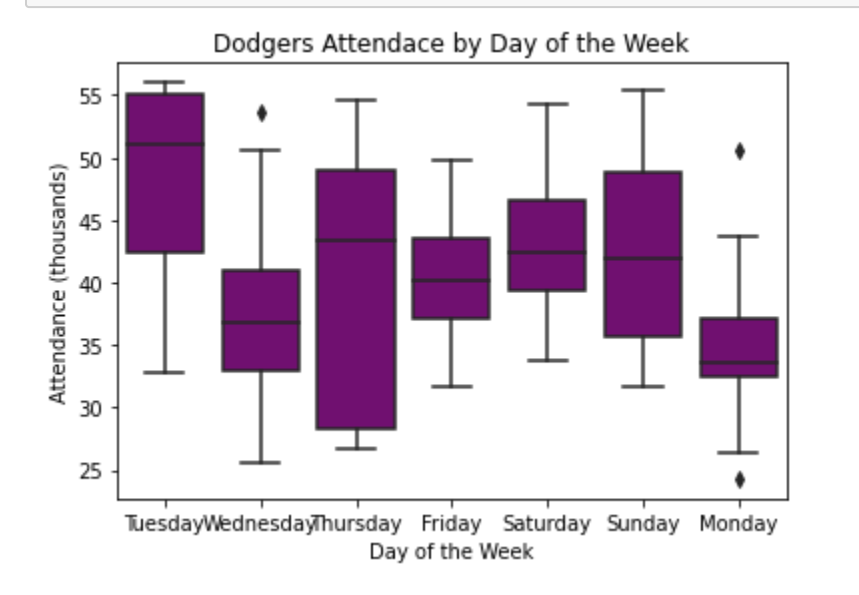
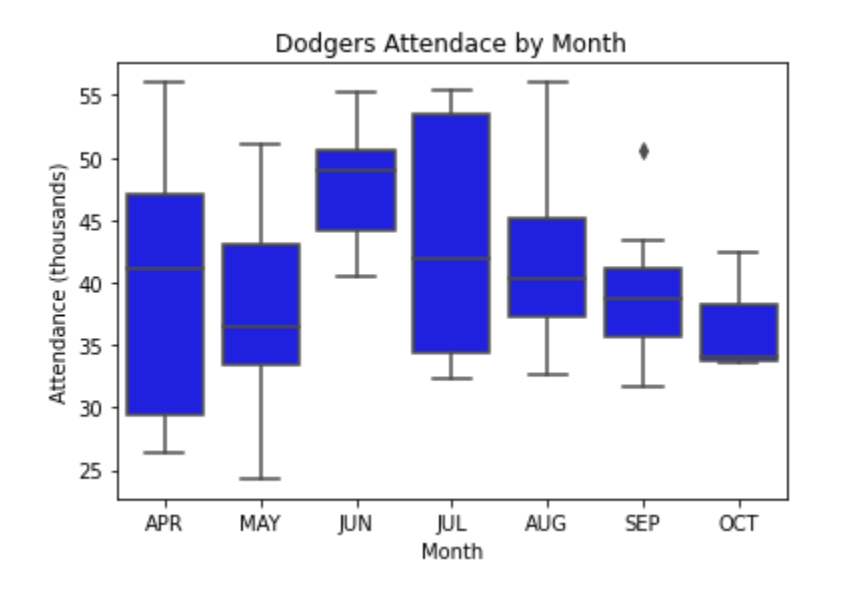
In [15]:
```
#Author: Laura C. Larregui
#Purpose: Week 2 Code Review

# 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("C:\\Users\\laura\\OneDrive\\Documents\\Syracuse iSchool\\IST 718\\Rep
os\\IST_718\\dodgers.csv")

print(dodgers.describe())
#print(dodgers.info)

#The following code id the data cleanup made in the Jupyter Notebook presented in the Async
 Video

dodgerDF = pd.DataFrame(dodgers)

dodgers['attend_000'] = dodgers['attend']/1000
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))
```

