Bike 데이터 분석 및 시각화 과제

```
In [1]: import pandas as pd
```

```
In [5]: train = pd.read_csv("data/biketrain.csv", parse_dates=['datetime'])
        test = pd.read_csv("data/biketest.csv", parse_dates=['datetime'])
```

데이터 행, 열

```
In [6]: train.shape

Out[6]: (10886, 12)
```

```
In [7]: test.shape

Out[7]: (6493, 9)
```

결측치 확인

In [8]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
datetime      10886 non-null datetime64[ns]
season        10886 non-null int64
holiday       10886 non-null int64
workingday    10886 non-null int64
weather       10886 non-null int64
temp          10886 non-null float64
atemp         10886 non-null float64
humidity      10886 non-null int64
windspeed     10886 non-null float64
casual        10886 non-null int64
registered    10886 non-null int64
count         10886 non-null int64
dtypes: datetime64[ns](1), float64(3), int64(8)
memory usage: 1020.6 KB
```

In [9]:

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6493 entries, 0 to 6492
Data columns (total 9 columns):
datetime      6493 non-null datetime64[ns]
season        6493 non-null int64
holiday       6493 non-null int64
workingday    6493 non-null int64
weather       6493 non-null int64
temp          6493 non-null float64
atemp         6493 non-null float64
humidity      6493 non-null int64
windspeed     6493 non-null float64
dtypes: datetime64[ns](1), float64(3), int64(5)
memory usage: 456.6 KB
```

컬럼명 확인

In [10]:

```
train.columns
```

Out[10]:

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

In [11]:

```
test.columns
```

Out[11]:

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

시각화

In [14]:

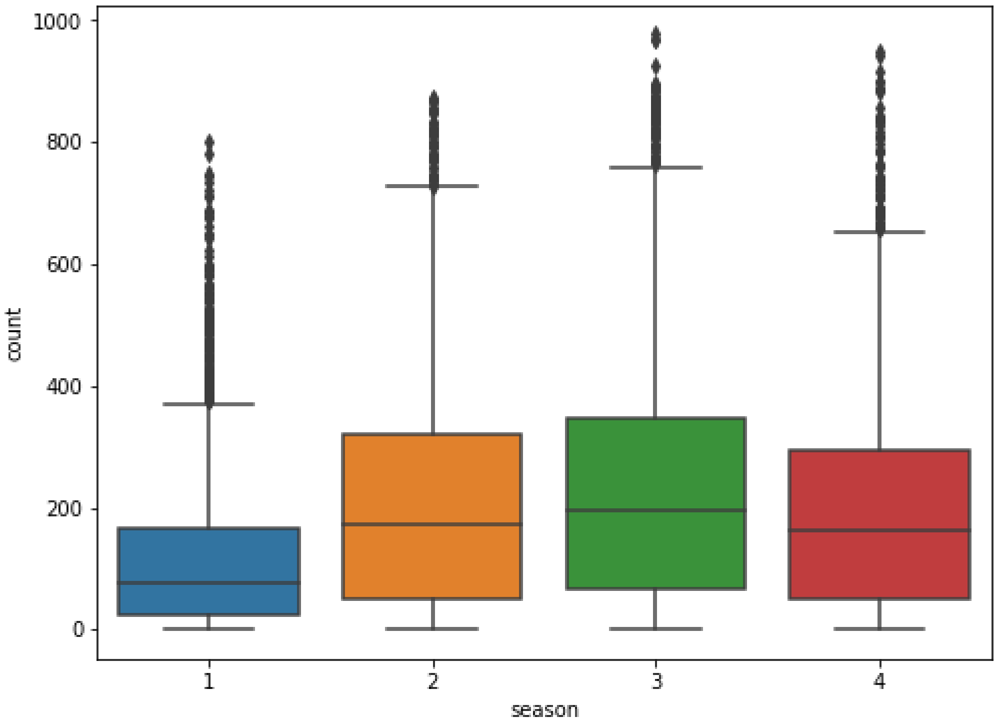
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [16]:

```
plt.figure(figsize=(8,6))
sns.boxplot(x='season', y='count', data=train)
```

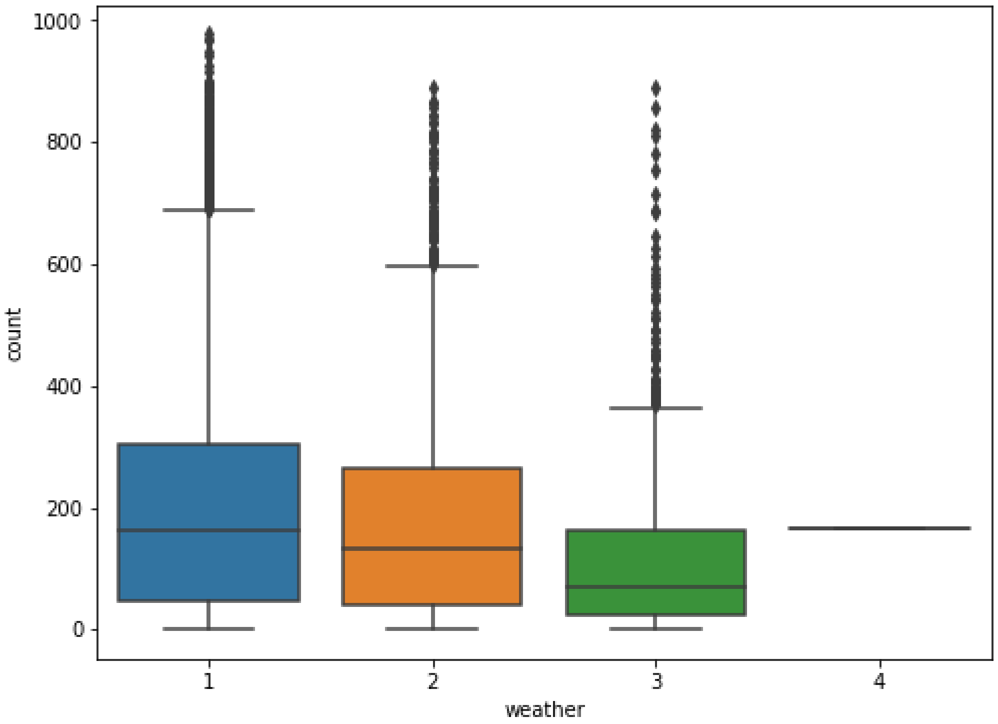
Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e55d02f98>
```



In [18]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='weather', y='count', data=train)
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21e560e2a58>
```

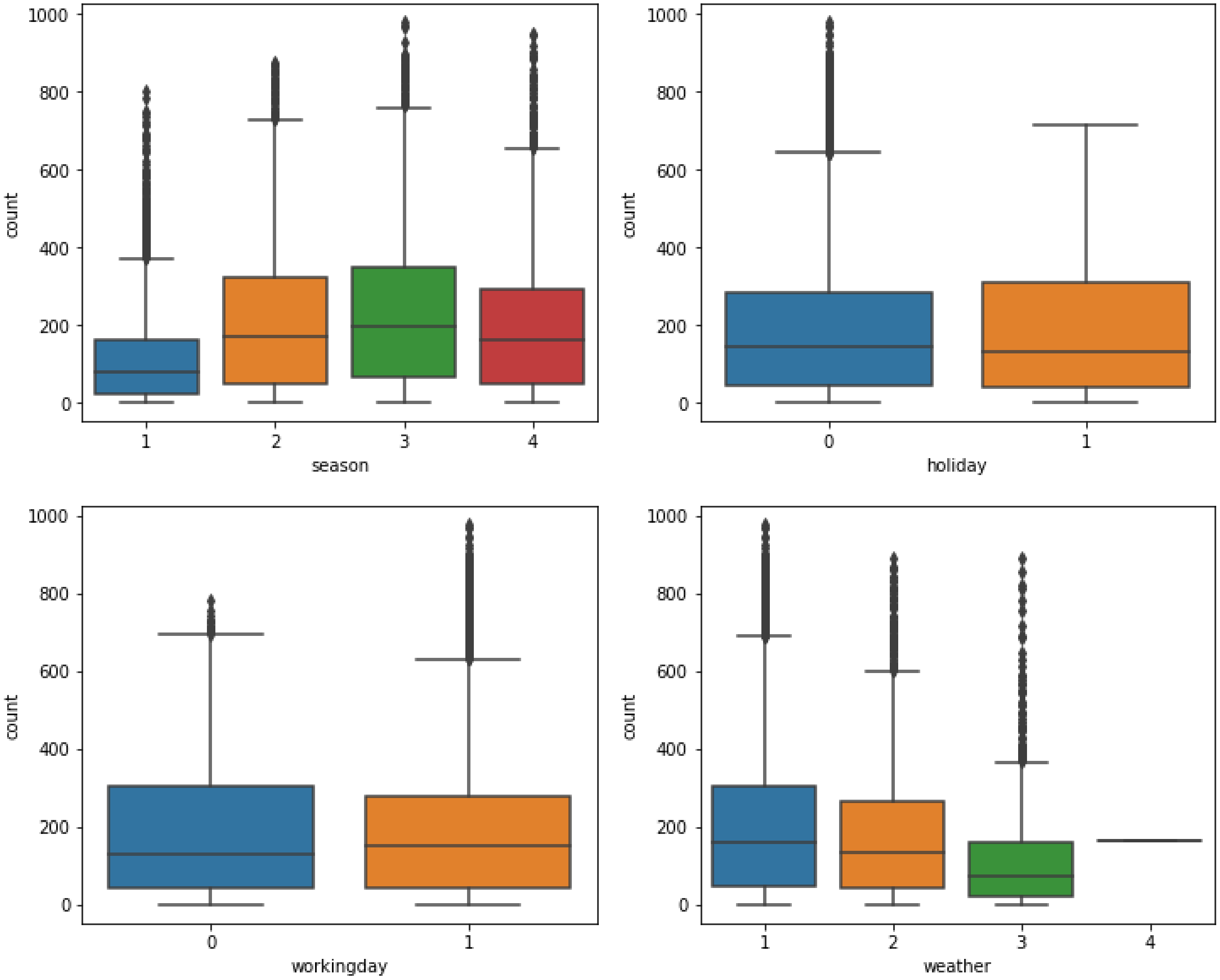


In [22]:

```
plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
sns.boxplot(x='season', y='count', data=train)
plt.subplot(2,2,2)
sns.boxplot(x='holiday', y='count', data=train)
plt.subplot(2,2,3)
sns.boxplot(x='workingday', y='count', data=train)
plt.subplot(2,2,4)
sns.boxplot(x='weather', y='count', data=train)
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e566434a8>
```



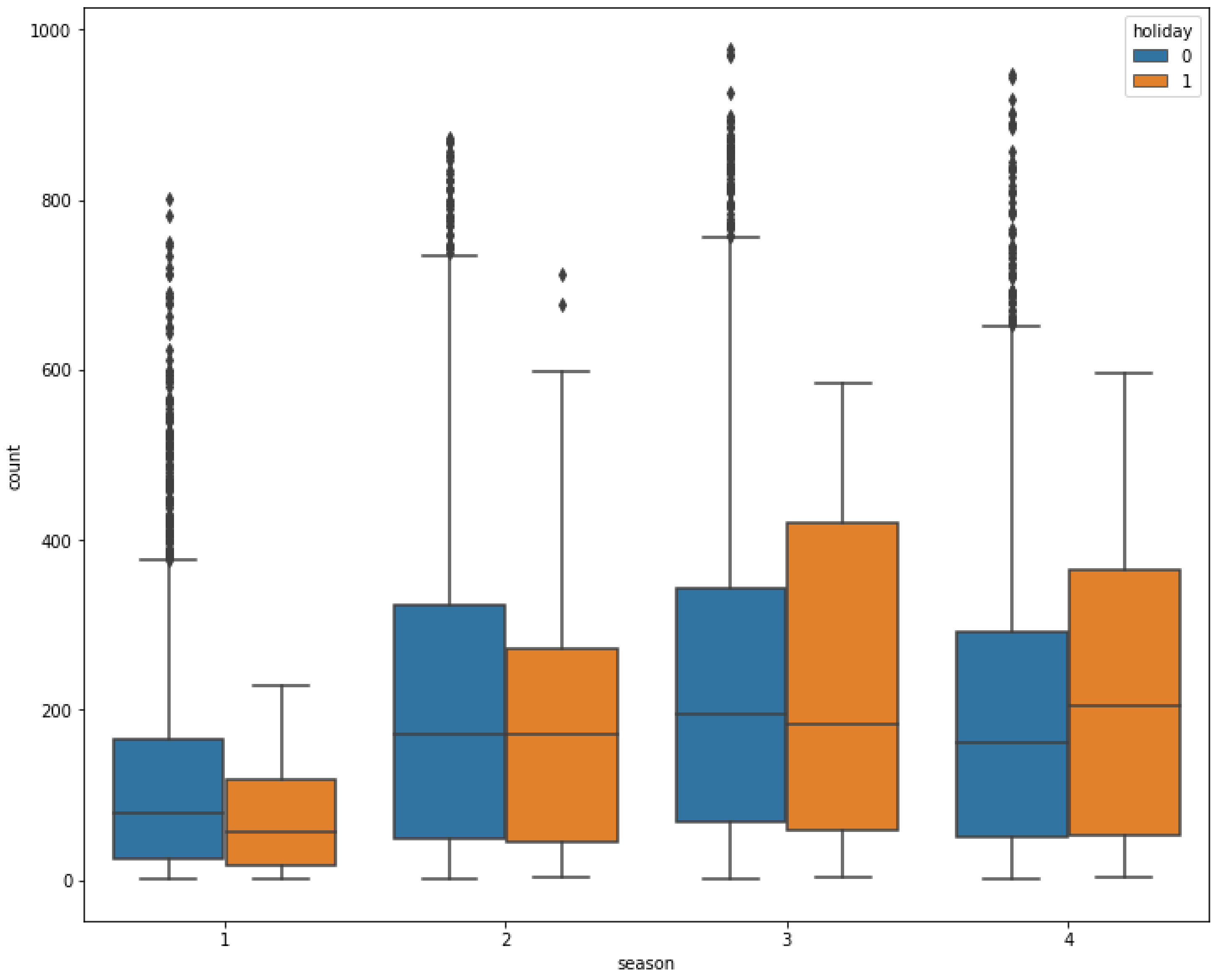
(다) season(1,2,3,4) 별 휴일인 날과 아닌 날을 비교해서 boxplot을 그려보자.

In [24]:

```
plt.figure(figsize=(12,10))
sns.boxplot(x='season', y='count', hue='holiday', data=train)
```

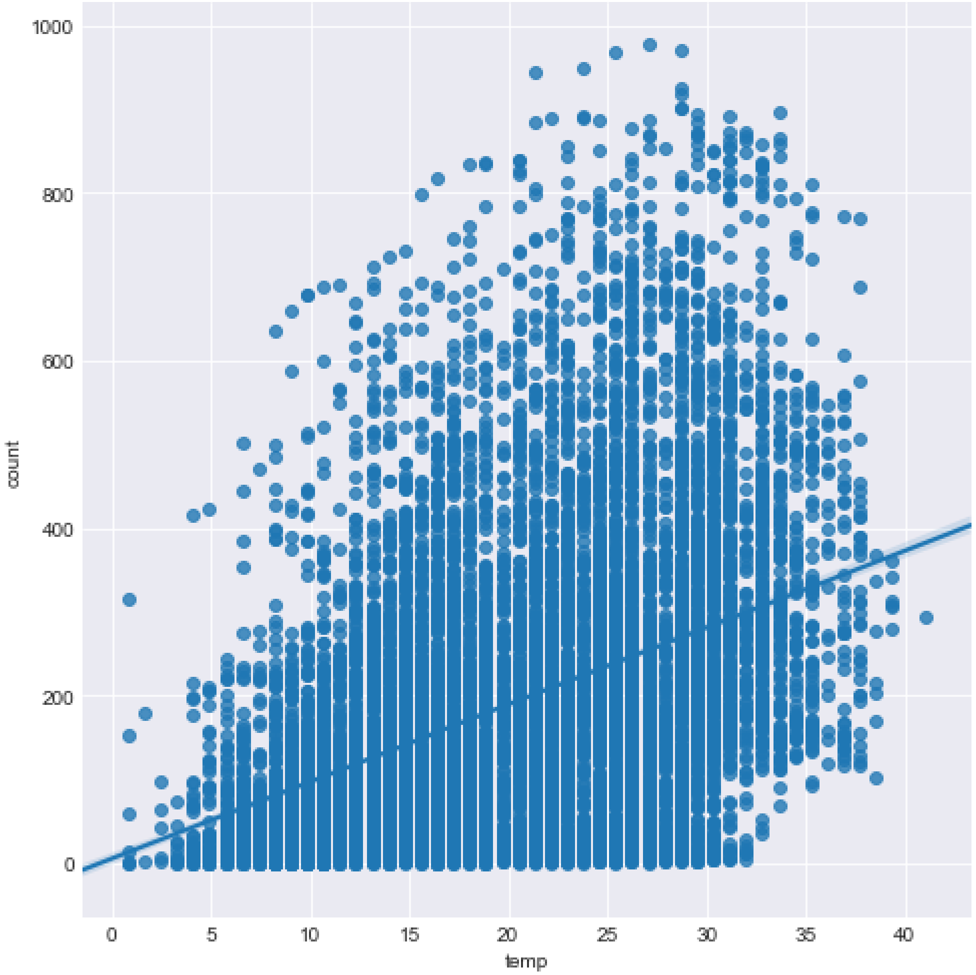
Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e569803c8>
```

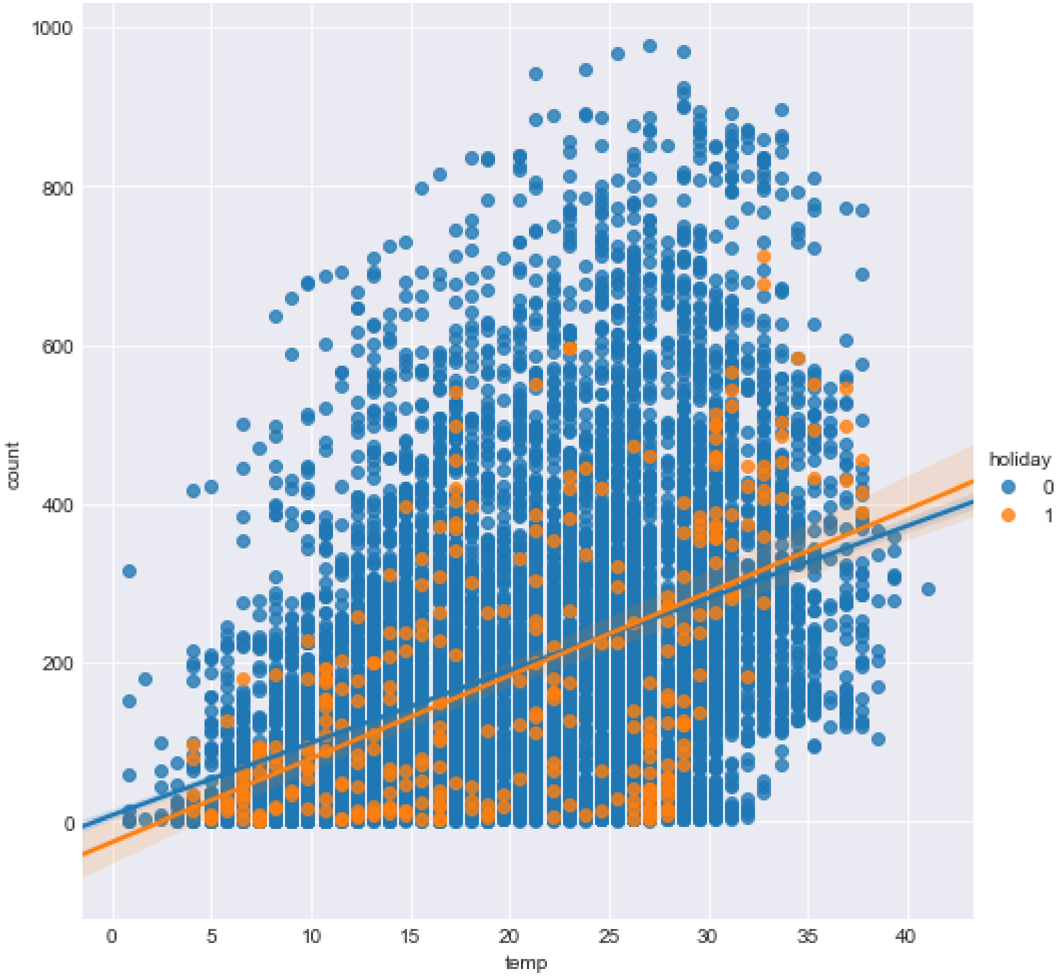


(라) 온도(temp)와 count의 관계를 알아보기 위해 seaborn의 lmplot을 그려보자.

In [25]:
```
sns.set_style('darkgrid')
sns.lmplot(x="temp", y='count',data=train, size=7)
plt.show()
```



In [26]:
```
sns.set_style('darkgrid')
sns.lmplot(x="temp", y='count',hue='holiday', data=train, size=7)
plt.show()
```



In [27]:
```
train.columns
```

Out[27]:
```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```

In [61]:

```
tr_sub = train.copy()
tr_sub.loc[tr_sub['season']==1,'season'] = 'spring'
tr_sub.loc[tr_sub['season']==2,'season']= 'summer'
tr_sub.loc[tr_sub['season']==3,'season'] = 'autumn'
tr_sub.loc[tr_sub['season']==4,'season'] = 'winter'
#tr_sub['season']=tr_sub['season'].astype("object")
```

In [62]:

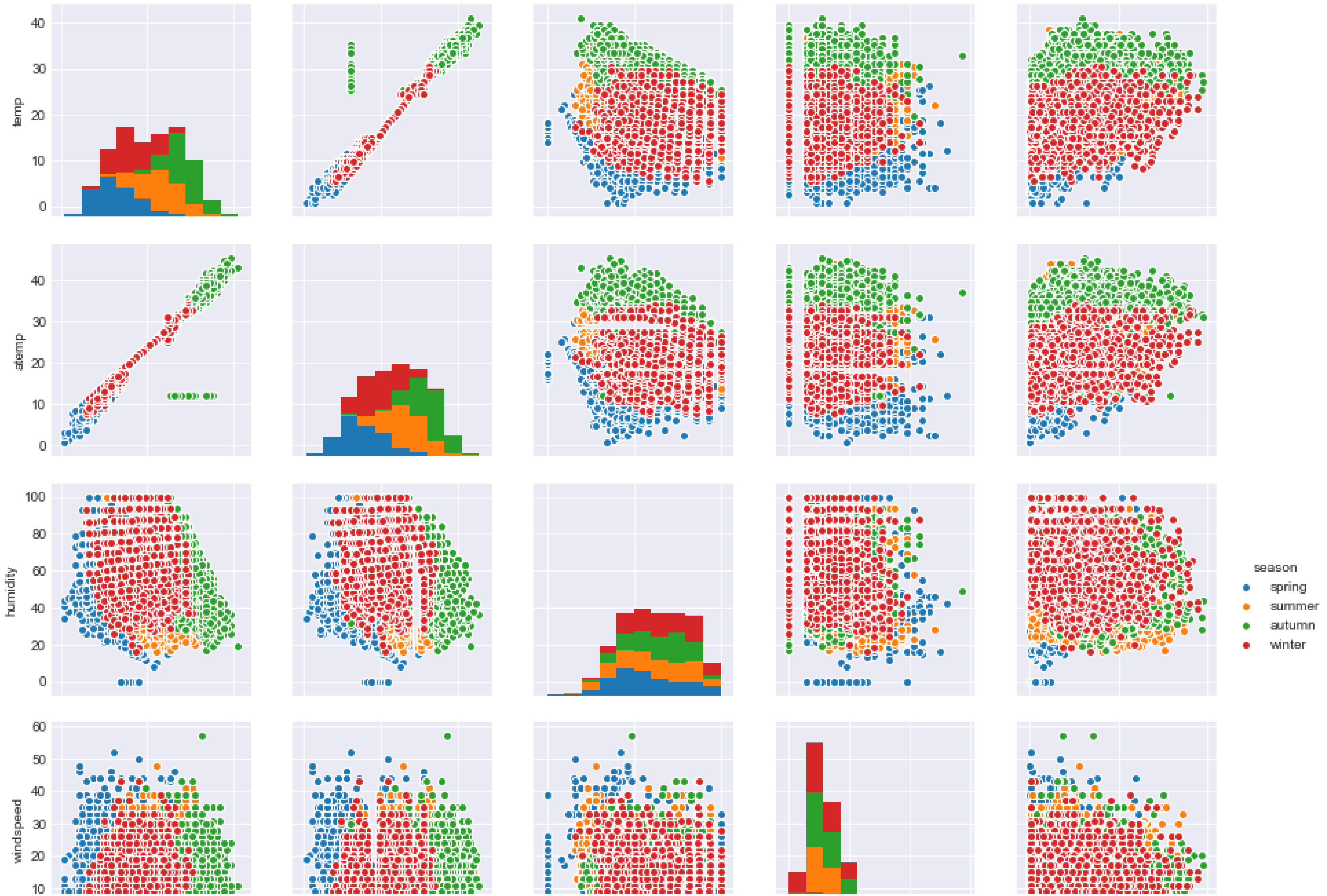
```
tr_sub['season'].value_counts()
```

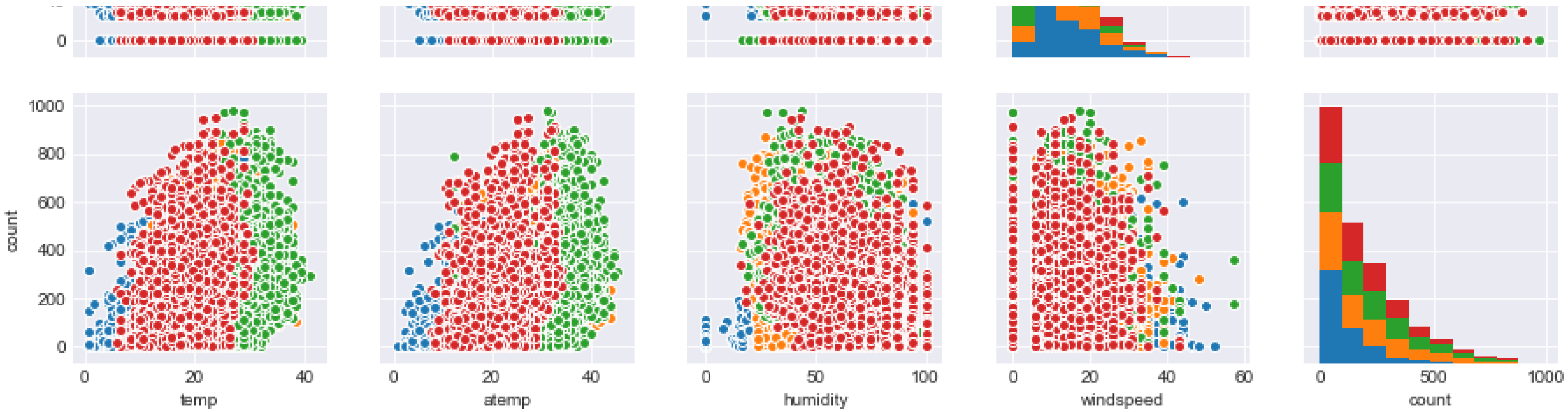
Out[62]:

```
winter    2734
autumn    2733
summer    2733
spring    2686
Name: season, dtype: int64
```

In [65]:
```
sel = ['temp', 'atemp', 'humidity', 'windspeed','count','season']
tr_sub = tr_sub[sel]
sns.pairplot(tr_sub, hue='season')
```

Out[65]:
```
<seaborn.axisgrid.PairGrid at 0x21e69e26470>
```




0
1000
800
600
400
200
0
count
0
20
40
temp
0
20
40
atemp
0
50
100
humidity
0
20
40
60
windspeed
0
500
1000
count

In [ ]: