Preference Learning - Part III

1. Groups' Price Responsiveness - K-mean based

In [1]:
```
%matplotlib inline
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import json
import seaborn as sns
import math
from pandas.io.json import json_normalize #package for flattening json in pandas df
import matplotlib.pyplot as plt
from datetime import date, timedelta, datetime
```

In [ ]:
```
# Import all preprocessed data necessary for the analysis
df_tou1h = pd.read_csv("C:\\Users\\Rockwell\\Desktop\\Paper4\\data_collection\\data_tables\\Consumption_tou2013_1h.csv")
df_Ntou1h = pd.read_csv("C:\\Users\\Rockwell\\Desktop\\Paper4\\data_collection\\data_tables\\Consumption_Ntou2013_1h.csv")
df_wea1h = pd.read_csv("C:\\Users\\Rockwell\\Desktop\\Paper4\\data_collection\\weather\\LondonWeather2013_interpolated.csv")
df_tariff_1h = pd.read_csv("C:\\Users\\Rockwell\\Desktop\\Paper4\\data_collection\\data_tables\\df_tariff_1h.csv")
```

In [2]:
```
import os
os.getcwd()
```

Out[2]:
```
'/Users/Rockwell/Documents/GitHub/Demand-Response'
```

In [3]:
```
# for ios system, import all data necessary for the analysis
df_tou1h = pd.read_csv('/Users/Rockwell/Desktop/PhD/Paper4PreferenceLearning/data/Consumption_tou2013_1h.csv')
df_Ntou1h = pd.read_csv('/Users/Rockwell/Desktop/PhD/Paper4PreferenceLearning/data/Consumption_Ntou2013_1h.csv')
df_wea1h = pd.read_csv('/Users/Rockwell/Desktop/PhD/Paper4PreferenceLearning/data/LondonWeather2013_interpolated.csv')
df_tariff_1h = pd.read_csv('/Users/Rockwell/Desktop/PhD/Paper4PreferenceLearning/data/df_tariff_1h.csv')
```

In [4]:
```
# first, create a list of days that belongs to event days
event_days = set()
event_series = df_tariff_1h[df_tariff_1h.Event_tags.notnull()].GMT
for i in event_series:
    event_days.add(datetime.strptime(i[:10], "%Y-%m-%d" ).date()) # add all event dates to the set
df_help = pd.DataFrame(pd.to_datetime(df_tariff_1h.GMT).dt.date) #str to datatime and extract date then make it to dataframe
# df_help[df_help['GMT'].isin(event_days)] #this shows the event days
# we can use ~df_help['GMT'].isin(event_days) to generate any non-flexible period items

# create TOU and non-TOU demand data in non-flexible hours
df_wea1h_nf = df_wea1h[~df_help['GMT'].isin(event_days)]
df_Ntou1h_nf = df_Ntou1h[~df_help['GMT'].isin(event_days)]
df_tou1h_nf = df_tou1h[~df_help['GMT'].isin(event_days)]

# create event data
df_tariff_1h_event = df_tariff_1h[df_tariff_1h.GMT.isin(event_series)]
df_wea1h_event = df_wea1h[df_wea1h.GMT.isin(event_series)]
df_tou1h_event = df_tou1h[df_tou1h.GMT.isin(event_series)]
```

In [5]:
```
# seperate the above data set based on the seasonal effect
# i.e., months of 11, 12, 1, 2, 3 are in a group - cold season
# months of 4, 5, 6, 7, 8, 9, 10 are in another group - warm season
cold_season = [11, 12, 1, 2, 3]
warm_season = [4, 5, 6, 7, 8, 9, 10]
df_help_season = pd.DataFrame(pd.to_datetime(df_wea1h_nf.GMT).dt.month)
df_wea1h_nf_cold = df_wea1h_nf[df_help_season['GMT'].isin(cold_season)]
df_wea1h_nf_warm = df_wea1h_nf[df_help_season['GMT'].isin(warm_season)]
df_Ntou1h_nf_cold = df_Ntou1h_nf[df_help_season['GMT'].isin(cold_season)]
df_Ntou1h_nf_warm = df_Ntou1h_nf[df_help_season['GMT'].isin(warm_season)]
df_tou1h_nf_cold = df_tou1h_nf[df_help_season['GMT'].isin(cold_season)]
df_tou1h_nf_warm = df_tou1h_nf[df_help_season['GMT'].isin(warm_season)]

df_help_season_event = pd.DataFrame(pd.to_datetime(df_wea1h_event.GMT).dt.month)
df_wea1h_event_cold = df_wea1h_event[df_help_season_event['GMT'].isin(cold_season)]
df_wea1h_event_warm = df_wea1h_event[df_help_season_event['GMT'].isin(warm_season)]
df_tou1h_event_cold = df_tou1h_event[df_help_season_event['GMT'].isin(cold_season)]
df_tou1h_event_warm = df_tou1h_event[df_help_season_event['GMT'].isin(warm_season)]
df_tariff_1h_event_cold = df_tariff_1h_event[df_help_season_event['GMT'].isin(cold_season)]
df_tariff_1h_event_warm = df_tariff_1h_event[df_help_season_event['GMT'].isin(warm_season)]
```

1. Groups' Price Responsiveness - K-mean based

The way we analyze the group price responsiveness has three steps:

1) for each hosehold group, build a new dataframe, the first column is GMT, the next columns are user ids in the group, the next column is temperature, then the price, day of a week, hour of a day, then the predicted consumption, and last the price responsiveness

2) build a function that can generate predicted consumption based on the day of a week, temp, hour to choose the coefficients and then make a prediction.

3) plot the properties based on from the smallest unit (day of week, hour, temp) to (day of week, hour), (day of week, temp), (hour, temp), (day of week), (hour), (temp) so we will have very comprehensive price responsiveness of three different groups

In [6]:
```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
n = 3 # cluster numbers
n_house = 1025 # total household numbers
houseLabel = [] # store the 24hour group labels for each hourshold
for i in range(1025):
    houseLabel.append('')
houseLabel = np.array(houseLabel)
cons_type = ['High Consumption', 'Medium Consumption', 'Low Consumption']
fig_all = plt.figure(figsize = (13,90))
for i in range(24):
    if i <= 9:
        x = df_tou1h_nf_cold[df_tou1h_nf_cold.GMT.str.contains('0' + str(i) + ':00:00')]
        x.set_index('GMT', inplace = True)
        x = x.fillna(x.mean()).transpose() # data for the specific hour, and fill in null value with mean
        # PCA without standardization
        pca_nstd = PCA()
        principleComponents_nstd = pca_nstd.fit_transform(x)
        principleDf_nstd = pd.DataFrame(data = principleComponents_nstd)
        kmeans = KMeans(n_clusters = n, init='k-means++').fit(principleDf_nstd.iloc[:,:3])
        houseLabel = np.core.defchararray.add(houseLabel, kmeans.labels_.astype('str'))
    else:
        x = df_tou1h_nf_cold[df_tou1h_nf_cold.GMT.str.contains(str(i) + ':00:00')]
        x.set_index('GMT', inplace = True)
        x = x.fillna(x.mean()).transpose() # data for the specific hour, and filtering out null value
        # PCA without standardization
        pca_nstd = PCA()
        principleComponents_nstd = pca_nstd.fit_transform(x)
        principleDf_nstd = pd.DataFrame(data = principleComponents_nstd)
        kmeans = KMeans(n_clusters = n, init='k-means++').fit(principleDf_nstd.iloc[:,:3])
        cluster_dict = {}
        houseLabel = np.core.defchararray.add(houseLabel, kmeans.labels_.astype('str'))
# We obtain the house Label shown below
# print(houseLabel)

# Group the house based on their 24 hour labels
houseLableDf = pd.DataFrame()
houseLableDf['House'] = df_tou1h_nf_cold.columns[1:]
houseLableDf['Label'] = houseLabel
houseGroupDf = houseLableDf.groupby('Label').size().reset_index(name='counts')
houseGroupDf = houseGroupDf.sort_values(by=['counts'], ascending=False)
houseGroupDf
```

Out[6]:

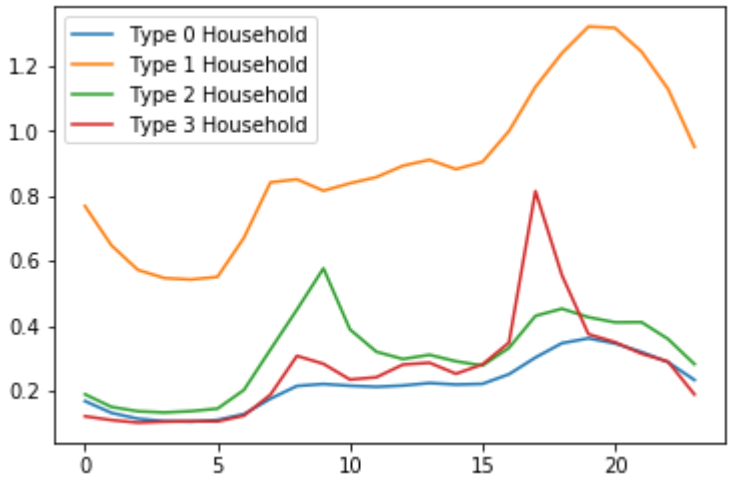
	Label	counts
90	0100001010001010111111110	411
374	2012220102220102020200002	29
151	0100001012001010111111110	11
104	0100001010001010121111110	10
54	0100001002220102020200002	8
96	0100001010001010111121110	6
75	0100001010001010111011110	6
14	0100000010001010111111110	5
271	1221112221112222120202221	5
289	2000001010001010111111110	5
354	2012220002220102020200002	5
422	2100001010001010111111110	4
110	0100001010001012111111110	4
161	0100001012201010111111110	4
45	0100001002001010111111110	4
97	0100001010001010120111110	4
290	2000001010001010111111112	4
85	0100001010001010111110110	3
442	2100001012220102020200002	3
105	0100001010001012011111110	3
414	2100001002220102020200002	3
74	0100001010001010021111110	3
446	2100001102220102020200002	3
68	0100001010001010011111110	3
33	0100001000001010111111110	3
424	2100001010001010120200002	3
307	2000001102220102020200002	3
30	0100000110001010111111110	3
132	0100001010201010111111110	3
60	0100001010000010111111110	3
...	...	...
140	0100001010220102020021000	1
139	0100001010220102020020110	1
138	0100001010220102020020002	1
137	0100001010220102020011002	1
160	0100001012201010110211110	1
162	0100001012201010111120002	1
163	0100001012201010111120110	1
164	0100001012201010111121010	1
189	0100001012221102020211110	1
188	0100001012221012011111110	1
186	0100001012221002020200002	1
185	0100001012220112020211110	1
184	0100001012220110021111110	1
183	0100001012220102120111110	1
182	0100001012220102111111110	1
181	0100001012220102021201112	1
180	0100001012220102021111110	1
178	0100001012220102020200002	1
176	0100001012220102020111110	1

	Label	counts
175	0100001012220102020000002	1
174	0100001012220102011111110	1
173	0100001012220102010200010	1
172	0100001012220102010200002	1
171	0100001012220010111201l0	1
170	0100001012220010111111110	1
169	0100001012201110111211l0	1
168	0100001012201010121111l0	1
167	0100001012201010120200l0	1
165	0100001012201010120000l0	1
451	2212222121112221202200O2	1

452 rows × 2 columns

```
<Figure size 936x6480 with 0 Axes>
```

In [7]:
```
for i in range(4):
    load = []
    for j in range(24):
        if j <= 9:
            x = df_tou1h_nf_cold[df_tou1h_nf_cold.GMT.str.contains('0' + str(j) + ':00:00')]
            x.set_index('GMT', inplace = True)
            x = x.fillna(x.mean())
            x = x[list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[i]].House)]
            load.append(x.mean().mean())
        else:
            x = df_tou1h_nf_cold[df_tou1h_nf_cold.GMT.str.contains(str(j) + ':00:00')]
            x.set_index('GMT', inplace = True)
            x = x.fillna(x.mean())
            x = x[list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[i]].House)]
            load.append(x.mean().mean())
    plt.plot(load, label = 'Type ' + str(i) + ' Household')
plt.legend()
plt.show()
```



In [18]:
```
df_type2
```

Out[18]:

	GMT	D0092	D0109	D0226	D0242	D0448	D0753	D0756	D0795	D0884	D0932	D1001	TempC	TempF	Price	Event_t
0	2013-01-04 14:00:00	0.239	0.173	0.845	0.102	0.274	0.215	0.226	0.107	0.244	0.026	0.078	10.333333	51.000000	0.0399	L3
1	2013-01-04 15:00:00	0.498	0.258	0.526	0.142	0.230	0.378	0.126	0.242	0.163	0.041	0.122	10.000000	51.000000	0.0399	L3
2	2013-01-04 16:00:00	0.789	0.442	0.337	0.062	0.383	0.390	0.211	0.092	0.308	0.042	0.307	9.333333	49.333333	0.0399	L3
3	2013-01-07 23:00:00	0.349	0.358	0.382	0.286	0.445	0.196	0.933	0.056	0.255	0.054	0.135	7.000000	44.000000	0.6720	H3
4	2013-01-08 00:00:00	0.138	0.170	0.238	0.139	0.354	0.093	0.189	0.054	0.211	0.025	0.099	7.000000	44.000000	0.6720	H3
5	2013-01-08 01:00:00	0.128	0.272	0.229	0.123	0.189	0.110	0.126	0.055	0.217	0.043	0.066	6.666667	43.333333	0.6720	H3
6	2013-01-10 02:00:00	0.113	0.249	0.231	0.328	0.106	0.109	0.118	0.055	0.220	0.046	0.097	1.333333	34.333333	0.0399	L3
7	2013-01-10 03:00:00	0.743	0.181	0.214	0.354	0.104	0.113	0.103	0.052	0.240	0.026	0.064	1.000000	34.000000	0.0399	L3
8	2013-01-10 04:00:00	0.810	0.388	0.185	0.356	0.185	0.170	0.099	0.044	0.224	0.046	0.087	1.000000	34.333333	0.0399	L3
9	2013-01-11 11:00:00	0.107	0.262	0.408	0.111	0.211	0.159	0.114	0.097	0.382	0.108	0.168	2.666667	36.666667	0.6720	H3
10	2013-01-11 12:00:00	0.195	0.234	0.388	0.110	0.210	0.312	0.103	0.090	0.341	0.302	0.070	4.000000	39.000000	0.6720	H3
11	2013-01-11 13:00:00	0.100	0.393	0.470	0.112	0.435	0.302	0.381	0.044	0.404	0.107	0.078	4.000000	39.000000	0.6720	H3
12	2013-01-13 05:00:00	0.101	0.207	0.266	0.073	0.098	0.097	0.119	0.032	0.211	0.026	0.065	-1.666667	29.333333	0.6720	H6
13	2013-01-13 06:00:00	0.117	0.258	0.211	0.077	0.125	0.092	0.117	0.016	0.244	0.049	0.288	-2.000000	29.000000	0.6720	H6
14	2013-01-13 07:00:00	0.159	0.325	0.256	0.083	0.279	0.127	0.252	0.033	0.278	0.026	0.312	-1.333333	30.000000	0.6720	H6
15	2013-01-13 08:00:00	0.602	0.304	0.253	0.076	0.449	0.471	0.695	0.134	0.322	0.045	0.447	-0.666667	31.000000	0.6720	H6
16	2013-01-13 09:00:00	0.844	0.384	0.493	0.070	0.368	0.910	0.266	1.978	0.412	0.062	0.222	0.000000	32.000000	0.6720	H6
17	2013-01-13 10:00:00	0.969	0.373	0.477	0.176	0.407	0.908	0.143	0.141	0.406	0.679	0.137	0.666667	33.000000	0.6720	H6
18	2013-01-16 23:00:00	0.246	0.218	0.375	1.611	0.338	0.236	0.295	0.235	0.267	0.062	0.118	-5.333333	22.333333	0.6720	H3
19	2013-01-17 00:00:00	0.132	0.212	0.182	0.162	0.261	0.189	0.128	0.051	0.220	0.066	0.083	-6.000000	21.000000	0.6720	H3

	GMT	D0092	D0109	D0226	D0242	D0448	D0753	D0756	D0795	D0884	D0932	D1001	TempC	TempF	Price	Event_t
20	2013-01-17 01:00:00	0.107	0.206	0.220	0.226	0.131	0.088	0.104	0.031	0.240	0.041	0.068	-5.333333	22.000000	0.6720	H3
21	2013-01-19 05:00:00	0.101	0.265	0.365	0.058	0.124	0.116	0.109	0.038	0.210	0.026	0.064	-1.666667	29.333333	0.0399	CM
22	2013-01-19 06:00:00	0.227	0.211	0.432	0.068	0.121	0.106	0.108	0.032	0.206	0.107	0.163	-2.000000	29.000000	0.0399	CM
23	2013-01-19 07:00:00	0.384	0.681	0.511	0.061	0.244	0.105	0.179	0.044	0.277	0.111	0.316	-1.666667	29.666667	0.0399	CM
24	2013-01-19 08:00:00	1.523	0.325	0.395	0.346	0.487	0.419	0.153	0.906	0.858	0.035	0.340	-1.333333	30.333333	0.0399	CM
25	2013-01-19 09:00:00	1.003	0.215	0.518	2.264	0.372	0.511	0.379	1.362	0.658	1.507	1.136	-1.000000	31.000000	0.0399	CM
26	2013-01-19 10:00:00	1.757	0.206	0.489	1.055	0.568	0.175	0.309	0.057	0.799	1.750	0.127	-0.666667	31.333333	0.0399	CM
27	2013-01-19 11:00:00	0.673	0.239	0.383	2.045	0.876	0.380	0.930	0.075	0.456	1.879	0.201	-0.333333	31.666667	0.0399	CM
28	2013-01-19 12:00:00	0.522	0.264	0.468	1.889	0.542	0.160	0.358	0.057	0.335	1.464	0.107	0.000000	32.000000	0.0399	CM
29	2013-01-19 13:00:00	0.713	0.214	0.414	0.277	0.413	0.332	0.260	0.024	0.299	1.003	0.066	0.333333	32.333333	0.0399	CM
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
549	2013-12-19 14:00:00	0.123	0.258	0.376	0.059	0.438	0.303	0.234	0.156	0.190	1.576	0.153	5.000000	42.000000	0.0399	L24
550	2013-12-19 15:00:00	0.167	0.269	0.508	0.123	0.259	0.195	0.134	0.062	0.261	0.164	0.175	5.000000	42.000000	0.0399	L24
551	2013-12-19 16:00:00	0.174	0.541	0.433	0.032	0.351	0.227	0.210	0.062	0.241	0.096	0.920	5.000000	42.000000	0.0399	L24
552	2013-12-19 17:00:00	0.435	0.701	0.503	0.055	0.560	0.684	0.311	0.061	0.485	0.112	0.576	5.000000	42.000000	0.0399	L24
553	2013-12-19 18:00:00	0.390	0.641	0.546	0.059	0.718	0.524	0.672	0.116	0.640	0.092	0.132	5.000000	42.000000	0.0399	L24
554	2013-12-19 19:00:00	0.577	0.635	0.492	0.053	0.655	0.442	0.480	0.134	0.642	0.226	0.233	5.000000	42.000000	0.0399	L24
555	2013-12-19 20:00:00	0.452	0.747	0.512	0.212	0.500	0.403	0.327	0.129	0.823	0.807	0.311	5.000000	42.000000	0.0399	L24
556	2013-12-19 21:00:00	0.375	0.588	0.632	0.332	0.373	0.241	0.410	0.255	0.685	0.148	0.357	5.000000	42.000000	0.0399	L24
557	2013-12-19 22:00:00	0.337	0.342	0.538	0.423	0.388	0.244	0.543	0.144	0.595	0.051	0.279	5.000000	41.333333	0.0399	L24
558	2013-12-19 23:00:00	0.161	0.305	0.418	0.232	0.367	0.252	0.589	0.031	0.450	0.066	0.201	5.000000	40.666667	0.0399	L24

	GMT	D0092	D0109	D0226	D0242	D0448	D0753	D0756	D0795	D0884	D0932	D1001	TempC	TempF	Price	Event_t
559	2013-12-20 00:00:00	0.124	0.195	0.241	0.185	0.281	0.133	0.809	0.032	0.238	0.037	0.117	5.000000	40.000000	0.0399	L24
560	2013-12-20 01:00:00	0.170	0.224	0.201	0.118	0.116	0.119	0.562	0.032	0.203	0.029	0.086	5.000000	40.000000	0.0399	L24
561	2013-12-20 02:00:00	0.164	0.199	0.231	0.100	0.154	0.117	0.123	0.021	0.238	0.035	0.070	5.000000	40.000000	0.0399	L24
562	2013-12-20 03:00:00	0.104	0.231	0.226	0.098	0.127	0.167	0.123	0.026	0.233	0.032	0.087	5.000000	40.000000	0.0399	L24
563	2013-12-20 04:00:00	0.113	0.224	0.209	0.105	0.110	0.191	0.123	0.032	0.201	0.029	0.062	5.000000	40.000000	0.0399	L24
564	2013-12-26 08:00:00	0.583	0.452	0.448	0.088	0.264	0.727	0.675	0.043	0.188	0.031	0.348	4.000000	40.000000	0.0399	L3
565	2013-12-26 09:00:00	1.126	0.859	0.435	0.176	0.182	0.516	0.729	0.023	0.232	0.366	0.970	4.000000	40.000000	0.0399	L3
566	2013-12-26 10:00:00	0.439	0.440	0.515	0.385	0.113	0.344	0.569	0.032	0.224	1.334	0.549	4.000000	40.000000	0.0399	L3
567	2013-12-29 17:00:00	0.164	0.661	0.404	0.277	0.570	0.297	0.469	0.198	0.623	1.312	0.246	3.000000	38.000000	0.0399	L12
568	2013-12-29 18:00:00	0.210	0.574	0.601	0.444	0.440	0.435	0.790	0.173	0.581	0.134	0.299	3.000000	38.000000	0.0399	L12
569	2013-12-29 19:00:00	0.229	0.541	0.656	0.294	0.516	0.441	0.588	0.185	0.575	0.087	0.317	3.000000	38.000000	0.0399	L12
570	2013-12-29 20:00:00	0.423	0.631	0.496	0.227	0.661	0.407	0.445	0.162	0.667	0.174	0.303	3.000000	38.000000	0.0399	L12
571	2013-12-29 21:00:00	0.297	0.550	0.651	0.275	0.494	0.222	0.348	0.301	0.583	0.172	0.343	3.000000	38.000000	0.0399	L12
572	2013-12-29 22:00:00	0.503	0.439	0.525	0.514	0.503	0.260	0.400	0.344	0.431	0.049	0.268	4.666667	40.666667	0.0399	L12
573	2013-12-29 23:00:00	0.178	0.387	0.540	0.314	0.384	0.182	0.631	0.194	0.267	0.032	0.162	6.333333	43.333333	0.0399	L12
574	2013-12-30 00:00:00	0.104	0.234	0.252	0.203	0.276	0.116	0.241	0.046	0.246	0.032	0.070	8.000000	46.000000	0.0399	L12
575	2013-12-30 01:00:00	0.122	0.301	0.214	0.155	0.146	0.156	0.215	0.027	0.210	0.032	0.068	8.000000	46.000000	0.0399	L12
576	2013-12-30 02:00:00	0.103	0.188	0.209	0.089	0.112	0.100	0.230	0.023	0.199	0.032	0.081	8.000000	46.000000	0.0399	L12
577	2013-12-30 03:00:00	0.120	0.173	0.207	0.089	0.186	0.096	0.230	0.033	0.283	0.033	0.083	8.000000	46.000000	0.0399	L12
578	2013-12-30 04:00:00	0.113	0.303	0.212	0.090	0.110	0.121	0.224	0.033	0.255	0.029	0.080	8.000000	46.000000	0.0399	L12

579 rows × 20 columns

In [20]:
```
# Type 3 household
# build a new dataframe with columns: GMT, user ids in the group
df_type3 = df_tou1h_event_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[3]].
House)]
df_type3 = pd.merge(df_type3, df_wea1h_event_cold, on = 'GMT') # add weather
df_type3 = pd.merge(df_type3, df_tariff_1h_event_cold, on = 'GMT') # add price
df_type3['Day of week'] = pd.to_datetime(df_type3.GMT).dt.dayofweek # add day of week
df_type3['Hour of day'] = pd.to_datetime(df_type3.GMT).dt.hour # add hour of day
df_type3['Household type'] = 3
df_type3
```

Out[20]:

	GMT	D0107	D0182	D0239	D0272	D0304	D0361	D0476	D0569	D0573	D0627	TempC	TempF	Price	Event_tags	D we
0	2013-01-04 14:00:00	1.403	0.279	0.348	0.422	0.161	1.116	0.217	0.436	0.094	0.257	10.333333	51.000000	0.0399	L3	4
1	2013-01-04 15:00:00	1.334	0.139	0.213	1.461	0.165	0.669	0.066	0.855	0.544	0.205	10.000000	51.000000	0.0399	L3	4
2	2013-01-04 16:00:00	1.745	0.151	0.487	1.618	0.080	1.747	0.089	0.392	0.128	0.289	9.333333	49.333333	0.0399	L3	4
3	2013-01-07 23:00:00	0.467	0.098	0.260	0.233	0.145	0.113	0.137	0.201	0.157	0.350	7.000000	44.000000	0.6720	H3	0
4	2013-01-08 00:00:00	0.210	0.142	0.163	0.182	0.086	0.057	0.114	0.077	0.108	0.153	7.000000	44.000000	0.6720	H3	1
5	2013-01-08 01:00:00	0.130	0.072	0.157	0.150	0.052	0.129	0.069	0.077	0.094	0.126	6.666667	43.333333	0.6720	H3	1
6	2013-01-10 02:00:00	0.119	0.063	0.157	0.195	0.060	0.048	0.068	0.072	0.171	0.156	1.333333	34.333333	0.0399	L3	3
7	2013-01-10 03:00:00	0.159	0.062	0.184	0.192	0.449	0.142	0.082	0.074	0.171	0.088	1.000000	34.000000	0.0399	L3	3
8	2013-01-10 04:00:00	0.134	0.052	0.221	0.146	0.108	0.045	0.085	0.078	0.169	0.132	1.000000	34.333333	0.0399	L3	3
9	2013-01-11 11:00:00	0.550	0.067	0.360	1.912	0.062	0.212	0.148	1.270	0.102	0.235	2.666667	36.666667	0.6720	H3	4
10	2013-01-11 12:00:00	0.159	0.098	0.285	0.288	0.088	0.257	0.122	0.374	0.103	0.223	4.000000	39.000000	0.6720	H3	4
11	2013-01-11 13:00:00	0.128	0.277	0.584	0.351	0.075	0.316	0.093	0.651	0.088	0.176	4.000000	39.000000	0.6720	H3	4
12	2013-01-13 05:00:00	0.157	0.049	0.152	0.145	0.061	0.077	0.060	0.087	0.105	0.129	-1.666667	29.333333	0.6720	H6	6
13	2013-01-13 06:00:00	0.168	0.060	0.146	0.150	0.074	0.137	0.157	0.069	0.114	0.113	-2.000000	29.000000	0.6720	H6	6
14	2013-01-13 07:00:00	0.229	0.061	0.191	0.402	0.081	0.047	0.088	0.082	0.208	0.125	-1.333333	30.000000	0.6720	H6	6
15	2013-01-13 08:00:00	0.292	0.052	0.248	0.244	0.126	0.363	0.335	0.377	0.182	0.220	-0.666667	31.000000	0.6720	H6	6
16	2013-01-13 09:00:00	0.207	0.275	0.442	0.140	0.172	0.313	0.379	1.188	0.242	0.451	0.000000	32.000000	0.6720	H6	6
17	2013-01-13 10:00:00	0.206	0.145	0.459	0.117	0.046	0.330	0.199	0.214	0.131	0.230	0.666667	33.000000	0.6720	H6	6
18	2013-01-16 23:00:00	0.302	0.071	0.253	0.338	0.096	0.046	0.062	0.167	0.272	0.336	-5.333333	22.333333	0.6720	H3	2
19	2013-01-17 00:00:00	0.263	0.058	0.151	0.137	0.093	0.111	0.068	0.108	0.098	0.167	-6.000000	21.000000	0.6720	H3	3

	GMT	D0107	D0182	D0239	D0272	D0304	D0361	D0476	D0569	D0573	D0627	TempC	TempF	Price	Event_tags	D we
20	2013-01-17 01:00:00	0.175	0.052	0.145	0.097	0.066	0.071	0.068	0.067	0.095	0.101	-5.333333	22.000000	0.6720	H3	3
21	2013-01-19 05:00:00	0.200	0.081	0.140	0.188	0.051	0.049	0.193	0.076	0.096	0.087	-1.666667	29.333333	0.0399	CM	5
22	2013-01-19 06:00:00	0.171	0.055	0.137	0.400	0.105	0.114	0.077	0.072	0.127	0.155	-2.000000	29.000000	0.0399	CM	5
23	2013-01-19 07:00:00	0.169	0.053	0.184	0.189	0.253	0.072	0.084	0.145	0.194	0.127	-1.666667	29.666667	0.0399	CM	5
24	2013-01-19 08:00:00	0.323	0.059	0.247	0.370	1.466	0.553	0.093	0.386	0.155	0.393	-1.333333	30.333333	0.0399	CM	5
25	2013-01-19 09:00:00	0.236	0.368	0.479	0.179	0.352	1.327	0.049	0.360	0.271	0.305	-1.000000	31.000000	0.0399	CM	5
26	2013-01-19 10:00:00	0.873	0.236	0.627	0.504	0.219	0.894	0.092	0.198	0.241	0.204	-0.666667	31.333333	0.0399	CM	5
27	2013-01-19 11:00:00	0.505	0.152	0.288	0.952	0.177	0.577	0.141	0.116	0.096	0.207	-0.333333	31.666667	0.0399	CM	5
28	2013-01-19 12:00:00	0.398	0.297	0.515	0.720	0.290	0.297	0.110	0.165	0.148	0.192	0.000000	32.000000	0.0399	CM	5
29	2013-01-19 13:00:00	0.814	0.214	0.571	1.898	0.285	0.327	0.367	0.290	0.207	0.218	0.333333	32.333333	0.0399	CM	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
549	2013-12-19 14:00:00	0.250	0.065	0.288	1.372	0.052	0.394	0.071	0.397	0.083	0.239	5.000000	42.000000	0.0399	L24	3
550	2013-12-19 15:00:00	0.195	0.231	0.482	1.440	0.187	0.567	0.132	0.350	0.453	1.027	5.000000	42.000000	0.0399	L24	3
551	2013-12-19 16:00:00	0.260	0.251	0.440	0.453	0.301	1.251	0.078	0.465	0.196	1.347	5.000000	42.000000	0.0399	L24	3
552	2013-12-19 17:00:00	0.226	1.161	0.704	1.449	0.915	1.600	0.414	0.980	0.595	2.114	5.000000	42.000000	0.0399	L24	3
553	2013-12-19 18:00:00	0.221	0.374	0.587	0.470	0.416	0.481	0.331	0.768	0.775	0.565	5.000000	42.000000	0.0399	L24	3
554	2013-12-19 19:00:00	0.293	0.147	0.401	0.299	0.301	0.484	0.180	0.291	0.422	0.242	5.000000	42.000000	0.0399	L24	3
555	2013-12-19 20:00:00	0.454	0.142	0.474	0.187	0.365	0.461	0.201	0.386	0.485	0.276	5.000000	42.000000	0.0399	L24	3
556	2013-12-19 21:00:00	0.440	0.118	0.590	0.154	0.232	0.374	0.218	0.359	0.397	0.295	5.000000	42.000000	0.0399	L24	3
557	2013-12-19 22:00:00	0.608	0.136	0.543	0.176	0.259	0.489	0.153	0.257	0.258	0.256	5.000000	41.333333	0.0399	L24	3
558	2013-12-19 23:00:00	0.311	0.078	0.275	0.138	0.070	0.122	0.118	0.141	0.127	0.271	5.000000	40.666667	0.0399	L24	3

	GMT	D0107	D0182	D0239	D0272	D0304	D0361	D0476	D0569	D0573	D0627	TempC	TempF	Price	Event_tags	D we
559	2013-12-20 00:00:00	0.143	0.062	0.150	0.129	0.136	0.061	0.075	0.090	0.055	0.183	5.000000	40.000000	0.0399	L24	4
560	2013-12-20 01:00:00	0.170	0.061	0.137	0.126	0.055	0.154	0.084	0.068	0.058	0.090	5.000000	40.000000	0.0399	L24	4
561	2013-12-20 02:00:00	0.153	0.071	0.123	0.109	0.042	0.108	0.076	0.087	0.051	0.115	5.000000	40.000000	0.0399	L24	4
562	2013-12-20 03:00:00	0.143	0.078	0.183	0.128	0.033	0.063	0.066	0.070	0.140	0.161	5.000000	40.000000	0.0399	L24	4
563	2013-12-20 04:00:00	0.139	0.059	0.172	0.119	0.057	0.165	0.061	0.071	0.100	0.089	5.000000	40.000000	0.0399	L24	4
564	2013-12-26 08:00:00	0.205	0.045	0.279	0.170	0.063	0.383	0.511	0.372	0.165	0.230	4.000000	40.000000	0.0399	L3	3
565	2013-12-26 09:00:00	0.236	0.254	0.317	0.308	0.248	0.769	0.394	0.837	0.101	0.157	4.000000	40.000000	0.0399	L3	3
566	2013-12-26 10:00:00	0.193	0.634	0.237	0.202	0.221	0.470	0.182	0.433	0.052	0.085	4.000000	40.000000	0.0399	L3	3
567	2013-12-29 17:00:00	0.185	0.103	0.500	0.588	0.412	0.542	0.260	1.000	0.268	0.343	3.000000	38.000000	0.0399	L12	6
568	2013-12-29 18:00:00	0.204	0.295	0.519	0.487	0.681	0.492	0.260	0.315	0.267	0.288	3.000000	38.000000	0.0399	L12	6
569	2013-12-29 19:00:00	0.202	0.113	0.573	0.437	0.303	0.646	0.229	0.279	0.273	0.273	3.000000	38.000000	0.0399	L12	6
570	2013-12-29 20:00:00	0.158	0.110	0.480	0.200	0.266	0.760	1.093	0.290	0.268	NaN	3.000000	38.000000	0.0399	L12	6
571	2013-12-29 21:00:00	0.220	0.091	0.626	0.265	0.144	0.634	0.325	0.238	0.270	0.290	3.000000	38.000000	0.0399	L12	6
572	2013-12-29 22:00:00	0.194	0.083	0.445	0.252	0.224	0.450	0.317	0.246	0.169	0.311	4.666667	40.666667	0.0399	L12	6
573	2013-12-29 23:00:00	0.173	0.048	0.266	NaN	0.089	0.102	0.106	0.085	0.128	0.208	6.333333	43.333333	0.0399	L12	6
574	2013-12-30 00:00:00	0.166	0.050	0.136	0.167	0.035	0.154	0.069	0.083	0.063	0.132	8.000000	46.000000	0.0399	L12	0
575	2013-12-30 01:00:00	0.178	0.087	0.143	0.148	0.062	0.160	0.083	0.064	0.077	0.138	8.000000	46.000000	0.0399	L12	0
576	2013-12-30 02:00:00	0.134	0.055	0.132	0.145	0.056	0.159	0.097	0.081	0.062	0.088	8.000000	46.000000	0.0399	L12	0
577	2013-12-30 03:00:00	0.188	0.056	0.178	0.128	0.020	0.150	0.066	0.062	0.065	0.116	8.000000	46.000000	0.0399	L12	0
578	2013-12-30 04:00:00	0.159	0.065	0.191	0.145	0.056	0.061	0.062	0.071	0.063	0.153	8.000000	46.000000	0.0399	L12	0

579 rows × 18 columns

In [8]:
```
# Type 0 household
# build a new dataframe with columns: GMT, user ids in the group
df_type0 = df_tou1h_event_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[0]].
House)]
df_type0 = pd.merge(df_type0, df_wea1h_event_cold, on = 'GMT') # add weather
df_type0 = pd.merge(df_type0, df_tariff_1h_event_cold, on = 'GMT') # add price
df_type0['Day of week'] = pd.to_datetime(df_type0.GMT).dt.dayofweek # add day of week
df_type0['Hour of day'] = pd.to_datetime(df_type0.GMT).dt.hour # add hour of day
df_type0['Household type'] = 0

# Type 1 household
# build a new dataframe with columns: GMT, user ids in the group
df_type1 = df_tou1h_event_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[1]].
House)]
df_type1 = pd.merge(df_type1, df_wea1h_event_cold, on = 'GMT') # add weather
df_type1 = pd.merge(df_type1, df_tariff_1h_event_cold, on = 'GMT') # add price
df_type1['Day of week'] = pd.to_datetime(df_type1.GMT).dt.dayofweek # add day of week
df_type1['Hour of day'] = pd.to_datetime(df_type1.GMT).dt.hour # add hour of day
df_type1['Household type'] = 1

# Type 2 household
# build a new dataframe with columns: GMT, user ids in the group
df_type2 = df_tou1h_event_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[2]].
House)]
df_type2 = pd.merge(df_type2, df_wea1h_event_cold, on = 'GMT') # add weather
df_type2 = pd.merge(df_type2, df_tariff_1h_event_cold, on = 'GMT') # add price
df_type2['Day of week'] = pd.to_datetime(df_type2.GMT).dt.dayofweek # add day of week
df_type2['Hour of day'] = pd.to_datetime(df_type2.GMT).dt.hour # add hour of day
df_type2['Household type'] = 2
```

With the household type and feature data ready, we need to build functions that can predict the consumptions during these time periods. Since we are considering group properties, the above users are all in the first group, for each time period, we only have one consumption predicted for the whole group, but since the regression coefficients depend on day of week, temperature and hour of day, so it should be quite different for different time.

In [9]:
```
def predict(dfRegression, dayOfWeek, hour, temperature):
    '''
    Predict the consumption duruing event time based on
    the regression, day of week, hour of a day and temperature
    '''
    weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
    output = dfRegression[weeks[dayOfWeek]].iloc[hour](temperature)
    return output
```

In [10]:
```
# store the regression coefficients
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
df_regression_0 = {} # store the regression coefficients
df_regression_1 = {}
df_regression_2 = {}
# df_regression_3 = {}
lm = LinearRegression()
for g in range(3): # doing this for three groups
    for day_of_week in range(7):
        coef_temp = []
        for i in range(24):
            if i <= 9:
                z = df_tou1h_nf_cold[df_tou1h_nf_cold.GMT.str.contains('0' + str(i) + ':00:00')]
                y_temp = z[['GMT'] + list(houseLableDf[(houseLableDf.Label == houseGroupDf.Label.iloc[g]) ].H
ouse)]
                y = y_temp[pd.to_datetime(z.GMT).dt.dayofweek == day_of_week] # find the users' data for Mond
ay
                y.set_index('GMT', inplace = True)
                y = y.fillna(y.mean()).values
                y = np.reshape(y, y.shape[0] * y.shape[1], order = 'F') # reconstruct the y variable
                x = df_wea1h_nf_cold.TempC[(df_wea1h.GMT.str.contains('0' + str(i) + ':00:00')) & (pd.to_date
time(z.GMT).dt.dayofweek == day_of_week)].values
                x = np.tile(x, reps = sum(houseLableDf.Label == houseGroupDf.Label.iloc[g])) # reconstruct th
e x variable
                z = np.polyfit(x, y, 2)
                p = np.poly1d(z)
                coef_temp.append(p)
            else:
                z = df_tou1h_nf_cold[df_tou1h_nf_cold.GMT.str.contains(str(i) + ':00:00')]
                y_temp = z[['GMT'] + list(houseLableDf[(houseLableDf.Label == houseGroupDf.Label.iloc[g]) ].H
ouse)]
                y = y_temp[pd.to_datetime(z.GMT).dt.dayofweek == day_of_week] # find the users' data for Mond
ay
                y.set_index('GMT', inplace = True)
                y = y.fillna(y.mean()).values
                y = np.reshape(y, y.shape[0] * y.shape[1], order = 'F') # reconstruct the y variable

                x = df_wea1h_nf_cold.TempC[(df_wea1h.GMT.str.contains(str(i) + ':00:00')) & (pd.to_datetime(z
.GMT).dt.dayofweek == day_of_week)].values
                x = np.tile(x, reps = sum(houseLableDf.Label == houseGroupDf.Label.iloc[g])) # reconstruct th
e x variable
                z = np.polyfit(x, y, 2)
                p = np.poly1d(z)
                coef_temp.append(p)
        if g == 0:
            df_regression_0[weeks[day_of_week]] = coef_temp
        elif g == 1:
            df_regression_1[weeks[day_of_week]] = coef_temp
        else:
            df_regression_2[weeks[day_of_week]] = coef_temp
df_regression_0 = pd.DataFrame(df_regression_0) # type 0 regression coefficient
df_regression_1 = pd.DataFrame(df_regression_1) # type 1
df_regression_2 = pd.DataFrame(df_regression_2) # type 2
```

In [14]:
```
df_regression_0.Monday[0]
```

Out[14]:
```
poly1d([ 1.50410346e-04, -1.97269603e-03,  1.64856830e-01])
```

In [11]:
```
# type 0 event time consumption prediction
pred_result = []
for i in range(df_type0.shape[0]):
    pred_result.append(predict(df_regression_0, df_type0['Day of week'].iloc[i], df_type0['Hour of day'].iloc
[i], df_type0['TempC'].iloc[i]))
df_type0['Predicted consumption'] = pred_result
# price responsiveness elementwise
df_type0_res = df_type0.copy()
df_type0_res.iloc[:,1:houseGroupDf.counts.iloc[0] + 1] = df_type0.iloc[:,1:houseGroupDf.counts.iloc[0] + 1].s
ub(df_type0['Predicted consumption'], axis = 0)

# type 1 event time consumption prediction
pred_result = []
for i in range(df_type1.shape[0]):
    pred_result.append(predict(df_regression_1, df_type1['Day of week'].iloc[i], df_type1['Hour of day'].iloc
[i], df_type1['TempC'].iloc[i]))
df_type1['Predicted consumption'] = pred_result

# price responsiveness elementwise
df_type1_res = df_type1.copy()
df_type1_res.iloc[:,1:houseGroupDf.counts.iloc[1] + 1] = df_type1.iloc[:,1:houseGroupDf.counts.iloc[1] + 1].s
ub(df_type1['Predicted consumption'], axis = 0)

# type 2 event time consumption prediction
pred_result = []
for i in range(df_type2.shape[0]):
    pred_result.append(predict(df_regression_2, df_type2['Day of week'].iloc[i], df_type2['Hour of day'].iloc
[i], df_type2['TempC'].iloc[i]))
df_type2['Predicted consumption'] = pred_result

# price responsiveness elementwise
df_type2_res = df_type2.copy()
df_type2_res.iloc[:,1:houseGroupDf.counts.iloc[2] + 1] = df_type2.iloc[:,1:houseGroupDf.counts.iloc[2] + 1].s
ub(df_type2['Predicted consumption'], axis = 0)
```

To visualize the price responsiveness, we need to reformat the data and melt them to a long dataframe, and all the user ID columns will be extracted to form a individual dataframe with adding household group label and we concatenate all of them in a long dataframe.

In [13]:
```
houseGroupDf.counts.iloc[0]
```

Out[13]:
```
407
```

In [14]:
```
df_type0_res.head()
```

Out[14]:

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024	
0	2013-01-04 14:00:00	-0.085446	0.051554	0.004554	-0.039446	-0.169446	0.404554	0.065554	-0.118446	-0.100446	...	-0.174446	-0.055446	10.
1	2013-01-04 15:00:00	-0.095975	0.823025	0.451025	-0.040975	-0.162975	0.836025	0.735025	-0.130975	0.168025	...	-0.160975	-0.105975	10.
2	2013-01-04 16:00:00	1.131656	0.084656	0.314656	0.365656	-0.203344	0.553656	0.507656	-0.173344	0.404656	...	-0.004344	-0.043344	9.3
3	2013-01-07 23:00:00	0.092168	0.122168	0.077168	0.034168	-0.195832	0.347168	-0.102832	-0.124832	0.303168	...	-0.095832	0.245168	7.0
4	2013-01-08 00:00:00	-0.066635	0.064365	0.112365	0.104365	-0.140635	0.043365	-0.061635	-0.094635	0.009365	...	-0.059635	-0.015635	7.0

5 rows × 416 columns

In [15]:
```
df_type0_res
```

Out[15]:

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024
0	2013-01-04 14:00:00	-0.085446	0.051554	0.004554	-0.039446	-0.169446	0.404554	0.065554	-0.118446	-0.100446	...	-0.174446	-0.055446
1	2013-01-04 15:00:00	-0.095975	0.823025	0.451025	-0.040975	-0.162975	0.836025	0.735025	-0.130975	0.168025	...	-0.160975	-0.105975
2	2013-01-04 16:00:00	1.131656	0.084656	0.314656	0.365656	-0.203344	0.553656	0.507656	-0.173344	0.404656	...	-0.004344	-0.043344
3	2013-01-07 23:00:00	0.092168	0.122168	0.077168	0.034168	-0.195832	0.347168	-0.102832	-0.124832	0.303168	...	-0.095832	0.245168
4	2013-01-08 00:00:00	-0.066635	0.064365	0.112365	0.104365	-0.140635	0.043365	-0.061635	-0.094635	0.009365	...	-0.059635	-0.015635
5	2013-01-08 01:00:00	-0.033712	0.034288	0.060288	0.041288	-0.100712	0.019288	-0.028712	-0.065712	-0.034712	...	-0.000712	0.006288
6	2013-01-10 02:00:00	-0.015063	0.008937	0.728937	0.061937	-0.089063	-0.013063	-0.012063	-0.040063	-0.040063	...	-0.081063	0.115937
7	2013-01-10 03:00:00	-0.015244	0.003756	0.382756	0.052756	-0.082244	-0.017244	-0.021244	-0.040244	-0.019244	...	-0.096244	0.086756
8	2013-01-10 04:00:00	0.113174	-0.032826	0.199174	0.094174	-0.086826	0.088174	0.000174	-0.036826	-0.051826	...	-0.049826	0.077174
9	2013-01-11 11:00:00	-0.099746	-0.005746	0.012254	0.117254	-0.184746	-0.099746	-0.048746	-0.133746	-0.103746	...	-0.027746	-0.047746
10	2013-01-11 12:00:00	-0.143034	-0.025034	-0.036034	0.188966	-0.082034	-0.034034	0.050966	-0.121034	-0.100034	...	-0.173034	-0.075034
11	2013-01-11 13:00:00	-0.101451	-0.003451	-0.068451	0.344549	-0.132451	-0.098451	-0.035451	-0.122451	-0.149451	...	-0.186451	0.146549
12	2013-01-13 05:00:00	0.130964	-0.017036	0.060964	0.037964	-0.087036	0.027964	-0.015036	-0.035036	-0.057036	...	-0.092036	0.040964
13	2013-01-13 06:00:00	-0.026602	0.038398	0.033398	0.285398	-0.087602	-0.022602	-0.001602	-0.042602	-0.029602	...	-0.061602	0.088398
14	2013-01-13 07:00:00	-0.079585	-0.069585	0.038415	0.300415	-0.130585	0.087415	-0.068585	-0.072585	0.130415	...	-0.133585	0.026415
15	2013-01-13 08:00:00	-0.111316	0.255684	0.079684	0.052684	-0.179316	0.050684	-0.084316	0.111684	0.252684	...	-0.197316	0.155684
16	2013-01-13 09:00:00	-0.144840	0.110160	-0.052840	0.012160	-0.227840	-0.001840	-0.061840	1.322160	0.060160	...	-0.187840	0.094160
17	2013-01-13 10:00:00	-0.182303	0.122697	-0.059303	0.069697	-0.225303	-0.069303	0.028697	0.029697	0.049697	...	-0.241303	-0.089303
18	2013-01-16 23:00:00	-0.172678	0.224322	0.188322	0.145322	-0.246678	0.133322	-0.128678	-0.161678	0.486322	...	-0.062678	0.119322
19	2013-01-17 00:00:00	-0.115459	-0.089459	0.130541	0.113541	-0.181459	-0.102459	-0.103459	-0.139459	0.209541	...	0.126541	0.061541

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024
20	2013-01-17 01:00:00	-0.071176	-0.029176	0.003824	0.095824	-0.130176	-0.025176	-0.063176	-0.078176	0.167824	...	0.012824	0.085824
21	2013-01-19 05:00:00	-0.028058	-0.030058	0.079942	0.275942	-0.080058	-0.011058	-0.005058	-0.054058	-0.001058	...	-0.094058	0.052942
22	2013-01-19 06:00:00	-0.024681	-0.011681	0.017319	0.236319	-0.100681	0.015319	-0.017681	-0.043681	-0.069681	...	-0.081681	0.085319
23	2013-01-19 07:00:00	-0.060236	-0.050236	0.016764	0.159764	-0.137236	0.050764	-0.066236	-0.098236	-0.098236	...	-0.114236	0.034764
24	2013-01-19 08:00:00	-0.146091	-0.033091	0.138909	0.189909	-0.182091	-0.038091	-0.061091	0.732909	-0.151091	...	-0.201091	-0.041091
25	2013-01-19 09:00:00	-0.111838	0.151162	0.160162	0.147162	-0.223838	0.074162	-0.014838	0.066162	-0.174838	...	-0.225838	0.157162
26	2013-01-19 10:00:00	0.692799	0.229799	0.040799	-0.027201	-0.247201	-0.107201	0.935799	0.196799	-0.067201	...	-0.195201	0.976799
27	2013-01-19 11:00:00	0.326361	-0.177639	0.090361	0.109361	-0.241639	0.170361	-0.047639	0.121361	-0.014639	...	-0.107639	0.038361
28	2013-01-19 12:00:00	0.077701	0.006701	0.086701	0.081701	-0.160299	1.068701	-0.066299	0.740701	-0.091299	...	-0.145299	-0.055299
29	2013-01-19 13:00:00	-0.156781	0.085219	0.091219	0.056219	-0.240781	-0.063781	0.328219	0.205219	-0.146781	...	-0.210781	-0.020781
...	...	...	...	...	...	...	...	...	...	...	...	...	...
549	2013-12-19 14:00:00	-0.093868	0.100132	0.126132	0.088132	-0.188868	0.209132	-0.081868	-0.128868	-0.174868	...	-0.151868	0.012132
550	2013-12-19 15:00:00	-0.102815	-0.050815	0.087185	0.169185	-0.192815	0.043185	0.016185	-0.124815	-0.175815	...	-0.183815	-0.035815
551	2013-12-19 16:00:00	-0.084363	-0.104363	0.168637	0.228637	-0.226363	0.125637	0.054637	0.512637	-0.172363	...	0.562637	0.554637
552	2013-12-19 17:00:00	0.231253	0.242253	0.123253	0.473253	-0.289747	0.152253	0.046253	-0.194747	-0.266747	...	-0.115747	0.627253
553	2013-12-19 18:00:00	0.206339	0.200339	0.088339	0.282339	-0.319661	0.175339	0.317339	-0.270661	-0.298661	...	-0.189661	-0.141661
554	2013-12-19 19:00:00	0.283742	0.231742	0.055742	0.142742	-0.362258	0.251742	0.559742	-0.311258	-0.168258	...	-0.216258	-0.200258
555	2013-12-19 20:00:00	0.153854	0.051854	0.158854	0.091854	-0.332146	0.172854	-0.056146	-0.082146	0.128854	...	-0.197146	-0.153146
556	2013-12-19 21:00:00	0.147662	0.027662	0.107662	0.037662	-0.298338	0.185662	-0.085338	-0.195338	-0.148338	...	-0.109338	-0.145338
557	2013-12-19 22:00:00	0.124157	0.165157	0.146157	0.022157	-0.271843	0.653157	-0.056843	-0.136843	-0.055843	...	-0.201843	0.253157
558	2013-12-19 23:00:00	0.038321	0.221321	0.175321	0.015321	-0.219679	0.336321	-0.008679	-0.068679	0.069321	...	-0.106679	0.045321

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024
559	2013-12-20 00:00:00	-0.066093	0.060907	0.239907	0.091907	-0.137093	-0.006093	0.028907	-0.039093	0.425907	...	-0.015093	0.045907
560	2013-12-20 01:00:00	-0.033146	-0.032146	0.219854	0.045854	-0.099146	-0.005146	-0.027146	-0.055146	-0.037146	...	-0.025146	0.238854
561	2013-12-20 02:00:00	-0.009884	-0.026884	0.161116	0.081116	-0.094884	0.003116	-0.001884	-0.035884	-0.040884	...	-0.004884	0.068116
562	2013-12-20 03:00:00	0.015113	-0.019887	0.113113	0.117113	-0.083887	0.031113	0.000113	-0.021887	-0.055887	...	-0.057887	0.148113
563	2013-12-20 04:00:00	0.265246	-0.000754	0.057246	0.038246	-0.074754	-0.003754	0.004246	-0.030754	-0.027754	...	-0.044754	0.118246
564	2013-12-26 08:00:00	-0.129868	0.021132	-0.063868	0.115132	-0.209868	-0.073868	-0.155868	-0.153868	-0.119868	...	-0.212868	-0.087868
565	2013-12-26 09:00:00	0.062386	0.095386	-0.053614	0.088386	-0.201614	-0.097614	0.380386	-0.095614	-0.076614	...	-0.205614	0.045386
566	2013-12-26 10:00:00	-0.048643	0.171357	-0.028643	0.133357	-0.179643	-0.075643	0.132357	0.090357	0.106357	...	-0.129643	-0.044643
567	2013-12-29 17:00:00	0.645779	0.157779	0.125779	0.145779	-0.280221	-0.162221	0.079779	-0.216221	0.194779	...	-0.243221	0.241779
568	2013-12-29 18:00:00	0.352694	0.023694	0.015694	0.179694	-0.303306	-0.218306	0.784694	-0.249306	-0.032306	...	-0.229306	-0.034306
569	2013-12-29 19:00:00	0.325393	-0.082607	-0.136607	0.838393	-0.335607	-0.118607	0.413393	-0.270607	0.004393	...	-0.246607	0.113393
570	2013-12-29 20:00:00	0.351702	-0.025298	-0.065298	0.079702	-0.318298	-0.127298	-0.034298	-0.253298	-0.050298	...	-0.143298	-0.055298
571	2013-12-29 21:00:00	0.307417	0.015417	-0.097583	0.050417	-0.295583	-0.196583	0.025417	-0.070583	0.333417	...	-0.135583	-0.015583
572	2013-12-29 22:00:00	0.237259	0.018259	-0.098741	0.070259	-0.257741	-0.163741	0.038259	0.014259	0.217259	...	-0.136741	-0.006741
573	2013-12-29 23:00:00	0.217968	0.090968	-0.026032	0.027968	-0.204032	-0.112032	-0.050032	0.229968	0.651968	...	-0.076032	0.263968
574	2013-12-30 00:00:00	0.134981	0.202981	0.021981	0.129981	-0.128019	-0.058019	-0.031019	0.021981	1.093981	...	-0.081019	0.032981
575	2013-12-30 01:00:00	0.168191	-0.017809	0.056191	0.102191	-0.104809	-0.015809	0.036191	0.033191	0.490191	...	-0.052809	0.086191
576	2013-12-30 02:00:00	0.032048	-0.023952	0.249048	0.038048	-0.090952	-0.008952	0.032048	-0.029952	-0.034952	...	-0.078952	0.091048
577	2013-12-30 03:00:00	0.006540	-0.039460	0.129540	0.036540	-0.085460	-0.017460	0.016540	-0.023460	-0.004460	...	-0.074460	0.098540
578	2013-12-30 04:00:00	0.439431	-0.046569	0.085431	0.036431	-0.084569	0.017431	0.000431	-0.020569	-0.007569	...	-0.055569	0.048431

579 rows × 416 columns

In [19]:
```
# select each user ID and form a seperate dataframe
# first change all the user ID to "Consumption"
new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[0]):
    new_col.append('Consumption')
new_col = new_col + list(df_type0_res.columns)[-8:] # the new column names
df_type0_res.columns = new_col
df_all_res_long = pd.DataFrame()
for i in range(1, houseGroupDf.counts.iloc[0] + 1):
    df_all_res_long = df_all_res_long.append(df_type0_res.iloc[:,[0,i,-8,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[1]): # for renaming the second type
    new_col.append('Consumption')
new_col = new_col + list(df_type1_res.columns)[-8:] # the new column names
df_type1_res.columns = new_col
for i in range(1, houseGroupDf.counts.iloc[1] + 1):
    df_all_res_long = df_all_res_long.append(df_type1_res.iloc[:,[0,i,-8,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[2]): # for renaming the third type
    new_col.append('Consumption')
new_col = new_col + list(df_type2_res.columns)[-8:] # the new column names
df_type2_res.columns = new_col
for i in range(1, houseGroupDf.counts.iloc[2] + 1):
    df_all_res_long = df_all_res_long.append(df_type2_res.iloc[:,[0,i,-8,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

# for clearer visulization, round the temperatures
df_all_res_long['TempC'] = df_all_res_long.TempC.round(0)
df_all_res_long['TempF'] = df_all_res_long.TempF.round(0)
df_all_res_long # ready for the price responsiveness analysis
```

Out[19]:

	GMT	Consumption	TempC	TempF	Price	Event_tags	Day of week	Hour of day	Household type	Predicted consumption
0	2013-01-04 14:00:00	-0.084761	10.0	51.0	0.0399	L3	4	14	0	0.186761
1	2013-01-04 15:00:00	-0.095933	10.0	51.0	0.0399	L3	4	15	0	0.192933
2	2013-01-04 16:00:00	1.132319	9.0	49.0	0.0399	L3	4	16	0	0.228681
3	2013-01-07 23:00:00	0.092587	7.0	44.0	0.6720	H3	0	23	0	0.221413
4	2013-01-08 00:00:00	-0.066445	7.0	44.0	0.6720	H3	1	0	0	0.164445
5	2013-01-08 01:00:00	-0.033424	7.0	43.0	0.6720	H3	1	1	0	0.132424
6	2013-01-10 02:00:00	-0.014933	1.0	34.0	0.0399	L3	3	2	0	0.111933
7	2013-01-10 03:00:00	-0.015324	1.0	34.0	0.0399	L3	3	3	0	0.110324
8	2013-01-10 04:00:00	0.113252	1.0	34.0	0.0399	L3	3	4	0	0.106748
9	2013-01-11 11:00:00	-0.100935	3.0	37.0	0.6720	H3	4	11	0	0.205935
10	2013-01-11 12:00:00	-0.144464	4.0	39.0	0.6720	H3	4	12	0	0.204464
11	2013-01-11 13:00:00	-0.103057	4.0	39.0	0.6720	H3	4	13	0	0.206057
12	2013-01-13 05:00:00	0.131004	-2.0	29.0	0.6720	H6	6	5	0	0.105996
13	2013-01-13 06:00:00	-0.026613	-2.0	29.0	0.6720	H6	6	6	0	0.116613
14	2013-01-13 07:00:00	-0.080359	-1.0	30.0	0.6720	H6	6	7	0	0.148359
15	2013-01-13 08:00:00	-0.112905	-1.0	31.0	0.6720	H6	6	8	0	0.212905
16	2013-01-13 09:00:00	-0.146042	0.0	32.0	0.6720	H6	6	9	0	0.246042
17	2013-01-13 10:00:00	-0.185176	1.0	33.0	0.6720	H6	6	10	0	0.257176
18	2013-01-16 23:00:00	-0.172620	-5.0	22.0	0.6720	H3	2	23	0	0.273620
19	2013-01-17 00:00:00	-0.117406	-6.0	21.0	0.6720	H3	3	0	0	0.205406
20	2013-01-17 01:00:00	-0.071150	-5.0	22.0	0.6720	H3	3	1	0	0.147150
21	2013-01-19 05:00:00	-0.027990	-2.0	29.0	0.0399	CM	5	5	0	0.107990
22	2013-01-19 06:00:00	-0.024402	-2.0	29.0	0.0399	CM	5	6	0	0.119402
23	2013-01-19 07:00:00	-0.060056	-2.0	30.0	0.0399	CM	5	7	0	0.154056
24	2013-01-19 08:00:00	-0.147660	-1.0	30.0	0.0399	CM	5	8	0	0.216660
25	2013-01-19 09:00:00	-0.115253	-1.0	31.0	0.0399	CM	5	9	0	0.245253
26	2013-01-19 10:00:00	0.688811	-1.0	31.0	0.0399	CM	5	10	0	0.268189
27	2013-01-19 11:00:00	0.325001	-0.0	32.0	0.0399	CM	5	11	0	0.273999
28	2013-01-19 12:00:00	0.074385	0.0	32.0	0.0399	CM	5	12	0	0.268615

	GMT	Consumption	TempC	TempF	Price	Event_tags	Day of week	Hour of day	Household type	Predicted consumption
29	2013-01-19 13:00:00	-0.159550	0.0	32.0	0.0399	CM	5	13	0	0.268550
...	...	...	...	...	...	...	...	...	...	...
258783	2013-12-19 14:00:00	0.026042	5.0	42.0	0.0399	L24	3	14	2	0.212958
258784	2013-12-19 15:00:00	0.752225	5.0	42.0	0.0399	L24	3	15	2	0.274775
258785	2013-12-19 16:00:00	1.037395	5.0	42.0	0.0399	L24	3	16	2	0.309605
258786	2013-12-19 17:00:00	1.230750	5.0	42.0	0.0399	L24	3	17	2	0.883250
258787	2013-12-19 18:00:00	0.048408	5.0	42.0	0.0399	L24	3	18	2	0.516592
258788	2013-12-19 19:00:00	-0.138476	5.0	42.0	0.0399	L24	3	19	2	0.380476
258789	2013-12-19 20:00:00	-0.088367	5.0	42.0	0.0399	L24	3	20	2	0.364367
258790	2013-12-19 21:00:00	-0.029095	5.0	42.0	0.0399	L24	3	21	2	0.324095
258791	2013-12-19 22:00:00	-0.043453	5.0	41.0	0.0399	L24	3	22	2	0.299453
258792	2013-12-19 23:00:00	0.079350	5.0	41.0	0.0399	L24	3	23	2	0.191650
258793	2013-12-20 00:00:00	0.057260	5.0	40.0	0.0399	L24	4	0	2	0.125740
258794	2013-12-20 01:00:00	-0.011436	5.0	40.0	0.0399	L24	4	1	2	0.101436
258795	2013-12-20 02:00:00	0.017666	5.0	40.0	0.0399	L24	4	2	2	0.097334
258796	2013-12-20 03:00:00	0.057762	5.0	40.0	0.0399	L24	4	3	2	0.103238
258797	2013-12-20 04:00:00	-0.012526	5.0	40.0	0.0399	L24	4	4	2	0.101526
258798	2013-12-26 08:00:00	-0.061602	4.0	40.0	0.0399	L3	3	8	2	0.291602
258799	2013-12-26 09:00:00	-0.096392	4.0	40.0	0.0399	L3	3	9	2	0.253392
258800	2013-12-26 10:00:00	-0.150750	4.0	40.0	0.0399	L3	3	10	2	0.235750
258801	2013-12-29 17:00:00	-0.255094	3.0	38.0	0.0399	L12	6	17	2	0.598094
258802	2013-12-29 18:00:00	-0.164974	3.0	38.0	0.0399	L12	6	18	2	0.452974
258803	2013-12-29 19:00:00	-0.069955	3.0	38.0	0.0399	L12	6	19	2	0.342955
258804	2013-12-29 20:00:00	NaN	3.0	38.0	0.0399	L12	6	20	2	0.343809
258805	2013-12-29 21:00:00	-0.022450	3.0	38.0	0.0399	L12	6	21	2	0.312450
258806	2013-12-29 22:00:00	0.035618	5.0	41.0	0.0399	L12	6	22	2	0.275382
258807	2013-12-29 23:00:00	0.033214	6.0	43.0	0.0399	L12	6	23	2	0.174786
258808	2013-12-30 00:00:00	0.016563	8.0	46.0	0.0399	L12	0	0	2	0.115437
258809	2013-12-30 01:00:00	0.036195	8.0	46.0	0.0399	L12	0	1	2	0.101805
258810	2013-12-30 02:00:00	-0.015848	8.0	46.0	0.0399	L12	0	2	2	0.103848

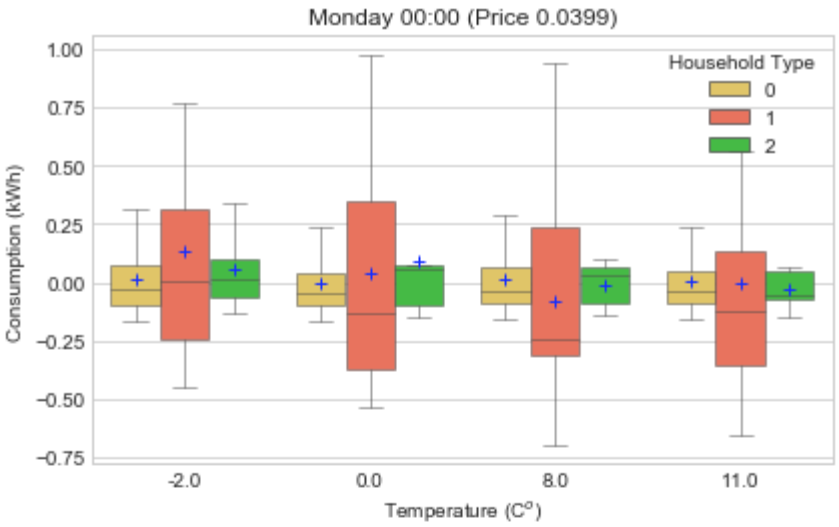
	GMT	Consumption	TempC	TempF	Price	Event_tags	Day of week	Hour of day	Household type	Predicted consumption
258811	2013-12-30 03:00:00	0.014469	8.0	46.0	0.0399	L12	0	3	2	0.101531
258812	2013-12-30 04:00:00	0.049480	8.0	46.0	0.0399	L12	0	4	2	0.103520

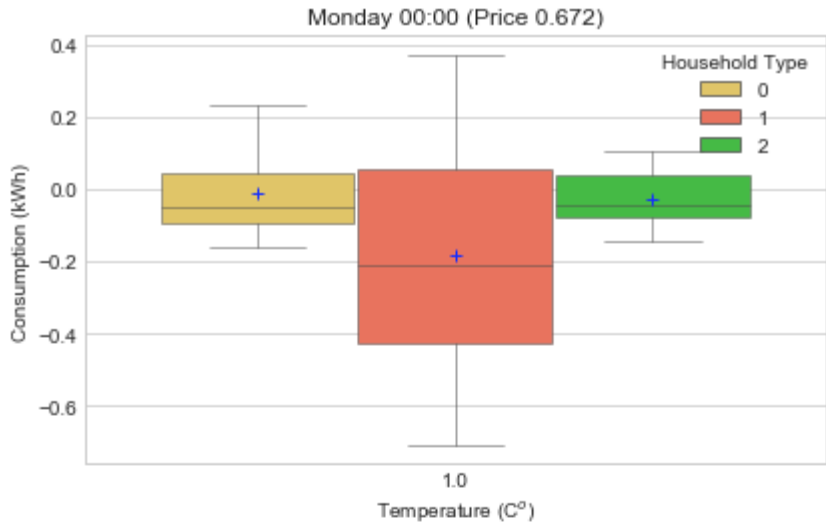
258813 rows × 10 columns

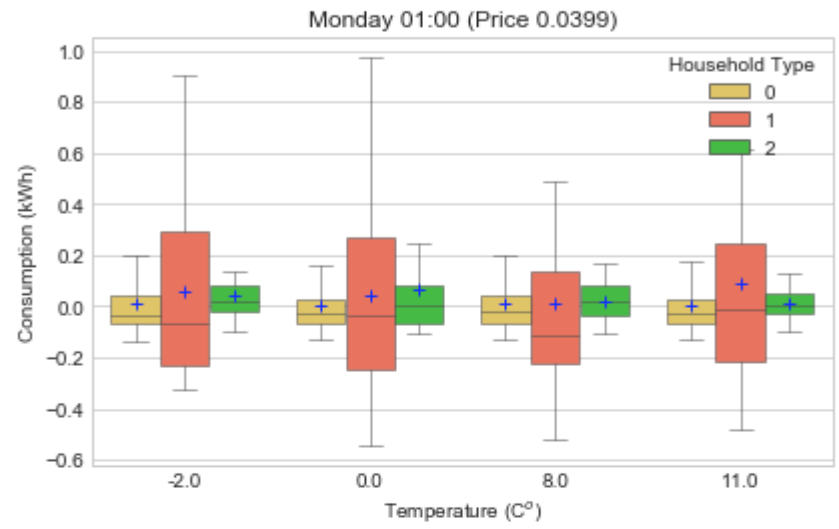
Then plot the properties based on from the smallest unit (day of week, hour, temp) to (day of week, hour), (day of week, temp), (hour, temp), (day of week), (hour), (temp) for different prices (high and low) so we will have very comprehensive price responsiveness of three different groups

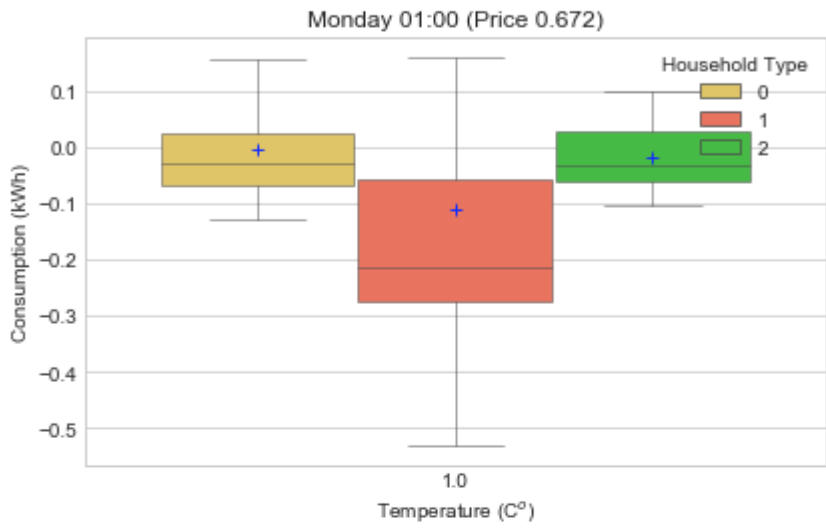
a) Day of week - hour - temperature : prices

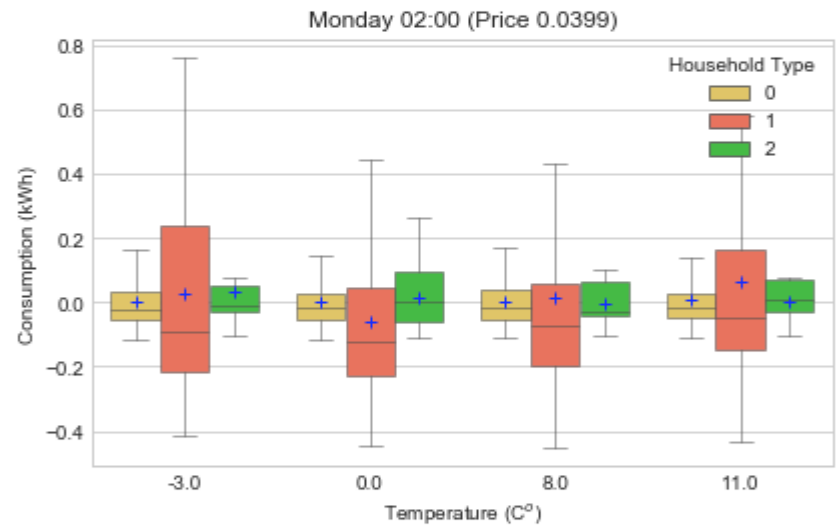
In [97]:
```
# Monday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 0 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

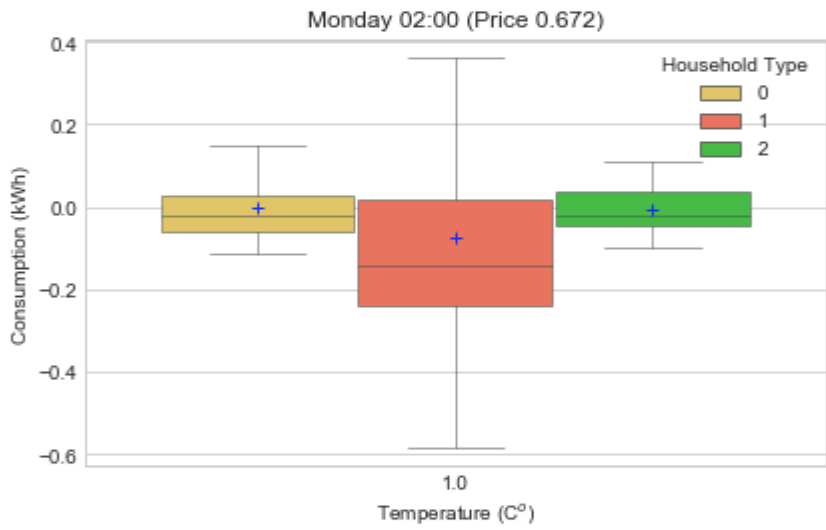

Monday 00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-2.0
0.0
8.0
11.0
Temperature (C°)

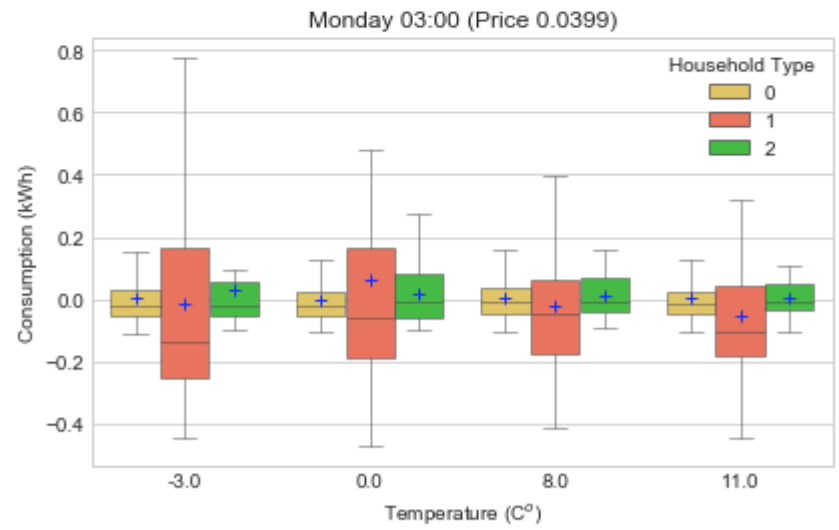

Monday 00:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
1.0
Temperature (C°)

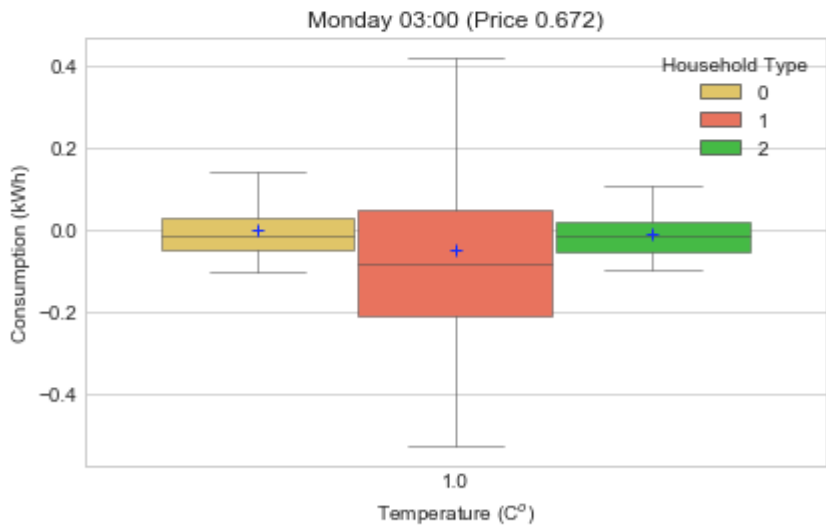

Monday 01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0
0.0
8.0
11.0
Temperature (C°)

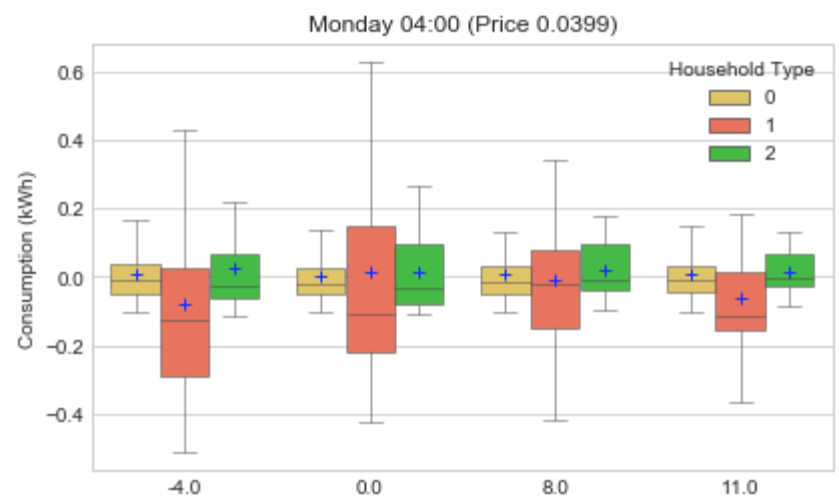

Monday 01:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
1.0
Temperature (C°)

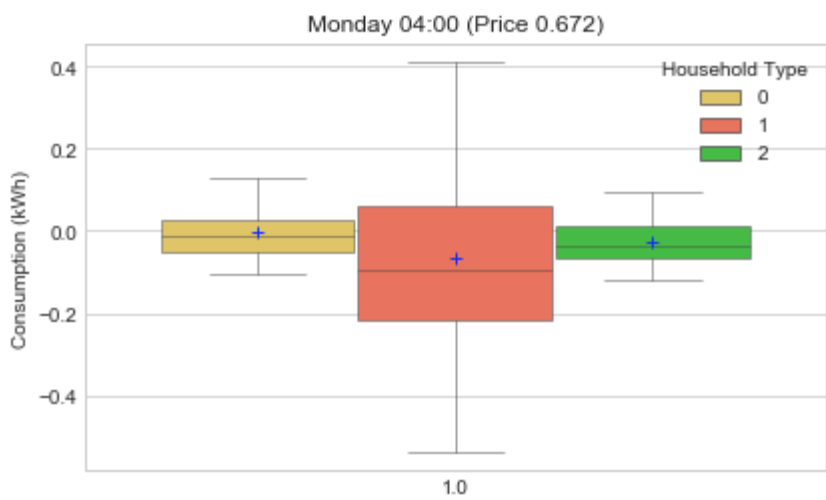

Monday 02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-3.0
0.0
8.0
11.0
Temperature (C°)


Monday 02:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
1.0
Temperature (C°)


Monday 03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-3.0
0.0
8.0
11.0
Temperature (C°)


Monday 03:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
1.0
Temperature (C°)

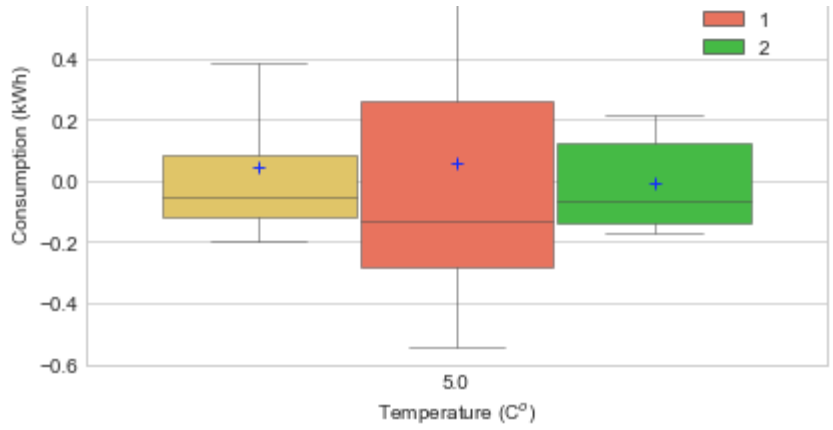

Monday 04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-4.0
0.0
8.0
11.0
Temperature (C°)

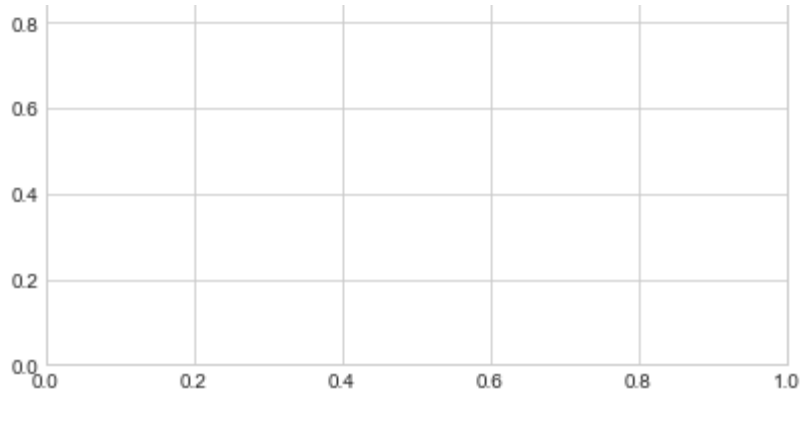

Monday 04:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
1.0
Temperature (C°)

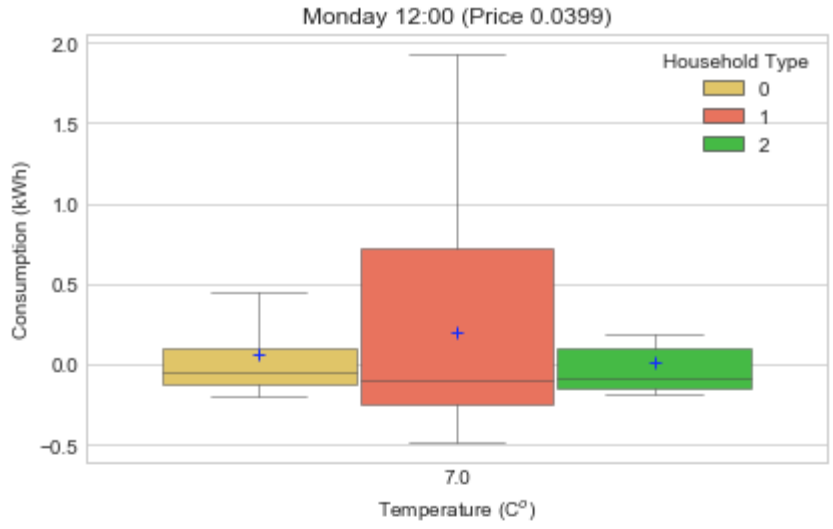
1.0
0.8
0.6
0.4

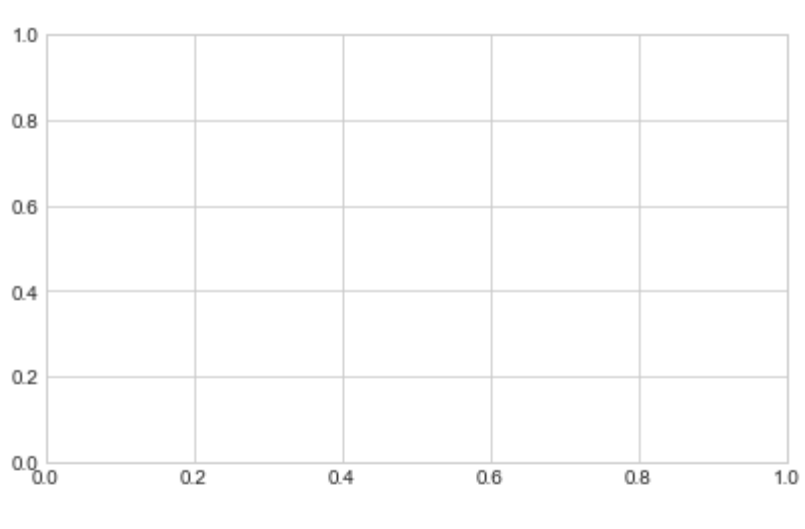
1.0
0.8
0.6
0.4

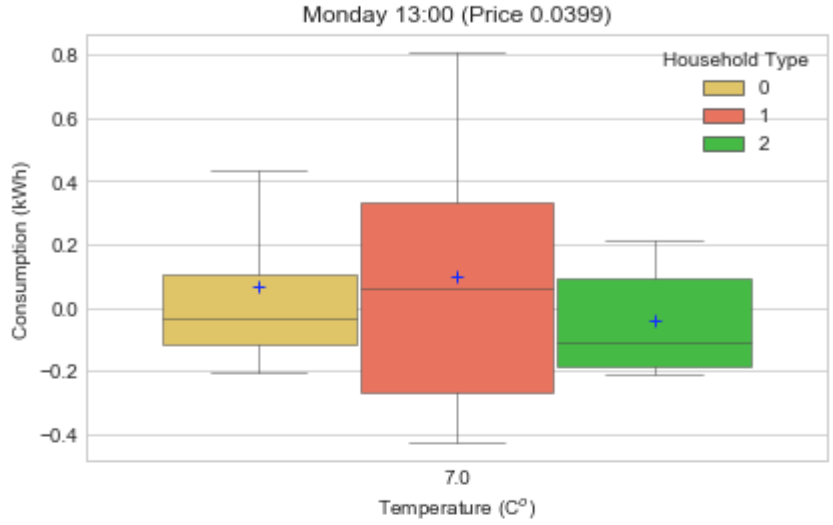

Monday 11:00 (Price 0.0399)
Household Type
0

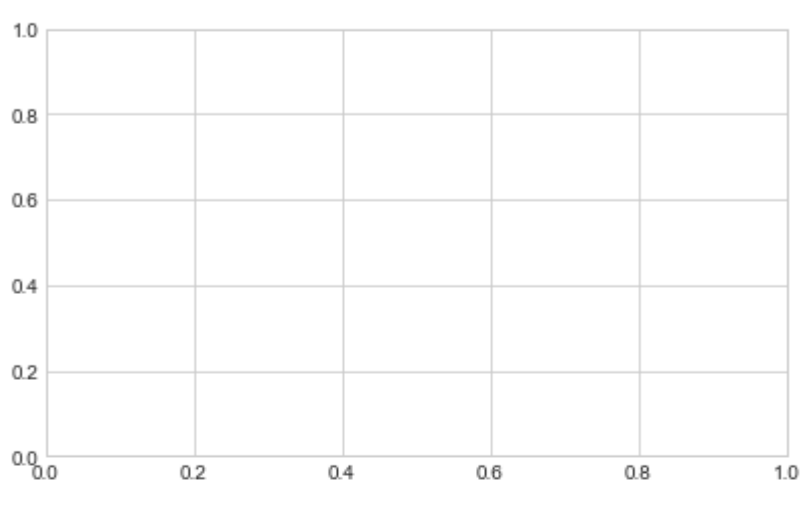

1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
5.0
Temperature (C°)

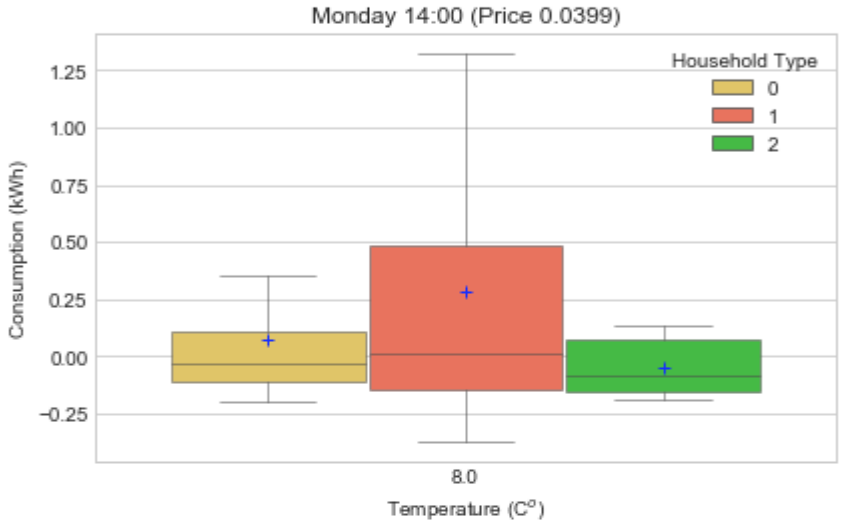

0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

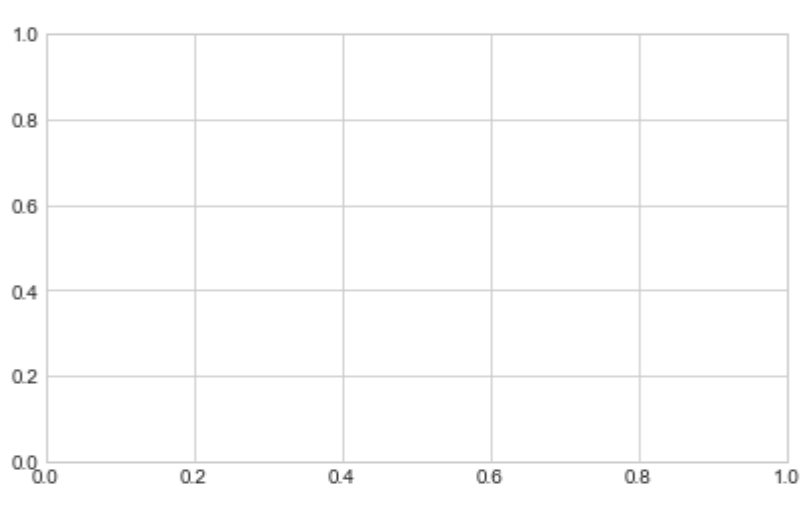

Monday 12:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
7.0
Temperature (C°)

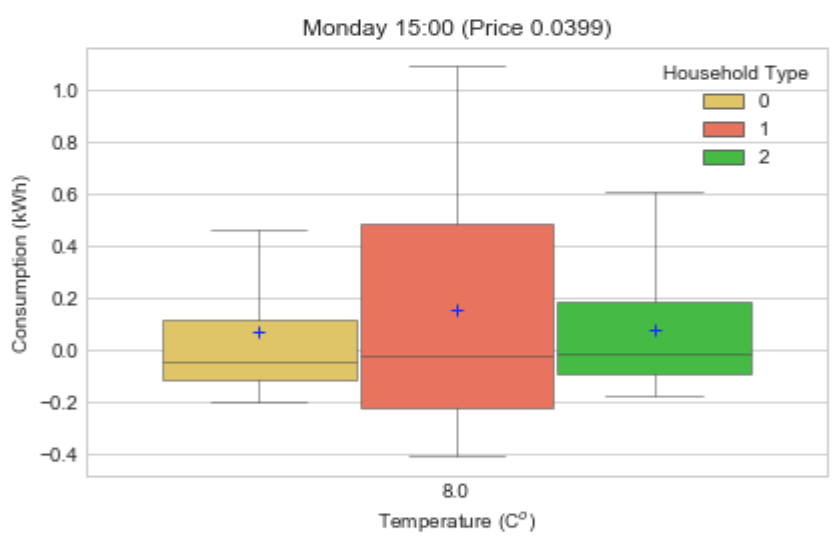

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

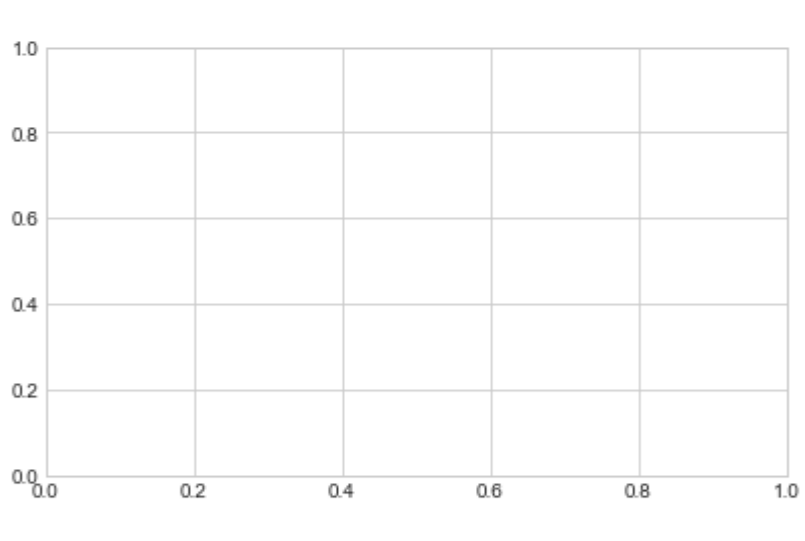

Monday 13:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
Temperature (C°)

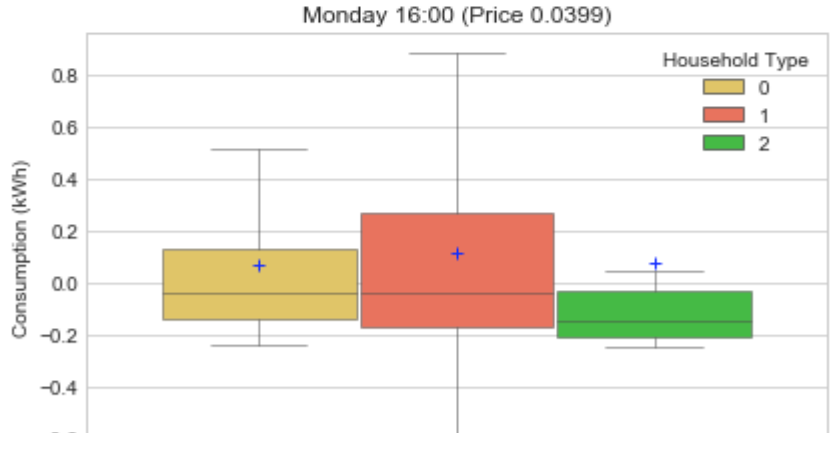

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

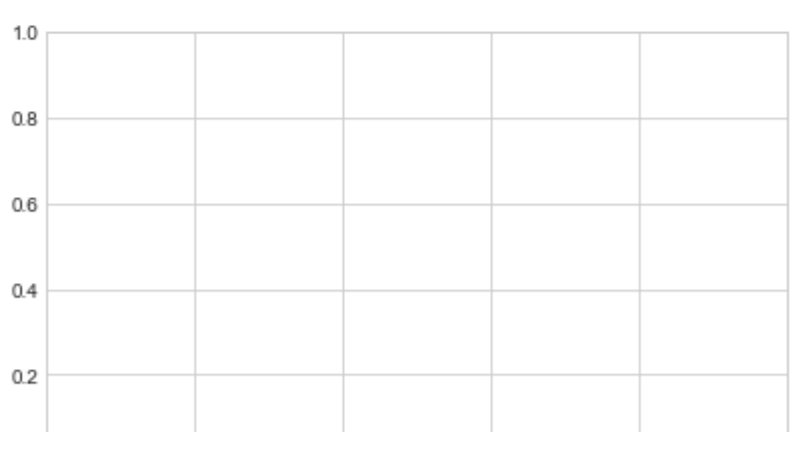

Monday 14:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
8.0
Temperature (C°)

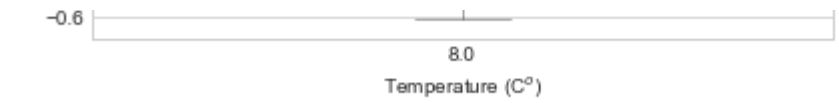

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Monday 15:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
8.0
Temperature (C°)

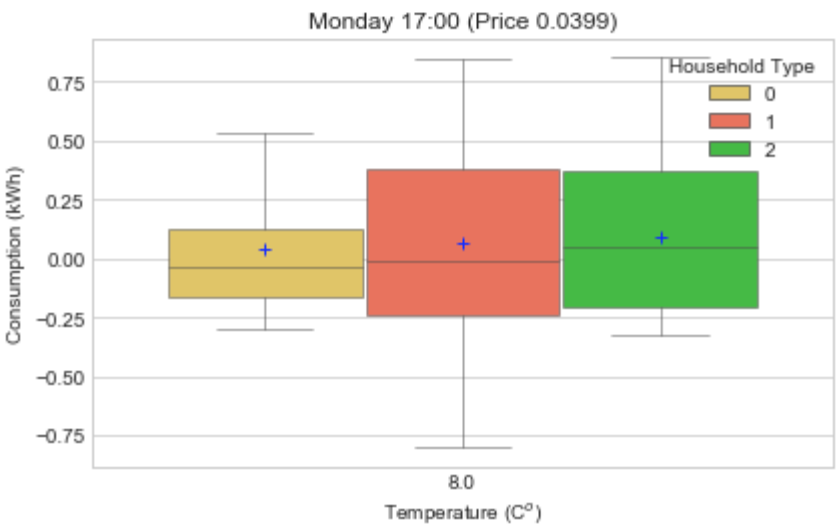

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

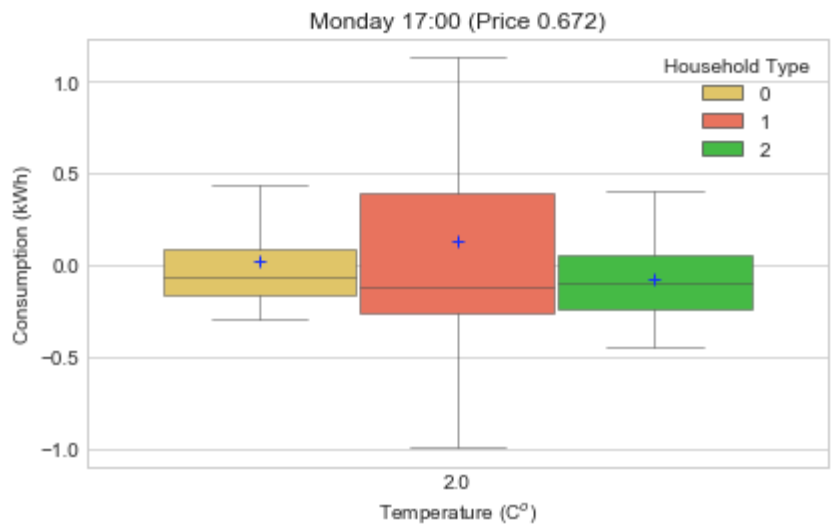

Monday 16:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

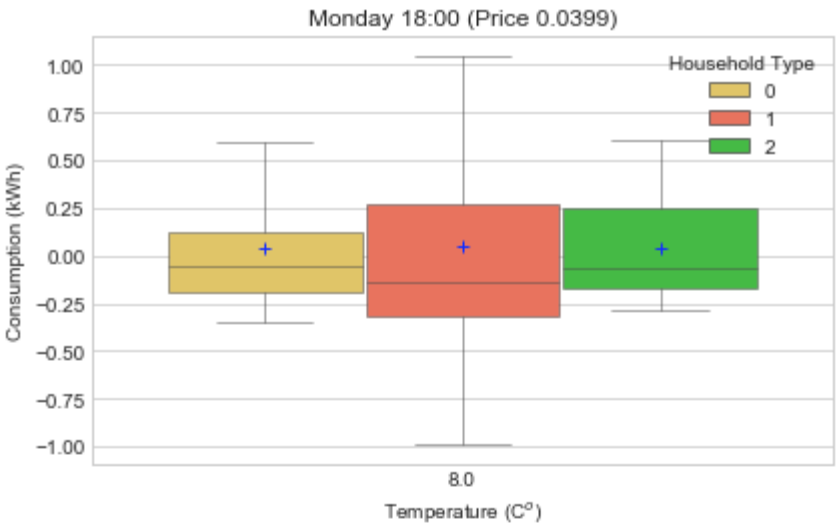

1.0
0.8
0.6
0.4
0.2

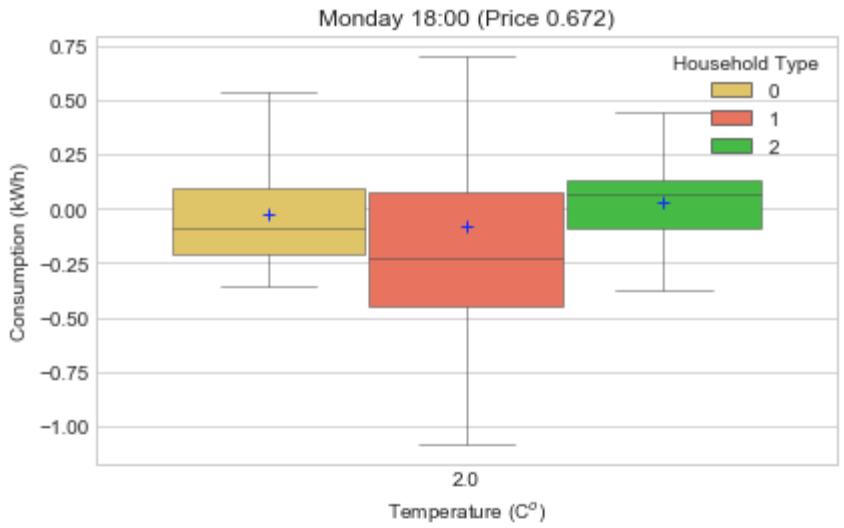

−0.6
8.0
Temperature (C°)

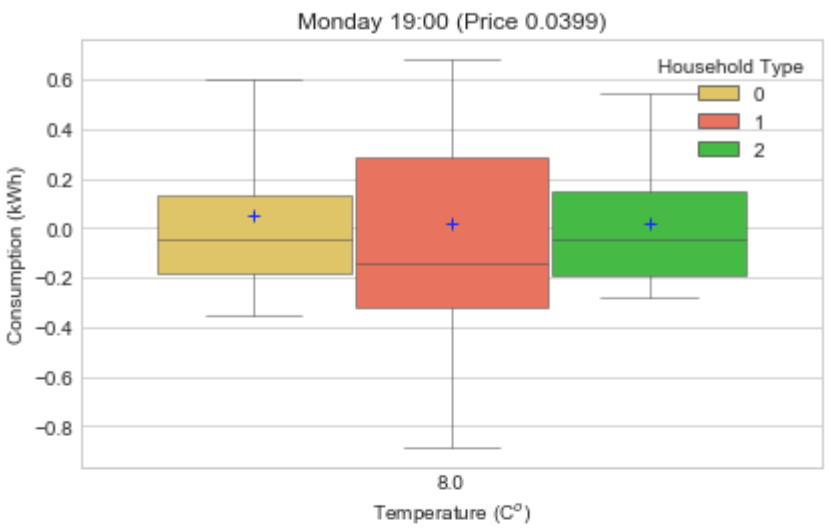

0.0
0.0
0.2
0.4
0.6
0.8
1.0

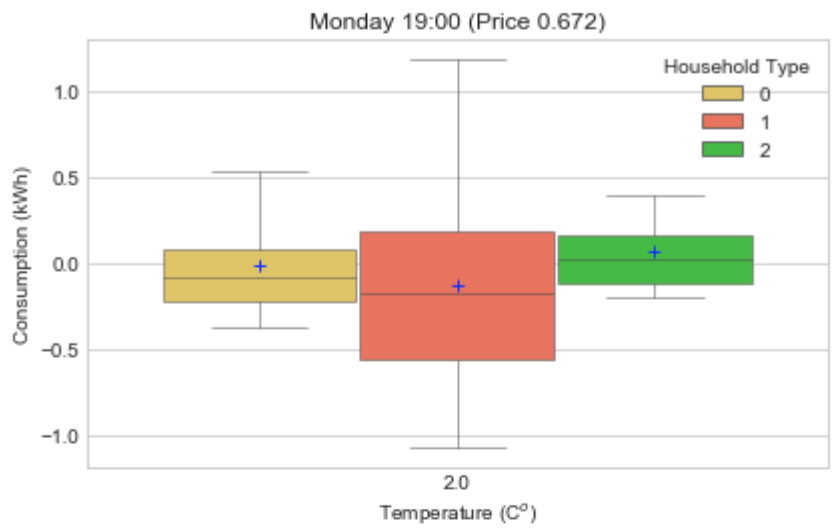

Monday 17:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
8.0
Temperature (C°)

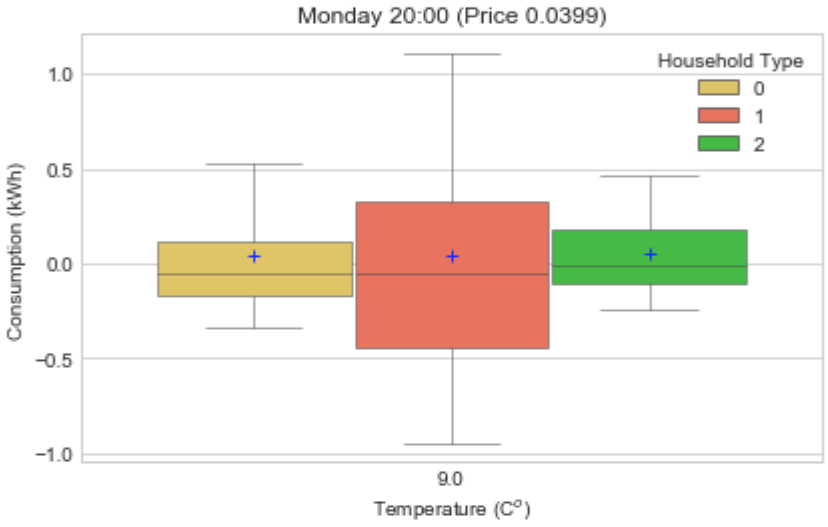

Monday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
Temperature (C°)

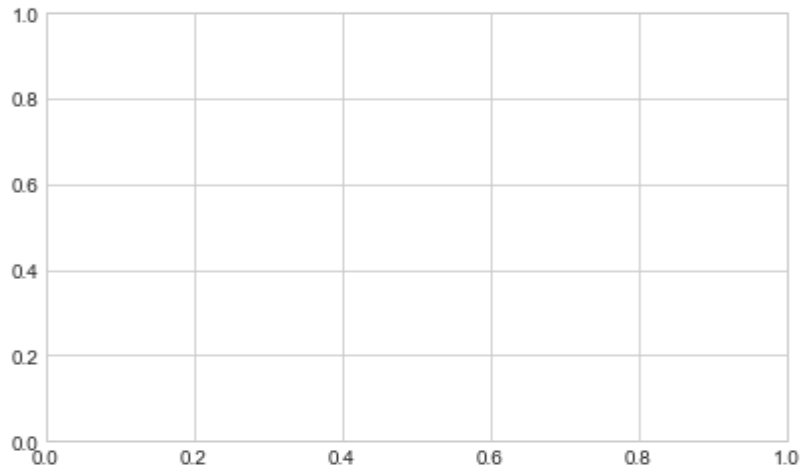

Monday 18:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
8.0
Temperature (C°)

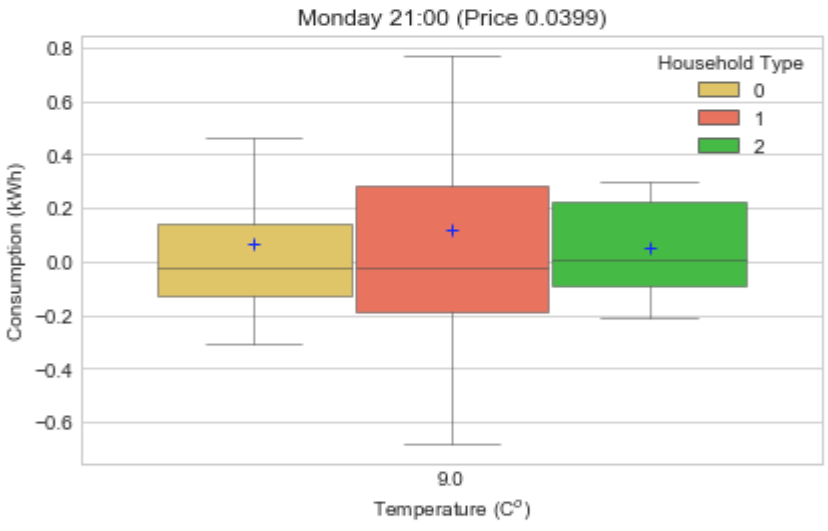

Monday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2.0
Temperature (C°)

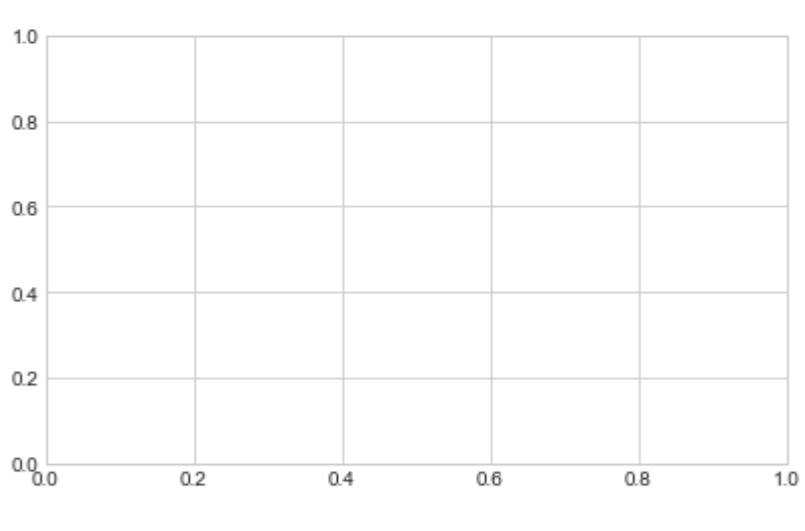

Monday 19:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
8.0
Temperature (C°)


Monday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
Temperature (C°)


Monday 20:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
9.0
Temperature (C°)

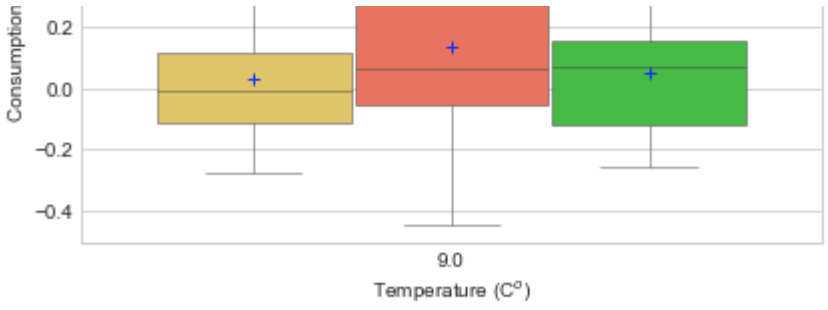

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

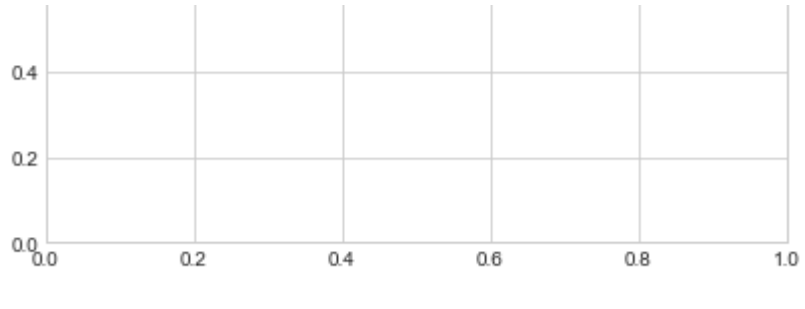

Monday 21:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
9.0
Temperature (C°)

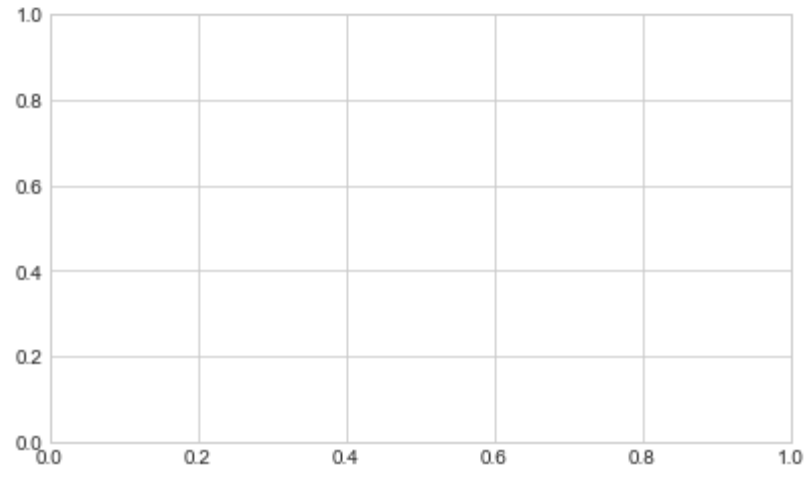

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

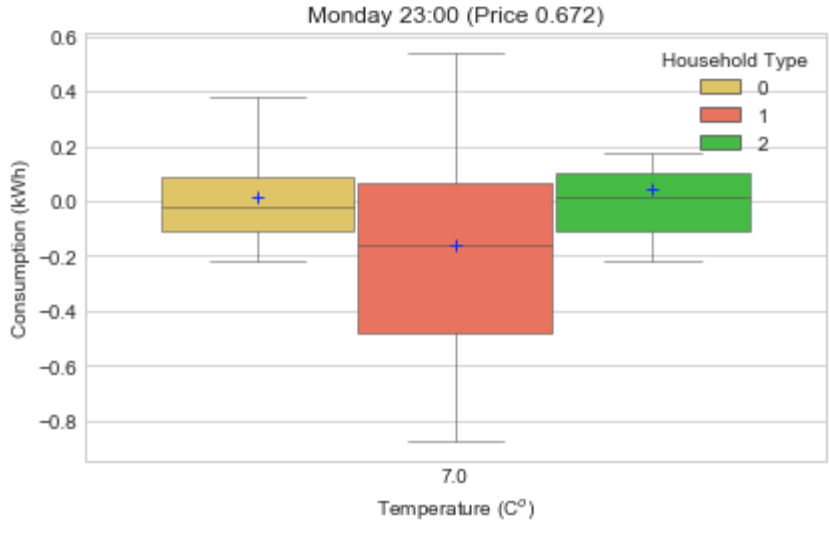
Monday 22:00 (Price 0.0399)
Household Type
0
1
2
0.8
0.6
0.4

1.0
0.8
0.6

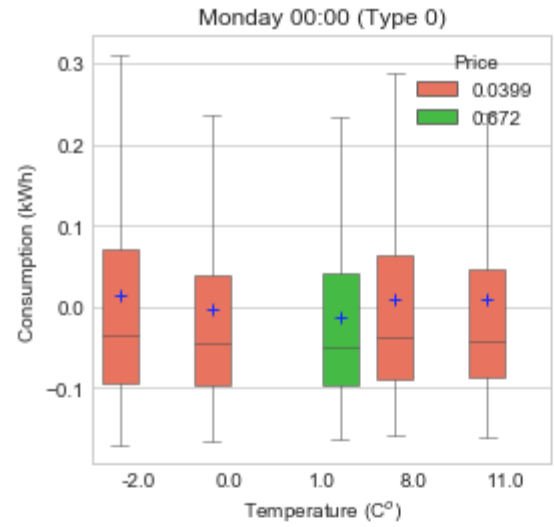

Consumption
0.2
0.0
−0.2
−0.4
9.0
Temperature (C°)

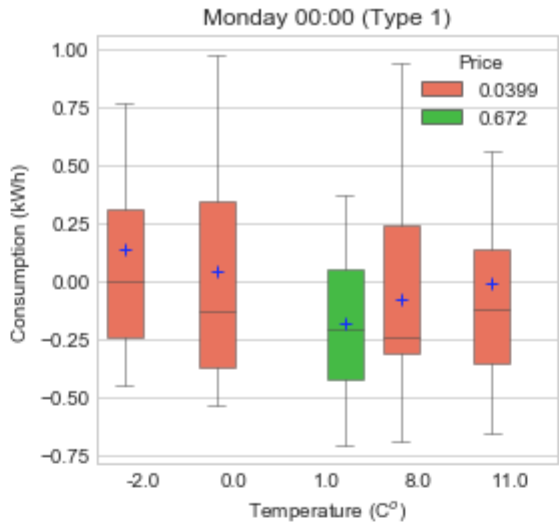

0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

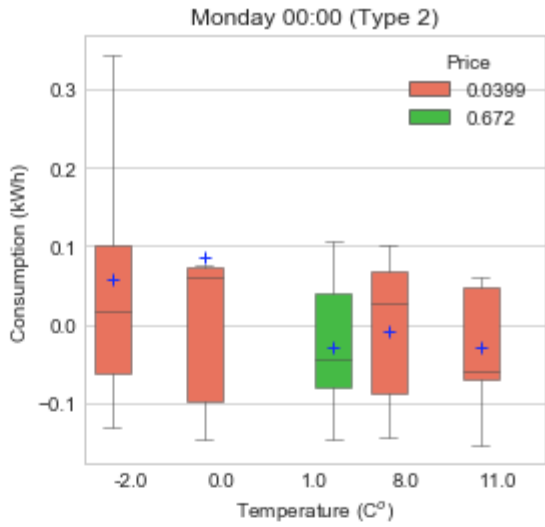

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

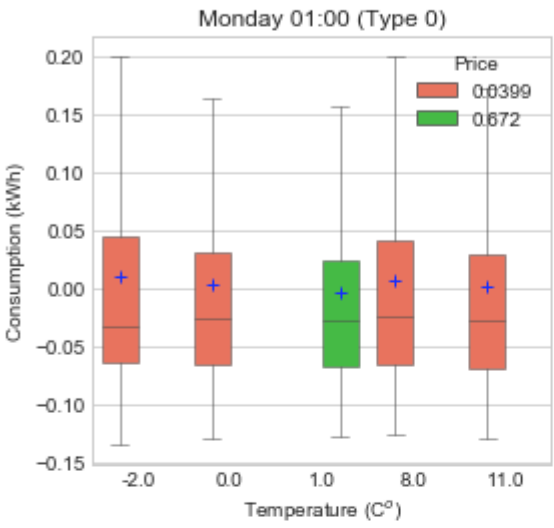

Monday 23:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
7.0
Temperature (C°)

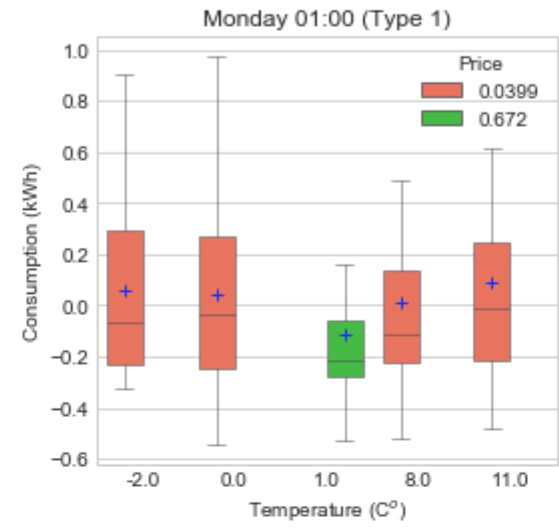
In [128]:
```
# Price comparison version
# Monday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 0 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3 * i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

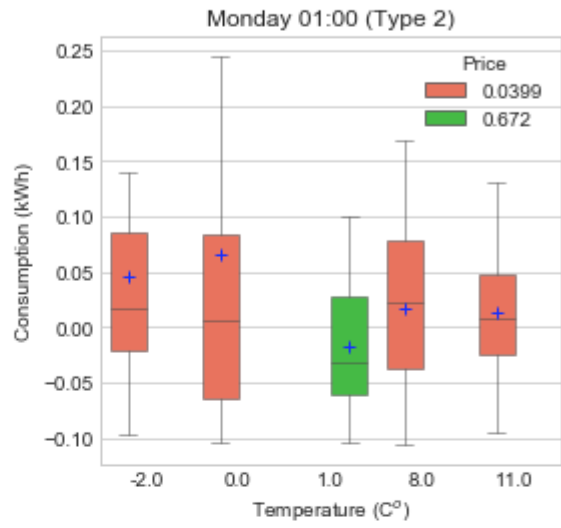

Monday 00:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

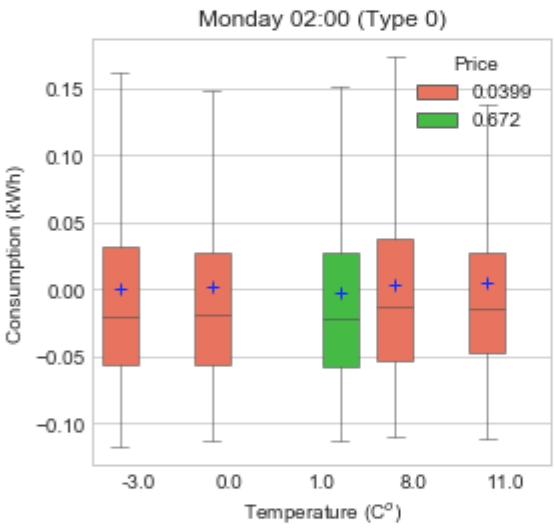

Monday 00:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

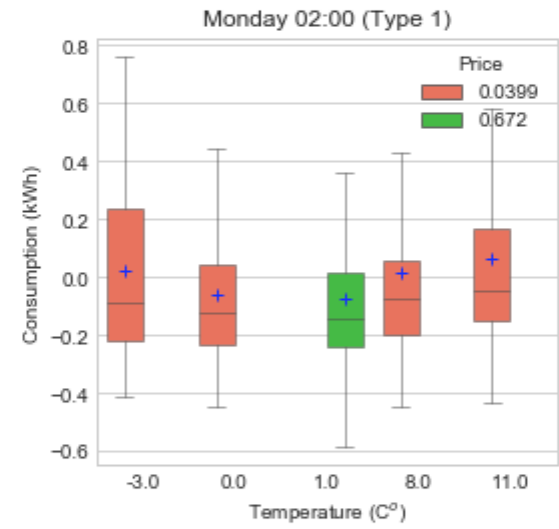

Monday 00:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

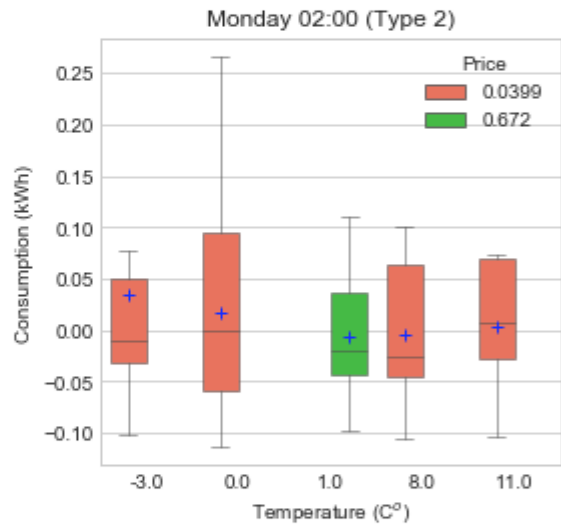

Monday 01:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

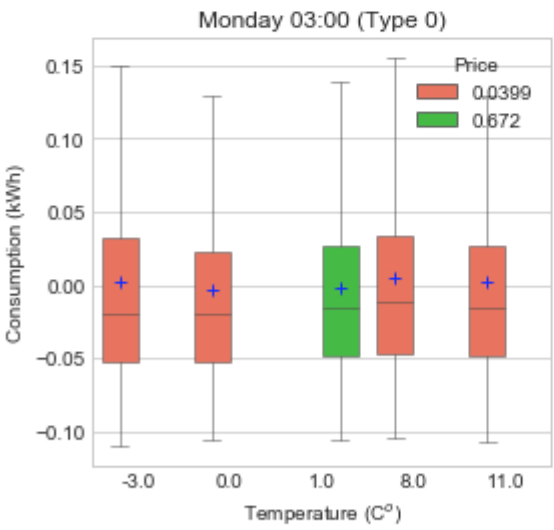

Monday 01:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

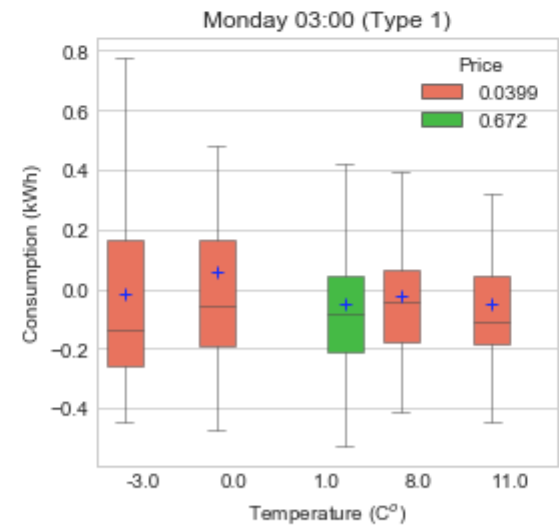

Monday 01:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

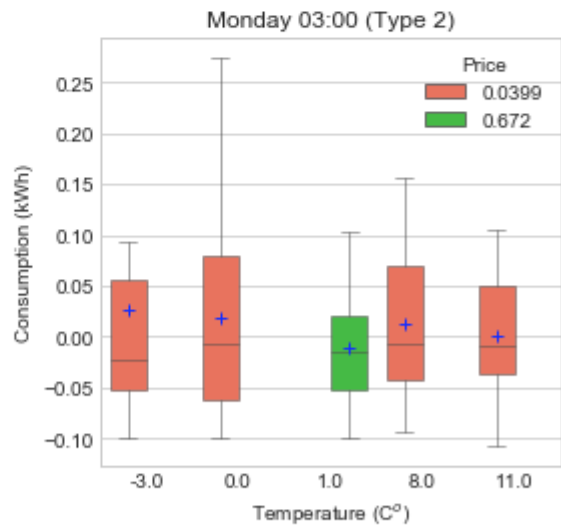

Monday 02:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

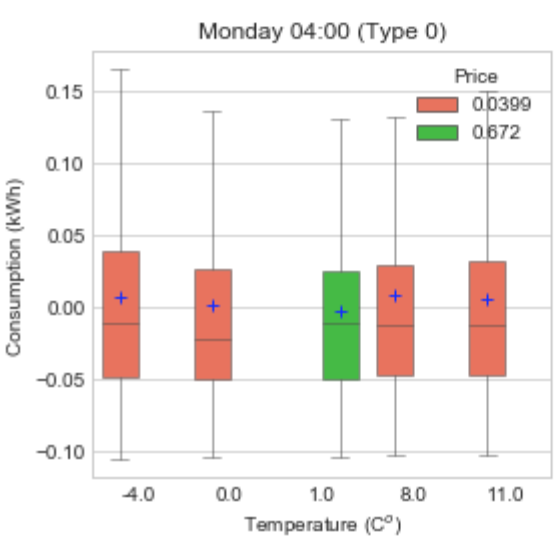

Monday 02:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

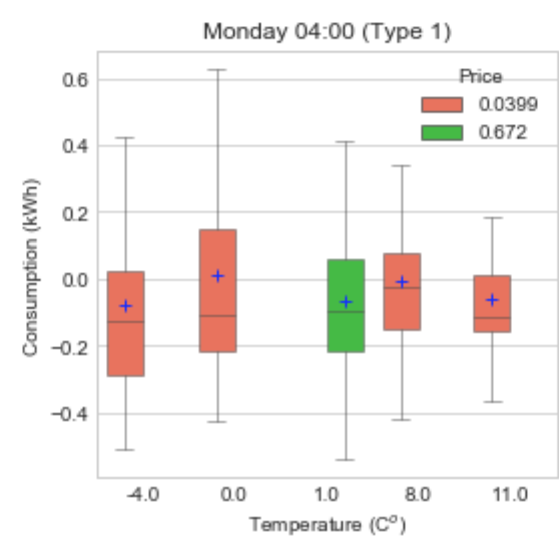

Monday 02:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

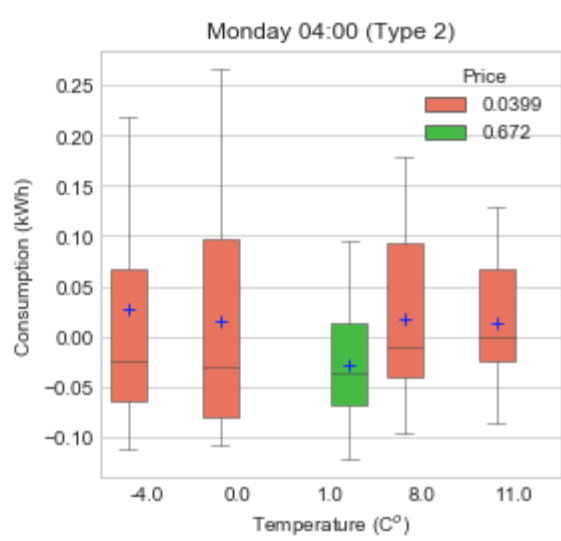

Monday 03:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

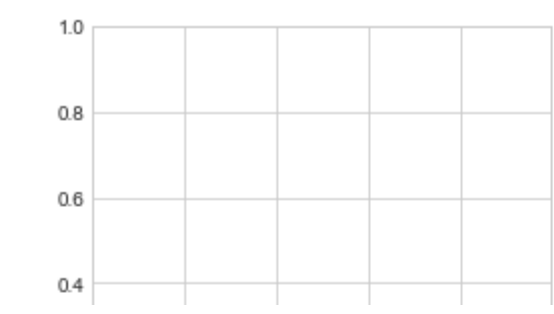

Monday 03:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

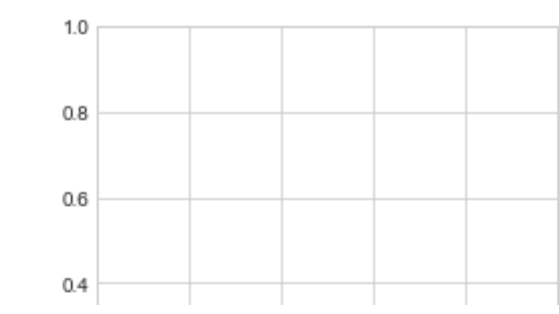

Monday 03:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)


Monday 04:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)


Monday 04:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

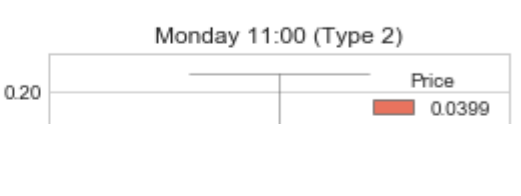

Monday 04:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

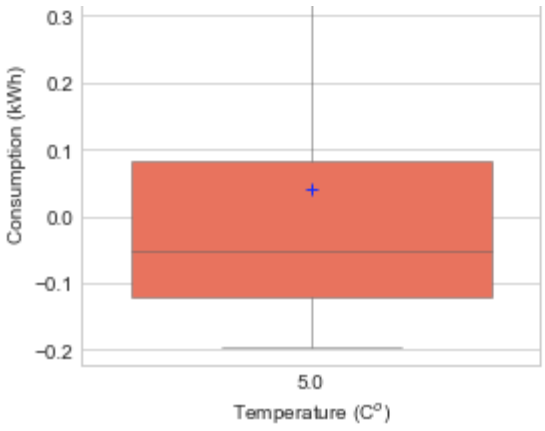


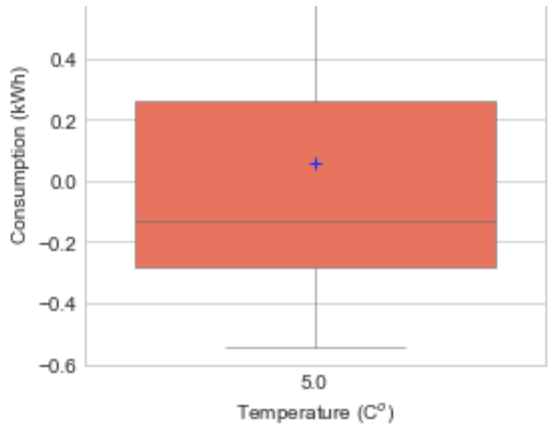


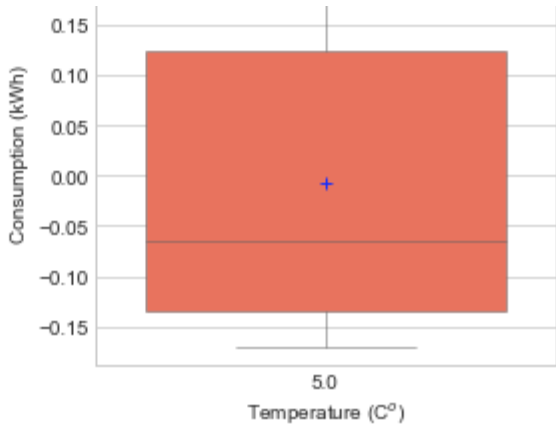

Monday 11:00 (Type 0)
Price
0.0399

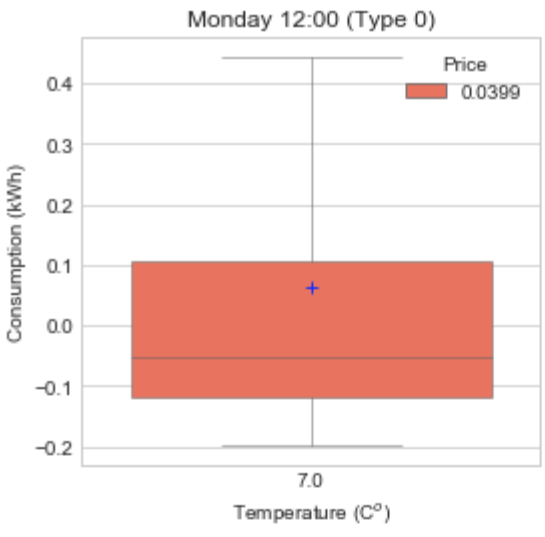

Monday 11:00 (Type 1)
Price
0.0399

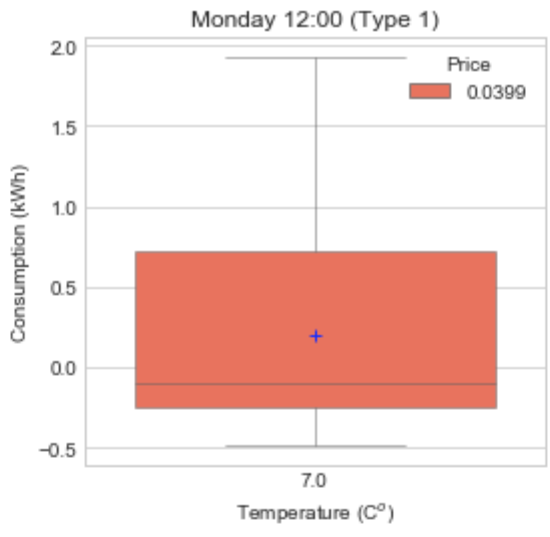

Monday 11:00 (Type 2)
Price
0.0399

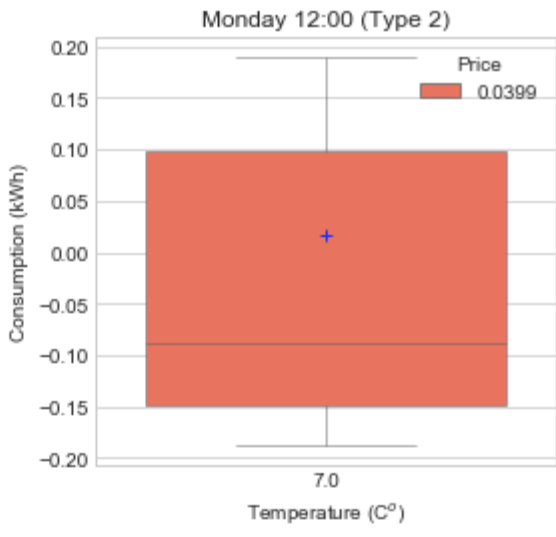

Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
5.0
Temperature (C°)

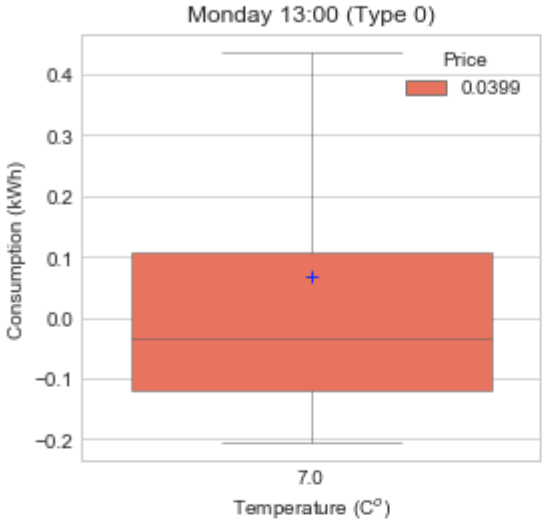

Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
5.0
Temperature (C°)

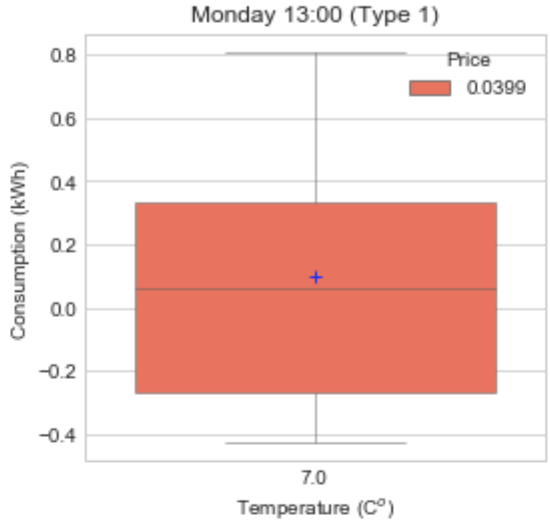

Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
5.0
Temperature (C°)

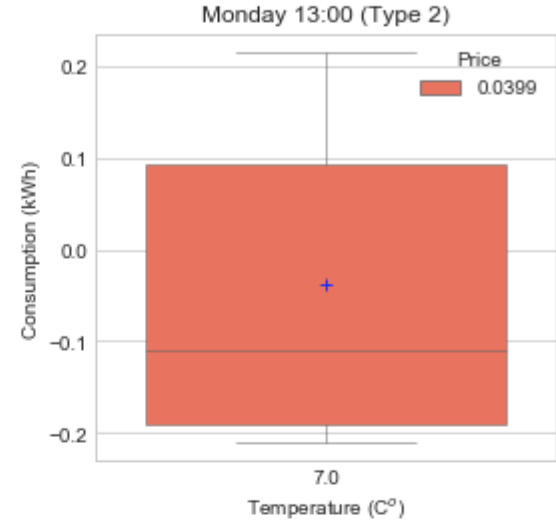

Monday 12:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
7.0
Temperature (C°)

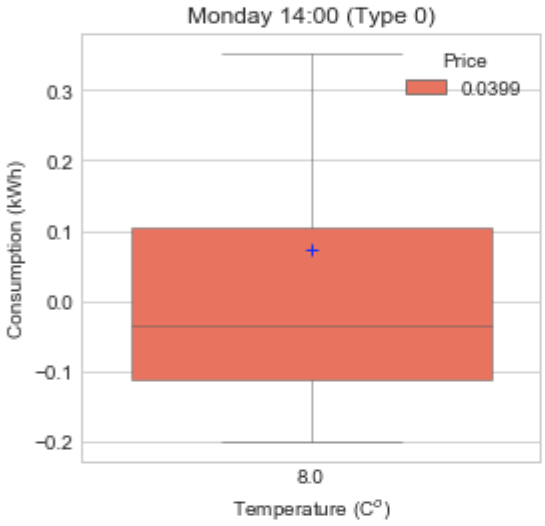

Monday 12:00 (Type 1)
Price
0.0399
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
7.0
Temperature (C°)

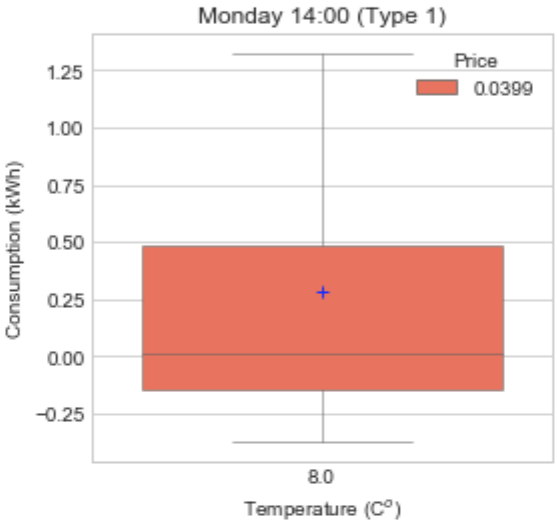

Monday 12:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
7.0
Temperature (C°)

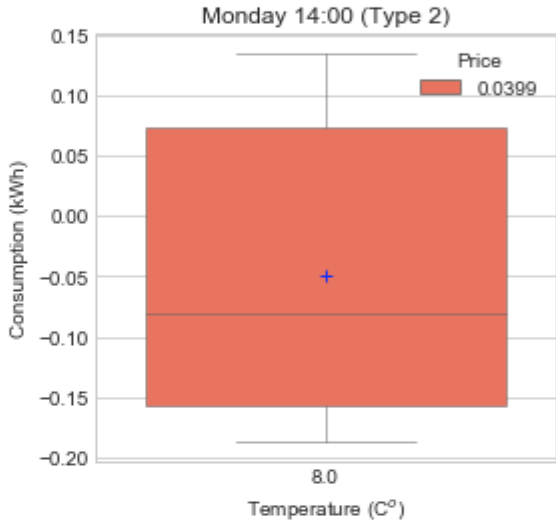

Monday 13:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
7.0
Temperature (C°)

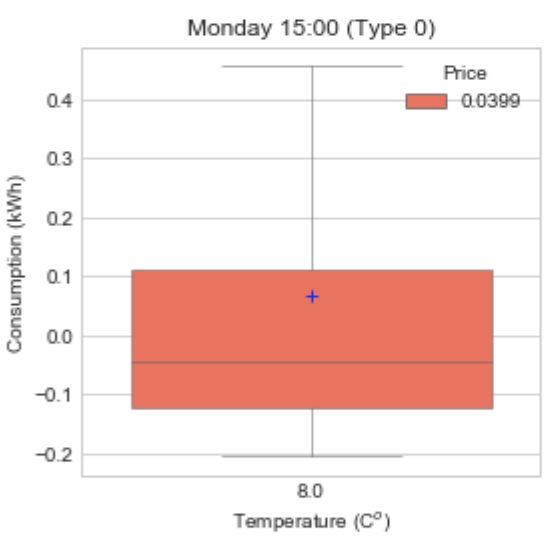

Monday 13:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
Temperature (C°)

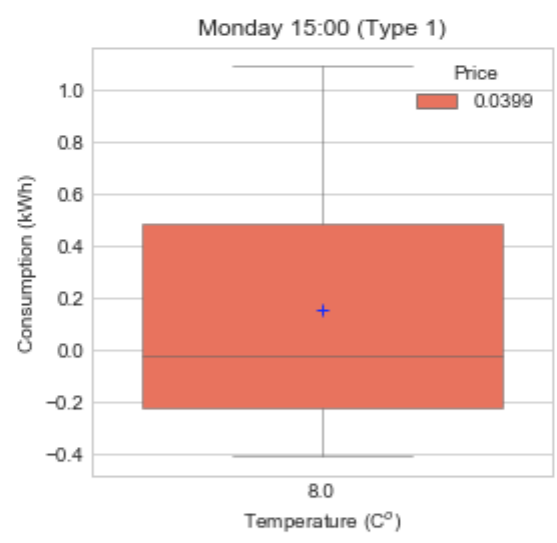

Monday 13:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.2
0.1
0.0
−0.1
−0.2
7.0
Temperature (C°)

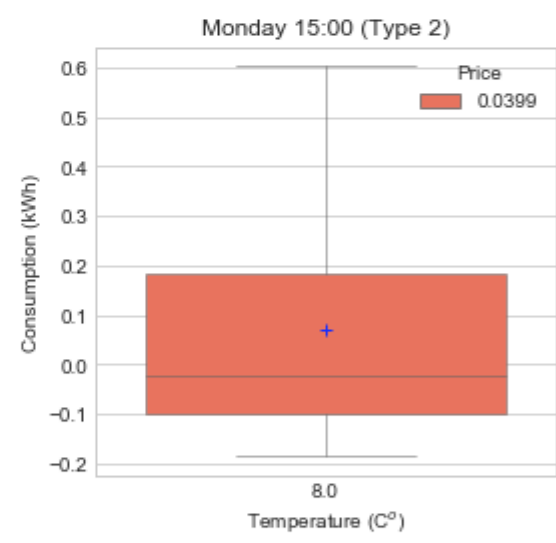

Monday 14:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
8.0
Temperature (C°)

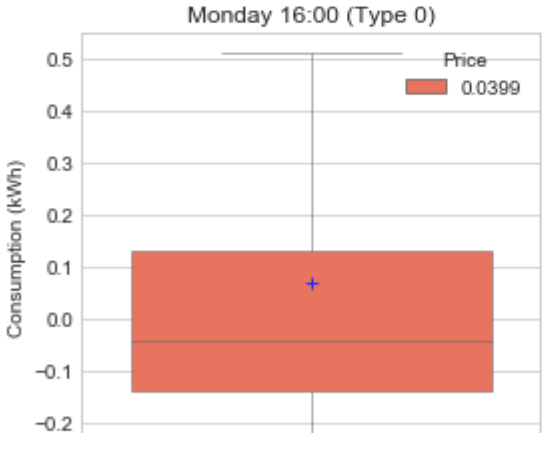

Monday 14:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
8.0
Temperature (C°)

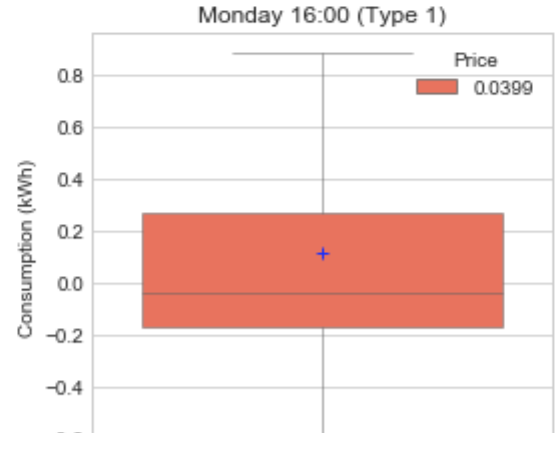

Monday 14:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
8.0
Temperature (C°)

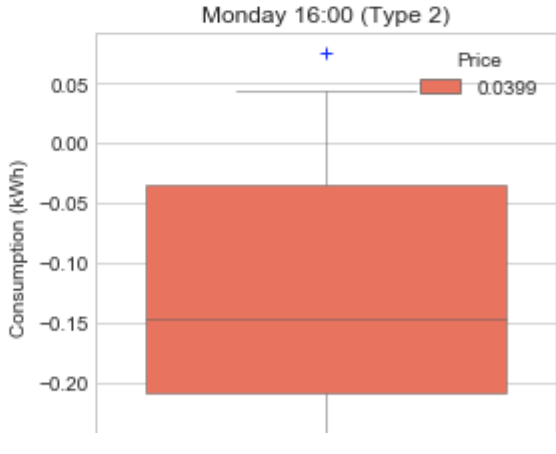

Monday 15:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
8.0
Temperature (C°)

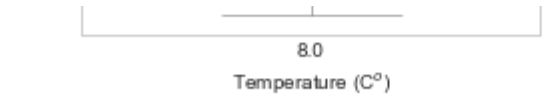

Monday 15:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
8.0
Temperature (C°)

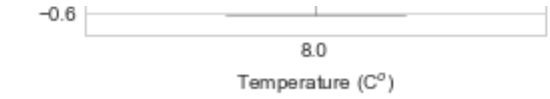

Monday 15:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
8.0
Temperature (C°)

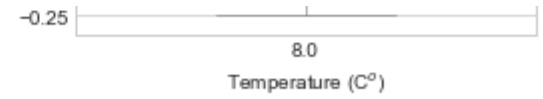

Monday 16:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2

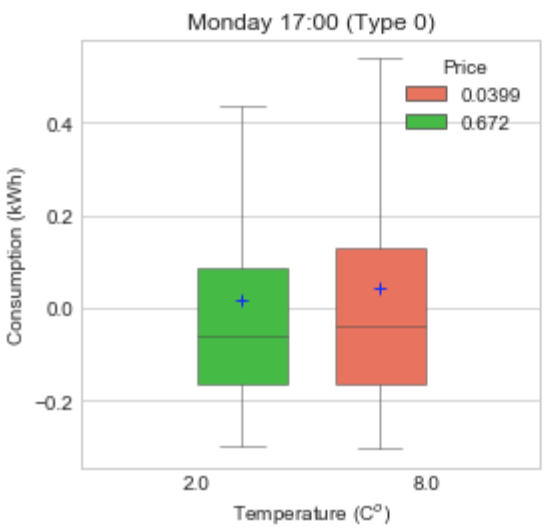

Monday 16:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

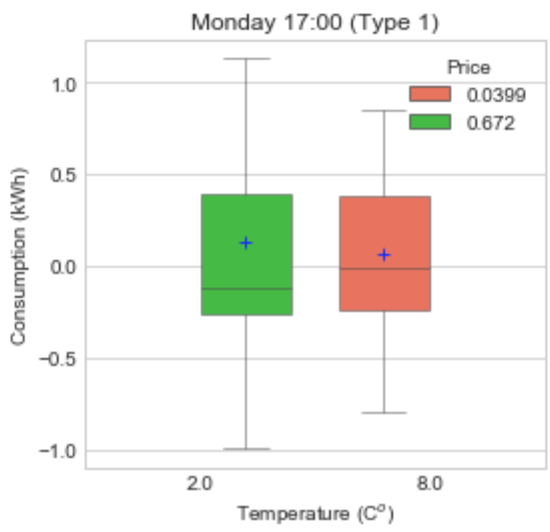

Monday 16:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.05
0.00
−0.05
−0.10
−0.15
−0.20

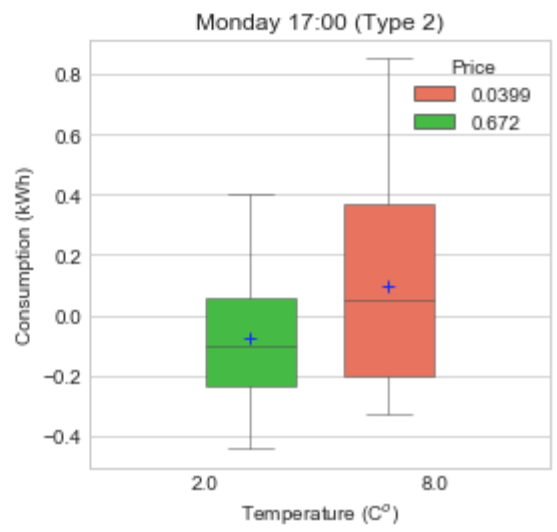

8.0
Temperature (C°)

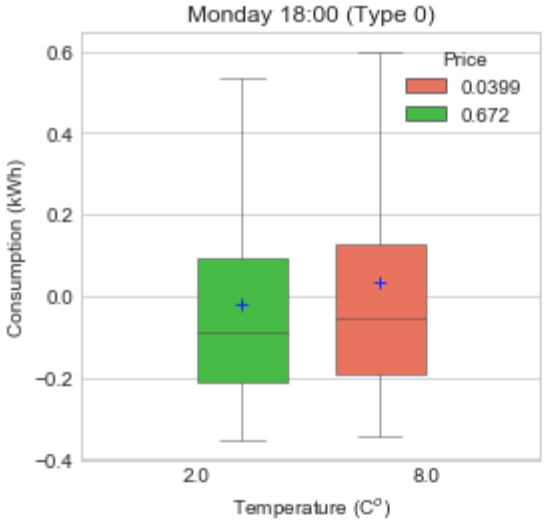

−0.6
8.0
Temperature (C°)

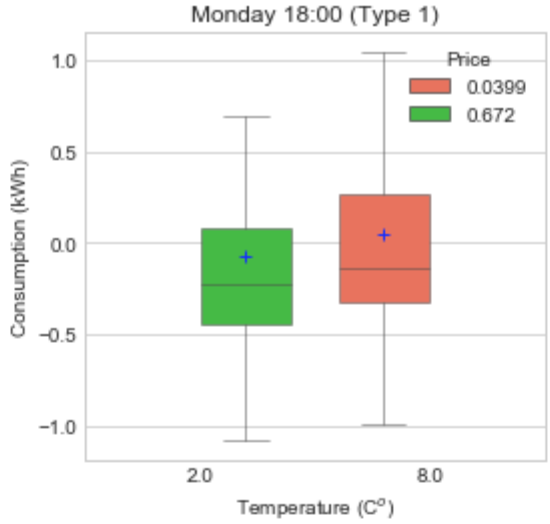

−0.25
8.0
Temperature (C°)

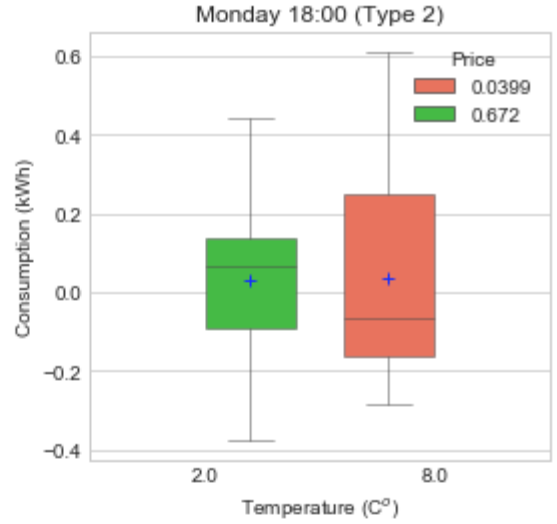

Monday 17:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
2.0
8.0
Temperature (C°)

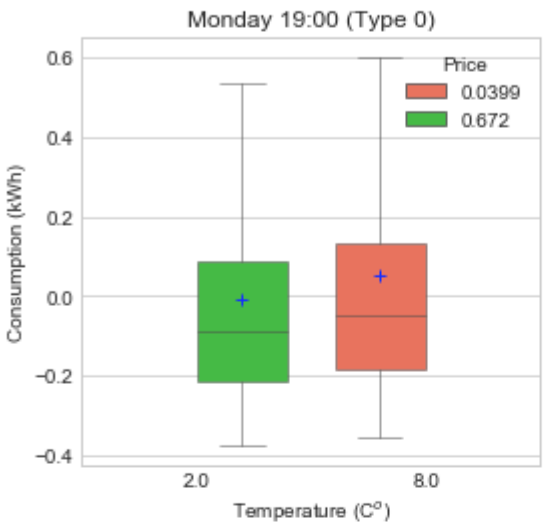

Monday 17:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
8.0
Temperature (C°)

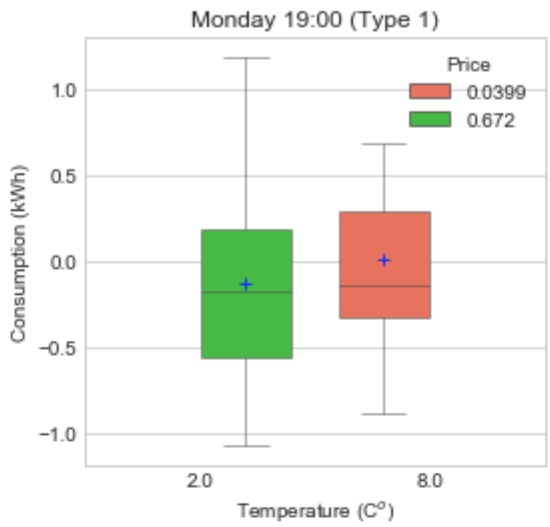

Monday 17:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
8.0
Temperature (C°)

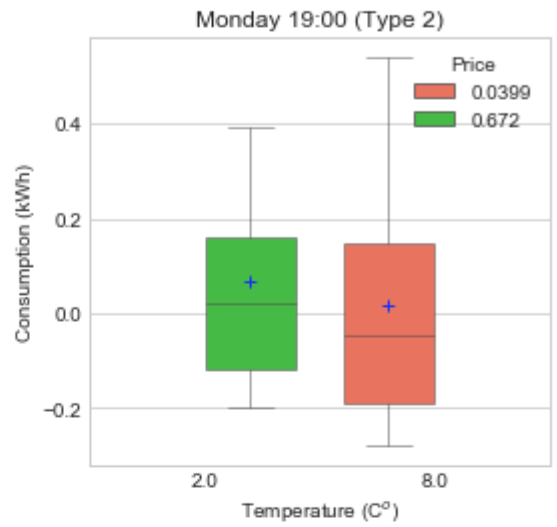

Monday 18:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
2.0
8.0
Temperature (C°)

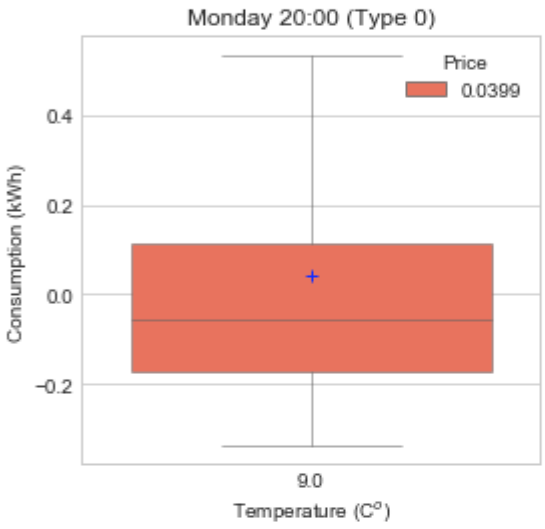

Monday 18:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
8.0
Temperature (C°)

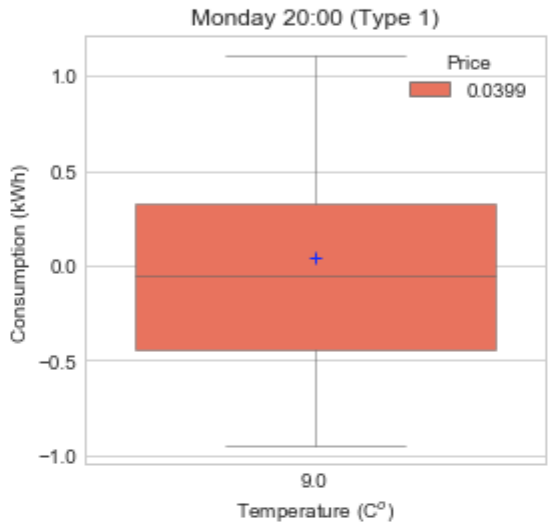

Monday 18:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
8.0
Temperature (C°)

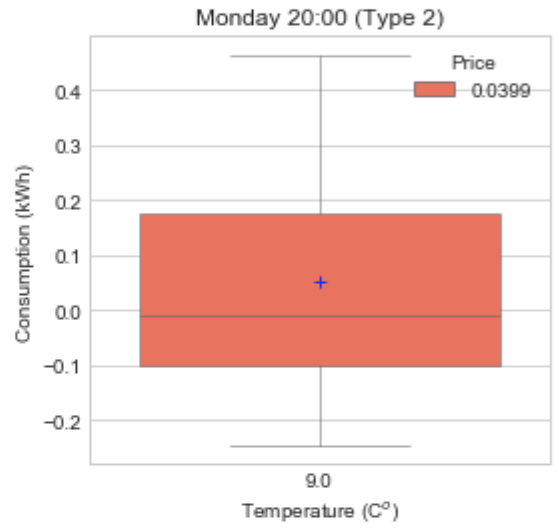

Monday 19:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
8.0
Temperature (C°)

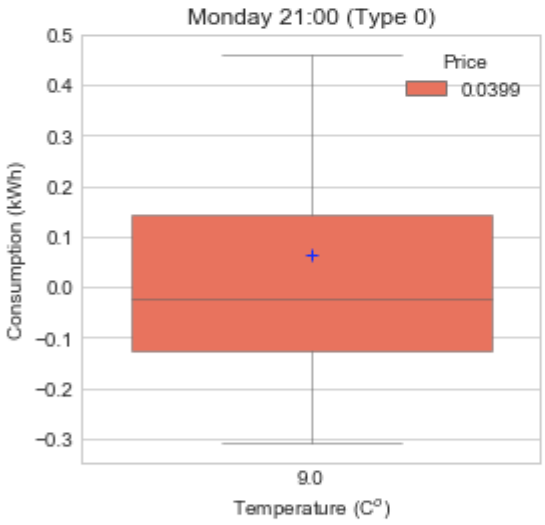

Monday 19:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
8.0
Temperature (C°)

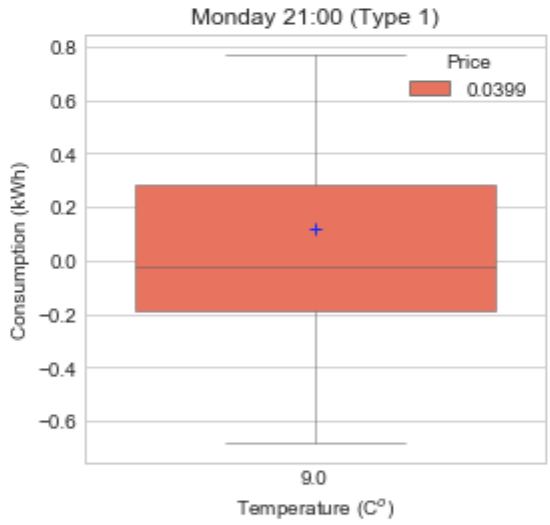

Monday 19:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
2.0
8.0
Temperature (C°)

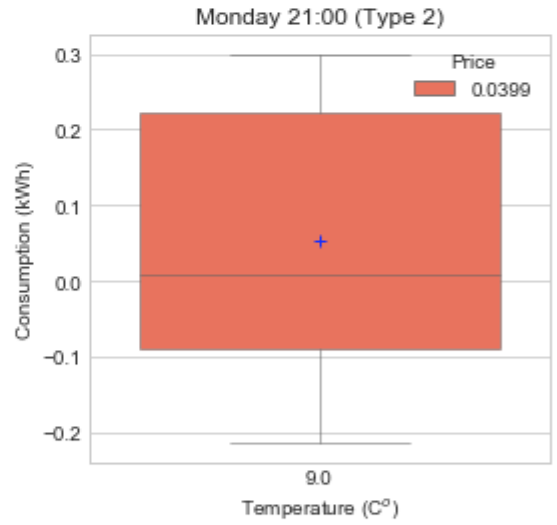

Monday 20:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
9.0
Temperature (C°)

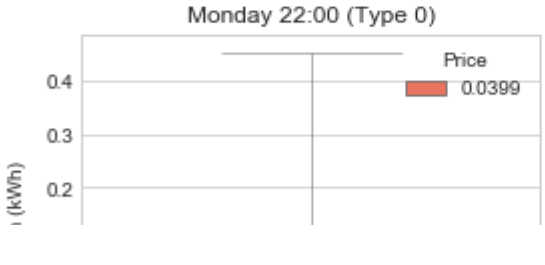

Monday 20:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.0
0.5
0.0
−0.5
9.0
Temperature (C°)

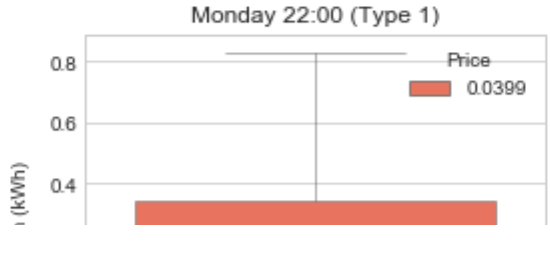

Monday 20:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
9.0
Temperature (C°)

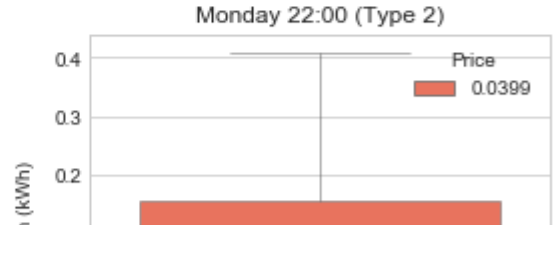

Monday 21:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
9.0
Temperature (C°)

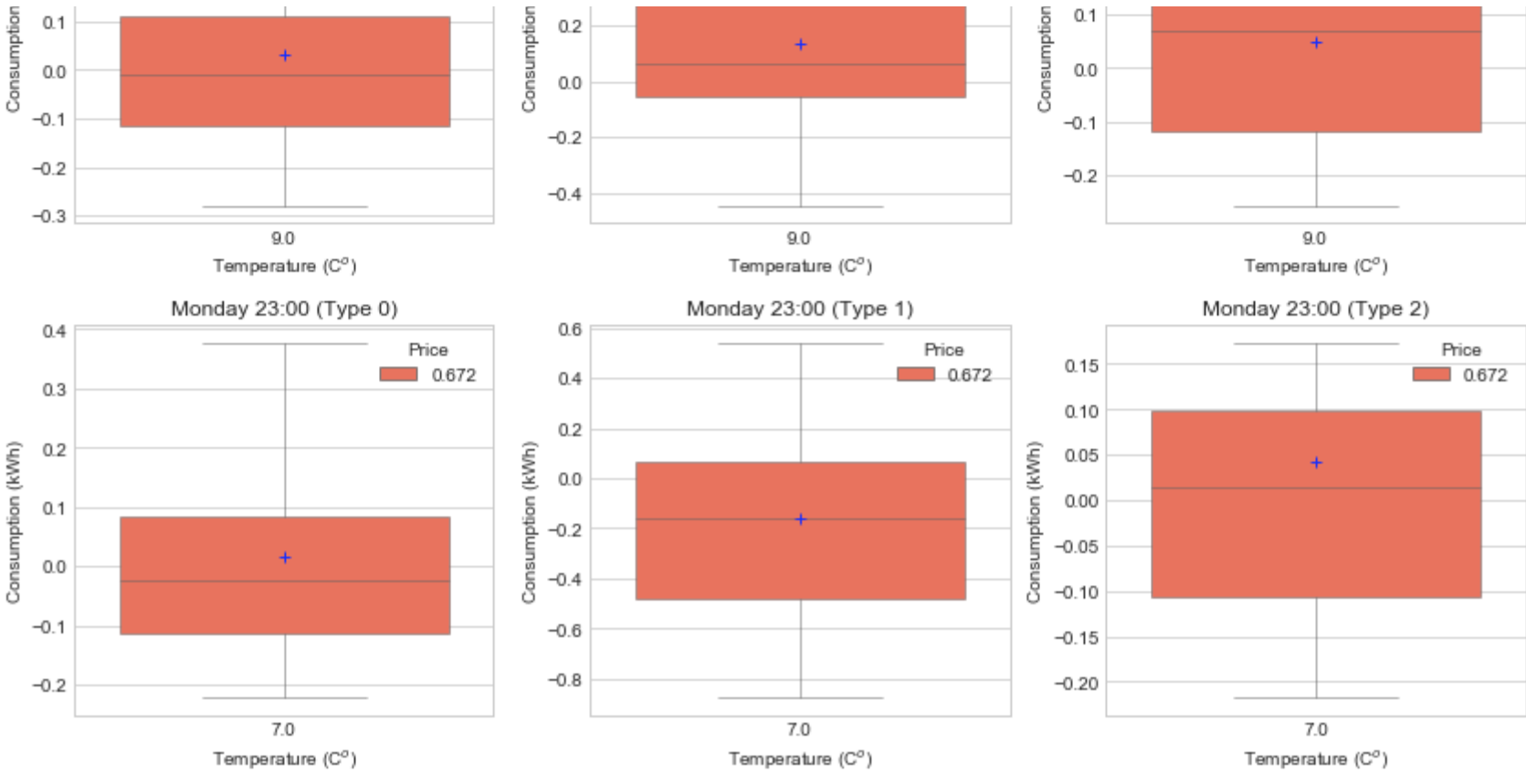

Monday 21:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
9.0
Temperature (C°)


Monday 21:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
9.0
Temperature (C°)

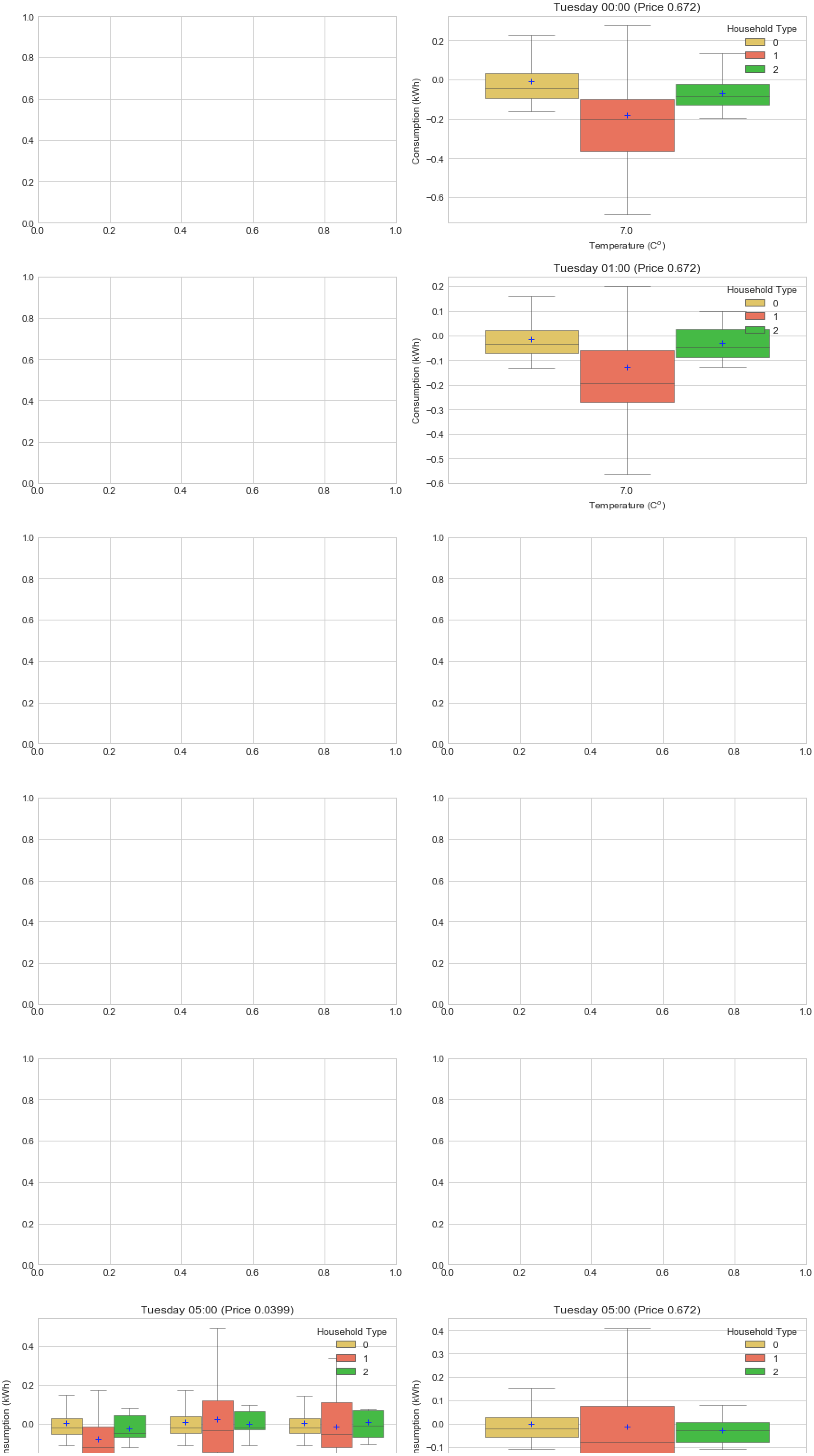

Monday 22:00 (Type 0)
Price
0.0399
0.4
0.3
0.2


Monday 22:00 (Type 1)
Price
0.0399
0.8
0.6
0.4

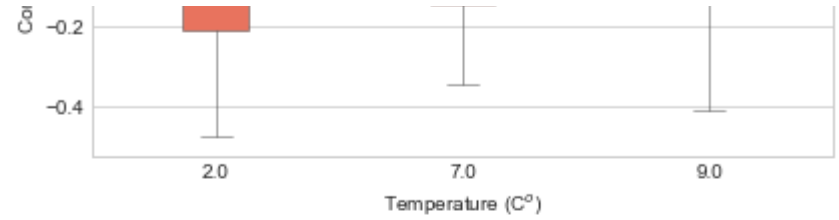

Monday 22:00 (Type 2)
Price
0.0399
0.4
0.3
0.2

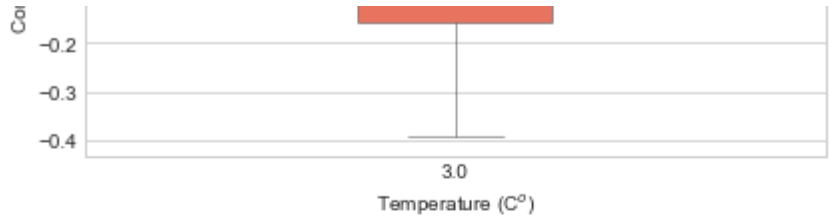

Consumption
Temperature (C°)
9.0
Monday 23:00 (Type 0)
Monday 23:00 (Type 1)
Monday 23:00 (Type 2)
Price
0.672
Consumption (kWh)
7.0

