Experiment Results

Results for Atlanta

In [17]:
```
import warnings
warnings.filterwarnings('ignore')

%run helper_functions.py
%matplotlib inline
```

Lets review the top 10 results for Atlanta, ordered by Mean Square Error (ascending)

In [5]:
```
city='Atlanta'
results_top_10 = get_results(city=city, top_hm_results=10)
results_top_10
```

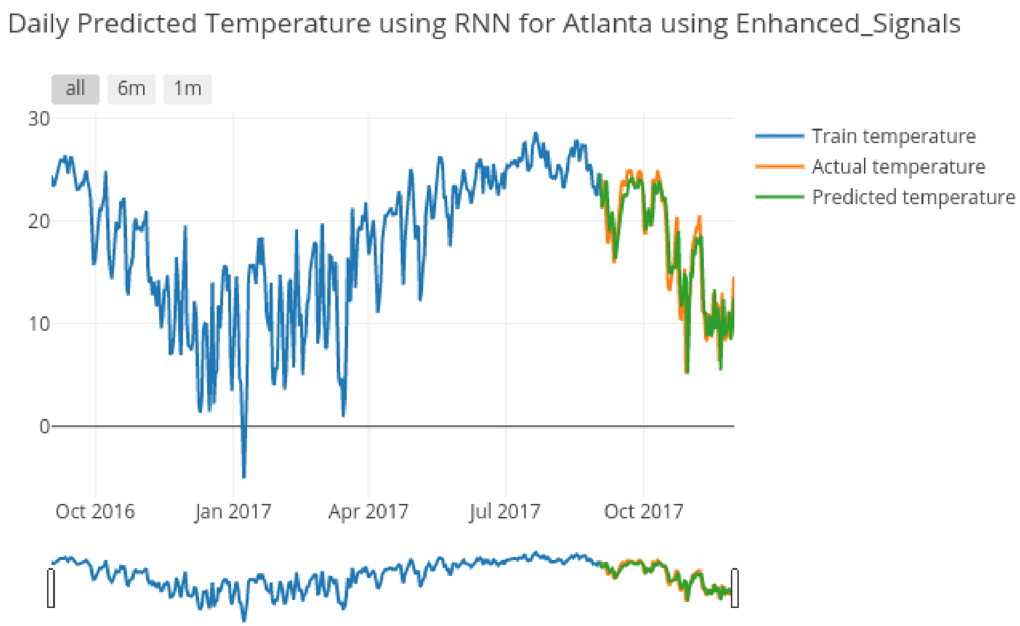
Out[5]:

	RUN_ID	DATETIME	MODEL_NAME	CITY	FEATURE_TYPE	HOST_MACHINE	MODEL_PARAMETERS	MODEL_RESULTS	MEAN_SQUARED_ERROR
1	50	2018-08-14 20:39:13.461675	RNN	Atlanta	Enhanced_Signals	DESKTOP-KN40C32	{'epochs': 50, 'Info': {'feature_set_type': 'E...	{'features': ['temperature_lag1', 'temperature...	4.966304
2	38	2018-08-14 20:34:21.973872	RNN	Atlanta	Basic	DESKTOP-KN40C32	{'epochs': 50, 'Info': {'feature_set_type': 'B...	{'features': ['temperature_lag1', 'temperature...	5.105819
3	26	2018-08-14 19:55:31.041998	RANDOM FOREST	Atlanta	Inc_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'n_estimators': 50, 'Info': {...	{'features': ['temperature_lag1', 'temperature...	5.434763
4	32	2018-08-14 19:56:23.951216	RANDOM FOREST	Atlanta	Enhanced_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'n_estimators': 50, 'Info': {...	{'features': ['temperature_lag1', 'temperature...	5.434763
5	20	2018-08-14 19:54:40.149363	RANDOM FOREST	Atlanta	Basic	DESKTOP-KN40C32	{'max_depth': 8, 'n_estimators': 50, 'Info': {...	{'features': ['temperature_lag1', 'temperature...	5.516615
6	44	2018-08-14 20:36:38.538124	RNN	Atlanta	Inc_Signals	DESKTOP-KN40C32	{'epochs': 50, 'Info': {'feature_set_type': 'I...	{'features': ['temperature_lag1', 'temperature...	5.628276
7	14	2018-08-14 19:44:07.609795	DECISION TREE	Atlanta	Enhanced_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'Info': {'feature_set_type': ...	{'features': ['temperature_lag1', 'temperature...	7.783687
8	8	2018-08-14 19:43:12.930159	DECISION TREE	Atlanta	Inc_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'Info': {'feature_set_type': ...	{'features': ['temperature_lag1', 'temperature...	7.992946
9	2	2018-08-14 19:42:13.654060	DECISION TREE	Atlanta	Basic	DESKTOP-KN40C32	{'max_depth': 8, 'Info': {'feature_set_type': ...	{'features': ['temperature_lag1', 'temperature...	8.436149