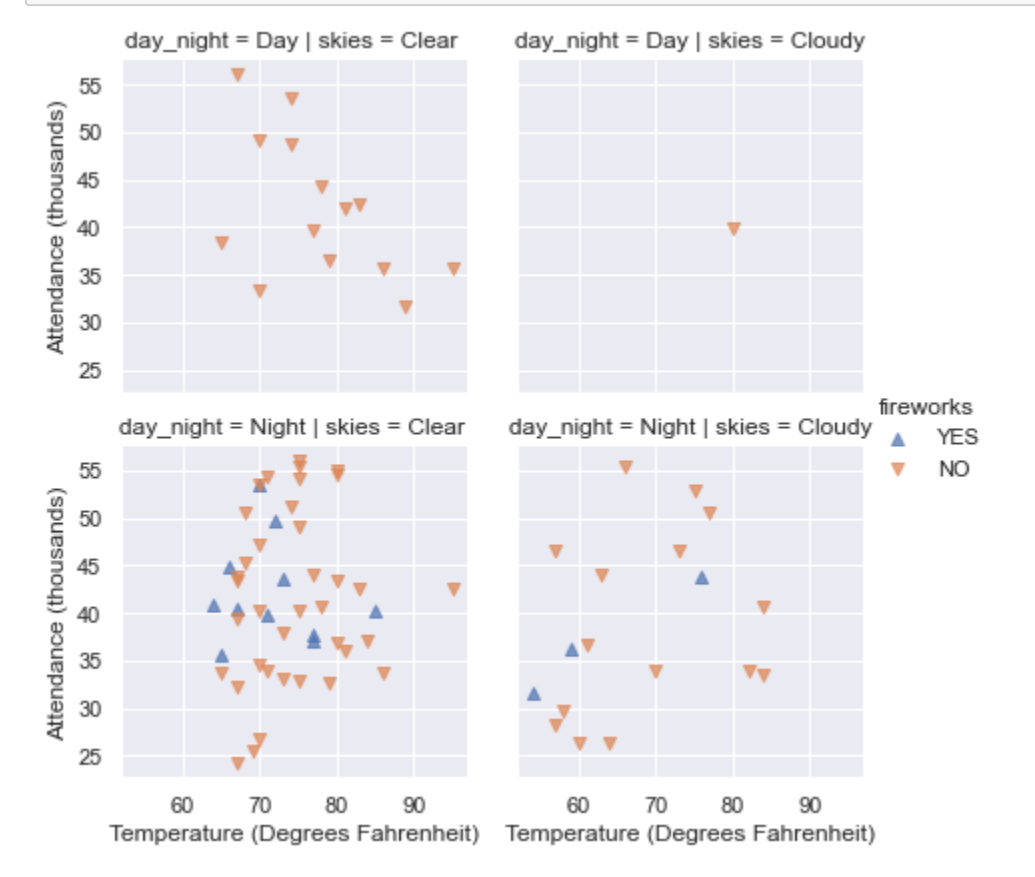
```
             day        attend       temp
count  81.000000     81.000000  81.000000
mean   16.135802  41040.074074  73.148148
std     9.605666   8297.539460   8.317318
min     1.000000  24312.000000  54.000000
25%     8.000000  34493.000000  67.000000
50%    15.000000  40284.000000  73.000000
75%    25.000000  46588.000000  79.000000
max    31.000000  56000.000000  95.000000
```

In [25]:
```
# Instructions: try to recreate  the graphics and the model from Chapter 2

# Mean
np.mean(dodgers['attend_000'])

#Figure 2.1: Dodgers Attendace by Day of the Week
sns.boxplot(x="day_of_week", y="attend_000", data=dodgers, color = "purple");
plt.title('Dodgers Attendace by Day of the Week')
plt.ylabel('Attendance (thousands)')
plt.xlabel('Day of the Week')
plt.show()

#Figure 2.2: Dodgers Attendace by Month
sns.boxplot(x="month", y="attend_000", data=dodgers, color = "blue");
plt.title('Dodgers Attendace by Month')
plt.ylabel('Attendance (thousands)')
plt.xlabel('Month')
plt.show()
```

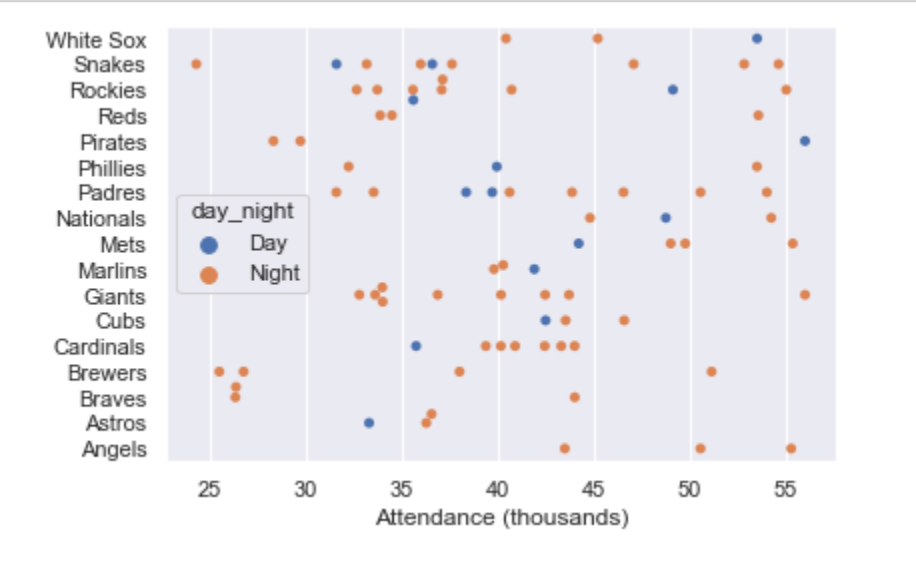




In [55]:
```
#Figure 2.3: Dodgers Weather, Fireworks, and Attendance

# trellis/lattice plot attendance by temp, conditioning on skies
# and day_night with bobblehead NO/YES shown in distinct colors
import seaborn as sns

sns.set(style="darkgrid")

g = sns.FacetGrid(dodgers, col="skies",row="day_night", hue="fireworks",
                  hue_order=["YES", "NO"],
                  hue_kws=dict(marker=["^", "v"]))
g.map(plt.scatter, "temp", "attend_000", alpha=.7)
g.add_legend();

g.set_axis_labels("Temperature (Degrees Fahrenheit)", "Attendance (thousands)")

plt.show()
```



In [66]:
```
#Figure 2.4: Dodgers Attendance by Visiting Team
sns.swarmplot(x="attend_000", y="opponent", hue="day_night",
              data=dodgers, order =ordered_team_names)

plt.xlabel('Attendance (thousands)')
plt.ylabel(' ')
g.add_legend(loc='lower right');
plt.show()
```



In [96]:
```
#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'}
dodgers['ordered_month'] = dodgers['month'].map(month_to_ordered_month)
```

In [97]:
```
# employ training-and-test regimen for model validation
np.random.seed(1234)
dodgers['runiform'] = uniform.rvs(loc = 0, scale = 1, size = len(dodgers))
dodgers_train = dodgers[dodgers['runiform'] >= 0.33]#more than 1/3 of the dataset
dodgers_test = dodgers[dodgers['runiform'] < 0.33]#less than 1/3 of the dataset
# check training data frame
print('\ndodgers_train data frame (rows, columns): ',dodgers_train.shape)
print(dodgers_train.head())
# check test data frame
print('\ndodgers_test data frame (rows, columns): ',dodgers_test.shape)
print(dodgers_test.head())

# specify a simple model with bobblehead entered last
my_model = str('attend_000 ~ ordered_month + ordered_day_of_week + bobblehead')

# fit the model to the training set
train_model_fit = smf.ols(my_model, data = dodgers_train).fit()
# summary of model fit to the training set
print(train_model_fit.summary())
# training set predictions from the model fit to the training set
dodgers_train['predict_attend'] = train_model_fit.fittedvalues

# test set predictions from the model fit to the training set
dodgers_test['predict_attend'] = train_model_fit.predict(dodgers_test)
```

```
dodgers_train data frame (rows, columns):  (57, 17)
  month  day  attend day_of_week opponent  temp   skies day_night cap shirt  \
1   APR   11   29729   Wednesday  Pirates    58  Cloudy     Night  NO    NO
2   APR   12   28328    Thursday  Pirates    57  Cloudy     Night  NO    NO
3   APR   13   31601      Friday   Padres    54  Cloudy     Night  NO    NO
4   APR   14   46549    Saturday   Padres    57  Cloudy     Night  NO    NO
7   APR   24   44014     Tuesday   Braves    63  Cloudy     Night  NO    NO

  fireworks bobblehead    division  attend_000  runiform ordered_day_of_week  \
1        NO         NO       Cross      29.729  0.622109          3Wednesday
2        NO         NO       Cross      28.328  0.437728           4Thursday
3       YES         NO       Cross      31.601  0.785359             5Friday
4        NO         NO  Divisional      46.549  0.779976           6Saturday
7        NO         NO  Divisional      44.014  0.801872            2Tuesday

  ordered_month
1        1April
2        1April
3        1April
4        1April
7        1April

dodgers_test data frame (rows, columns):  (24, 17)
   month  day  attend day_of_week   opponent  temp   skies day_night cap  \
0    APR   10   56000     Tuesday    Pirates    67  Clear       Day  NO
5    APR   15   38359      Sunday     Padres    65  Clear       Day  NO
6    APR   23   26376      Monday     Braves    60  Cloudy     Night  NO
17   MAY   13   49124      Sunday    Rockies    70  Clear       Day  NO
22   MAY   20   44005      Sunday  Cardinals    77  Clear     Night  NO

   shirt fireworks bobblehead    division  attend_000  runiform  \
0     NO        NO         NO  Divisional      56.000  0.191519
5     NO        NO         NO       Cross      38.359  0.272593
6     NO        NO         NO       Cross      26.376  0.276464
17    NO        NO         NO  Divisional      49.124  0.013768
22    NO        NO         NO  Divisional      44.005  0.075381

   ordered_day_of_week ordered_month
0             2Tuesday        1April
5              7Sunday        1April
6              1Monday        1April
17             7Sunday          2May
22             7Sunday          2May
                            OLS Regression Results
==============================================================================
Dep. Variable:             attend_000   R-squared:                       0.634
Model:                            OLS   Adj. R-squared:                  0.532
Method:                 Least Squares   F-statistic:                     6.219
Date:                Wed, 07 Oct 2020   Prob (F-statistic):           3.44e-06
Time:                        12:18:39   Log-Likelihood:                -170.59
No. Observations:                  56   AIC:                             367.2
Df Residuals:                      43   BIC:                             393.5
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
========
                                    coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept                        35.9897      3.133     11.486      0.000      29.671
42.309
ordered_month[T.2May]            -3.0495      2.532     -1.204      0.235      -8.157
2.058
ordered_month[T.3June]            8.7768      2.979      2.946      0.005       2.769
14.785
ordered_month[T.4July]            3.7056      3.238      1.145      0.259      -2.824
10.235
ordered_month[T.5Aug]             2.0276      2.698      0.751      0.456      -3.414
7.469
ordered_month[T.6Sept]            1.2808      2.864      0.447      0.657      -4.494
7.056
ordered_day_of_week[T.2Tuesday]       4.7343      3.467      1.366      0.179      -2.257
11.726
ordered_day_of_week[T.3Wednesday]    -0.7414      3.228     -0.230      0.819      -7.252
5.769
ordered_day_of_week[T.4Thursday]     -0.8790      3.849     -0.228      0.820      -8.642
6.884
ordered_day_of_week[T.5Friday]        3.4208      2.886      1.185      0.242      -2.399
9.240
ordered_day_of_week[T.6Saturday]      3.0605      2.966      1.032      0.308      -2.920
9.041
ordered_day_of_week[T.7Sunday]        2.7964      3.153      0.887      0.380      -3.563
9.156
bobblehead[T.YES]                12.3293      2.683      4.595      0.000       6.918
17.740
==============================================================================
Omnibus:                        2.223   Durbin-Watson:                   2.071
Prob(Omnibus):                  0.329   Jarque-Bera (JB):                1.873
Skew:                           0.447   Prob(JB):                        0.392
Kurtosis:                       2.941   Cond. No.                         10.5
==============================================================================

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

```
<ipython-input-97-6b90c7be4325>:21: 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_guid
e/indexing.html#returning-a-view-versus-a-copy
  dodgers_train['predict_attend'] = train_model_fit.fittedvalues
<ipython-input-97-6b90c7be4325>:24: 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_guid
e/indexing.html#returning-a-view-versus-a-copy
  dodgers_test['predict_attend'] = train_model_fit.predict(dodgers_test)
```

In [98]:
```
# compute the proportion of response variance
# accounted for when predicting out-of-sample
print('\nProportion of Test Set Variance Accounted for: ',\
    round(np.power(dodgers_test['attend_000'].corr(dodgers_test['predict_attend']),2),3))

# use the full data set to obtain an estimate of the increase in
# attendance due to bobbleheads, controlling for other factors
my_model_fit = smf.ols(my_model, data = dodgers).fit()
print(my_model_fit.summary())

print('\nEstimated Effect of Bobblehead Promotion on Attendance: ',\
    round(my_model_fit.params[13],0))
```

```
Proportion of Test Set Variance Accounted for:  0.201
                            OLS Regression Results
==============================================================================
Dep. Variable:             attend_000   R-squared:                       0.536
Model:                            OLS   Adj. R-squared:                  0.451
Method:                 Least Squares   F-statistic:                     6.262
Date:                Wed, 07 Oct 2020   Prob (F-statistic):           3.40e-07
Time:                        12:19:16   Log-Likelihood:                -245.97
No. Observations:                  78   AIC:                             517.9
Df Residuals:                      65   BIC:                             548.6
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
========
                                    coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept                        33.8780      2.606     13.001      0.000      28.674
39.082
ordered_month[T.2May]            -2.3782      2.325     -1.023      0.310      -7.021
2.264
ordered_month[T.3June]            7.1507      2.773      2.579      0.012       1.613
12.688
ordered_month[T.4July]            2.8466      2.617      1.088      0.281      -2.380
8.073
ordered_month[T.5Aug]             2.3551      2.439      0.966      0.338      -2.515
7.226
ordered_month[T.6Sept]            0.0397      2.557      0.016      0.988      -5.068
5.147
ordered_day_of_week[T.2Tuesday]       7.7833      2.886      2.697      0.009       2.020
13.547
ordered_day_of_week[T.3Wednesday]     2.6507      2.665      0.995      0.324      -2.671
7.972
ordered_day_of_week[T.4Thursday]      0.7679      3.579      0.215      0.831      -6.380
7.916
ordered_day_of_week[T.5Friday]        4.9191      2.580      1.906      0.061      -0.234
10.072
ordered_day_of_week[T.6Saturday]      6.3885      2.634      2.425      0.018       1.128
11.649
ordered_day_of_week[T.7Sunday]        6.7479      2.585      2.610      0.011       1.585
11.911
bobblehead[T.YES]                10.8202      2.484      4.355      0.000       5.858
15.782
==============================================================================
Omnibus:                        5.711   Durbin-Watson:                   2.117
Prob(Omnibus):                  0.058   Jarque-Bera (JB):                5.367
Skew:                           0.641   Prob(JB):                       0.0683
Kurtosis:                       3.088   Cond. No.                         9.53
==============================================================================

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

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_value(self, series, key)
   4404         try:
-> 4405             return self._engine.get_value(s, k, tz=getattr(series.dtype, "tz", None))
   4406         except KeyError as e1:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_value()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_value()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item
()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item
()

KeyError: 13

During handling of the above exception, another exception occurred:

IndexError                                Traceback (most recent call last)
<ipython-input-98-ef2b2049da85> in <module>
     10
     11 print('\nEstimated Effect of Bobblehead Promotion on Attendance: ',\
---> 12     round(my_model_fit.params[13],0))

~\anaconda3\lib\site-packages\pandas\core\series.py in __getitem__(self, key)
    869         key = com.apply_if_callable(key, self)
    870         try:
--> 871             result = self.index.get_value(self, key)
    872
    873             if not is_scalar(result):

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_value(self, series, key)
   4409
   4410             try:
-> 4411                 return libindex.get_value_at(s, key)
   4412             except IndexError:
   4413                 raise

pandas\_libs\index.pyx in pandas._libs.index.get_value_at()

pandas\_libs\index.pyx in pandas._libs.index.get_value_at()

pandas\_libs\util.pxd in pandas._libs.util.get_value_at()

pandas\_libs\util.pxd in pandas._libs.util.validate_indexer()

IndexError: index out of bounds
```

In [99]:
```
#Figure 2.5 Regression Model Performance: Bobbleheads and Attendance

#Add set column with Test/Train
dodgers_test['set']='Test'
dodgers_train['set']='Train'

#combine test/train to one set
combined_set = dodgers_test.append(dodgers_train, ignore_index=True)

g = sns.FacetGrid(combined_set, col="set", hue="bobblehead",
                  hue_order=["YES", "NO"],
                  #reorder col
                  col_order=["Train","Test"],
                  hue_kws=dict(marker=["^", "v"]))

#rename axis
g.map(plt.scatter, "attend", "predict_attend",
```