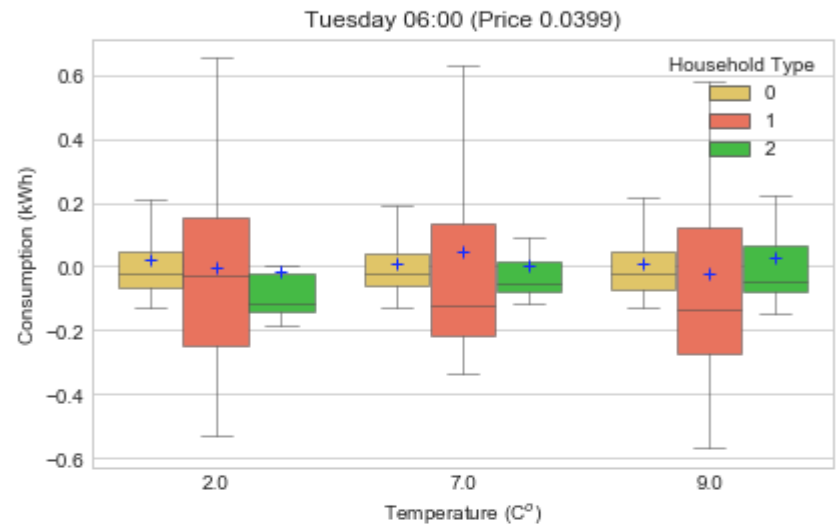
In [98]:
```
# Tuesday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 1 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

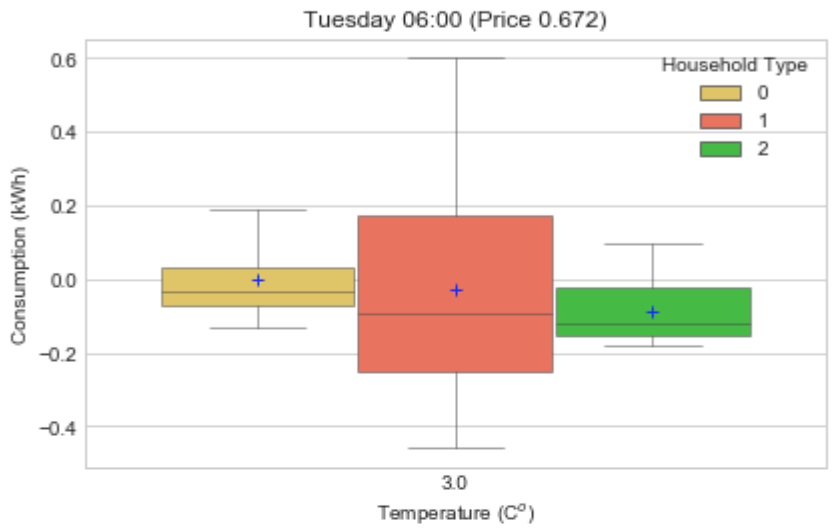


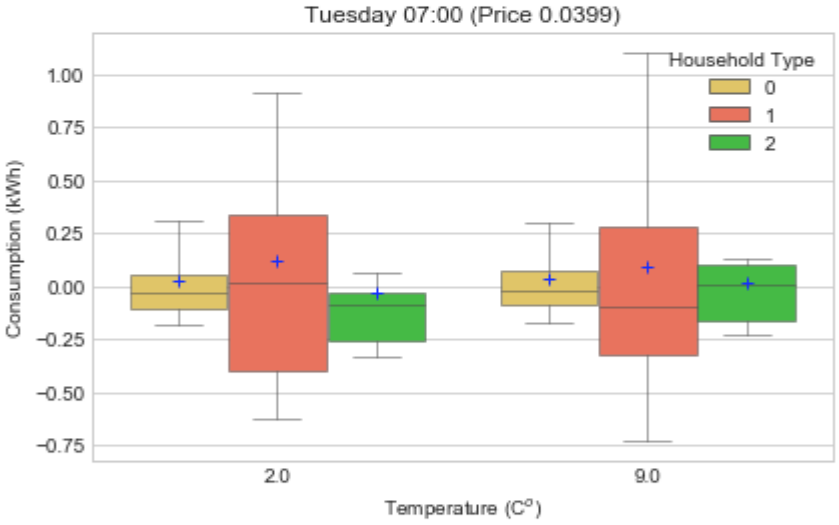
Tuesday 00:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
7.0
Temperature (C°)
Tuesday 01:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
7.0
Temperature (C°)
Tuesday 05:00 (Price 0.0399)
Household Type
0
1
2
nsumption (kWh)
Tuesday 05:00 (Price 0.672)
Household Type
0
1
2
nsumption (kWh)

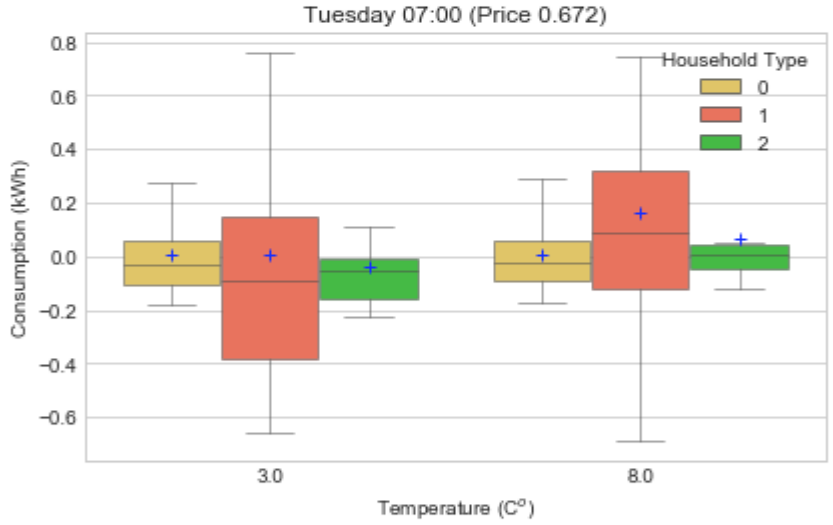

−0.2
−0.4
2.0
7.0
9.0
Temperature (C°)

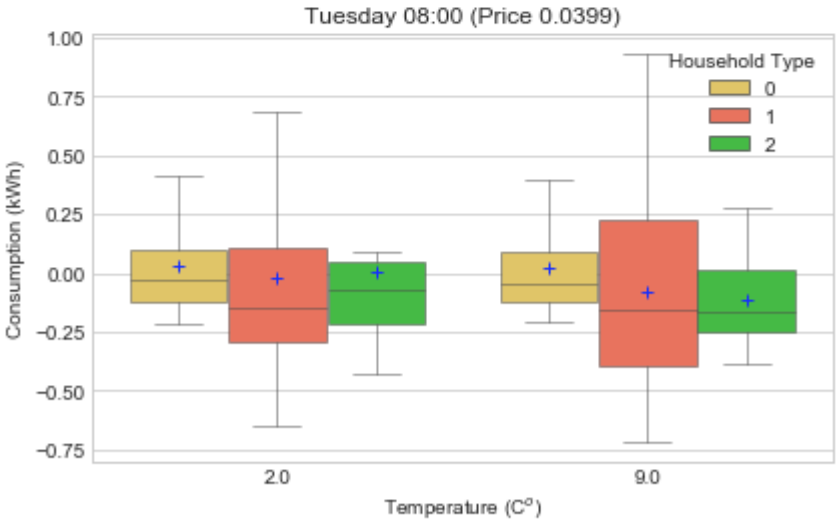

−0.2
−0.3
−0.4
3.0
Temperature (C°)

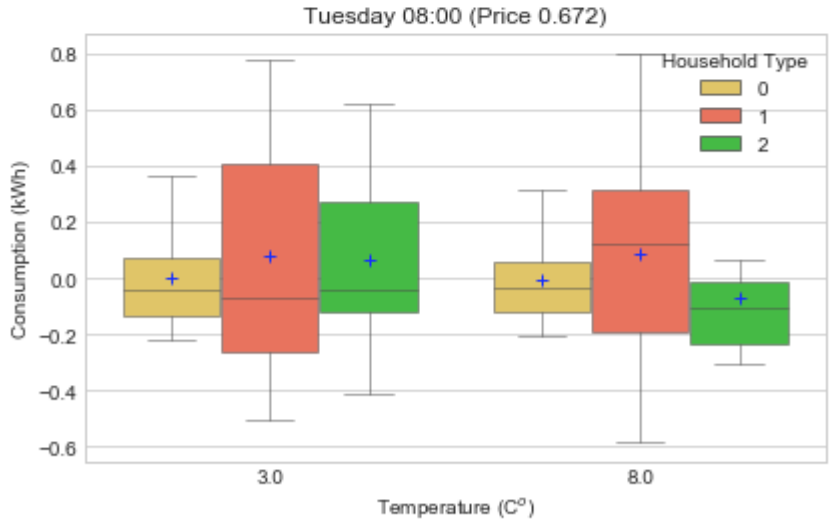

Tuesday 06:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
7.0
9.0
Temperature (C°)

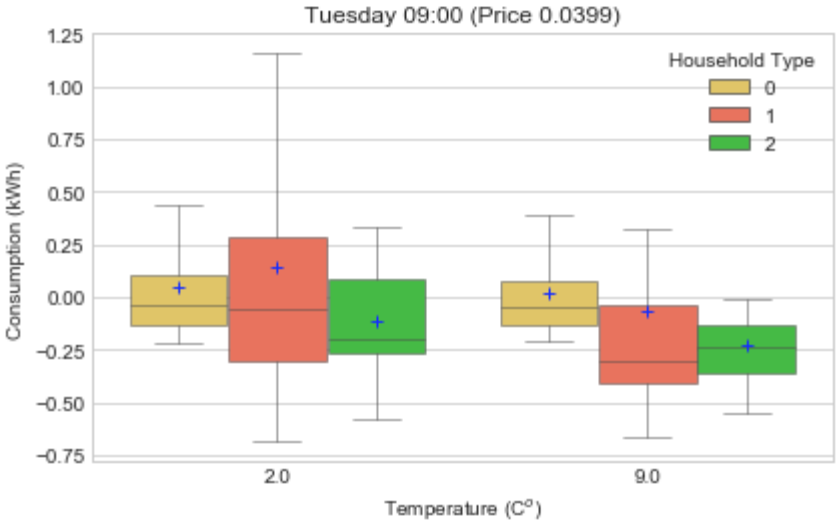

Tuesday 06:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
3.0
Temperature (C°)

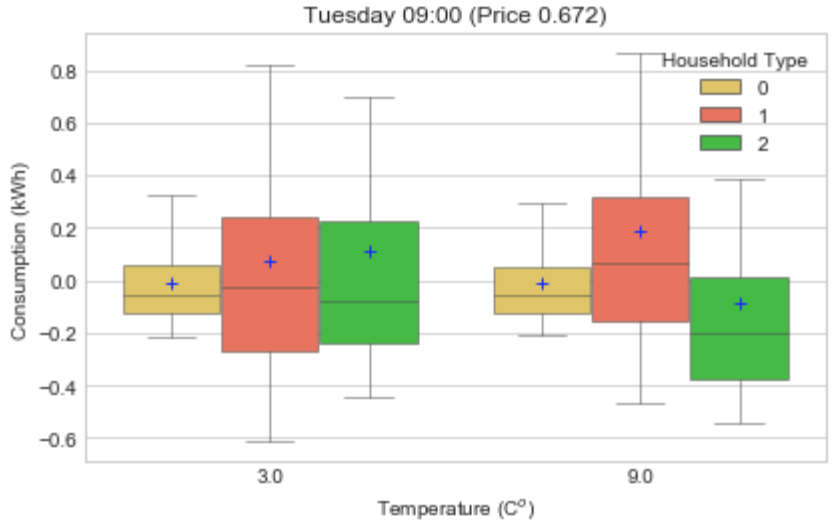

Tuesday 07:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
9.0
Temperature (C°)

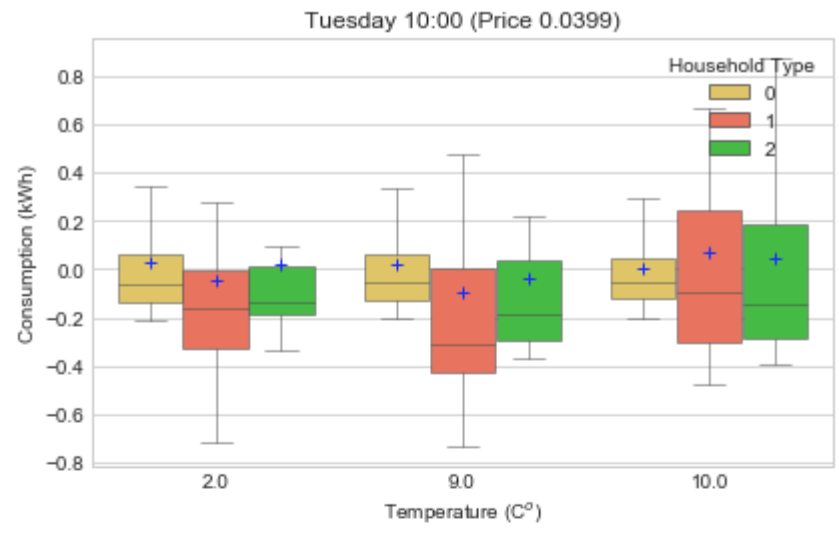

Tuesday 07:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
8.0
Temperature (C°)

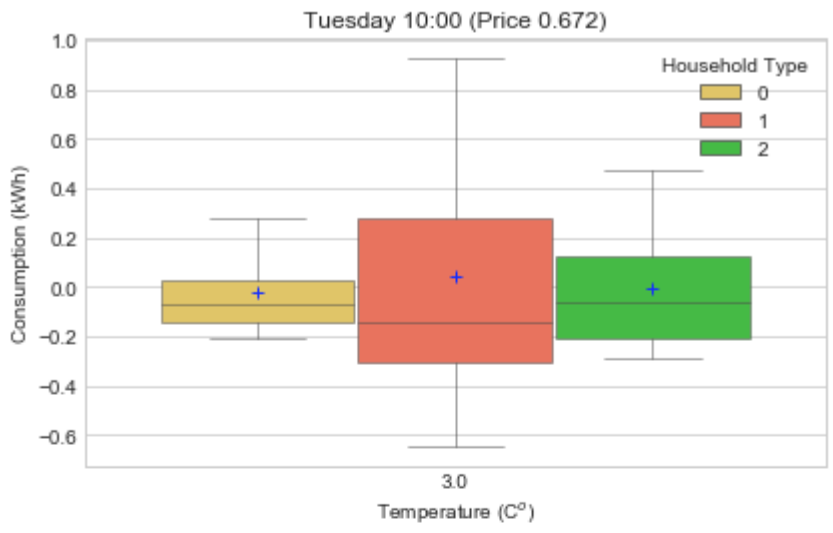

Tuesday 08:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
9.0
Temperature (C°)

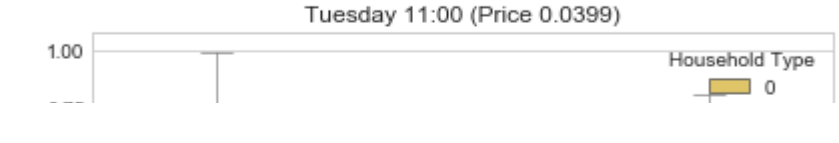

Tuesday 08:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
8.0
Temperature (C°)


Tuesday 09:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
9.0
Temperature (C°)

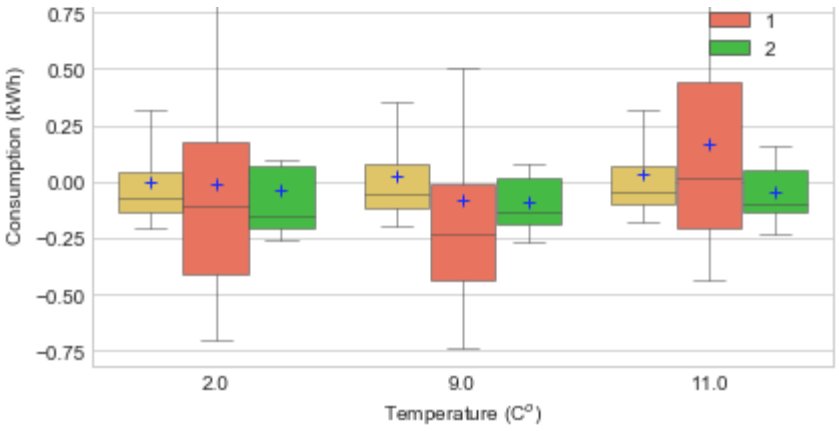

Tuesday 09:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
9.0
Temperature (C°)

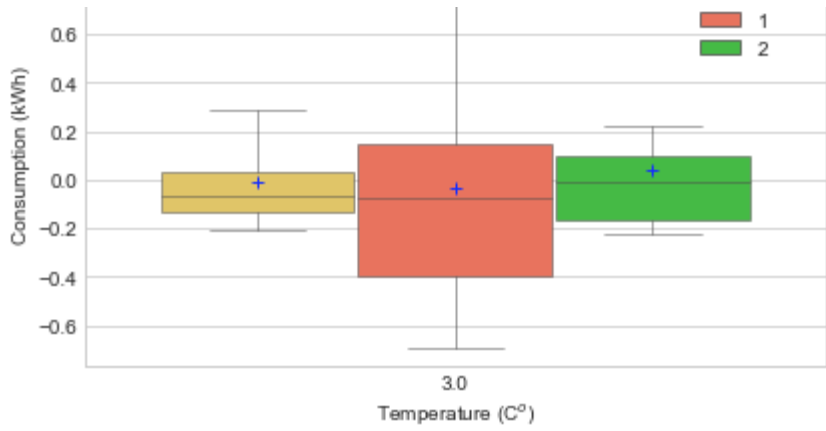

Tuesday 10:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
2.0
9.0
10.0
Temperature (C°)

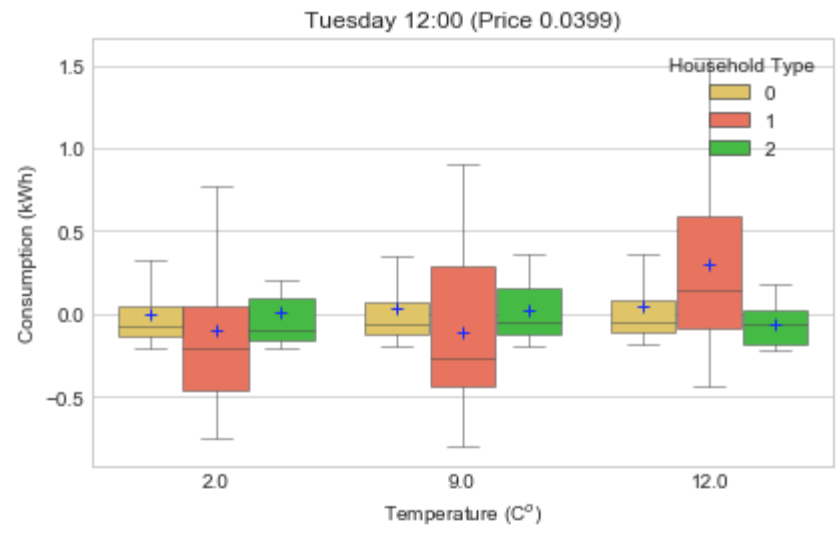

Tuesday 10:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
Temperature (C°)

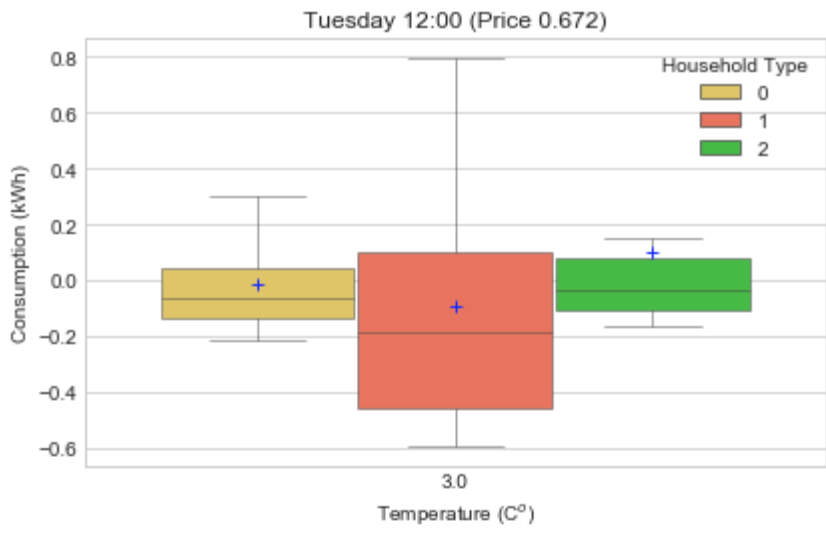

Tuesday 11:00 (Price 0.0399)
Household Type
0
1.00

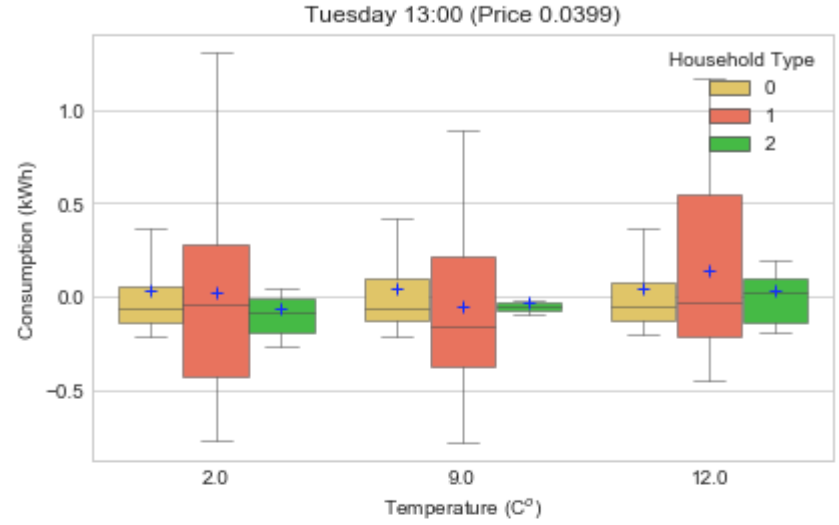

Tuesday 11:00 (Price 0.672)
Household Type
0
0.8

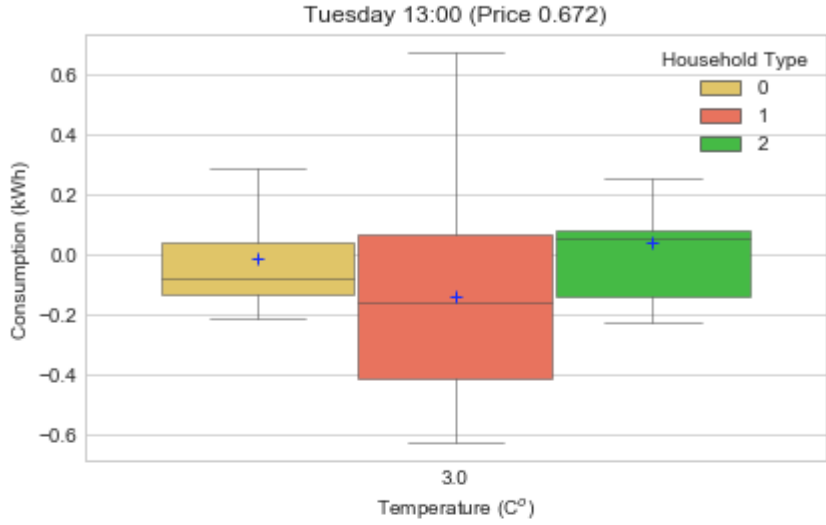

1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
9.0
11.0
Temperature (C°)

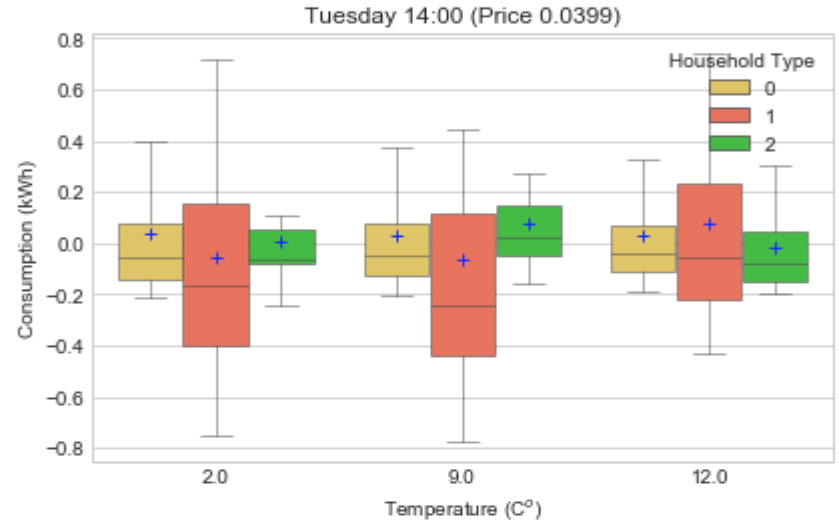

1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
Temperature (C°)

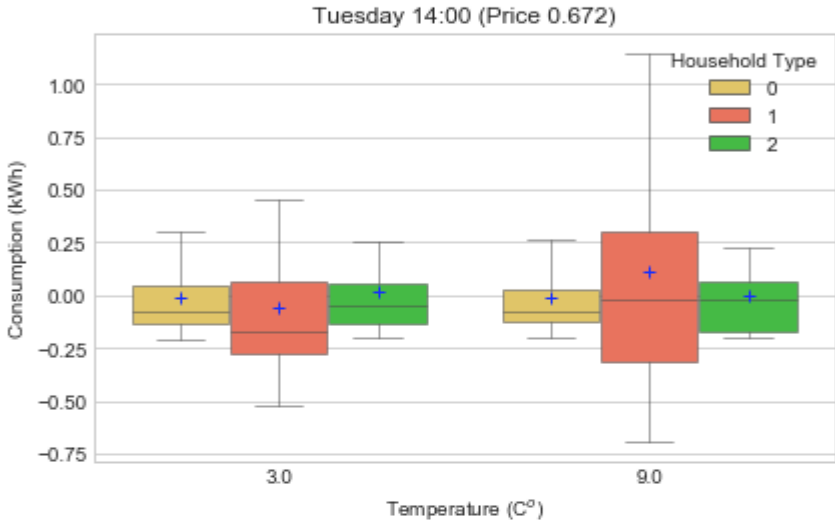

Tuesday 12:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
2.0
9.0
12.0
Temperature (C°)

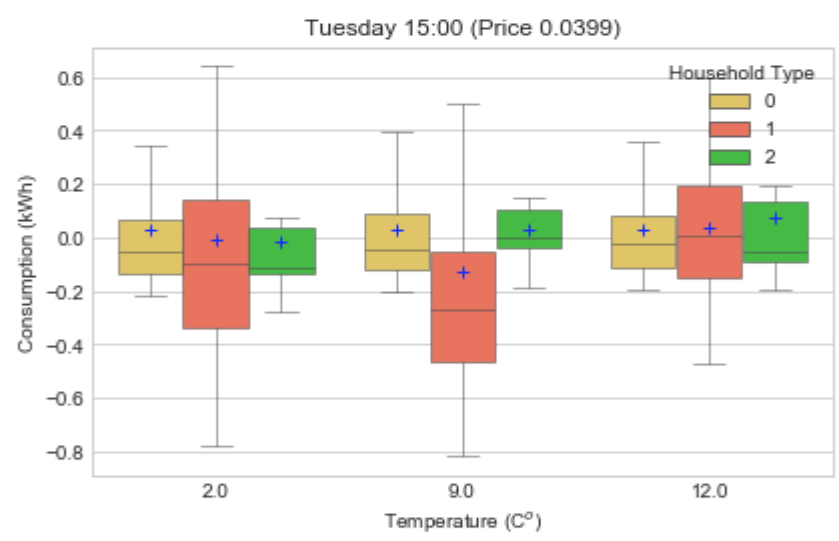

Tuesday 12:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
Temperature (C°)

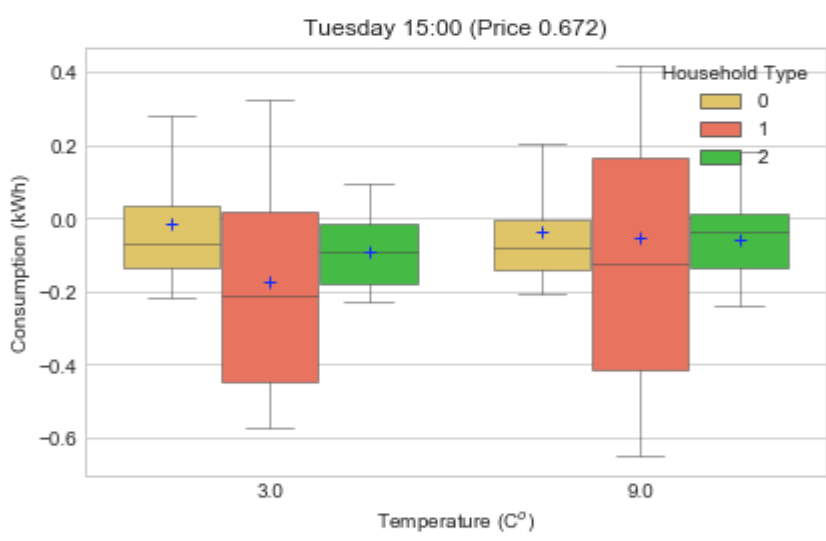

Tuesday 13:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
2.0
9.0
12.0
Temperature (C°)

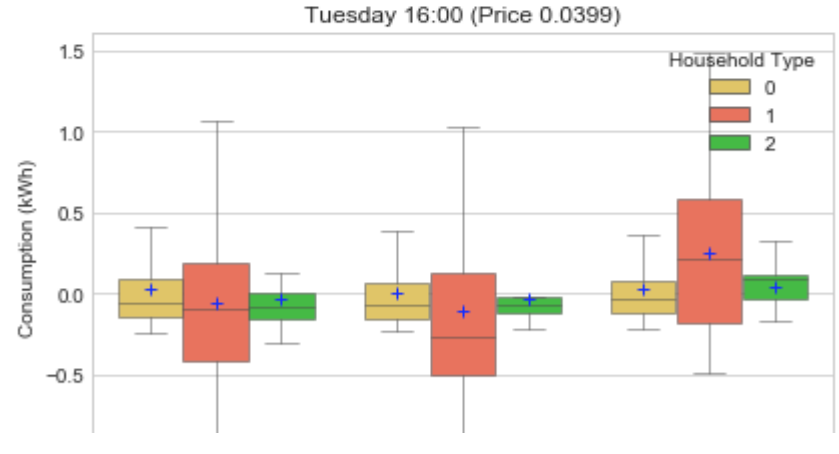

Tuesday 13:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
Temperature (C°)

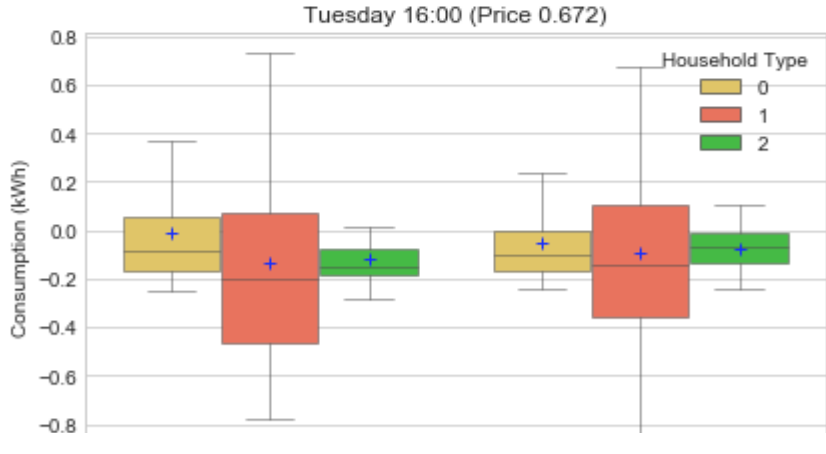

Tuesday 14:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
2.0
9.0
12.0
Temperature (C°)

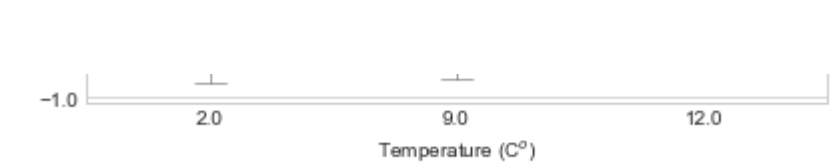

Tuesday 14:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
9.0
Temperature (C°)


Tuesday 15:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
2.0
9.0
12.0
Temperature (C°)

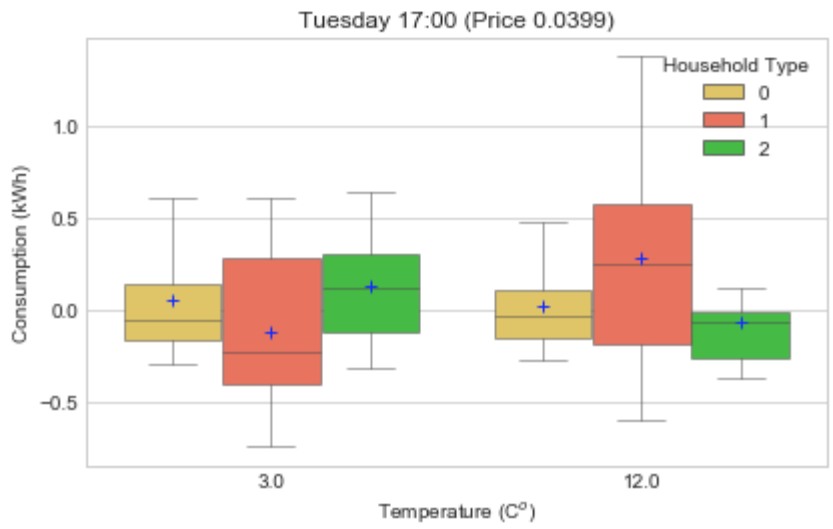

Tuesday 15:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
9.0
Temperature (C°)

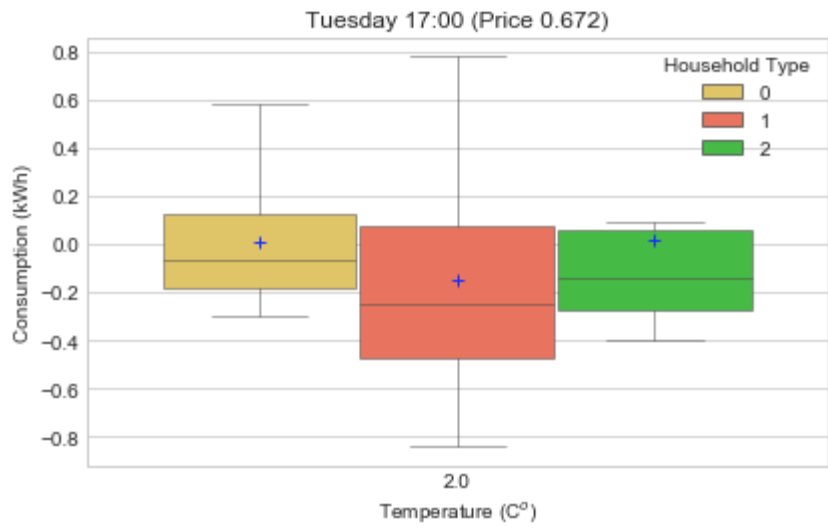

Tuesday 16:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5

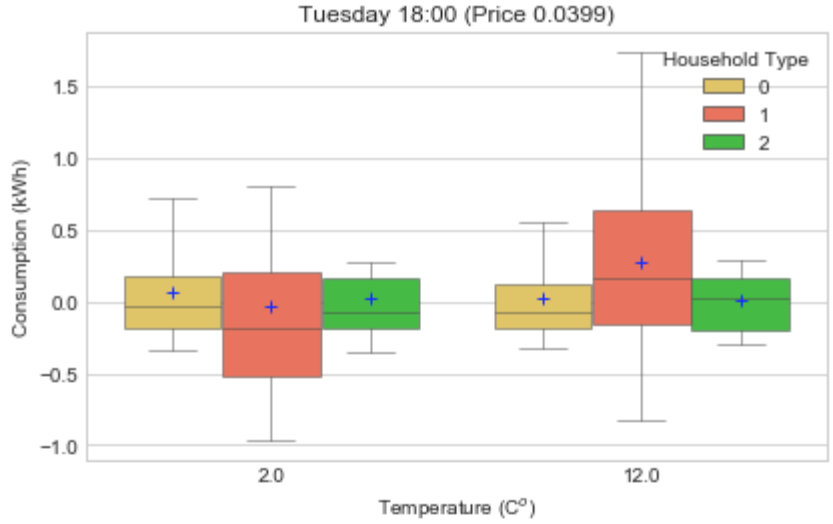

Tuesday 16:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8

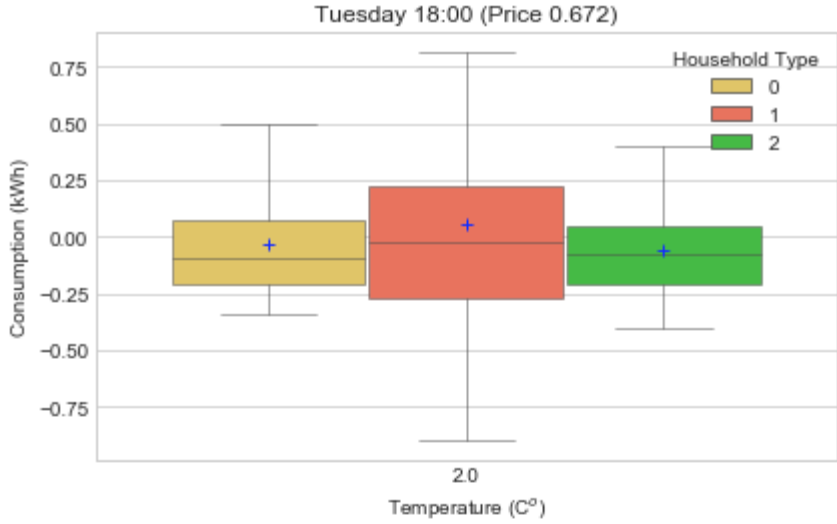

2.0
9.0
12.0
Temperature (C°)

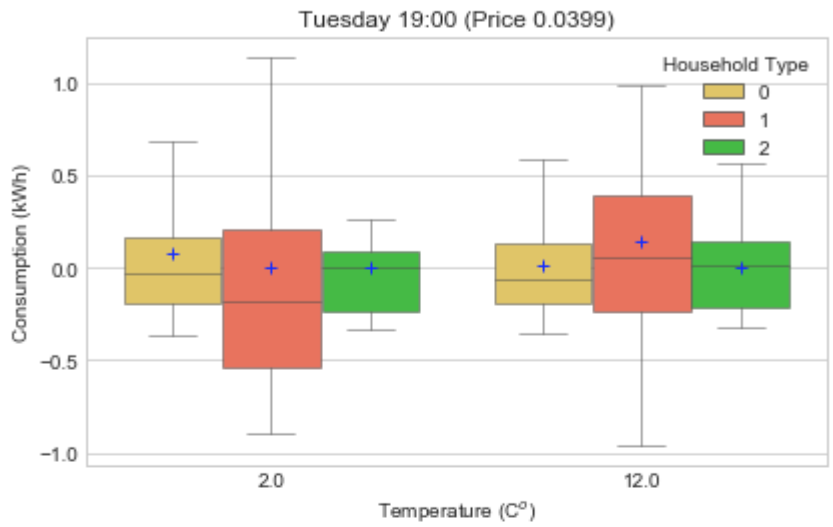

3.0
7.0
Temperature (C°)

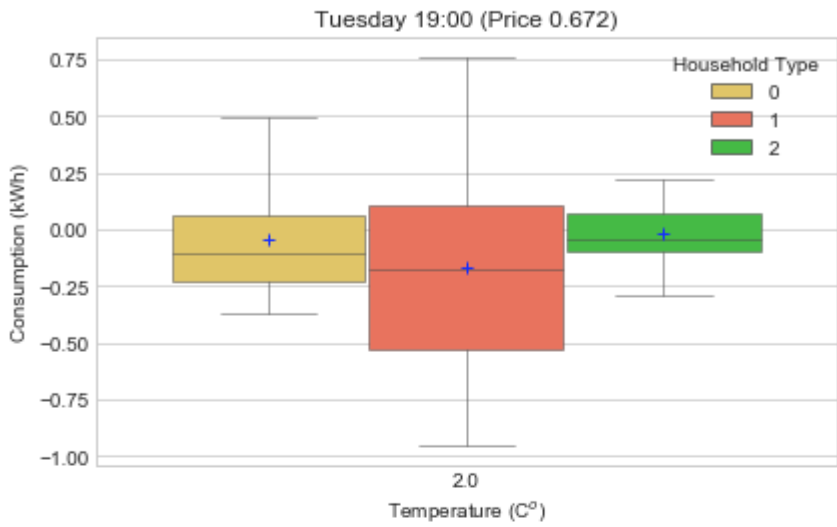

Tuesday 17:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
3.0
12.0
Temperature (C°)

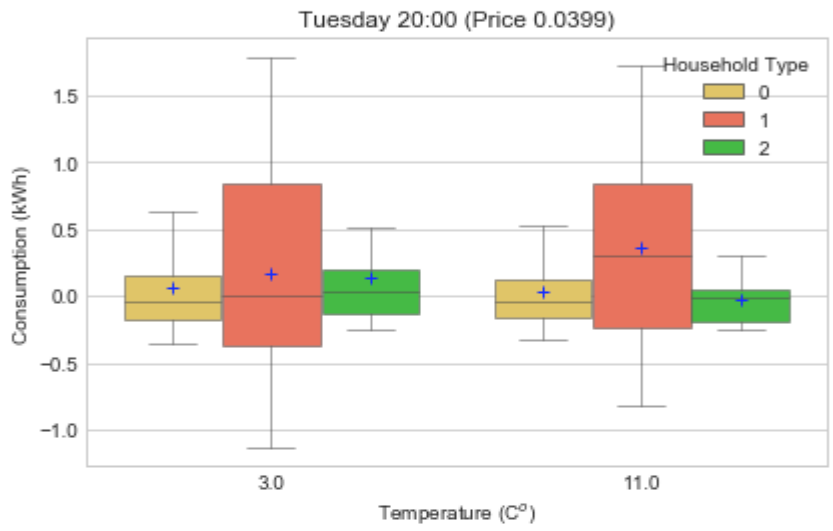

Tuesday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
Temperature (C°)

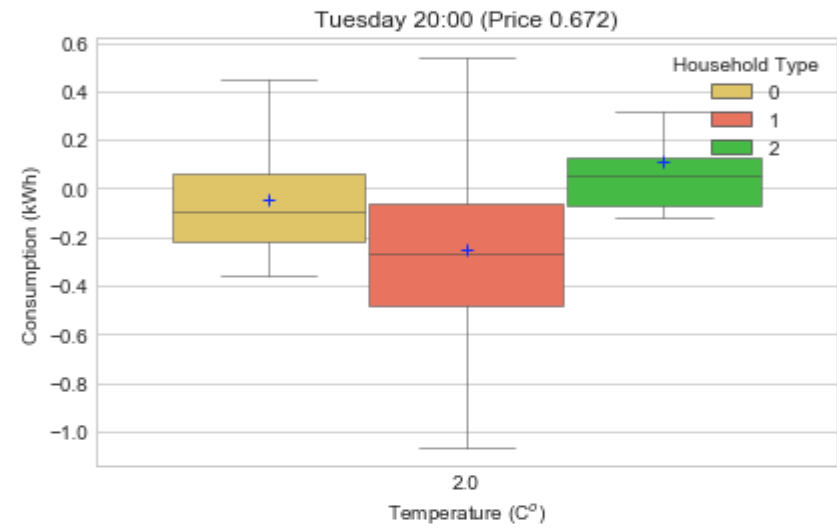

Tuesday 18:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.0
12.0
Temperature (C°)

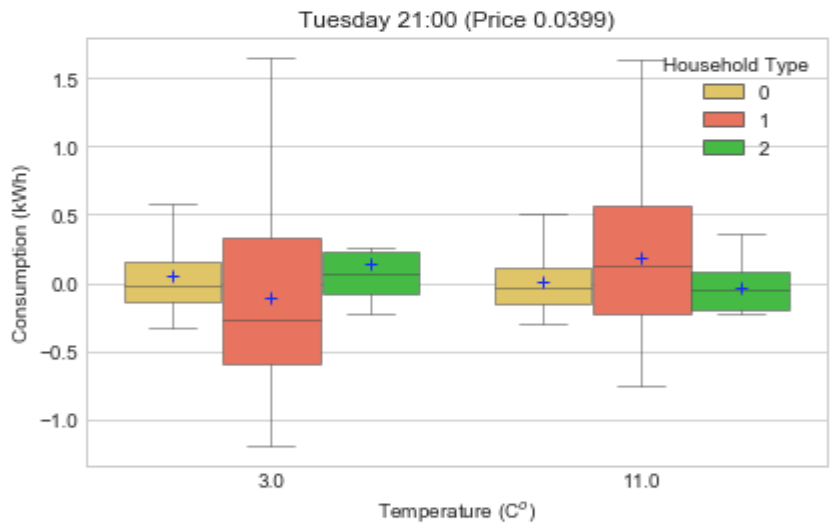

Tuesday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
Temperature (C°)

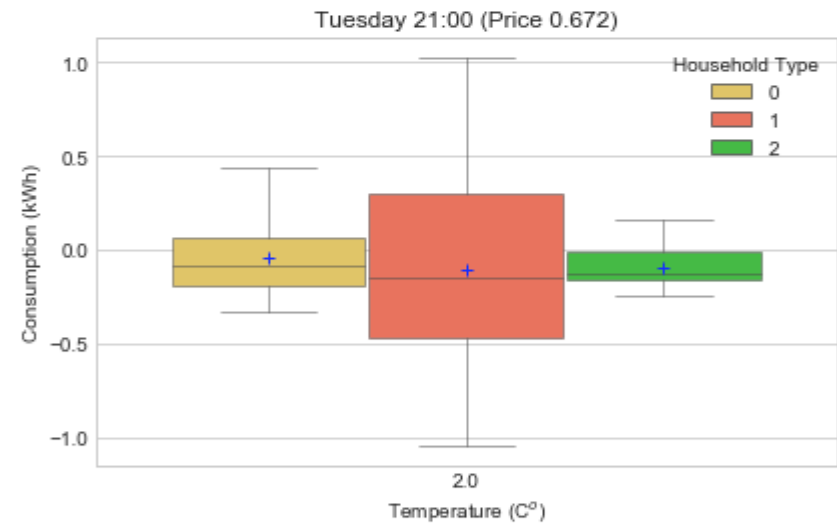

Tuesday 19:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.0
12.0
Temperature (C°)


Tuesday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
Temperature (C°)

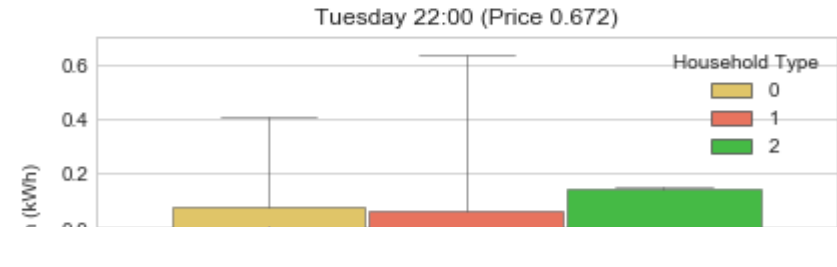

Tuesday 20:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
3.0
11.0
Temperature (C°)

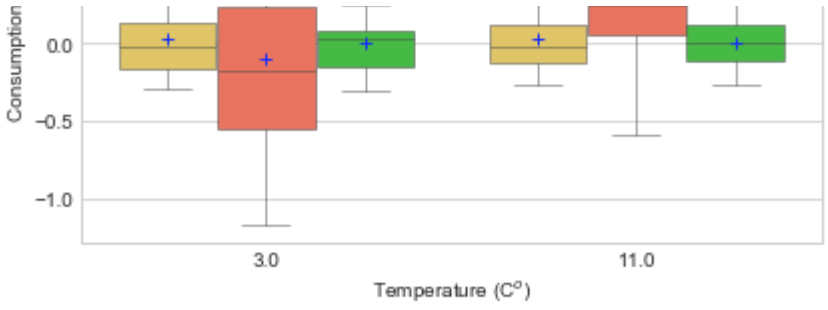

Tuesday 20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
Temperature (C°)

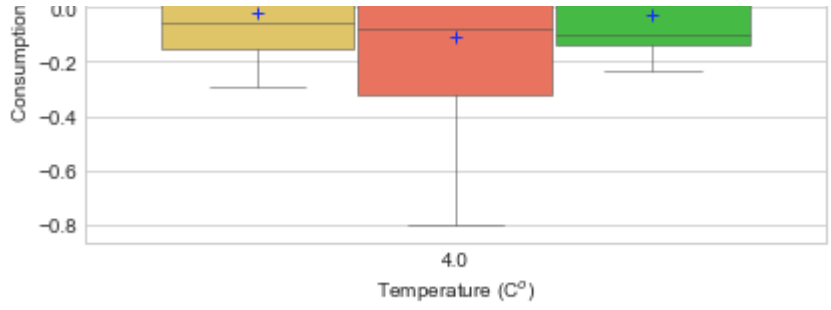

Tuesday 21:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
3.0
11.0
Temperature (C°)

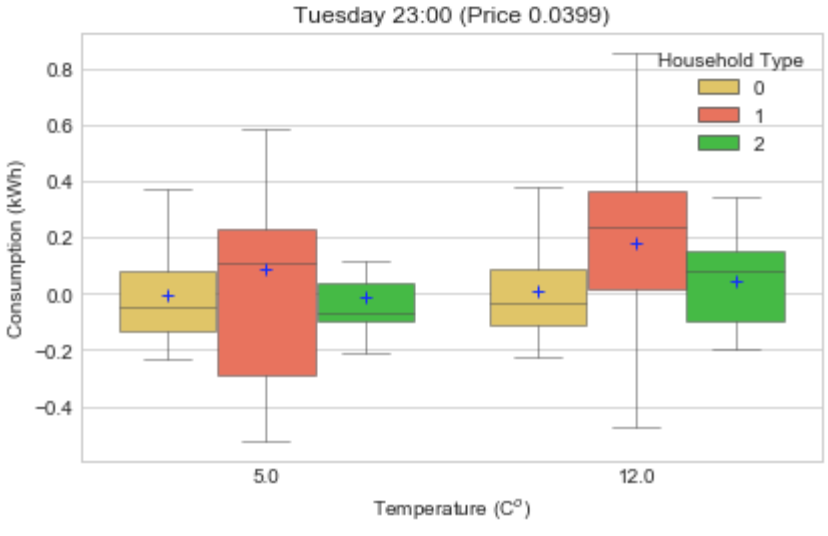

Tuesday 21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
Temperature (C°)

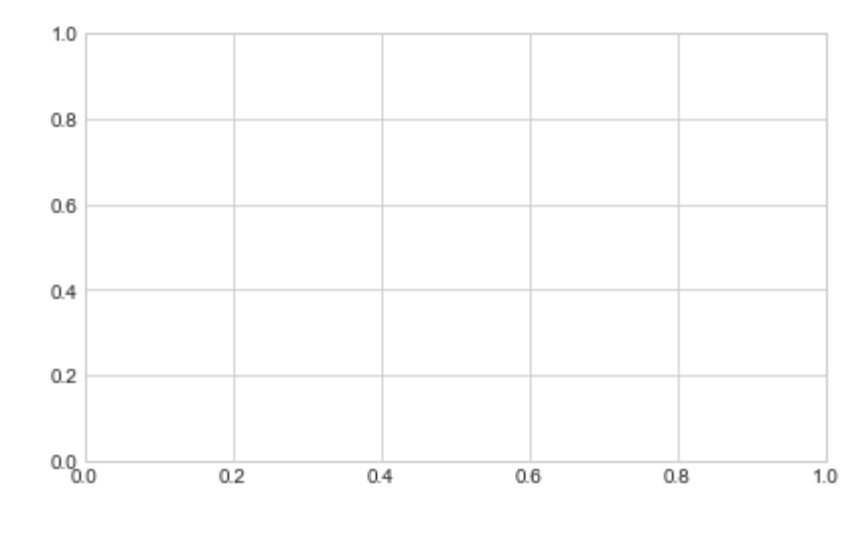

Tuesday 22:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)


Tuesday 22:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)

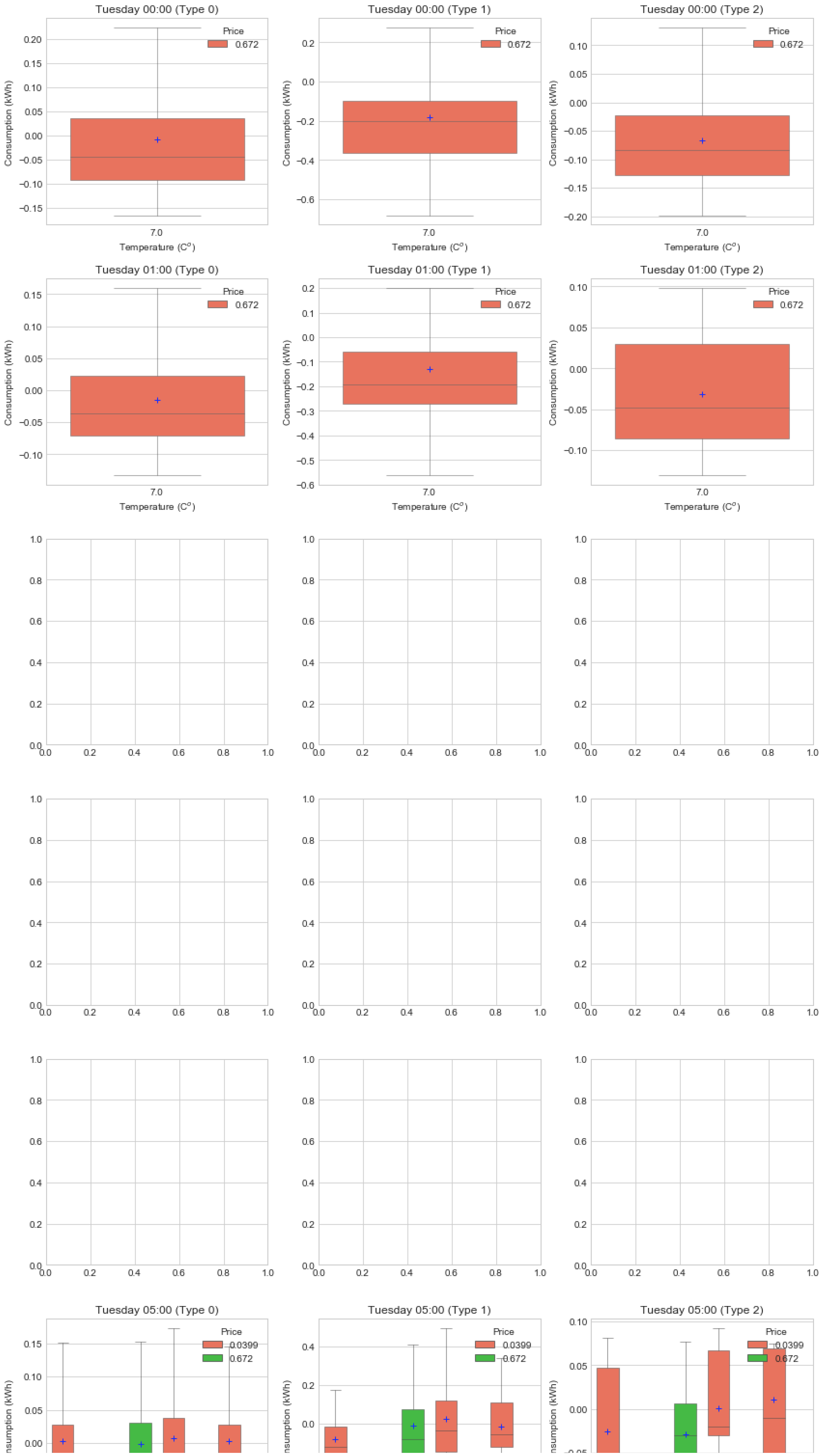

Consumption
0.0
−0.5
−1.0
3.0
11.0
Temperature (C°)

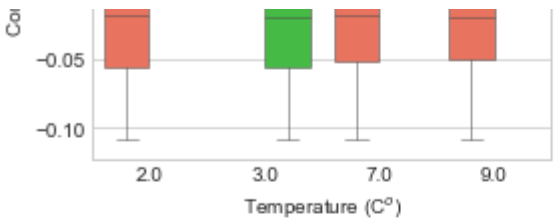

Consumption
0.0
−0.2
−0.4
−0.6
−0.8
4.0
Temperature (C°)

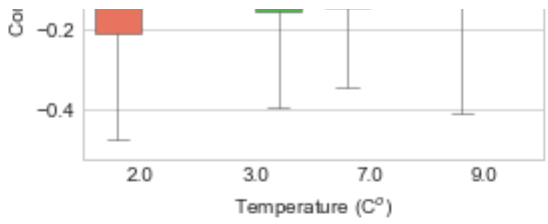

Tuesday 23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
5.0
12.0
Temperature (C°)

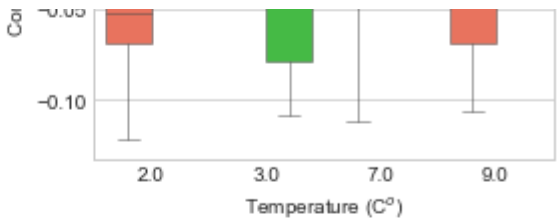

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

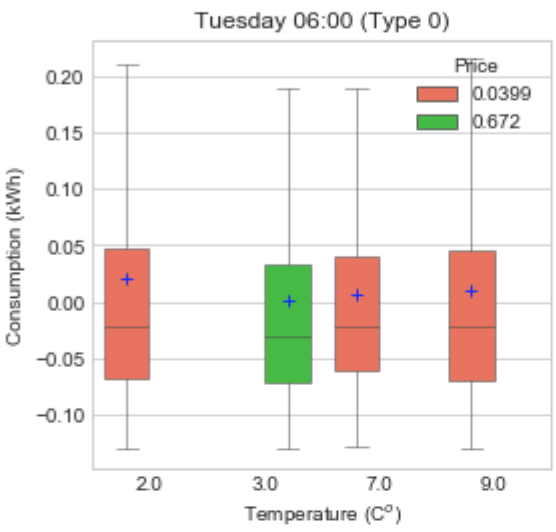
In [129]:
```
# Price comparison
# Tuesday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 1 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

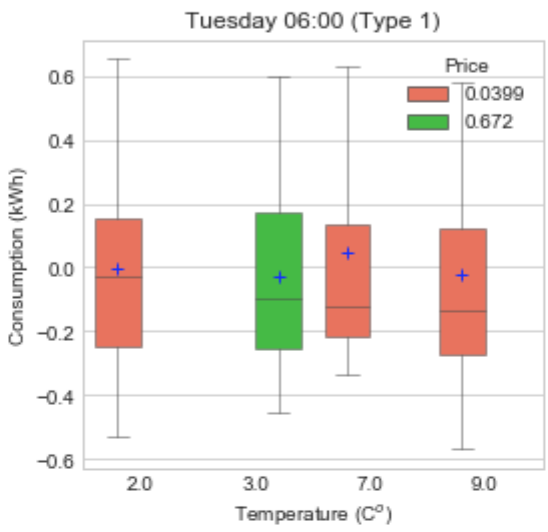

Tuesday 00:00 (Type 0)
Tuesday 00:00 (Type 1)
Tuesday 00:00 (Type 2)
Tuesday 01:00 (Type 0)
Tuesday 01:00 (Type 1)
Tuesday 01:00 (Type 2)
Tuesday 05:00 (Type 0)
Tuesday 05:00 (Type 1)
Tuesday 05:00 (Type 2)
Price
0.672
0.0399
Consumption (kWh)
Temperature (C°)
7.0

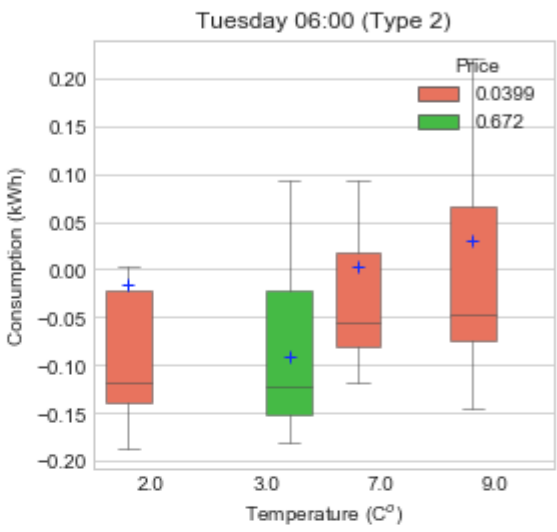

−0.05
−0.10
2.0
3.0
7.0
9.0
Temperature (C°)

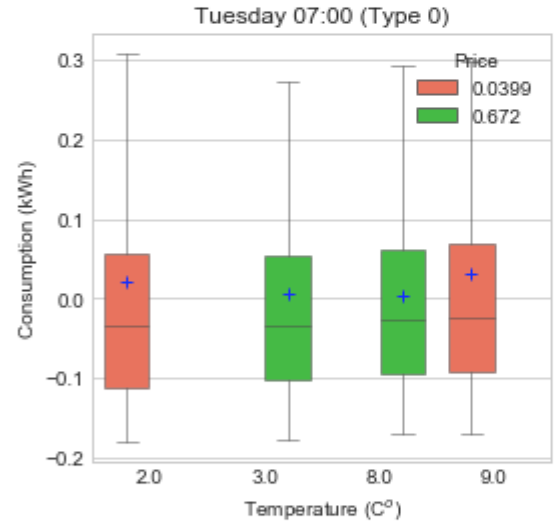

−0.2
−0.4
2.0
3.0
7.0
9.0
Temperature (C°)

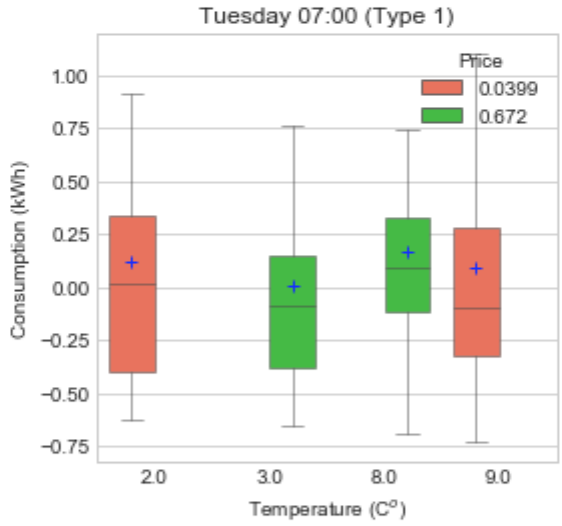

−0.10
2.0
3.0
7.0
9.0
Temperature (C°)

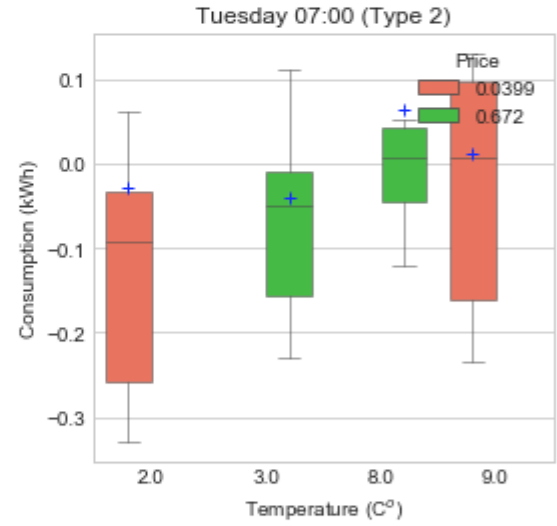

Tuesday 06:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
2.0
3.0
7.0
9.0
Temperature (C°)

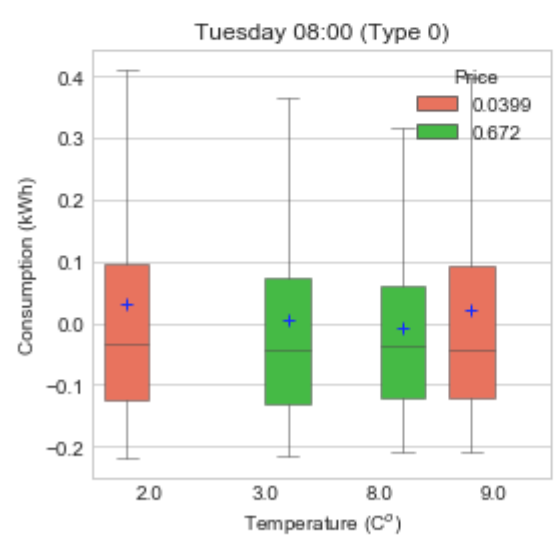

Tuesday 06:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
3.0
7.0
9.0
Temperature (C°)

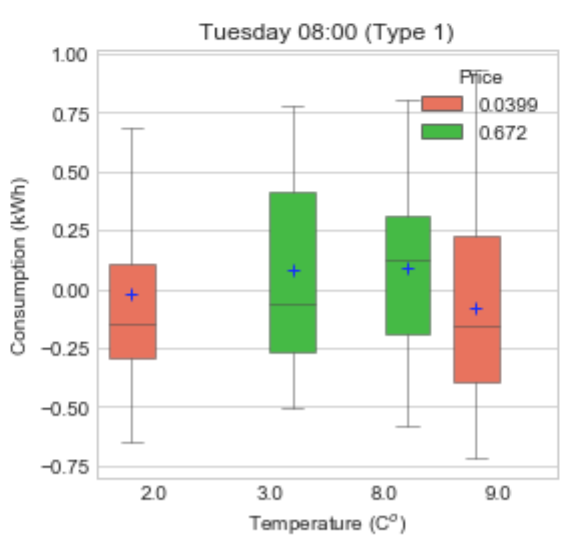

Tuesday 06:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
2.0
3.0
7.0
9.0
Temperature (C°)

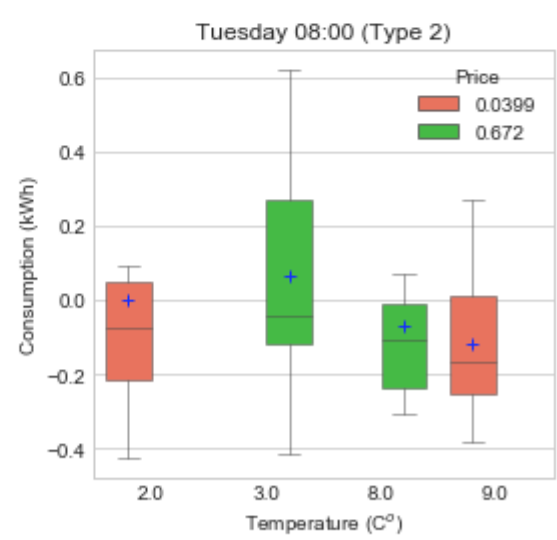

Tuesday 07:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
8.0
9.0
Temperature (C°)

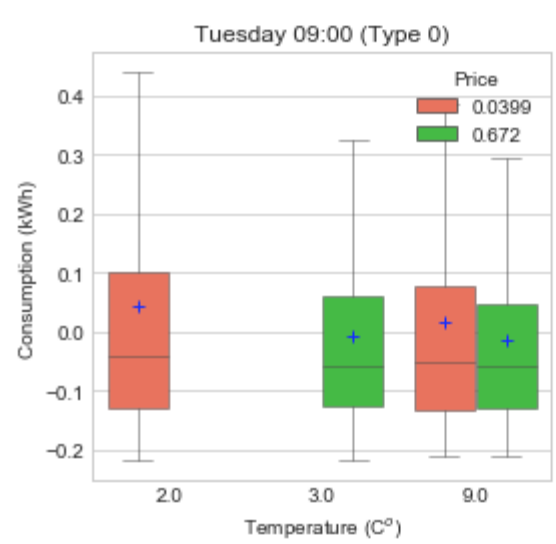

Tuesday 07:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
3.0
8.0
9.0
Temperature (C°)

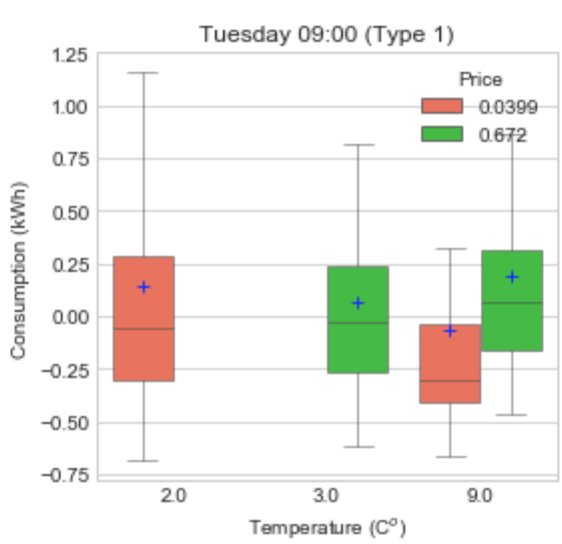

Tuesday 07:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.1
0.0
−0.1
−0.2
−0.3
2.0
3.0
8.0
9.0
Temperature (C°)

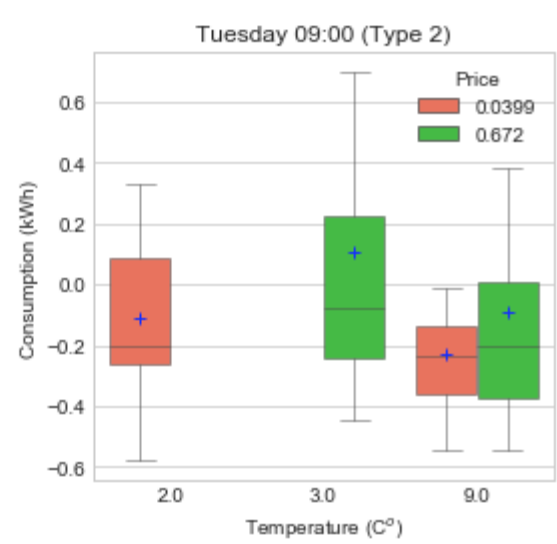

Tuesday 08:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
8.0
9.0
Temperature (C°)

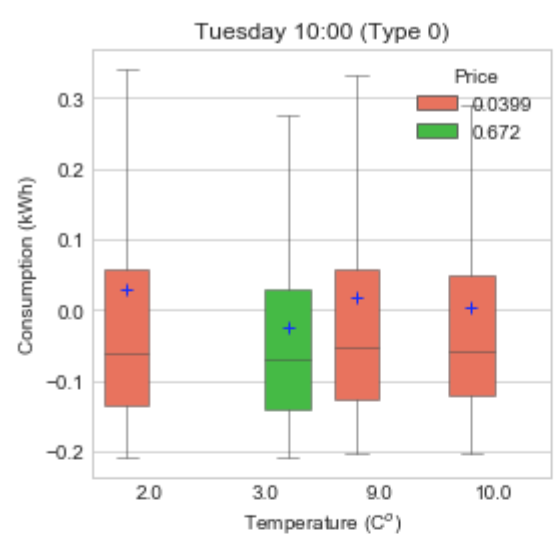

Tuesday 08:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
3.0
8.0
9.0
Temperature (C°)

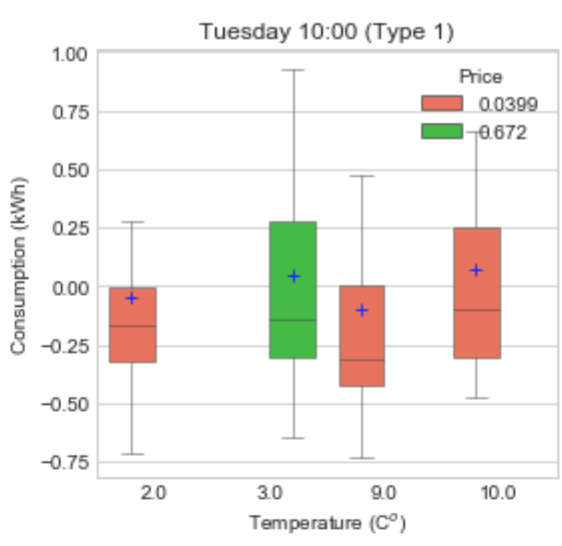

Tuesday 08:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
3.0
8.0
9.0
Temperature (C°)

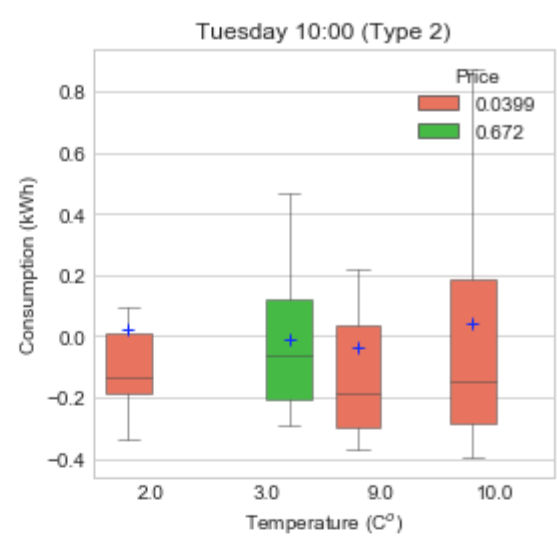

Tuesday 09:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
9.0
Temperature (C°)


Tuesday 09:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
3.0
9.0
Temperature (C°)

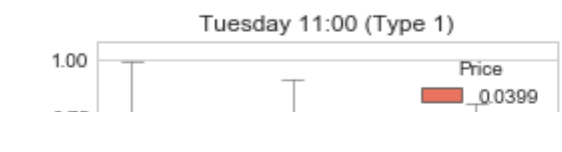

Tuesday 09:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
3.0
9.0
Temperature (C°)


Tuesday 10:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
9.0
10.0
Temperature (C°)

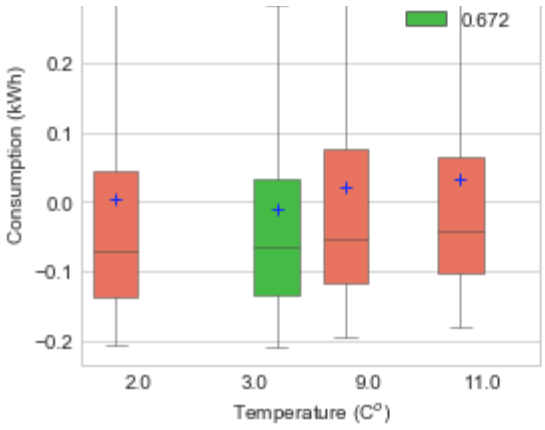

Tuesday 10:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
3.0
9.0
10.0
Temperature (C°)

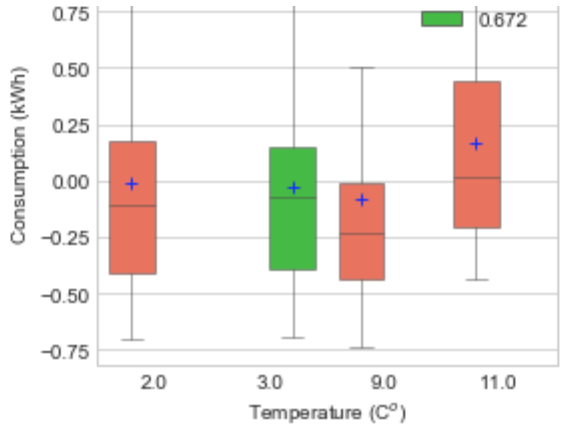

Tuesday 10:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
3.0
9.0
10.0
Temperature (C°)

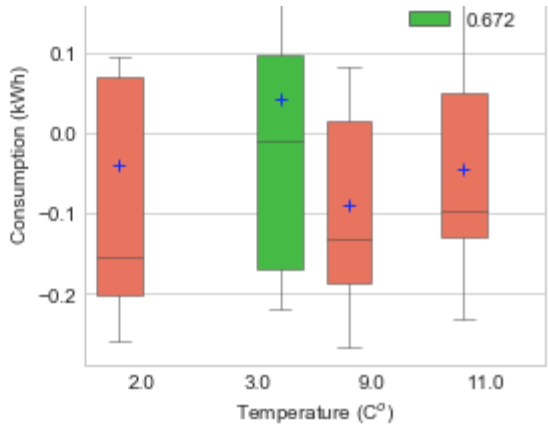

Tuesday 11:00 (Type 0)
Price
0.0399
0.3

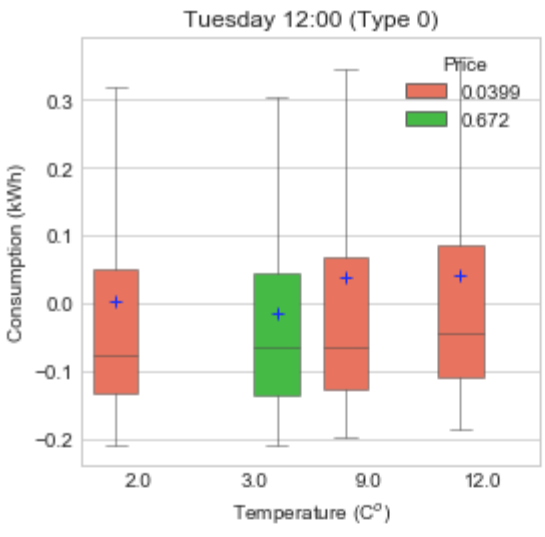

Tuesday 11:00 (Type 1)
Price
0.0399
1.00

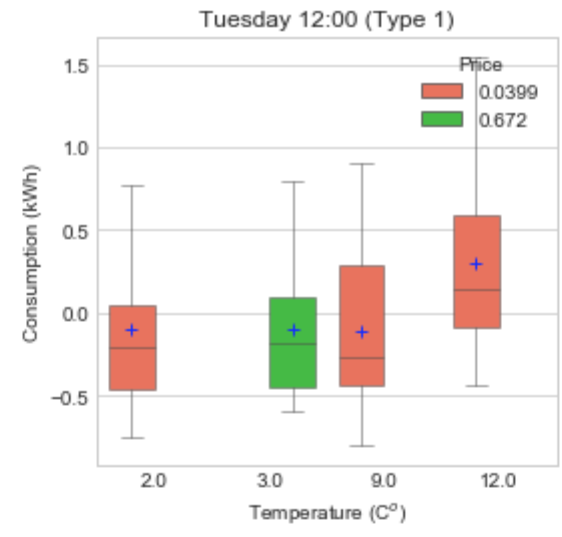

Tuesday 11:00 (Type 2)
Price
0.0399
0.2

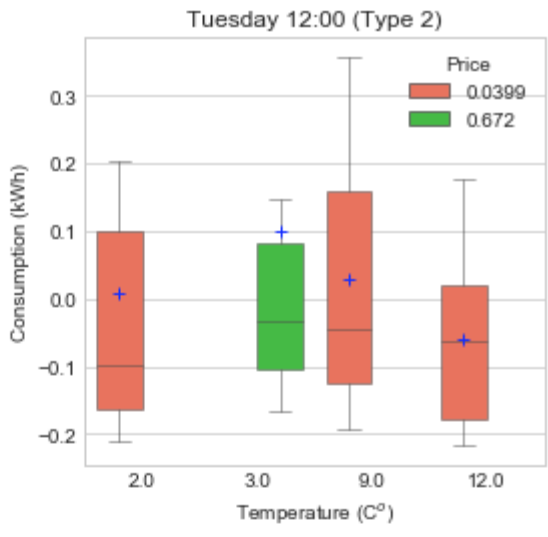

0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
11.0

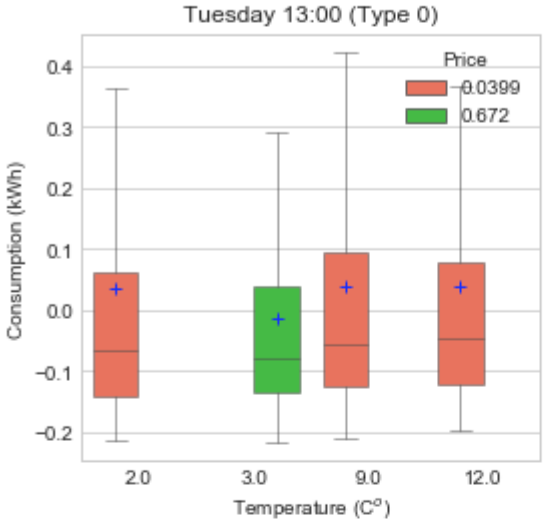

0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
11.0

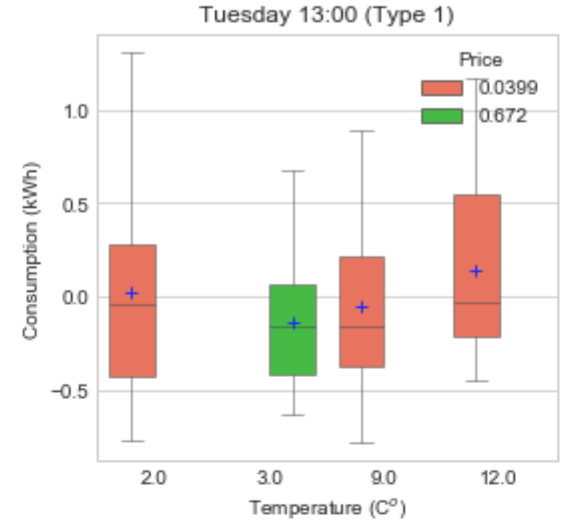

0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
11.0

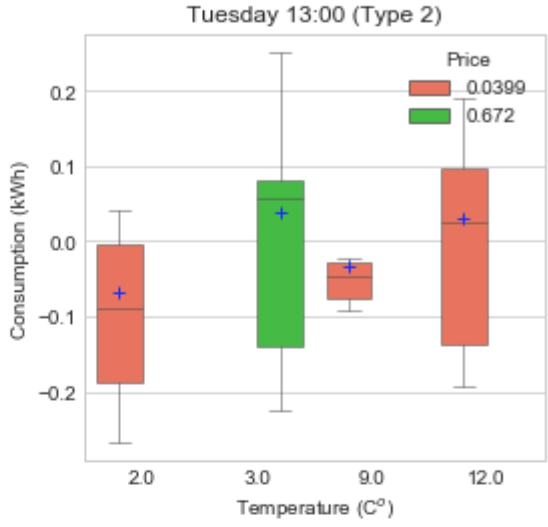

Tuesday 12:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

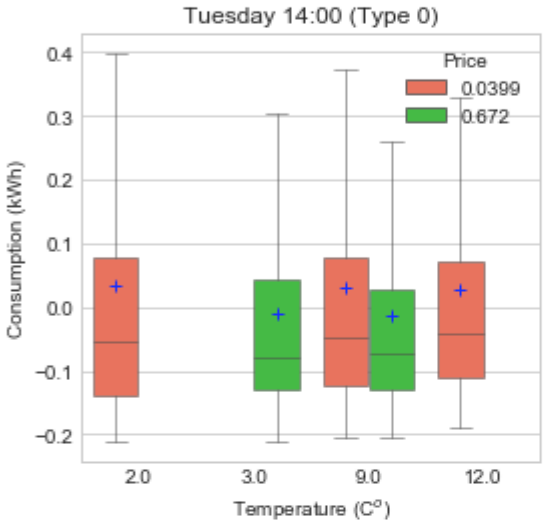

Tuesday 12:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

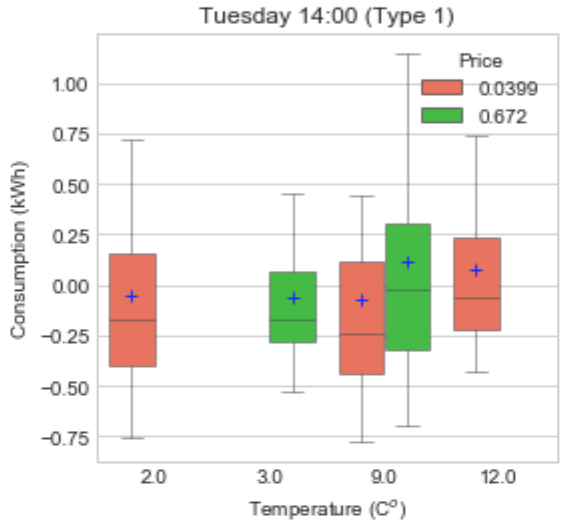

Tuesday 12:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

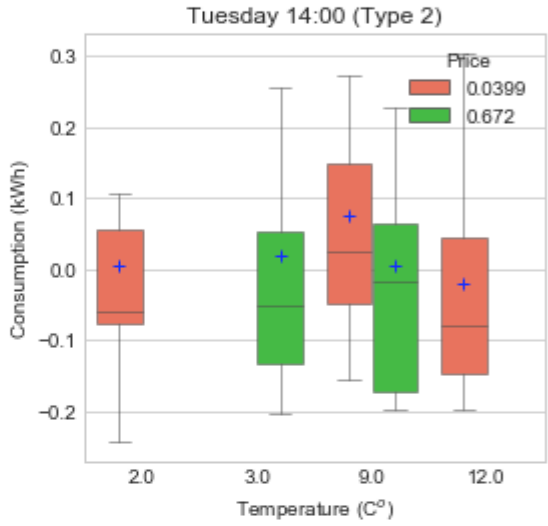

Tuesday 13:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

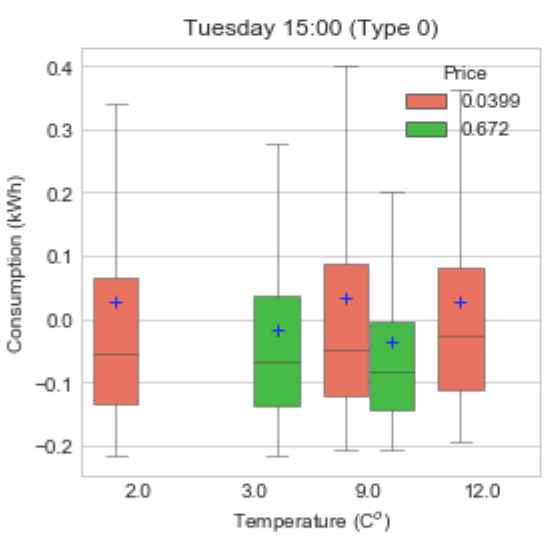

Tuesday 13:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

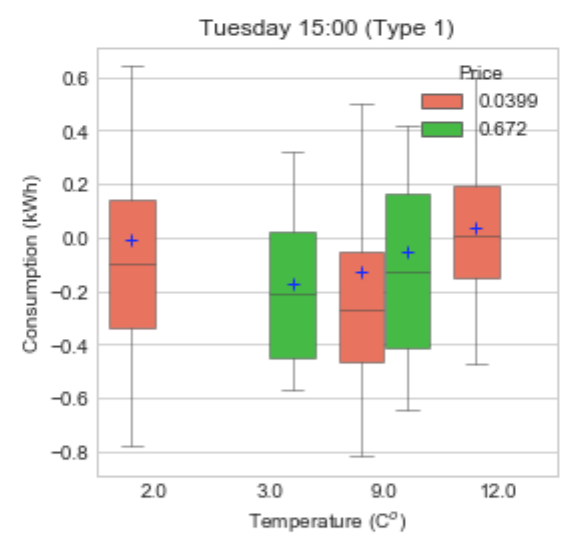

Tuesday 13:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

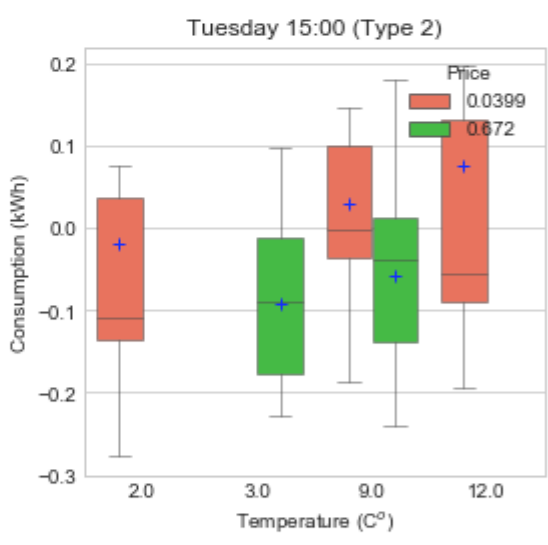

Tuesday 14:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

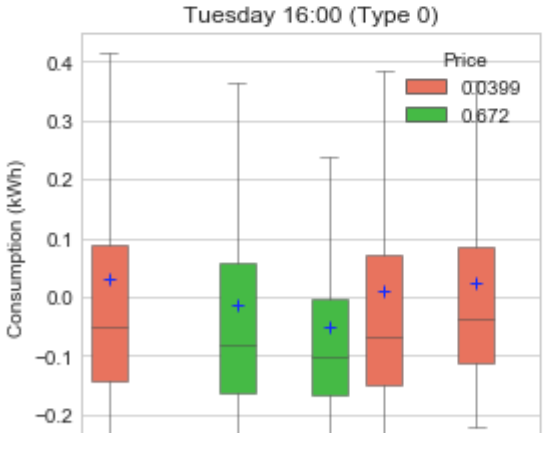

Tuesday 14:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

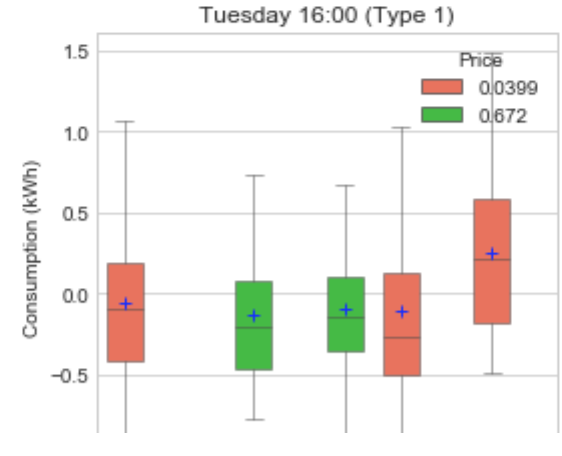

Tuesday 14:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0

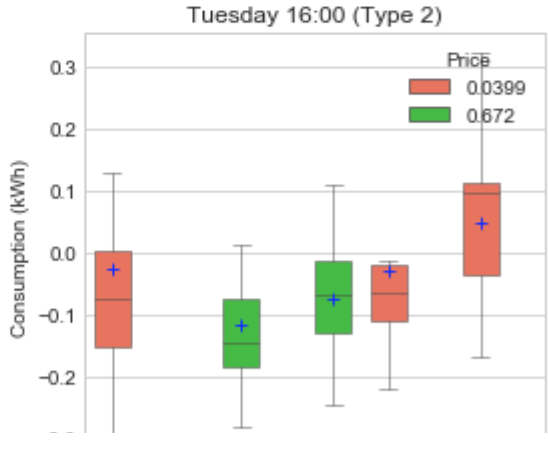

Tuesday 15:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0


Tuesday 15:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0


Tuesday 15:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
2.0
3.0
9.0
12.0


Tuesday 16:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)

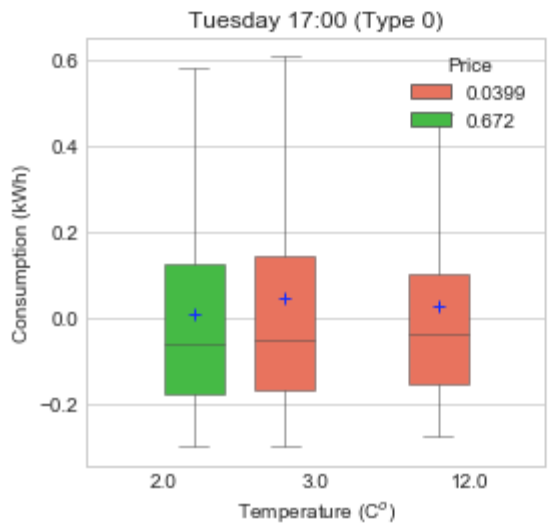

Tuesday 16:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)

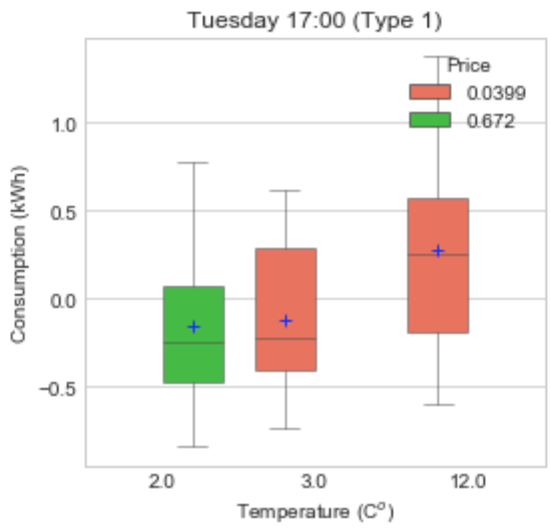

Tuesday 16:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)

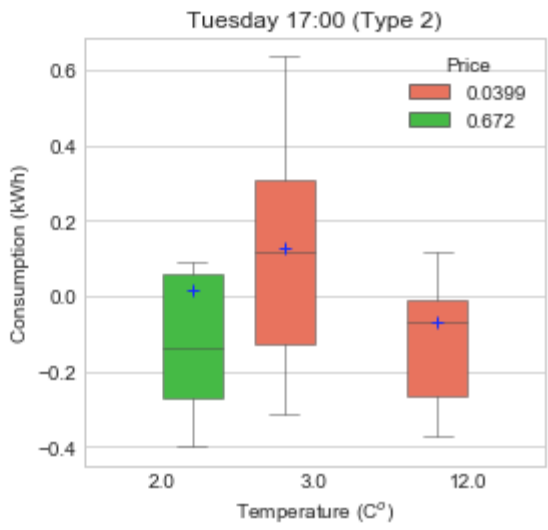

2.0
3.0
7.0
9.0
12.0
Temperature (C°)

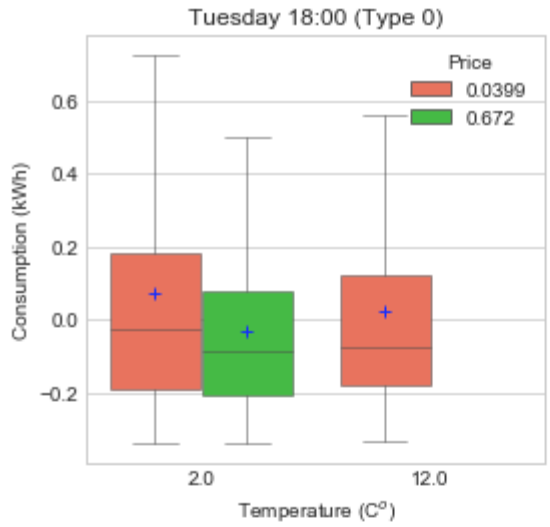

−1.0
2.0
3.0
7.0
9.0
12.0
Temperature (C°)

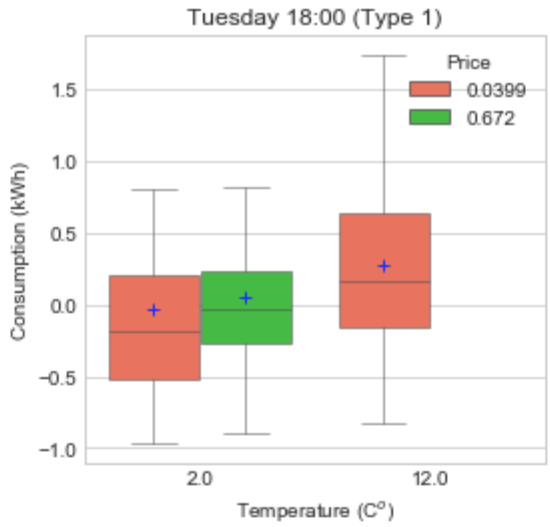

−0.3
2.0
3.0
7.0
9.0
12.0
Temperature (C°)

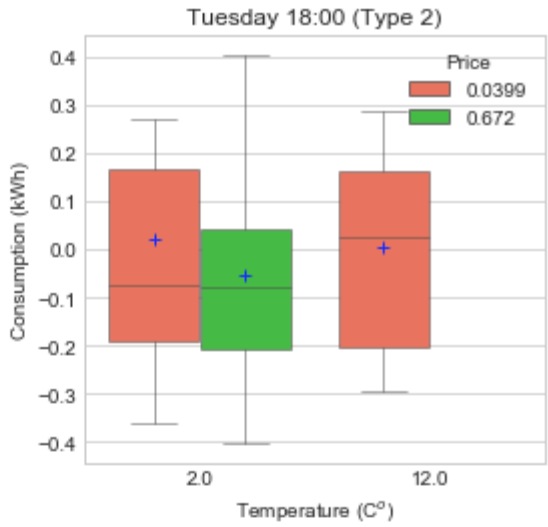

Tuesday 17:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
2.0
3.0
12.0
Temperature (C°)

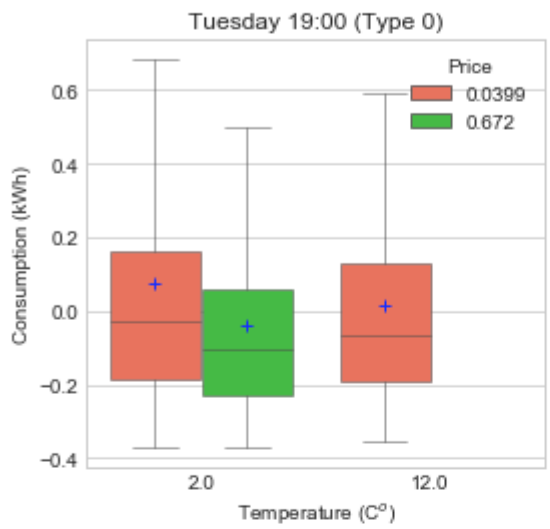

Tuesday 17:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
2.0
3.0
12.0
Temperature (C°)

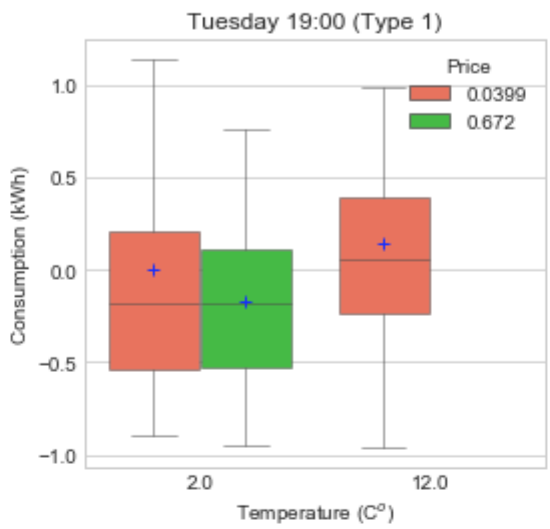

Tuesday 17:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
3.0
12.0
Temperature (C°)

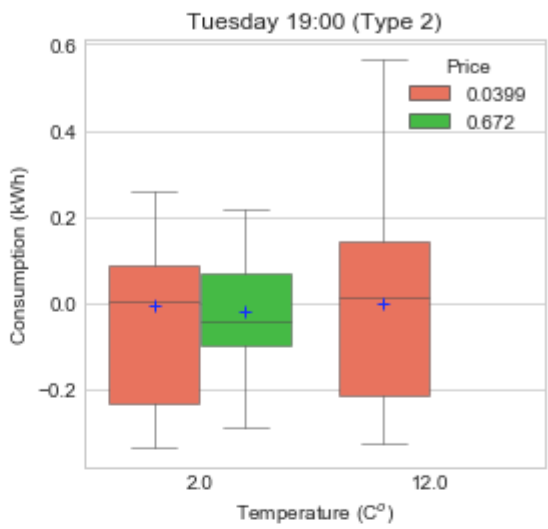

Tuesday 18:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
2.0
12.0
Temperature (C°)

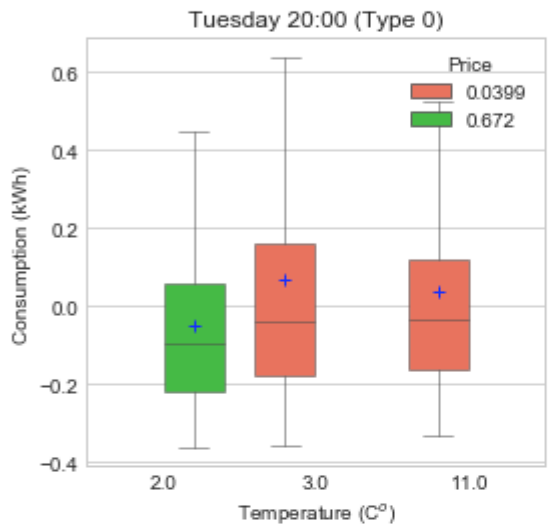

Tuesday 18:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
2.0
12.0
Temperature (C°)

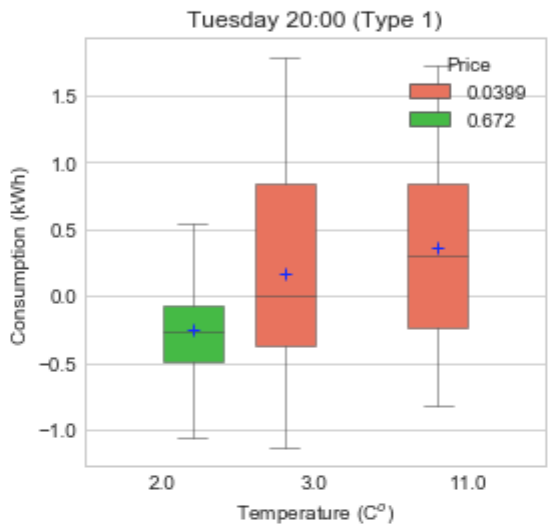

Tuesday 18:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
2.0
12.0
Temperature (C°)

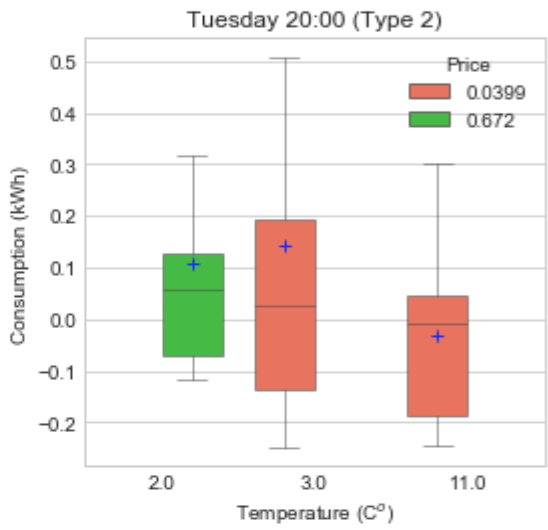

Tuesday 19:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
12.0
Temperature (C°)

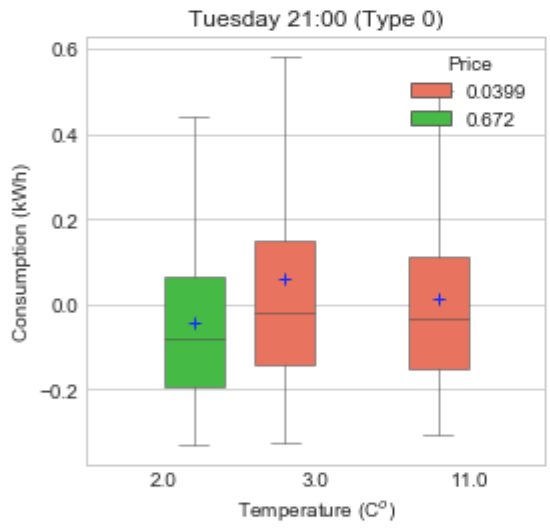

Tuesday 19:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
12.0
Temperature (C°)

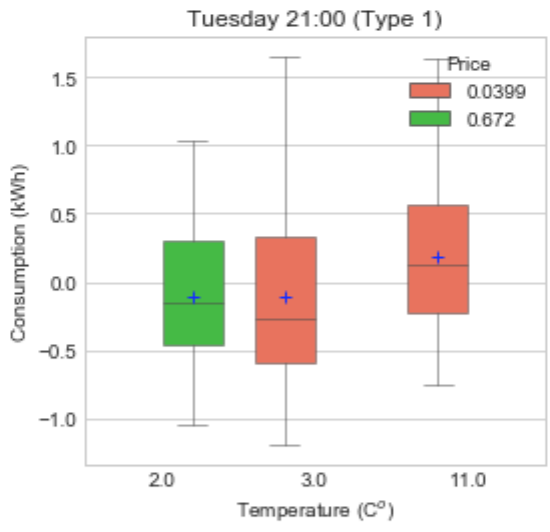

Tuesday 19:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
2.0
12.0
Temperature (C°)

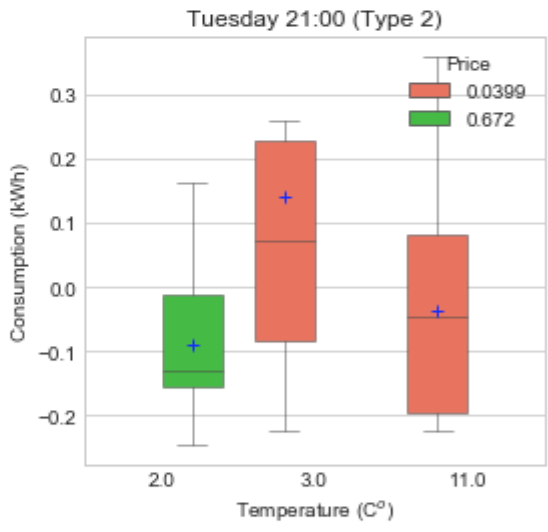

Tuesday 20:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
3.0
11.0
Temperature (C°)

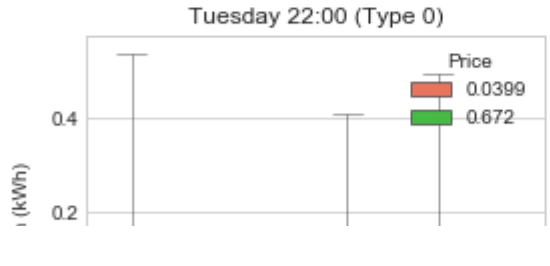

Tuesday 20:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
2.0
3.0
11.0
Temperature (C°)

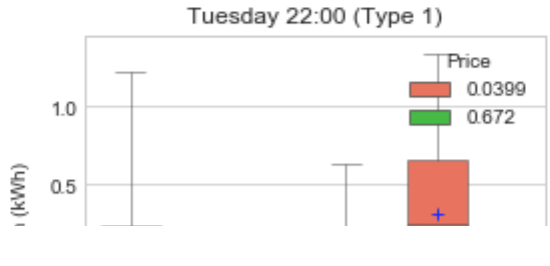

Tuesday 20:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
11.0
Temperature (C°)

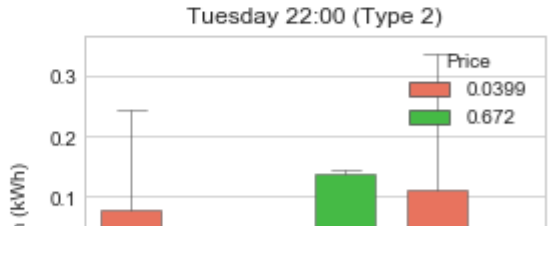

Tuesday 21:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
2.0
3.0
11.0
Temperature (C°)

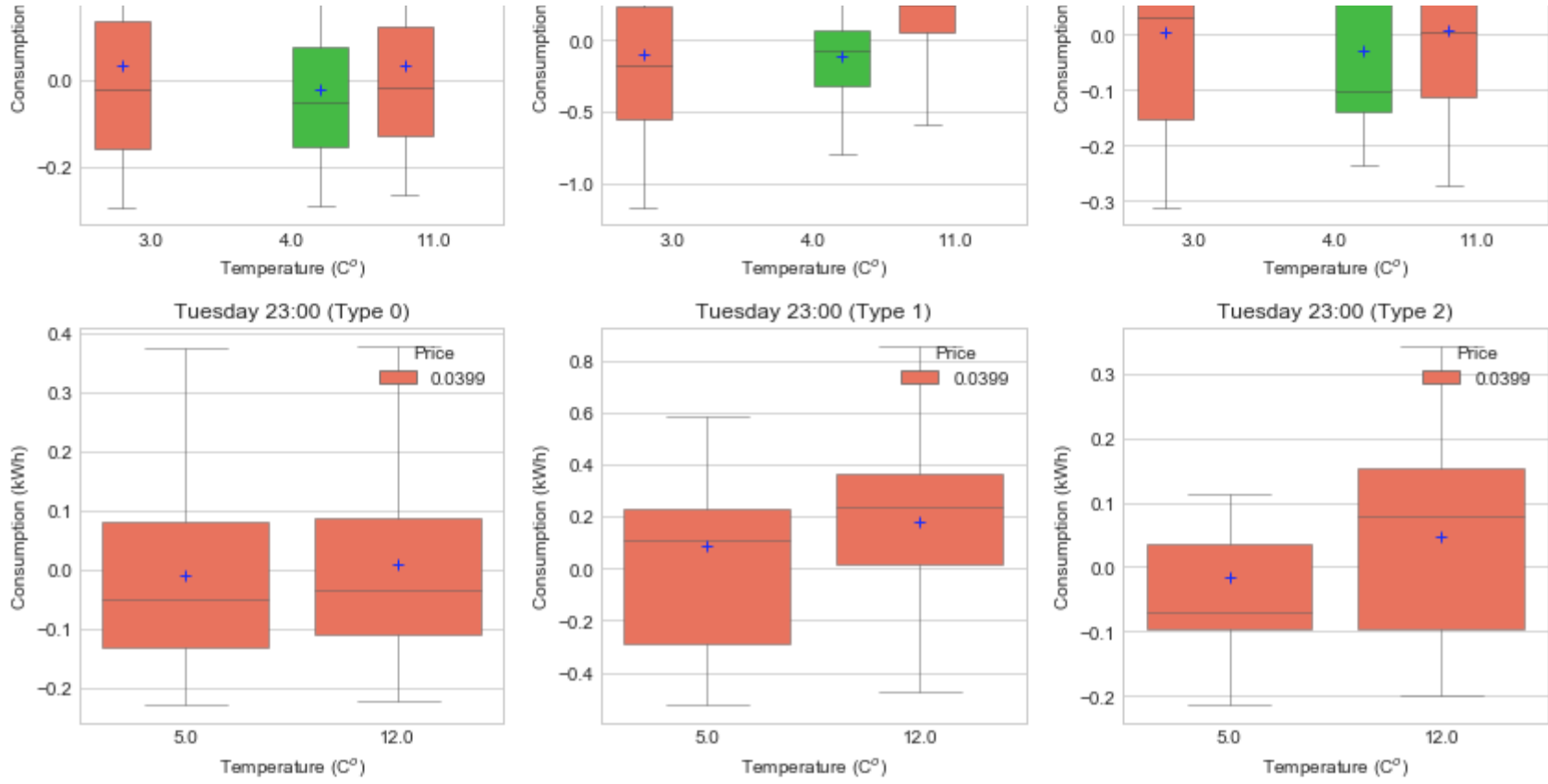

Tuesday 21:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
2.0
3.0
11.0
Temperature (C°)


Tuesday 21:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
11.0
Temperature (C°)


Tuesday 22:00 (Type 0)
Price
0.0399
0.672
0.4
0.2

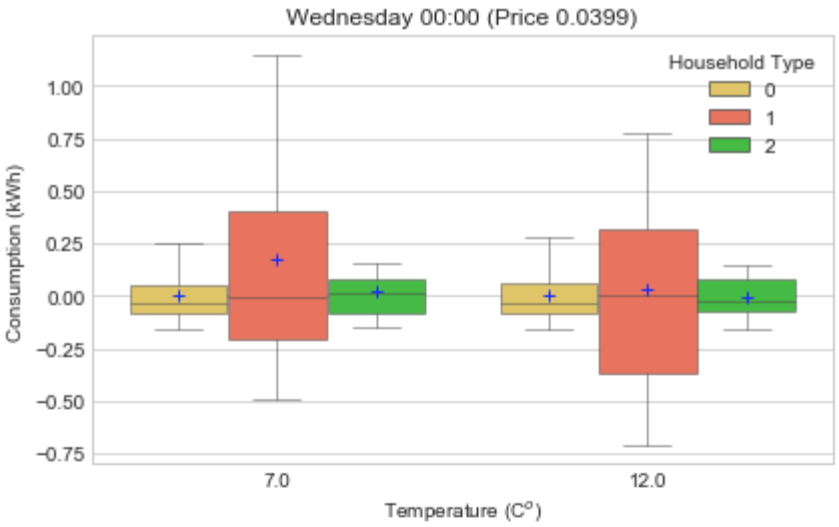

Tuesday 22:00 (Type 1)
Price
0.0399
0.672
1.0
0.5

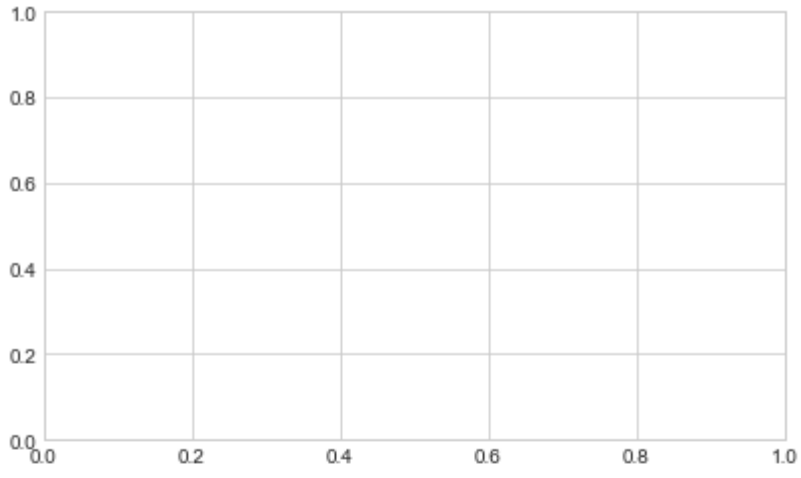

Tuesday 22:00 (Type 2)
Price
0.0399
0.672
0.3
0.2
0.1

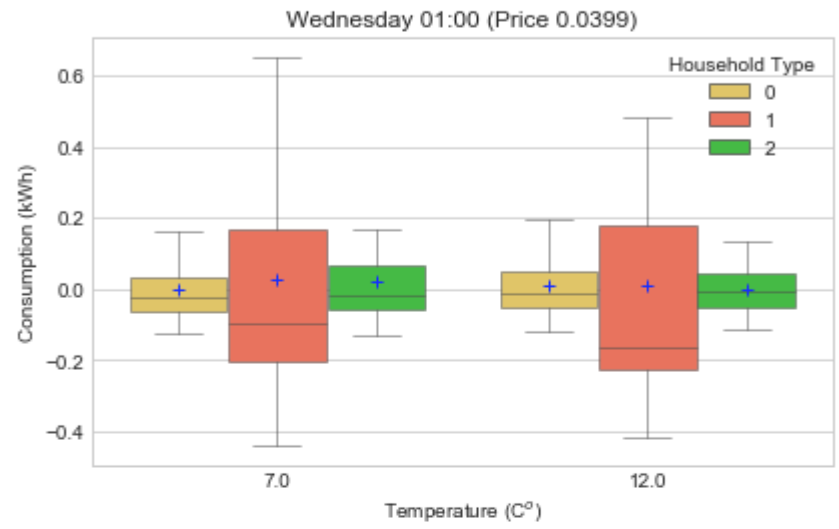


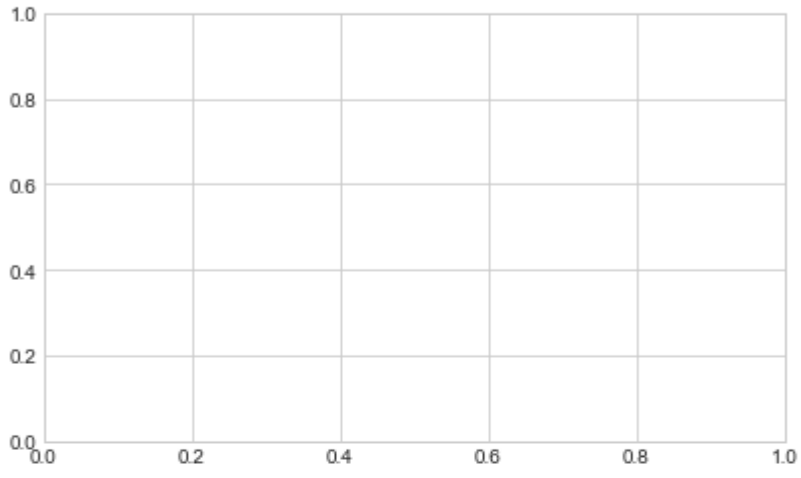
See type0 at 6pm and 7pm mean values. We find type 0 households are actually good candidates for demand response, since they have a right direction of energy usage change. But for type 1, their response sometimes fluctuates.