RNN has done better than Random forest but the results are close so there is no clear winner, lets take a look at that chart.

In [6]:
```
display_results(results_top_10.head(1), chart_type='predict')
```

Experiment #50 - run by DESKTOP-KN40C32 on 2018-08-14 20:39:13.461675



We can see the forecast follows the trend and it does a better job vs the prior example of Miami with respect to following the peaks and troughs of the actual temperature pattern experiences.

Lets examine the top 10 features across the model runs for this location in a pivot table form.

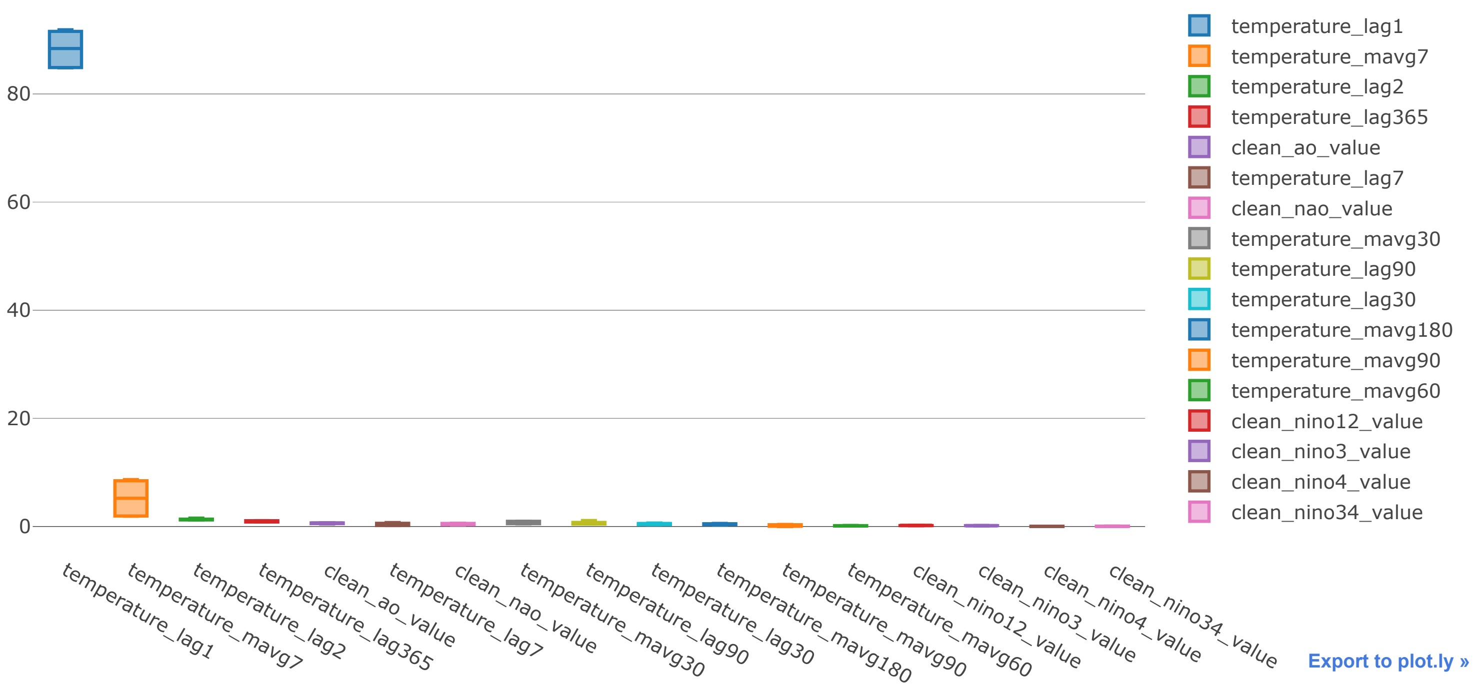
In [7]:
```
features_df = get_feature_importances(results_top_10)
features_df.pivot_table(index=['FEATURE'], columns=['MODEL_NAME','FEATURE_TYPE'], values="IMPORTANCE")
```

