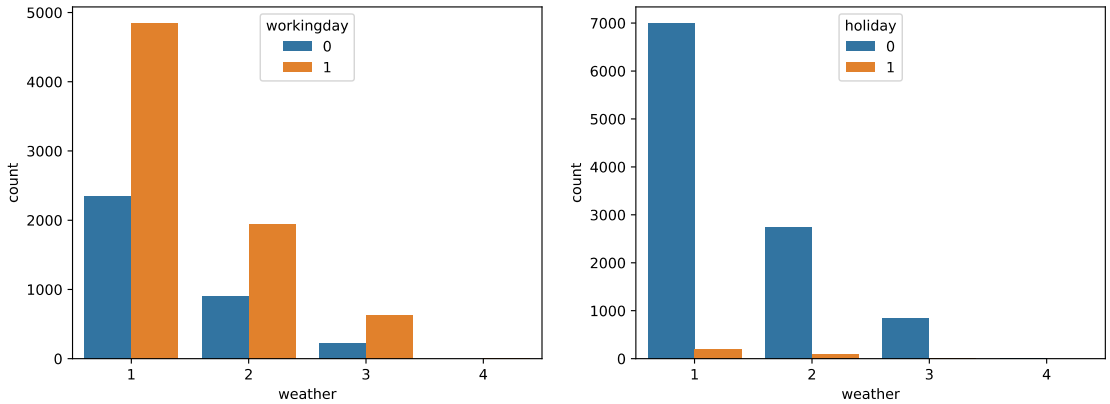
```
<AxesSubplot:xlabel='workingday', ylabel='count'>
```




```
plt.figure(figsize=(13,10))
plt.subplot(221)
sns.countplot(data=df_yulu, x="season", hue="workingday", dodge=True)
plt.subplot(222)
sns.countplot(data=df_yulu, x="season", hue="holiday", dodge=True)
plt.subplot(223)
sns.countplot(data=df_yulu, x="weather", hue="workingday", dodge=True)
plt.subplot(224)
sns.countplot(data=df_yulu, x="weather", hue="holiday", dodge=True)
```

```
<AxesSubplot:xlabel='weather', ylabel='count'>
```

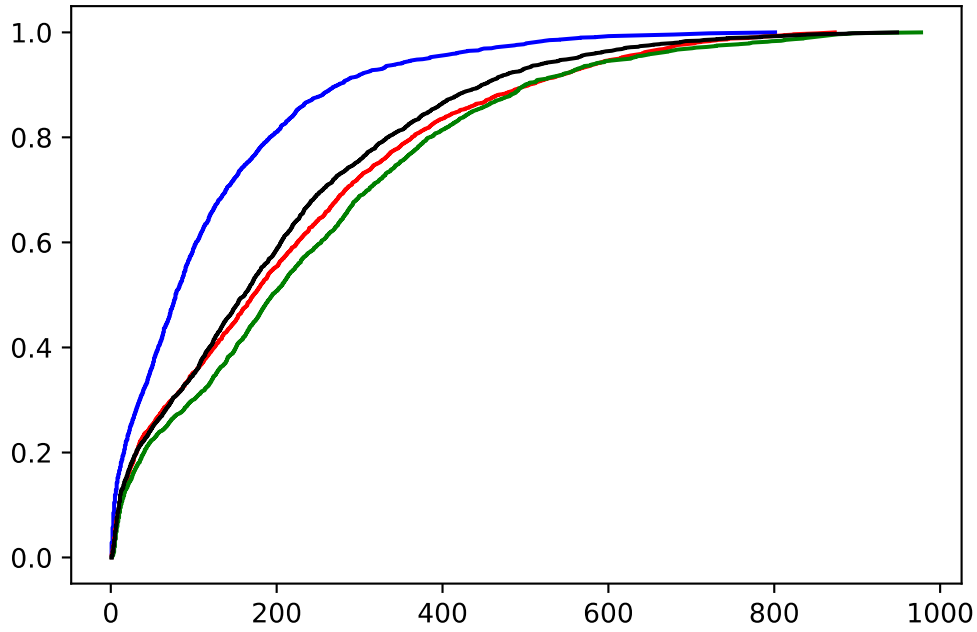





```
#Alpha is 0.05
df_yulu_working = df_yulu[df_yulu["workingday"]==1]

df_yulu_Not_working = df_yulu[df_yulu["workingday"]==0]

ttest_ind(df_yulu_working["count"],df_yulu_Not_working["count"])

Ttest_indResult(statistic=1.2096277376026694,
pvalue=0.22644804226361348)

df_yulu_season_a = df_yulu[df_yulu["season"] == 1]["count"]
df_yulu_season_b = df_yulu[df_yulu["season"] == 2]["count"]
df_yulu_season_c = df_yulu[df_yulu["season"] == 3]["count"]
df_yulu_season_d = df_yulu[df_yulu["season"] == 4]["count"]

#H0 = season has no effect on the no of cycles rented
f_oneway(df_yulu_season_a,df_yulu_season_b,df_yulu_season_c,df_yulu_se
ason_d)

F_onewayResult(statistic=236.94671081032106,
pvalue=6.164843386499654e-149)

e1 = ECDF(df_yulu_season_a)
e2 = ECDF(df_yulu_season_b)
e3 = ECDF(df_yulu_season_c)
e4 = ECDF(df_yulu_season_d)
```

```
plt.plot(e1.x, e1.y,c='b')
plt.plot(e2.x, e2.y,c='r')
plt.plot(e3.x, e3.y,c='g')
plt.plot(e4.x, e4.y,c='k')
```

```
[<matplotlib.lines.Line2D at 0x7fafb233ed00>]
```




```
#P-value is very low means we can reject null hypothesis and consider
that seasons has an effect on the no of cycles rented
```

```
df_yulu_weather_a = df_yulu[df_yulu["weather"] == 1]["count"]
df_yulu_weather_b = df_yulu[df_yulu["weather"] == 2]["count"]
df_yulu_weather_c = df_yulu[df_yulu["weather"] == 3]["count"]
df_yulu_weather_d = df_yulu[df_yulu["weather"] == 4]["count"]
```

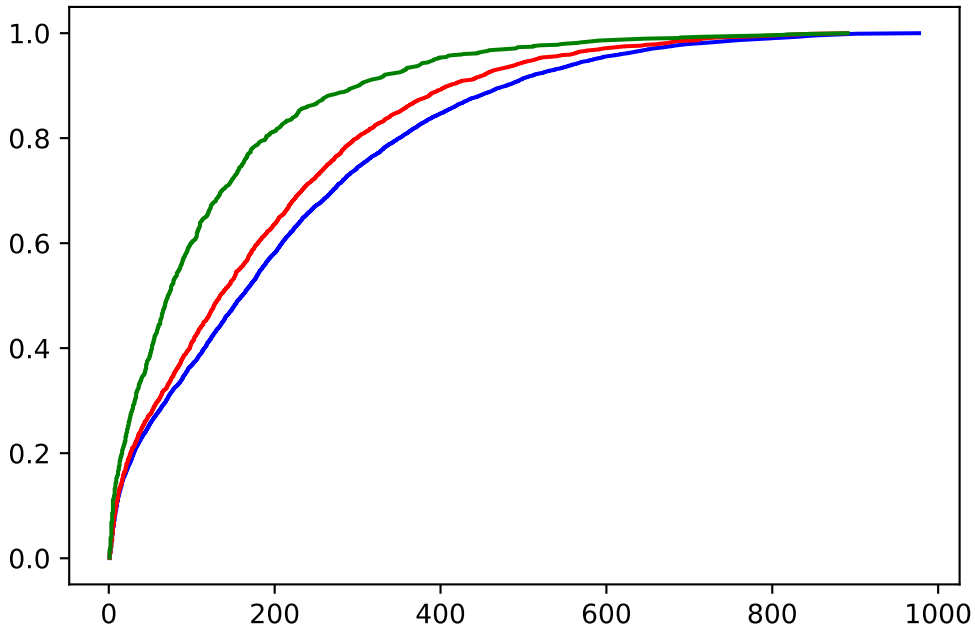
```
#H0 = weather has no effect on the no of cycles rented
f_oneway(df_yulu_weather_a,df_yulu_weather_b,df_yulu_weather_c,df_yulu
_weather_d)
```

```
F_onewayResult(statistic=65.53024112793271, pvalue=5.482069475935669e-
42)
```

```
e1 = ECDF(df_yulu_weather_a)
e2 = ECDF(df_yulu_weather_b)
e3 = ECDF(df_yulu_weather_c)
e4 = ECDF(df_yulu_weather_d)
```

```
plt.plot(e1.x, e1.y,c='b')
plt.plot(e2.x, e2.y,c='r')
plt.plot(e3.x, e3.y,c='g')
plt.plot(e4.x, e4.y,c='k')
```

```
[<matplotlib.lines.Line2D at 0x7fafb1669be0>]
```




```
#P-value is very low means we can reject null hypothesis and consider
that weather has an effect on the no of cycles rented

WS = pd.crosstab(index = df_yulu["weather"],columns =
df_yulu["season"])

WS
```

season	1	2	3	4
weather				
1	1759	1801	1930	1702
2	715	708	604	807
3	211	224	199	225
4	1	0	0	0

```
#H0 = weather and season are independent
chi2_contingency(WS)
```

```
(49.15865559689363,
 1.5499250736864862e-07,
 9,
 array([[1.77454639e+03, 1.80559765e+03, 1.80559765e+03,
1.80625831e+03],
        [6.99258130e+02, 7.11493845e+02, 7.11493845e+02,
7.11754180e+02],
        [2.11948742e+02, 2.15657450e+02, 2.15657450e+02,
2.15736359e+02],
        [2.46738931e-01, 2.51056403e-01, 2.51056403e-01, 2.51148264e-
01]]))
```

```
#P-value is very less so we can reject null hypothesis means weather and season are dependant
```

```
plt.figure(figsize=(13,10))
sns.heatmap(data=df_yulu.corr(),cmap="Blues",annot=True)
```

```
<AxesSubplot:>
```