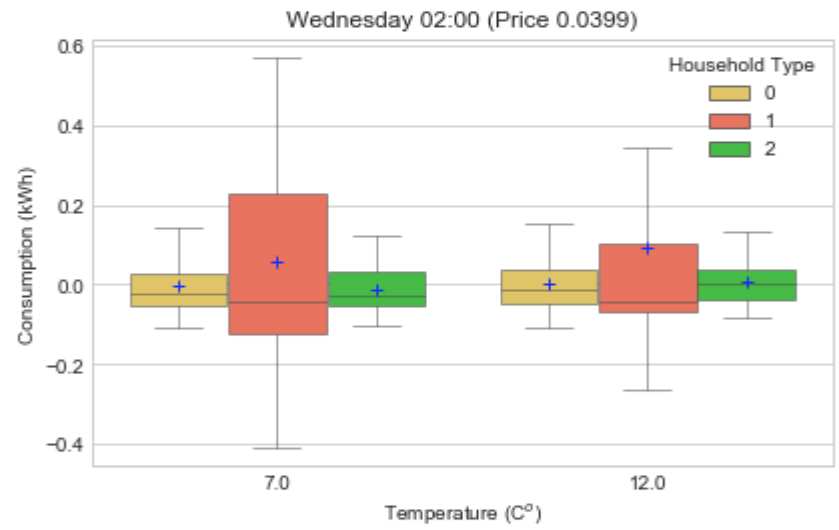
In [99]:
```
# Wednesday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 2 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

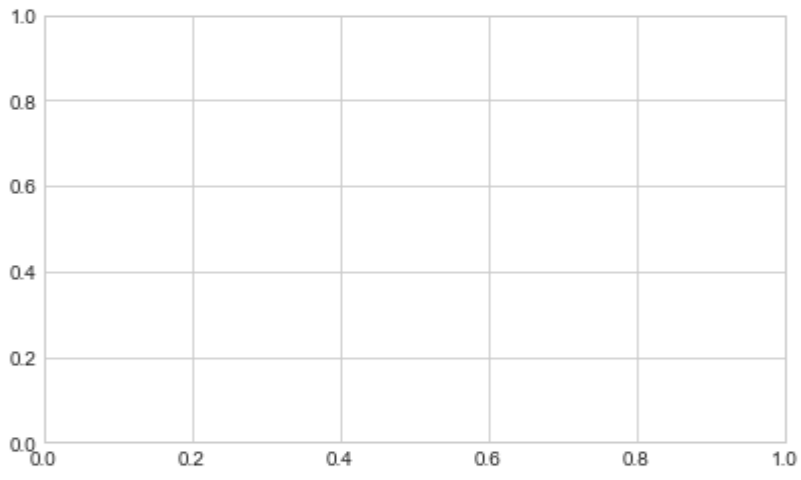

Wednesday 00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
7.0
12.0
Temperature (C°)

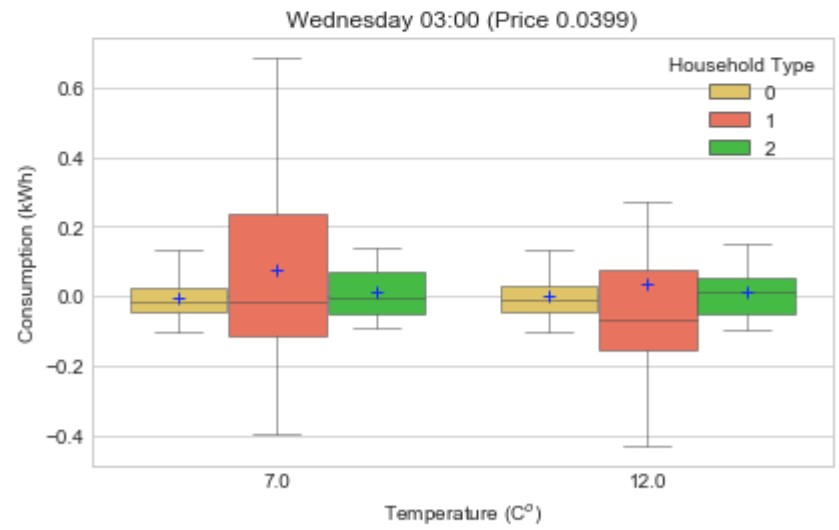

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

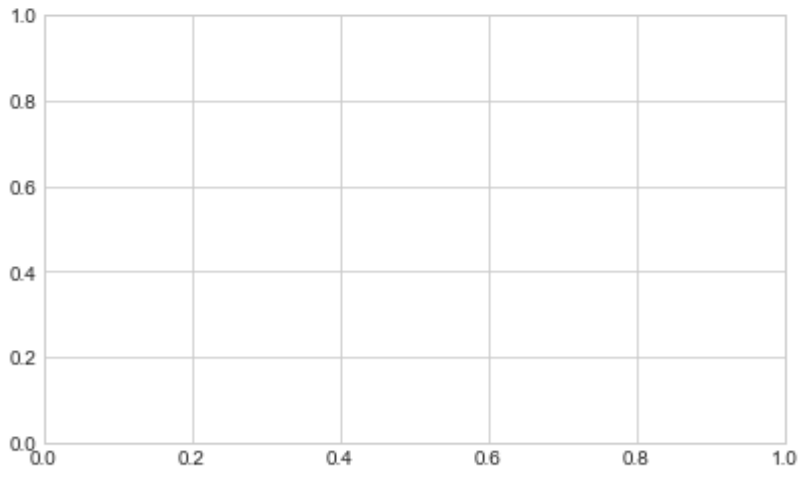

Wednesday 01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

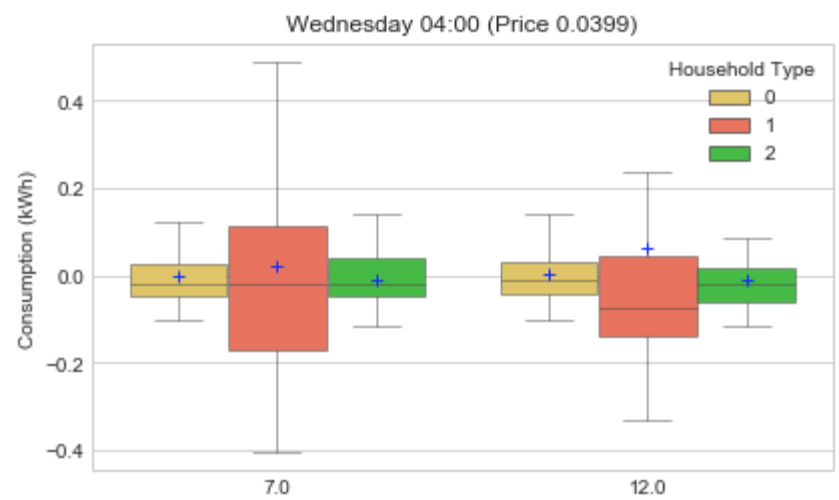

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

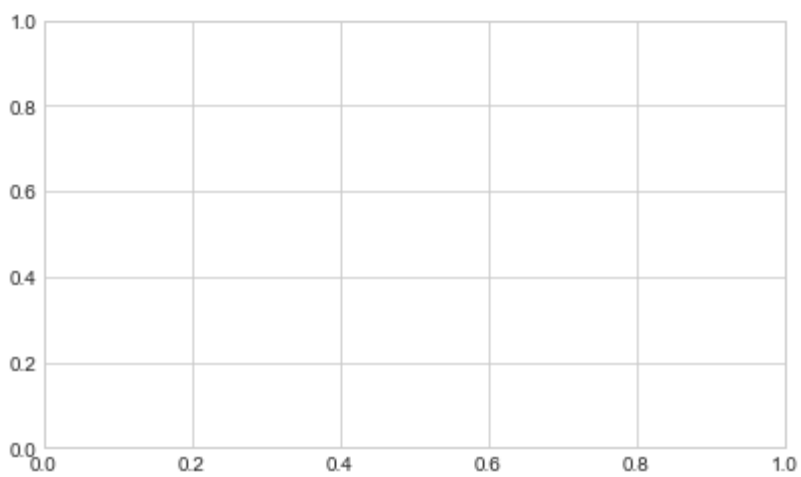

Wednesday 02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

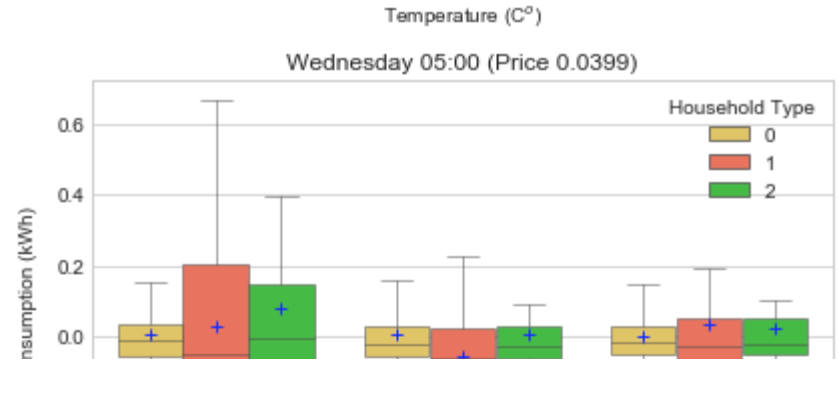

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

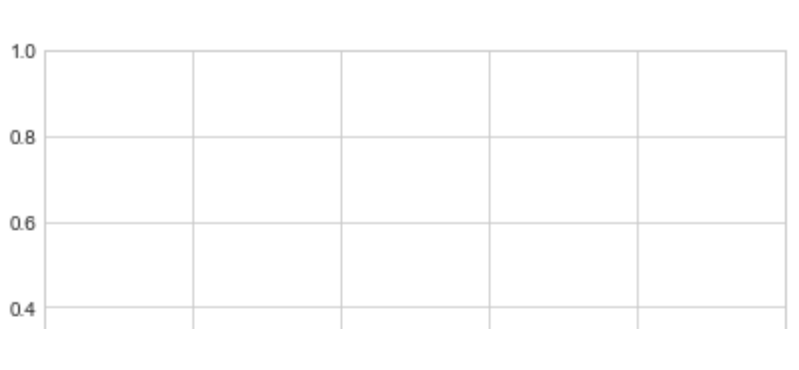

Wednesday 03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

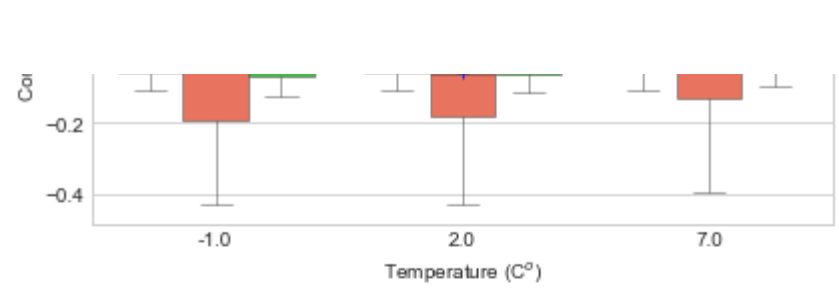

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

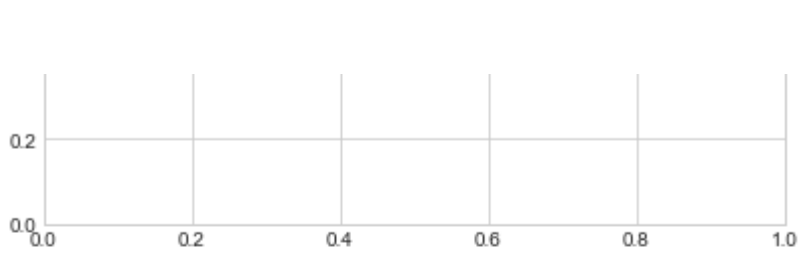

Wednesday 04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

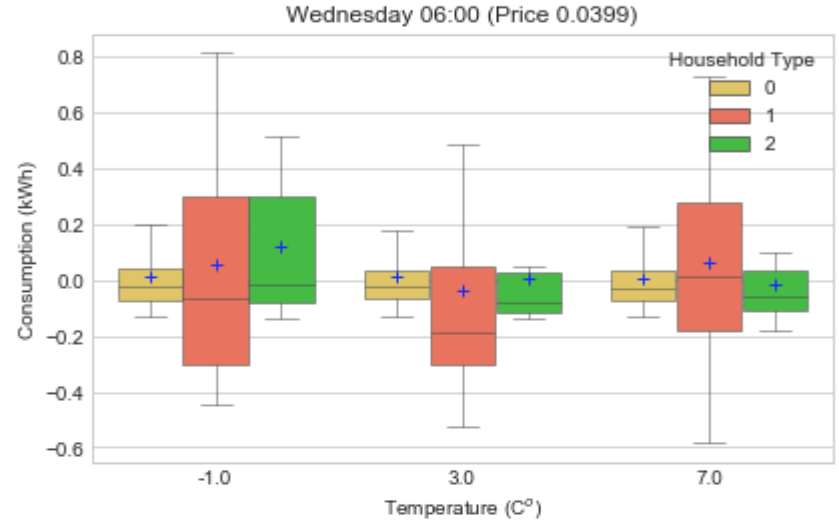

Wednesday 05:00 (Price 0.0399)
Household Type
0
1
2
0.6
0.4
0.2
0.0

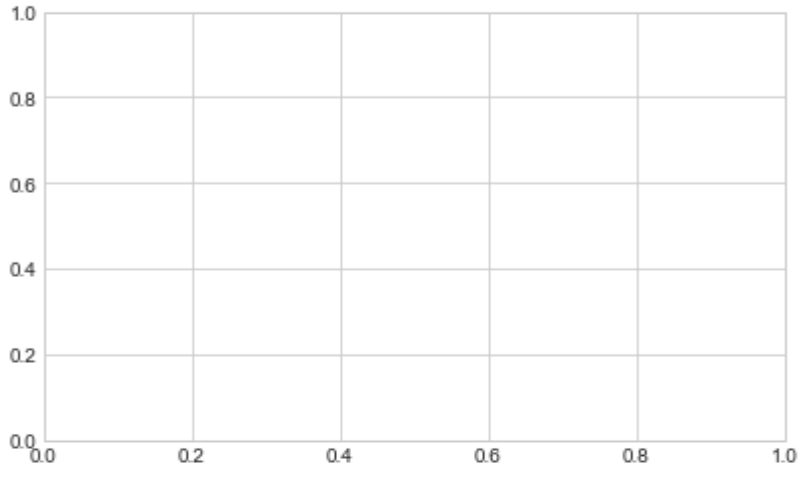

1.0
0.8
0.6
0.4



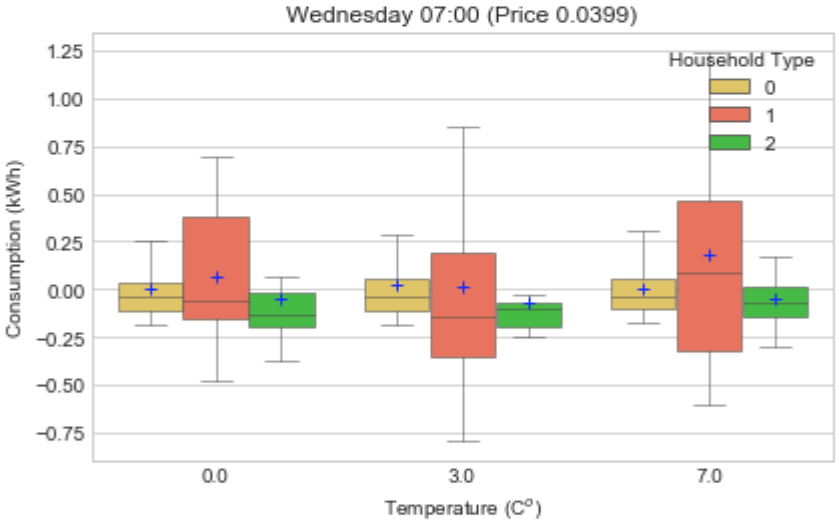


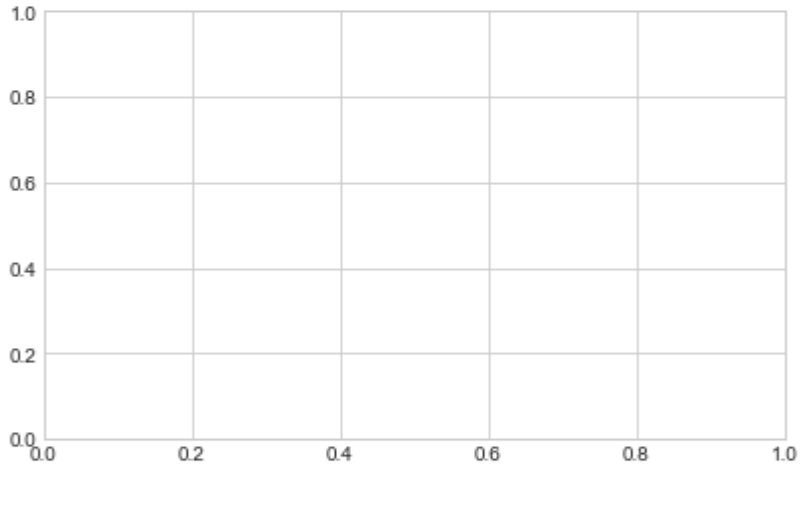
Wednesday 06:00 (Price 0.0399)



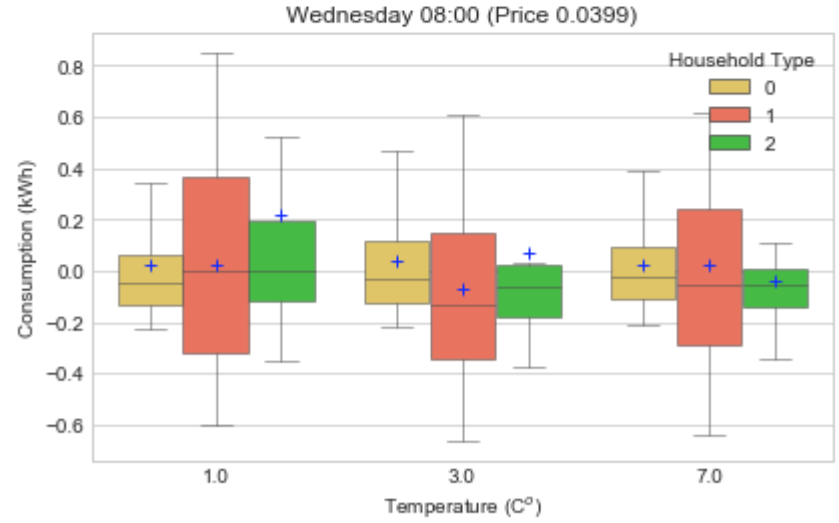


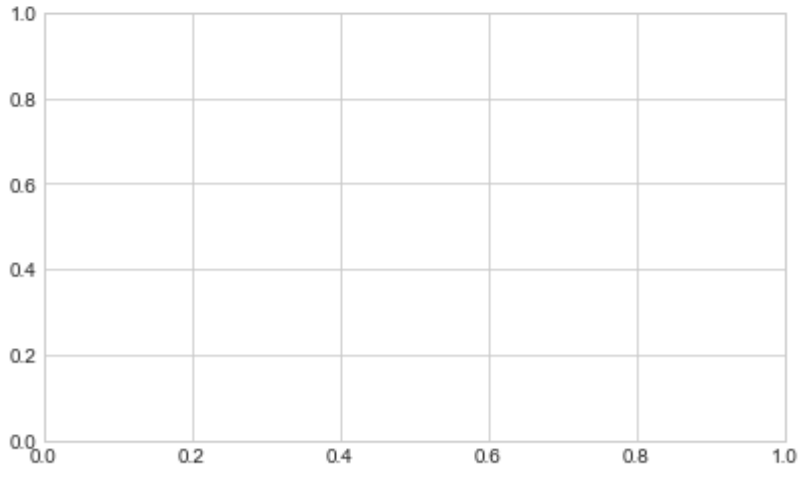
Wednesday 07:00 (Price 0.0399)



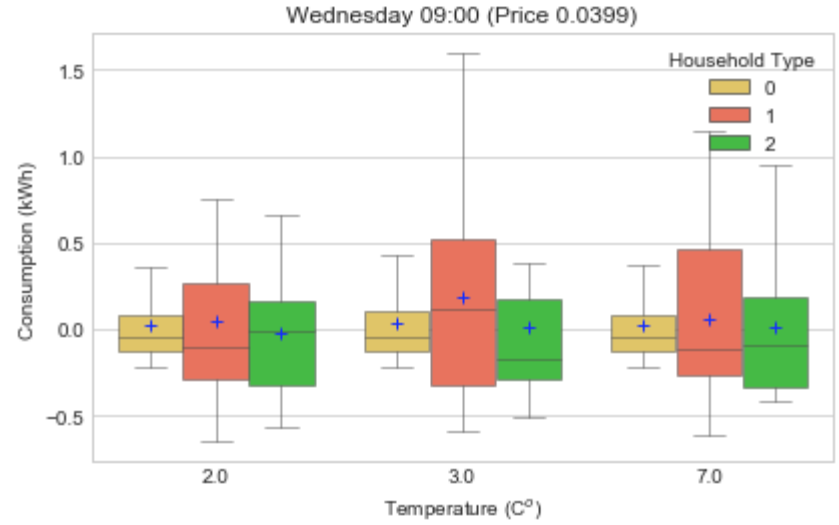


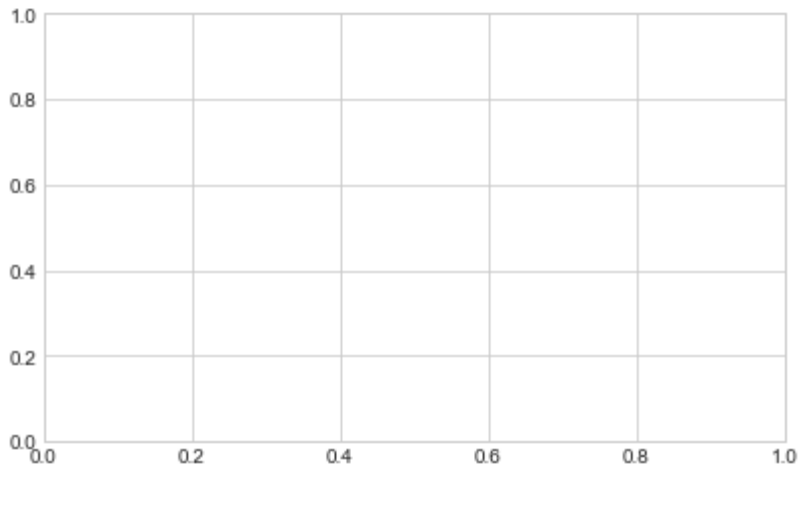
Wednesday 08:00 (Price 0.0399)



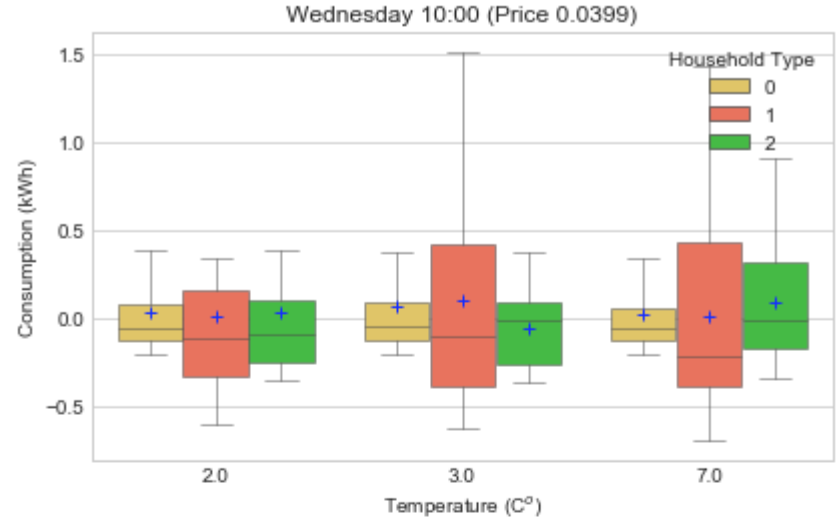


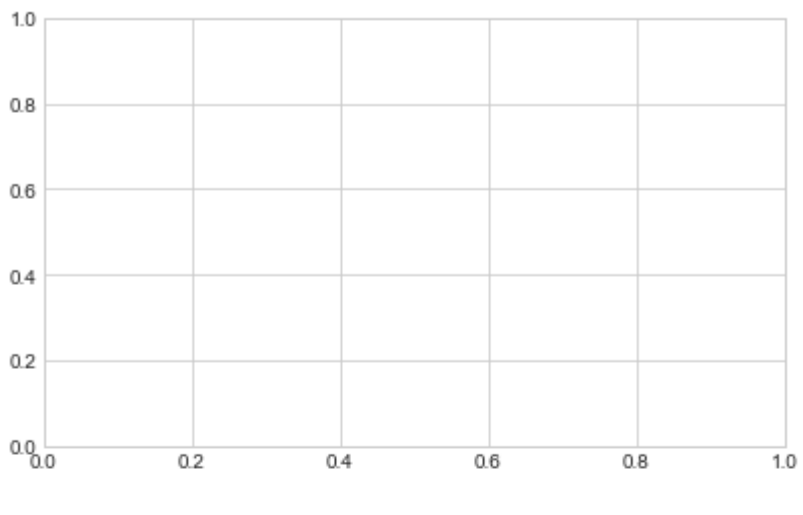
Wednesday 09:00 (Price 0.0399)



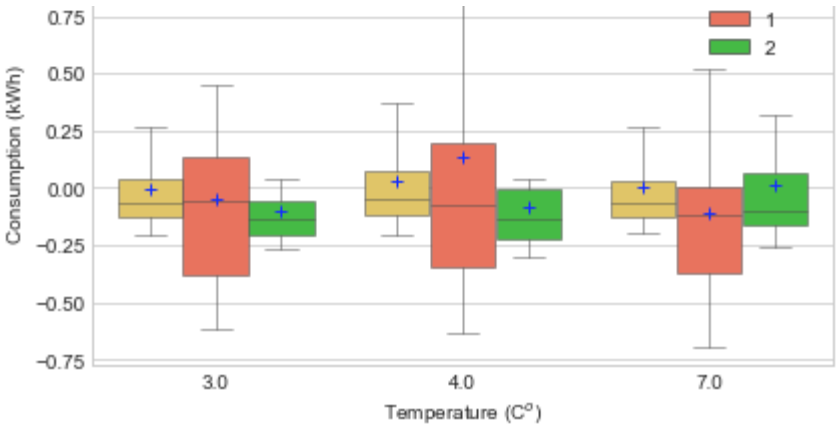


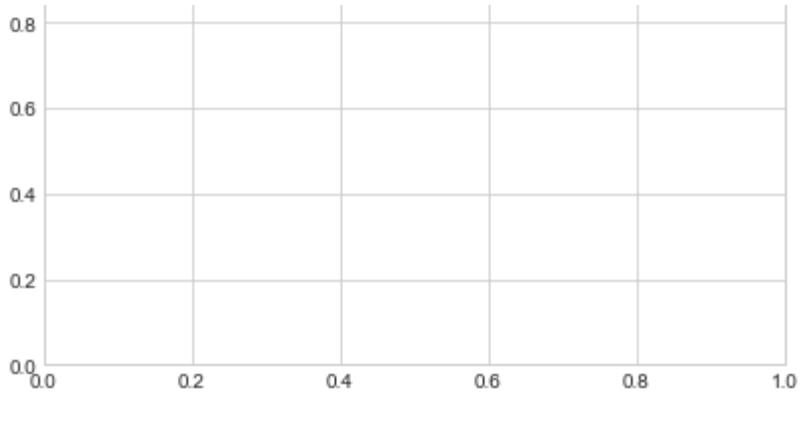
Wednesday 10:00 (Price 0.0399)

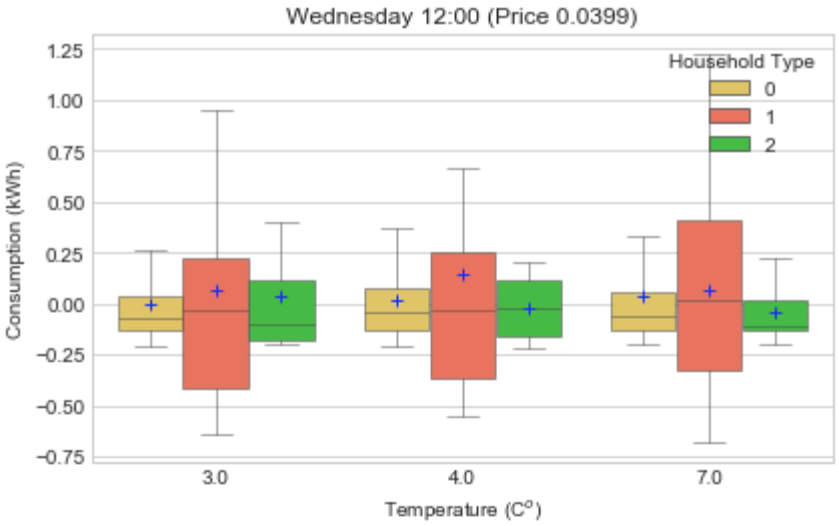


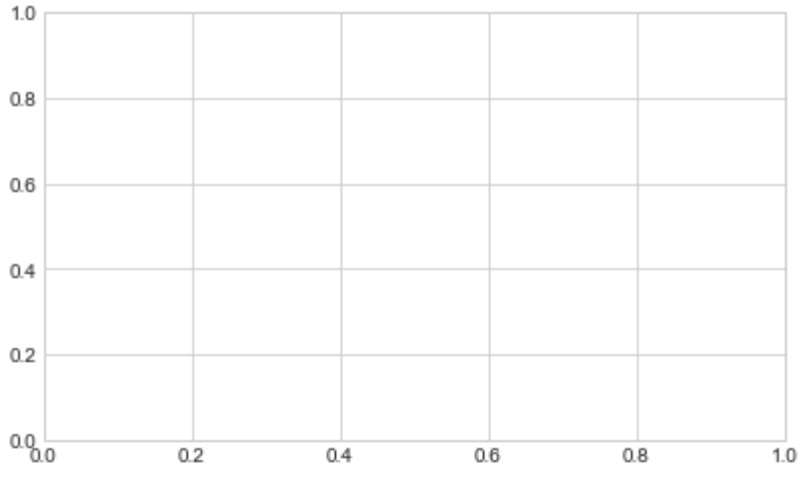


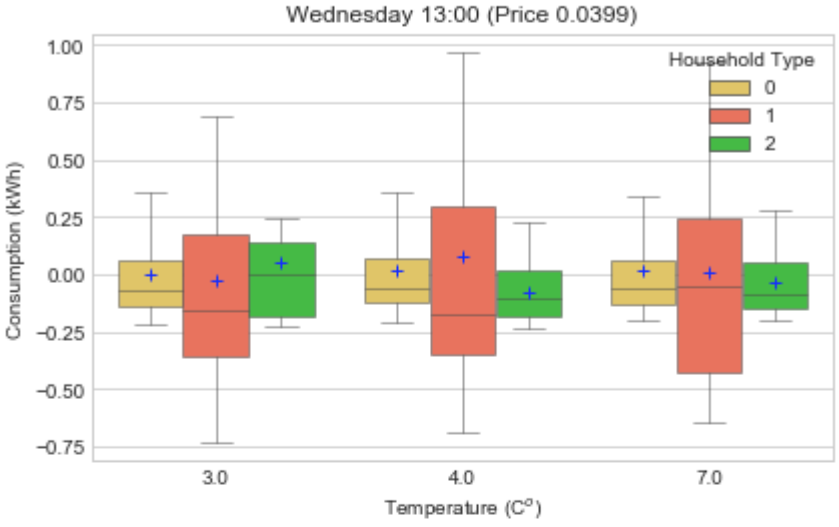
Wednesday 11:00 (Price 0.0399)

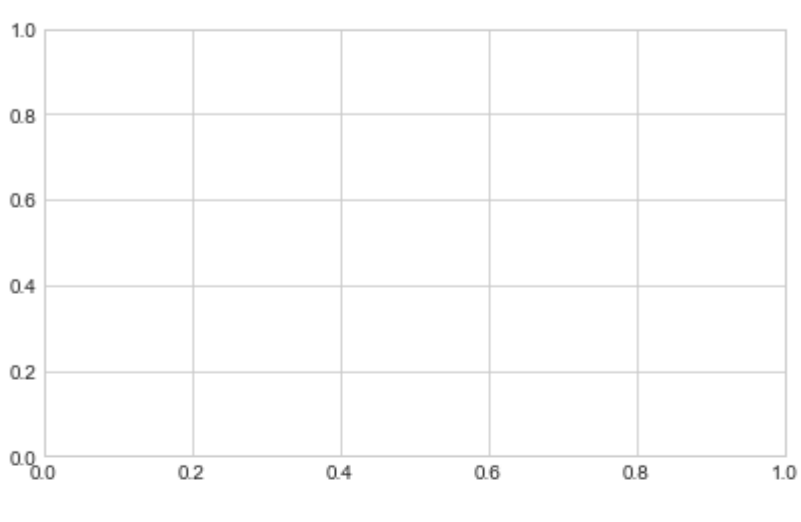

1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
4.0
7.0
Temperature (C°)

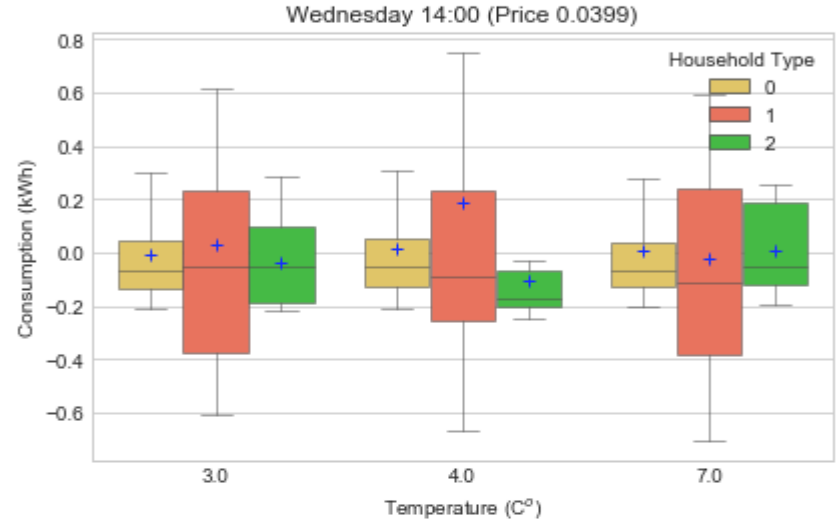

0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

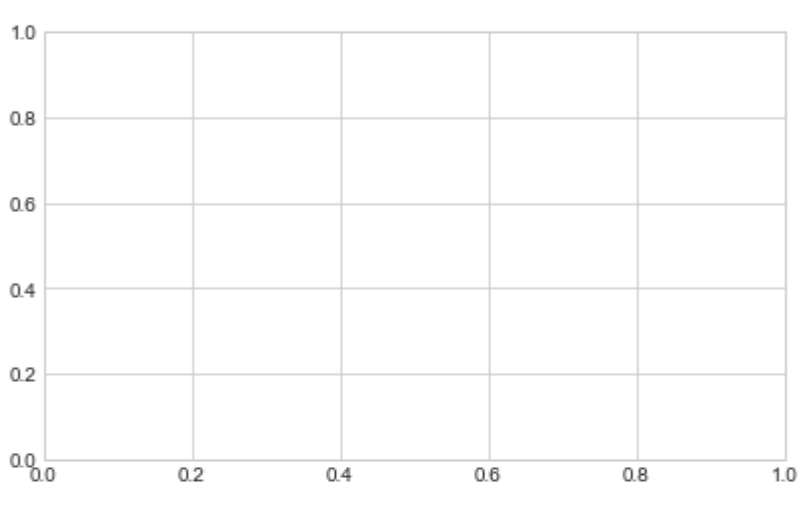

Wednesday 12:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
4.0
7.0
Temperature (C°)

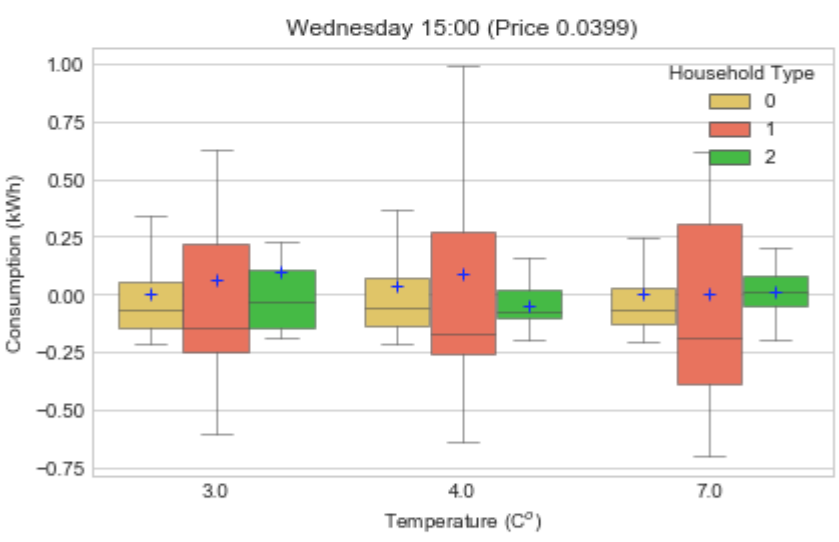

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

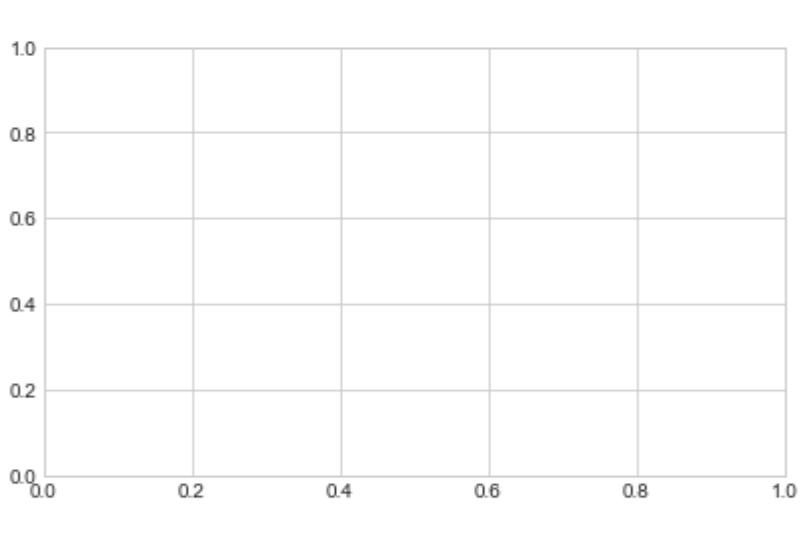

Wednesday 13:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
4.0
7.0
Temperature (C°)

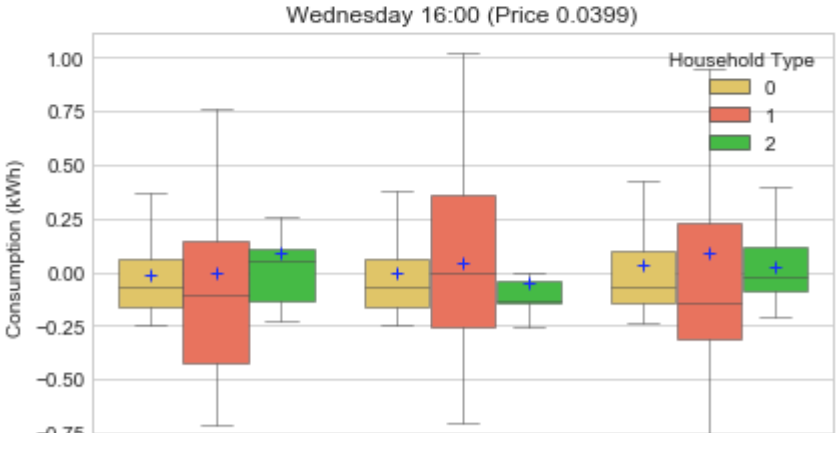

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

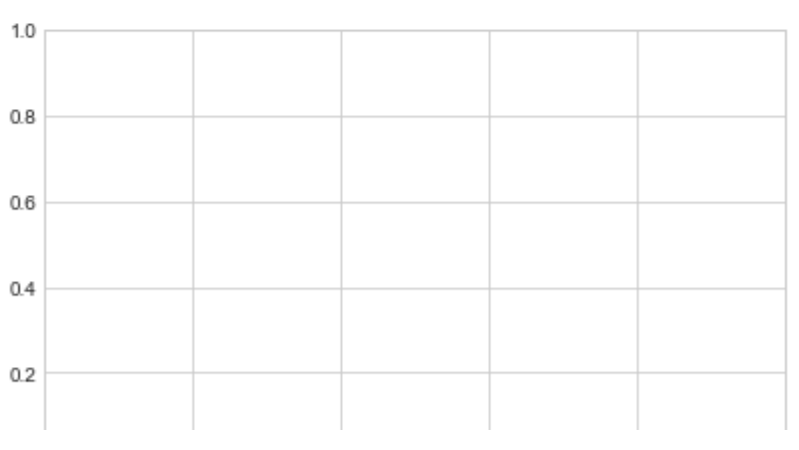

Wednesday 14:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
4.0
7.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Wednesday 15:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
4.0
7.0
Temperature (C°)

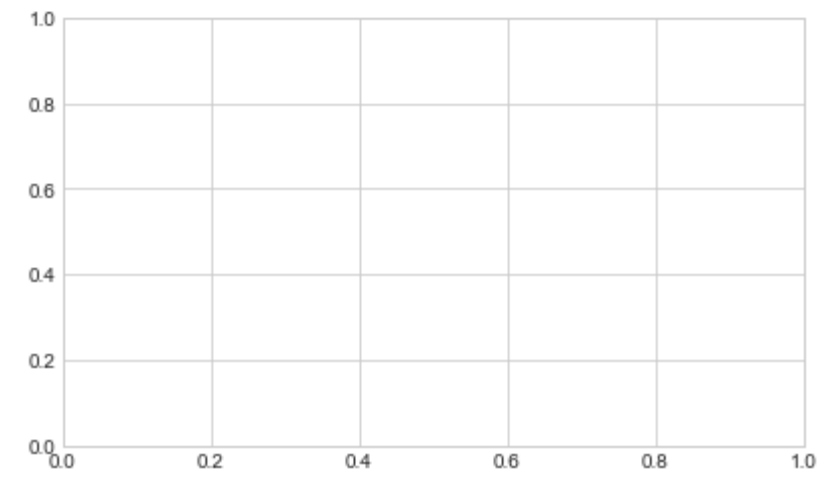

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

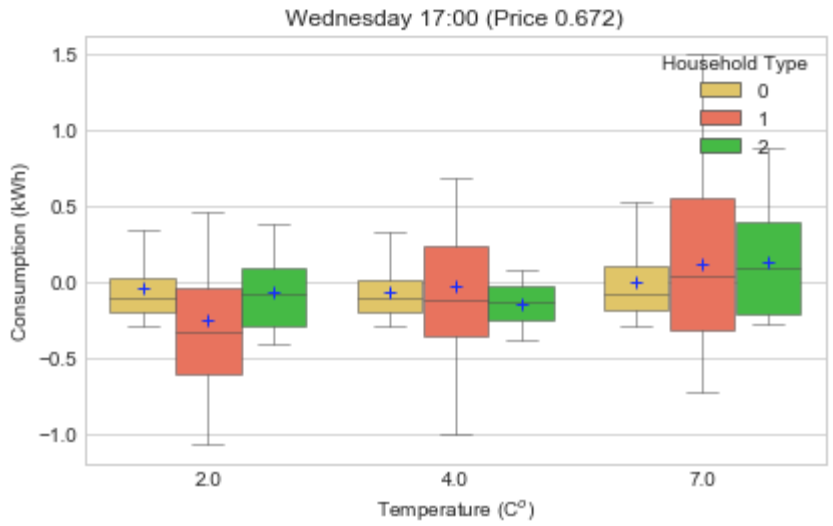

Wednesday 16:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50

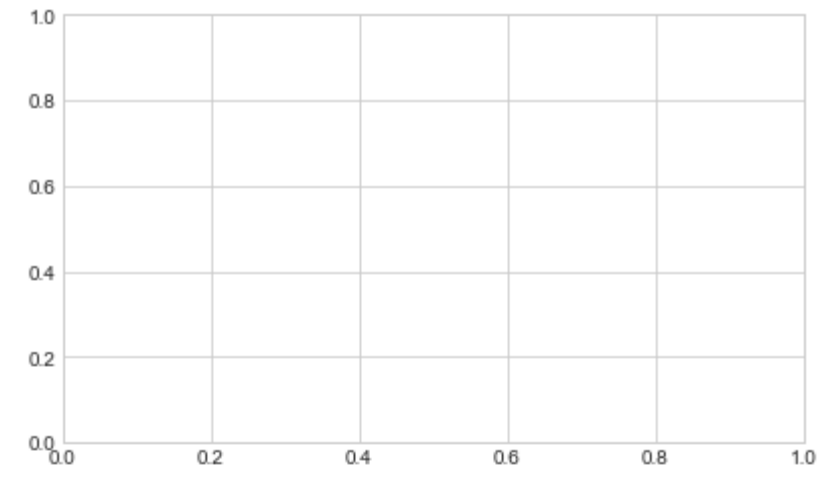

1.0
0.8
0.6
0.4
0.2

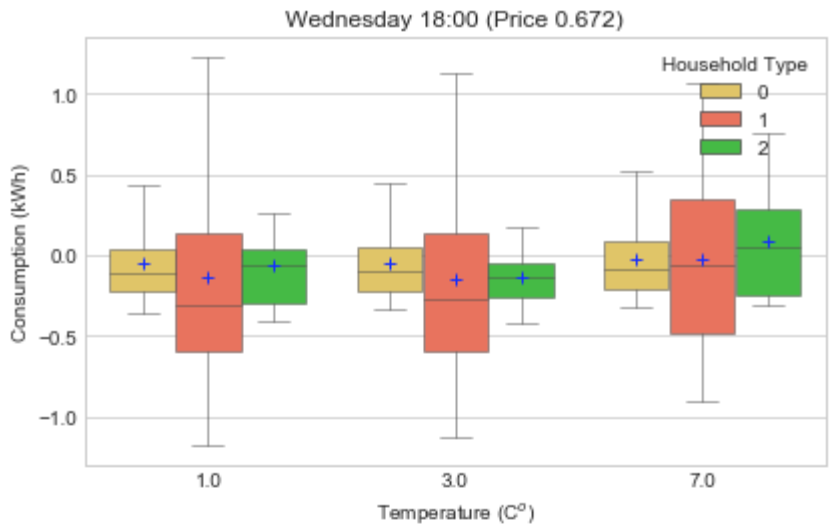

2.0
3.0
7.0
Temperature (C°)

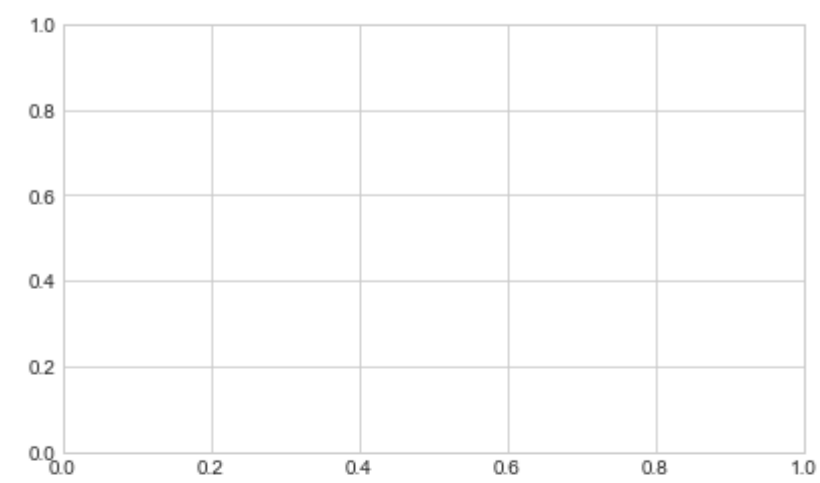

0.0
0.2
0.4
0.6
0.8
1.0

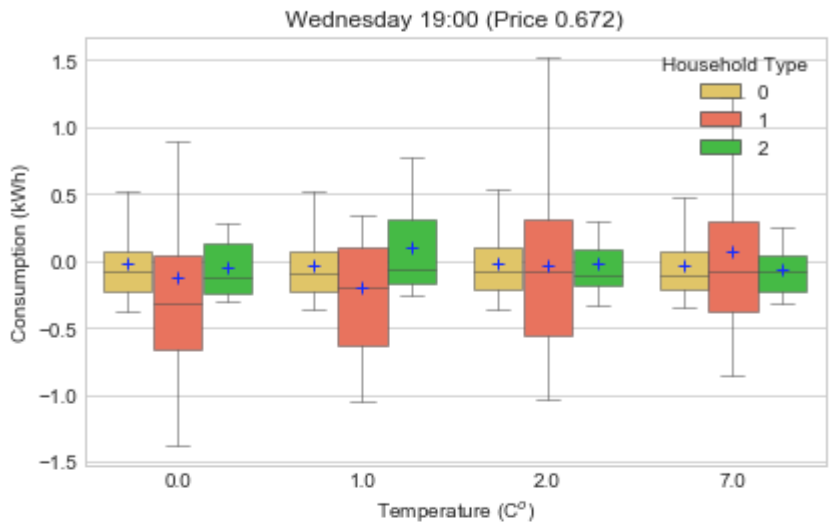

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

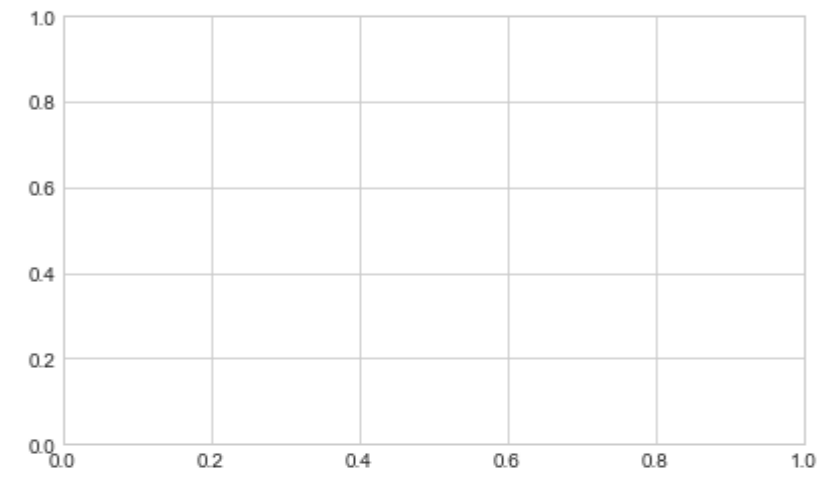

Wednesday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
2.0
4.0
7.0
Temperature (C°)

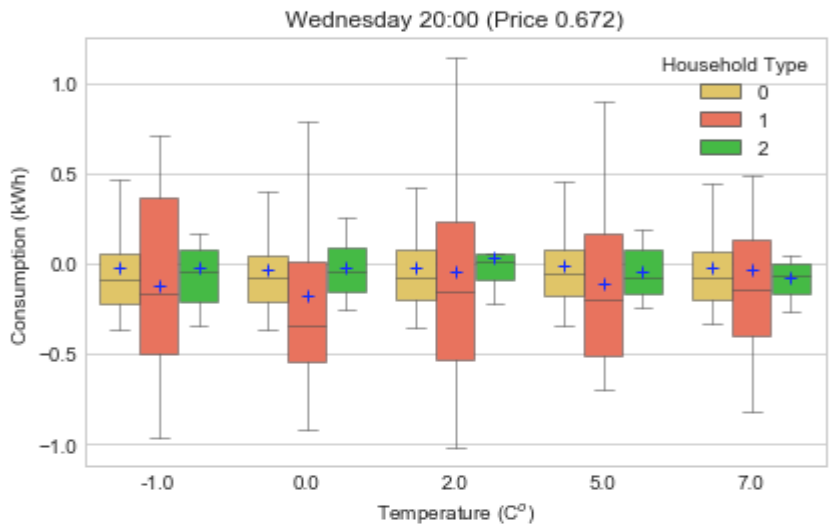

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

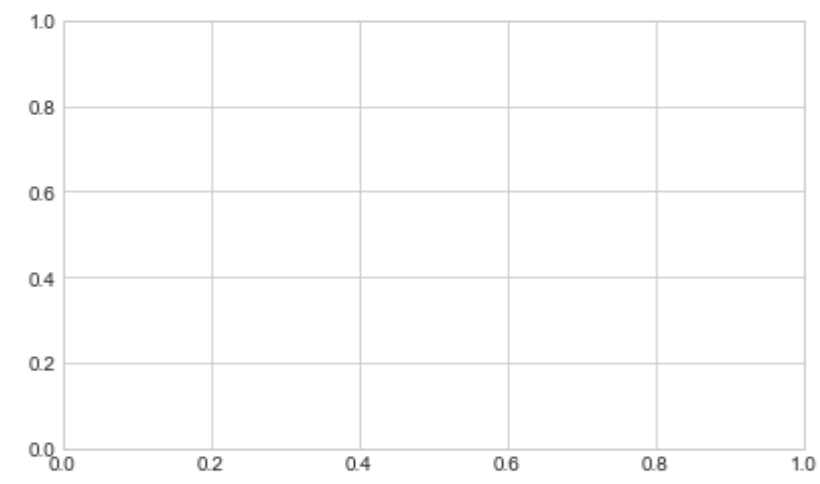

Wednesday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
1.0
3.0
7.0
Temperature (C°)

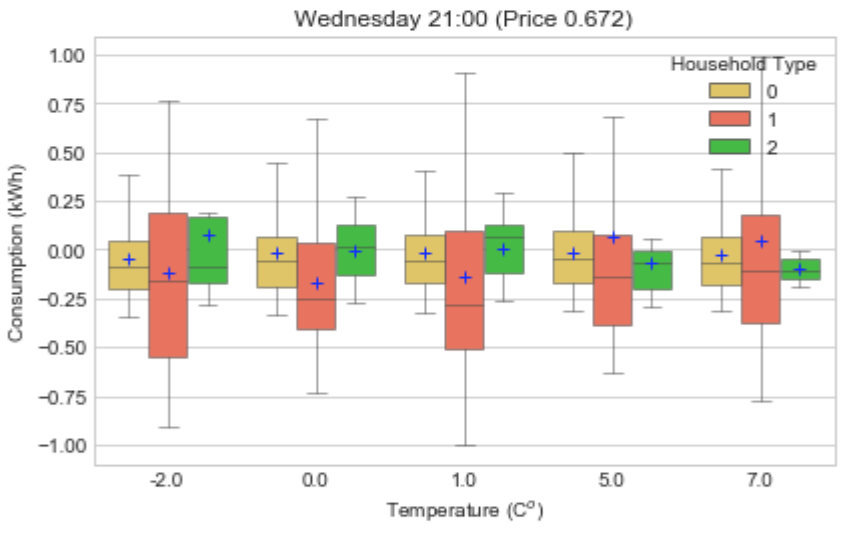

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Wednesday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
1.0
2.0
7.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

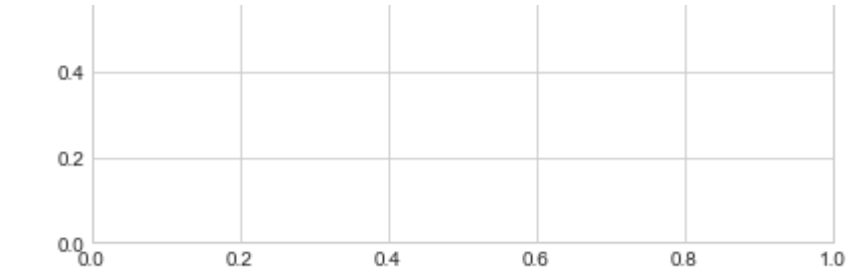

Wednesday 20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
0.0
2.0
5.0
7.0
Temperature (C°)

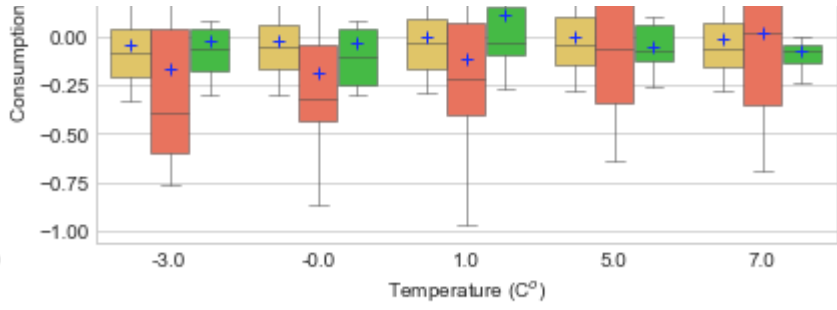

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

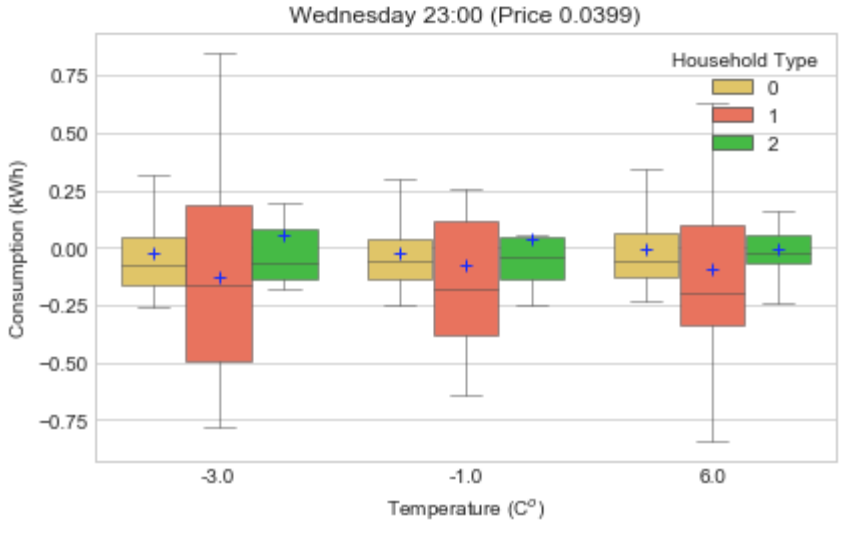

Wednesday 21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-2.0
0.0
1.0
5.0
7.0
Temperature (C°)

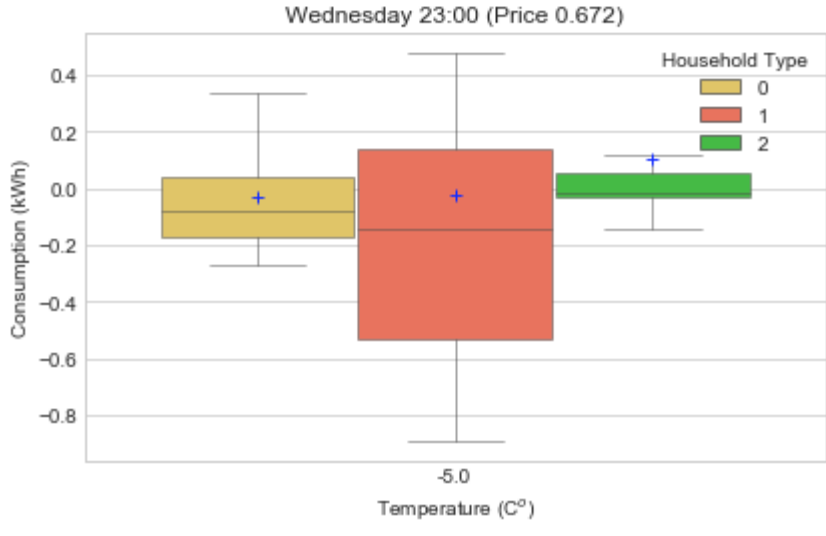
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Wednesday 22:00 (Price 0.672)
Household Type
0
1
2
1.00
0.75
0.50
0.25

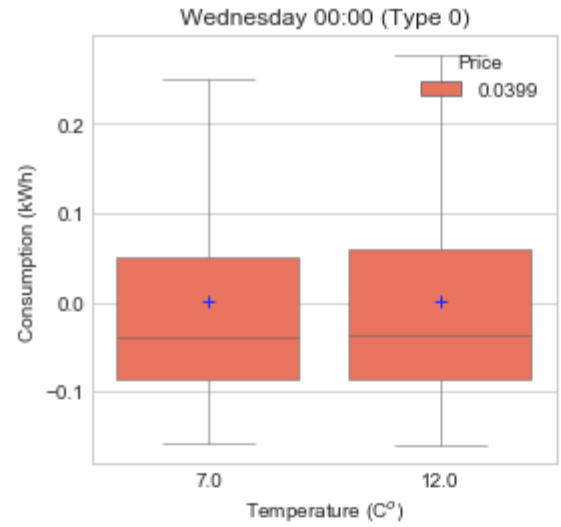

0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

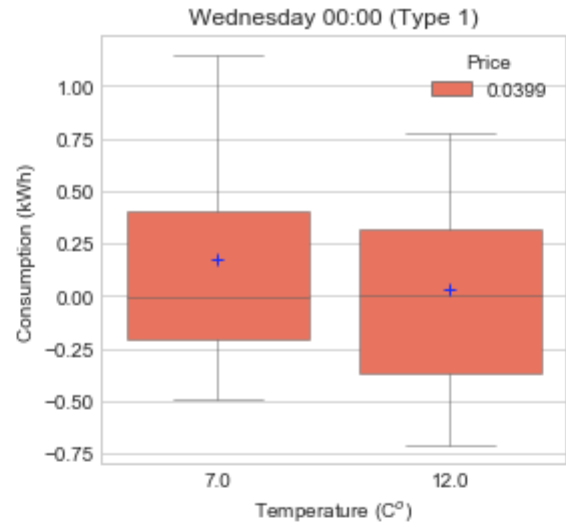

Consumption
0.00
−0.25
−0.50
−0.75
−1.00
-3.0
-0.0
1.0
5.0
7.0
Temperature (C°)

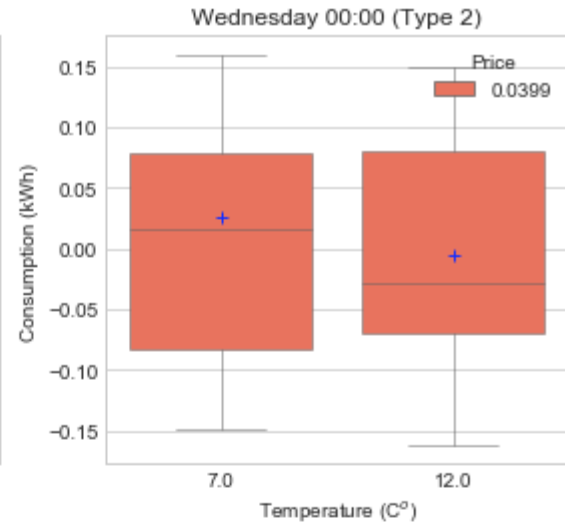

Wednesday 23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-3.0
-1.0
6.0
Temperature (C°)

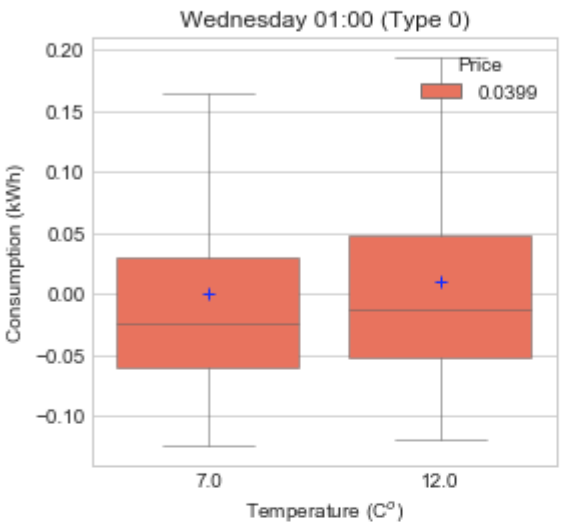

Wednesday 23:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
-5.0
Temperature (C°)

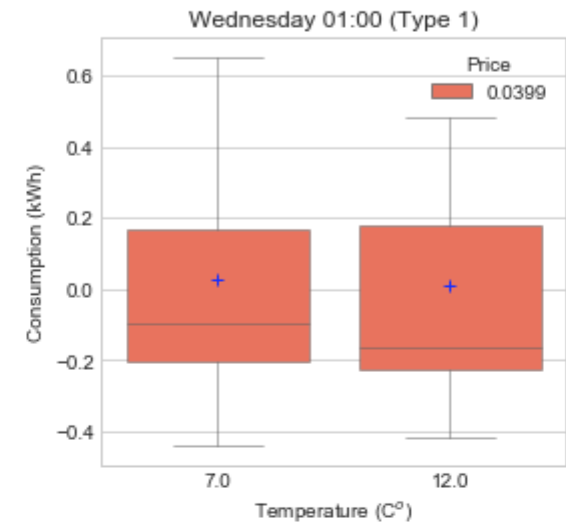
In [130]:
```
# price comparison
# Wednesday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 2 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

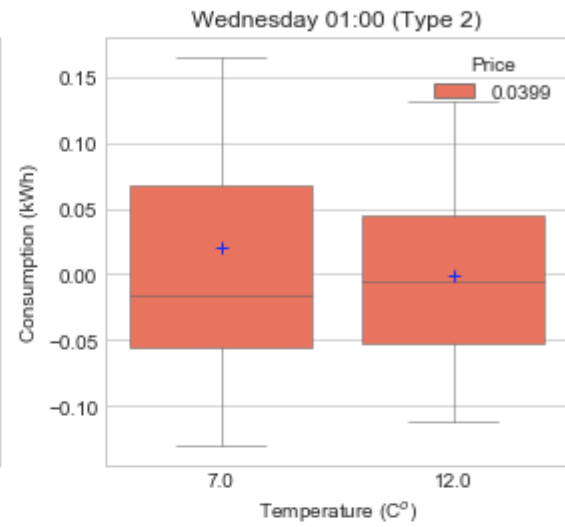

Wednesday 00:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.2
0.1
0.0
−0.1
7.0
12.0
Temperature (C°)

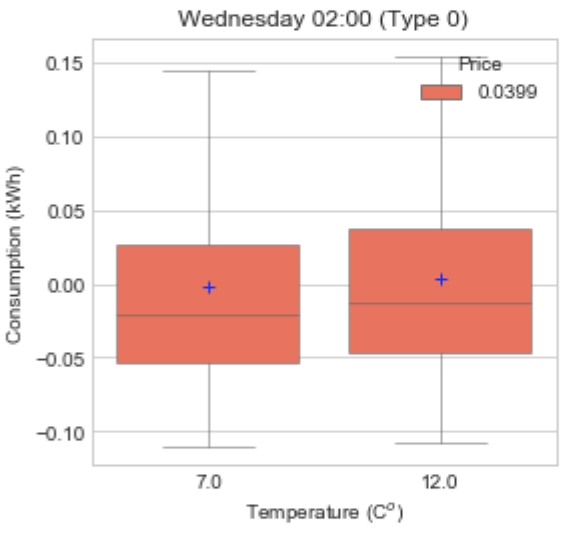

Wednesday 00:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
7.0
12.0
Temperature (C°)

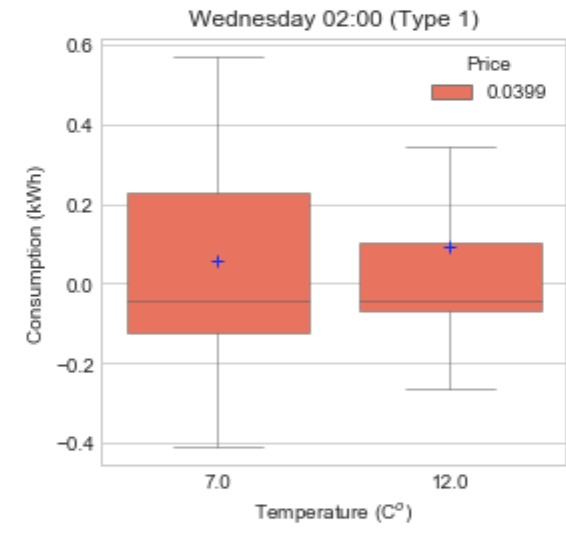

Wednesday 00:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
7.0
12.0
Temperature (C°)

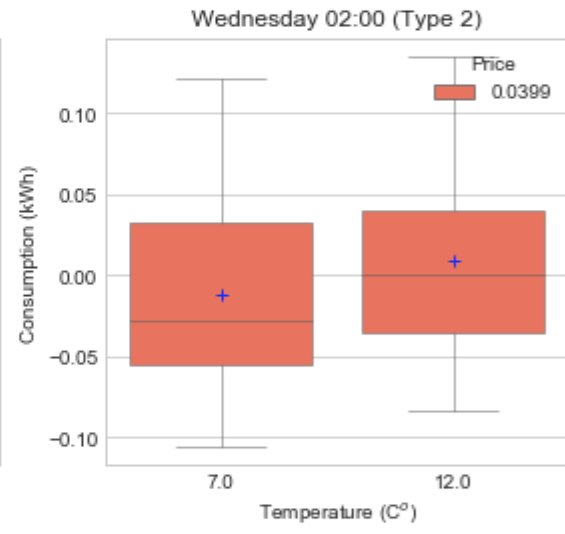

Wednesday 01:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

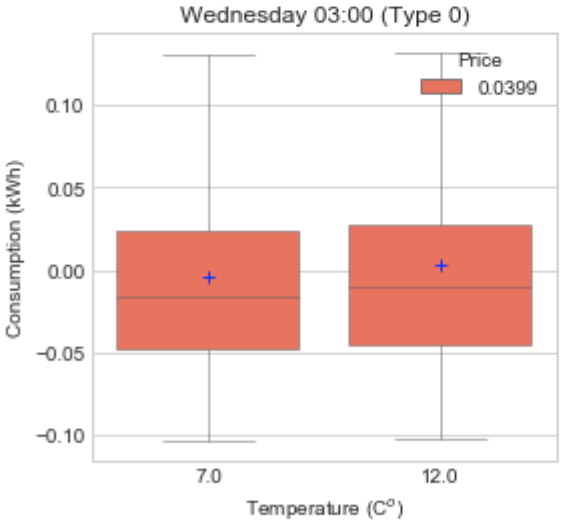

Wednesday 01:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

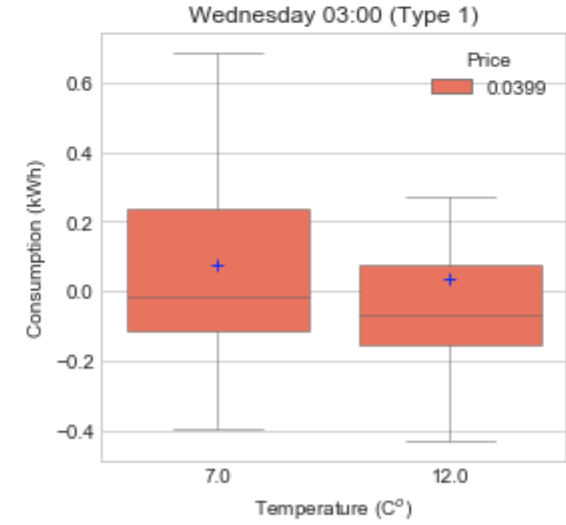

Wednesday 01:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

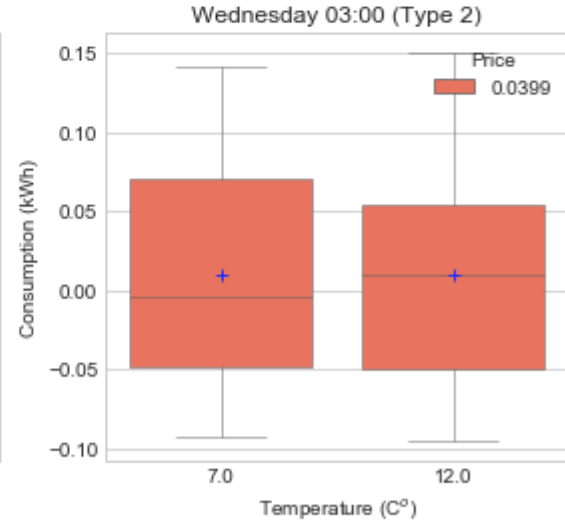

Wednesday 02:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

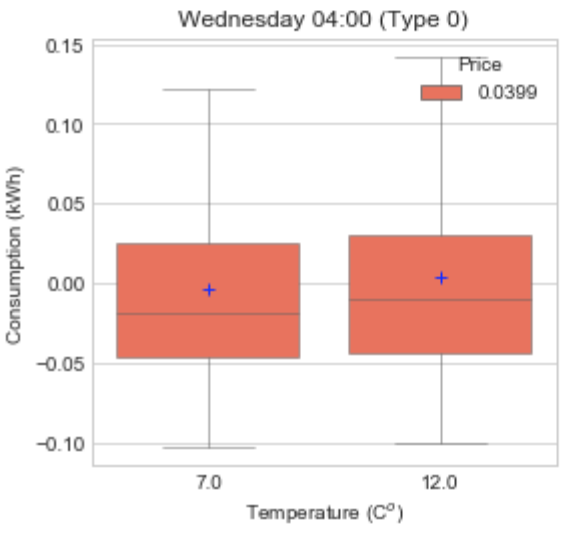

Wednesday 02:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

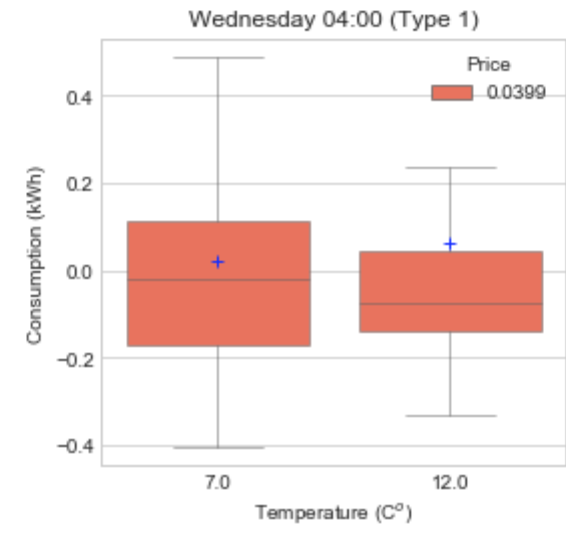

Wednesday 02:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

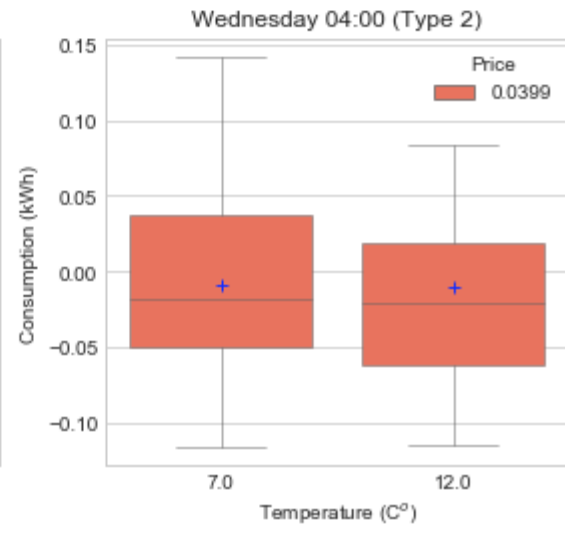

Wednesday 03:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

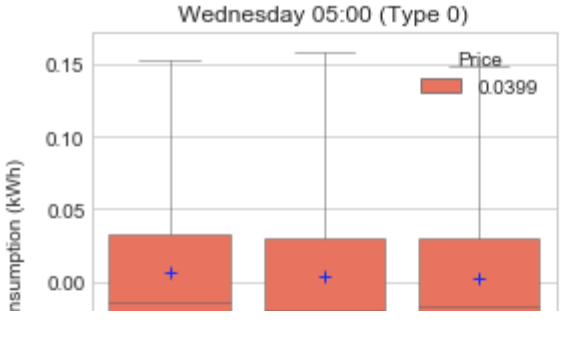

Wednesday 03:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

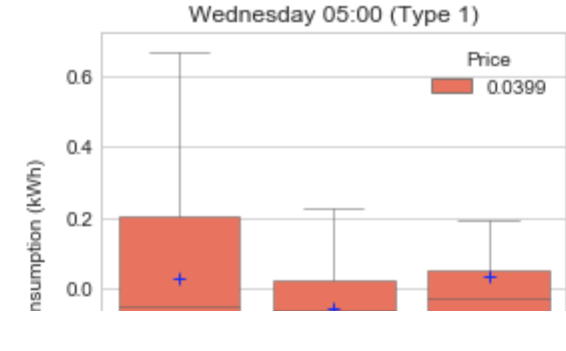

Wednesday 03:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

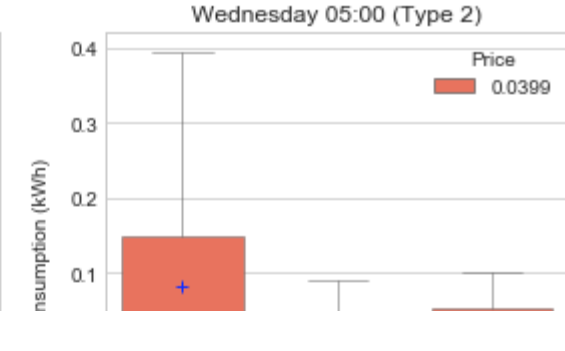

Wednesday 04:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

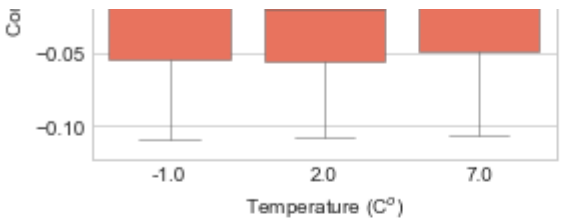

Wednesday 04:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
7.0
12.0
Temperature (C°)

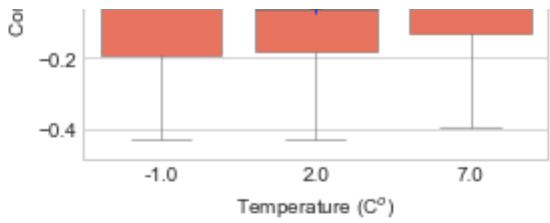

Wednesday 04:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
7.0
12.0
Temperature (C°)

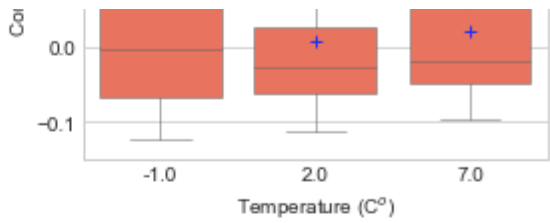

Wednesday 05:00 (Type 0)
Price
0.0399
0.15
0.10
0.05
0.00

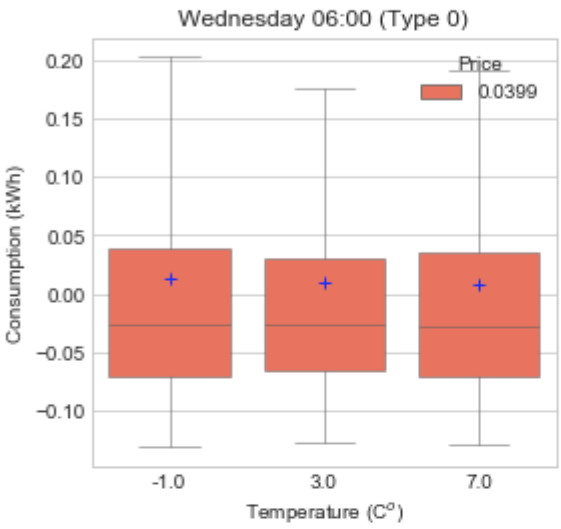

Wednesday 05:00 (Type 1)
Price
0.0399
0.6
0.4
0.2
0.0

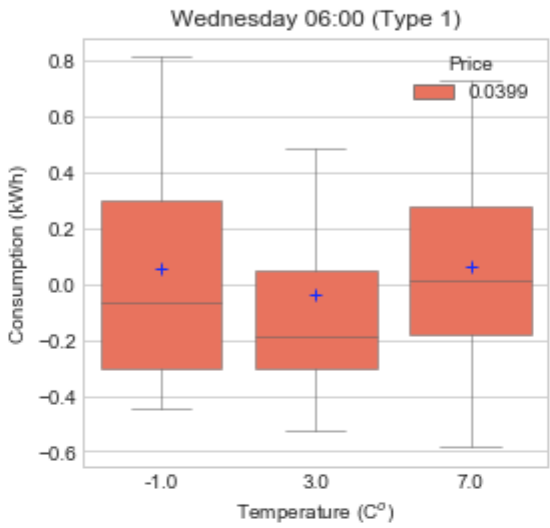

Wednesday 05:00 (Type 2)
Price
0.0399
0.4
0.3
0.2
0.1

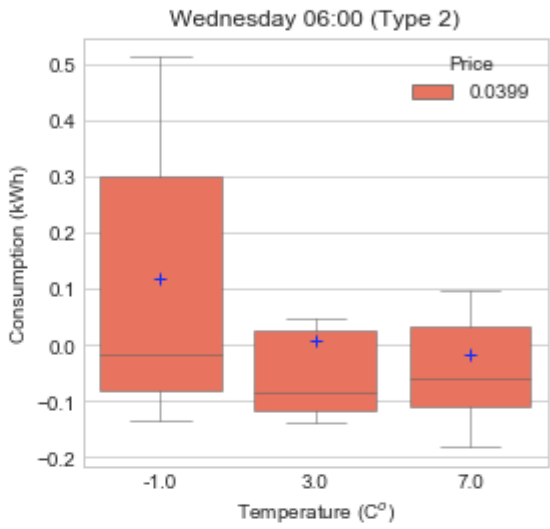

-0.05
-0.10
-1.0
2.0
7.0
Temperature (C°)

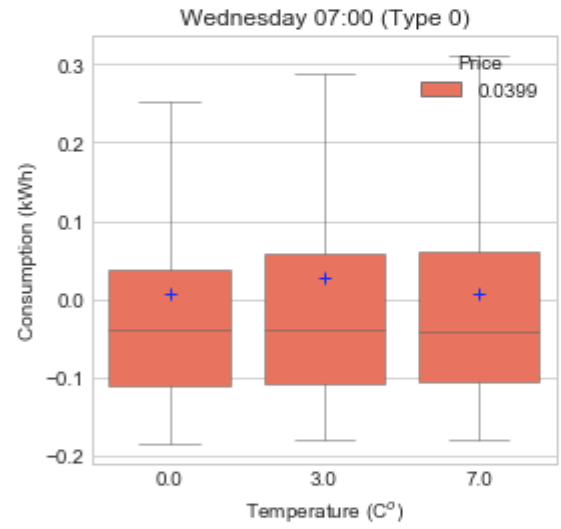

-0.2
-0.4
-1.0
2.0
7.0
Temperature (C°)

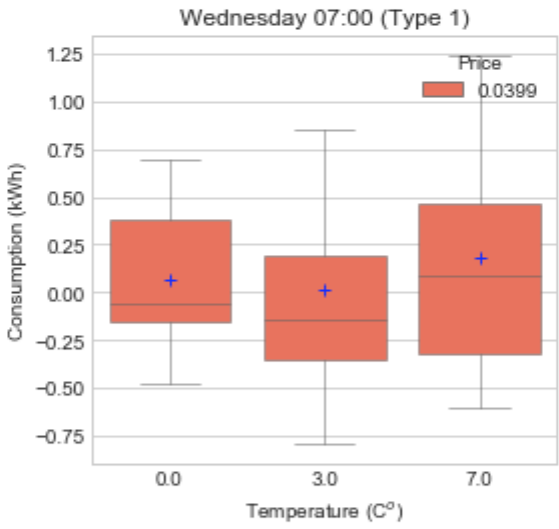

0.0
-0.1
-1.0
2.0
7.0
Temperature (C°)

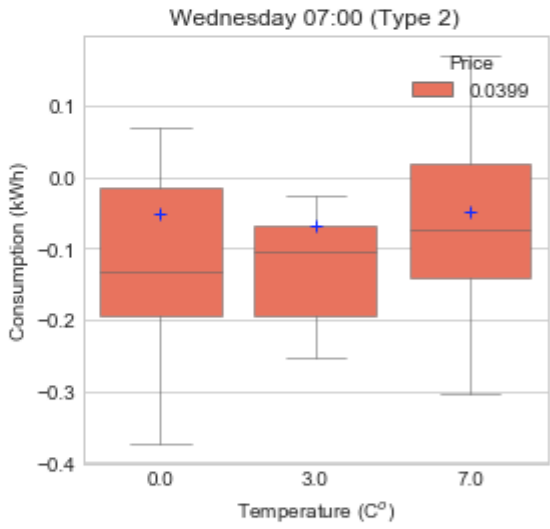

Wednesday 06:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
-1.0
3.0
7.0
Temperature (C°)

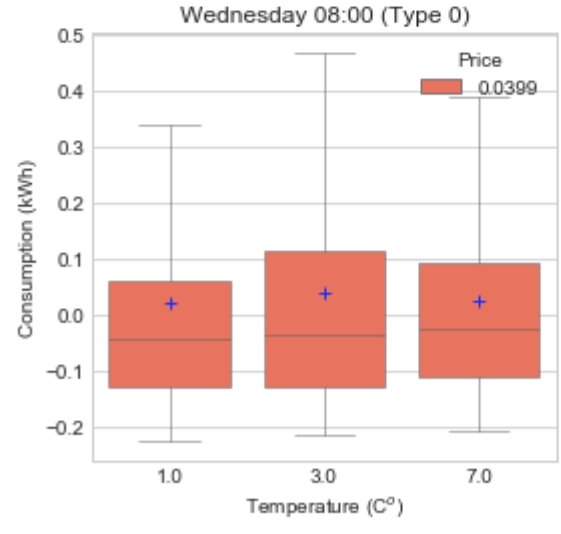

Wednesday 06:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-1.0
3.0
7.0
Temperature (C°)

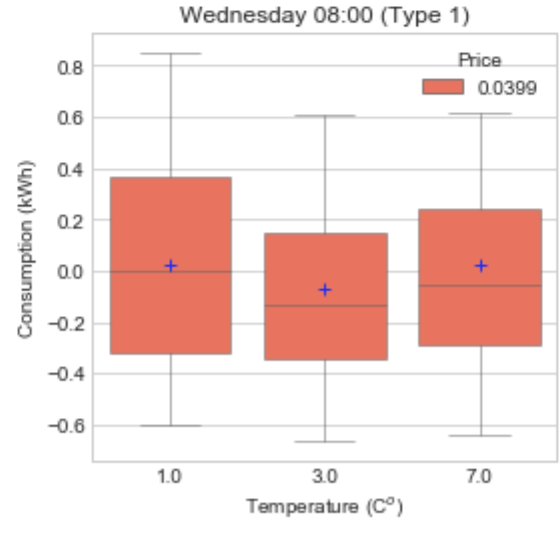

Wednesday 06:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-1.0
3.0
7.0
Temperature (C°)

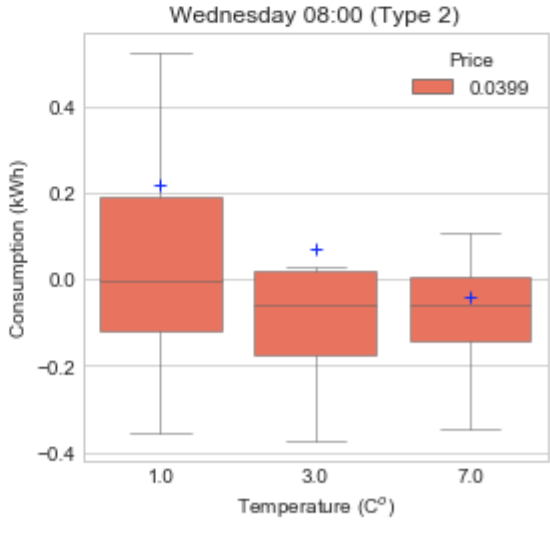

Wednesday 07:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
3.0
7.0
Temperature (C°)

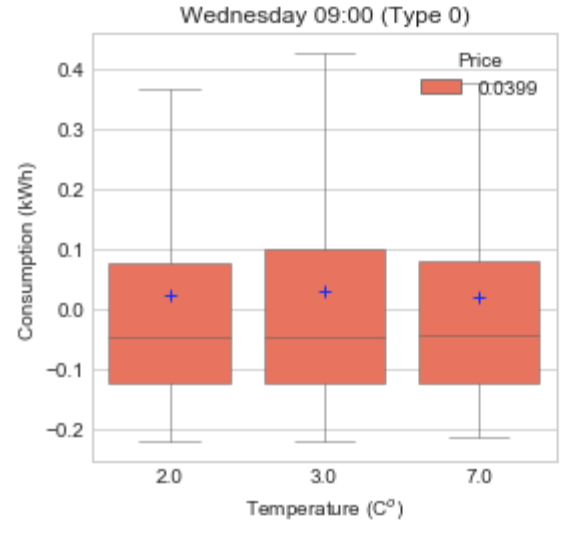

Wednesday 07:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0.0
3.0
7.0
Temperature (C°)

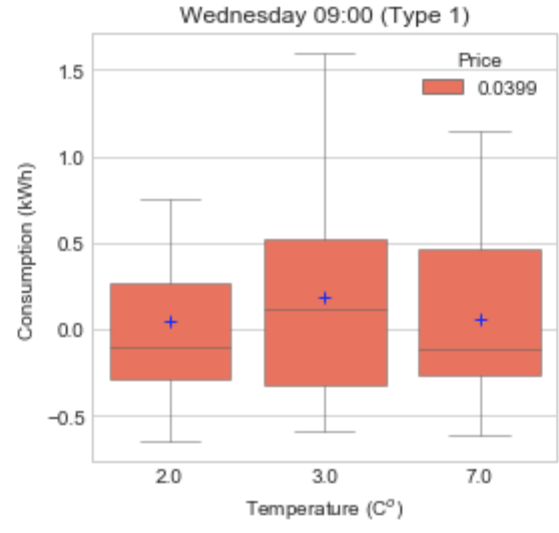

Wednesday 07:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0.0
3.0
7.0
Temperature (C°)

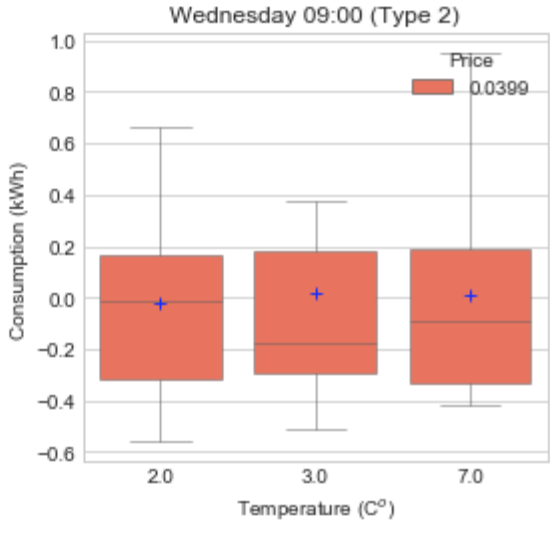

Wednesday 08:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
1.0
3.0
7.0
Temperature (C°)

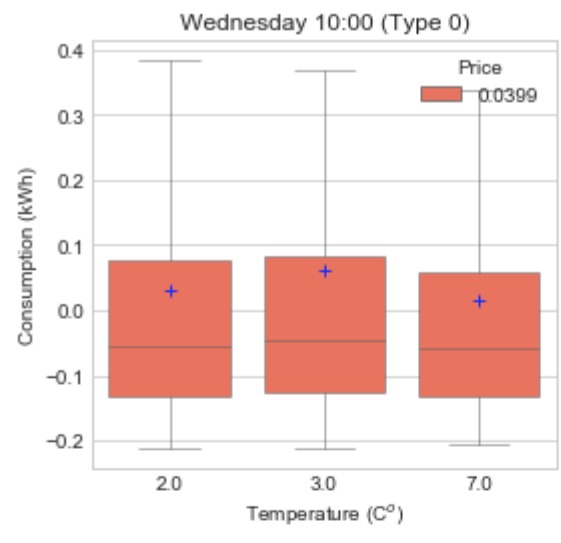

Wednesday 08:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
1.0
3.0
7.0
Temperature (C°)

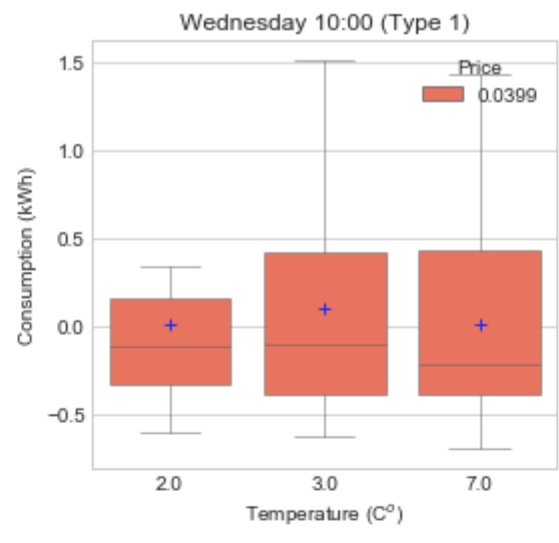

Wednesday 08:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
1.0
3.0
7.0
Temperature (C°)

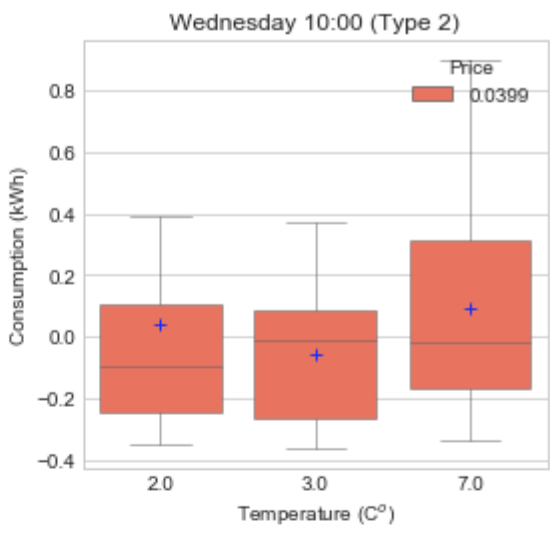

Wednesday 09:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
7.0
Temperature (C°)


Wednesday 09:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
2.0
3.0
7.0
Temperature (C°)


Wednesday 09:00 (Type 2)
Price
0.0399
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
3.0
7.0
Temperature (C°)

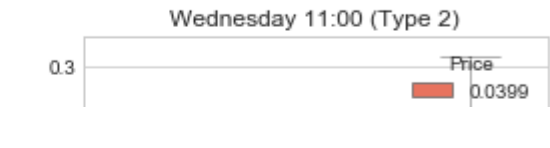

Wednesday 10:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
3.0
7.0
Temperature (C°)

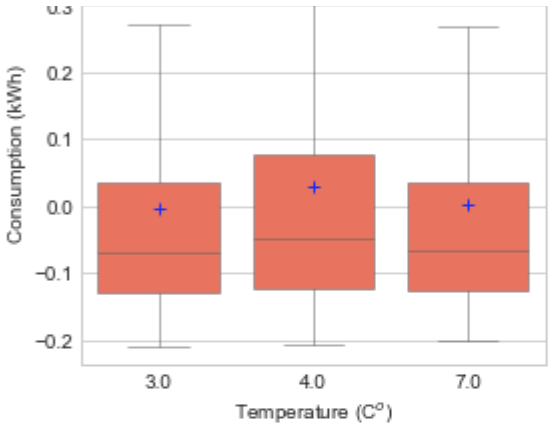

Wednesday 10:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
2.0
3.0
7.0
Temperature (C°)

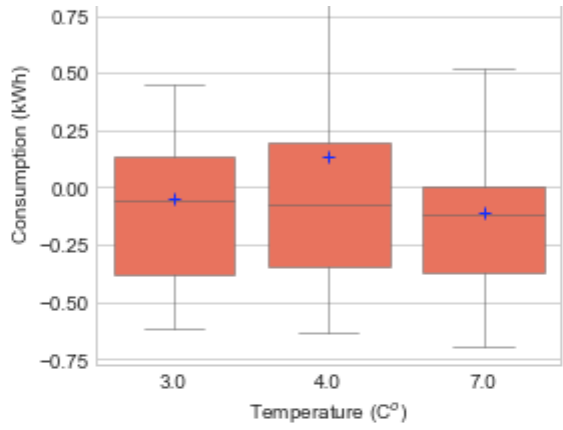

Wednesday 10:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
3.0
7.0
Temperature (C°)