Out[7]:

MODEL_NAME	DECISION TREE			RANDOM FOREST		
FEATURE_TYPE	Basic	Enhanced_Signals	Inc_Signals	Basic	Enhanced_Signals	Inc_Signals
FEATURE						
clean_ao_value	NaN	0.484511	0.586696	NaN	0.667402	0.667402
clean_nao_value	NaN	0.387084	0.319784	NaN	0.570843	0.570843
clean_nino12_value	NaN	0.176243	0.171550	NaN	NaN	NaN
clean_nino34_value	NaN	0.014362	0.075990	NaN	NaN	NaN
clean_nino3_value	NaN	0.155625	0.162820	NaN	NaN	NaN
clean_nino4_value	NaN	0.050394	0.000808	NaN	NaN	NaN
temperature_lag1	91.866531	91.563345	91.569282	85.242322	84.878510	84.878510
temperature_lag2	1.239663	1.220060	1.194826	1.553500	1.365561	1.365561
temperature_lag30	0.551914	0.641604	0.527405	0.532966	0.353330	0.353330
temperature_lag365	1.069122	0.947275	0.986347	1.080280	0.836627	0.836627
temperature_lag7	0.186437	0.173366	0.185205	0.730078	0.593266	0.593266
temperature_lag90	1.104324	0.751362	0.739947	0.587154	0.454484	0.454484
temperature_mavg180	0.570359	0.293828	0.448047	0.404703	NaN	NaN
temperature_mavg30	0.980080	0.966142	0.957687	0.671558	0.515406	0.515406
temperature_mavg60	0.107090	0.180291	0.096090	NaN	NaN	NaN
temperature_mavg7	1.944451	1.978938	1.905487	8.675354	8.460200	8.460200
temperature_mavg90	0.380028	0.015570	0.072030	0.274919	NaN	NaN

Lets look a boxplot of this data to see how the distributions look across our experiments for this location.

In [19]:
```
traces = create_boxplot_traces_for_features(features_df)
iplot(traces)
```



Lets review the means for the features

In [10]:
```
features_df = get_feature_importances(results_top_10)
pivot_df = features_df.pivot_table(index=['FEATURE'], columns=[], values="IMPORTANCE", aggfunc= [np.mean])
pivot_df
```

Out[10]:

	mean
	IMPORTANCE
FEATURE	
clean_ao_value	0.601503
clean_nao_value	0.462139
clean_nino12_value	0.173896
clean_nino34_value	0.045176
clean_nino3_value	0.159222
clean_nino4_value	0.025601
temperature_lag1	88.333084
temperature_lag2	1.323195
temperature_lag30	0.493425
temperature_lag365	0.959380
temperature_lag7	0.410270
temperature_lag90	0.681959
temperature_mavg180	0.429234
temperature_mavg30	0.767713
temperature_mavg60	0.127823
temperature_mavg7	5.237438
temperature_mavg90	0.185637

Now lets review the top 5 only

In [11]:
```
pivot_df.columns = pivot_df.columns.get_level_values(0)
pivot_df.sort_values(['mean'], ascending=False).head(5)
```

Out[11]:

	mean
FEATURE	
temperature_lag1	88.333084
temperature_mavg7	5.237438
temperature_lag2	1.323195
temperature_lag365	0.959380
temperature_mavg30	0.767713

In [ ]: