

```
In [4]: import datetime as dt, numpy as np, pandas as pd
from pandas import DataFrame as DF
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
from django_pandas.io import read_frame
from utils import *
import os
from os.path import join
import calendar
pwd=os.getcwd()

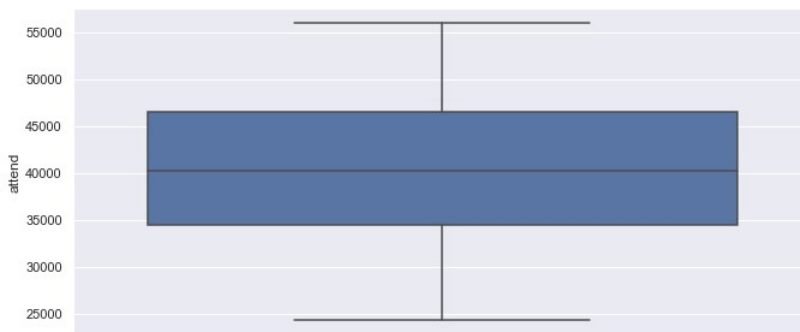
from sklearn.model_selection import train_test_split
import statsmodels.api as sm # statistical models (including regression)
import statsmodels.formula.api as smf # R-like model specification
```

```
In [5]: %matplotlib inline
plt.style.use('ggplot')
matplotlib.rcParams['figure.figsize'] = (11, 5)
matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['axes.titlesize'] = 20
```

```
In [36]: pwd=os.getcwd()
path = join(pwd, 'z_data\dodgers.csv')
df=pd.read_csv(path)
df['year']=2012
month_dict=dict((v.upper(),k) for k,v in enumerate(calendar.month_abbr))
month_dict.pop('')
df['MON']=df['month']
df['month']=df['month'].map(month_dict)
df['gamedate']=pd.to_datetime(df[['year','month','day']]).dt.date
df['dow']=pd.to_datetime(df['gamedate']).dt.weekday
df['bh_bit']=df['bobblehead'].map(lambda val: 1 if val=="YES" else 0)
```

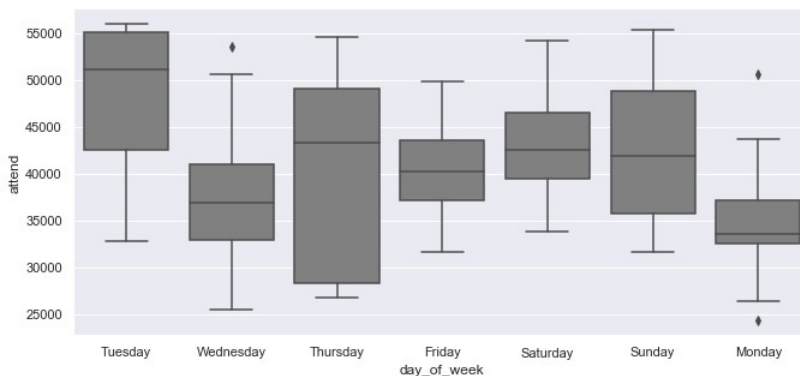
```
In [37]: sns.boxplot(y="attend", data=df)
```

```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x2c950e0b1c8>
```



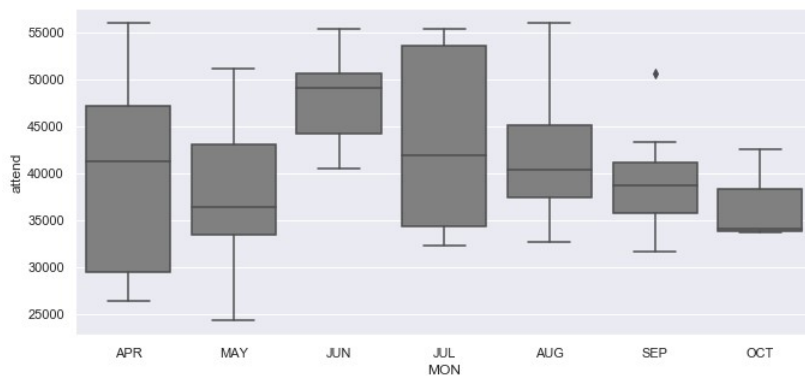
```
In [38]: sns.boxplot(x="day_of_week", y="attend", data=df, color = "gray")
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x2c950b84888>
```



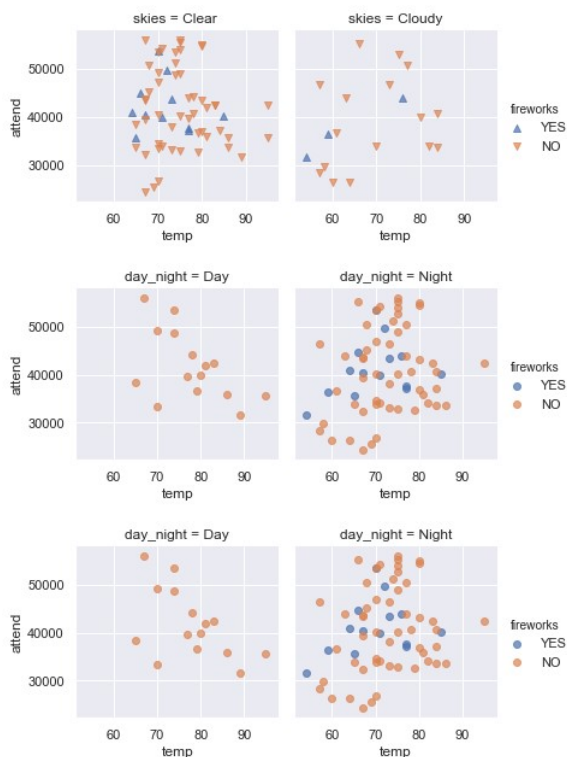
```
In [39]: sns.boxplot(x="MON", y="attend", data=df, color = "gray")
```

```
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x2c950a470c8>
```



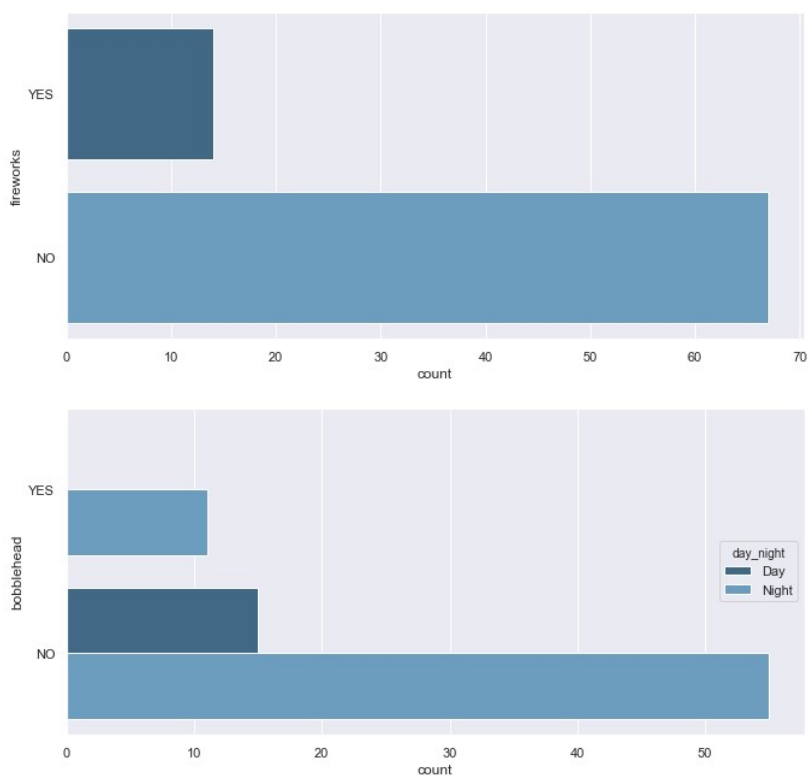
```
In [40]: sns.set(style="darkgrid")
```

```
g = sns.FacetGrid(df, col="skies", hue="fireworks",
                  hue_order=["YES", "NO"],
                  hue_kws=dict(marker=["^", "v"]))
g.map(plt.scatter, "temp", "attend", alpha=.7)
g.add_legend();
plt.show()
g = sns.FacetGrid(df, col="day_night", hue="fireworks",
                  hue_order=["YES", "NO"],)
g.map(plt.scatter, "temp", "attend", alpha=.7)
g.add_legend();
plt.show()
g = sns.FacetGrid(df, col="day_night", hue="fireworks",
                  hue_order=["YES", "NO"],)
g.map(plt.scatter, "temp", "attend", alpha=.7)
g.add_legend();
plt.show()
```

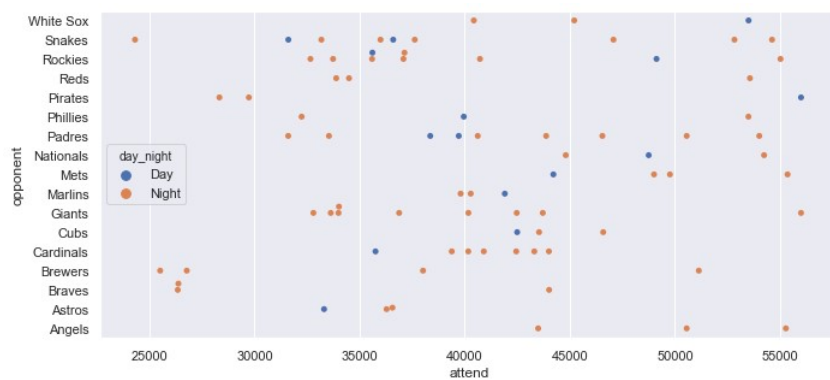


```
In [41]: sns.countplot(y="fireworks", data=df, palette="Blues_d",
                      order=["YES", "NO"])
g.add_legend();
plt.show()

sns.countplot(y="bubblehead", hue="day_night",
              data=df, palette="Blues_d",
              order=["YES", "NO"])
g.add_legend();
plt.show()
```

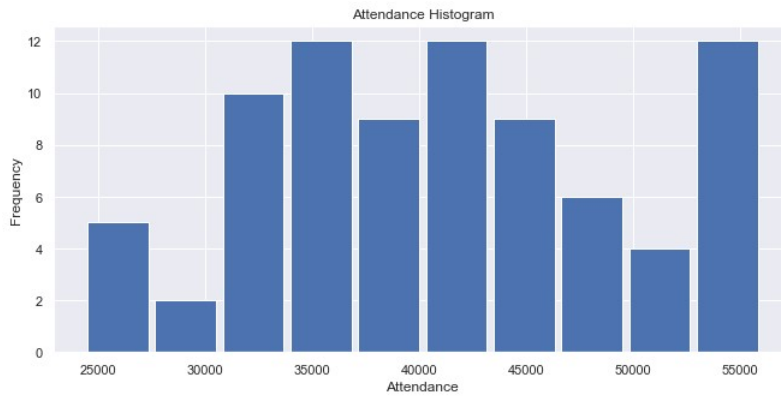


```
In [42]: sns.swarmplot(x="attend", y="opponent", hue="day_night",
                      data=df, order=df['opponent'].sort_values(ascending=False).unique())
g.add_legend();
plt.show()
```



```
In [43]: plt.hist(df['attend'], density = False, stacked = False, rwidth = .9)
plt.title("Attendance Histogram")
plt.xlabel('Attendance')
plt.ylabel('Frequency')

plt.show()
```



```
In [58]: X = df[['month', 'dow', 'bh_bit']]
y = df['attend']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42
)
ols = sm.OLS(y.to_numpy(), X.to_numpy())
ols = ols.fit()
print(ols.summary())
```

```

OLS Regression Results
=====
Dep. Variable:          y      R-squared (uncentered):      0.936
Model:                  OLS    Adj. R-squared (uncentered):    0.933
Method:                  Least Squares    F-statistic:          378.7
Date:                    Sun, 19 Jan 2020    Prob (F-statistic):    2.22e-46
Time:                    15:33:38    Log-Likelihood:       -865.77
No. Observations:        81    AIC:                  1738.
Df Residuals:            78    BIC:                  1745.
Df Model:                 3
Covariance Type:         nonrobust
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
x1             4474.2492    296.752     15.077     0.000    3883.461    5065.038
x2             2301.6352    531.919      4.327     0.000    1242.667    3360.604
x3             1.841e+04    3479.912      5.289     0.000    1.15e+04    2.53e+04
=====
Omnibus:                 3.880    Durbin-Watson:           0.838
Prob(Omnibus):            0.144    Jarque-Bera (JB):         3.122
Skew:                     0.379    Prob(JB):                 0.210
Kurtosis:                 3.592    Cond. No.                 21.7
=====
```

```

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
In [ ]: 
```

```
In [ ]: 
```

```
In [ ]: 
```

```
In [ ]: 
```