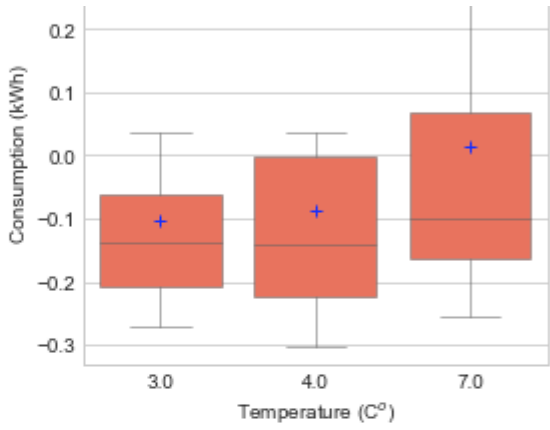

Wednesday 11:00 (Type 0)
Price
0.0399
0.4

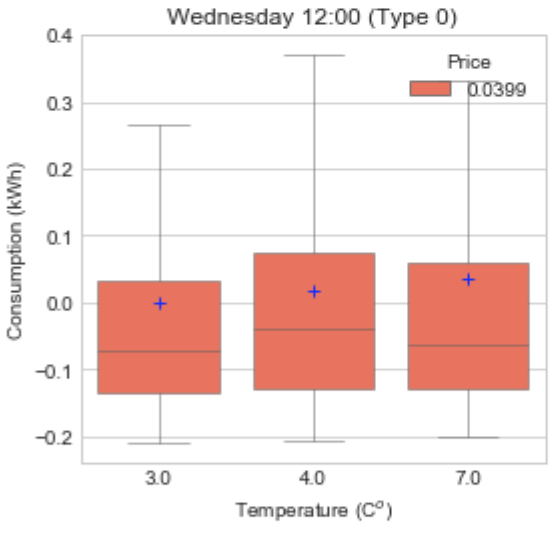

Wednesday 11:00 (Type 1)
Price
0.0399
1.00

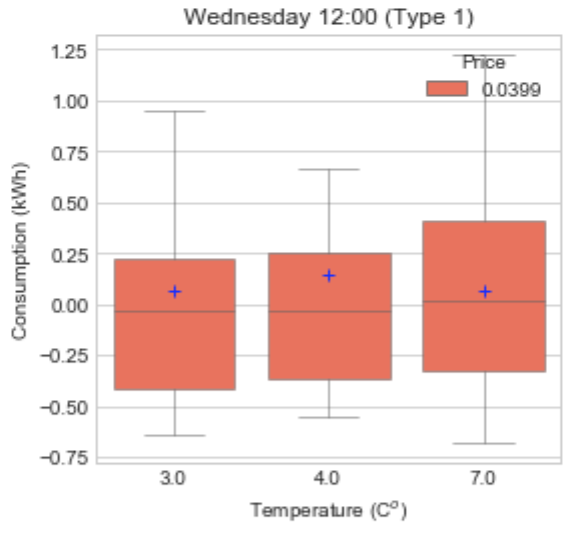

Wednesday 11:00 (Type 2)
Price
0.0399
0.3

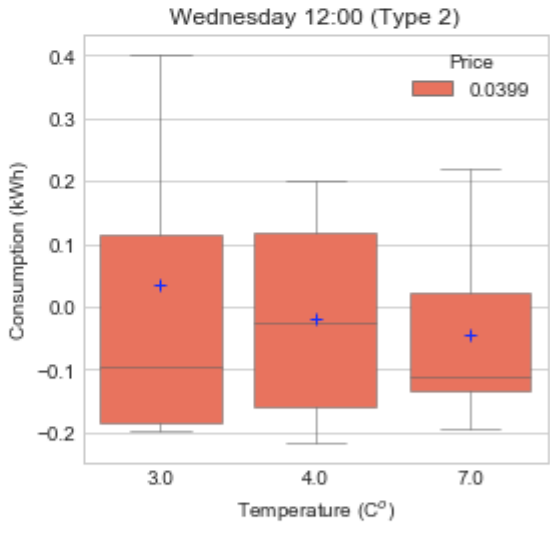

Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

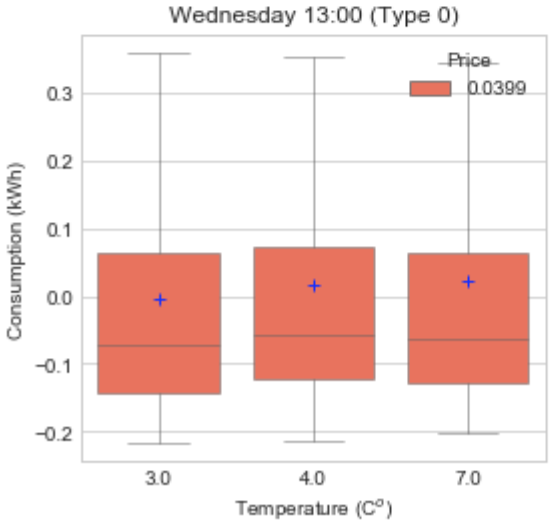

Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

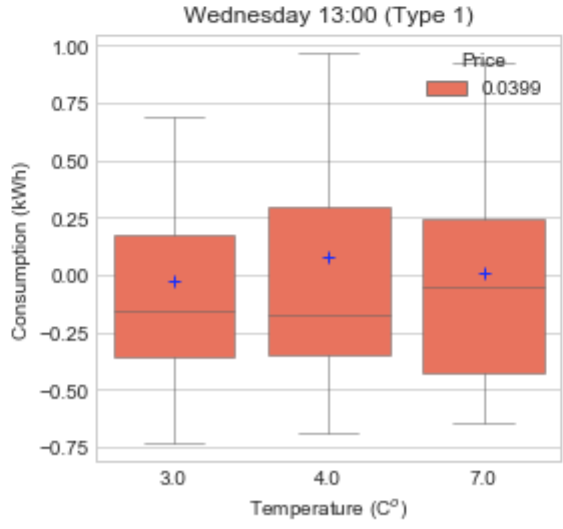

Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

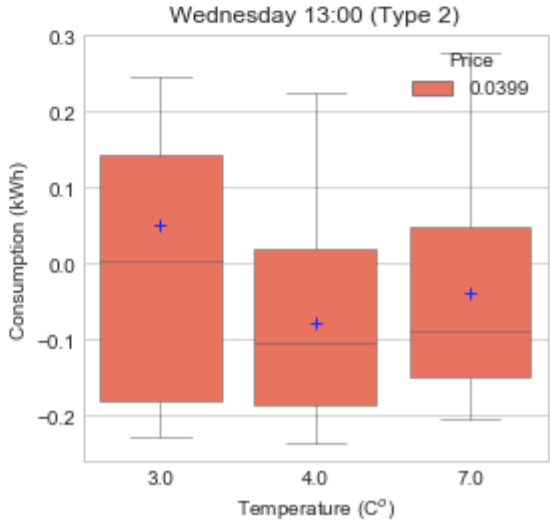

Wednesday 12:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

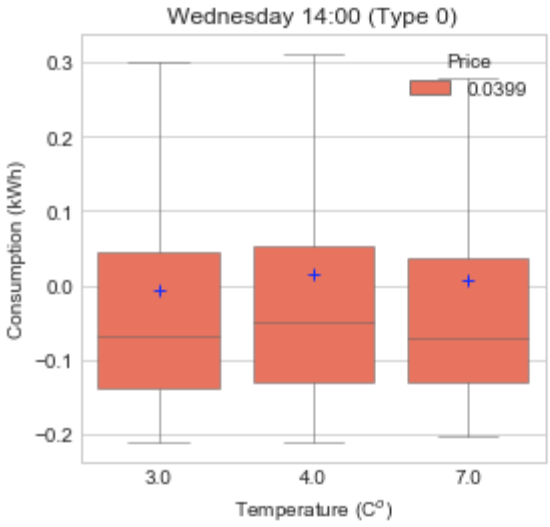

Wednesday 12:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

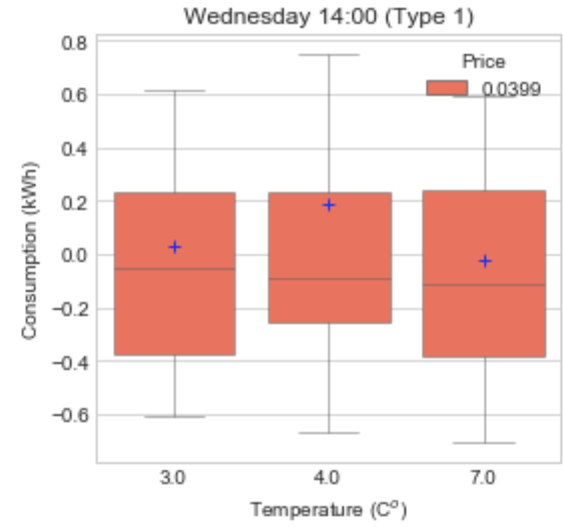

Wednesday 12:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

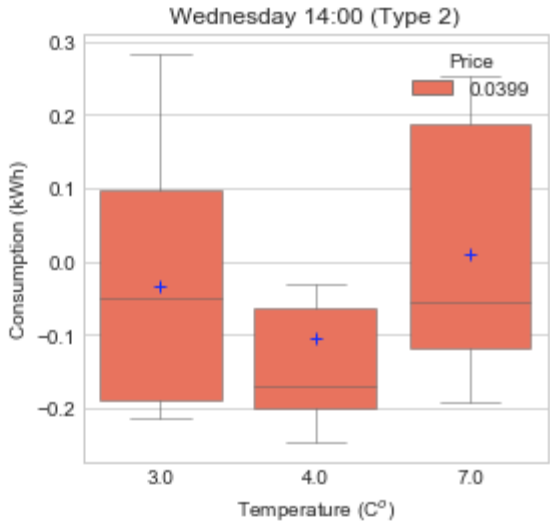

Wednesday 13:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

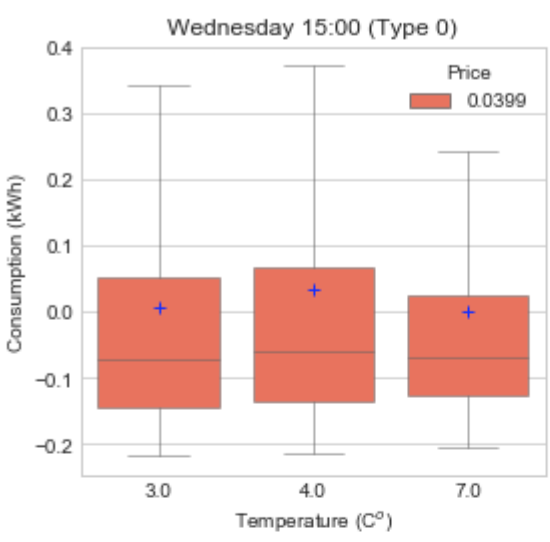

Wednesday 13:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

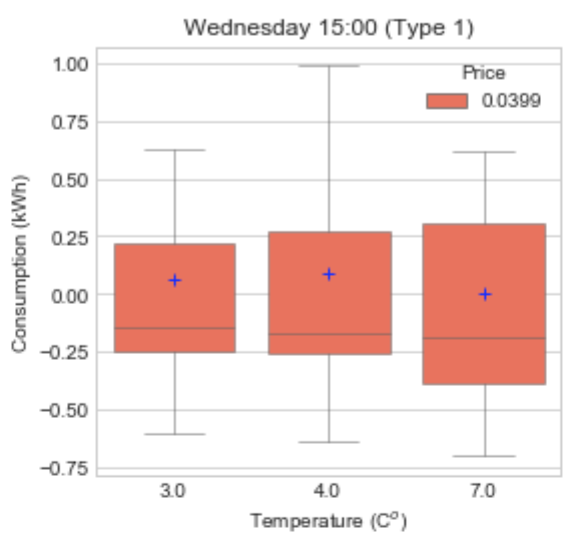

Wednesday 13:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

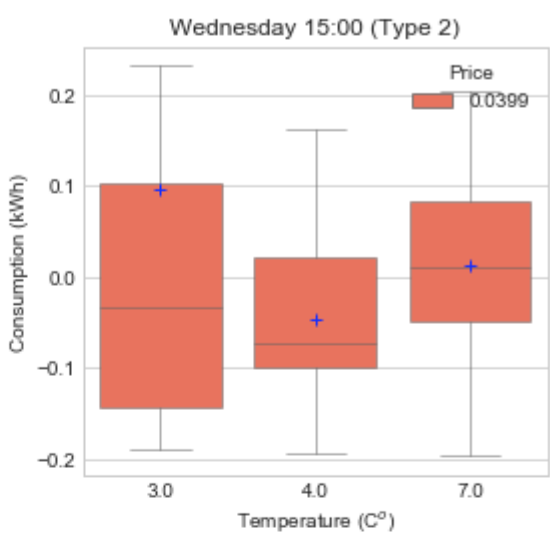

Wednesday 14:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

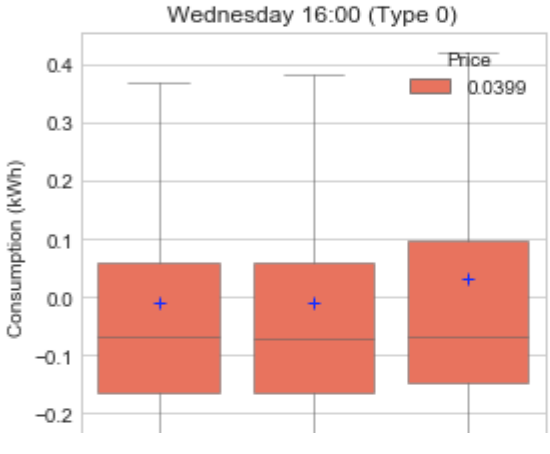

Wednesday 14:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

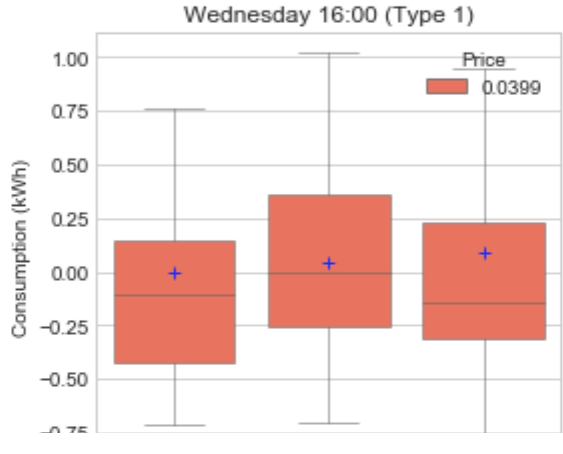

Wednesday 14:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

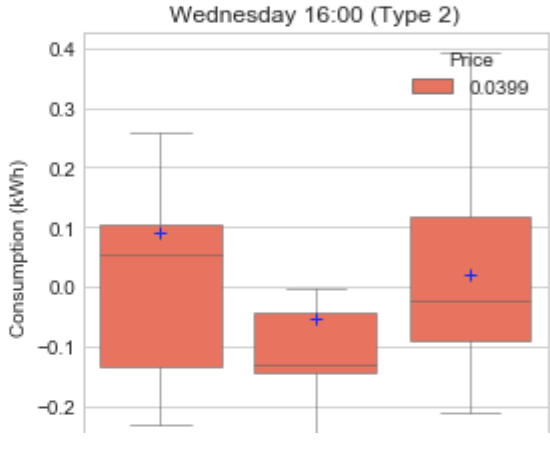

Wednesday 15:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

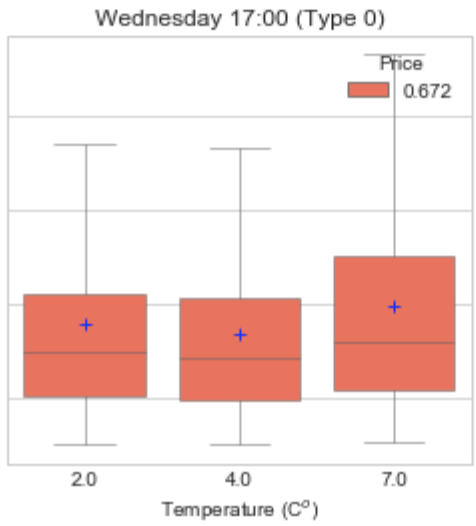

Wednesday 15:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

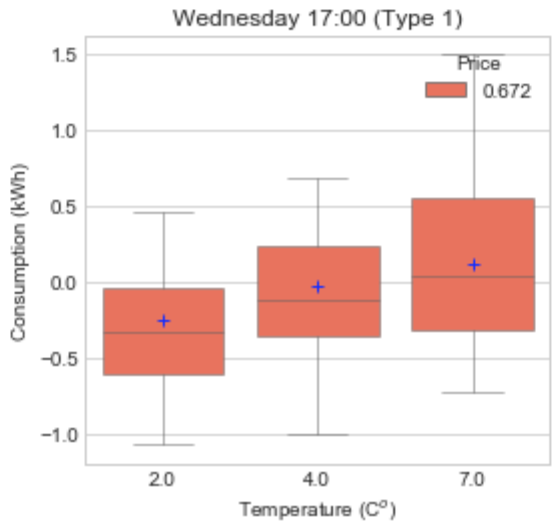

Wednesday 15:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
3.0
4.0
7.0

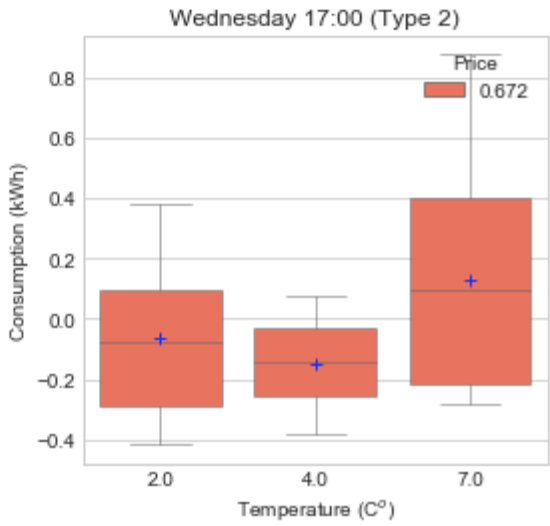

Wednesday 16:00 (Type 0)
Price
0.0399
Consumption (kWh)

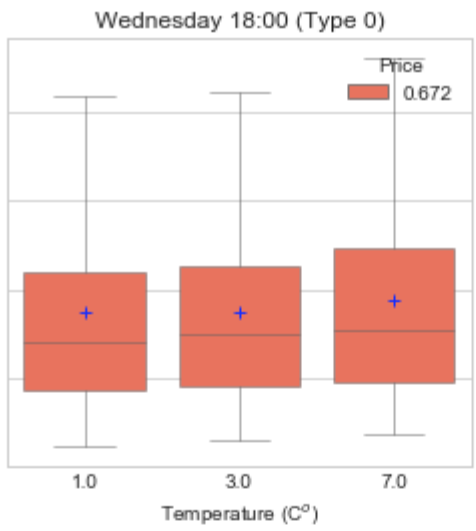

Wednesday 16:00 (Type 1)
Price
0.0399
Consumption (kWh)

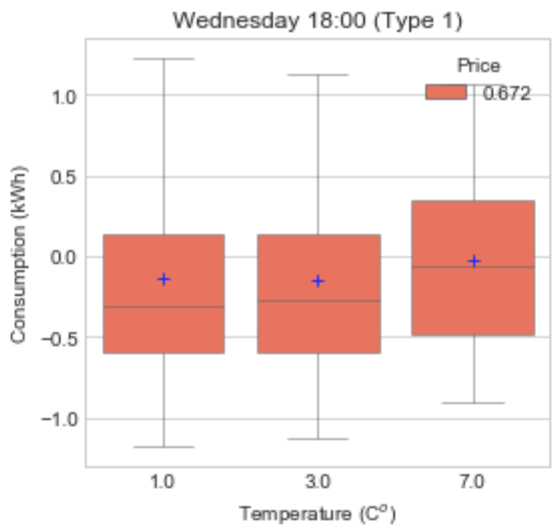

Wednesday 16:00 (Type 2)
Price
0.0399
Consumption (kWh)

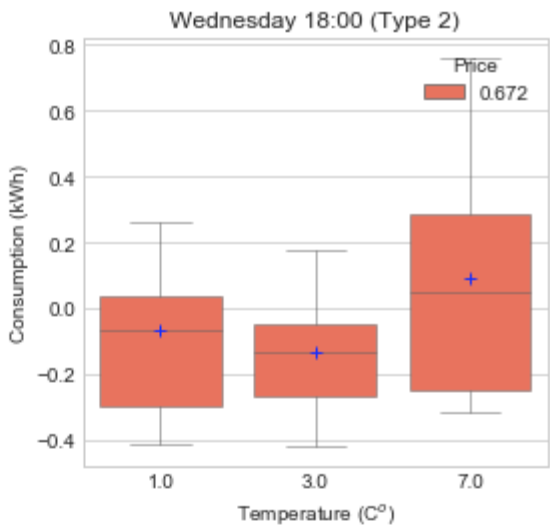

Wednesday 17:00 (Type 0)
Price
0.672
Consumption (kWh)
Temperature (C°)
2.0
4.0
7.0

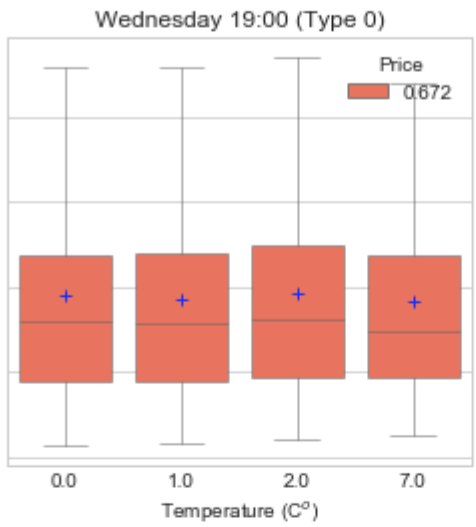

Wednesday 17:00 (Type 1)
Price
0.672
Consumption (kWh)
Temperature (C°)
2.0
4.0
7.0

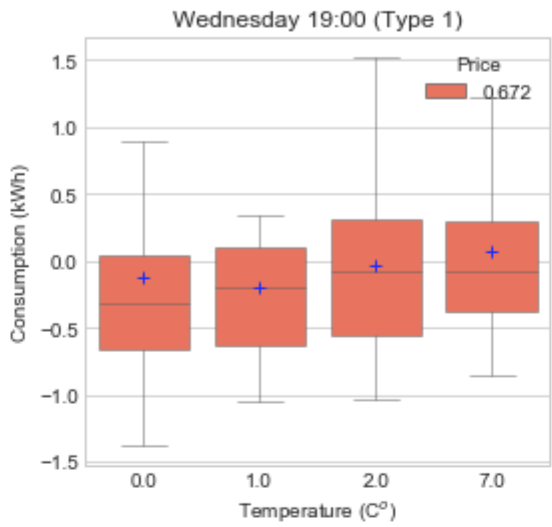

Wednesday 17:00 (Type 2)
Price
0.672
Consumption (kWh)
Temperature (C°)
2.0
4.0
7.0

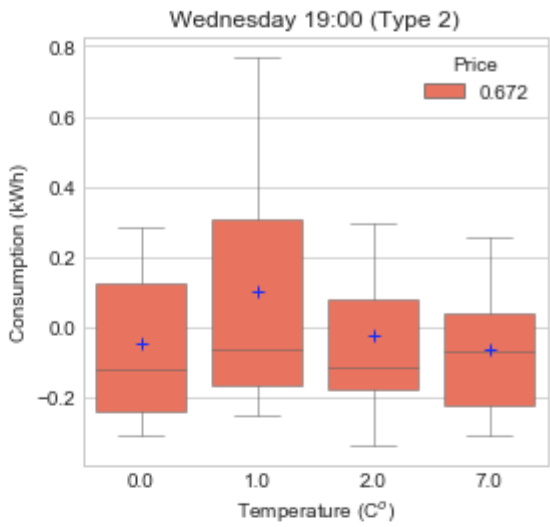

Wednesday 18:00 (Type 0)
Price
0.672
Consumption (kWh)
Temperature (C°)
1.0
3.0
7.0

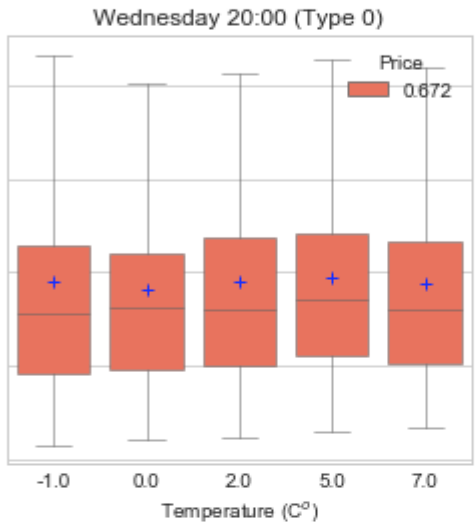

Wednesday 18:00 (Type 1)
Price
0.672
Consumption (kWh)
Temperature (C°)
1.0
3.0
7.0

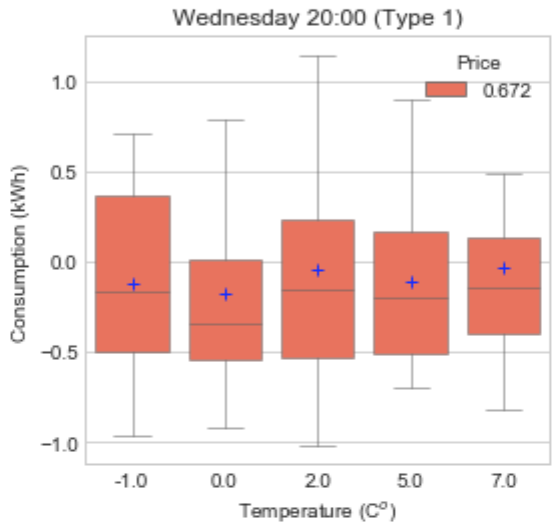

Wednesday 18:00 (Type 2)
Price
0.672
Consumption (kWh)
Temperature (C°)
1.0
3.0
7.0

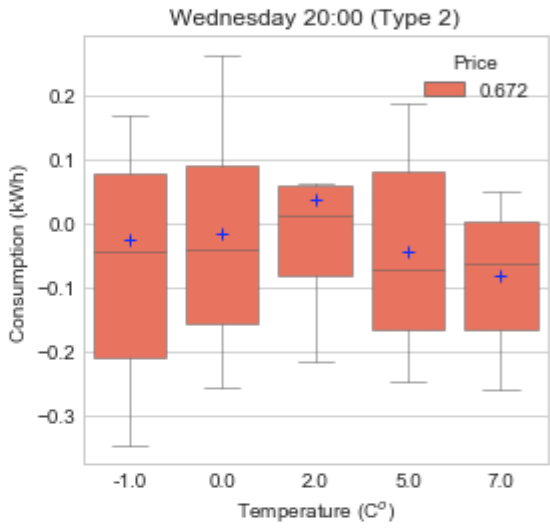

Wednesday 19:00 (Type 0)
Price
0.672
Consumption (kWh)
Temperature (C°)
0.0
1.0
2.0
7.0

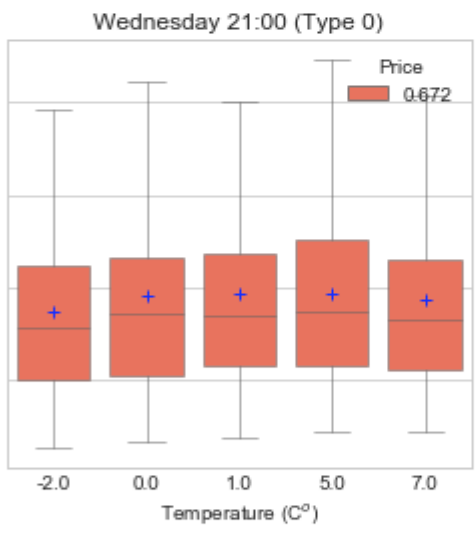

Wednesday 19:00 (Type 1)
Price
0.672
Consumption (kWh)
Temperature (C°)
0.0
1.0
2.0
7.0

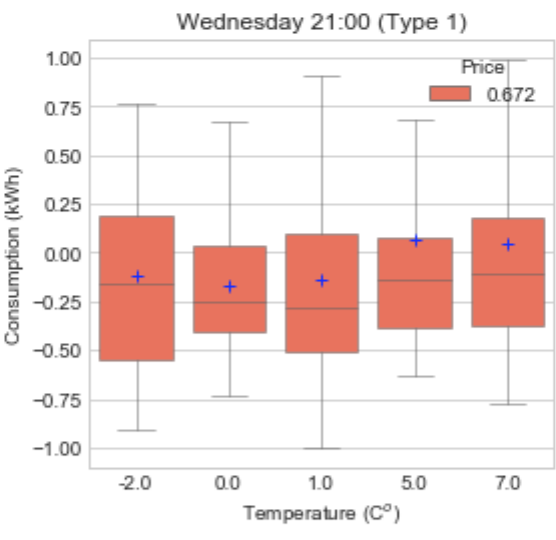

Wednesday 19:00 (Type 2)
Price
0.672
Consumption (kWh)
Temperature (C°)
0.0
1.0
2.0
7.0

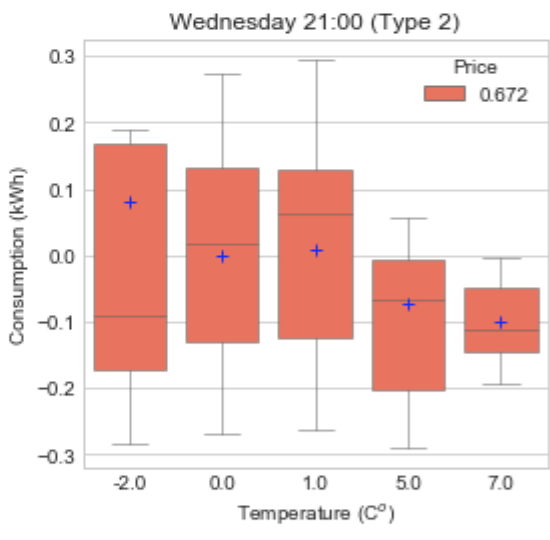

Wednesday 20:00 (Type 0)
Price
0.672
Consumption (kWh)
Temperature (C°)
-1.0
0.0
2.0
5.0
7.0


Wednesday 20:00 (Type 1)
Price
0.672
Consumption (kWh)
Temperature (C°)
-1.0
0.0
2.0
5.0
7.0


Wednesday 20:00 (Type 2)
Price
0.672
Consumption (kWh)
Temperature (C°)
-1.0
0.0
2.0
5.0
7.0


Wednesday 21:00 (Type 0)
Price
0.672
Consumption (kWh)
Temperature (C°)
-2.0
0.0
1.0
5.0
7.0

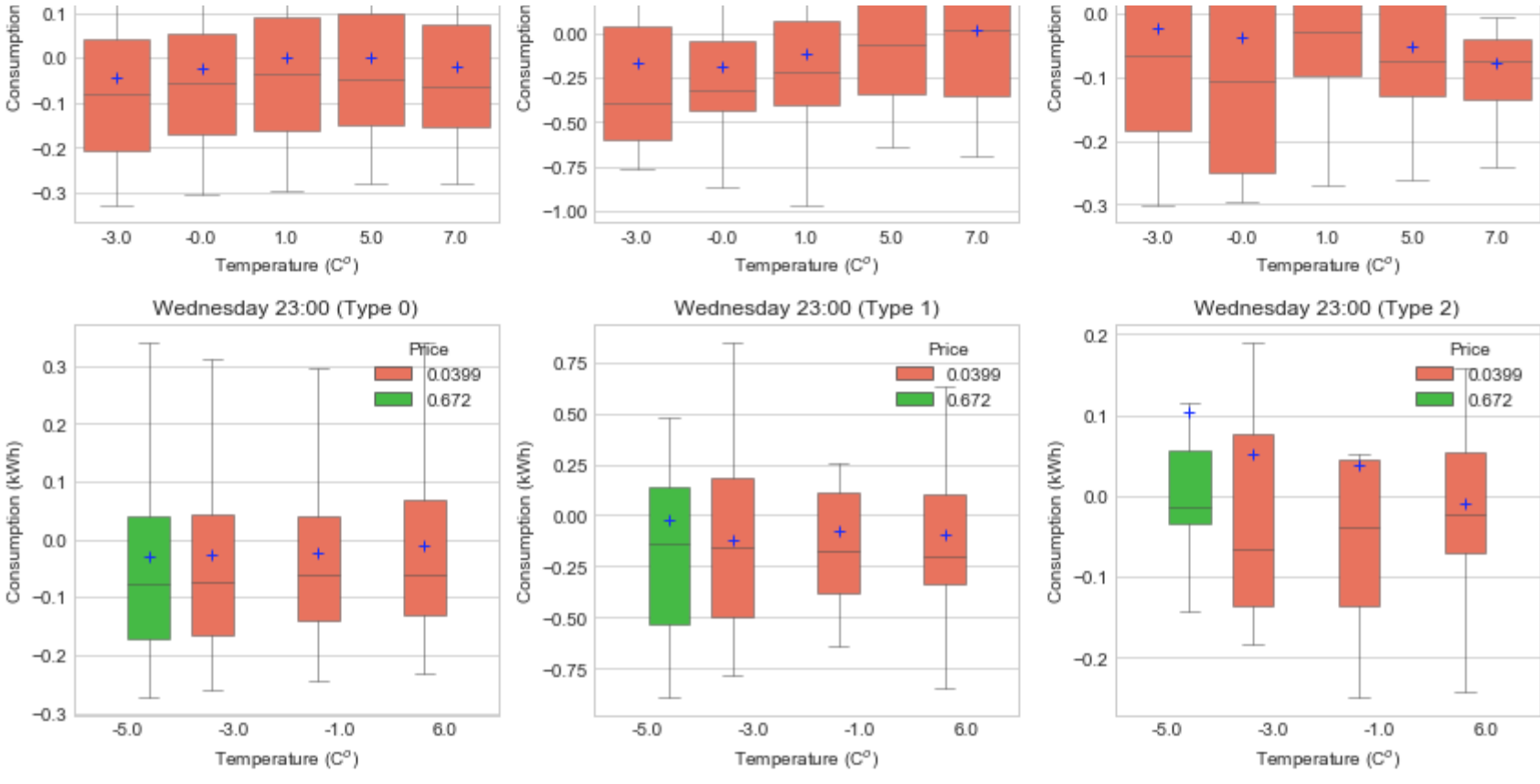

Wednesday 21:00 (Type 1)
Price
0.672
Consumption (kWh)
Temperature (C°)
-2.0
0.0
1.0
5.0
7.0


Wednesday 21:00 (Type 2)
Price
0.672
Consumption (kWh)
Temperature (C°)
-2.0
0.0
1.0
5.0
7.0

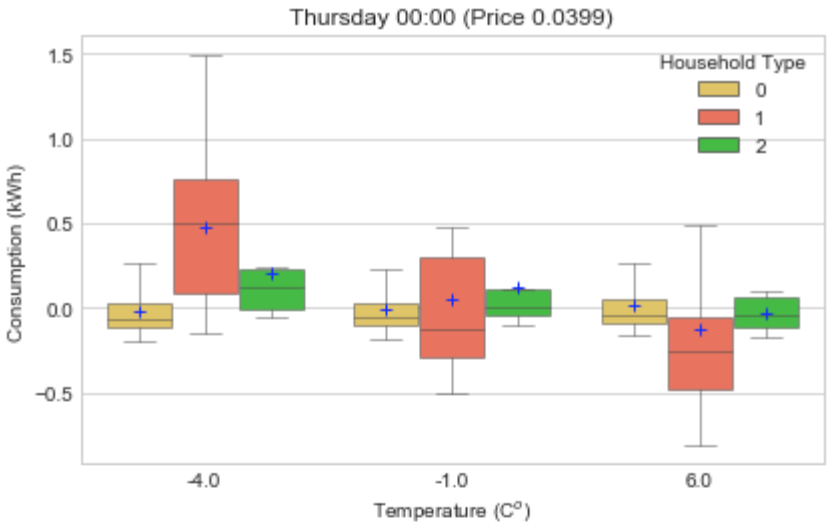
Wednesday 22:00 (Type 0)
Price
0.672
Temperature (C°)
-1.0
0.0
2.0
5.0
7.0

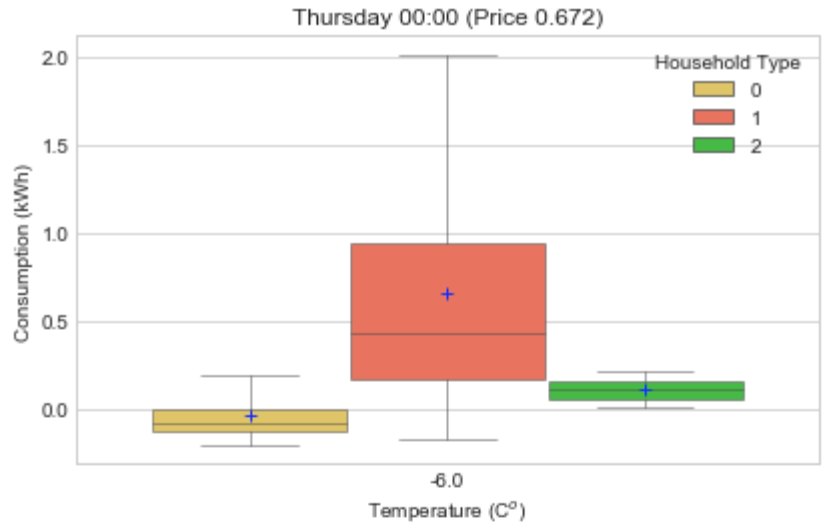
Wednesday 22:00 (Type 1)
Price
0.672
Temperature (C°)
-1.0
0.0
2.0
5.0
7.0

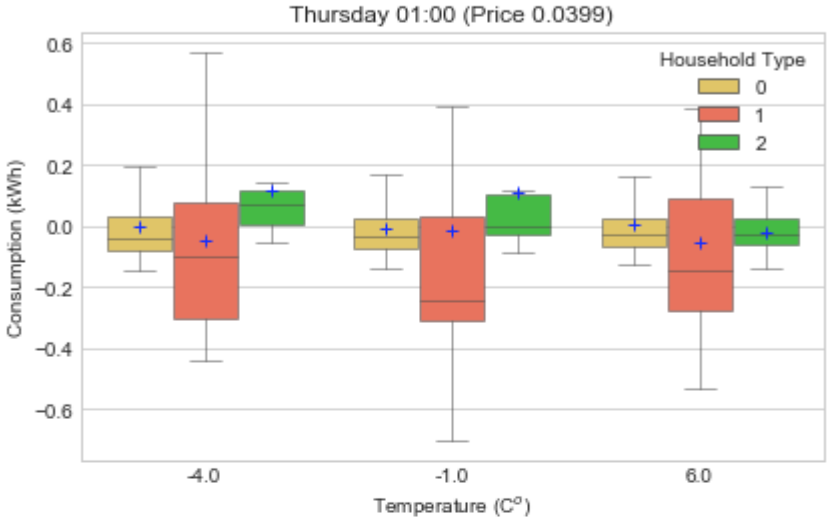
Wednesday 22:00 (Type 2)
Price
0.672
Temperature (C°)
-1.0
0.0
2.0
5.0
7.0

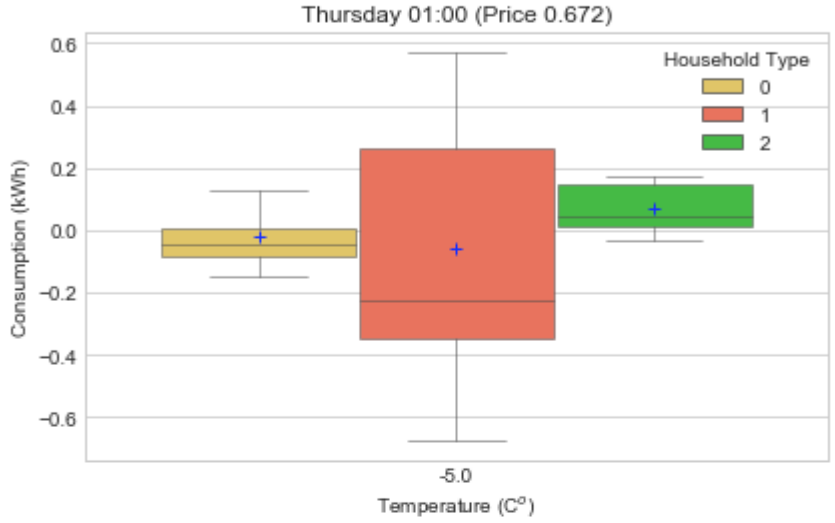

Consumption
Temperature (C°)
-3.0
-0.0
1.0
5.0
7.0
Wednesday 23:00 (Type 0)
Wednesday 23:00 (Type 1)
Wednesday 23:00 (Type 2)
Consumption (kWh)
Price
0.0399
0.672
-5.0
-3.0
-1.0
6.0

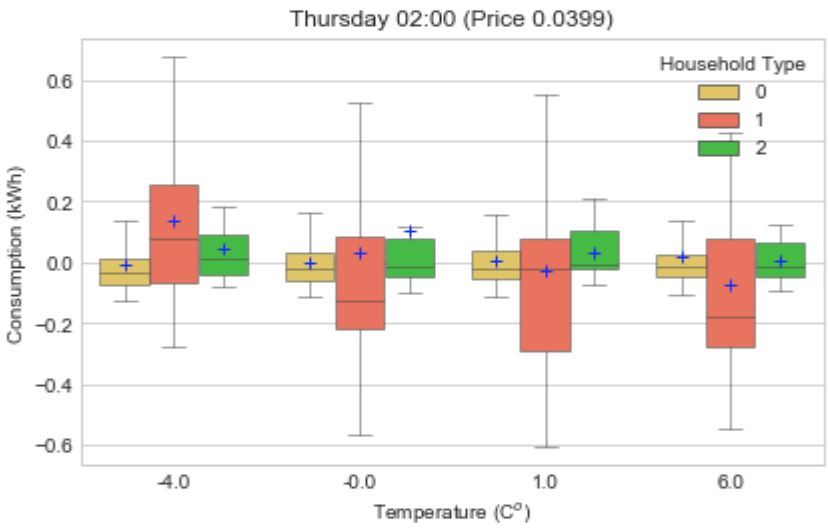
In [100]:
```
# Thursday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 3 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

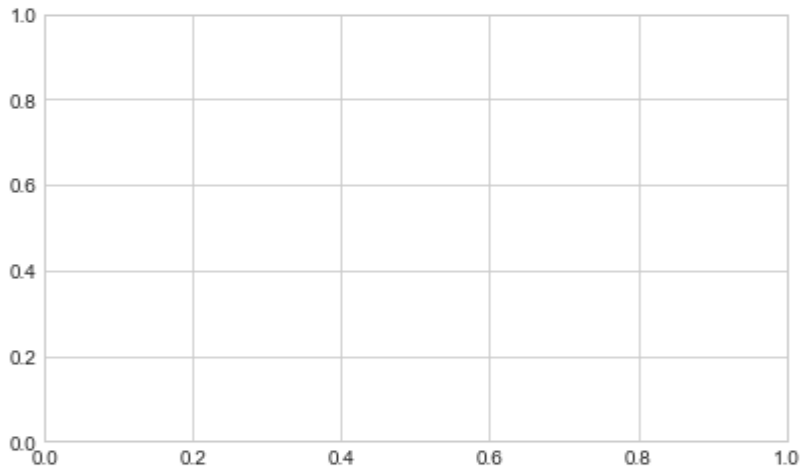

Thursday 00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-4.0
-1.0
6.0
Temperature (C°)

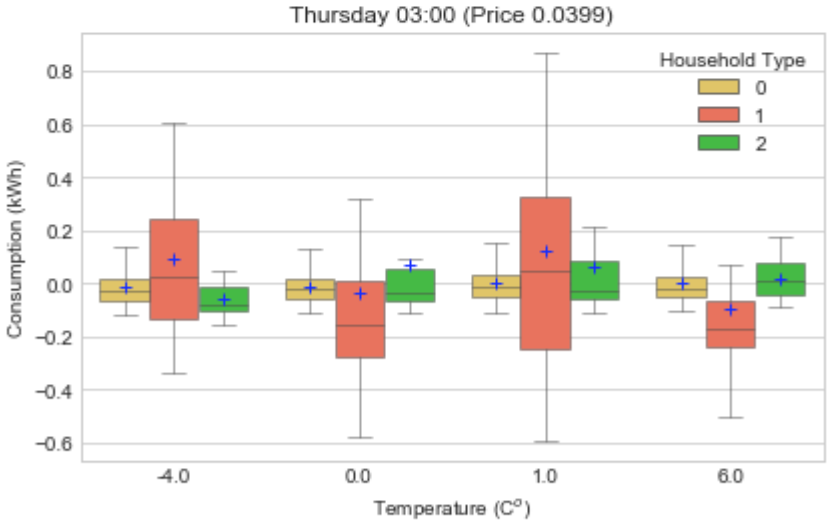

Thursday 00:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
-6.0
Temperature (C°)

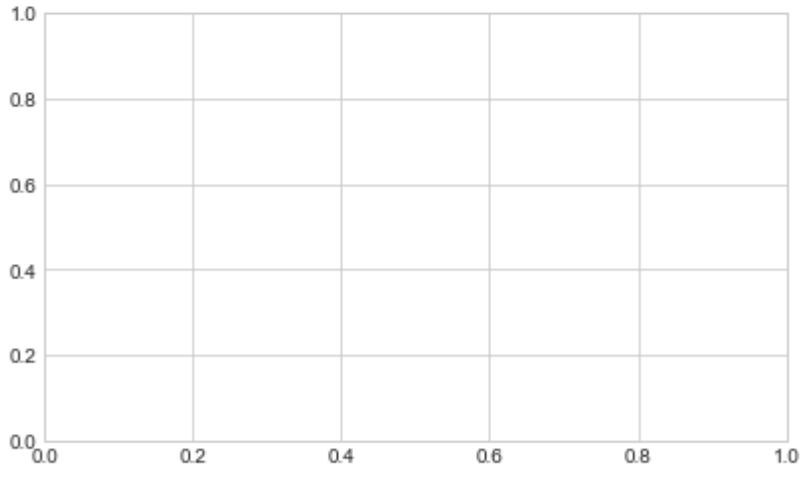

Thursday 01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
-1.0
6.0
Temperature (C°)

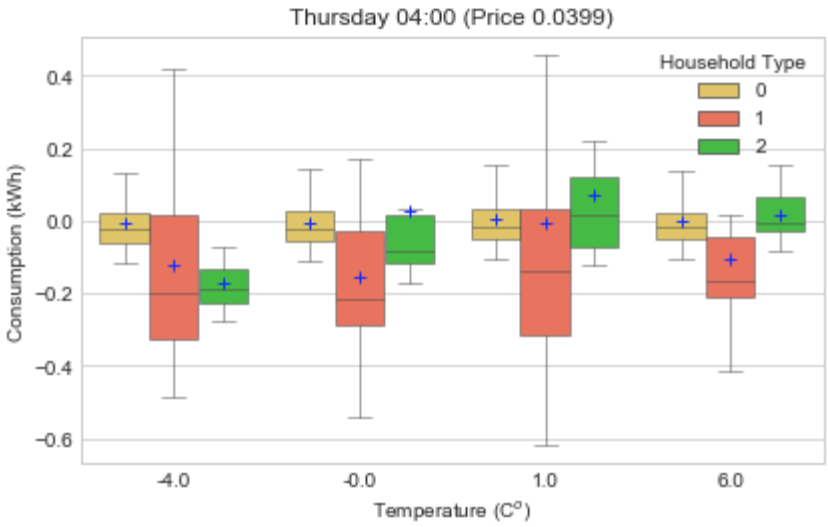

Thursday 01:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-5.0
Temperature (C°)

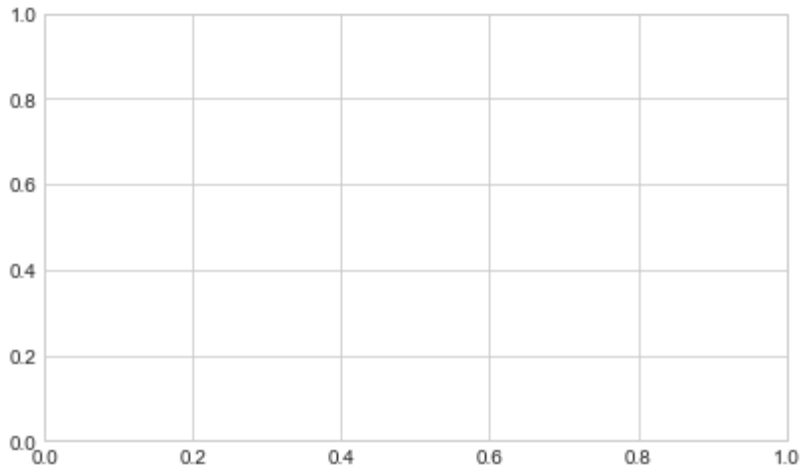

Thursday 02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
-0.0
1.0
6.0
Temperature (C°)

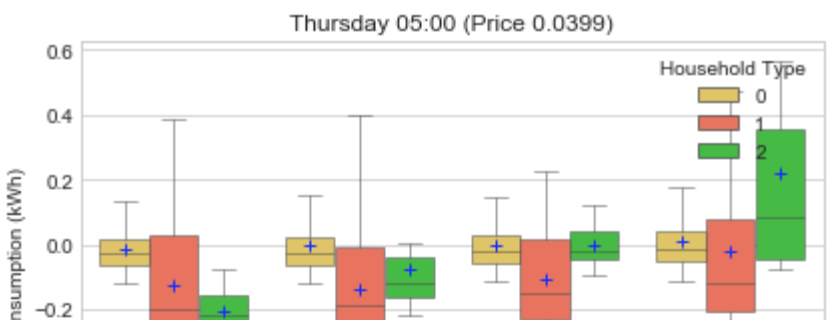

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

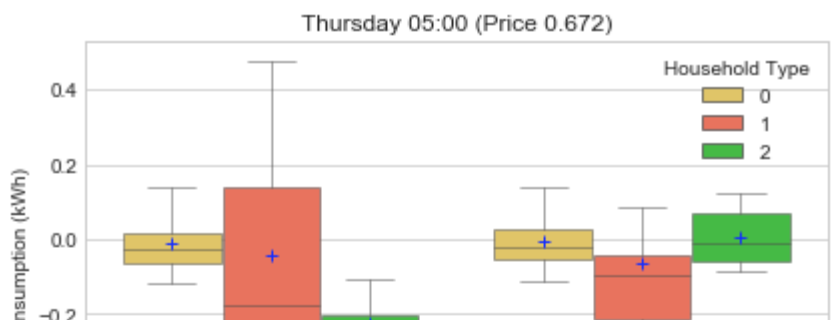

Thursday 03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
0.0
1.0
6.0
Temperature (C°)

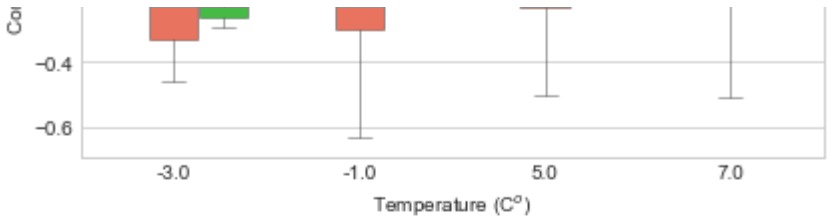

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

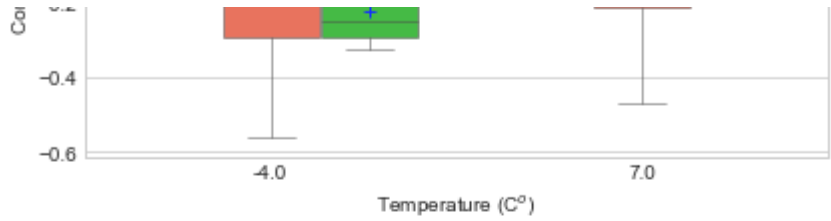

Thursday 04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
-0.0
1.0
6.0
Temperature (C°)

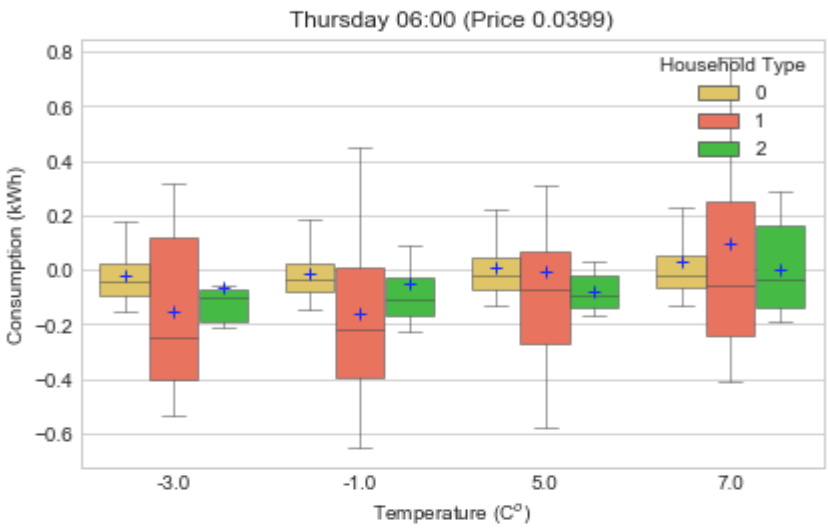

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

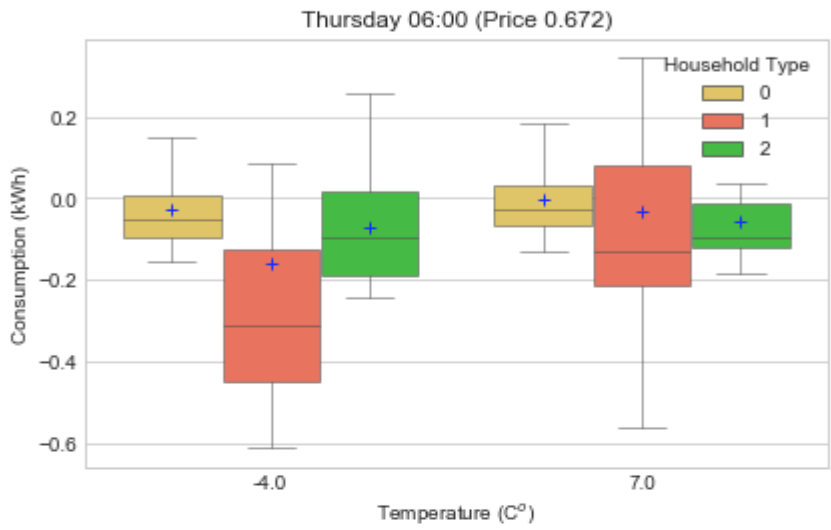

Thursday 05:00 (Price 0.0399)
Household Type
0
1
2
0.6
0.4
0.2
0.0
−0.2

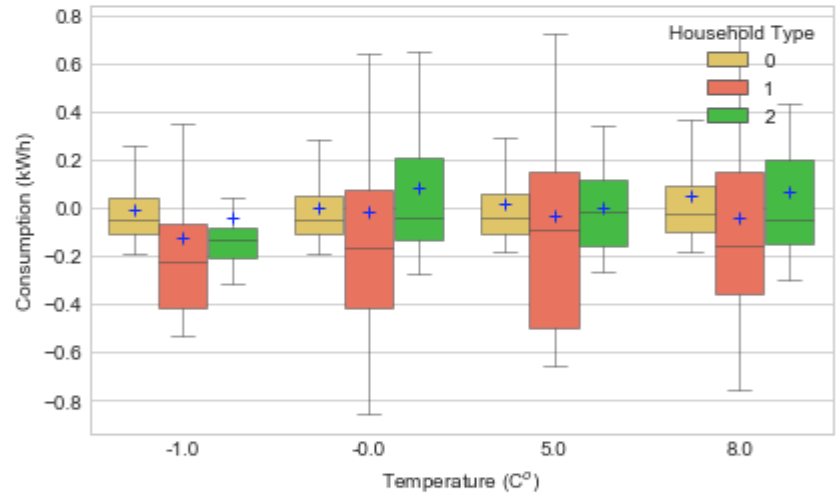

Thursday 05:00 (Price 0.672)
Household Type
0
1
2
0.4
0.2
0.0

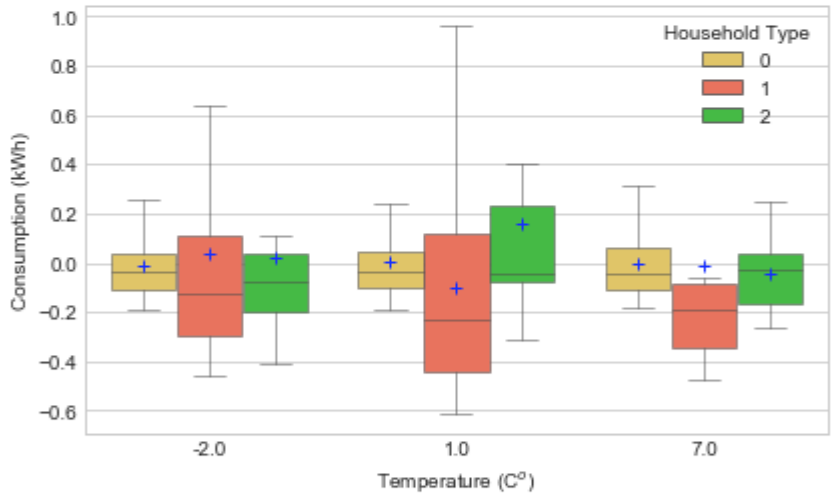

-0.4
-0.6
-3.0
-1.0
5.0
7.0
Temperature (C°)

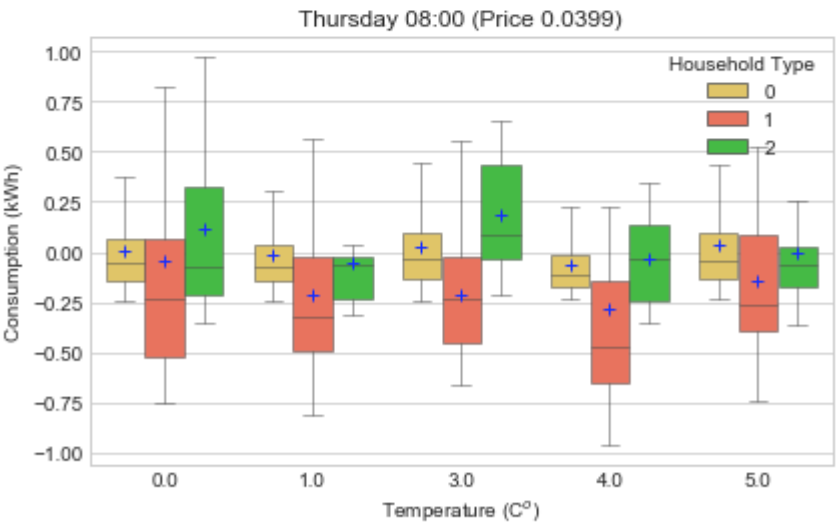

-0.4
-0.6
-4.0
7.0
Temperature (C°)

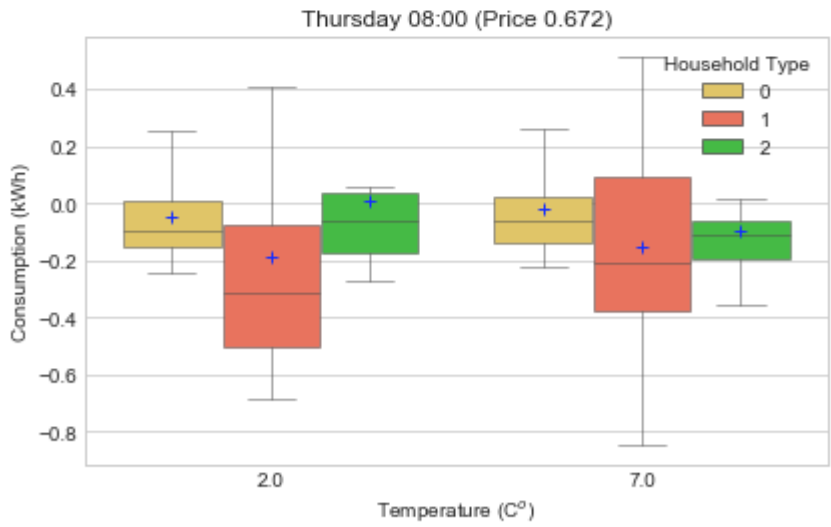

Thursday 06:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-3.0
-1.0
5.0
7.0
Temperature (C°)

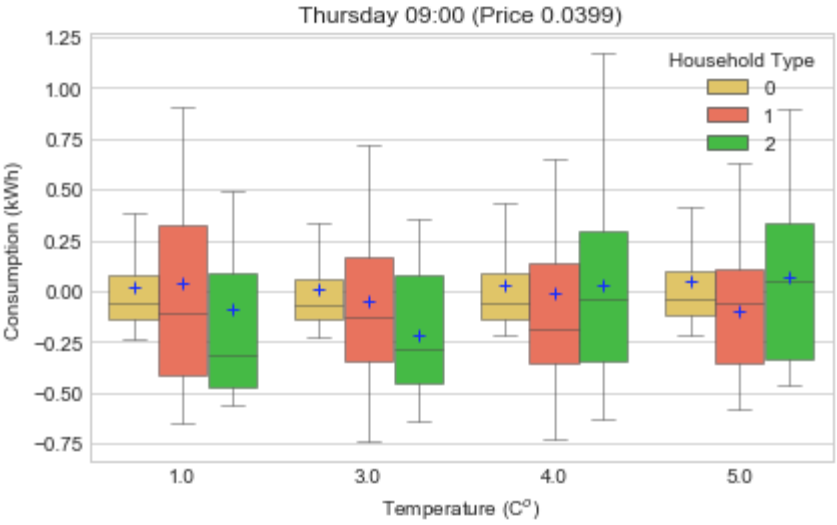

Thursday 06:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.2
0.0
-0.2
-0.4
-0.6
-4.0
7.0
Temperature (C°)

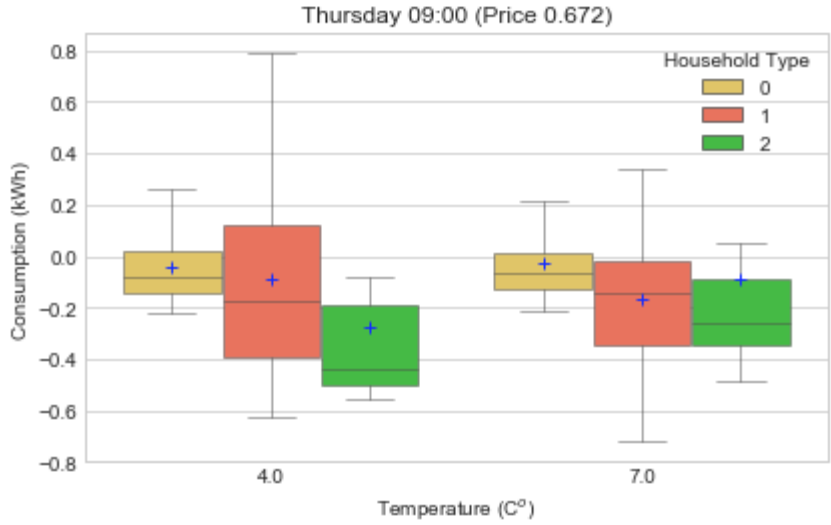

Thursday 07:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-0.0
5.0
8.0
Temperature (C°)

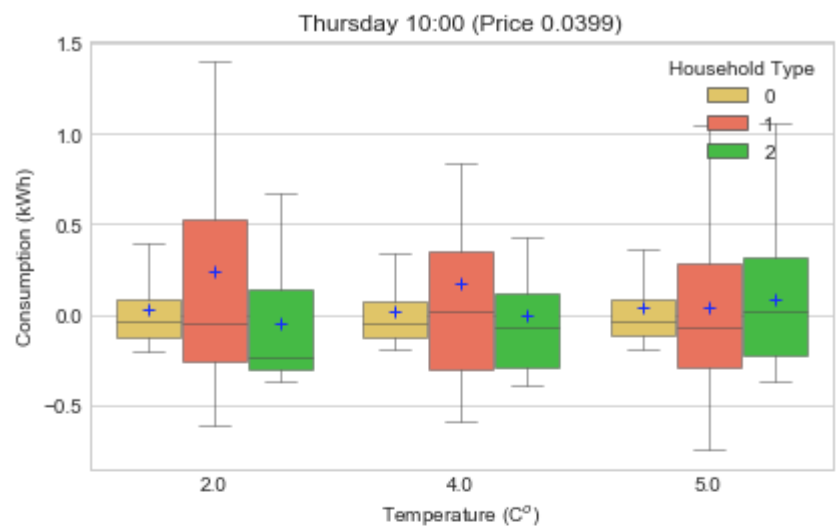

Thursday 07:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-2.0
1.0
7.0
Temperature (C°)

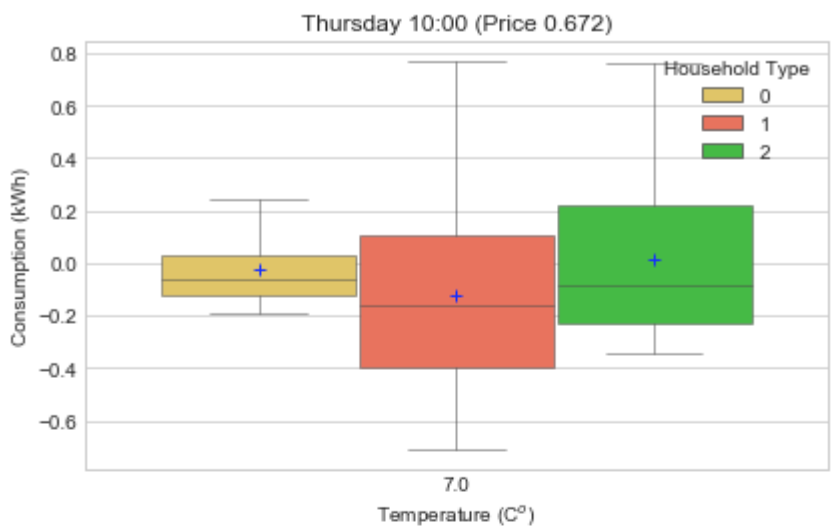

Thursday 08:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00
0.0
1.0
3.0
4.0
5.0
Temperature (C°)


Thursday 08:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
2.0
7.0
Temperature (C°)


Thursday 09:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
1.0
3.0
4.0
5.0
Temperature (C°)

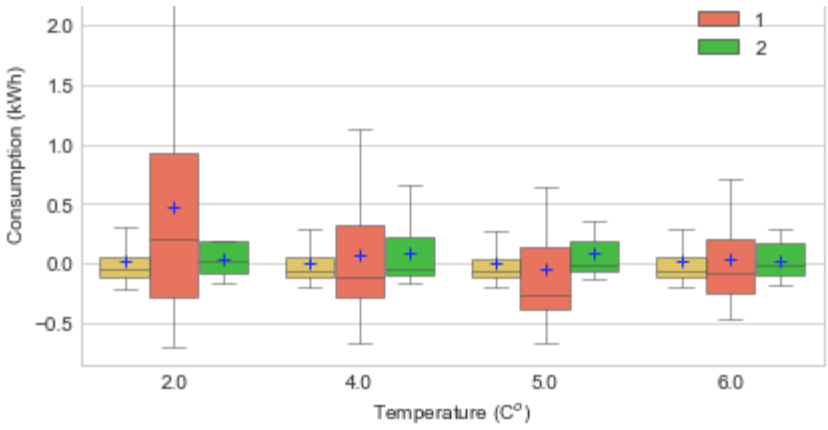

Thursday 09:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
4.0
7.0
Temperature (C°)

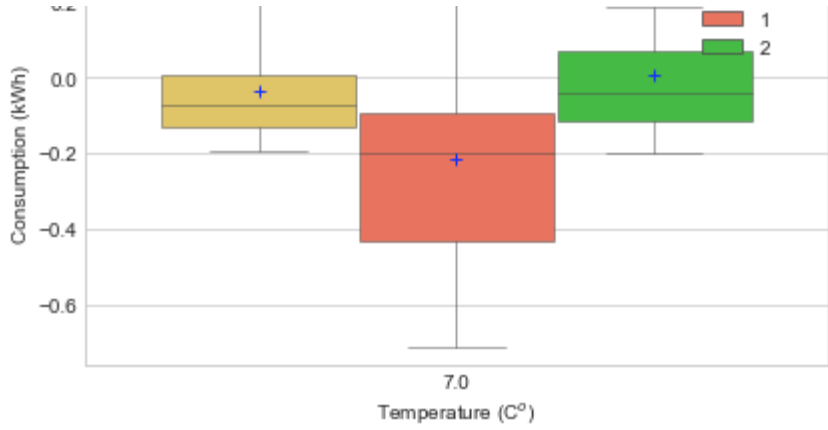

Thursday 10:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
-0.5
2.0
4.0
5.0
Temperature (C°)

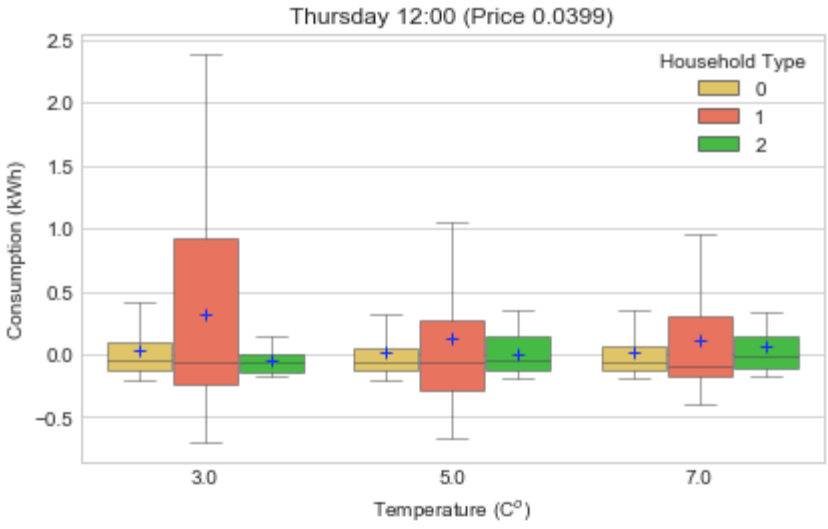

Thursday 10:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
7.0
Temperature (C°)

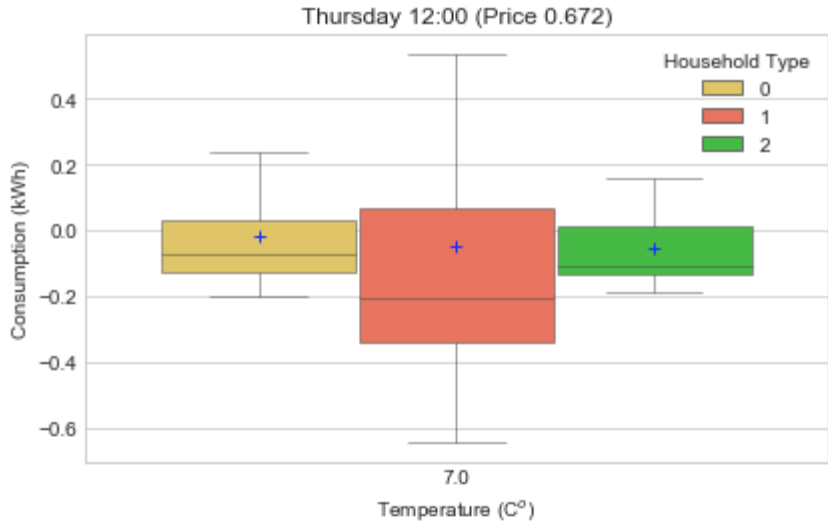
Thursday 11:00 (Price 0.0399)
Household Type
0
2.5

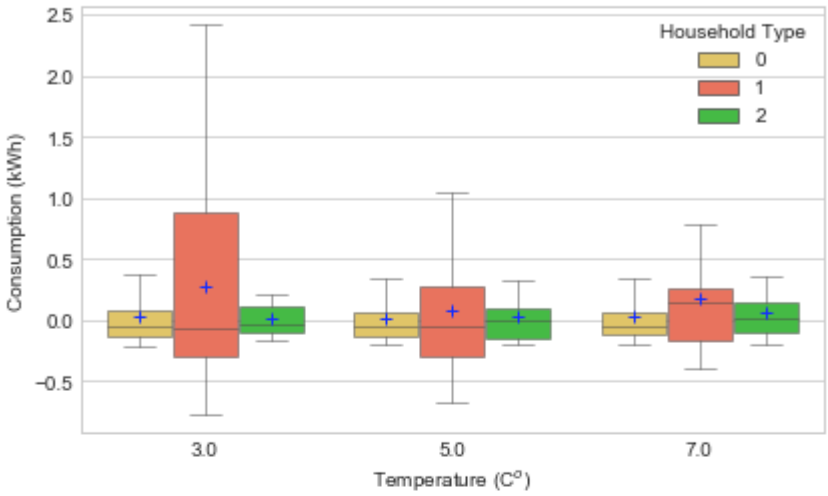
Thursday 11:00 (Price 0.672)
Household Type
0

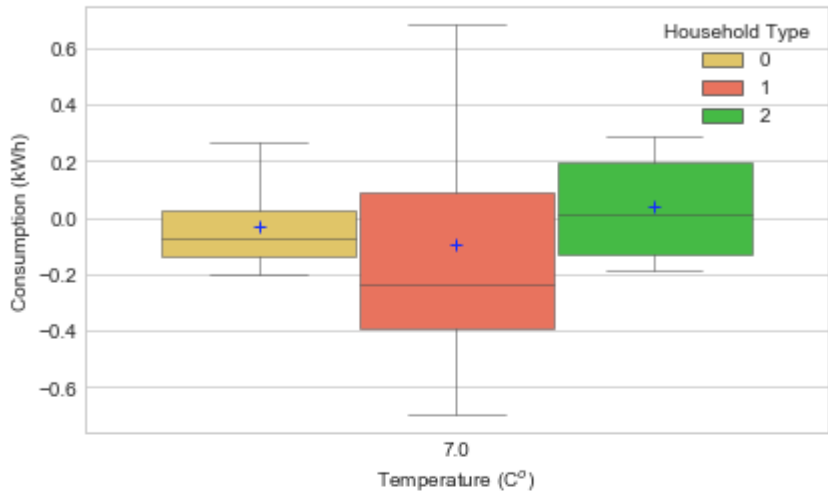

1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
2.0
4.0
5.0
6.0
Temperature (C°)

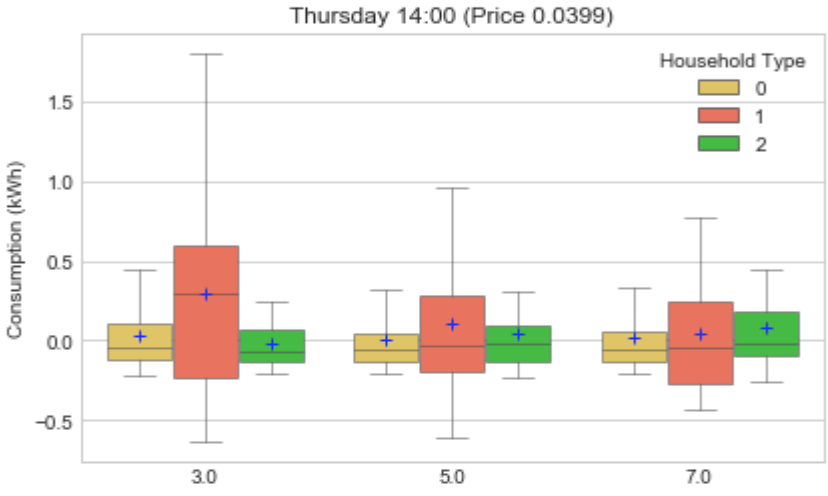

1
2
Consumption (kWh)
0.0
−0.2
−0.4
−0.6
7.0
Temperature (C°)

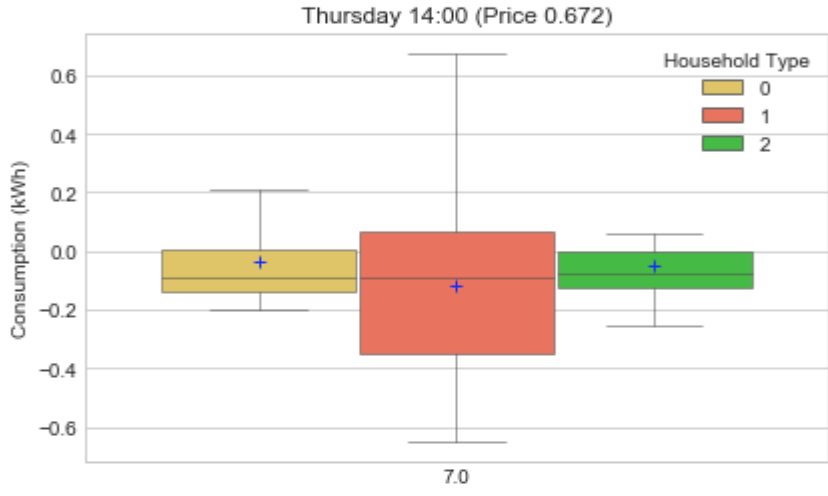

Thursday 12:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
3.0
5.0
7.0
Temperature (C°)

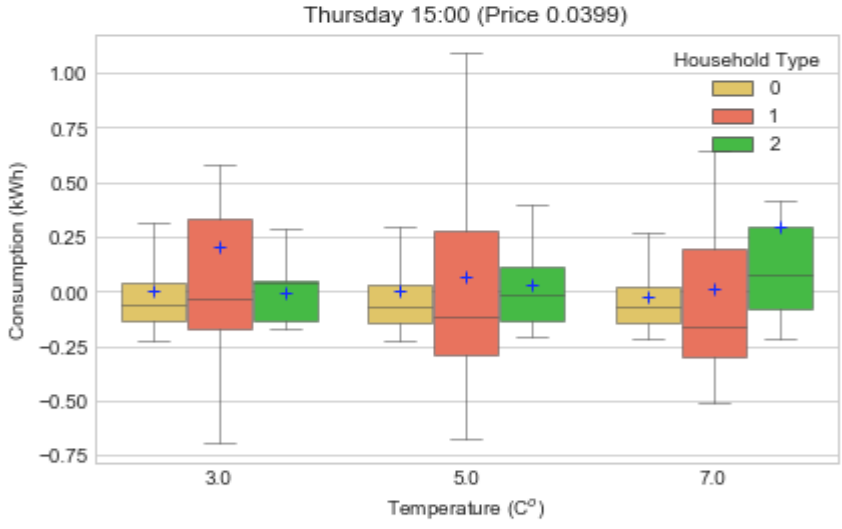

Thursday 12:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
7.0
Temperature (C°)

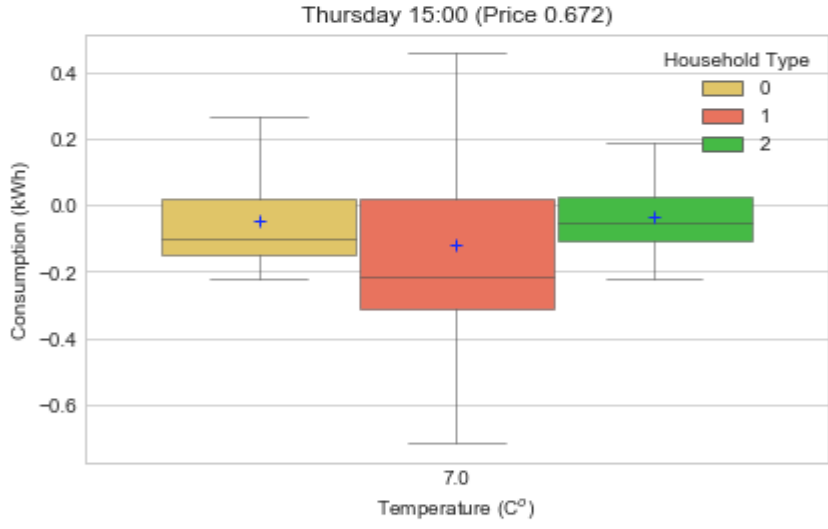

Thursday 13:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
3.0
5.0
7.0
Temperature (C°)

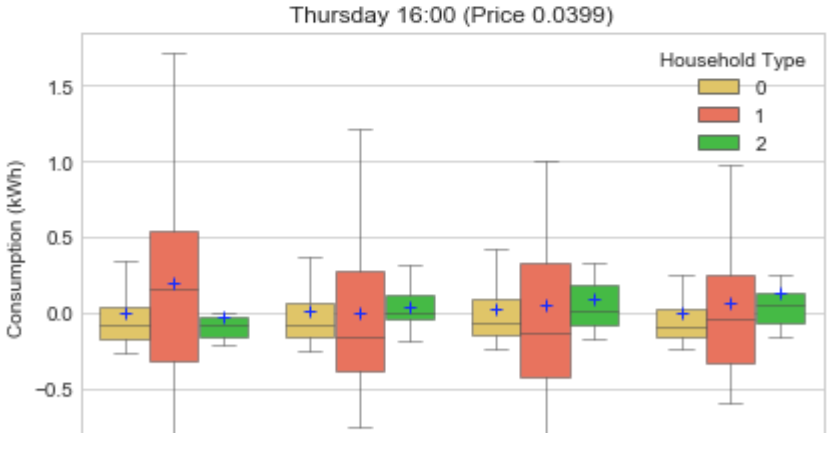

Thursday 13:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
7.0
Temperature (C°)

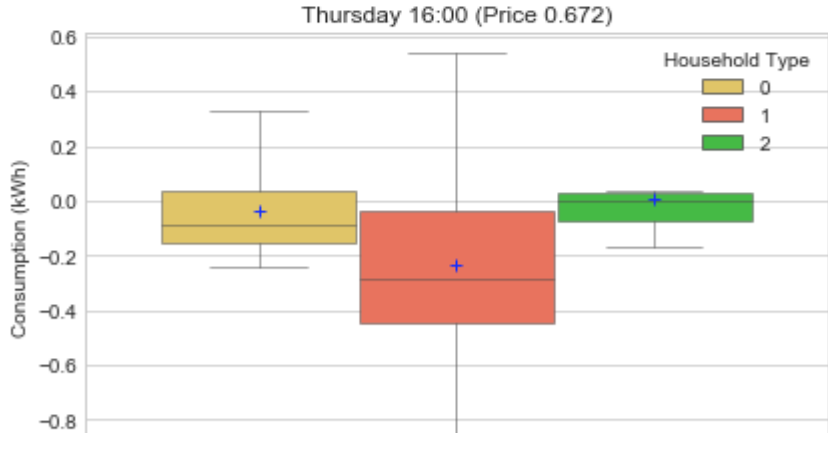

Thursday 14:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
3.0
5.0
7.0
Temperature (C°)

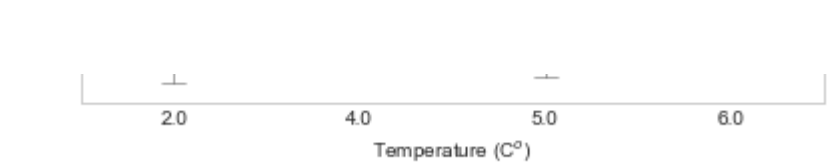

Thursday 14:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
7.0
Temperature (C°)

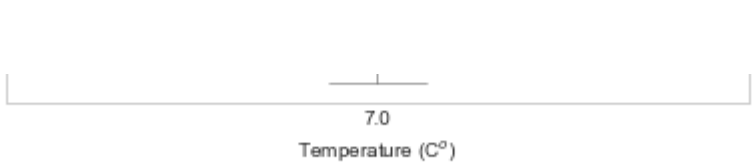

Thursday 15:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
5.0
7.0
Temperature (C°)

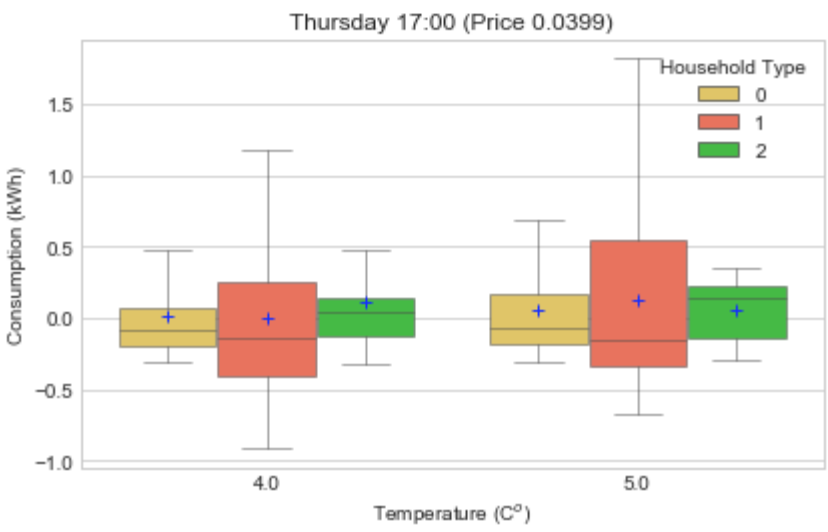

Thursday 15:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
7.0
Temperature (C°)

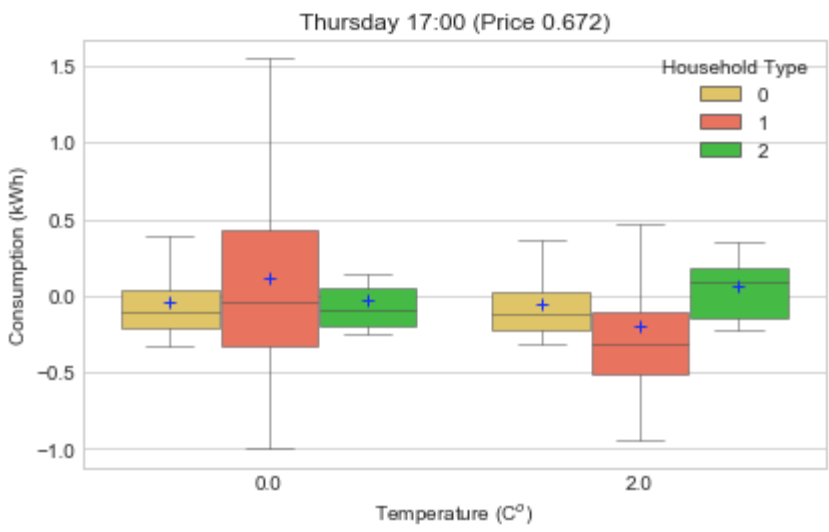

Thursday 16:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5

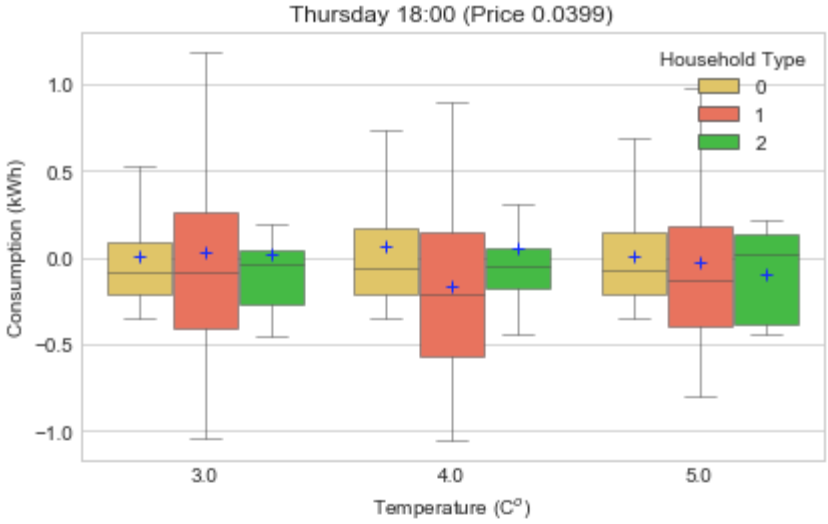

Thursday 16:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8

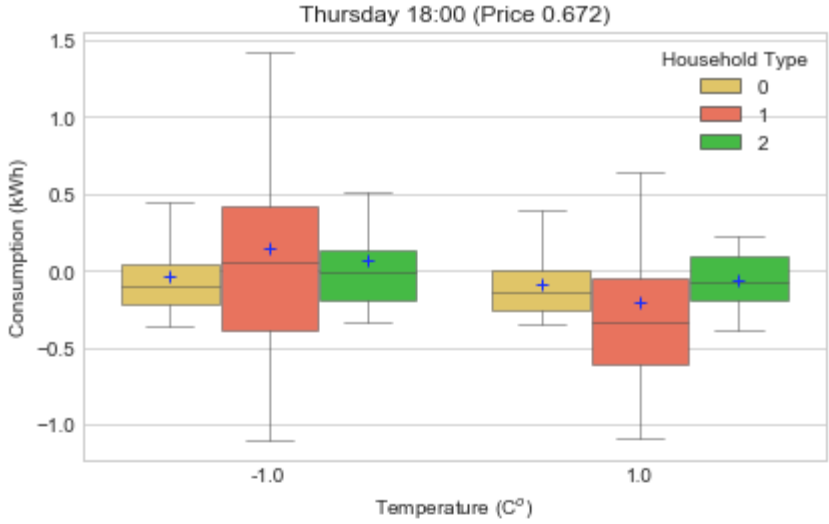

2.0
4.0
5.0
6.0
Temperature (C°)

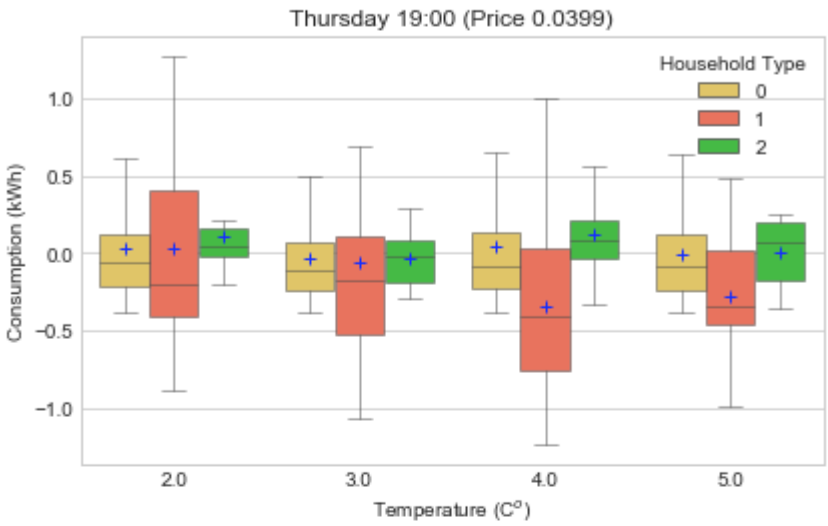

7.0
Temperature (C°)

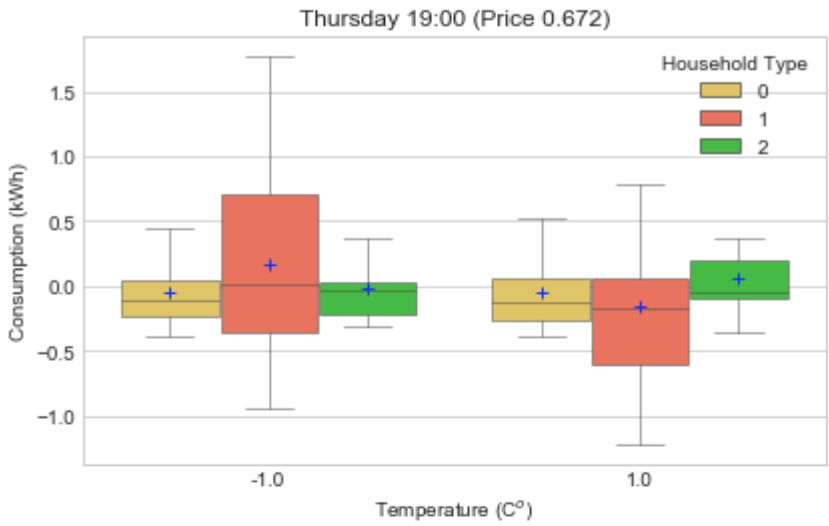

Thursday 17:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
4.0
5.0
Temperature (C°)

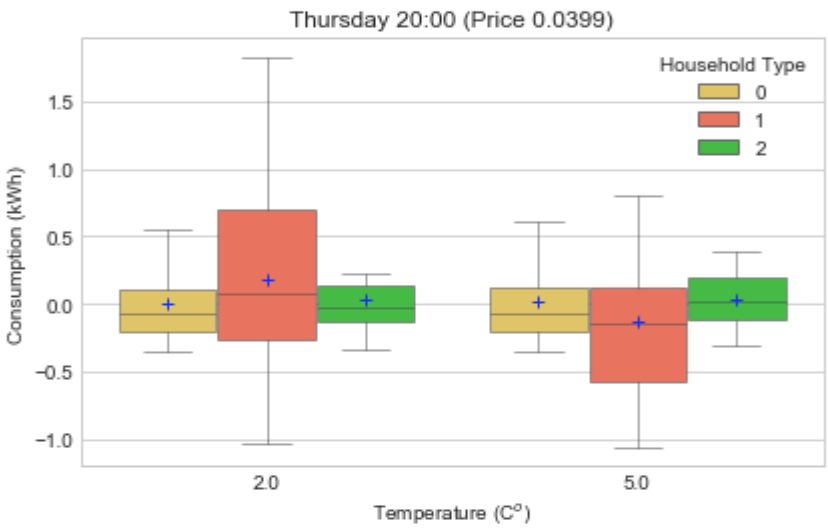

Thursday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
2.0
Temperature (C°)

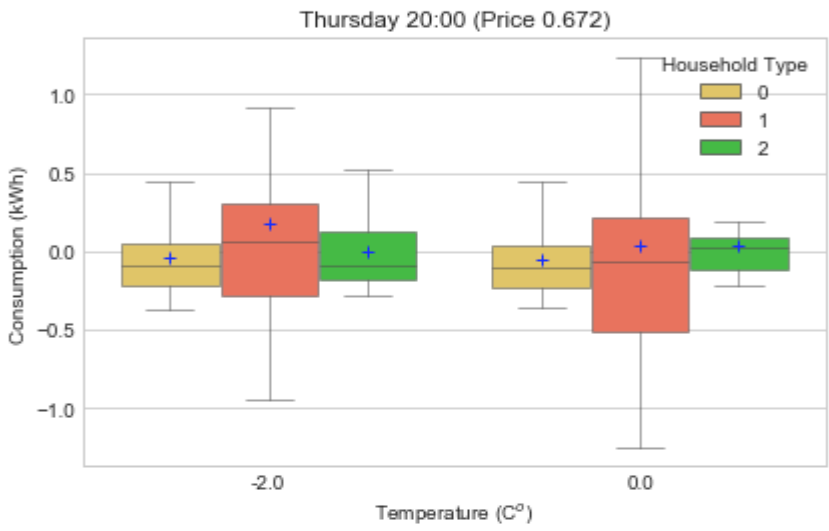

Thursday 18:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
3.0
4.0
5.0
Temperature (C°)

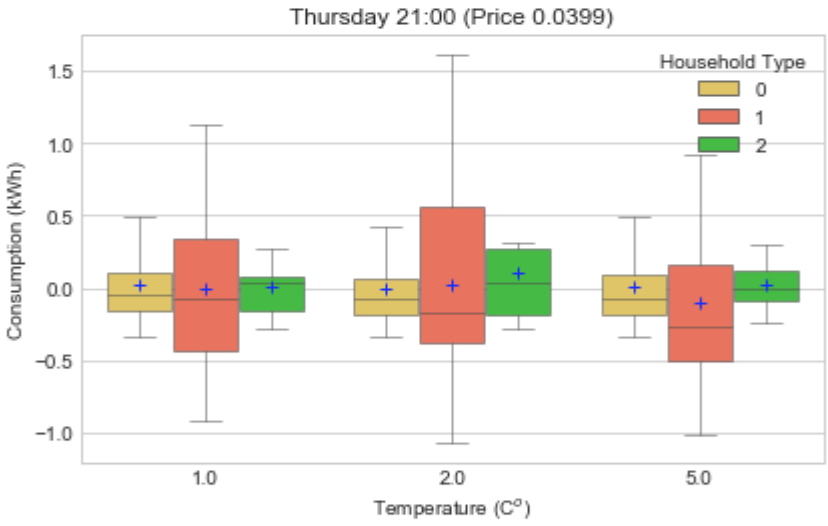

Thursday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
1.0
Temperature (C°)

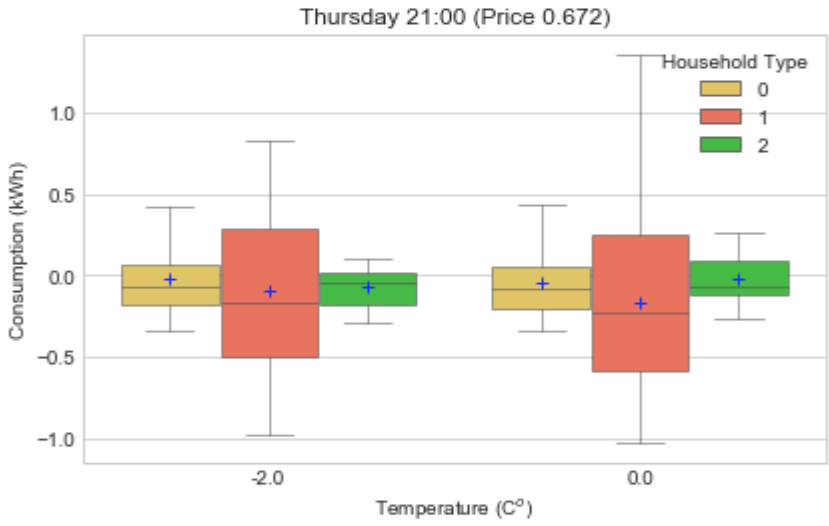

Thursday 19:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
3.0
4.0
5.0
Temperature (C°)


Thursday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
1.0
Temperature (C°)


Thursday 20:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
2.0
5.0
Temperature (C°)

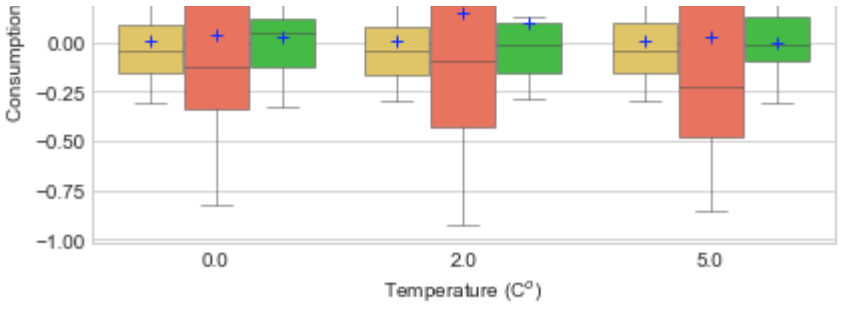

Thursday 20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-2.0
0.0
Temperature (C°)

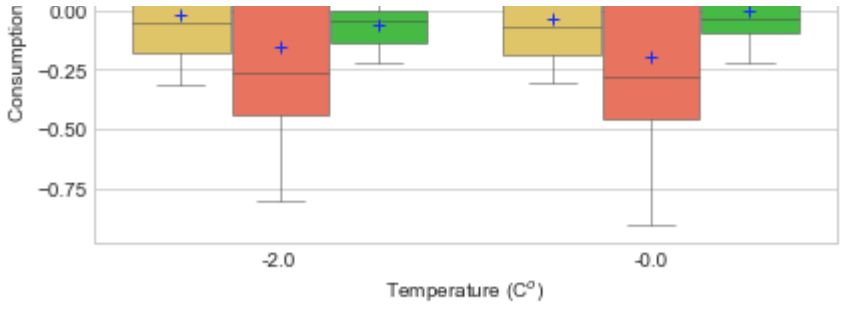

Thursday 21:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
1.0
2.0
5.0
Temperature (C°)

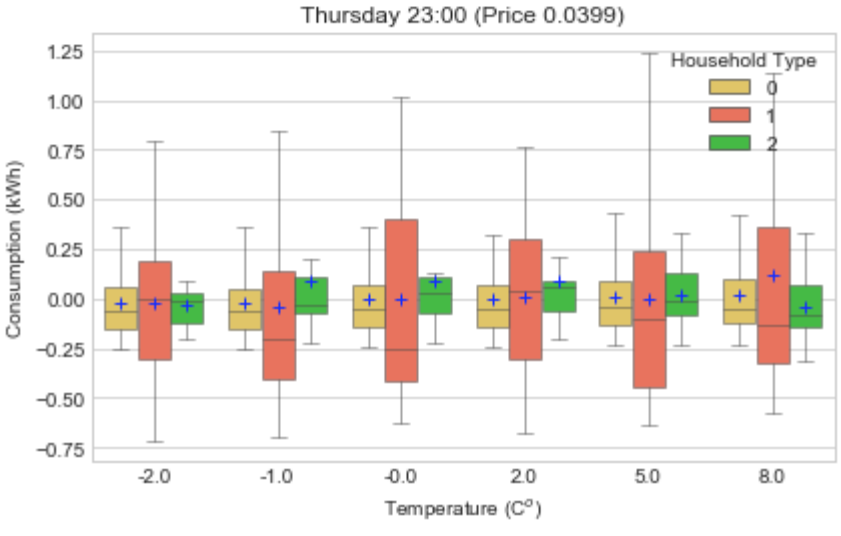

Thursday 21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-2.0
0.0
Temperature (C°)

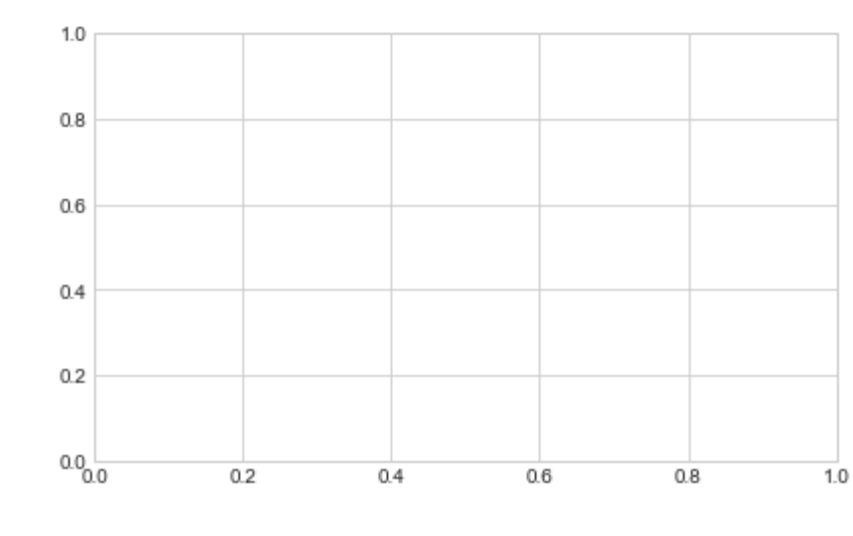
Thursday 22:00 (Price 0.0399)
Household Type
0
1
2
1.00
0.75
0.50
0.25

Thursday 22:00 (Price 0.672)
Household Type
0
1
2
0.75
0.50
0.25

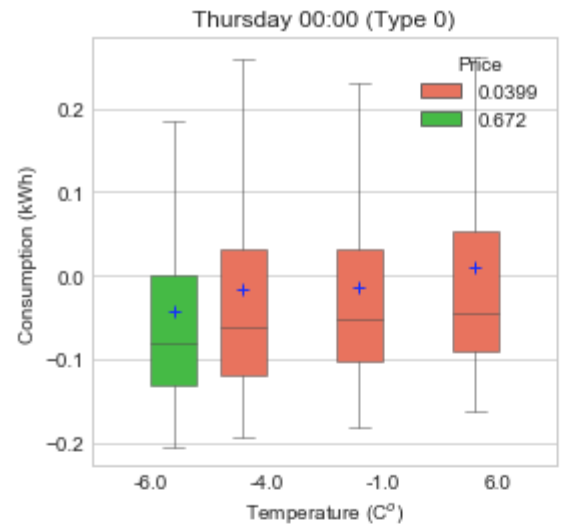

Consumption
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.0
5.0
Temperature (C°)

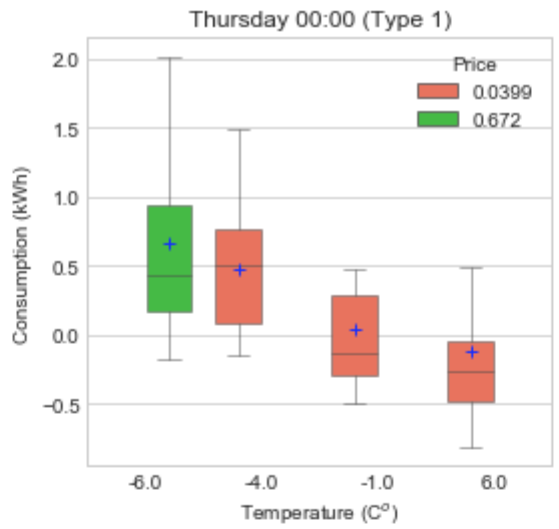

Consumption
0.00
−0.25
−0.50
−0.75
-2.0
-0.0
Temperature (C°)

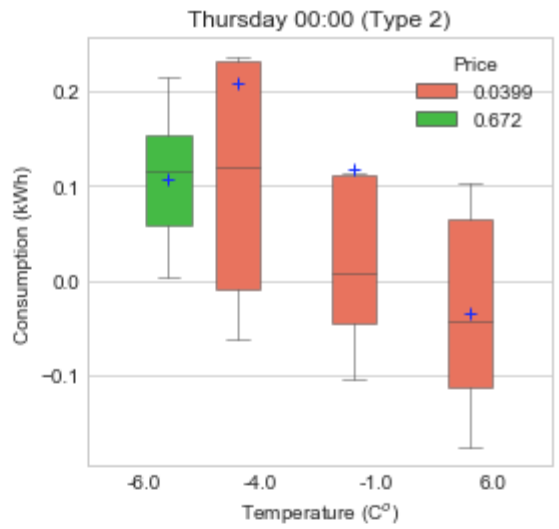

Thursday 23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-2.0
-1.0
-0.0
2.0
5.0
8.0
Temperature (C°)

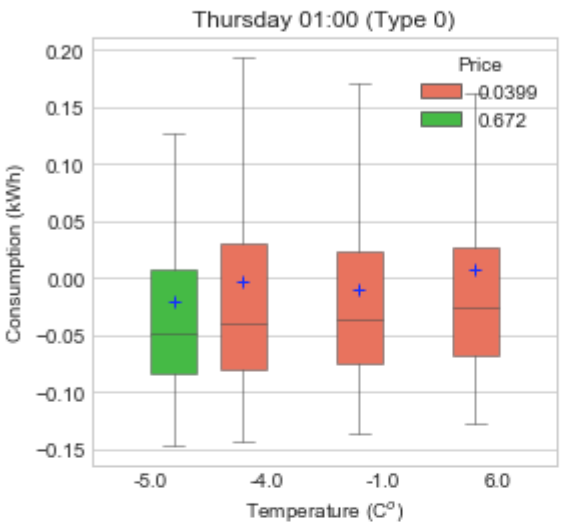

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

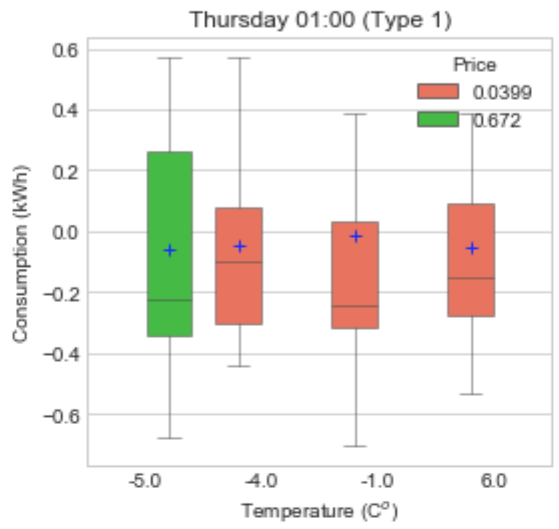
In [131]:
```
# price comparison
# Thursday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 3 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

