In [2]:

```
# Predictive Model for Los Angeles Dodgers Promotion and Attendance (Python)

# BASED ON EXHIBIT 2.1 FROM MILLER (2015)

# import packages for analysis and modeling
import pandas as pd  # data frame operations

import numpy as np  # arrays and math functions
from scipy.stats import uniform  # for training-and-test split
import statsmodels.api as sm  # statistical models (including regression)
import statsmodels.formula.api as smf  # R-like model specification
import matplotlib.pyplot as plt  # 2D plotting

import seaborn as sns  # PROVIDES TRELLIS AND SMALL MULTIPLE PLOTTING

# read in Dodgers bobbleheads data and create data frame
dodgers = pd.read_csv("/content/dodgers.csv")

# examine the structure of the data frame
print("\nContents of dodgers data frame ---------------")

# attendance in thousands for plotting
dodgers['attend_000'] = dodgers['attend']/1000

# print the first five rows of the data frame
print(pd.DataFrame.head(dodgers))
dodgerDF = pd.DataFrame(dodgers)

mondays = dodgers[dodgers['day_of_week'] == 'Monday']
tuesdays = dodgers[dodgers['day_of_week'] == 'Tuesday']
wednesdays = dodgers[dodgers['day_of_week'] == 'Wednesday']
thursdays = dodgers[dodgers['day_of_week'] == 'Thursday']
fridays = dodgers[dodgers['day_of_week'] == 'Friday']
saturdays = dodgers[dodgers['day_of_week'] == 'Saturday']
sundays = dodgers[dodgers['day_of_week'] == 'Sunday']

# convert days' attendance into list of vectors for box plot
data = [mondays['attend_000'], tuesdays['attend_000'],
    wednesdays['attend_000'], thursdays['attend_000'],
    fridays['attend_000'], saturdays['attend_000'],
    sundays['attend_000']]
ordered_day_names = ['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']

ordered_team_names = (sorted(set(dodgers['opponent']), reverse = True))
```

```
Contents of dodgers data frame ---------------
  month  day  attend day_of_week  ... shirt  fireworks bobblehead attend_000
0   APR   10   56000     Tuesday  ...    NO         NO         NO     56.000
1   APR   11   29729   Wednesday  ...    NO         NO         NO     29.729
2   APR   12   28328    Thursday  ...    NO         NO         NO     28.328
3   APR   13   31601      Friday  ...    NO        YES         NO     31.601
4   APR   14   46549    Saturday  ...    NO         NO         NO     46.549

[5 rows x 13 columns]
```

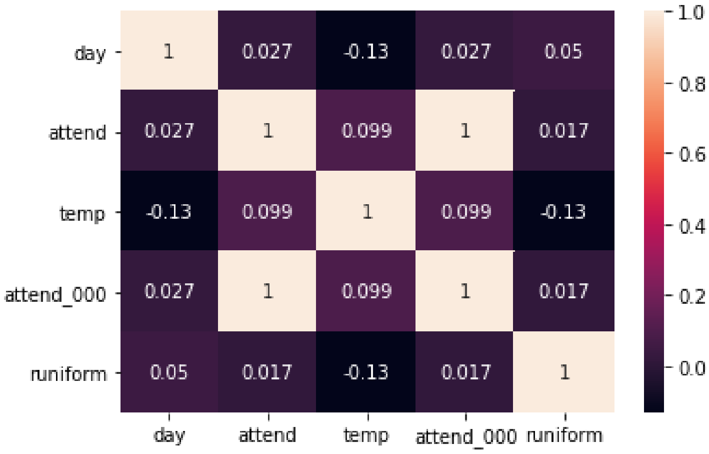
In [23]:

```
# ORDERING DATA

# map day_of_week to ordered_day_of_week
day_to_ordered_day = {'Monday' : '1Monday',
    'Tuesday' : '2Tuesday',
    'Wednesday' : '3Wednesday',
    'Thursday' : '4Thursday',
    'Friday' : '5Friday',
    'Saturday' : '6Saturday',
    'Sunday' : '7Sunday'}
```

```
dodgers['ordered_day_of_week'] = dodgers['day_of_week'].map(day_to_ordered_day)

# map month to ordered_month
month_to_ordered_month = {'APR' : '1April',
     'MAY' : '2May',
     'JUN' : '3June',
     'JUL' : '4July',
     'AUG' : '5Aug',
     'SEP' : '6Sept',
     'OCT' : '7Oct'}
dodgers['ordered_month'] = dodgers['month'].map(month_to_ordered_month)
```

In [25]:

```
corrMatrix = dodgers.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



In [19]:

```
# employ training-and-test regimen for model validation
np.random.seed(1234)
dodgers['runiform'] = uniform.rvs(loc = 0, scale = 1, size = len(dodgers))
train = dodgers[dodgers['runiform'] >= 0.33]
test = dodgers[dodgers['runiform'] < 0.33]

# Model 1
my_model = str('attend ~ ordered_month + ordered_day_of_week + skies +bobblehead')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = train).fit()

# summary of model fit to the training set
print(train_model_fit.summary())

train['predict_attend'] = train_model_fit.fittedvalues

test['predict_attend'] = train_model_fit.predict(test)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 attend   R-squared:                       0.643
Model:                            OLS   Adj. R-squared:                  0.524
Method:                 Least Squares   F-statistic:                     5.397
Date:                Sat, 31 Oct 2020   Prob (F-statistic):           1.00e-05
Time:                        14:24:09   Log-Likelihood:                -566.60
No. Observations:                  57   AIC:                             1163.
Df Residuals:                      42   BIC:                             1194.
Df Model:                          14
Covariance Type:            nonrobust
========================================================================================
============
                                        coef    std err          t      P>|t|      [0.0
25      0.975]
--------------------------------------------------------------------------------------
--------------
```

```
Intercept                       3.676e+04   3383.325     10.866      0.000    2.99e+0
4    4.36e+04
ordered_month[T.2May]          -3804.5270   2815.175     -1.351      0.184   -9485.78
1    1876.727
ordered_month[T.3June]          8048.1063   3213.265      2.505      0.016    1563.47
4    1.45e+04
ordered_month[T.4July]          3162.0657   3371.592      0.938      0.354   -3642.08
3    9966.215
ordered_month[T.5Aug]           1089.9430   3094.284      0.352      0.726   -5154.57
5    7334.461
ordered_month[T.6Sept]           724.4633   3014.531      0.240      0.811   -5359.10
6    6808.032
ordered_month[T.7Oct]           -933.1412   6469.958     -0.144      0.886    -1.4e+0
4    1.21e+04
ordered_day_of_week[T.2Tuesday]    5148.7993   3551.931      1.450      0.155   -2019.288
1.23e+04
ordered_day_of_week[T.3Wednesday]  -310.0199   3321.586     -0.093      0.926   -7013.252
6393.212
ordered_day_of_week[T.4Thursday]   -659.6697   3891.713     -0.170      0.866   -8513.464
7194.125
ordered_day_of_week[T.5Friday]     3651.4235   2928.772      1.247      0.219   -2259.077
9561.924
ordered_day_of_week[T.6Saturday]   3311.3463   3012.610      1.099      0.278   -2768.346
9391.039
ordered_day_of_week[T.7Sunday]     2627.7652   3186.642      0.825      0.414   -3803.139
9058.669
skies[T.Cloudy]                -1505.7707   2377.701     -0.633      0.530   -6304.16
5    3292.624
bobblehead[T.YES]               1.211e+04   2723.232      4.448      0.000    6618.13
7    1.76e+04
==============================================================================
Omnibus:                        3.219   Durbin-Watson:                   2.121
Prob(Omnibus):                  0.200   Jarque-Bera (JB):                2.542
Skew:                           0.511   Prob(JB):                        0.281
Kurtosis:                       3.160   Cond. No.                         11.0
==============================================================================

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

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  app.launch_new_instance()
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
```

In [20]:

```
#Using full dataset
my_model_fit = smf.ols(my_model, data = dodgers).fit()
print(my_model_fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 attend   R-squared:                       0.559
Model:                            OLS   Adj. R-squared:                  0.465
Method:                 Least Squares   F-statistic:                     5.968
Date:                Sat, 31 Oct 2020   Prob (F-statistic):           2.17e-07
Time:                        14:24:12   Log-Likelihood:                -812.22
No. Observations:                  81   AIC:                             1654.
Df Residuals:                      66   BIC:                             1690.
Df Model:                          14
Covariance Type:            nonrobust
```

```
=========================================================================================
============
                                        coef    std err          t      P>|t|      [0.0
25      0.975]
---------------------------------------------------------------------------------------
--------------
Intercept                          3.53e+04   2674.948     13.196      0.000       3e+0
4    4.06e+04
ordered_month[T.2May]            -3619.4053   2423.439     -1.493      0.140   -8457.95
7    1219.146
ordered_month[T.3June]            5898.1922   2844.386      2.074      0.042     219.19
3    1.16e+04
ordered_month[T.4July]            2231.8287   2591.594      0.861      0.392   -2942.45
5    7406.113
ordered_month[T.5Aug]              981.9946   2566.679      0.383      0.703   -4142.54
5    6106.534
ordered_month[T.6Sept]            -793.2216   2562.463     -0.310      0.758   -5909.34
5    4322.902
ordered_month[T.7Oct]            -1490.6548   4052.171     -0.368      0.714   -9581.07
5    6599.766
ordered_day_of_week[T.2Tuesday]   8294.4599   2692.260      3.081      0.003    2919.190
1.37e+04
ordered_day_of_week[T.3Wednesday] 3098.6730   2530.840      1.224      0.225   -1954.312
8151.658
ordered_day_of_week[T.4Thursday]   934.1158   3458.565      0.270      0.788   -5971.133
7839.365
ordered_day_of_week[T.5Friday]    5094.2917   2487.772      2.048      0.045     127.295
1.01e+04
ordered_day_of_week[T.6Saturday]  6771.0858   2545.297      2.660      0.010    1689.236
1.19e+04
ordered_day_of_week[T.7Sunday]    6228.2311   2508.653      2.483      0.016    1219.543
1.12e+04
skies[T.Cloudy]                  -2706.5489   1850.049     -1.463      0.148   -6400.29
0     987.192
bobblehead[T.YES]                 1.056e+04   2401.657      4.395      0.000    5760.65
9    1.54e+04
==============================================================================
Omnibus:                        6.129   Durbin-Watson:                   2.143
Prob(Omnibus):                  0.047   Jarque-Bera (JB):                5.827
Skew:                           0.655   Prob(JB):                       0.0543
Kurtosis:                       3.102   Cond. No.                         10.2
==============================================================================

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

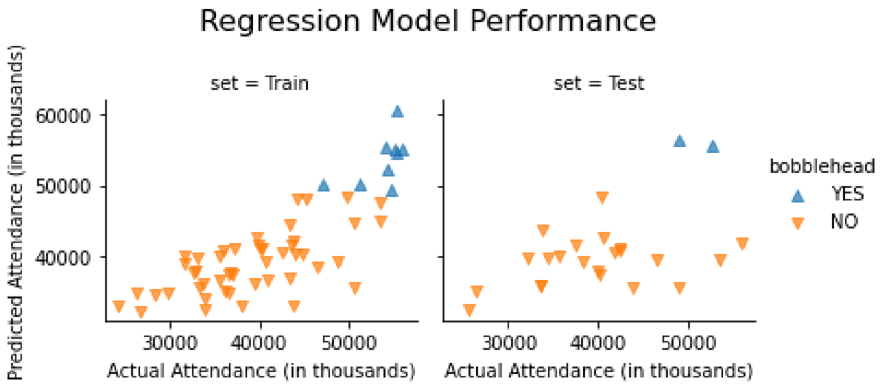
In [21]:

```
#Add set column
test['set']='Test'
train['set']='Train'

#combine datasets
combo = test.append(train, ignore_index=True)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [22]:

```
g = sns.FacetGrid(combo, col="set", hue="bobblehead",
                  hue_order=["YES", "NO"],
                  #reorder col
                  col_order=["Train","Test"],
                  hue_kws=dict(marker=["^", "v"]))
g.map(plt.scatter, "attend", "predict_attend",
      alpha=.7).set_axis_labels("Actual Attendance (in thousands)",
                                "Predicted Attendance (in thousands)")
g.add_legend();
plt.subplots_adjust(top=0.75)
g.fig.suptitle('Regression Model Performance', fontsize = 16)
plt.show()
```


Regression Model Performance
set = Train
set = Test
Predicted Attendance (in thousands)
Actual Attendance (in thousands)
Actual Attendance (in thousands)
bobblehead
▲ YES
▼ NO