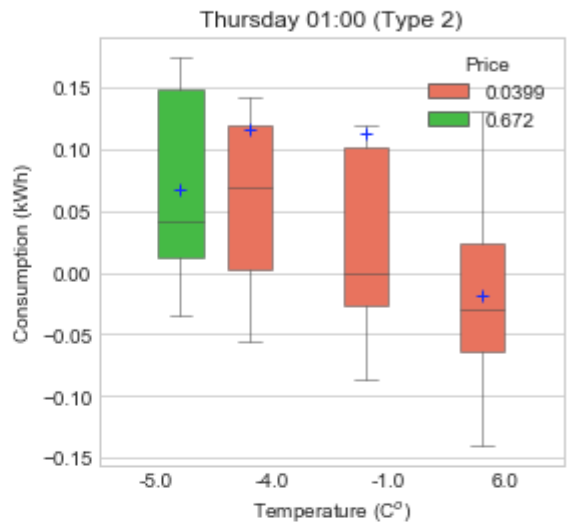

Thursday 00:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-6.0
-4.0
-1.0
6.0

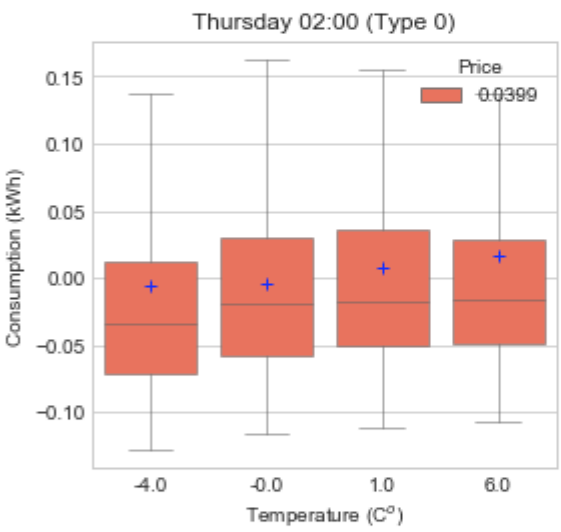

Thursday 00:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-6.0
-4.0
-1.0
6.0

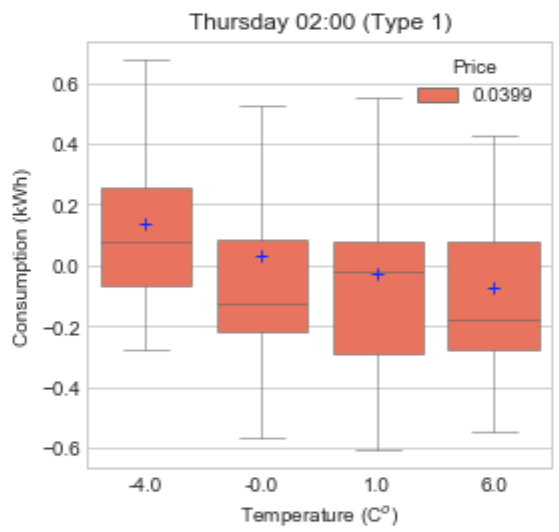

Thursday 00:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-6.0
-4.0
-1.0
6.0

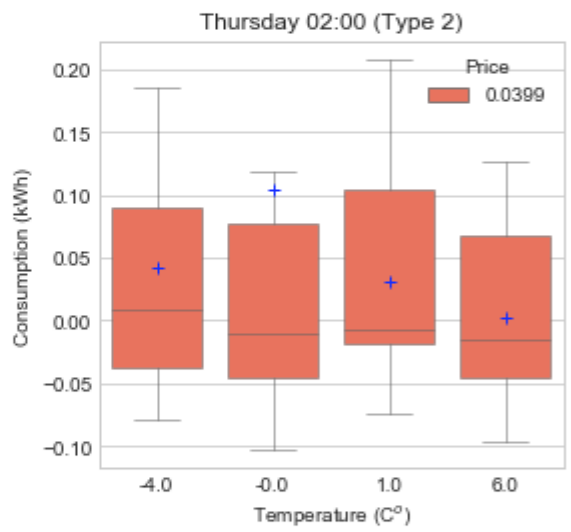

Thursday 01:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-5.0
-4.0
-1.0
6.0

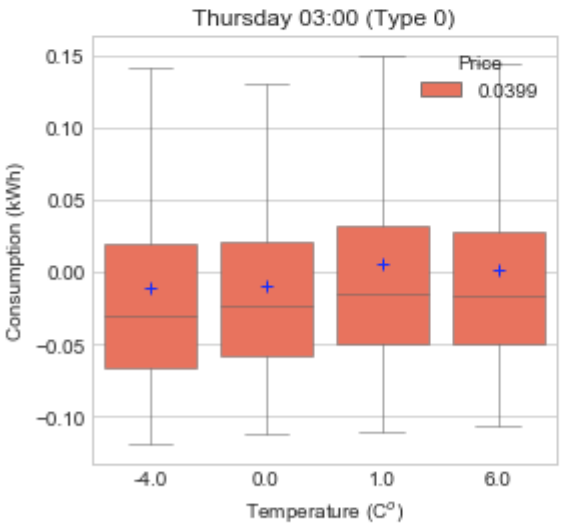

Thursday 01:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-5.0
-4.0
-1.0
6.0

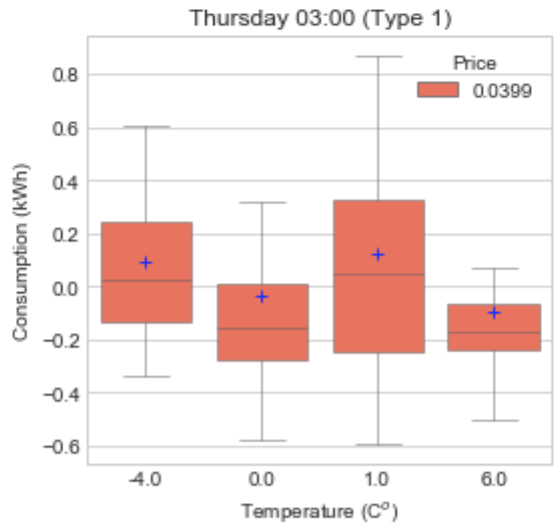

Thursday 01:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-5.0
-4.0
-1.0
6.0

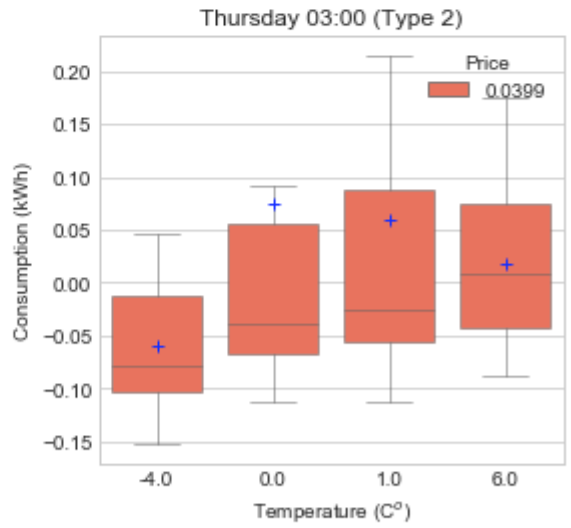

Thursday 02:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
-0.0
1.0
6.0

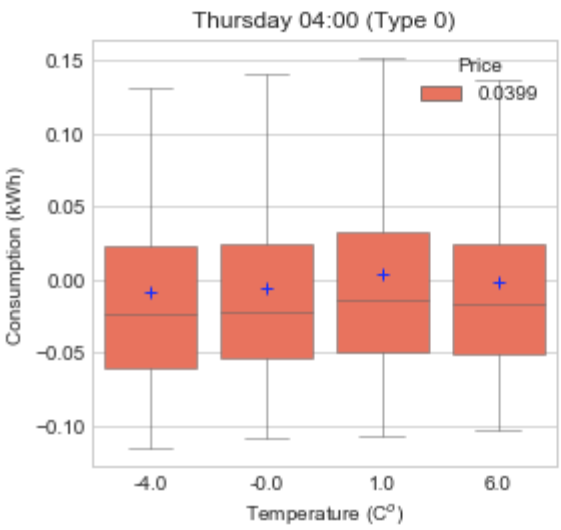

Thursday 02:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
-0.0
1.0
6.0

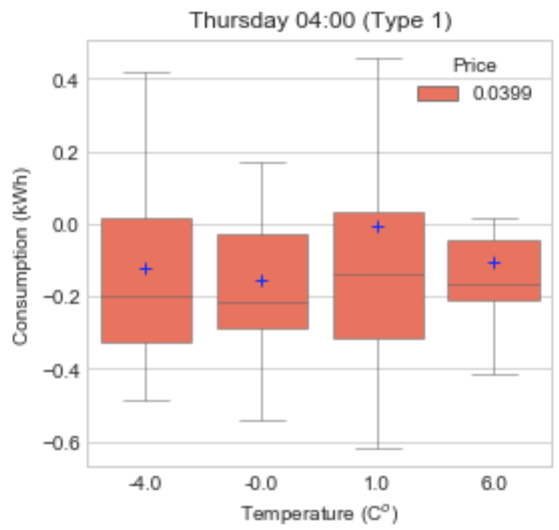

Thursday 02:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
-0.0
1.0
6.0

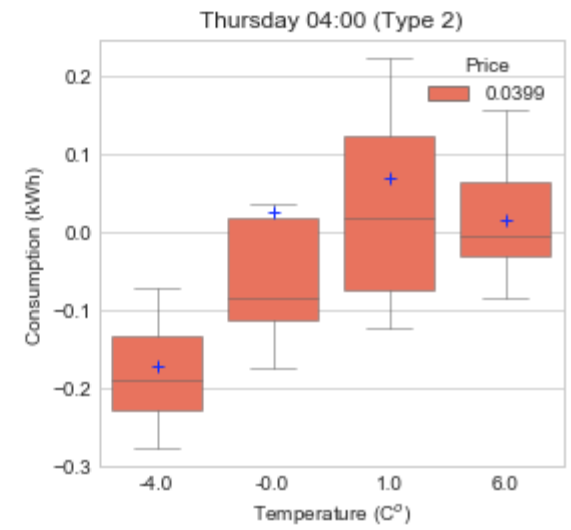

Thursday 03:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
0.0
1.0
6.0

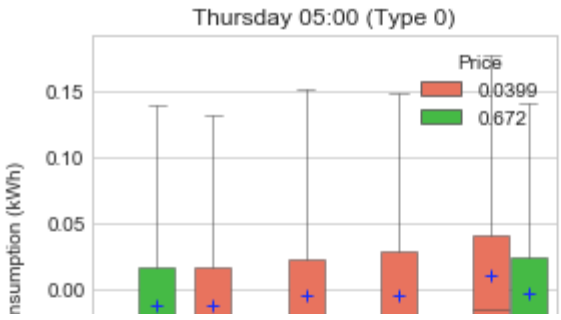

Thursday 03:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
0.0
1.0
6.0

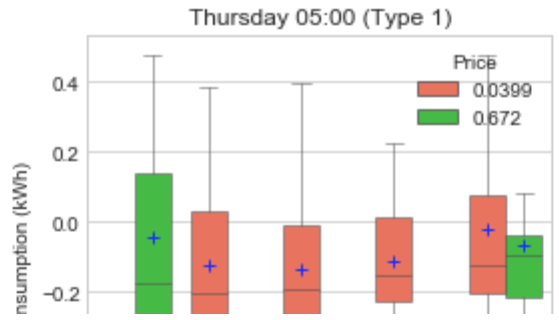

Thursday 03:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
0.0
1.0
6.0

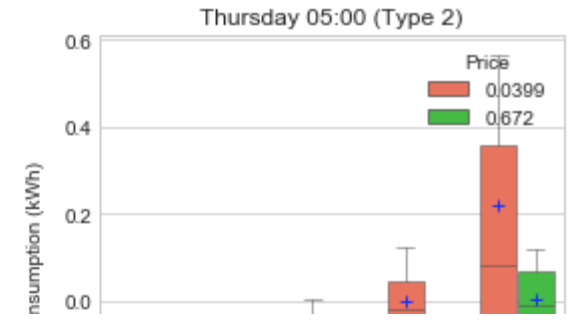

Thursday 04:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
-0.0
1.0
6.0

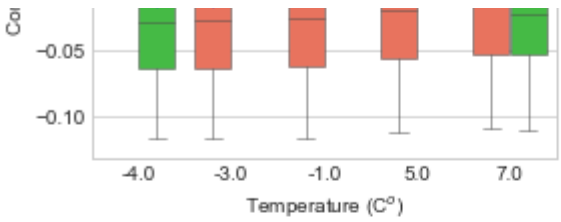

Thursday 04:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
-0.0
1.0
6.0

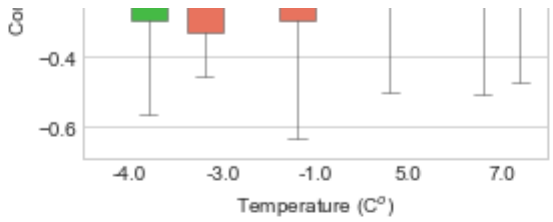

Thursday 04:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-4.0
-0.0
1.0
6.0

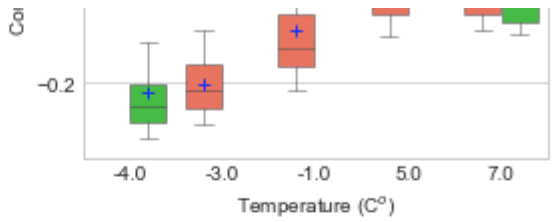

Thursday 05:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)

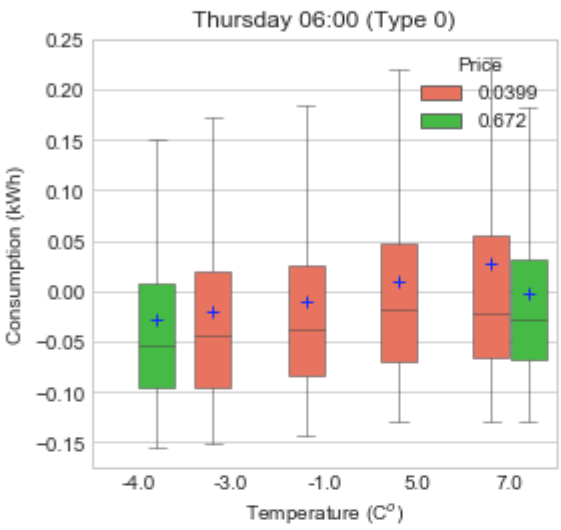

Thursday 05:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)

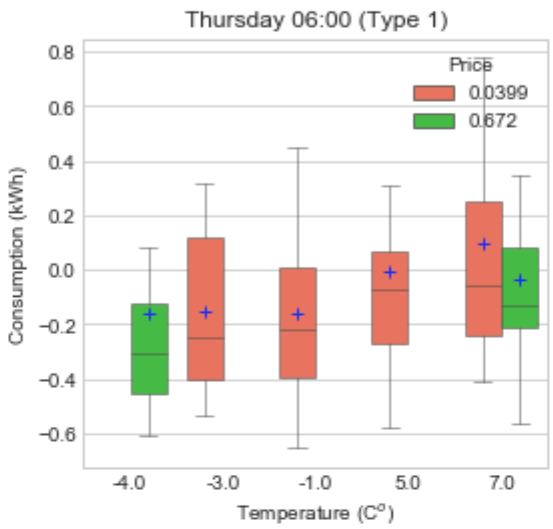

Thursday 05:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)

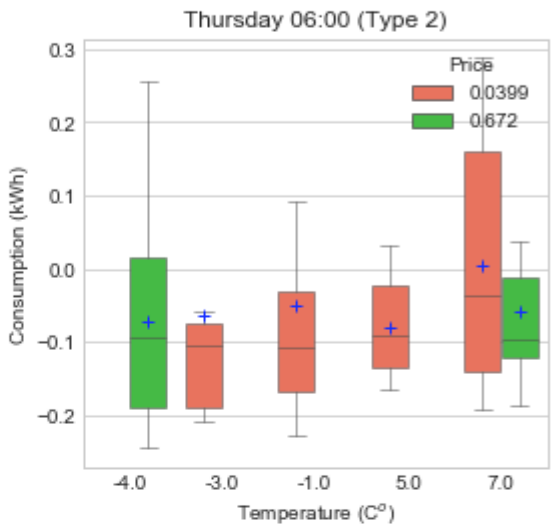

−0.05
−0.10
-4.0
-3.0
-1.0
5.0
7.0
Temperature (C°)

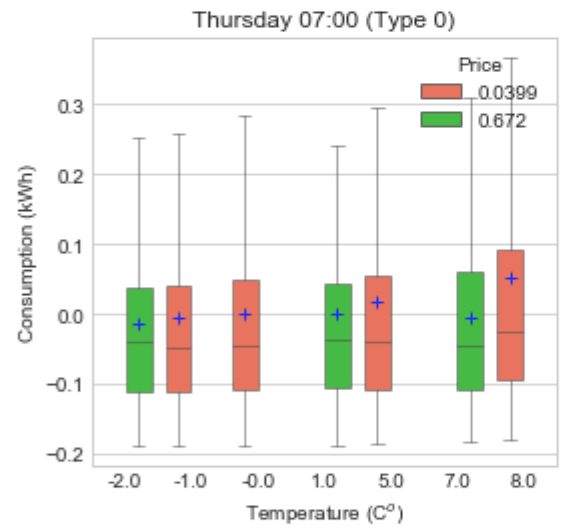

−0.4
−0.6
-4.0
-3.0
-1.0
5.0
7.0
Temperature (C°)

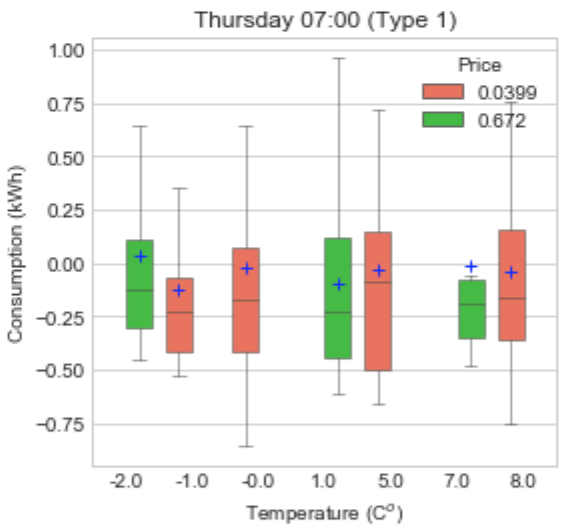

−0.2
-4.0
-3.0
-1.0
5.0
7.0
Temperature (C°)

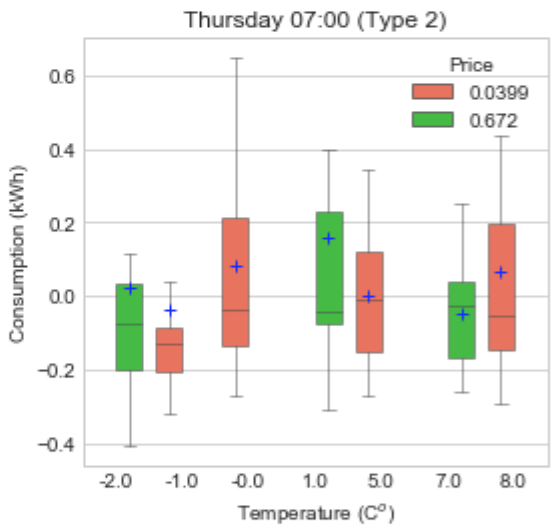

Thursday 06:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
-4.0
-3.0
-1.0
5.0
7.0
Temperature (C°)

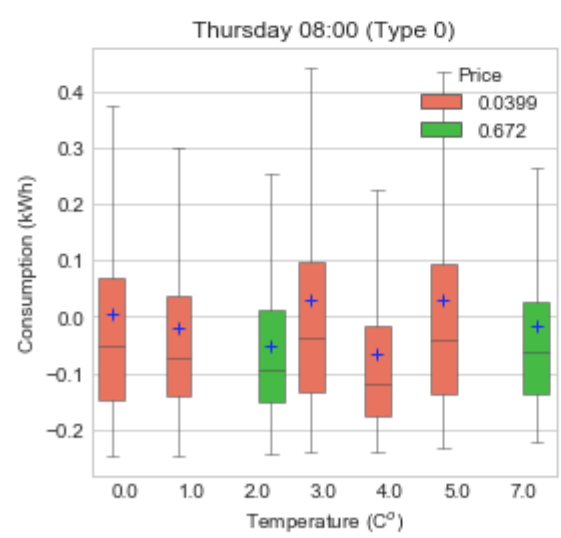

Thursday 06:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
-3.0
-1.0
5.0
7.0
Temperature (C°)

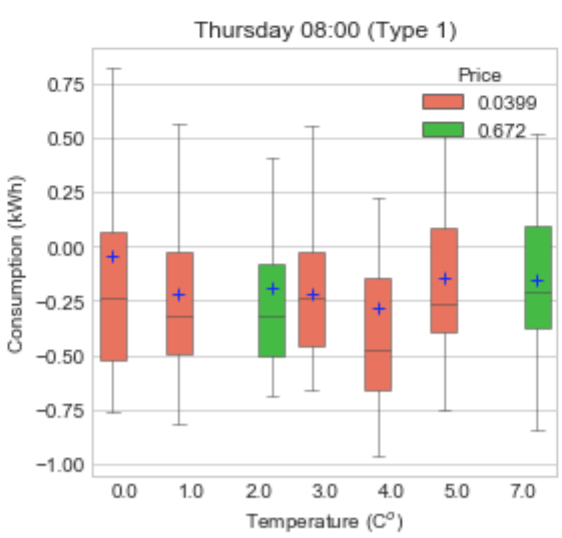

Thursday 06:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
-4.0
-3.0
-1.0
5.0
7.0
Temperature (C°)

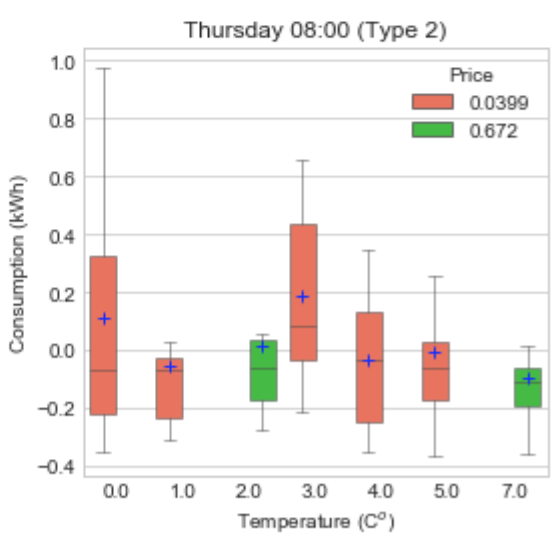

Thursday 07:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
-2.0
-1.0
-0.0
1.0
5.0
7.0
8.0
Temperature (C°)

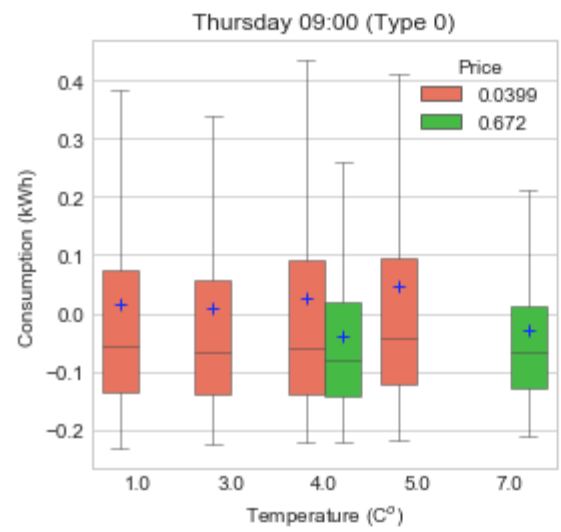

Thursday 07:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-2.0
-1.0
-0.0
1.0
5.0
7.0
8.0
Temperature (C°)

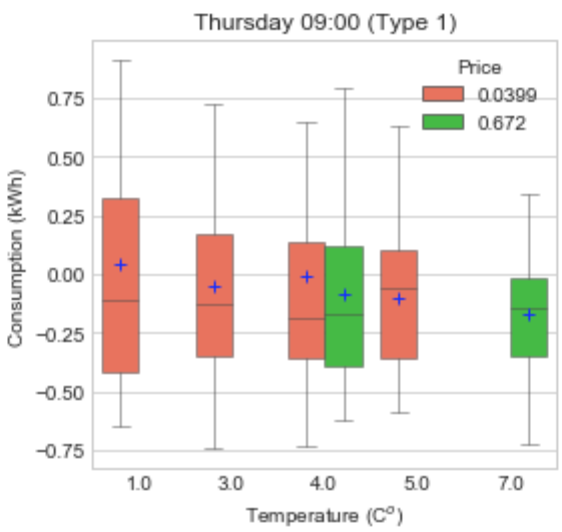

Thursday 07:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0
-1.0
-0.0
1.0
5.0
7.0
8.0
Temperature (C°)

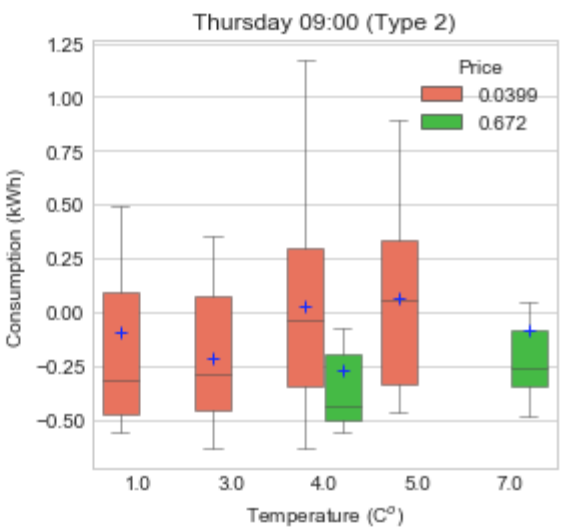

Thursday 08:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
1.0
2.0
3.0
4.0
5.0
7.0
Temperature (C°)

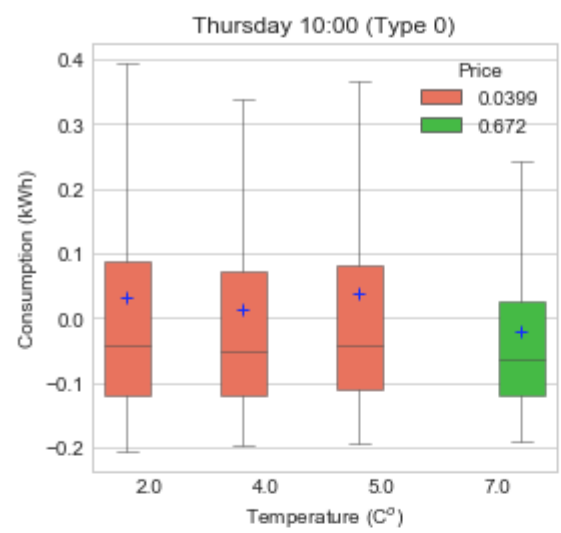

Thursday 08:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
1.0
2.0
3.0
4.0
5.0
7.0
Temperature (C°)

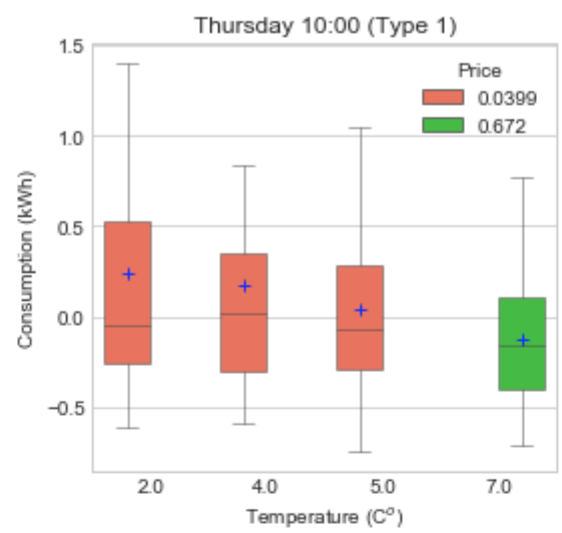

Thursday 08:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
1.0
2.0
3.0
4.0
5.0
7.0
Temperature (C°)

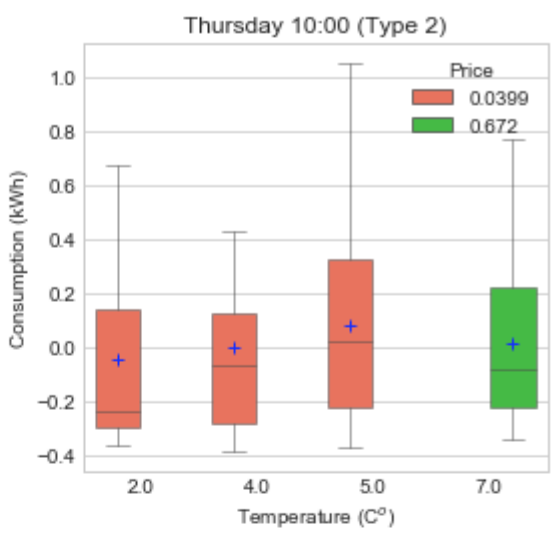

Thursday 09:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
1.0
3.0
4.0
5.0
7.0
Temperature (C°)

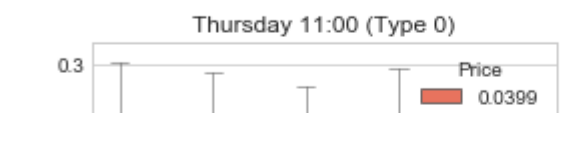

Thursday 09:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
1.0
3.0
4.0
5.0
7.0
Temperature (C°)

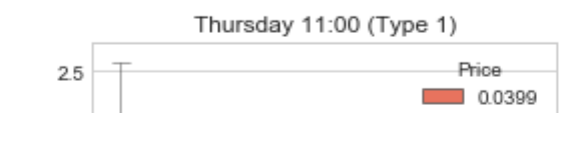

Thursday 09:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
1.0
3.0
4.0
5.0
7.0
Temperature (C°)


Thursday 10:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.0
4.0
5.0
7.0
Temperature (C°)

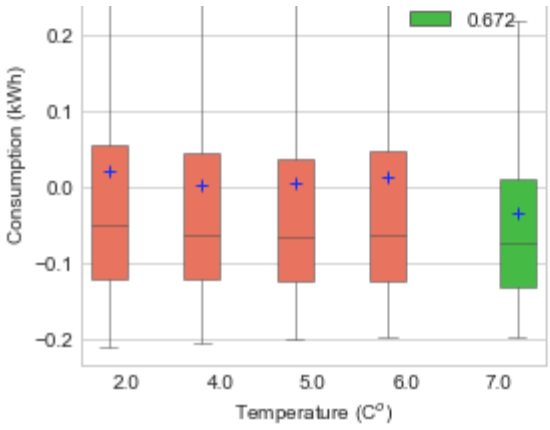

Thursday 10:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
2.0
4.0
5.0
7.0
Temperature (C°)

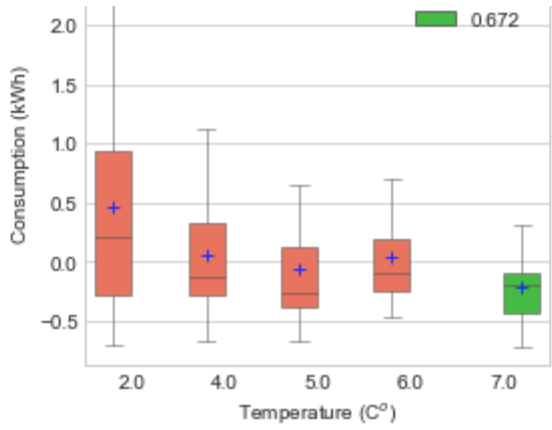

Thursday 10:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
2.0
4.0
5.0
7.0
Temperature (C°)

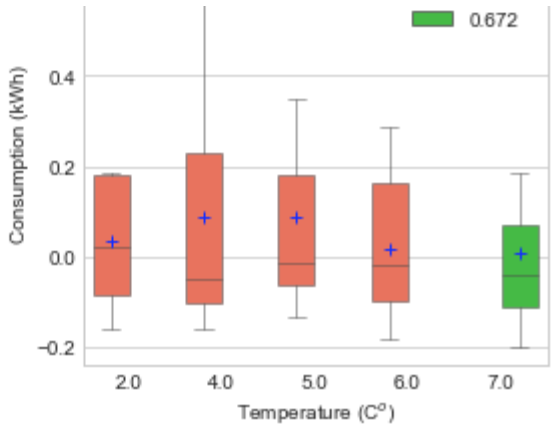

Thursday 11:00 (Type 0)
Price
0.0399
0.3

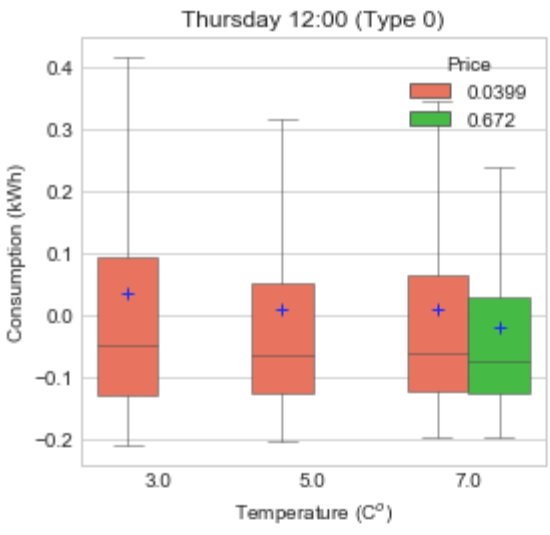

Thursday 11:00 (Type 1)
Price
0.0399
2.5

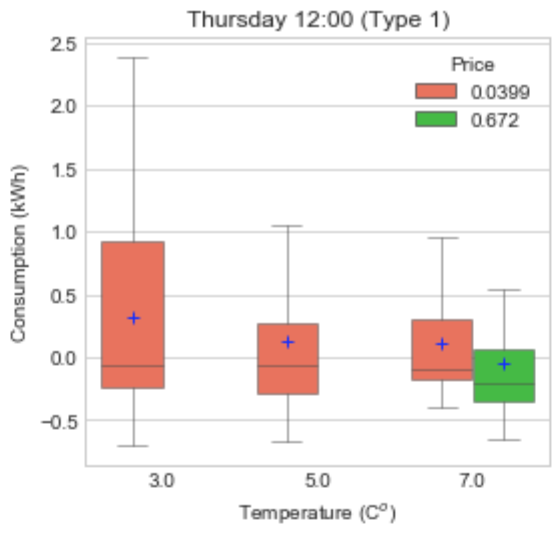

Thursday 11:00 (Type 2)
Price
0.0399
0.6

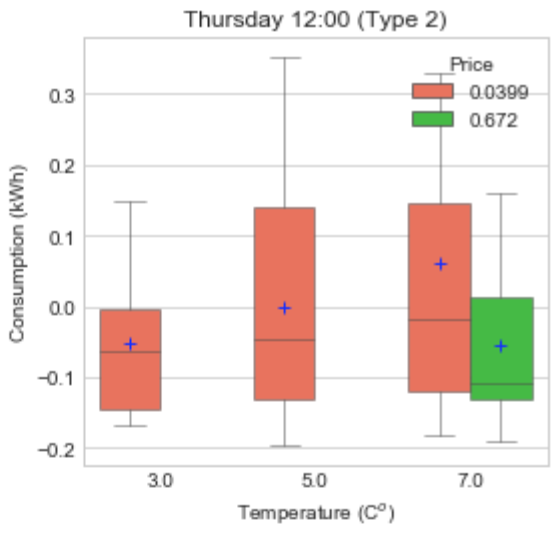

0.672
Consumption (kWh)
Temperature (C°)
2.0
4.0
5.0
6.0
7.0

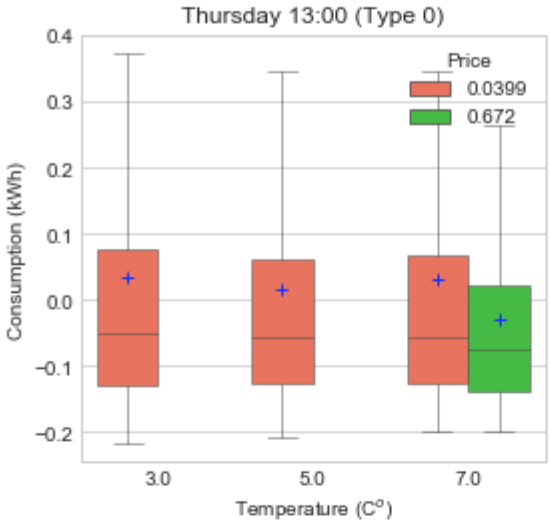

0.672
Consumption (kWh)
Temperature (C°)
2.0
4.0
5.0
6.0
7.0

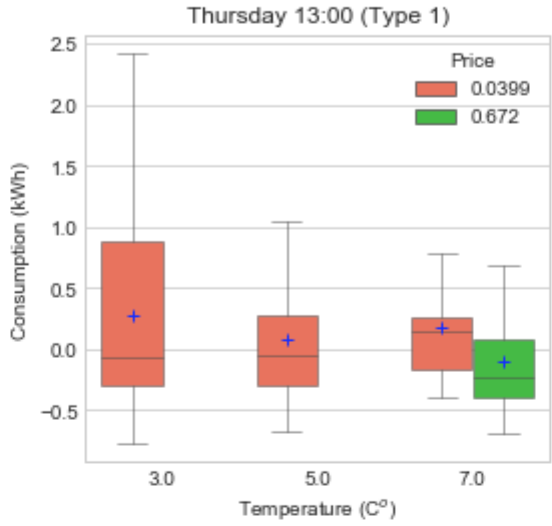

0.672
Consumption (kWh)
Temperature (C°)
2.0
4.0
5.0
6.0
7.0

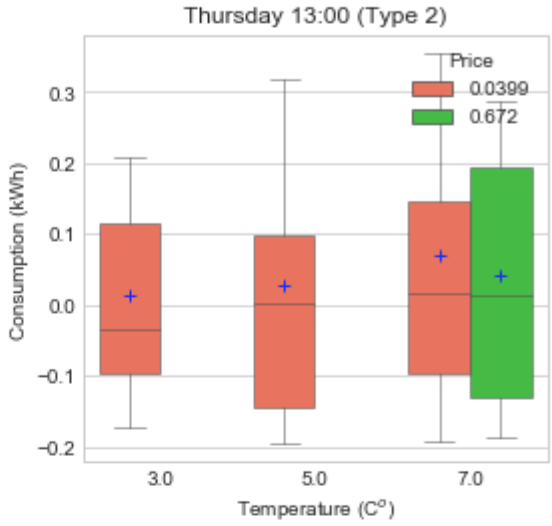

Thursday 12:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

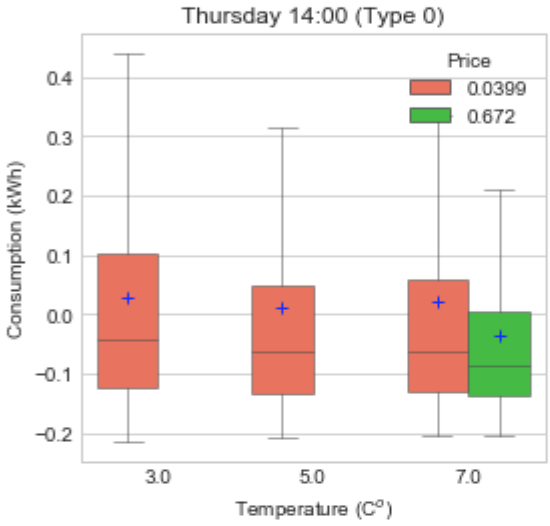

Thursday 12:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

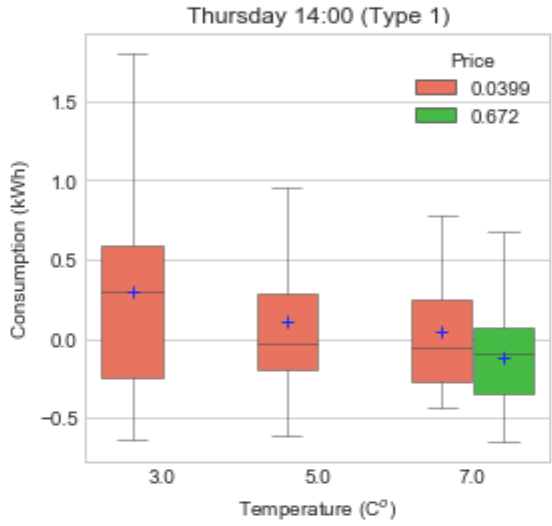

Thursday 12:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

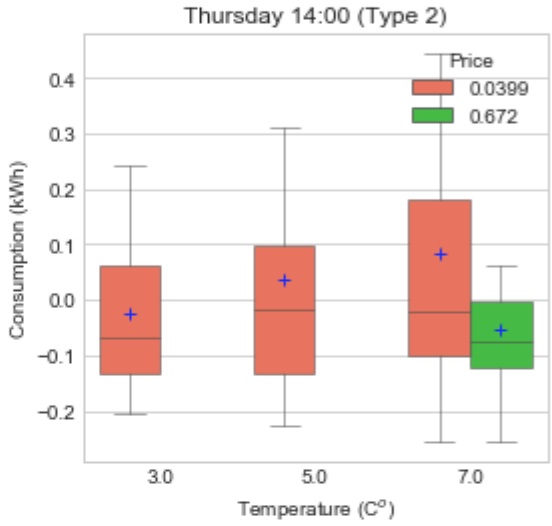

Thursday 13:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

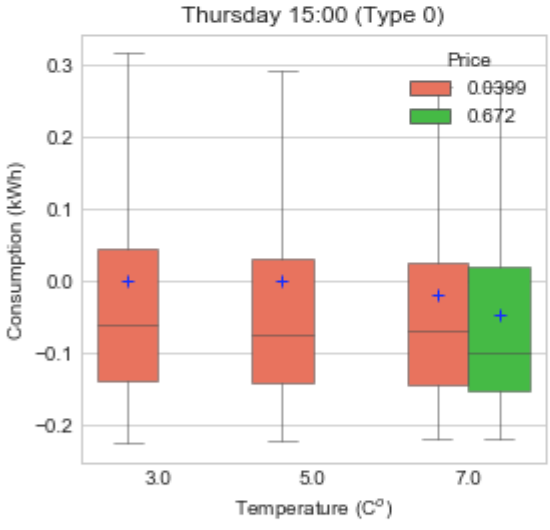

Thursday 13:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

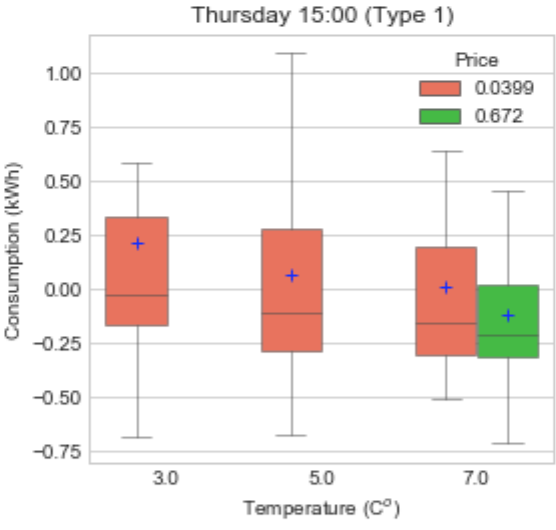

Thursday 13:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

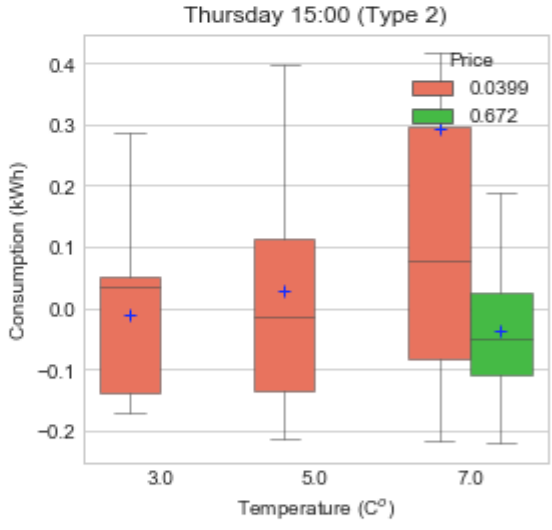

Thursday 14:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

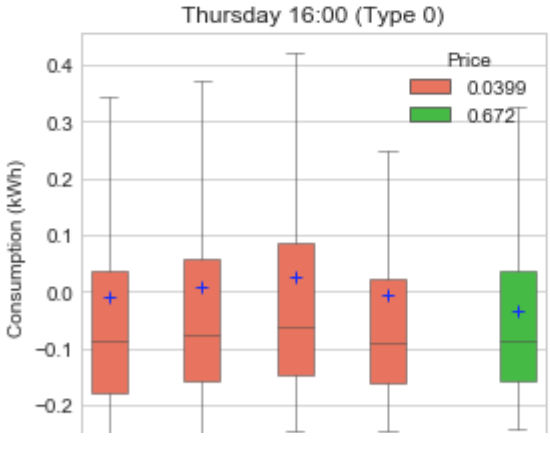

Thursday 14:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

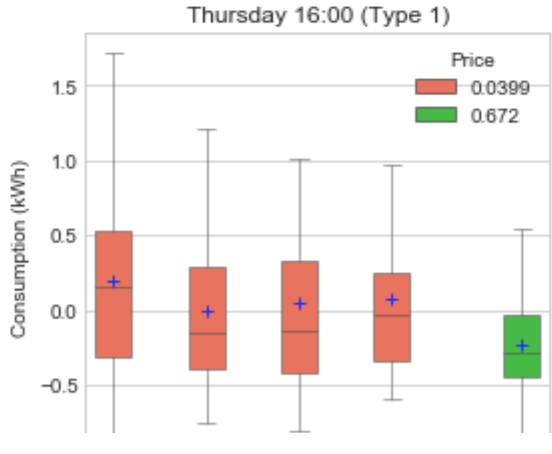

Thursday 14:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

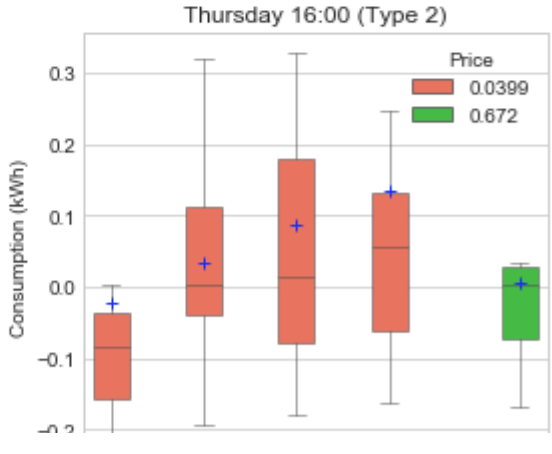

Thursday 15:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

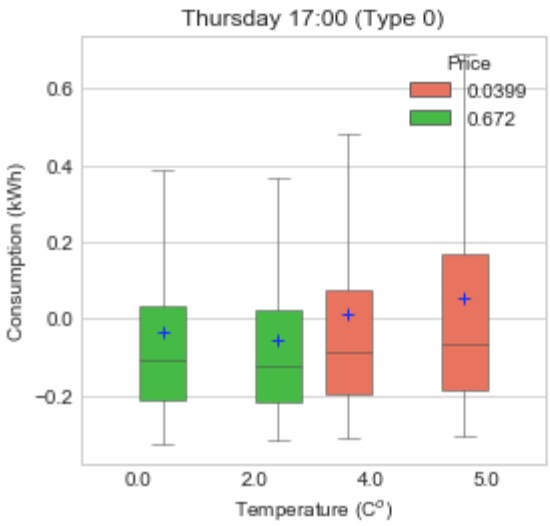

Thursday 15:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

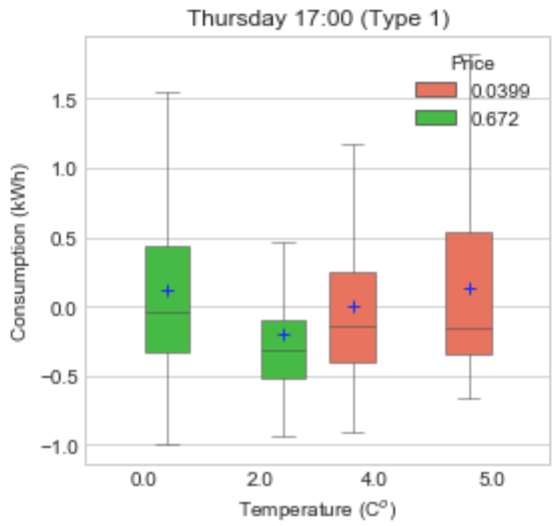

Thursday 15:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
3.0
5.0
7.0

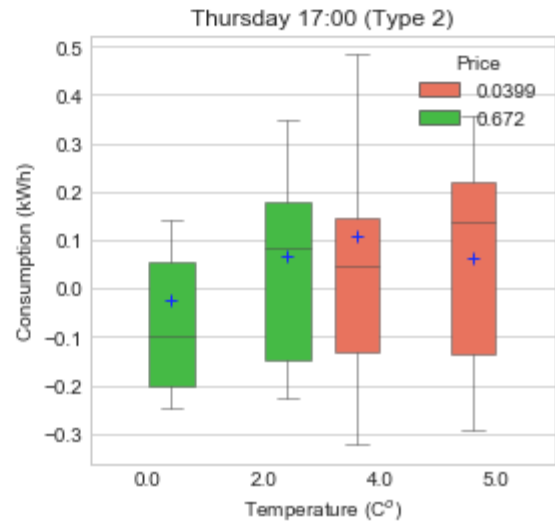

Thursday 16:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)

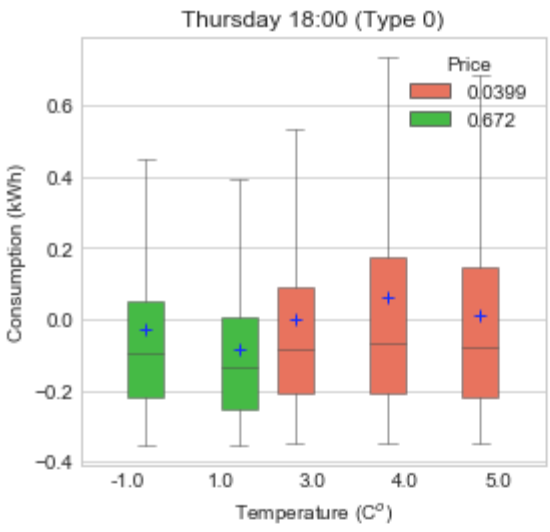

Thursday 16:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)

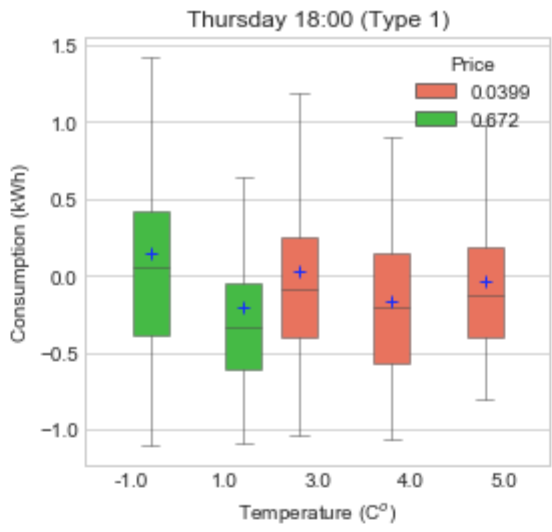

Thursday 16:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)

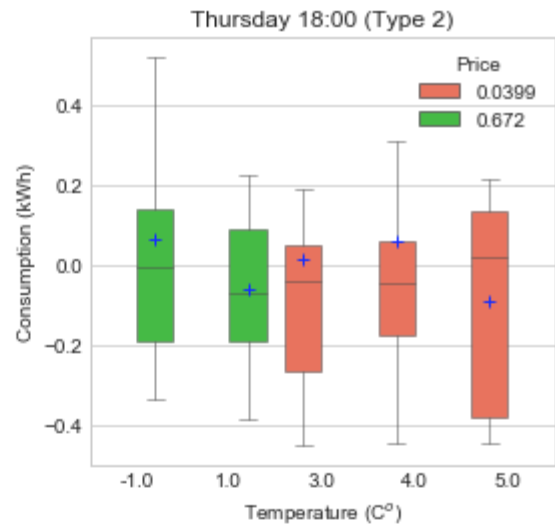

Thursday 17:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

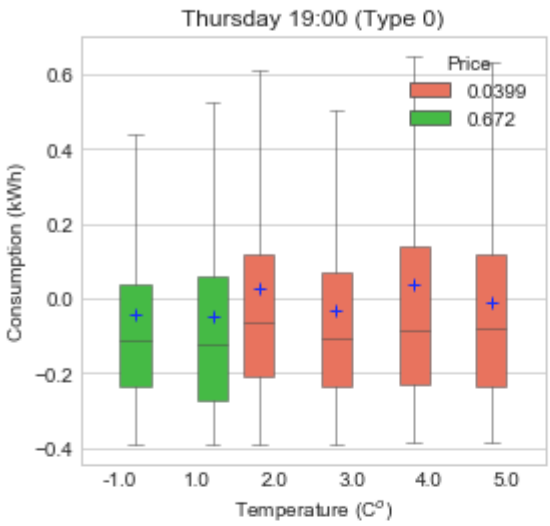

Thursday 17:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

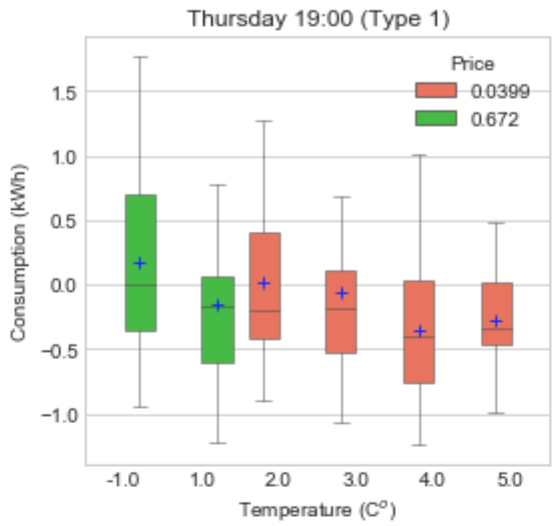

Thursday 17:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

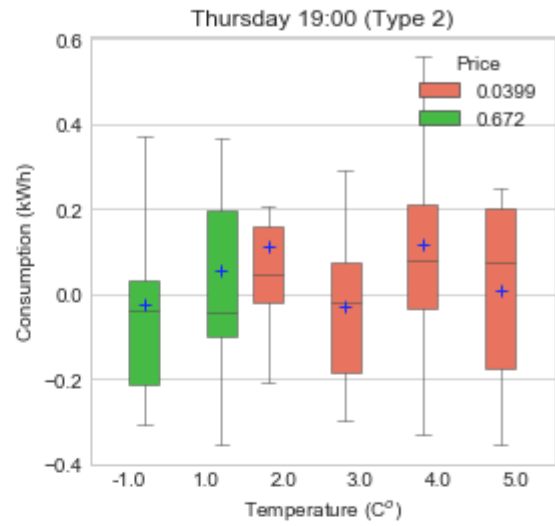

Thursday 18:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

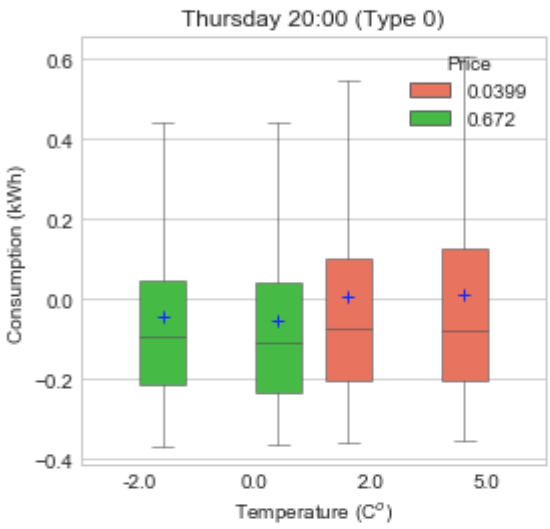

Thursday 18:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

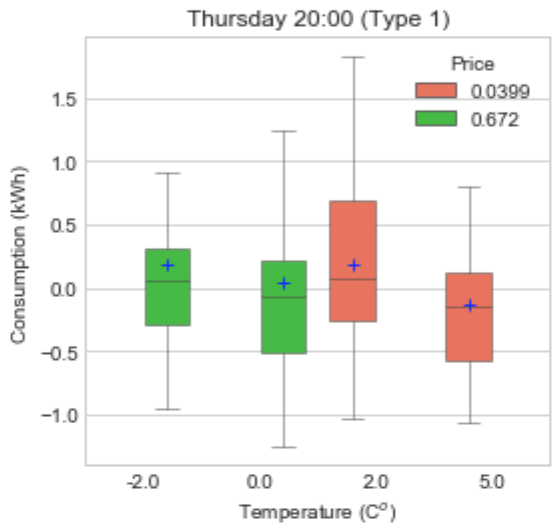

Thursday 18:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

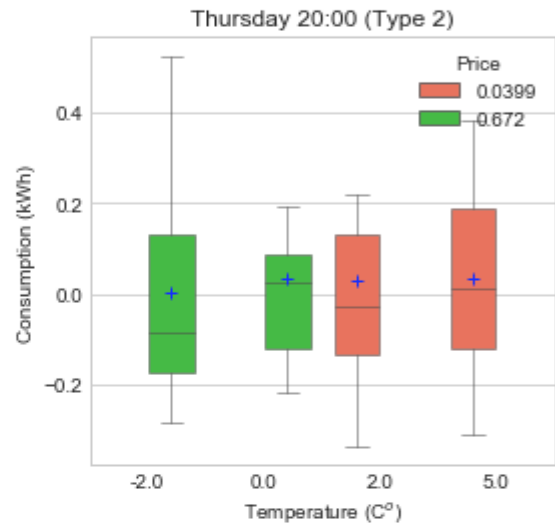

Thursday 19:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

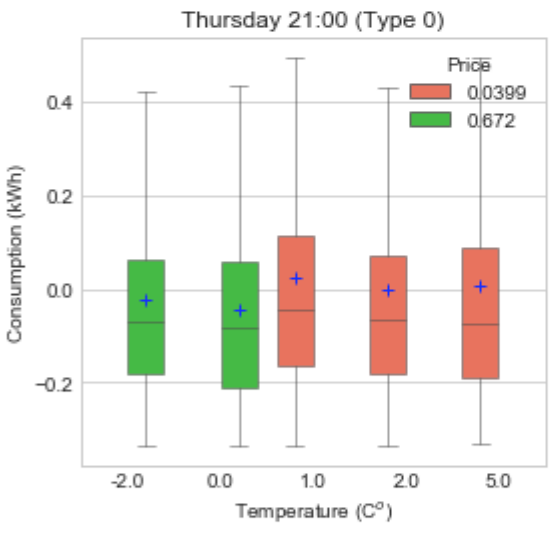

Thursday 19:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

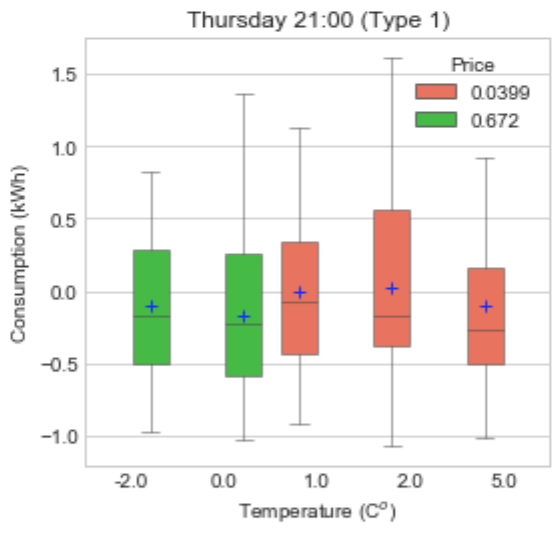

Thursday 19:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

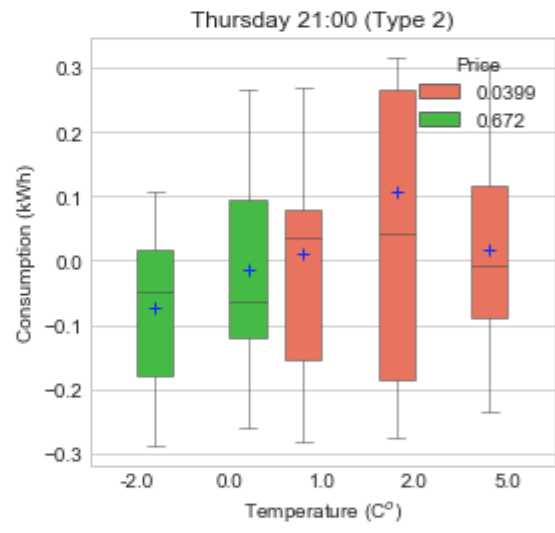

Thursday 20:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

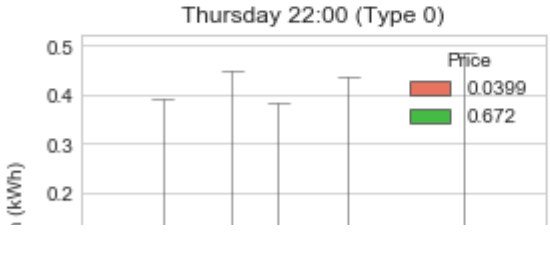

Thursday 20:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

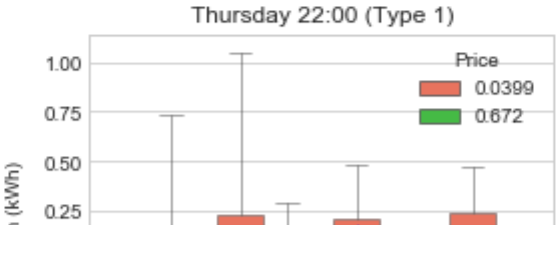

Thursday 20:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

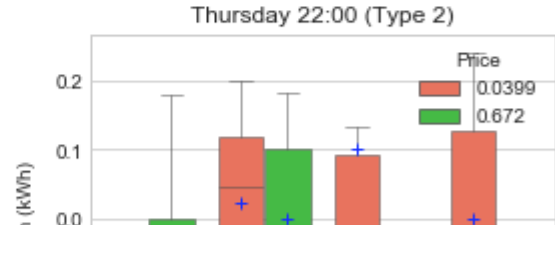

Thursday 21:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

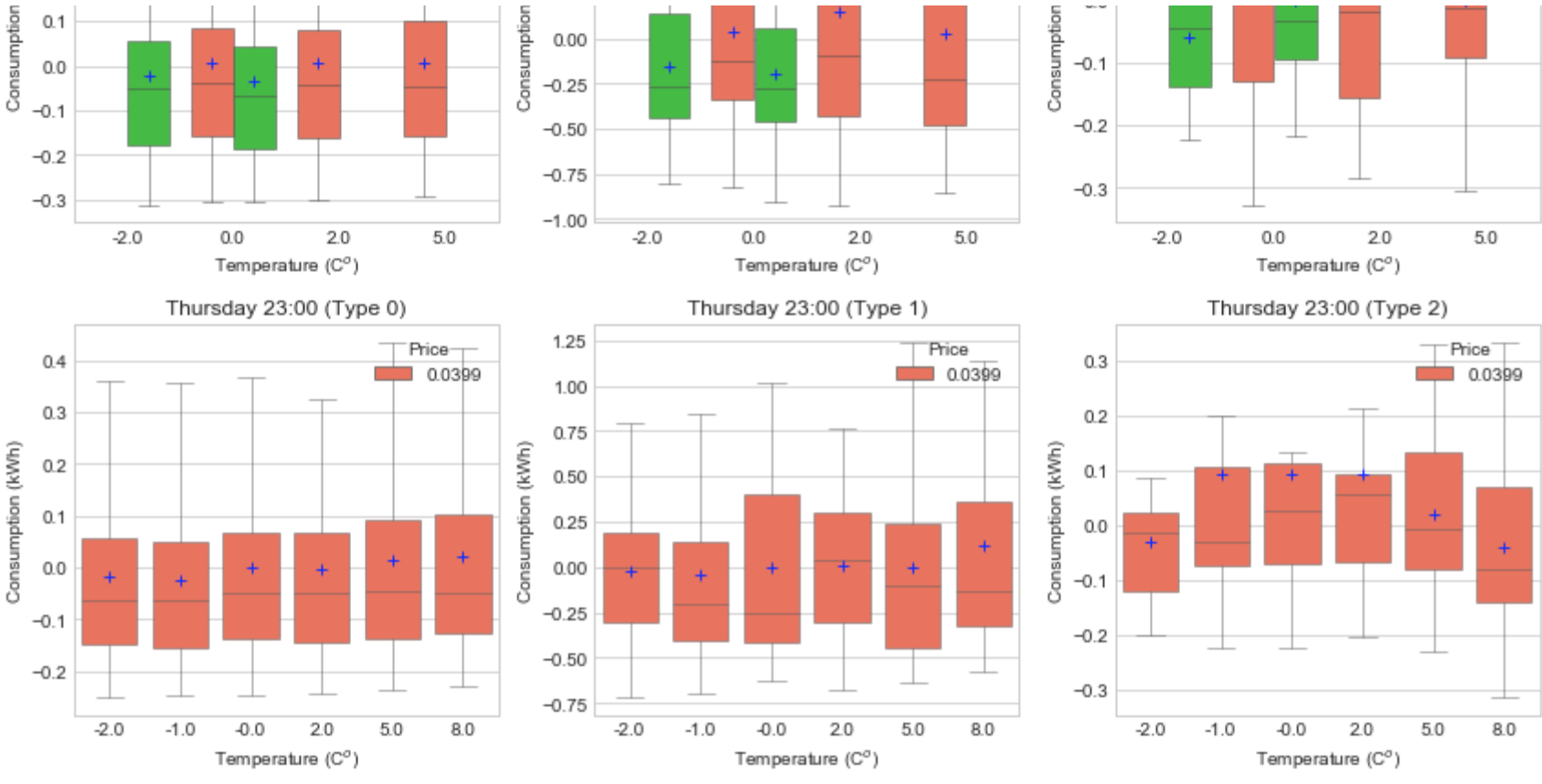

Thursday 21:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)


Thursday 21:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

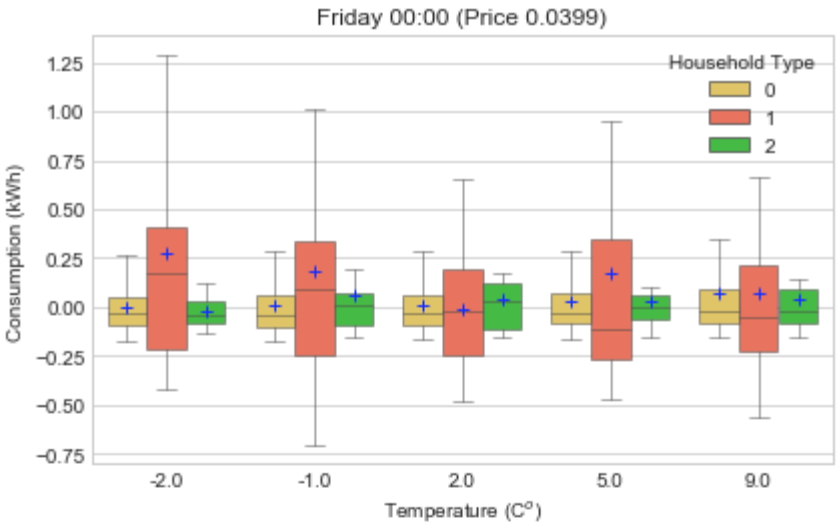

Thursday 22:00 (Type 0)
Price
0.0399
0.672
Temperature (C°)

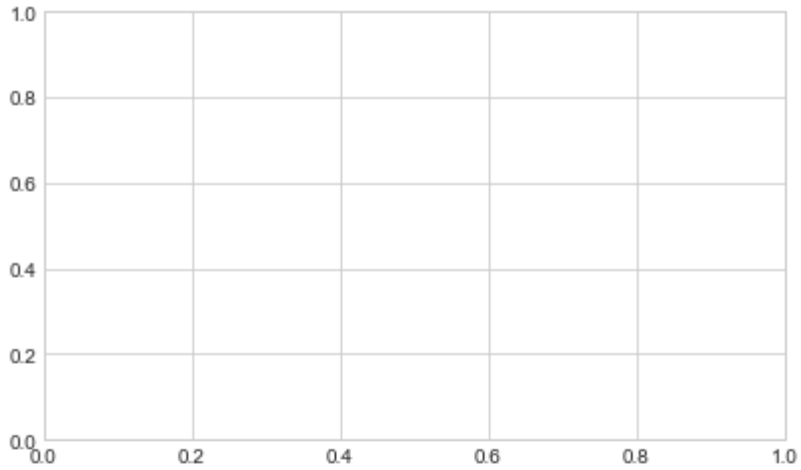

Thursday 22:00 (Type 1)
Price
0.0399
0.672
Temperature (C°)

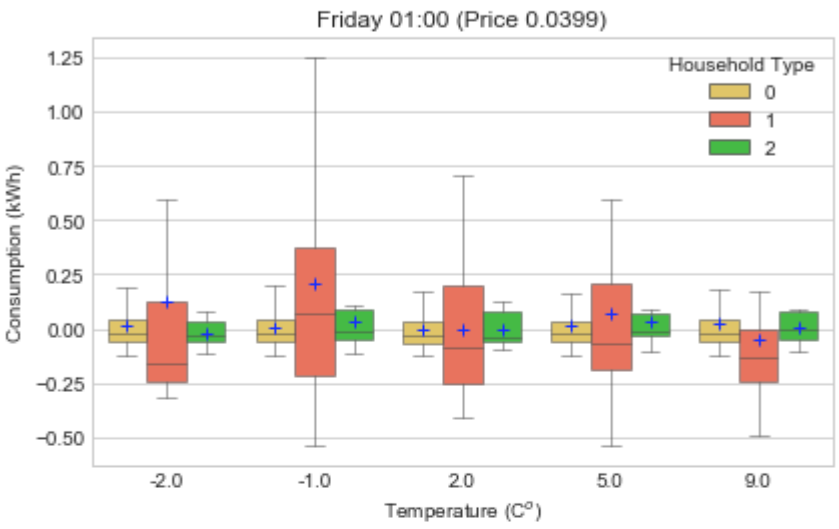

Thursday 22:00 (Type 2)
Price
0.0399
0.672
Temperature (C°)

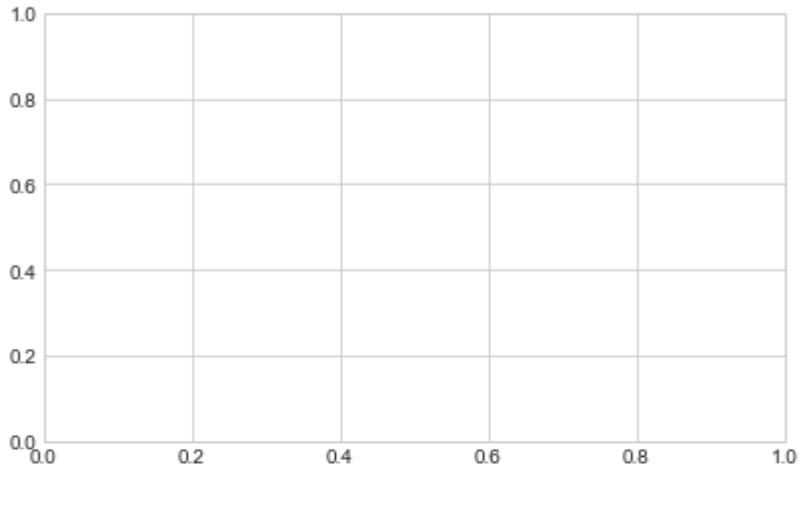

Consumption
Temperature (C°)
Thursday 23:00 (Type 0)
Thursday 23:00 (Type 1)
Thursday 23:00 (Type 2)
Consumption (kWh)
Price
0.0399
-2.0
-1.0
-0.0
2.0
5.0
8.0

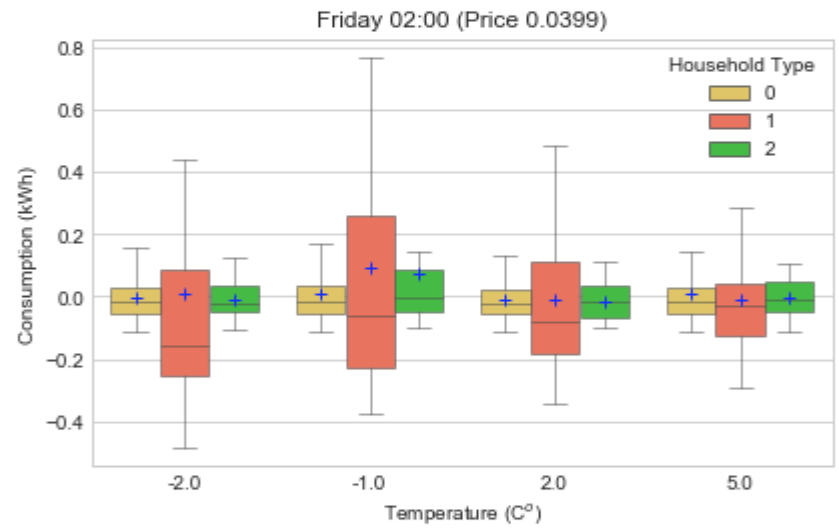
In [101]:
```
# Friday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 4 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers=False, showmeans=True, meanprops={"marker":"+","ma
rkerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers=False, showmeans=True, meanprops={"marker":"+","ma
rkerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

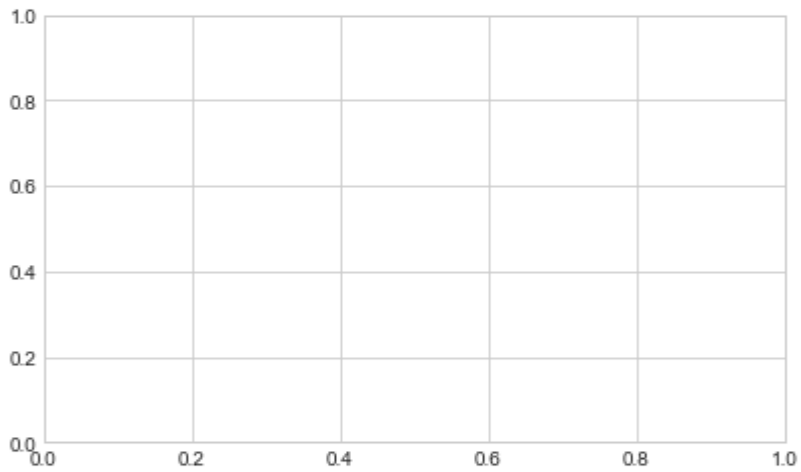

Friday 00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-2.0
-1.0
2.0
5.0
9.0
Temperature (C°)

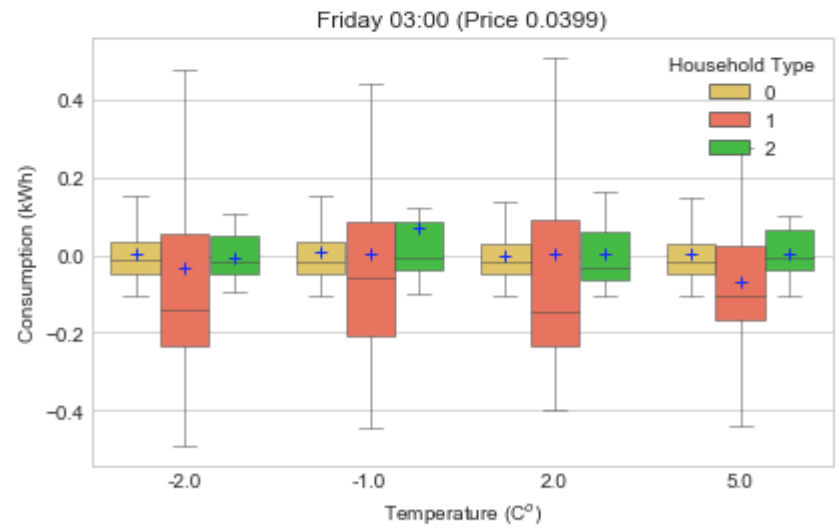

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

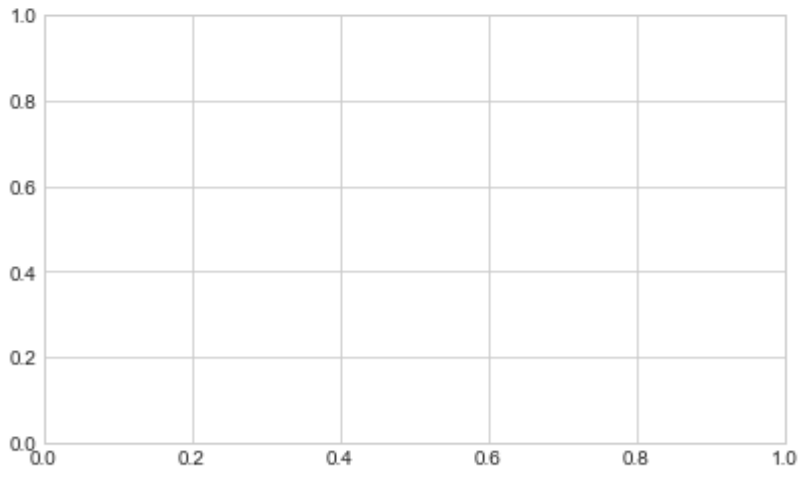

Friday 01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
-2.0
-1.0
2.0
5.0
9.0
Temperature (C°)

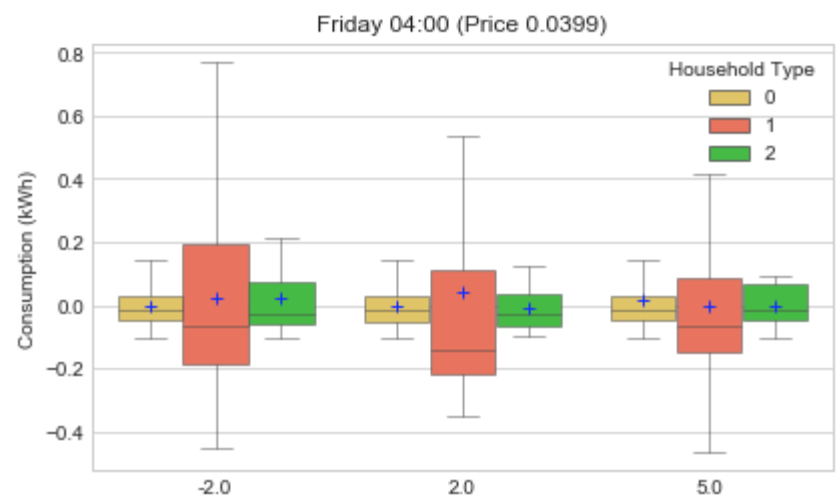

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

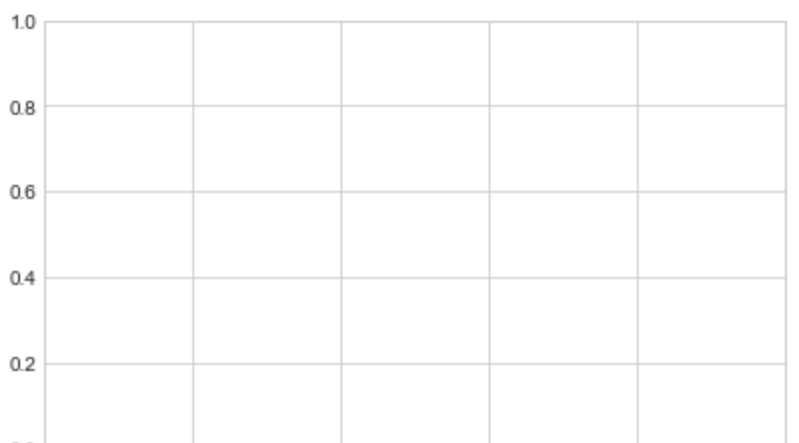

Friday 02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0
-1.0
2.0
5.0
Temperature (C°)

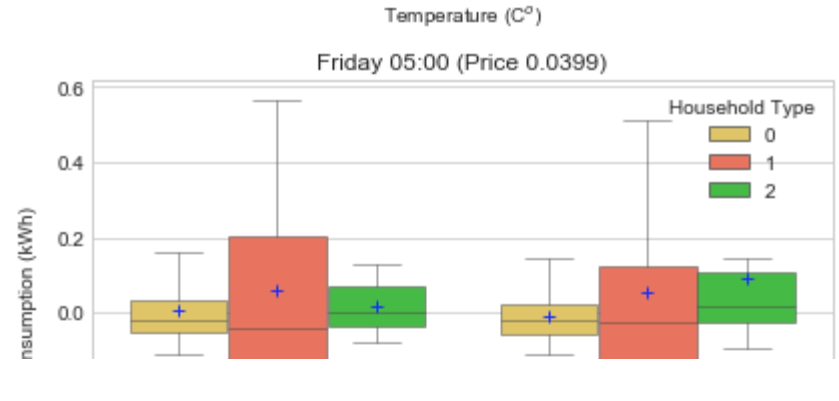

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

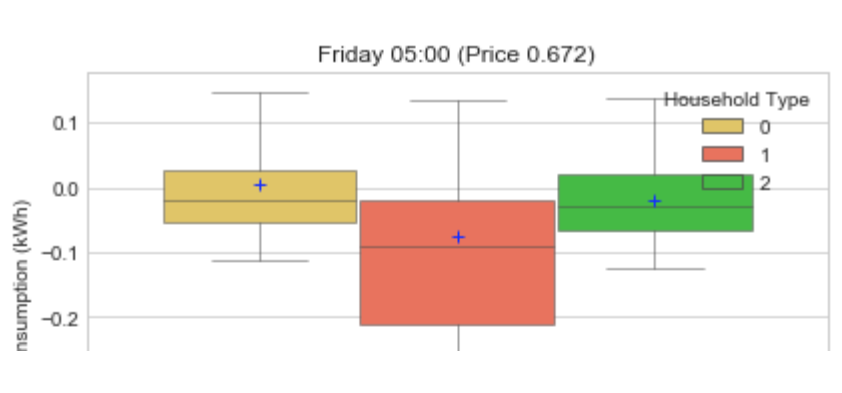

Friday 03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
-2.0
-1.0
2.0
5.0
Temperature (C°)

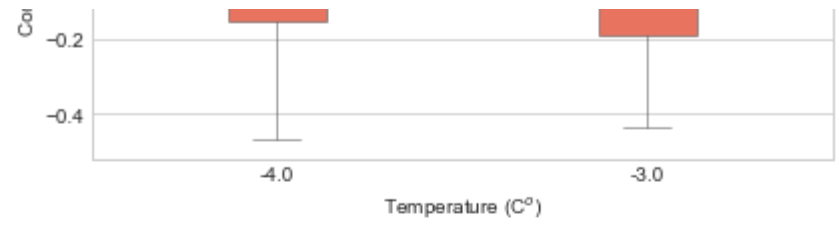

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

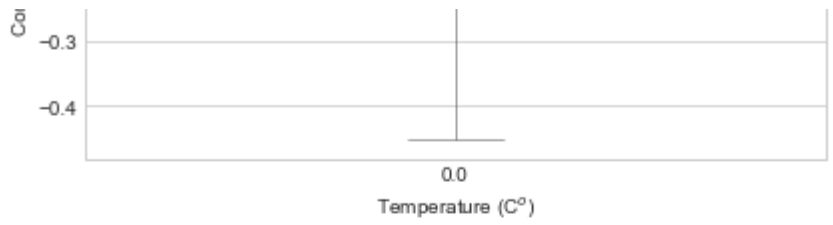

Friday 04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0
2.0
5.0
Temperature (C°)

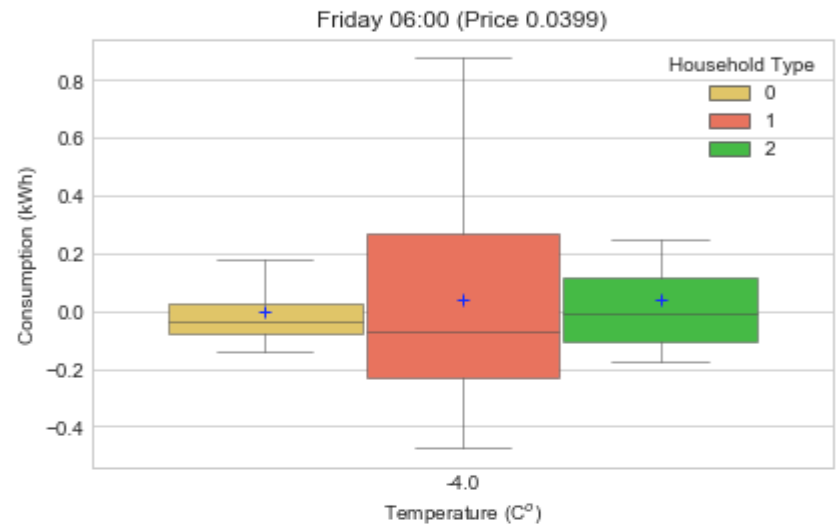

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

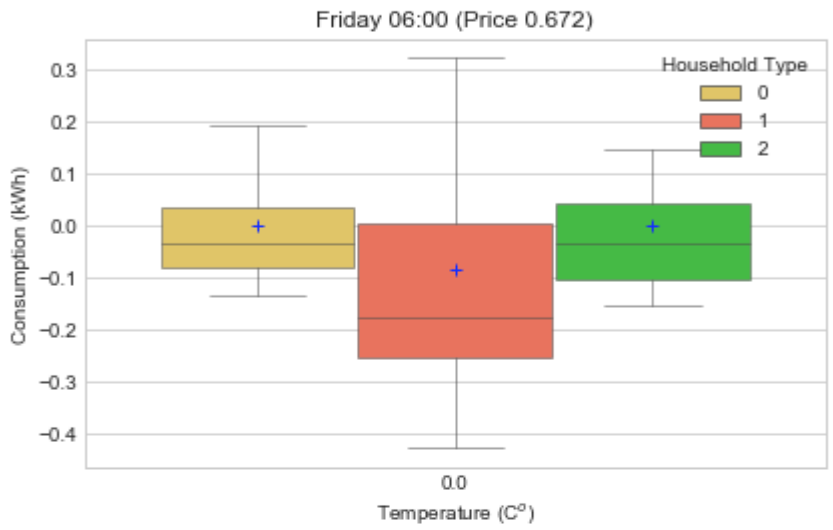

Friday 05:00 (Price 0.0399)
Household Type
0
1
2
0.6
0.4
0.2
0.0

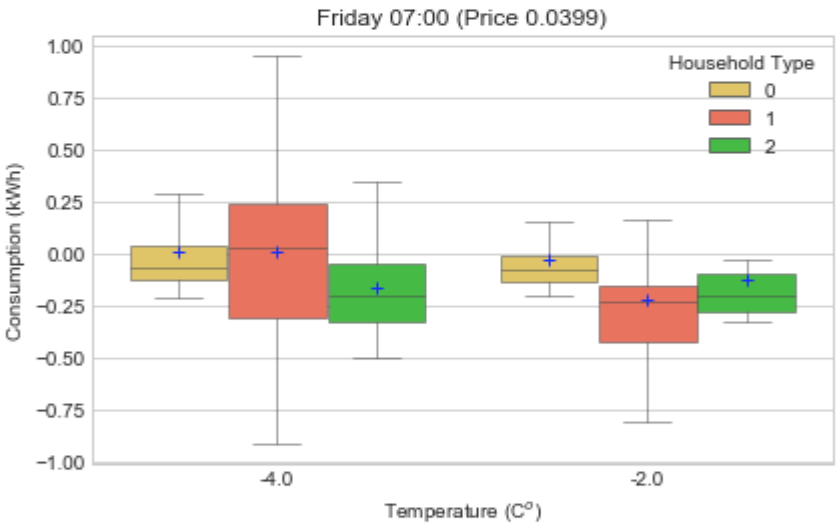

Friday 05:00 (Price 0.672)
Household Type
0
1
2
0.1
0.0
−0.1
−0.2

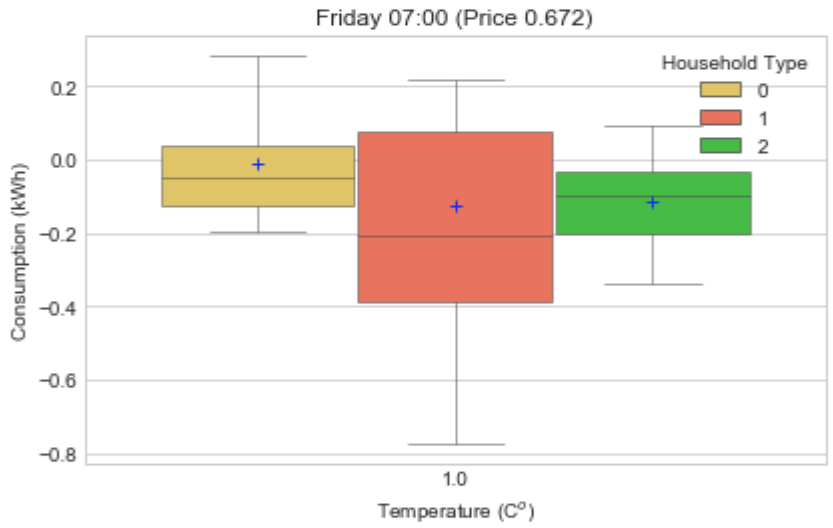

-0.2
-0.4
-4.0
-3.0
Temperature (C°)

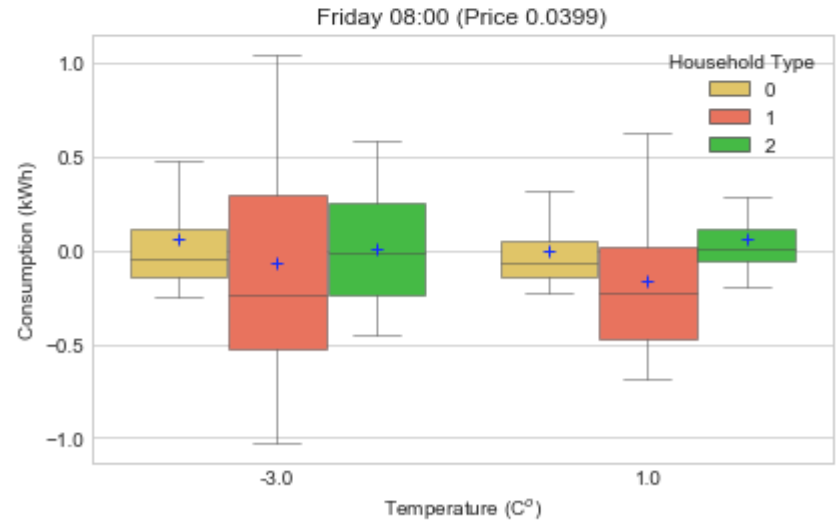

-0.3
-0.4
0.0
Temperature (C°)

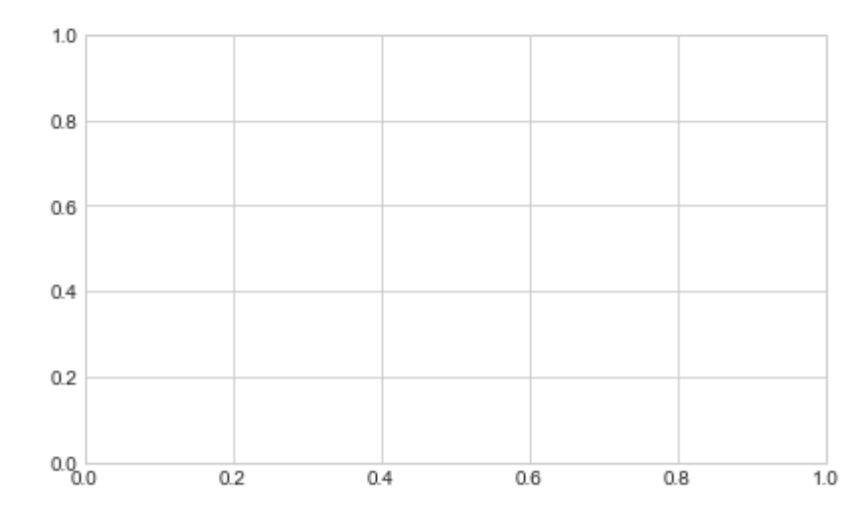

Friday 06:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-4.0
Temperature (C°)

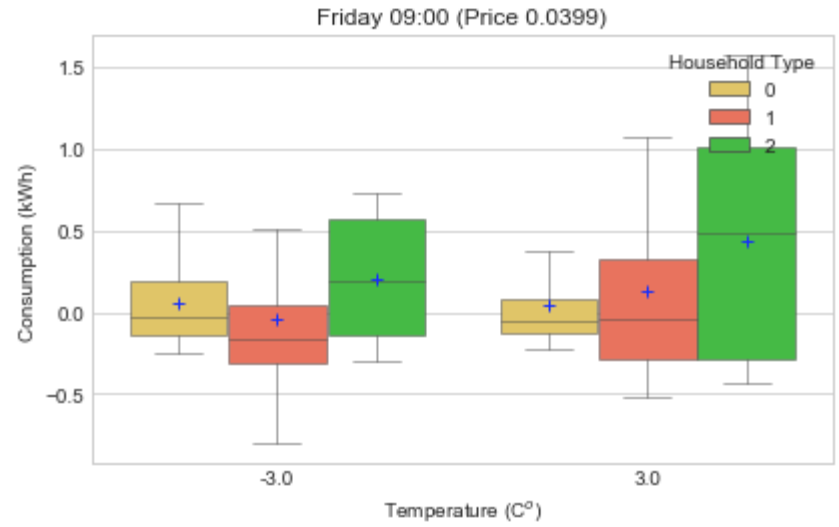

Friday 06:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0.0
Temperature (C°)

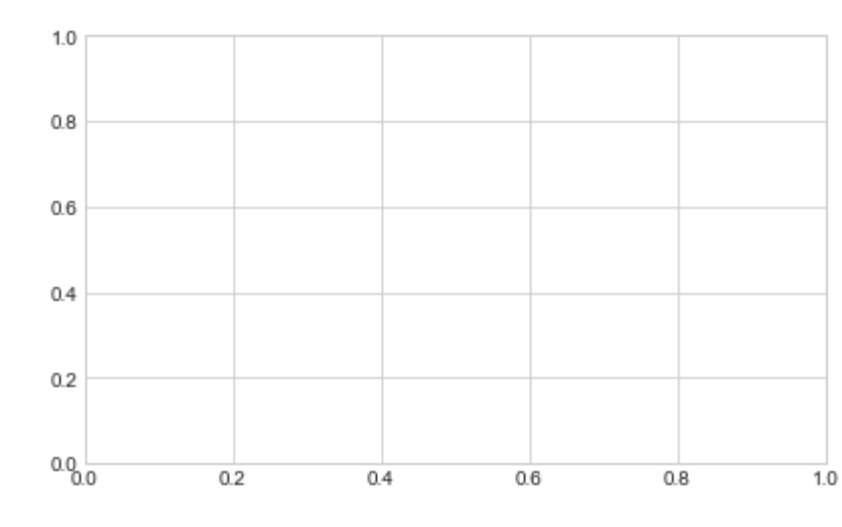

Friday 07:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-4.0
-2.0
Temperature (C°)

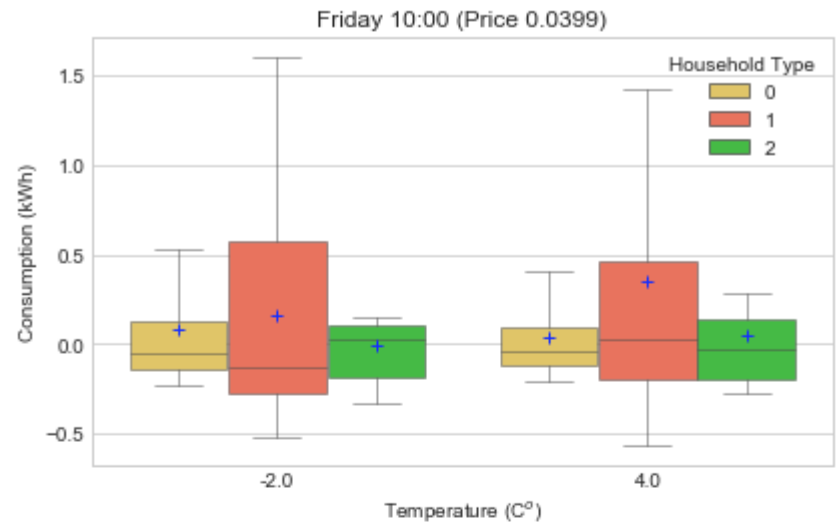

Friday 07:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.2
0.0
−0.2
−0.4
−0.6
−0.8
1.0
Temperature (C°)

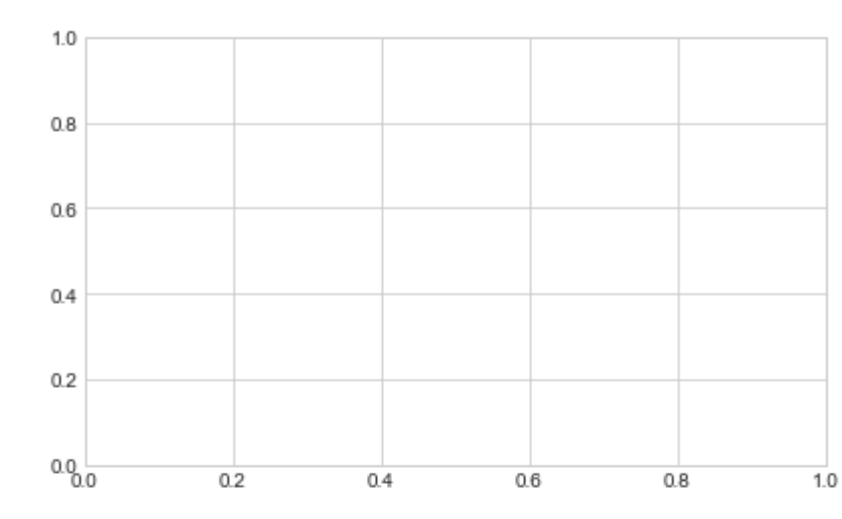

Friday 08:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-3.0
1.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

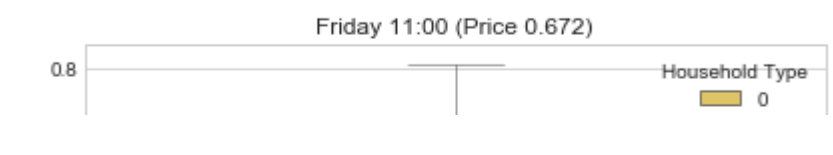

Friday 09:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-3.0
3.0
Temperature (C°)

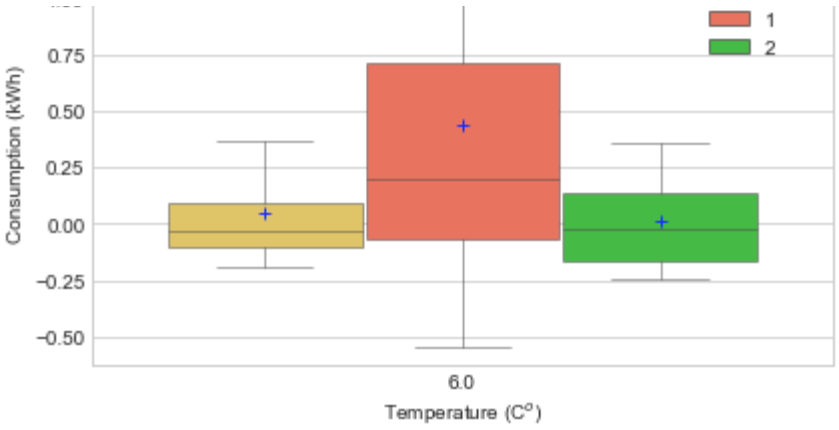

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

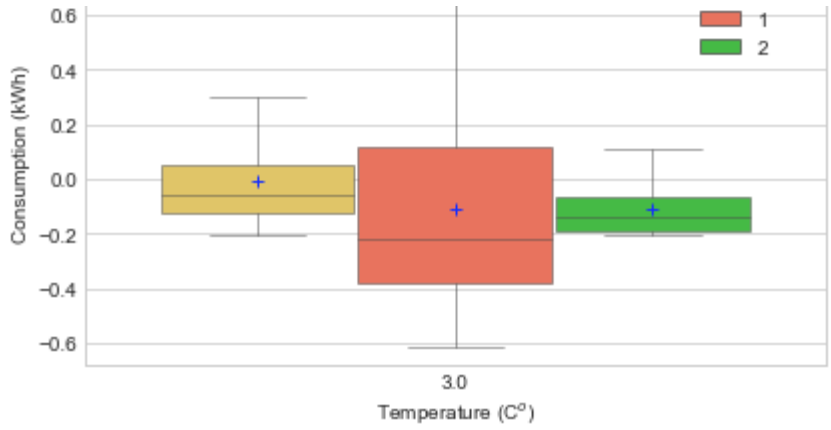

Friday 10:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-2.0
4.0
Temperature (C°)

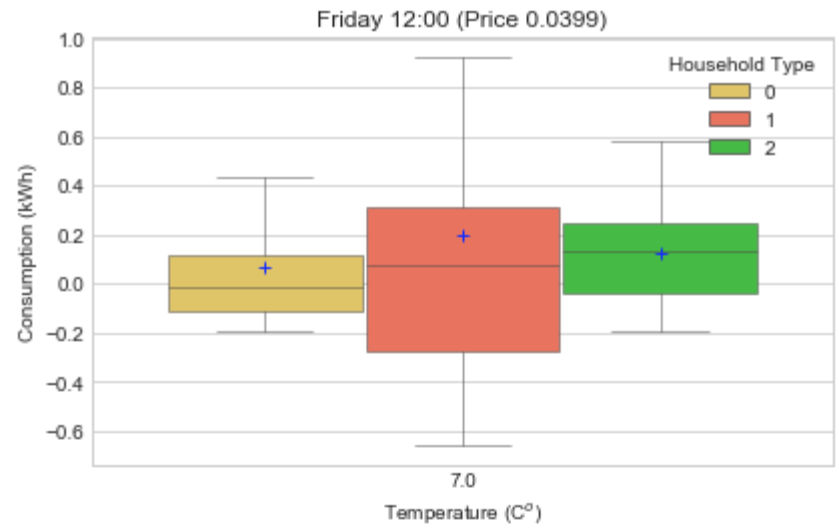

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

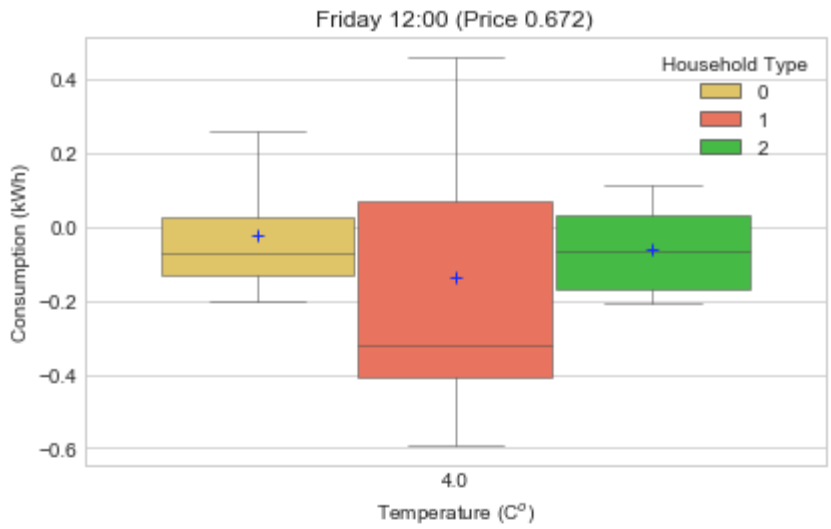

Friday 11:00 (Price 0.0399)
Household Type
0
1.25
1.00

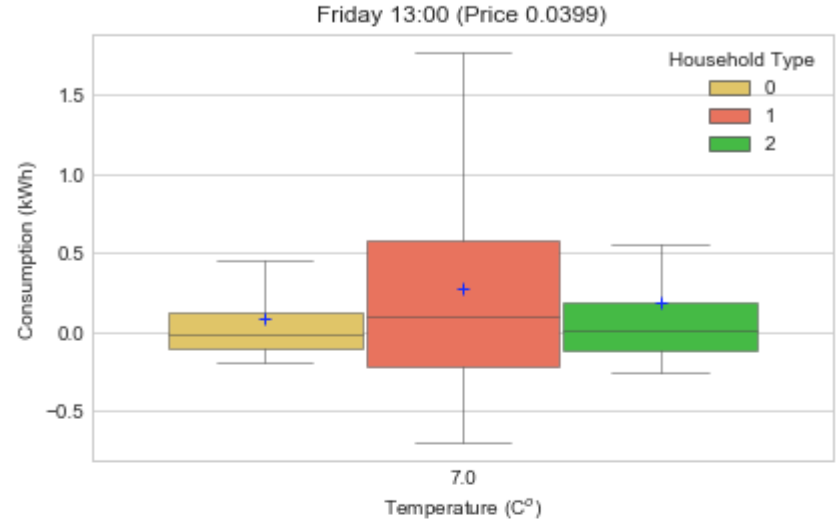

Friday 11:00 (Price 0.672)
Household Type
0
0.8

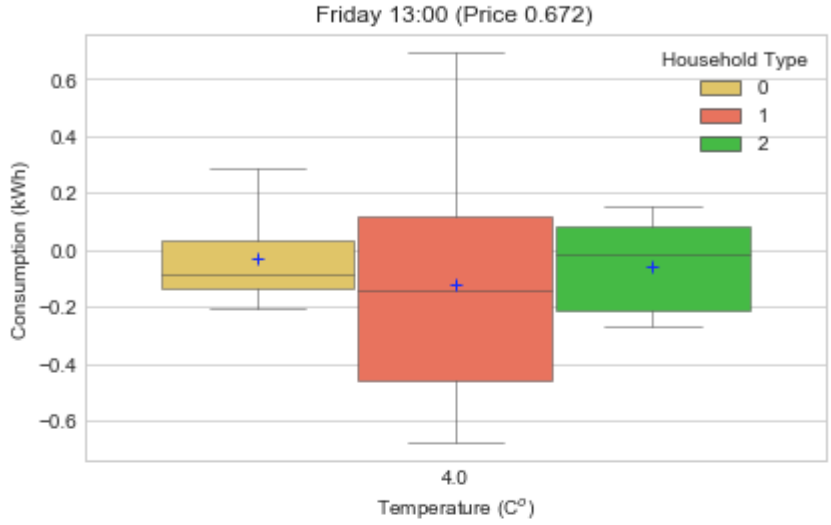

1
2
0.75
0.50
0.25
0.00
−0.25
−0.50
Consumption (kWh)
6.0
Temperature (C°)

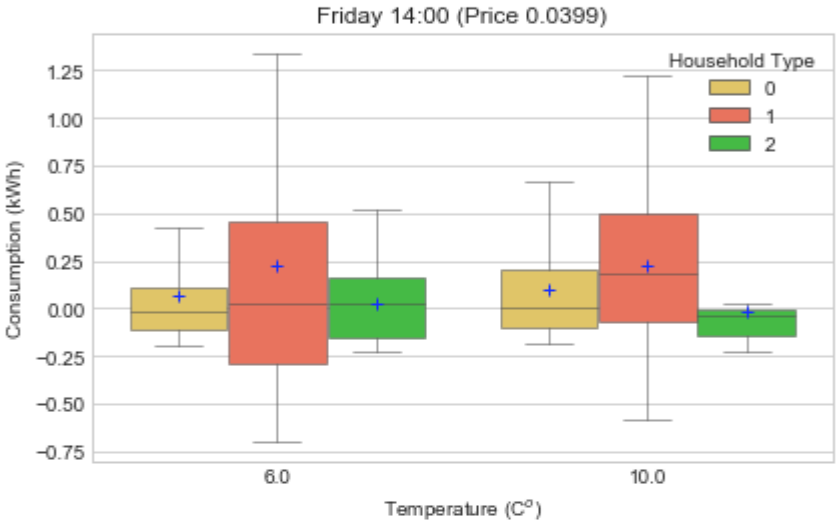

1
2
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)
3.0
Temperature (C°)

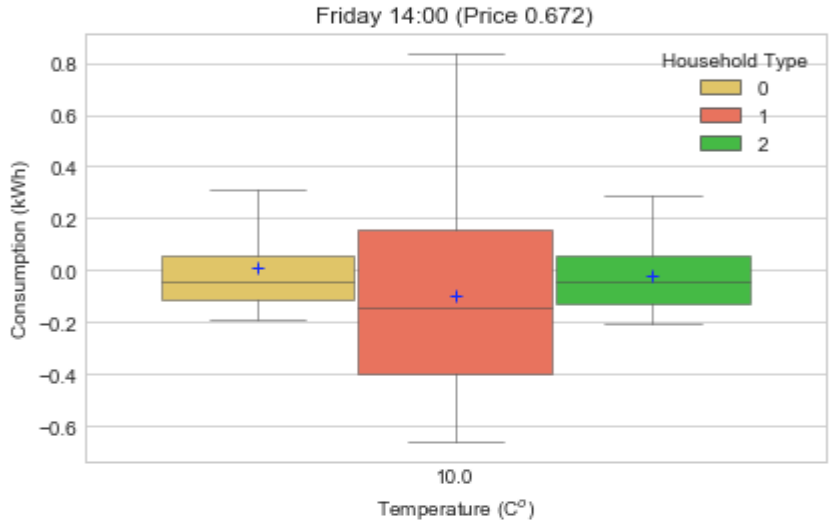

Friday 12:00 (Price 0.0399)
Household Type
0
1
2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)
7.0
Temperature (C°)

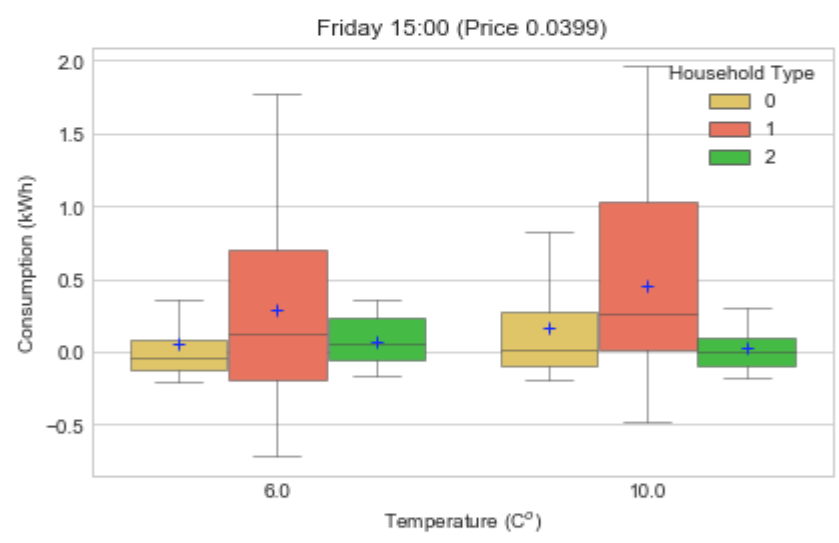

Friday 12:00 (Price 0.672)
Household Type
0
1
2
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)
4.0
Temperature (C°)

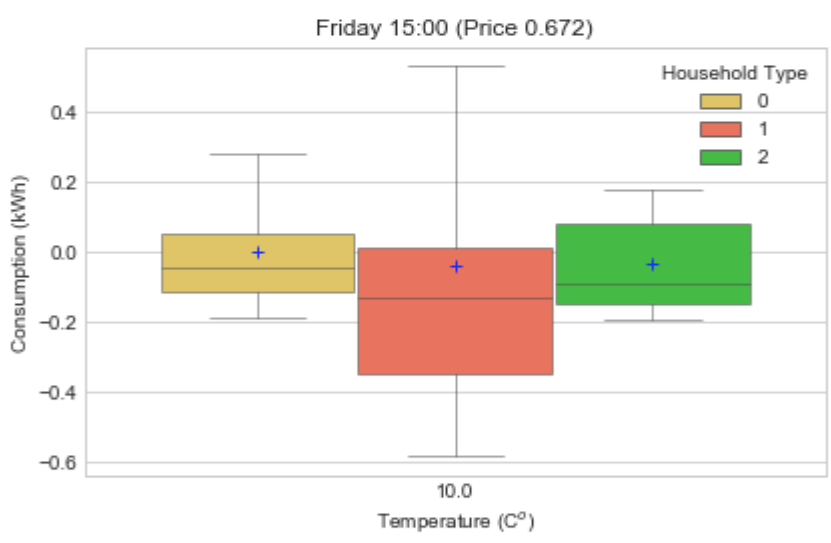

Friday 13:00 (Price 0.0399)
Household Type
0
1
2
1.5
1.0
0.5
0.0
−0.5
Consumption (kWh)
7.0
Temperature (C°)

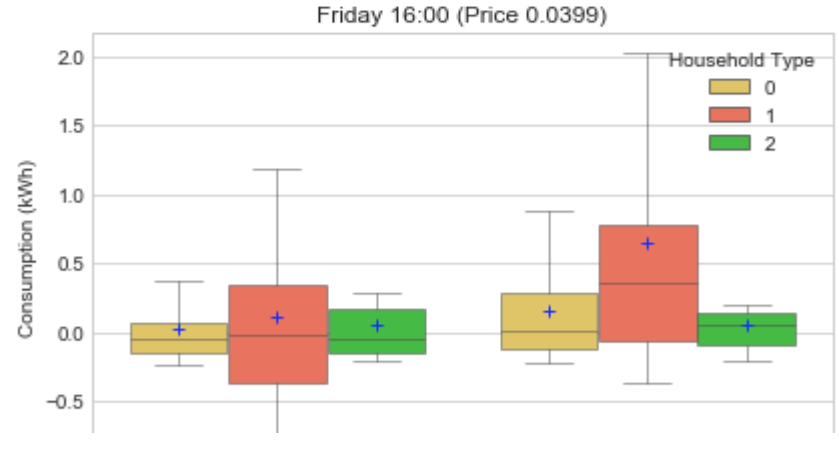

Friday 13:00 (Price 0.672)
Household Type
0
1
2
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)
4.0
Temperature (C°)

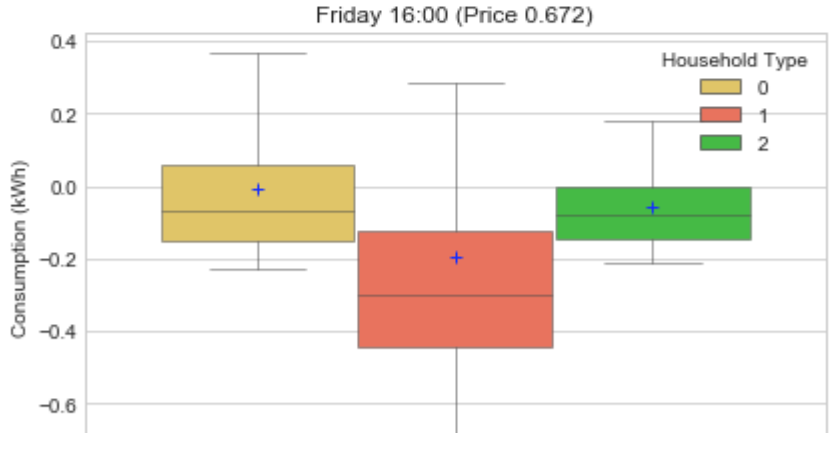

Friday 14:00 (Price 0.0399)
Household Type
0
1
2
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
Consumption (kWh)
6.0
10.0
Temperature (C°)


Friday 14:00 (Price 0.672)
Household Type
0
1
2
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)
10.0
Temperature (C°)

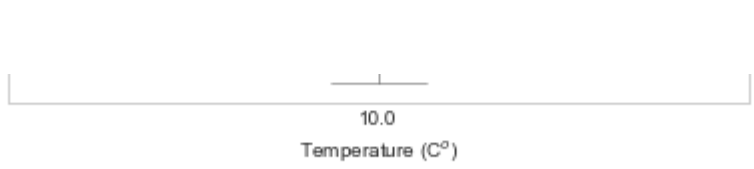

Friday 15:00 (Price 0.0399)
Household Type
0
1
2
2.0
1.5
1.0
0.5
0.0
−0.5
Consumption (kWh)
6.0
10.0
Temperature (C°)

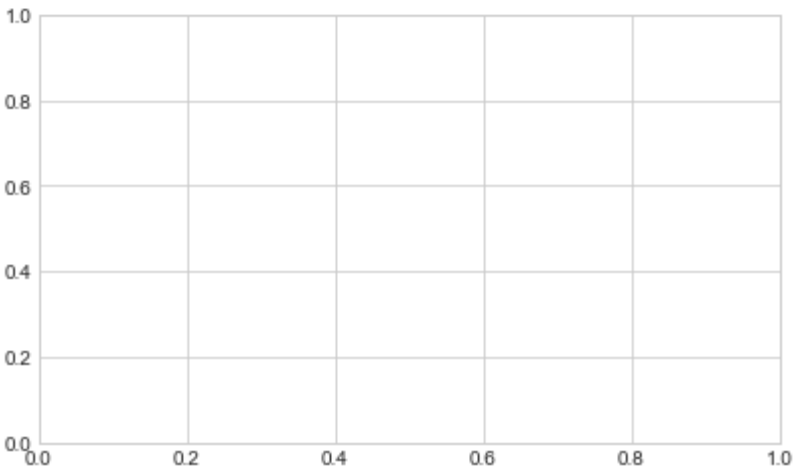

Friday 15:00 (Price 0.672)
Household Type
0
1
2
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)
10.0
Temperature (C°)

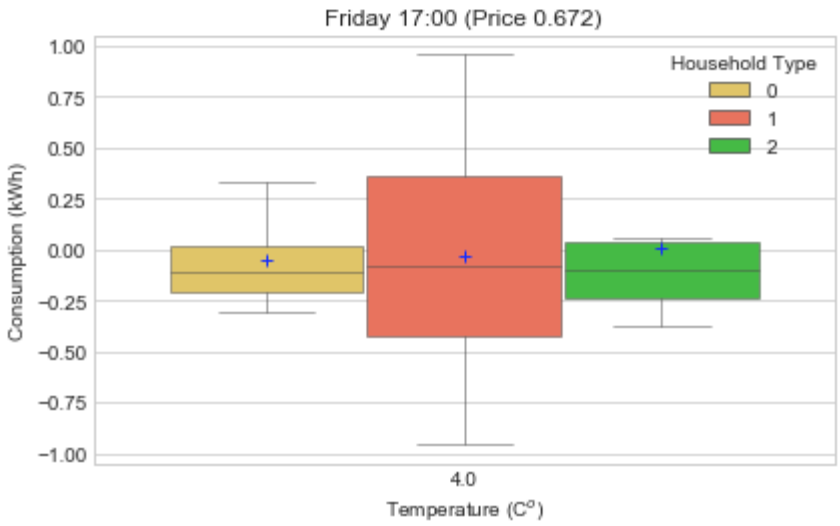

Friday 16:00 (Price 0.0399)
Household Type
0
1
2
2.0
1.5
1.0
0.5
0.0
−0.5
Consumption (kWh)

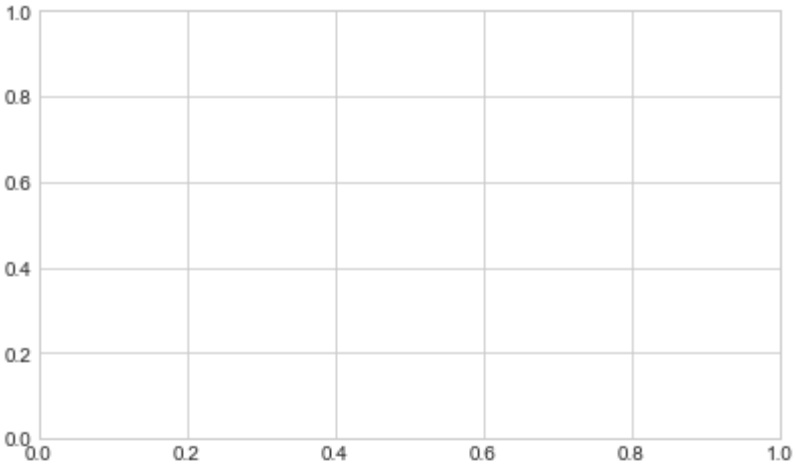

Friday 16:00 (Price 0.672)
Household Type
0
1
2
0.4
0.2
0.0
−0.2
−0.4
−0.6
Consumption (kWh)

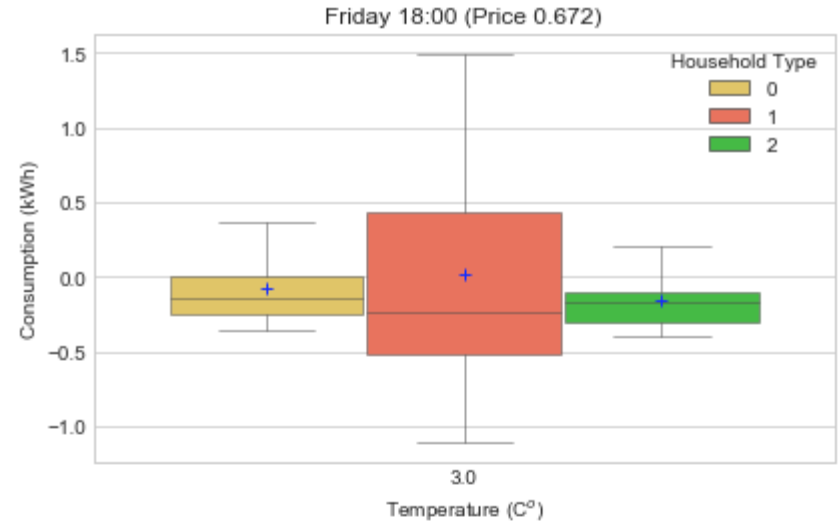

5.0
9.0
Temperature (C°)

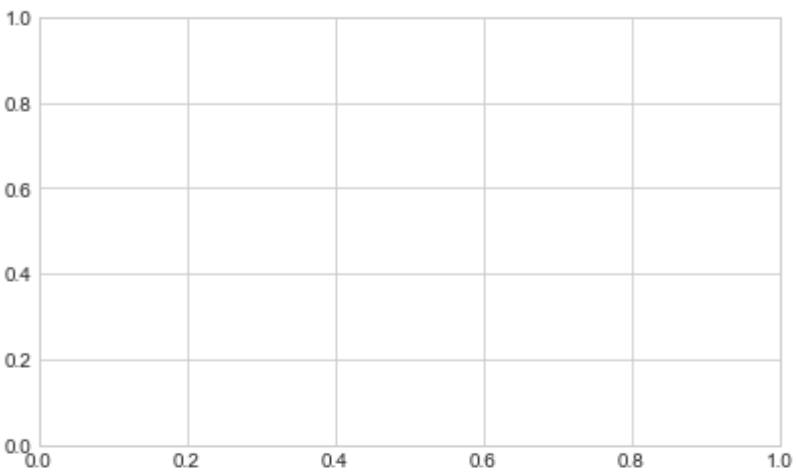

10.0
Temperature (C°)

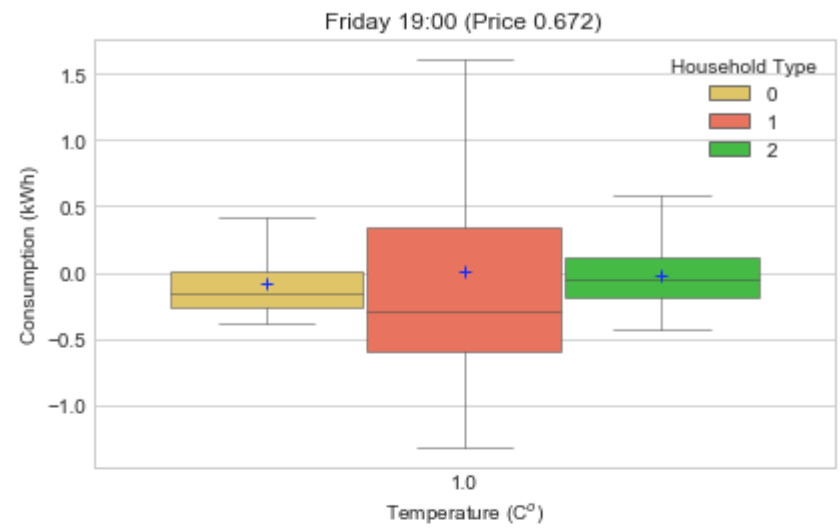

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

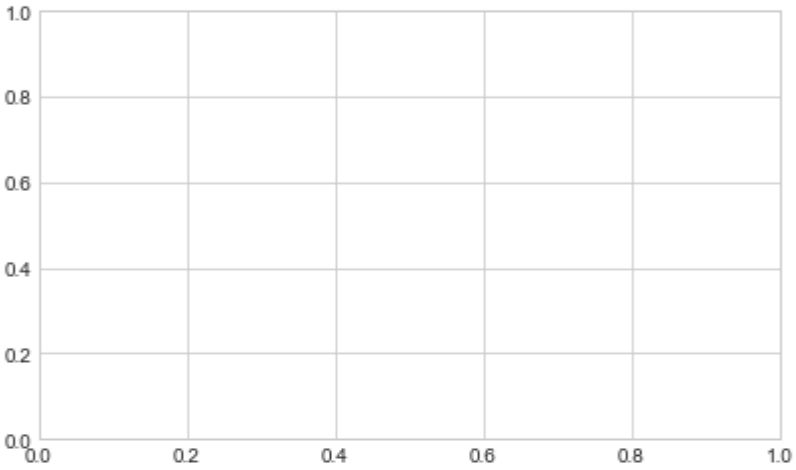

Friday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
4.0
Temperature (C°)

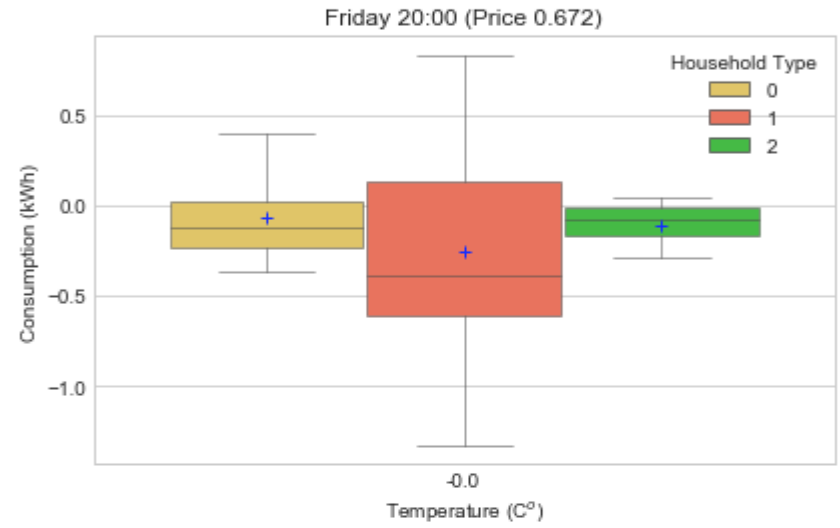

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

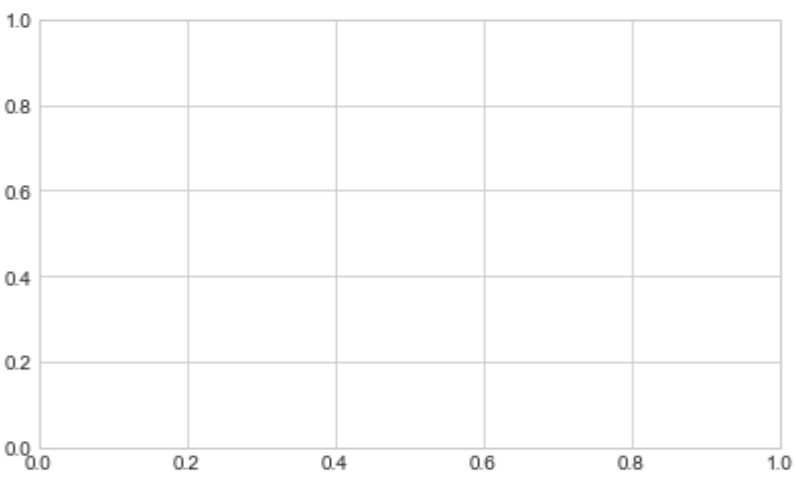

Friday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
3.0
Temperature (C°)

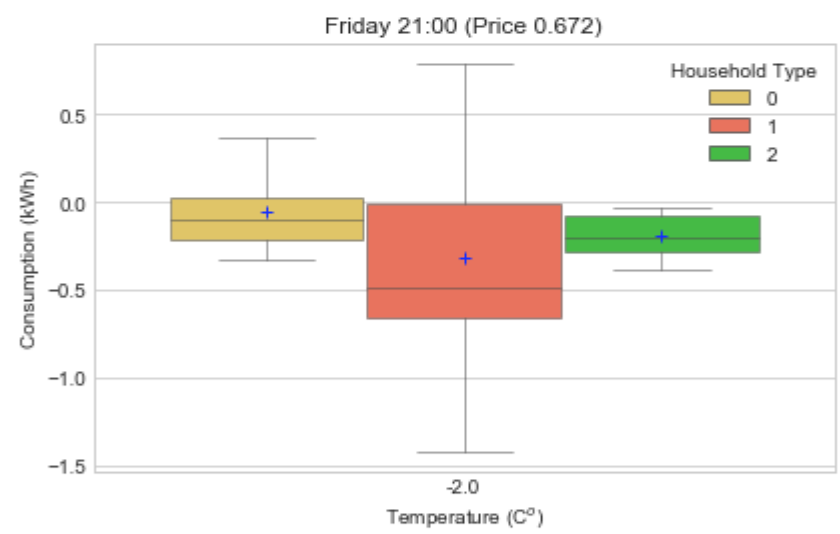

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

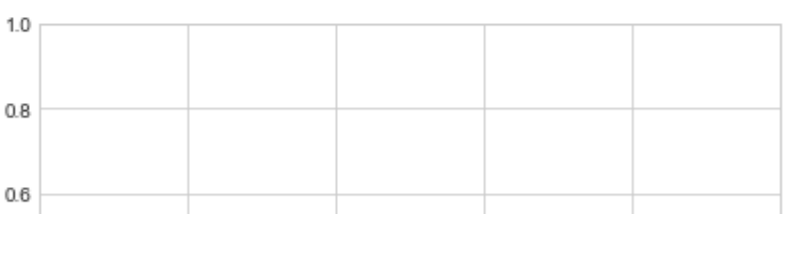

Friday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
1.0
Temperature (C°)

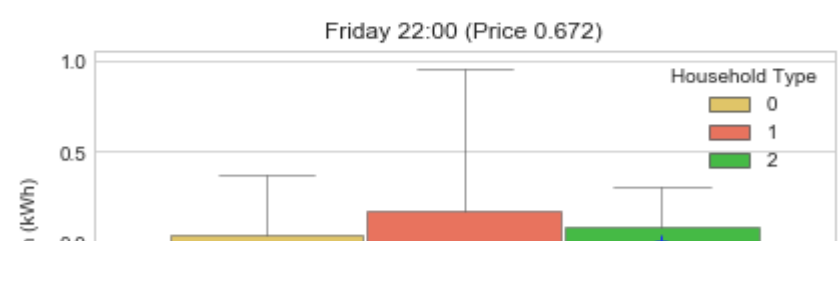

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

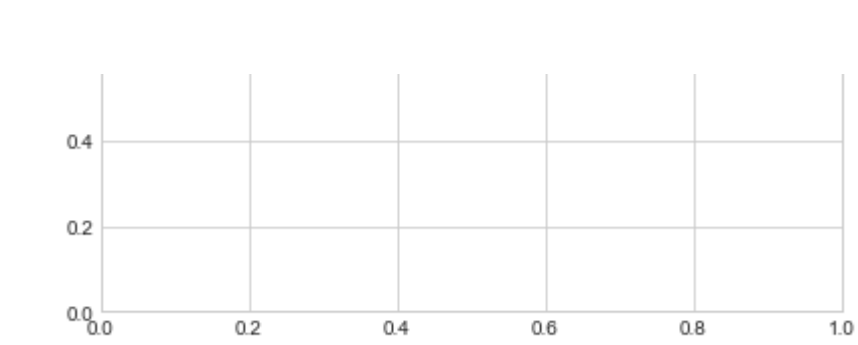

Friday 20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.5
0.0
−0.5
−1.0
-0.0
Temperature (C°)

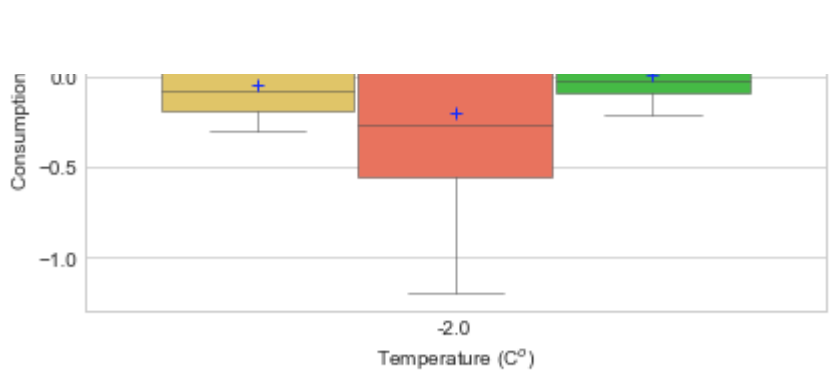

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

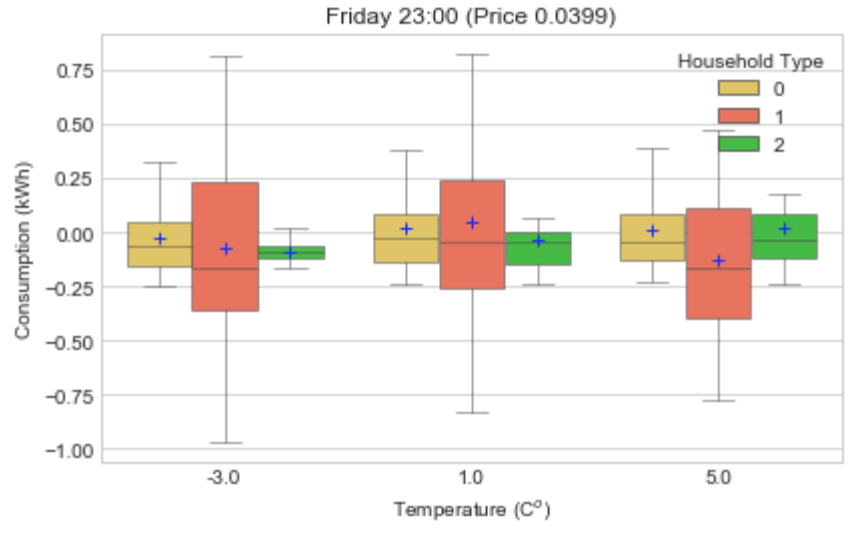

Friday 21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.5
0.0
−0.5
−1.0
−1.5
-2.0
Temperature (C°)

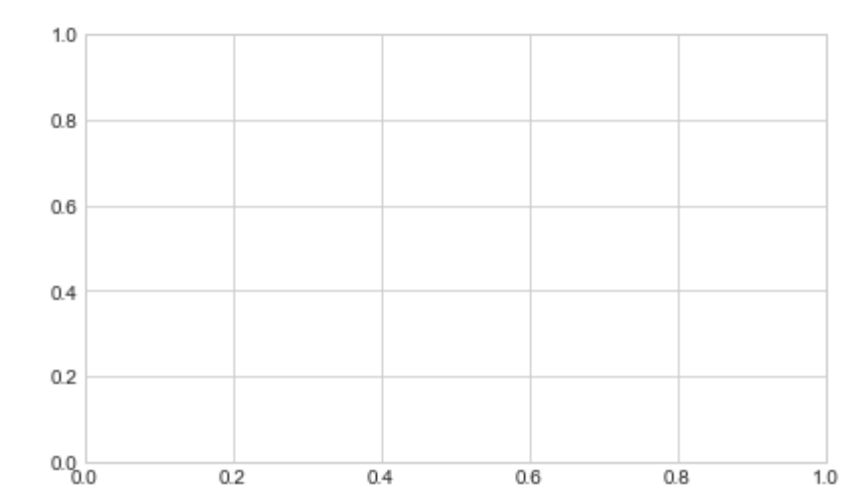

1.0
0.8
0.6


Friday 22:00 (Price 0.672)
Household Type
0
1
2
1.0
0.5

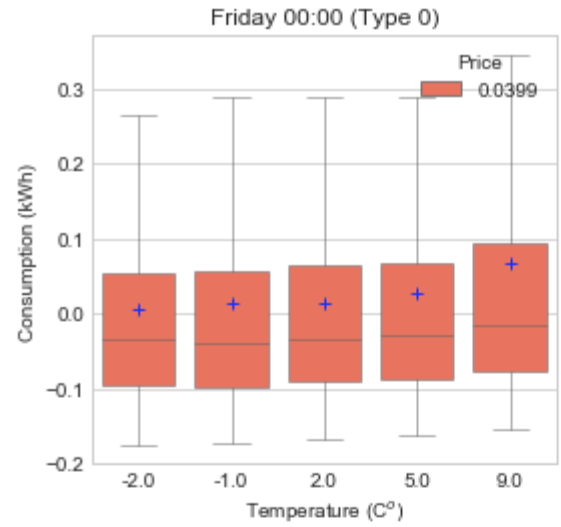

0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

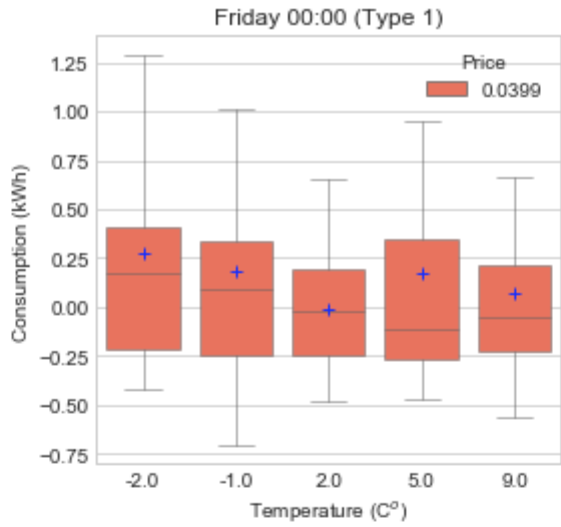

Consumption
0.0
−0.5
−1.0
-2.0
Temperature (C°)

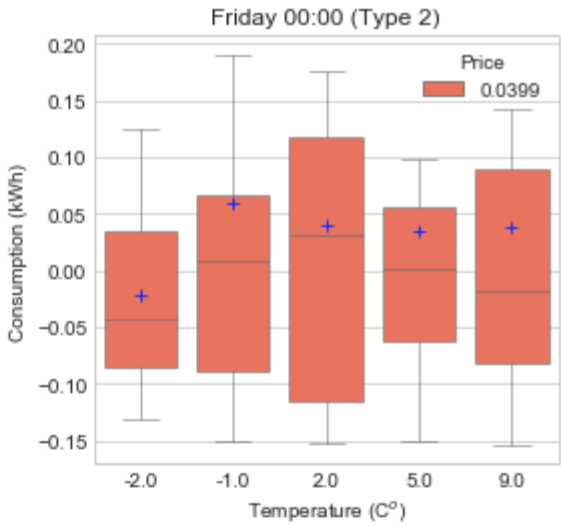

Friday 23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-3.0
1.0
5.0
Temperature (C°)

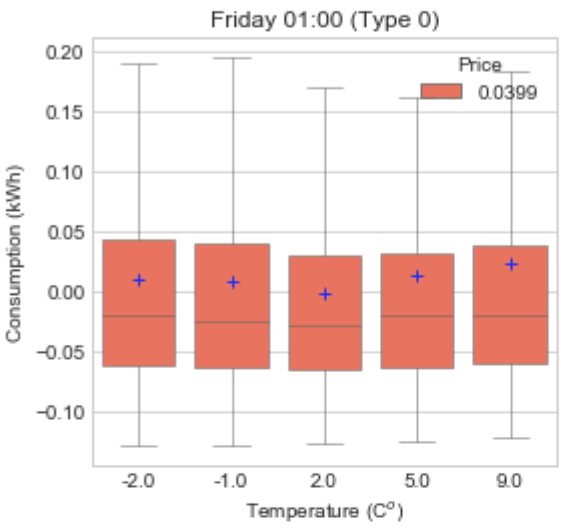

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

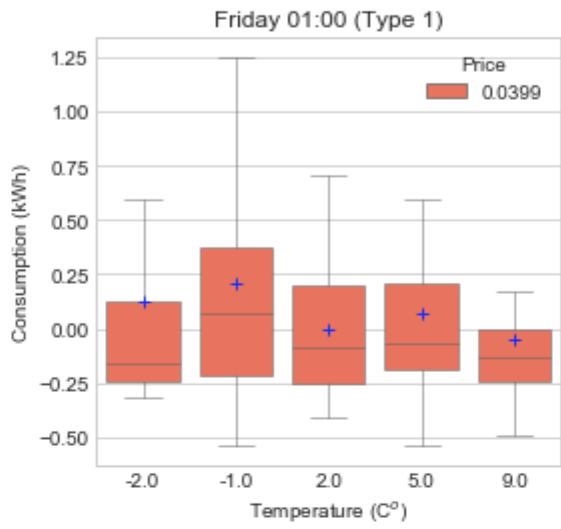
In [132]:
```
# price comparison
# Friday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 4 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers=False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:v
ivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers=False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:v
ivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

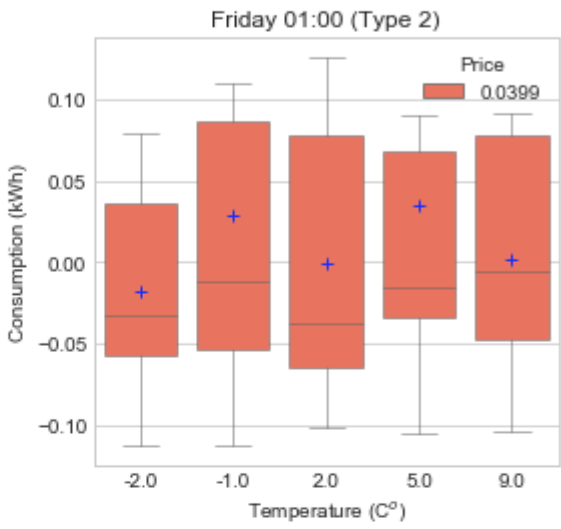

Friday 00:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)

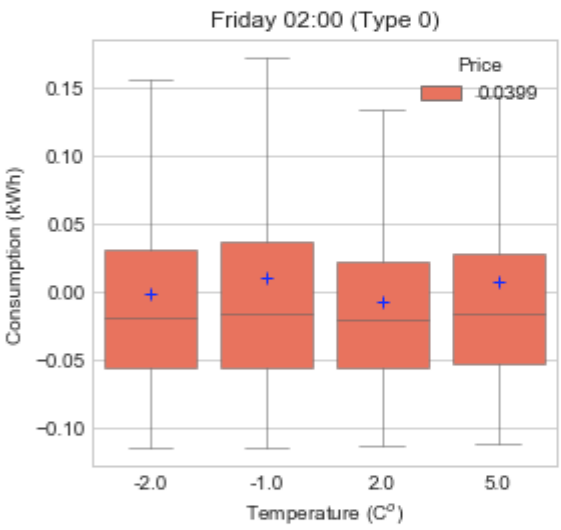

Friday 00:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)

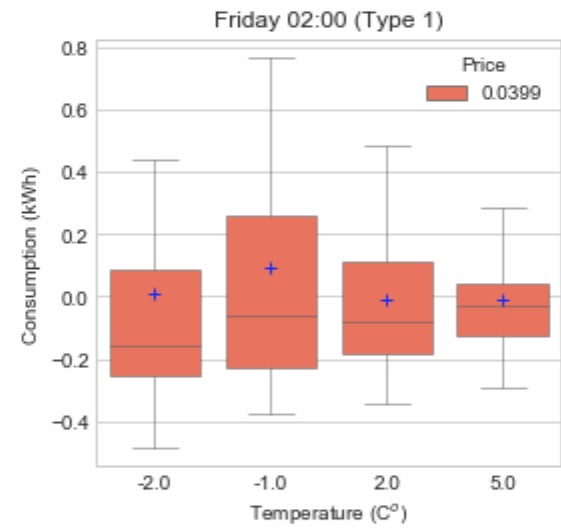

Friday 00:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)

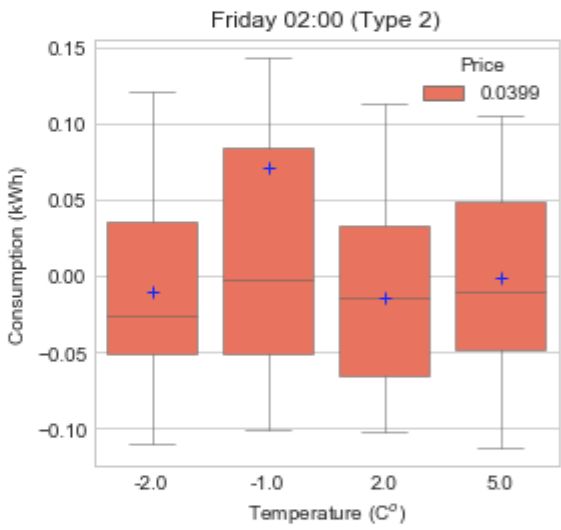

Friday 01:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)

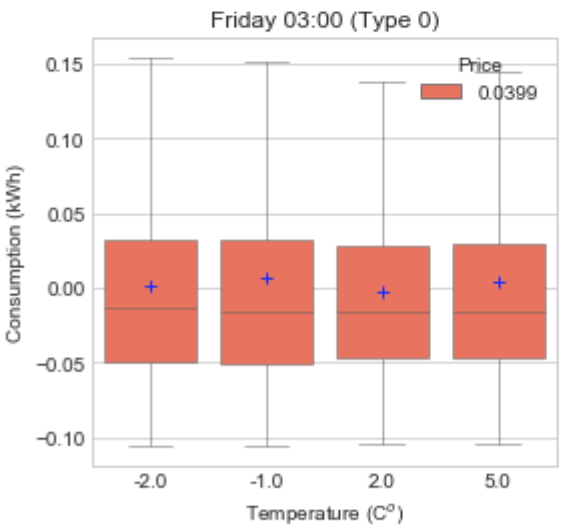

Friday 01:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)

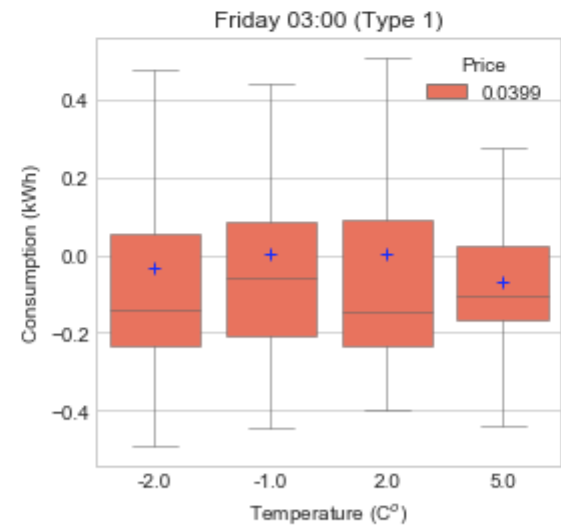

Friday 01:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)

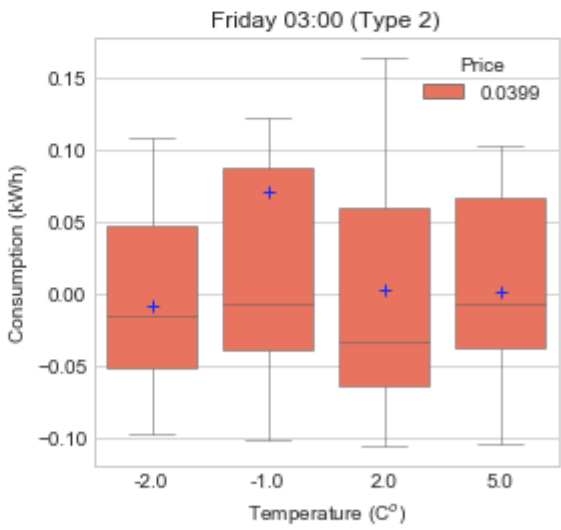

Friday 02:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)

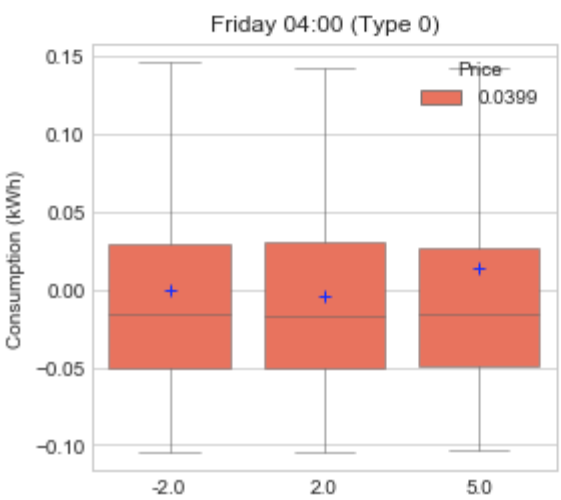

Friday 02:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)

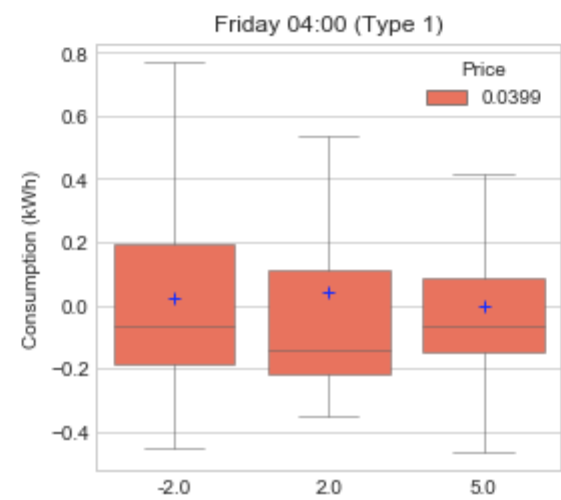

Friday 02:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)

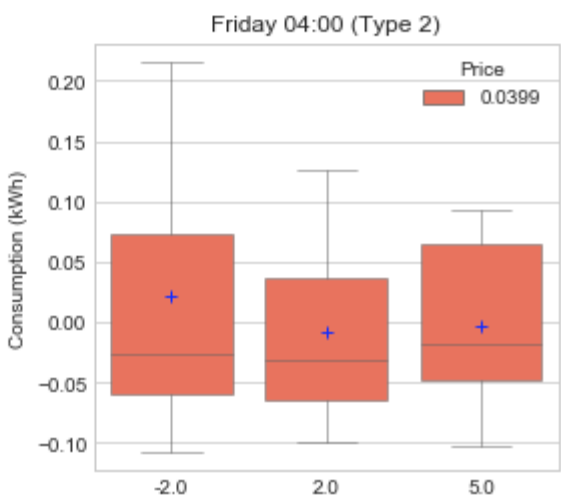

Friday 03:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)

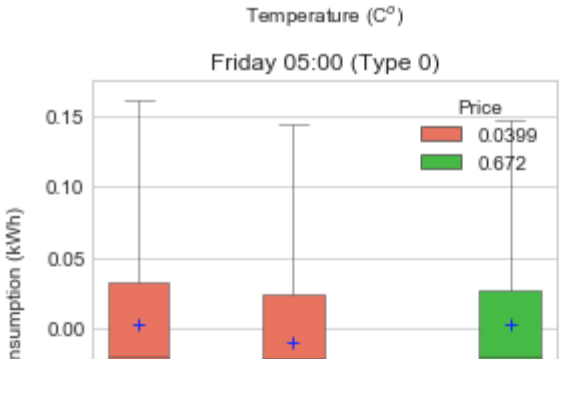

Friday 03:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)

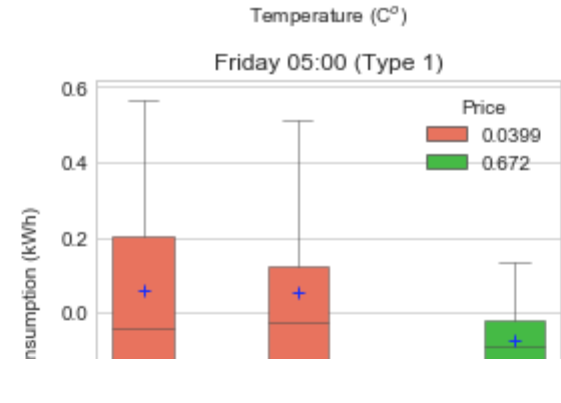

Friday 03:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)

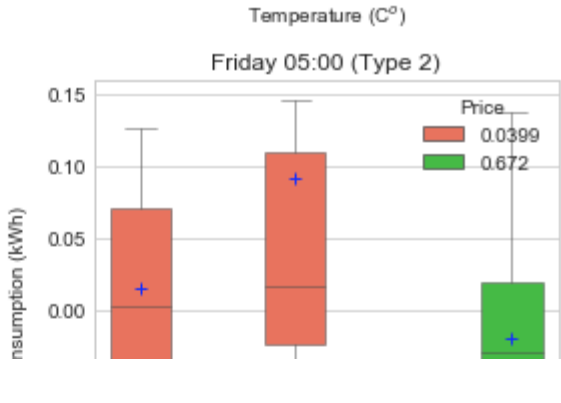

Friday 04:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)

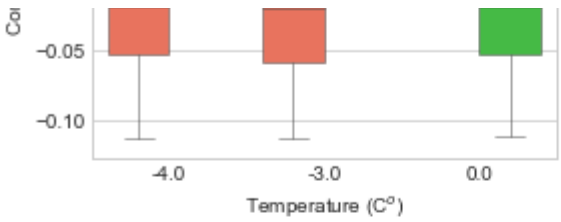

Friday 04:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)

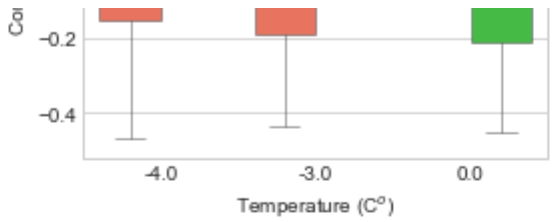

Friday 04:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)

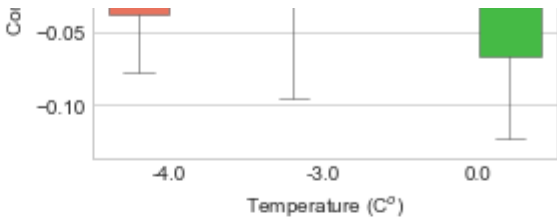

Friday 05:00 (Type 0)
Price
0.0399
0.672
Temperature (C°)

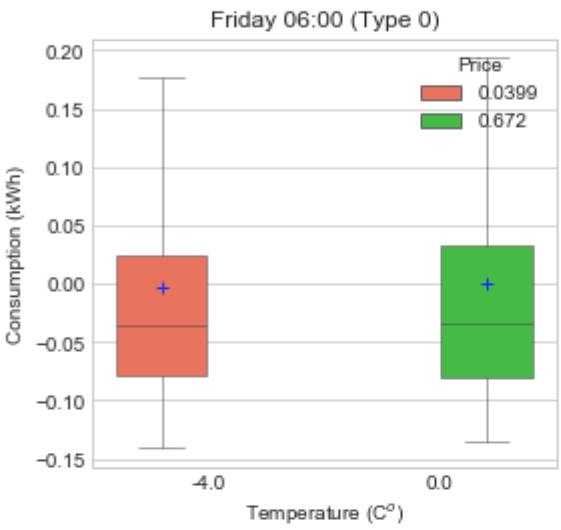

Friday 05:00 (Type 1)
Price
0.0399
0.672
Temperature (C°)

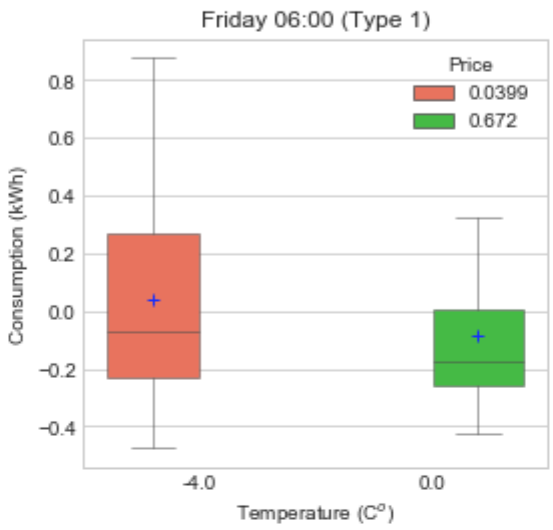

Friday 05:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

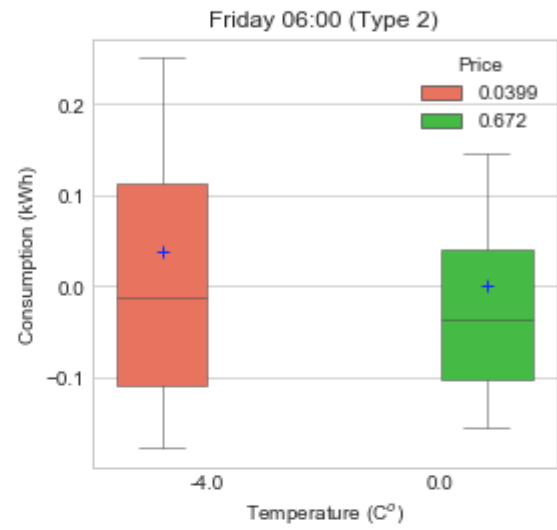

-0.05
-0.10
-4.0
-3.0
0.0
Temperature (C°)

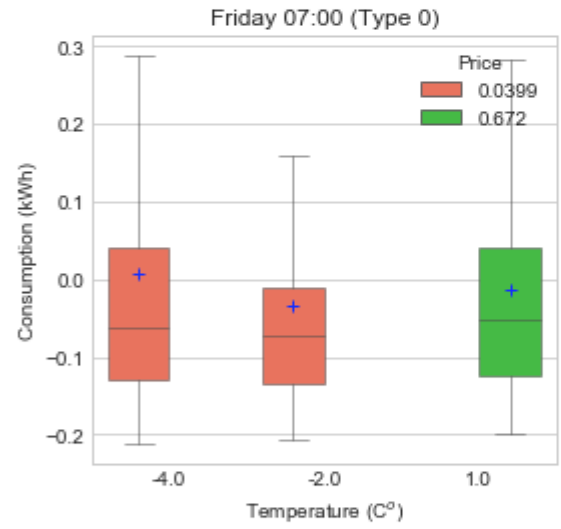

-0.2
-0.4
-4.0
-3.0
0.0
Temperature (C°)

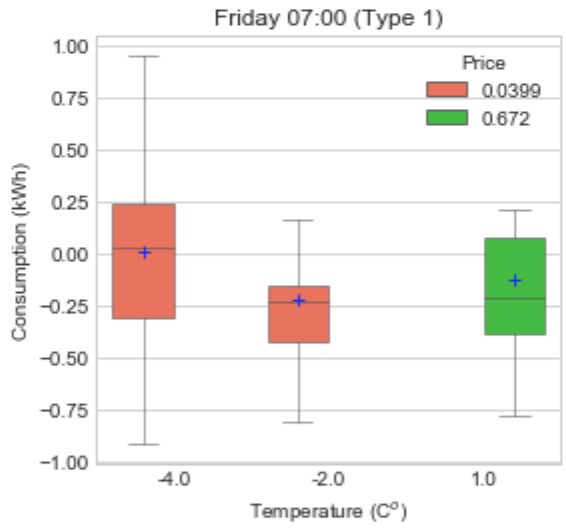

-0.05
-0.10
-4.0
-3.0
0.0
Temperature (C°)

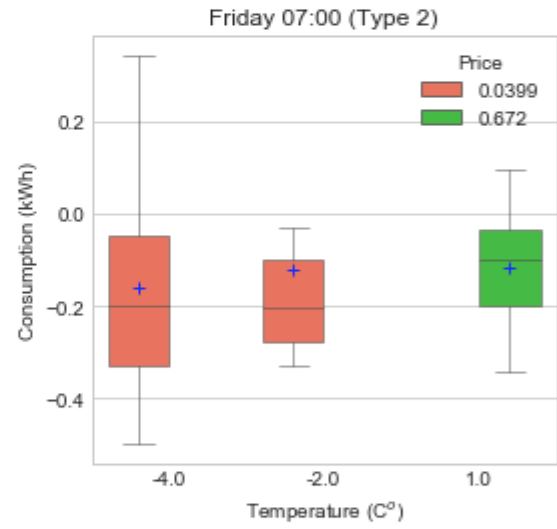

Friday 06:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
-4.0
0.0
Temperature (C°)

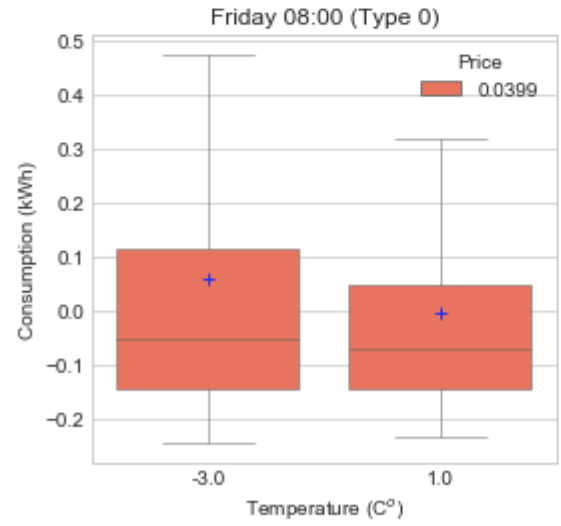

Friday 06:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-4.0
0.0
Temperature (C°)

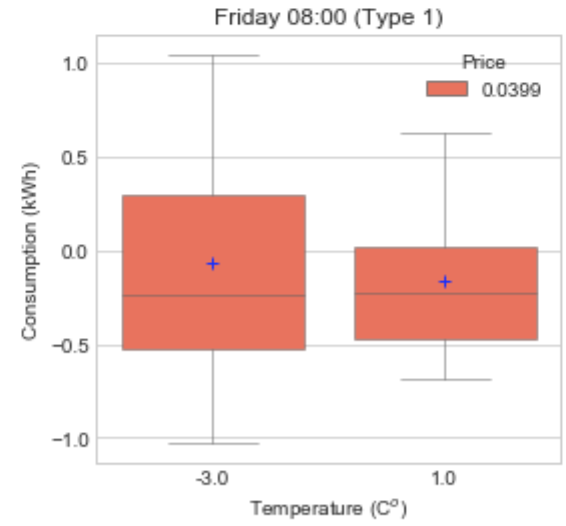

Friday 06:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.2
0.1
0.0
−0.1
-4.0
0.0
Temperature (C°)

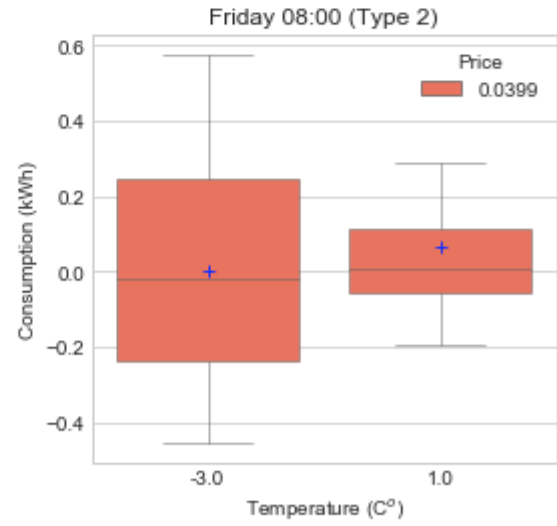

Friday 07:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
-4.0
-2.0
1.0
Temperature (C°)

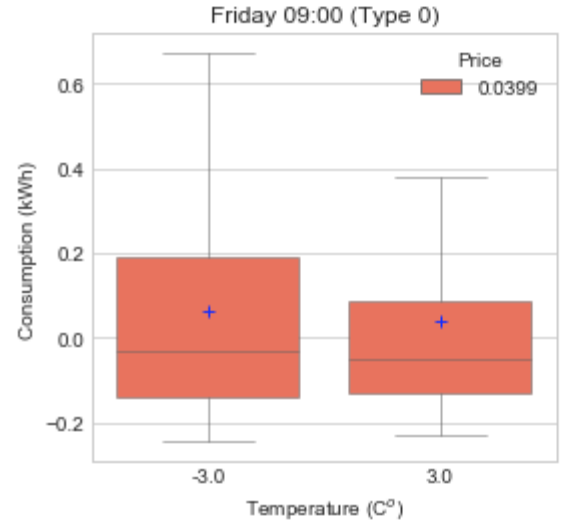

Friday 07:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-4.0
-2.0
1.0
Temperature (C°)

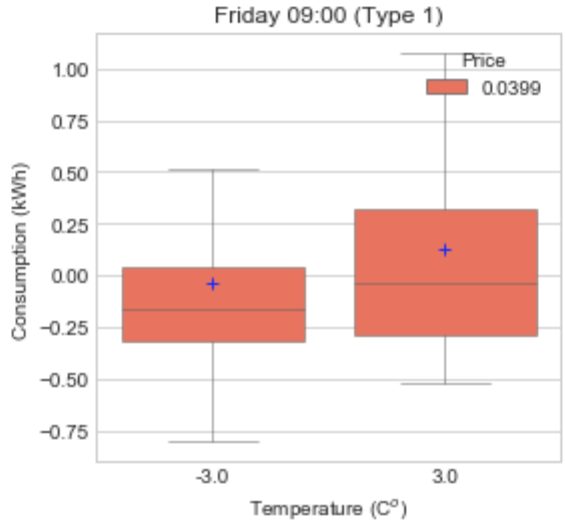

Friday 07:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.2
0.0
−0.2
−0.4
-4.0
-2.0
1.0
Temperature (C°)

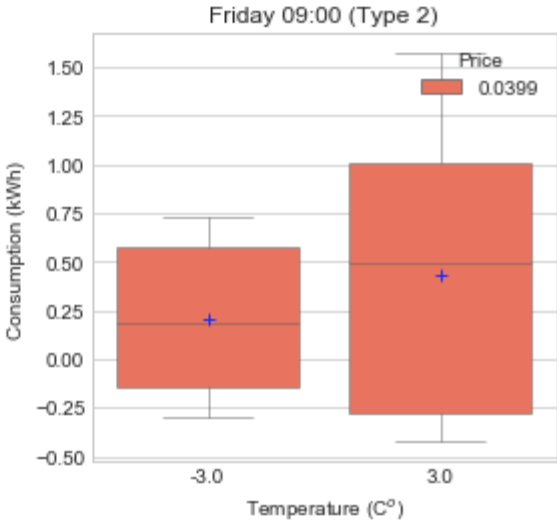

Friday 08:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-3.0
1.0
Temperature (C°)

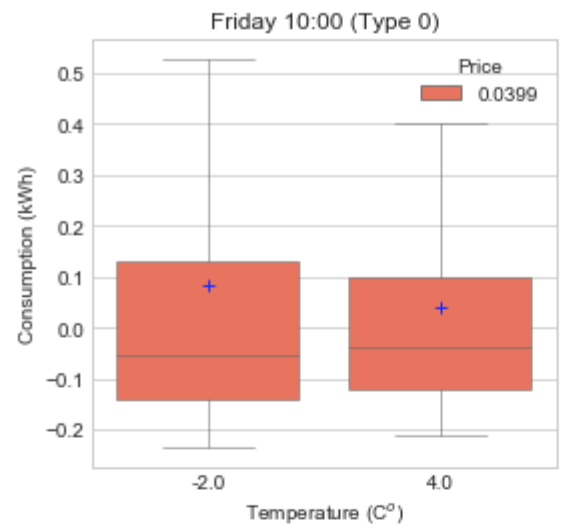

Friday 08:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-3.0
1.0
Temperature (C°)

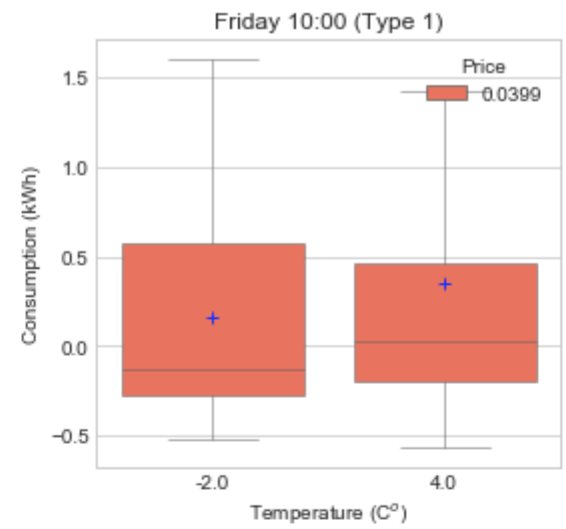

Friday 08:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-3.0
1.0
Temperature (C°)

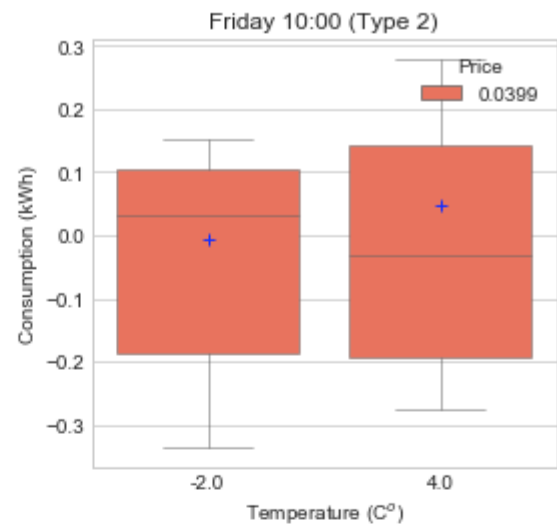

Friday 09:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
-3.0
3.0
Temperature (C°)


Friday 09:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-3.0
3.0
Temperature (C°)


Friday 09:00 (Type 2)
Price
0.0399
Consumption (kWh)
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
-3.0
3.0
Temperature (C°)


Friday 10:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-2.0
4.0
Temperature (C°)

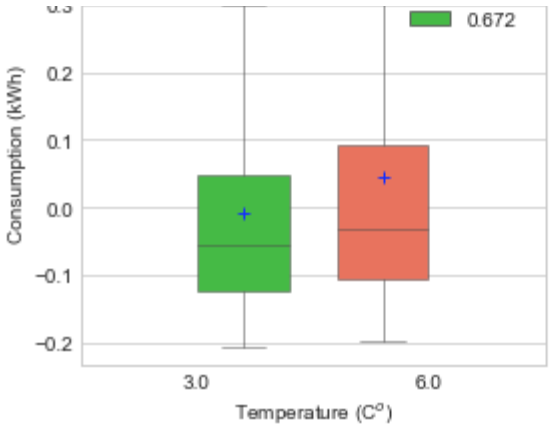

Friday 10:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-2.0
4.0
Temperature (C°)

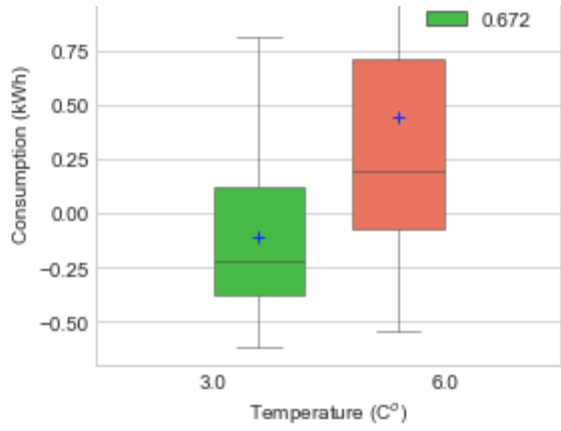

Friday 10:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-2.0
4.0
Temperature (C°)

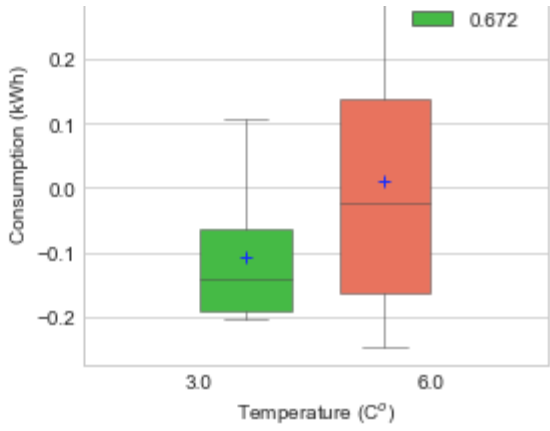
Friday 11:00 (Type 0)
Price
0.0399

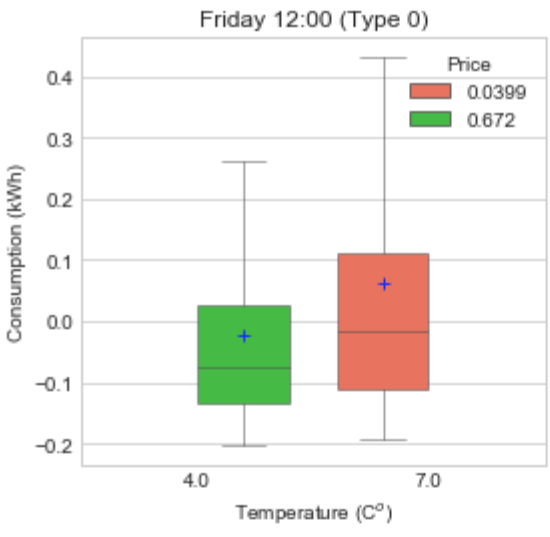
Friday 11:00 (Type 1)
Price
0.0399

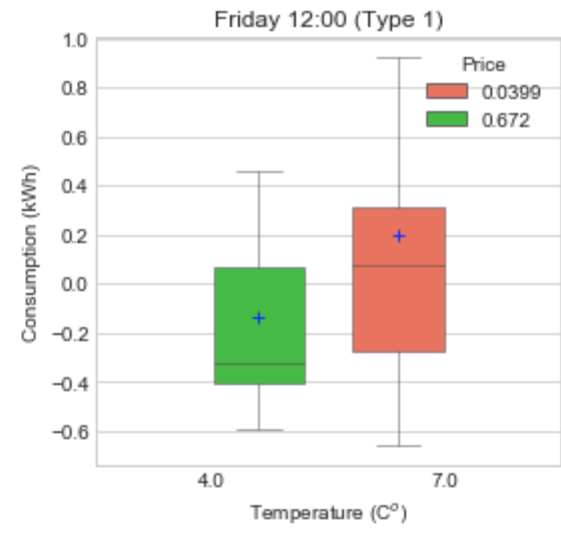
Friday 11:00 (Type 2)
Price
0.0399

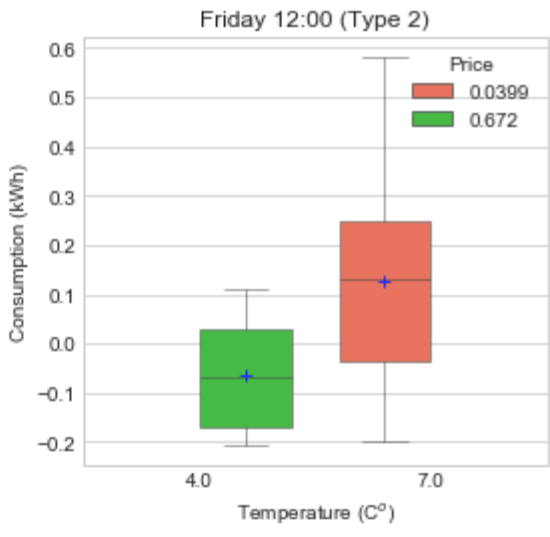

0.672
Consumption (kWh)
Temperature (C°)
3.0
6.0

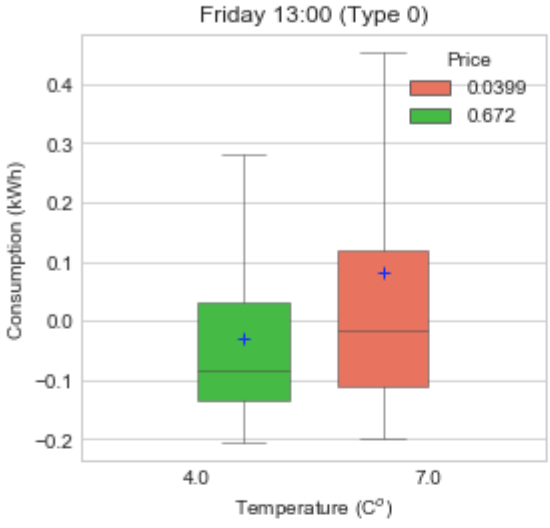

0.672
Consumption (kWh)
Temperature (C°)
3.0
6.0

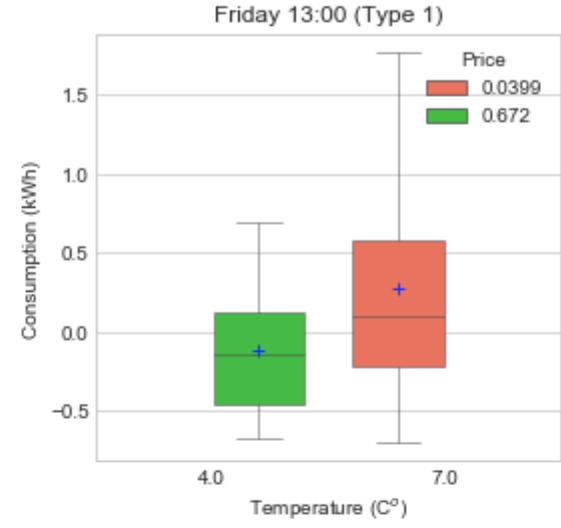

0.672
Consumption (kWh)
Temperature (C°)
3.0
6.0

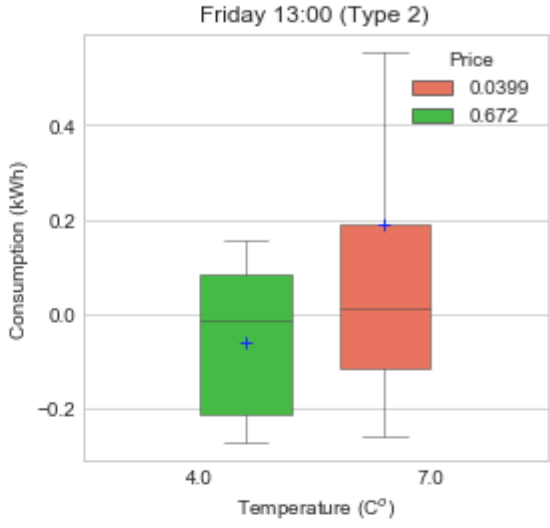

Friday 12:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
4.0
7.0

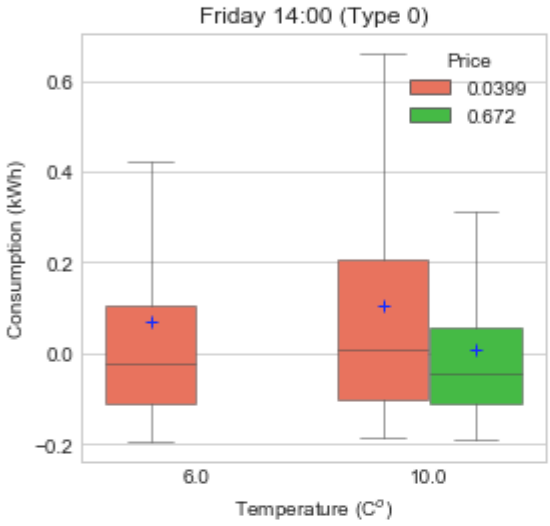

Friday 12:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
4.0
7.0

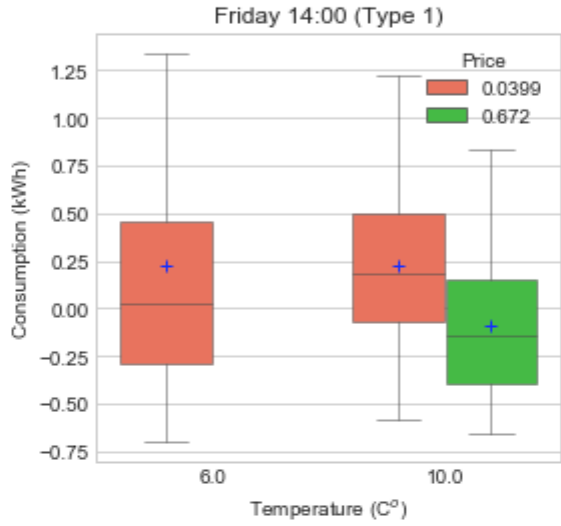

Friday 12:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
4.0
7.0

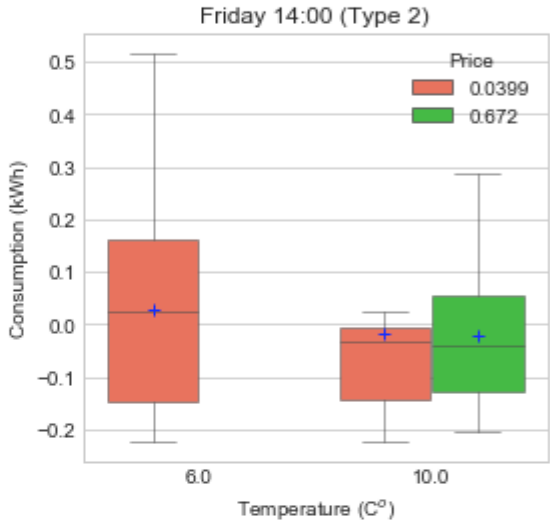

Friday 13:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
4.0
7.0

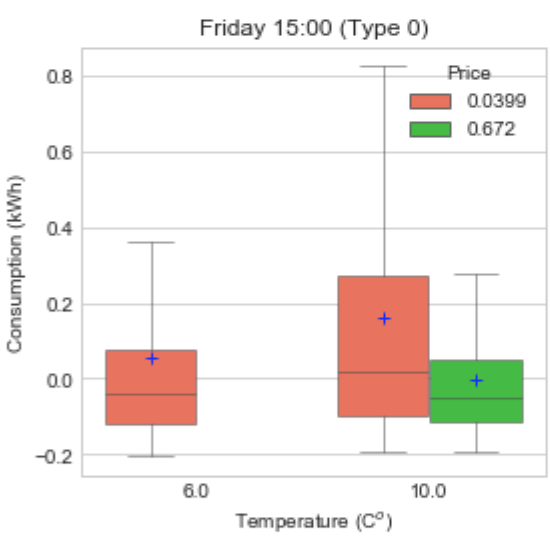

Friday 13:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
4.0
7.0

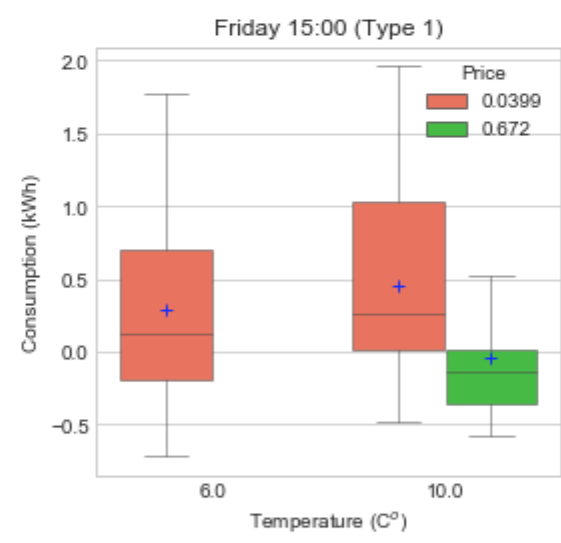

Friday 13:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
4.0
7.0

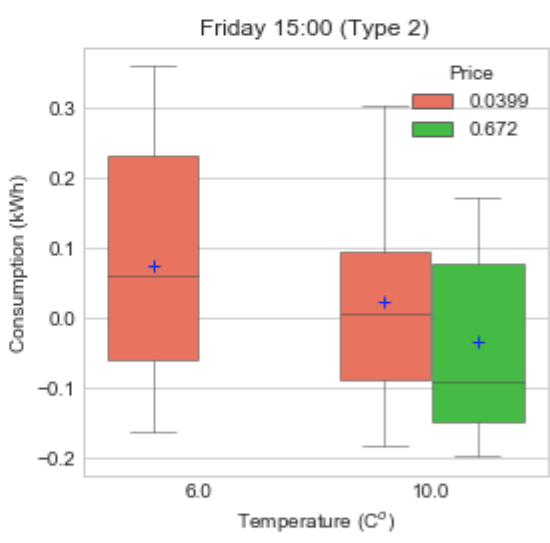

Friday 14:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
6.0
10.0

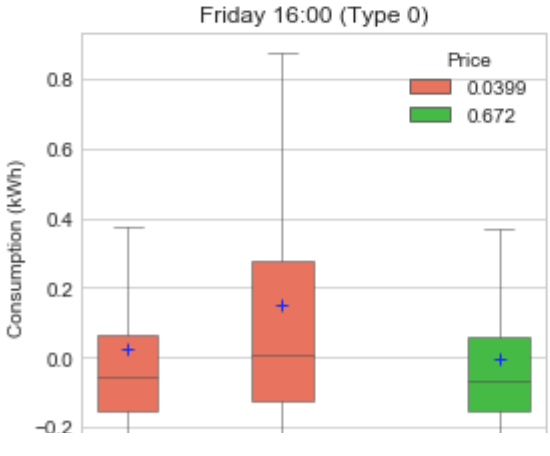

Friday 14:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
6.0
10.0

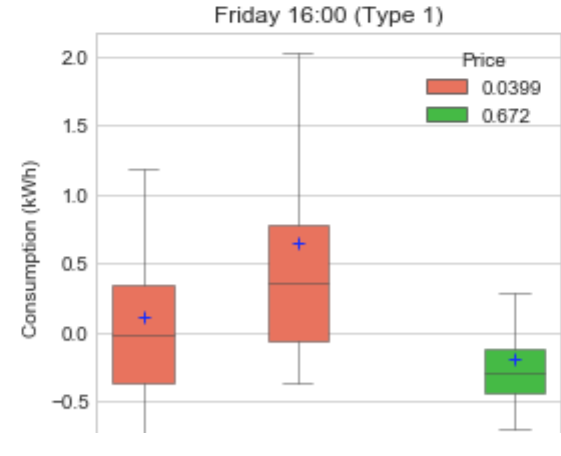

Friday 14:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
6.0
10.0

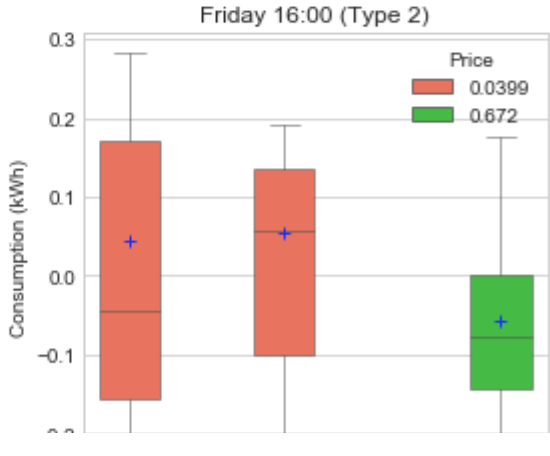

Friday 15:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
6.0
10.0

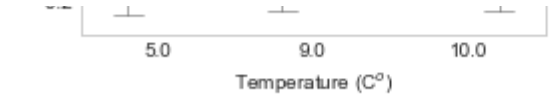

Friday 15:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
6.0
10.0

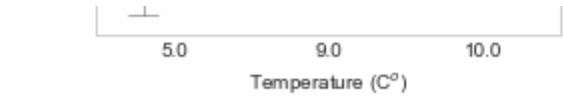

Friday 15:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
6.0
10.0

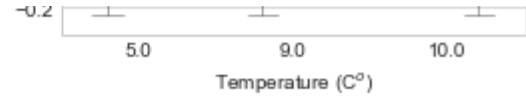

Friday 16:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)

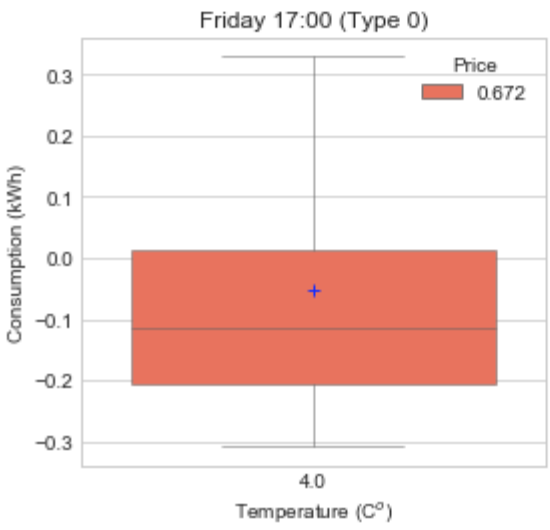

Friday 16:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)

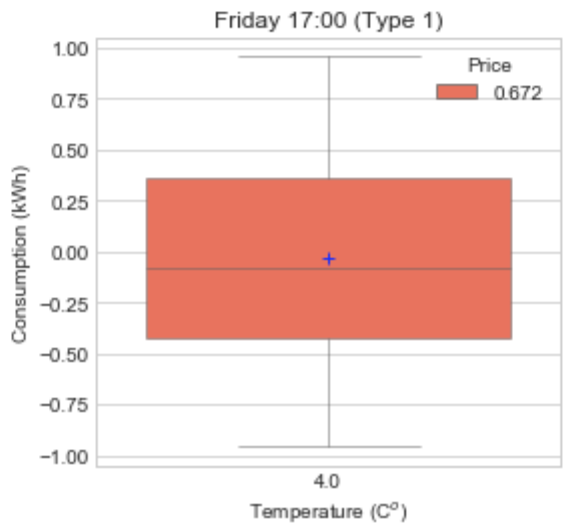

Friday 16:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)

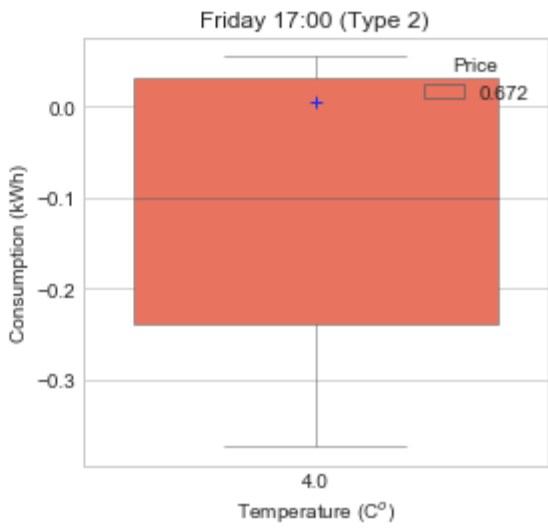

5.0
9.0
10.0
Temperature (C°)

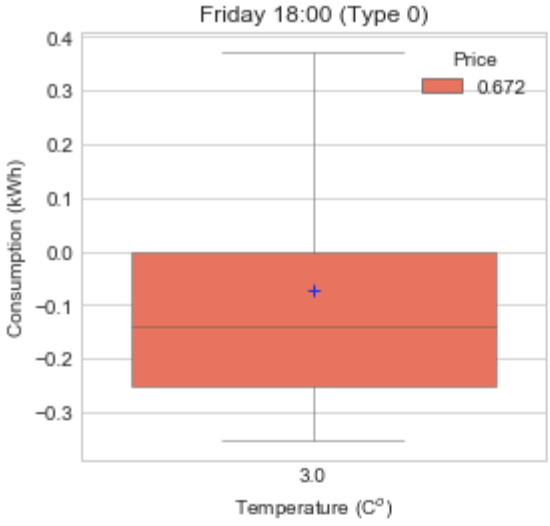

5.0
9.0
10.0
Temperature (C°)

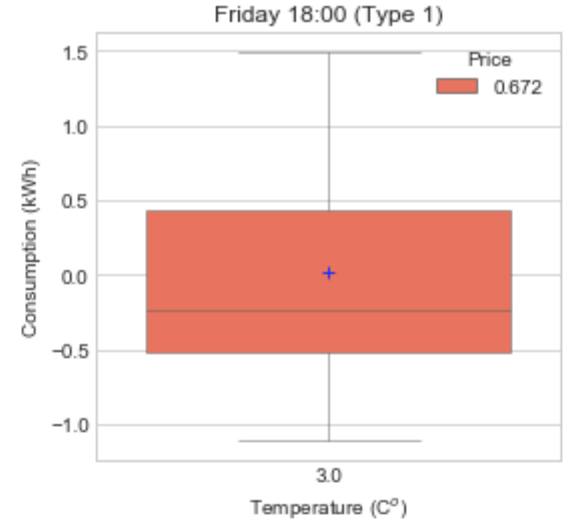

5.0
9.0
10.0
Temperature (C°)

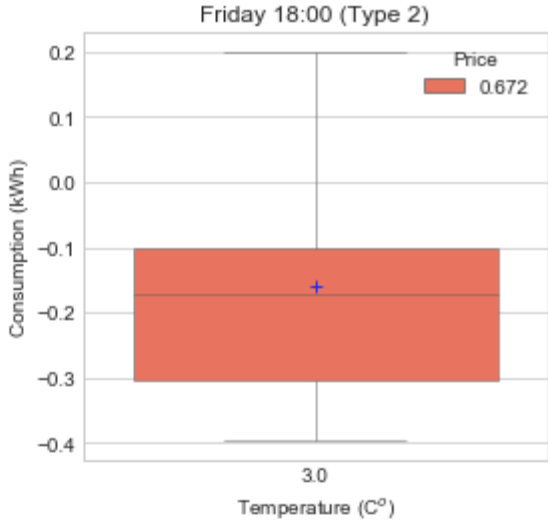

Friday 17:00 (Type 0)
Price
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
4.0
Temperature (C°)

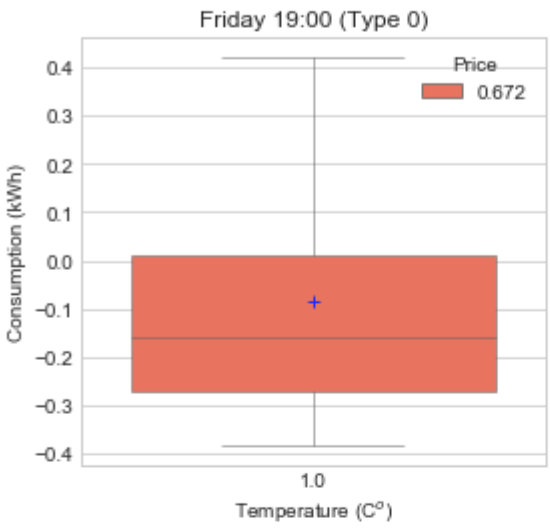

Friday 17:00 (Type 1)
Price
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
4.0
Temperature (C°)

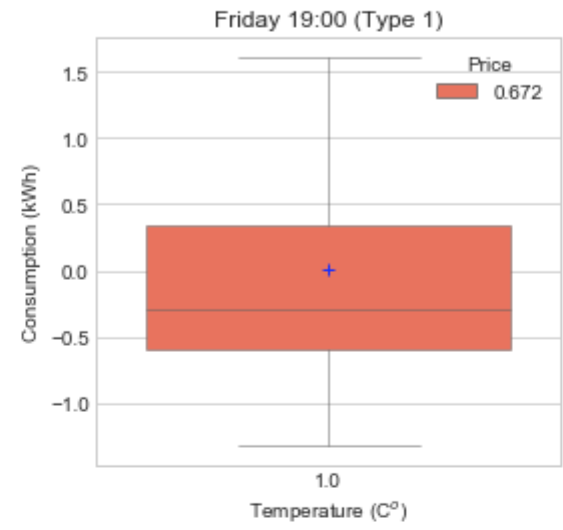

Friday 17:00 (Type 2)
Price
0.672
Consumption (kWh)
0.0
−0.1
−0.2
−0.3
4.0
Temperature (C°)

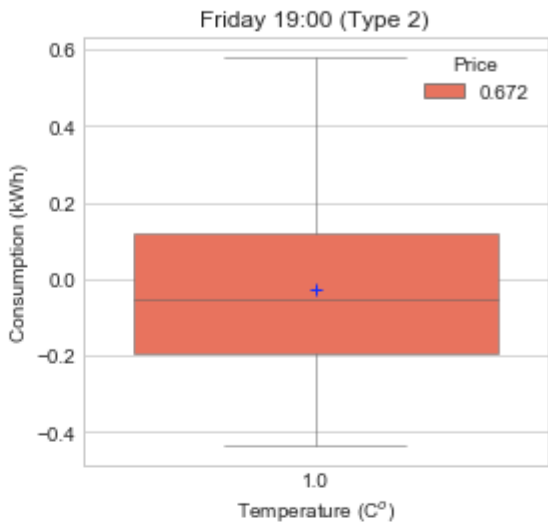

Friday 18:00 (Type 0)
Price
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
3.0
Temperature (C°)

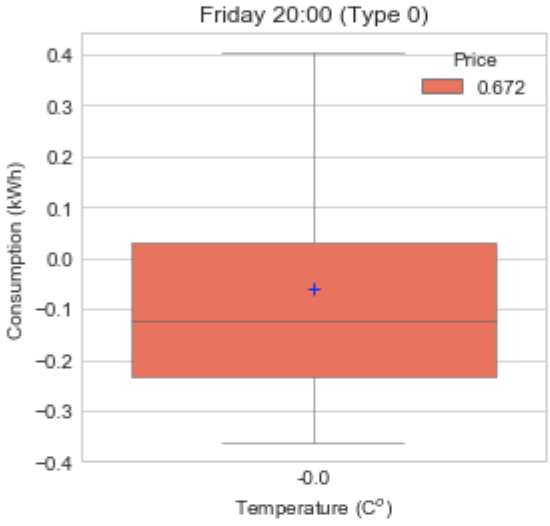

Friday 18:00 (Type 1)
Price
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
3.0
Temperature (C°)

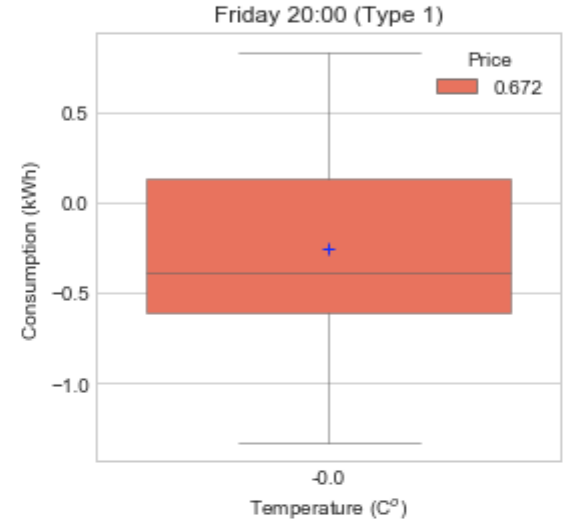

Friday 18:00 (Type 2)
Price
0.672
Consumption (kWh)
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
3.0
Temperature (C°)

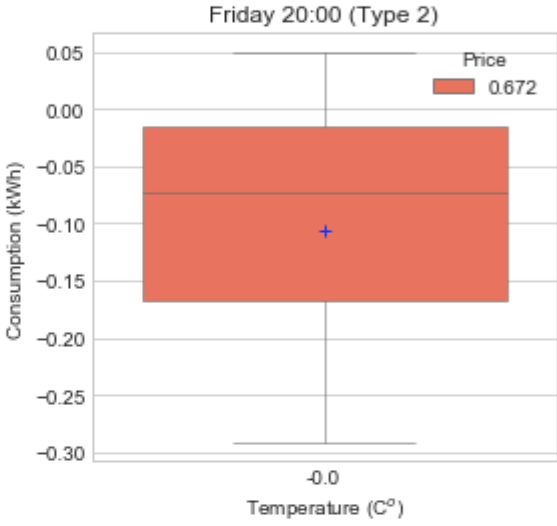

Friday 19:00 (Type 0)
Price
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
1.0
Temperature (C°)

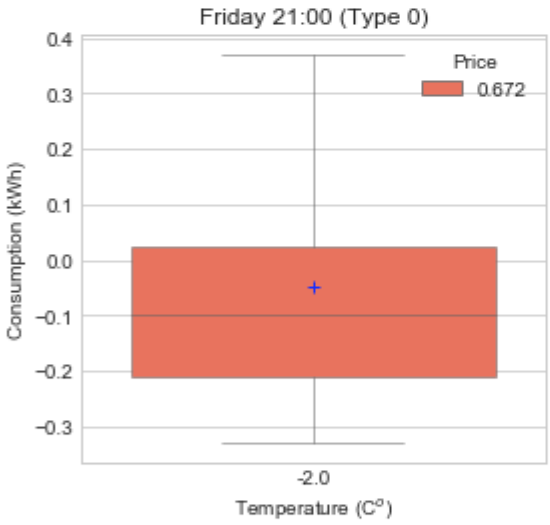

Friday 19:00 (Type 1)
Price
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
1.0
Temperature (C°)

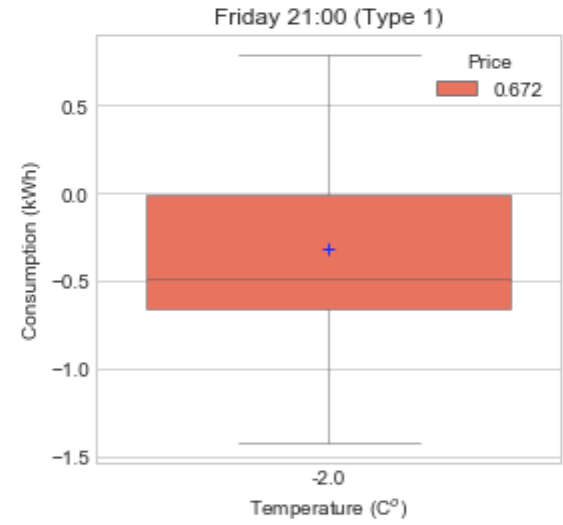

Friday 19:00 (Type 2)
Price
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
1.0
Temperature (C°)

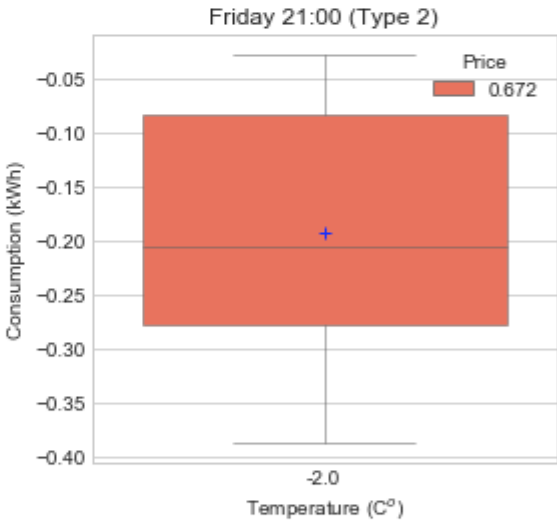

Friday 20:00 (Type 0)
Price
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
-0.0
Temperature (C°)


Friday 20:00 (Type 1)
Price
0.672
Consumption (kWh)
0.5
0.0
−0.5
−1.0
-0.0
Temperature (C°)

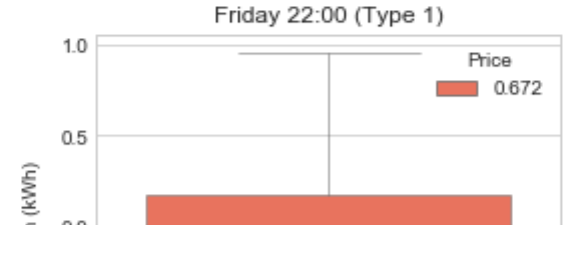

Friday 20:00 (Type 2)
Price
0.672
Consumption (kWh)
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
-0.0
Temperature (C°)

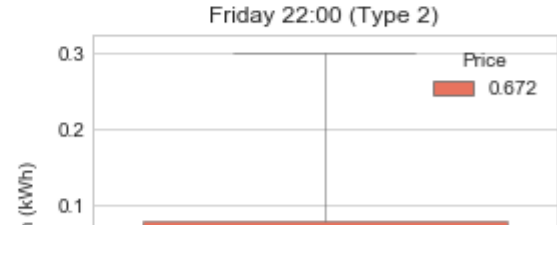

Friday 21:00 (Type 0)
Price
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-2.0
Temperature (C°)

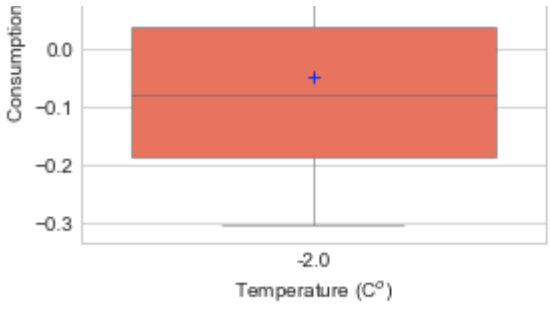

Friday 21:00 (Type 1)
Price
0.672
Consumption (kWh)
0.5
0.0
−0.5
−1.0
−1.5
-2.0
Temperature (C°)

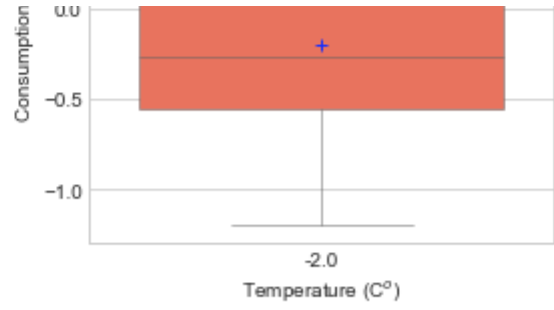

Friday 21:00 (Type 2)
Price
0.672
Consumption (kWh)
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
−0.35
−0.40
-2.0
Temperature (C°)

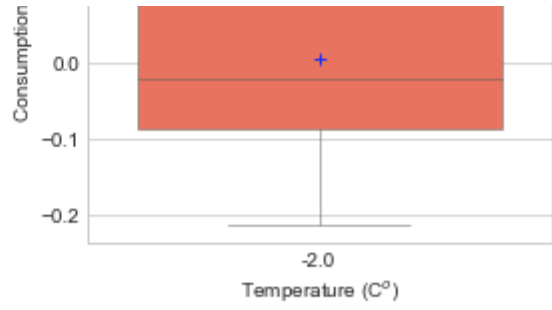

Friday 22:00 (Type 0)
Price
0.672
0.4
0.3
0.2
0.1

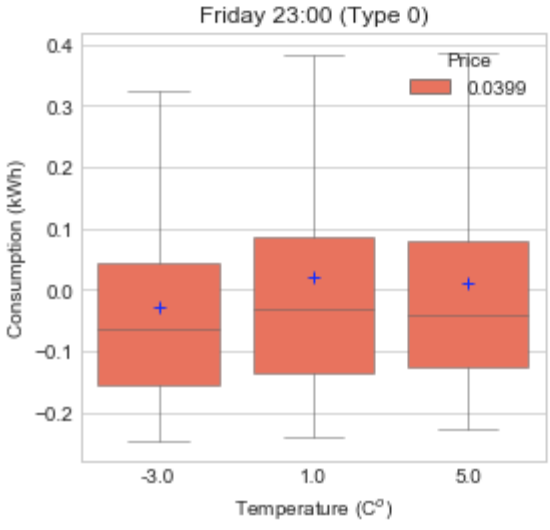

Friday 22:00 (Type 1)
Price
0.672
1.0
0.5

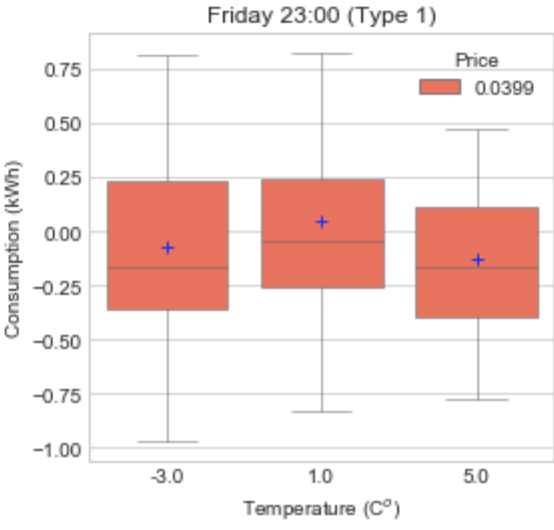

Friday 22:00 (Type 2)
Price
0.672
0.3
0.2
0.1

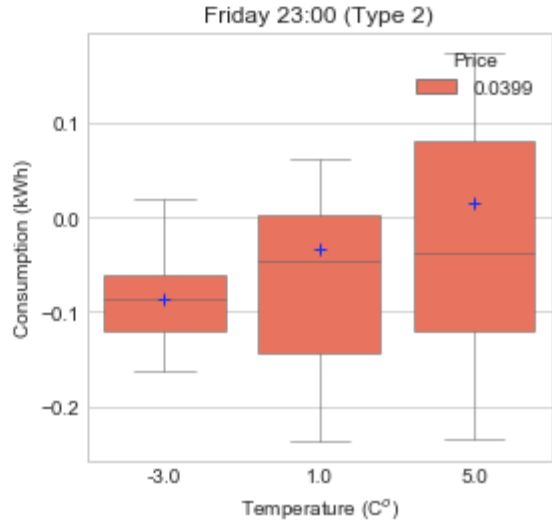

Consumption
0.0
−0.1
−0.2
−0.3
-2.0
Temperature (C°)


Consumption
0.0
−0.5
−1.0
-2.0
Temperature (C°)

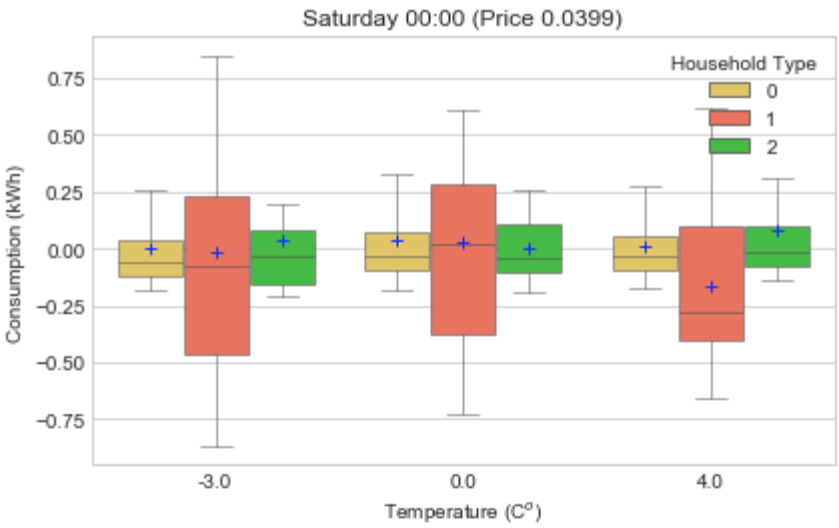

Consumption
0.0
−0.1
−0.2
-2.0
Temperature (C°)

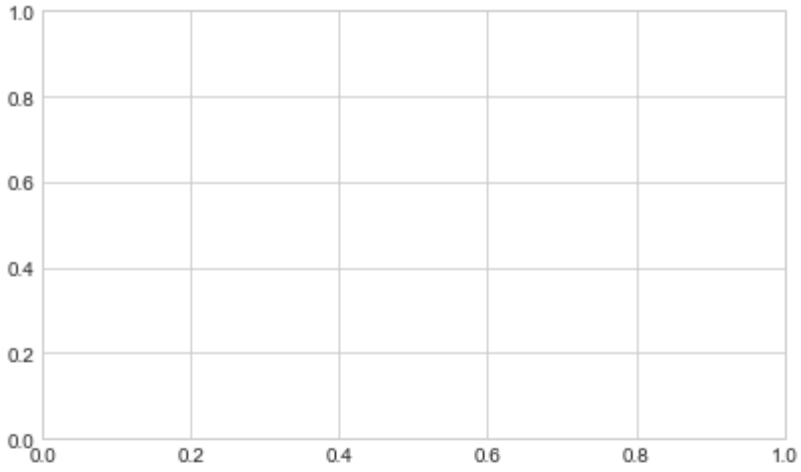

Friday 23:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-3.0
1.0
5.0
Temperature (C°)

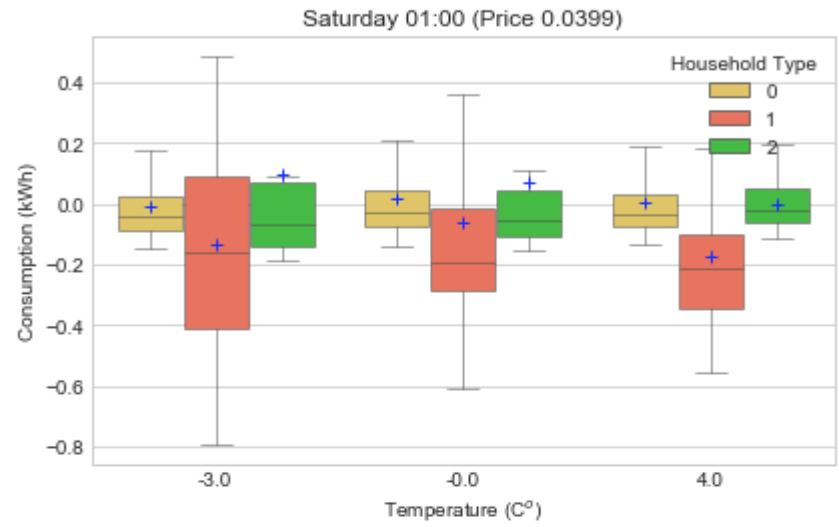

Friday 23:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-3.0
1.0
5.0
Temperature (C°)

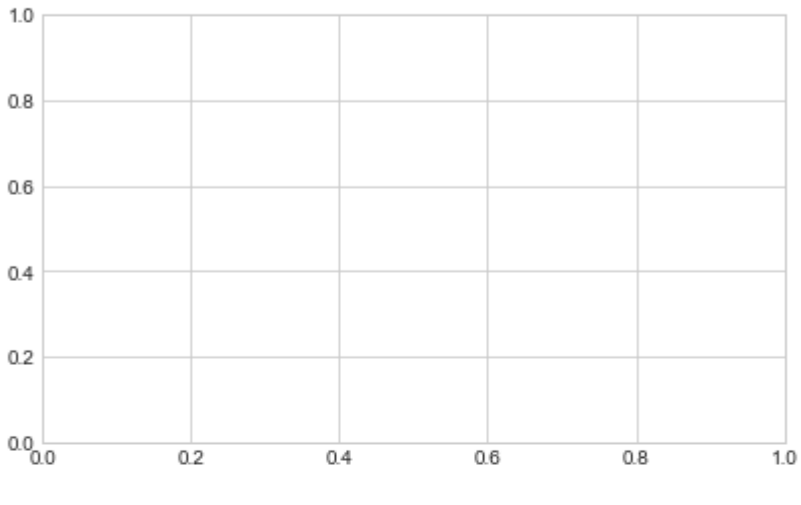

Friday 23:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.1
0.0
−0.1
−0.2
-3.0
1.0
5.0
Temperature (C°)

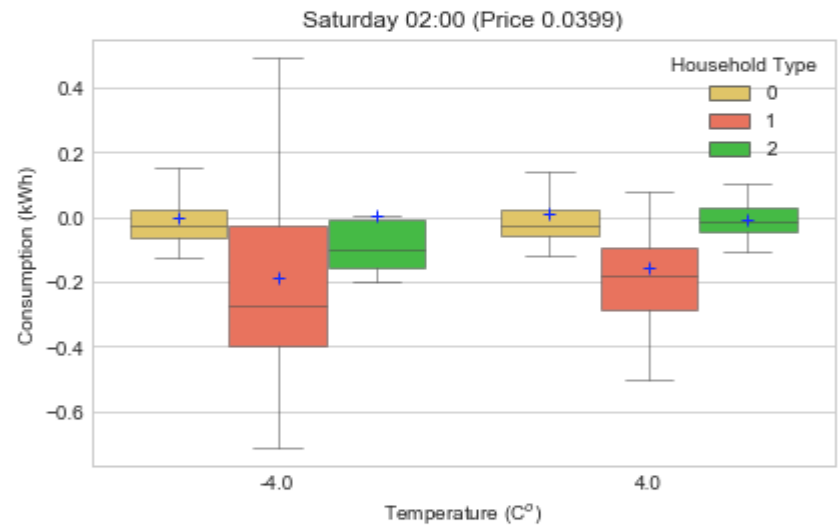
In [102]:
```
# Saturday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 5 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

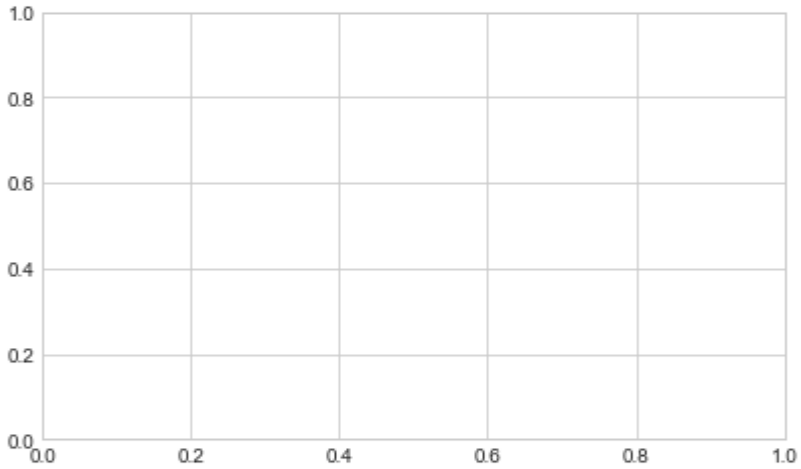

Saturday 00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-3.0
0.0
4.0
Temperature (C°)

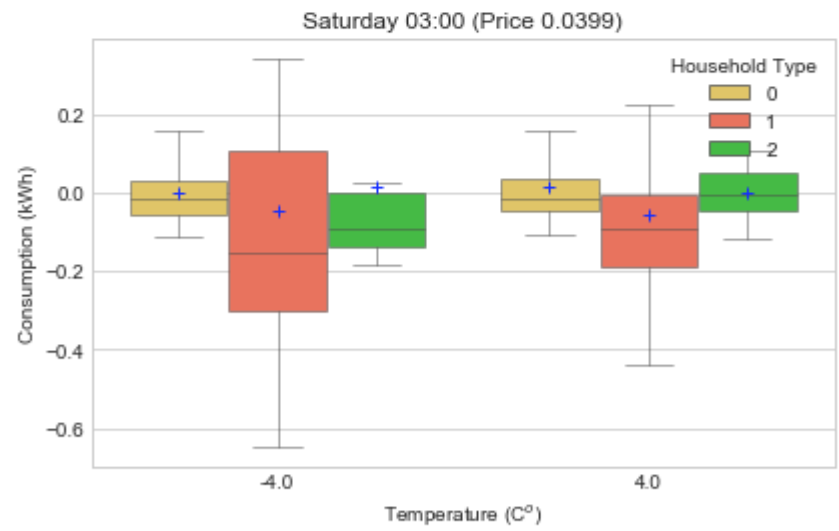

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

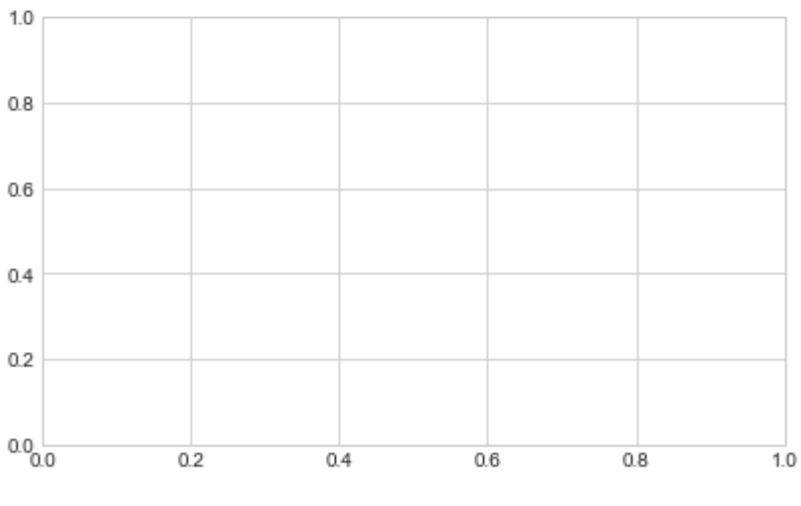

Saturday 01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
-3.0
-0.0
4.0
Temperature (C°)

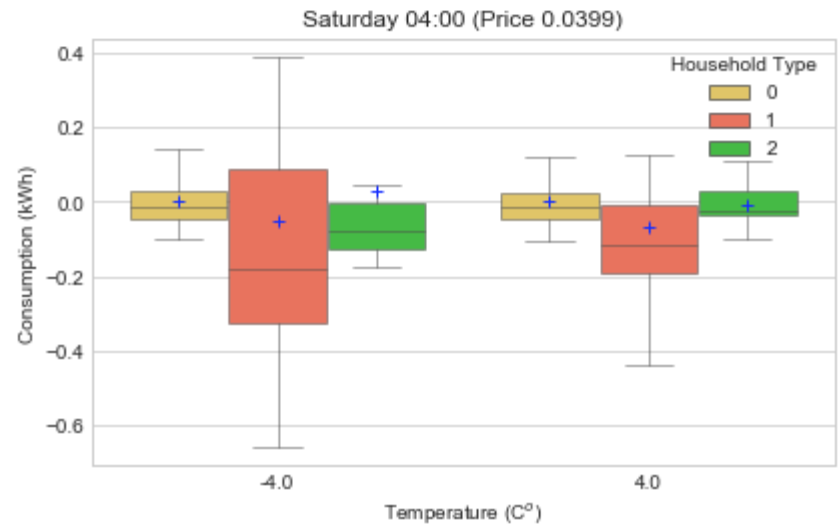

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

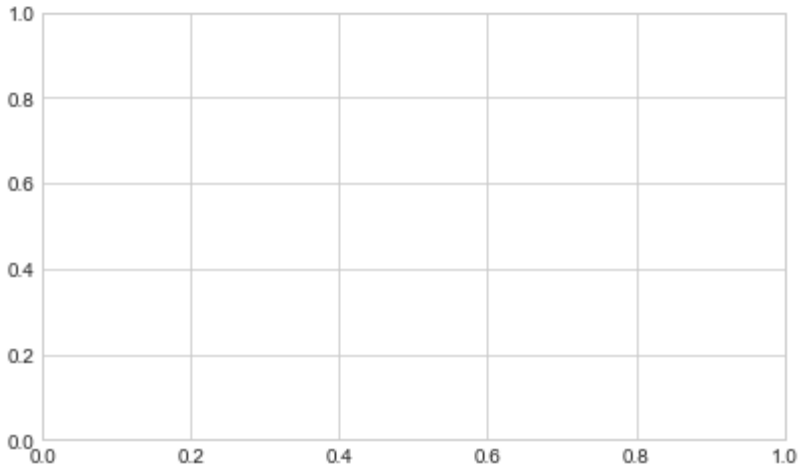

Saturday 02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
4.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

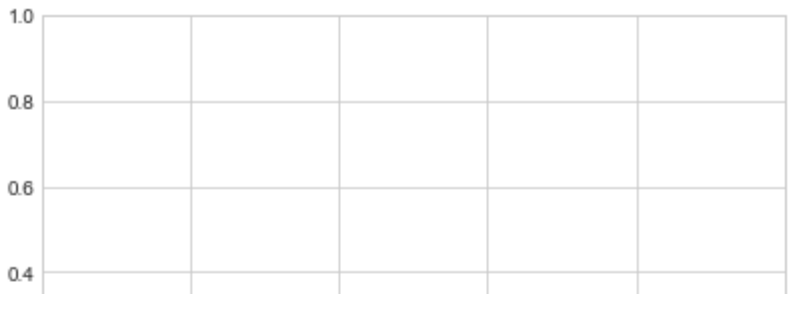

Saturday 03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.2
0.0
−0.2
−0.4
−0.6
-4.0
4.0
Temperature (C°)

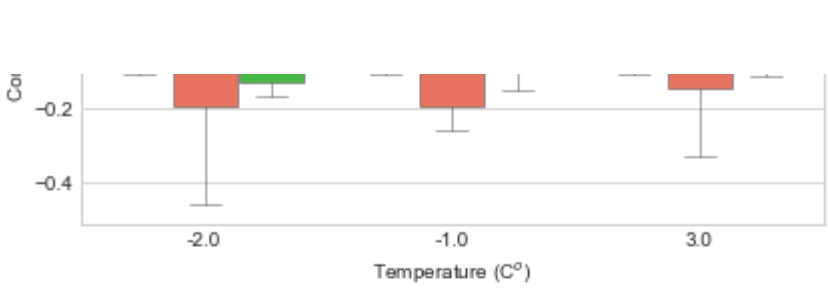

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

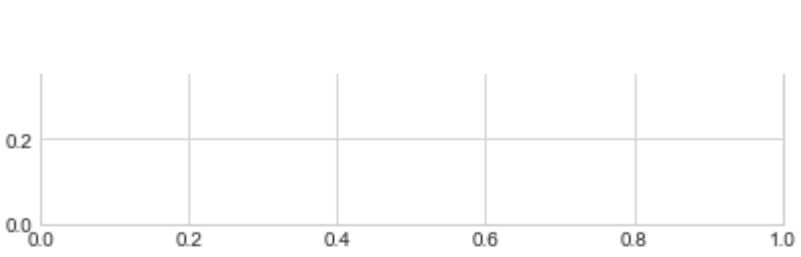

Saturday 04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
4.0
Temperature (C°)

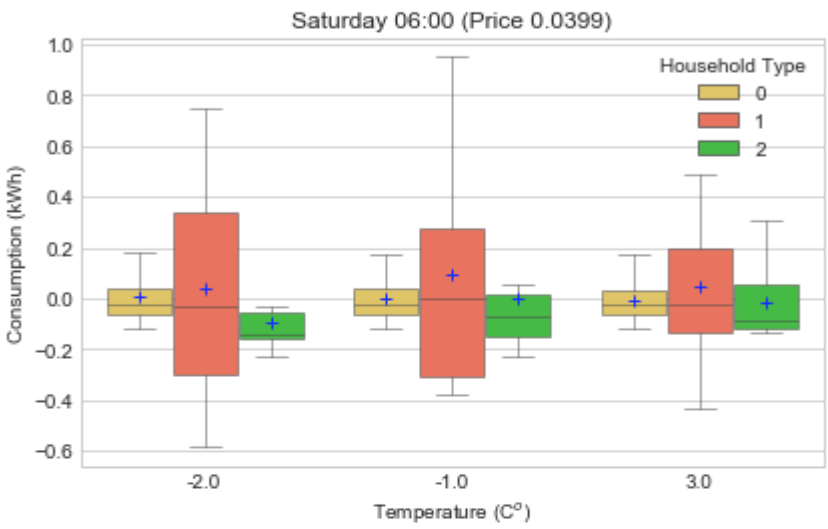

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

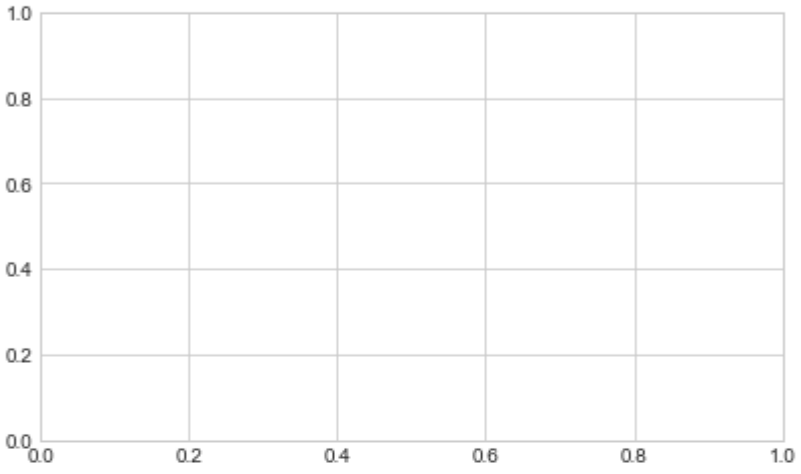

Saturday 05:00 (Price 0.0399)
Household Type
0
1
2
0.6
0.4
0.2
0.0

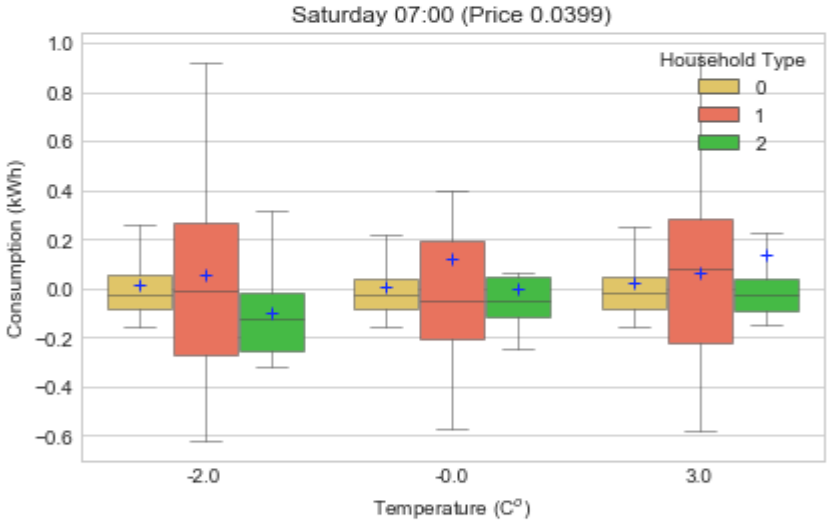

1.0
0.8
0.6
0.4

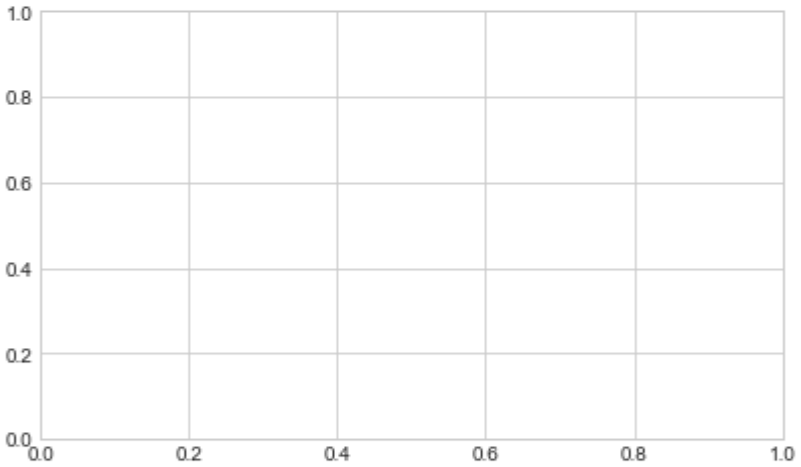

-0.2
-0.4
-2.0
-1.0
3.0
Temperature (C°)

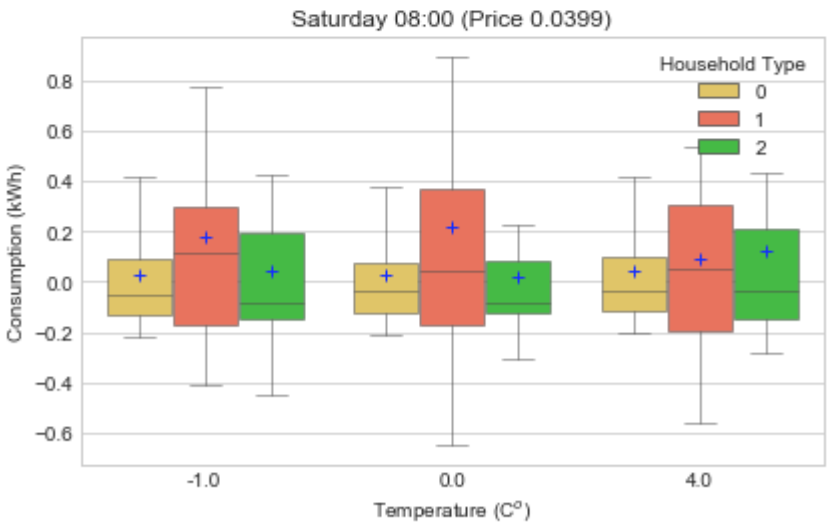

0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

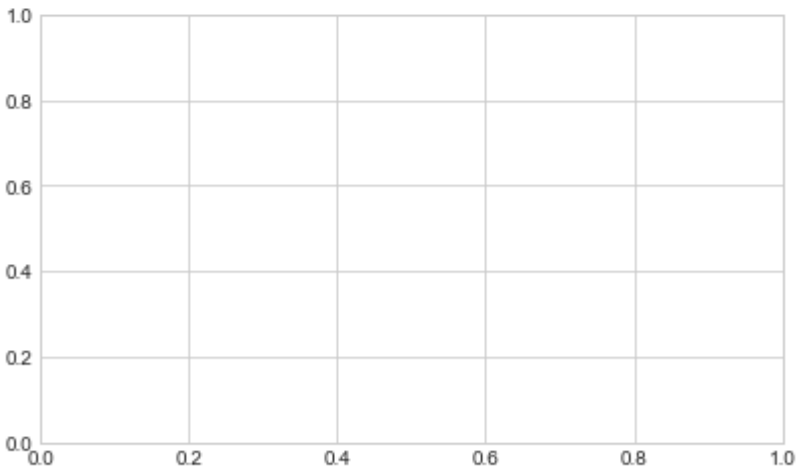

Saturday 06:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-2.0
-1.0
3.0
Temperature (C°)

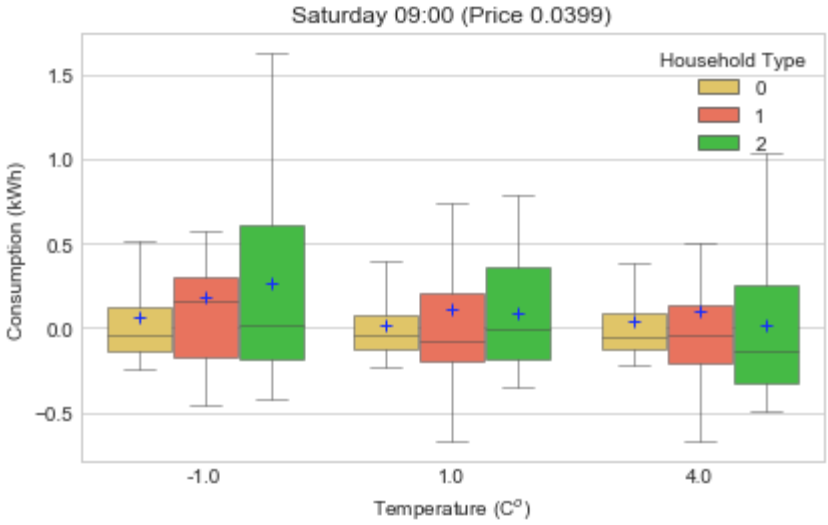

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

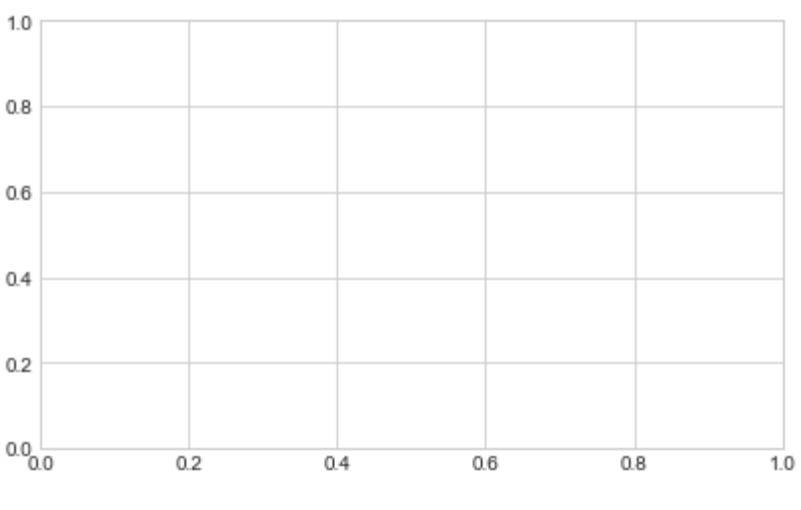

Saturday 07:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-2.0
-0.0
3.0
Temperature (C°)

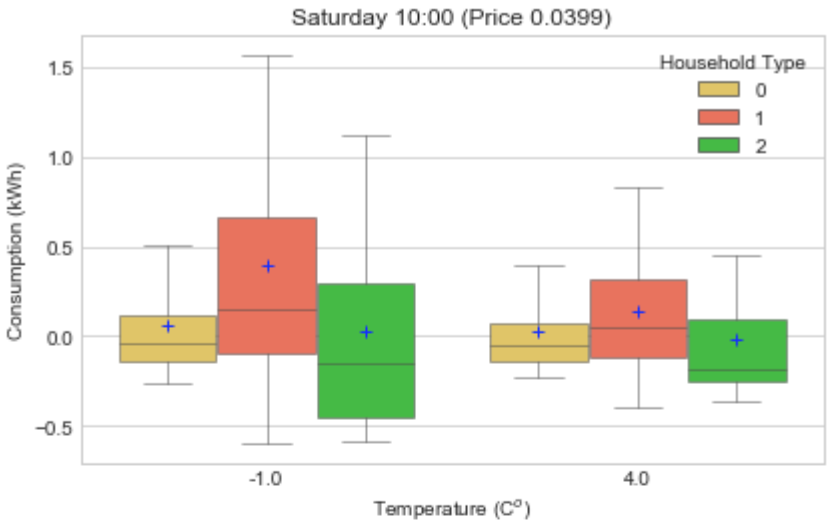

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

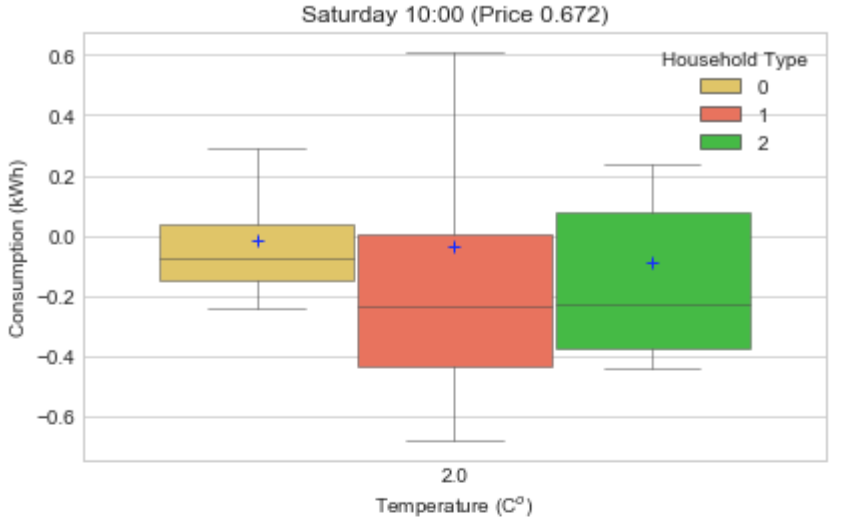

Saturday 08:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-1.0
0.0
4.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Saturday 09:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
-0.5
-1.0
1.0
4.0
Temperature (C°)

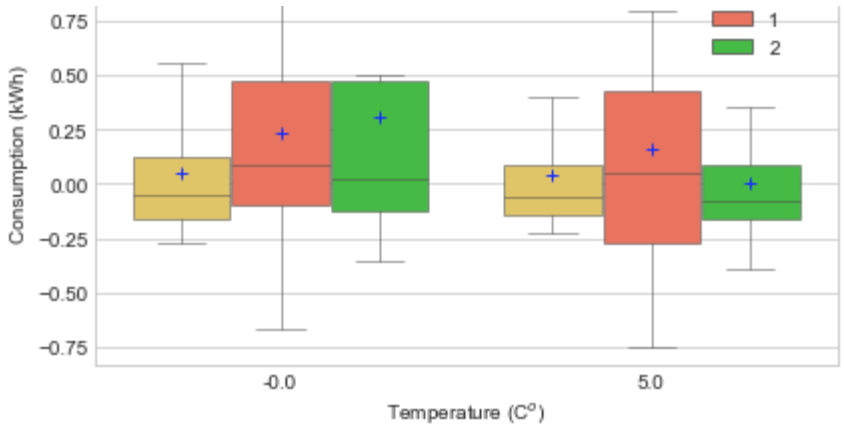

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

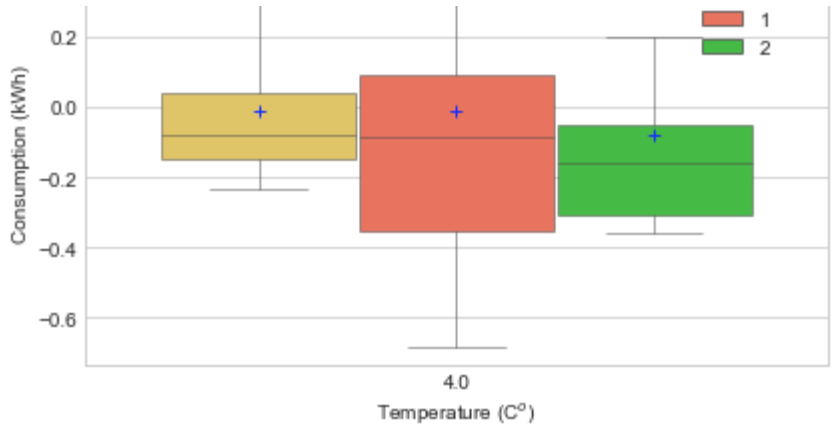

Saturday 10:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
-0.5
-1.0
4.0
Temperature (C°)

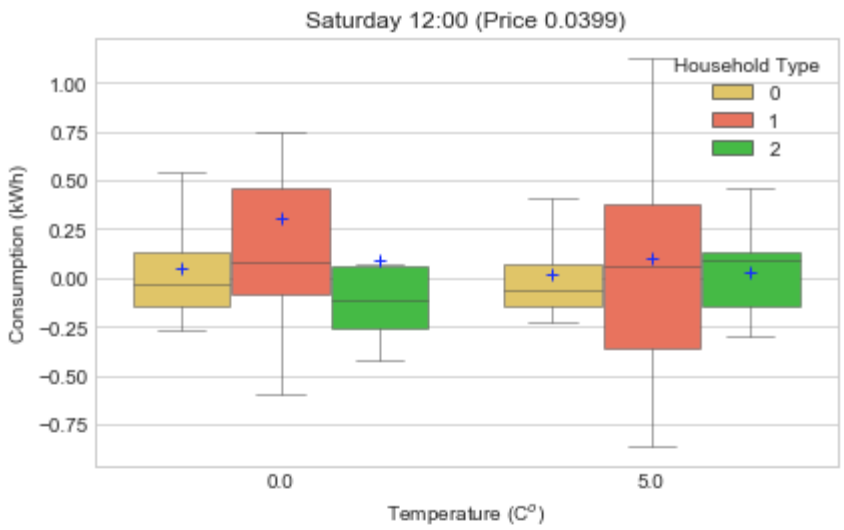

Saturday 10:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
2.0
Temperature (C°)

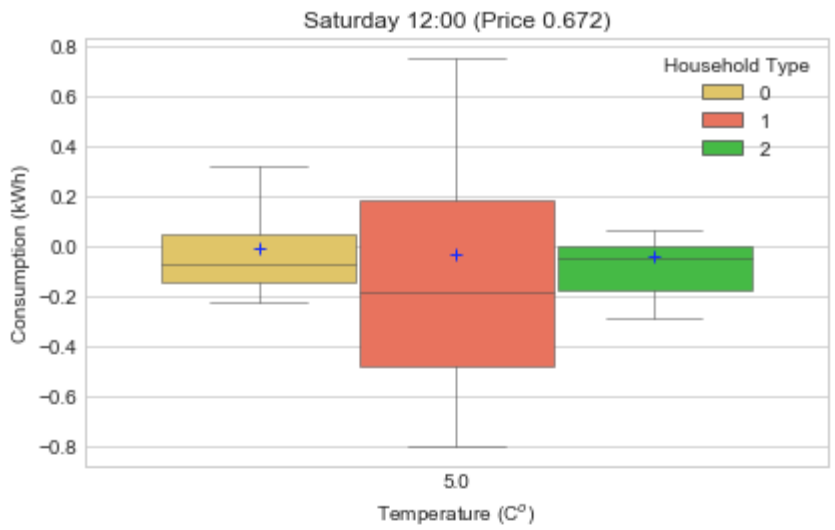
Saturday 11:00 (Price 0.0399)
Household Type
0
1.00

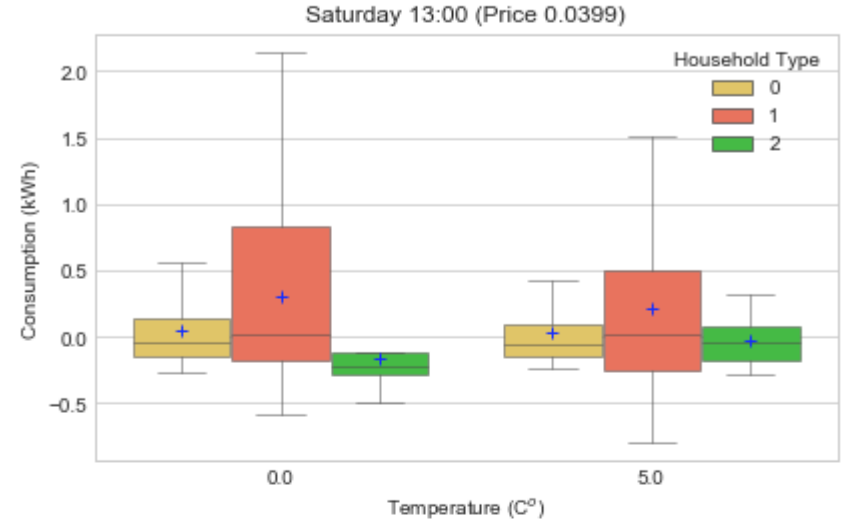
Saturday 11:00 (Price 0.672)
Household Type
0
0.4

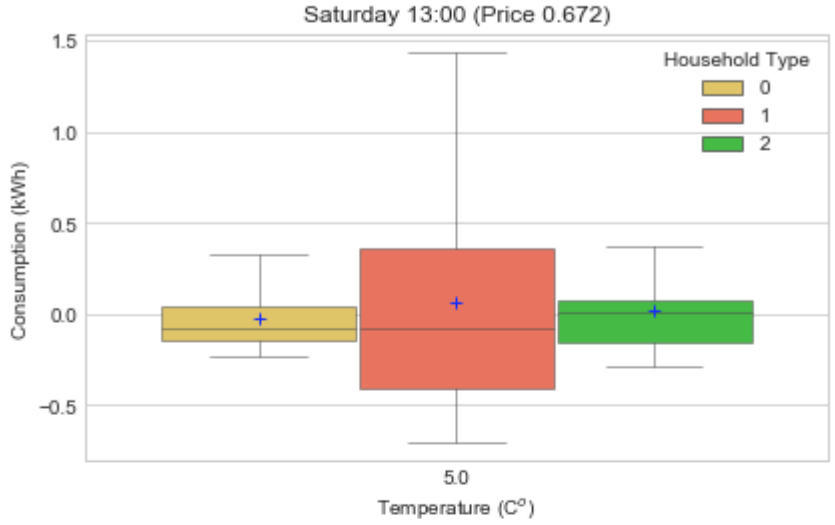

Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
1
2
-0.0
5.0
Temperature (C°)

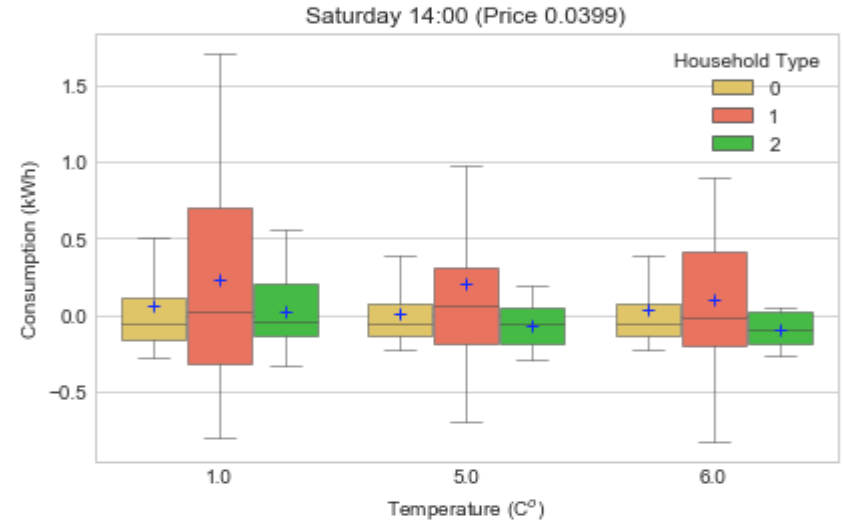

Consumption (kWh)
0.2
0.0
−0.2
−0.4
−0.6
1
2
4.0
Temperature (C°)

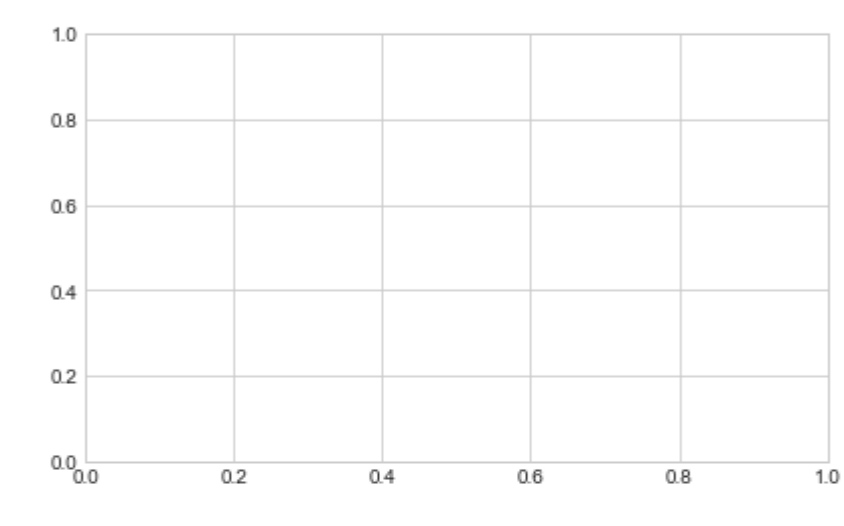

Saturday 12:00 (Price 0.0399)
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
Household Type
0
1
2
0.0
5.0
Temperature (C°)

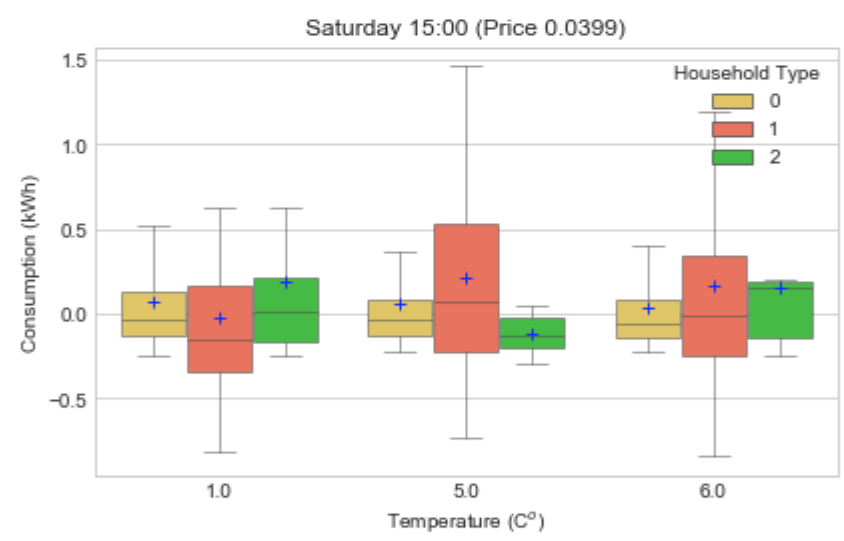

Saturday 12:00 (Price 0.672)
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
Household Type
0
1
2
5.0
Temperature (C°)

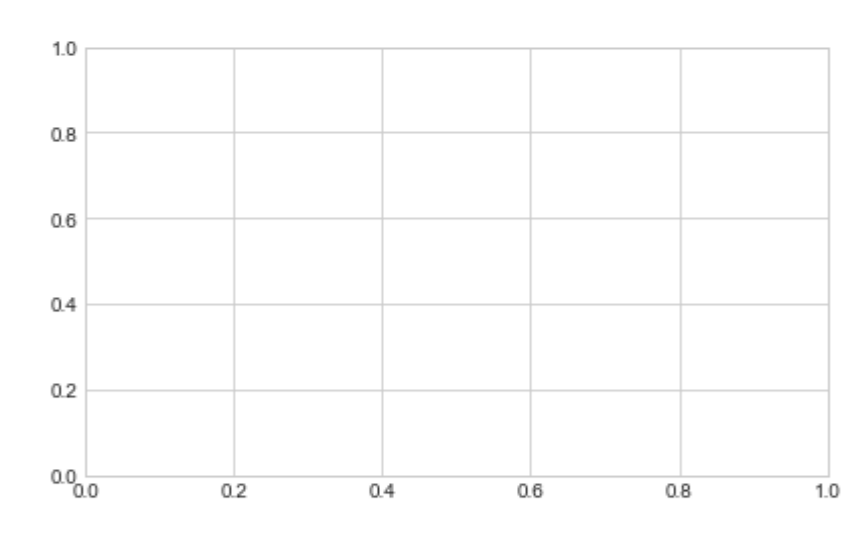

Saturday 13:00 (Price 0.0399)
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
Household Type
0
1
2
0.0
5.0
Temperature (C°)

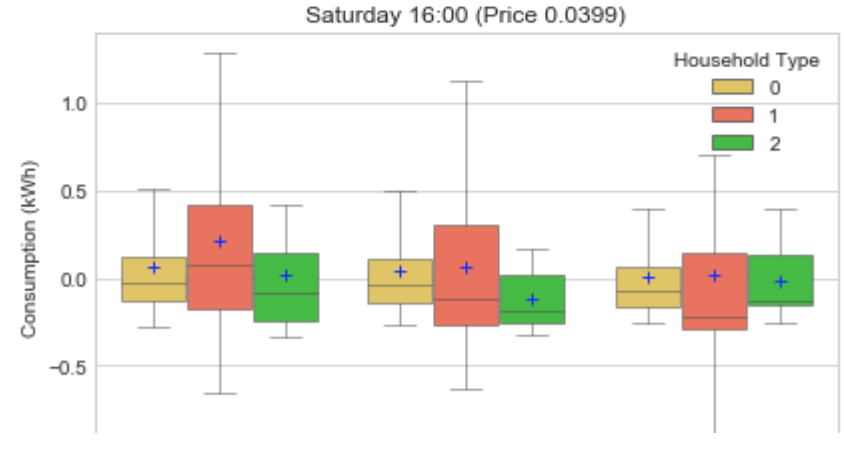

Saturday 13:00 (Price 0.672)
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
Household Type
0
1
2
5.0
Temperature (C°)

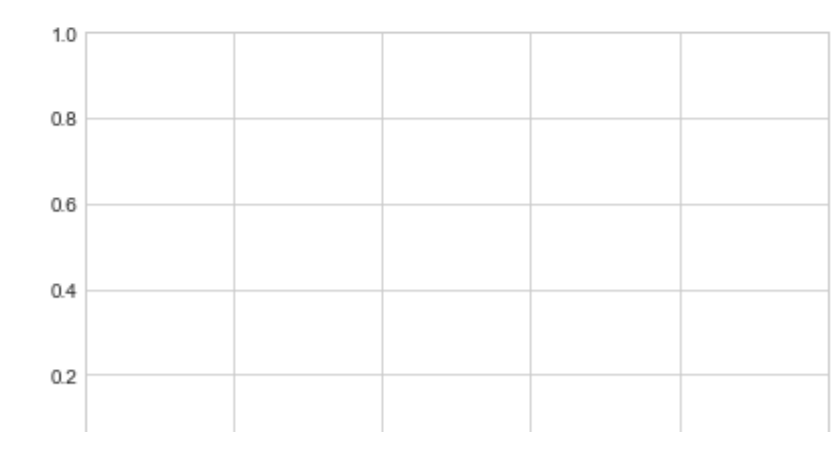

Saturday 14:00 (Price 0.0399)
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
Household Type
0
1
2
1.0
5.0
6.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Saturday 15:00 (Price 0.0399)
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
Household Type
0
1
2
1.0
5.0
6.0
Temperature (C°)

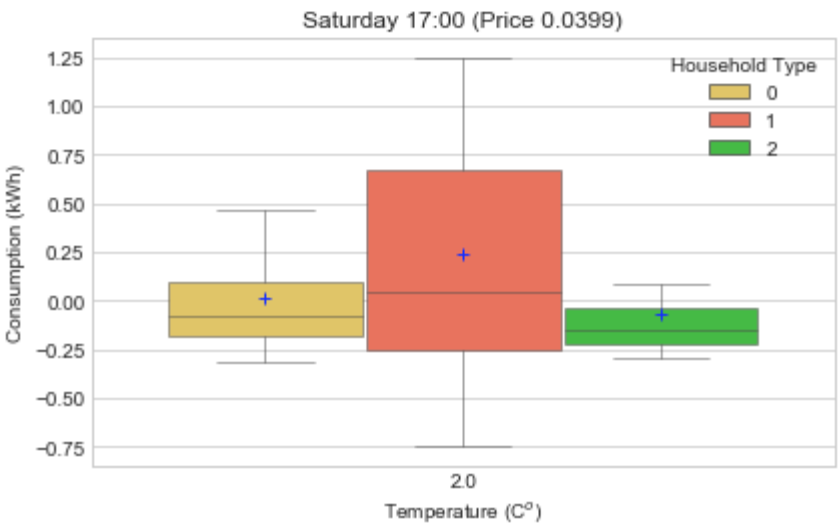

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

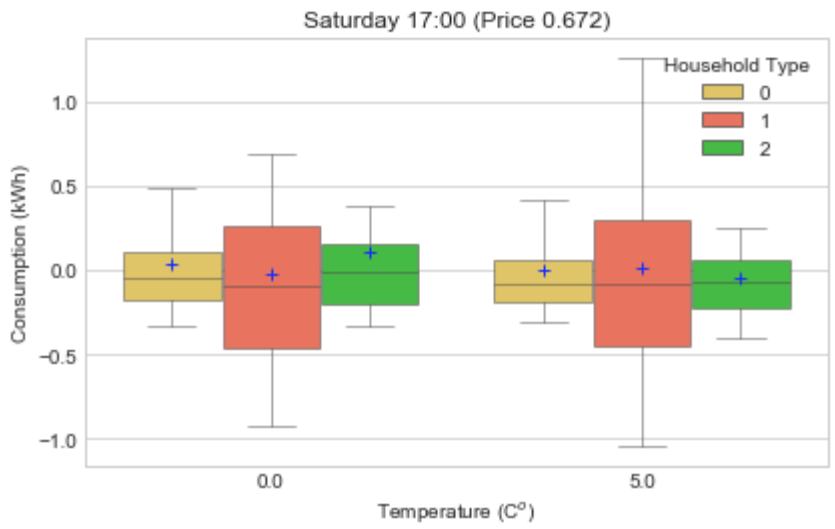

Saturday 16:00 (Price 0.0399)
Consumption (kWh)
1.0
0.5
0.0
−0.5
Household Type
0
1
2

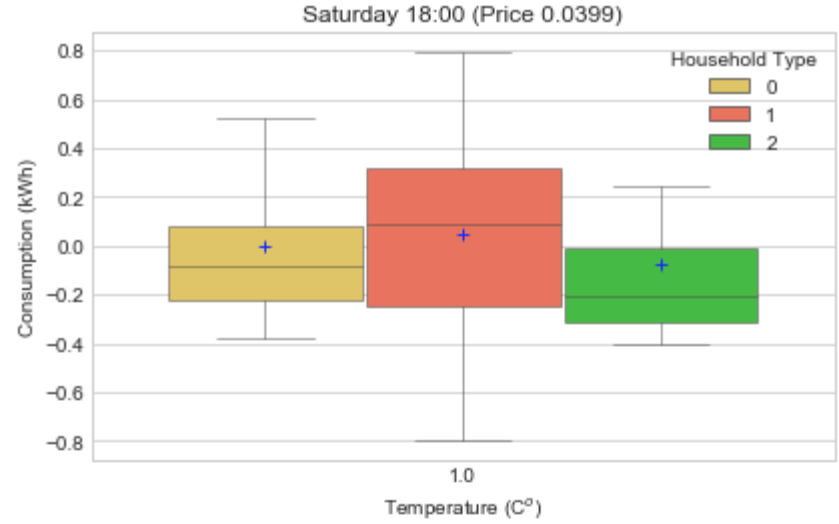

1.0
0.8
0.6
0.4
0.2

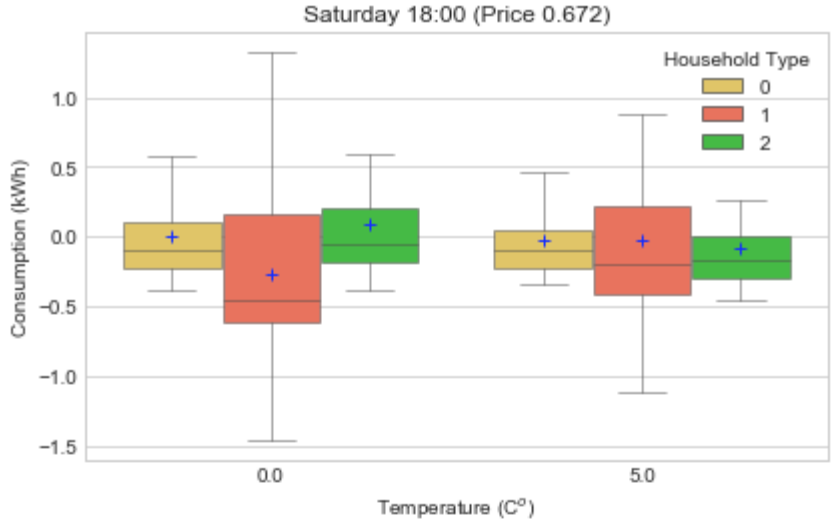

−1.0
1.0
4.0
6.0
Temperature (C°)

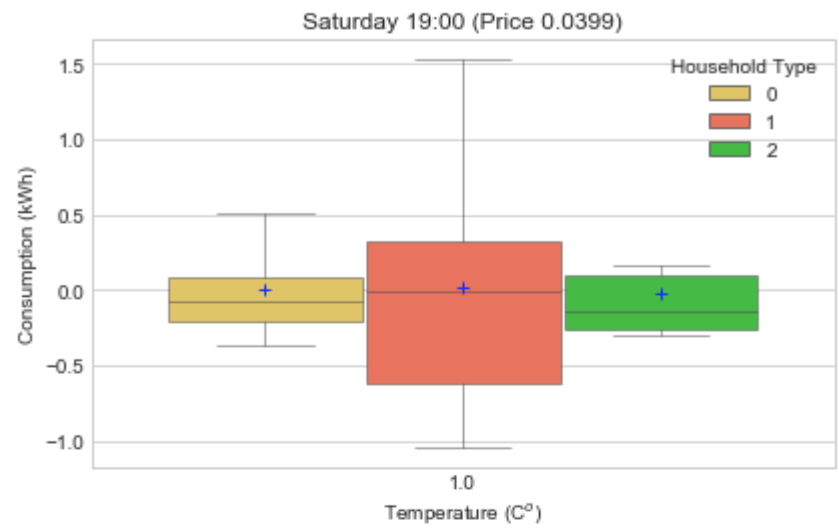

0.0
0.0
0.2
0.4
0.6
0.8
1.0

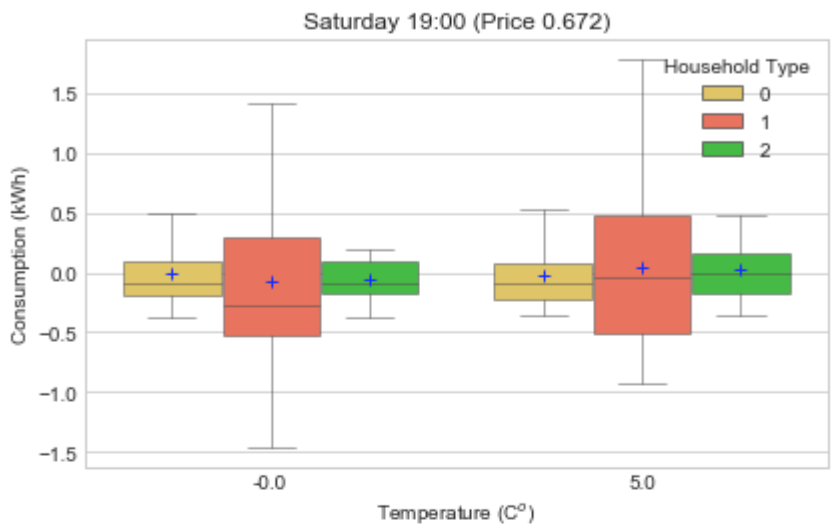

Saturday 17:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
2.0
Temperature (C°)

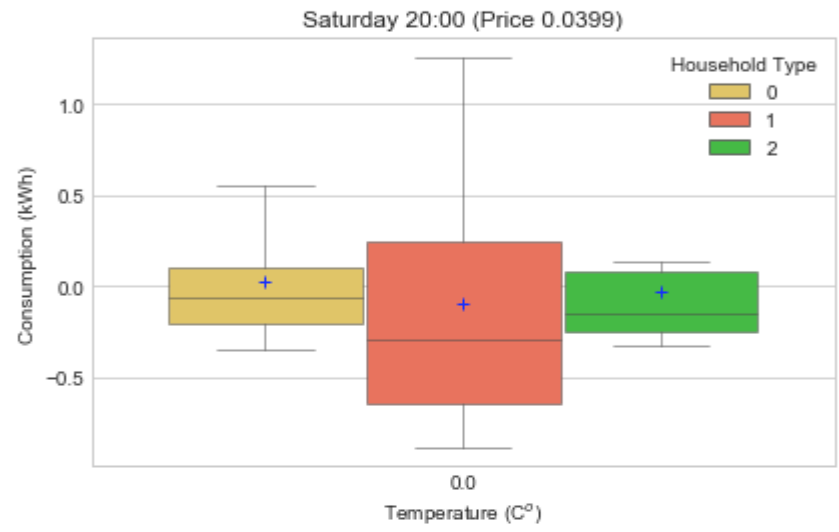

Saturday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
0.0
5.0
Temperature (C°)

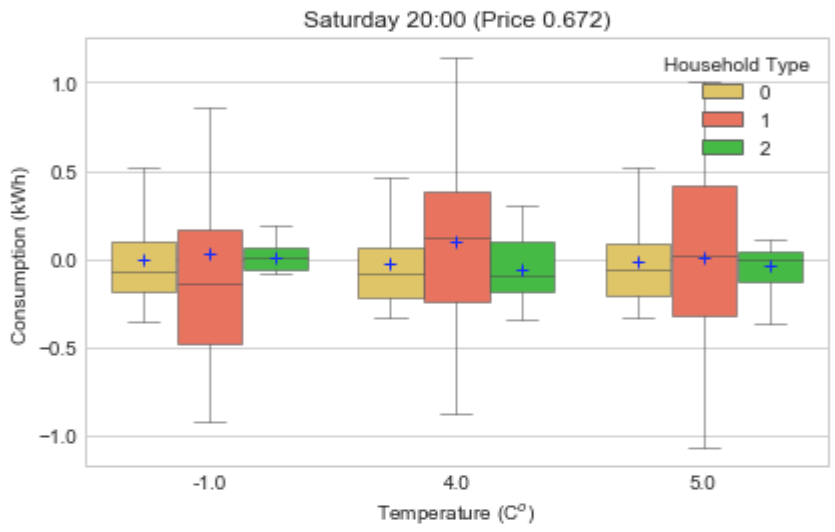

Saturday 18:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
1.0
Temperature (C°)

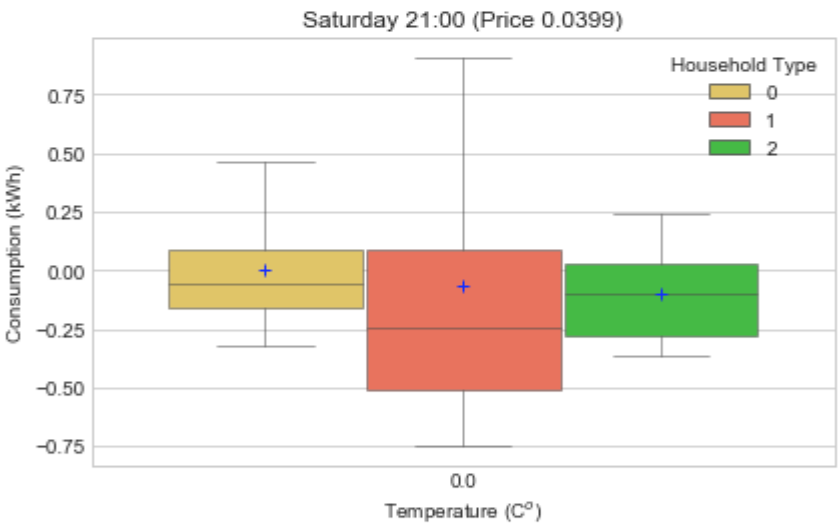

Saturday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
5.0
Temperature (C°)

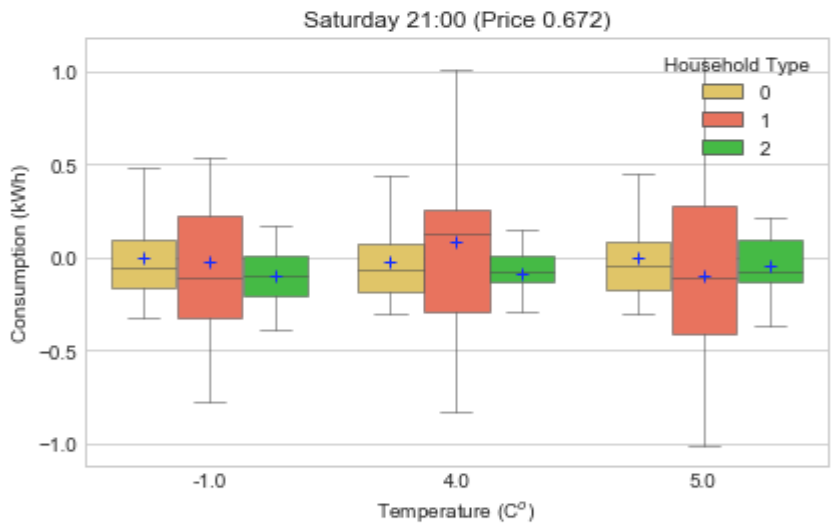

Saturday 19:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
1.0
Temperature (C°)


Saturday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
-0.0
5.0
Temperature (C°)


Saturday 20:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
0.0
Temperature (C°)

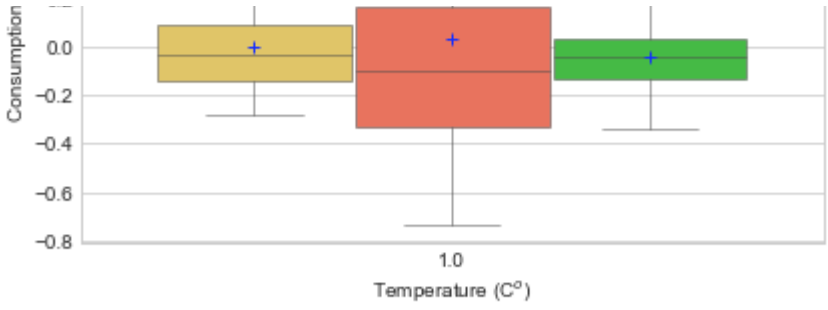

Saturday 20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
4.0
5.0
Temperature (C°)

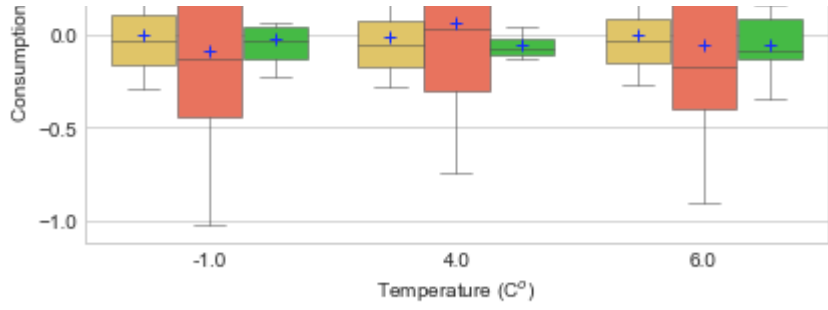

Saturday 21:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0.0
Temperature (C°)

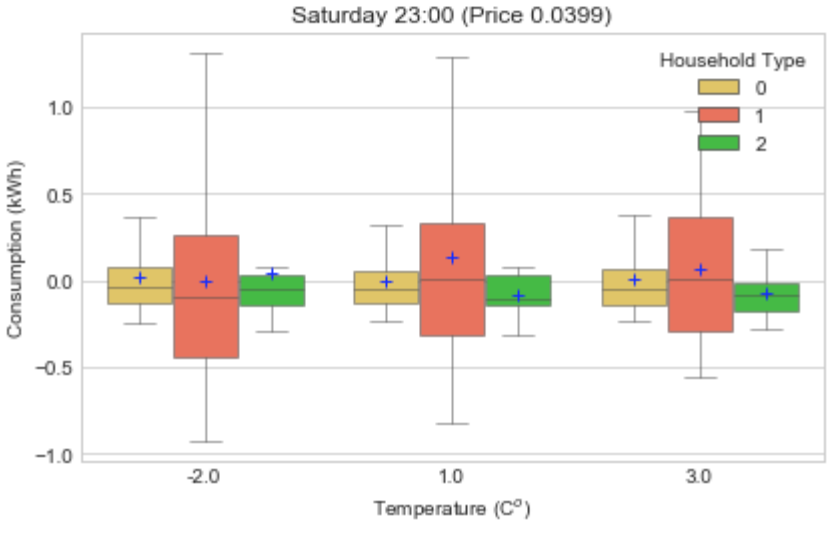

Saturday 21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
4.0
5.0
Temperature (C°)

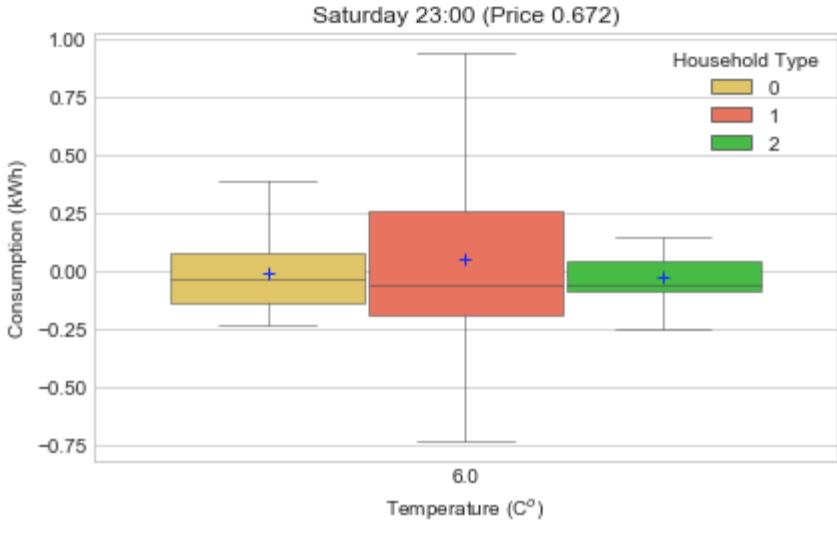
Saturday 22:00 (Price 0.0399)
Household Type
0
1
2
0.8
0.6
0.4
0.2

Saturday 22:00 (Price 0.672)
Household Type
0
1
2
1.0
0.5

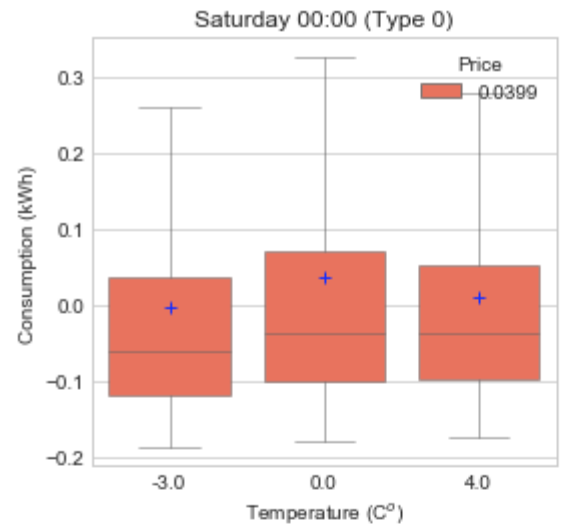

Consumption
0.0
−0.2
−0.4
−0.6
−0.8
1.0
Temperature (C°)

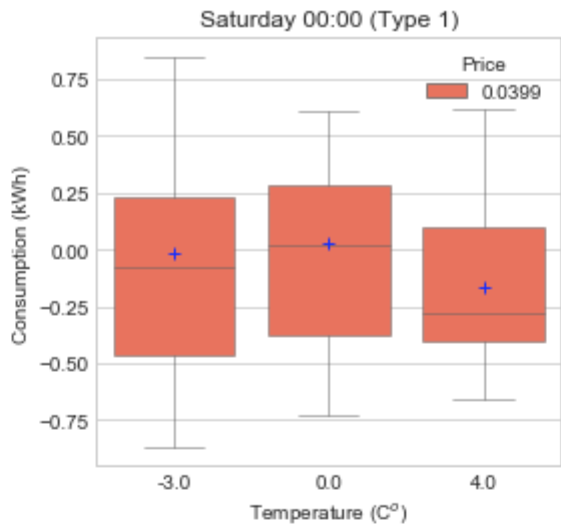

Consumption
0.0
−0.5
−1.0
-1.0
4.0
6.0
Temperature (C°)

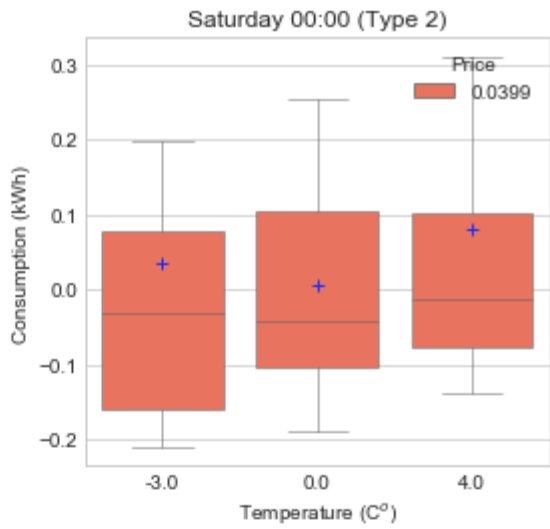

Saturday 23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-2.0
1.0
3.0
Temperature (C°)

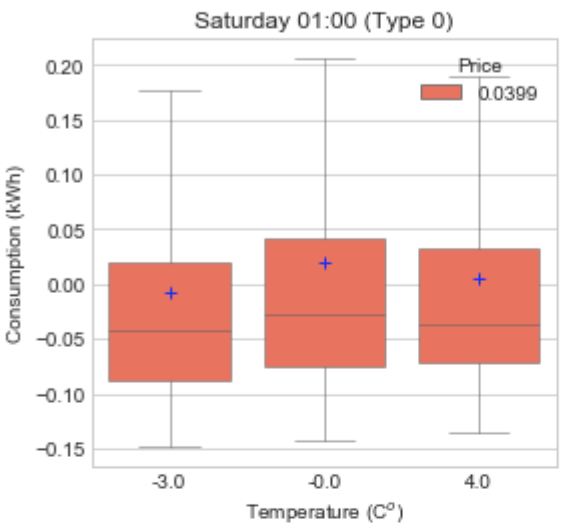

Saturday 23:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
6.0
Temperature (C°)

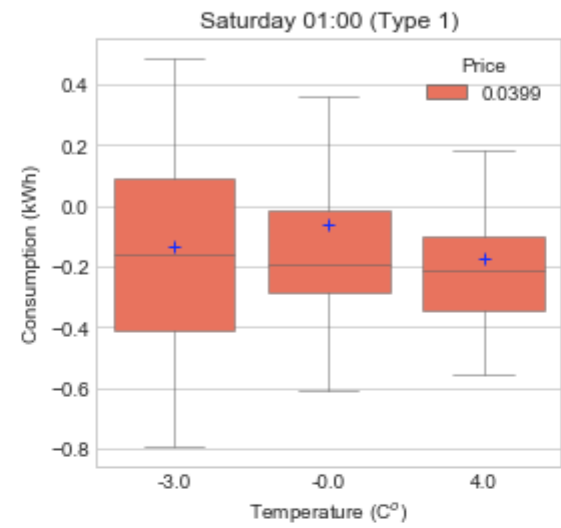
In [133]:
```
# price comparison
# Saturday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 5 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

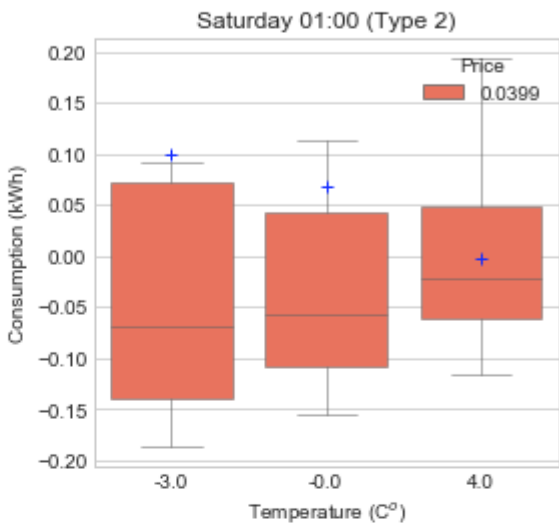

Saturday 00:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
-3.0
0.0
4.0
Temperature (C°)

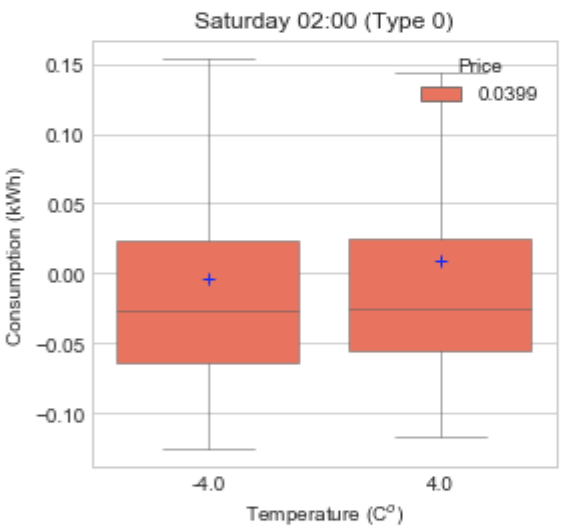

Saturday 00:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-3.0
0.0
4.0
Temperature (C°)

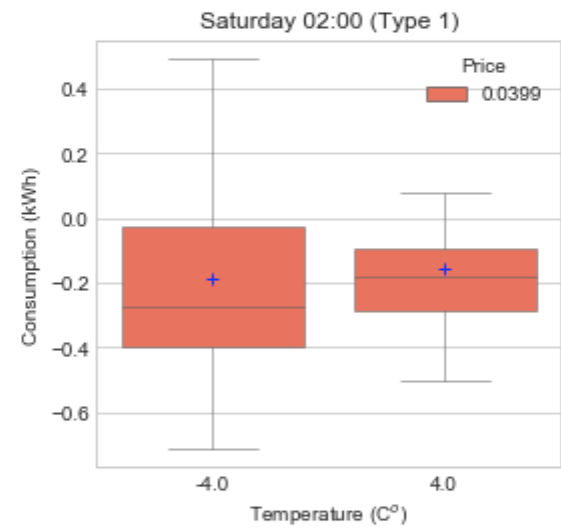

Saturday 00:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
-3.0
0.0
4.0
Temperature (C°)

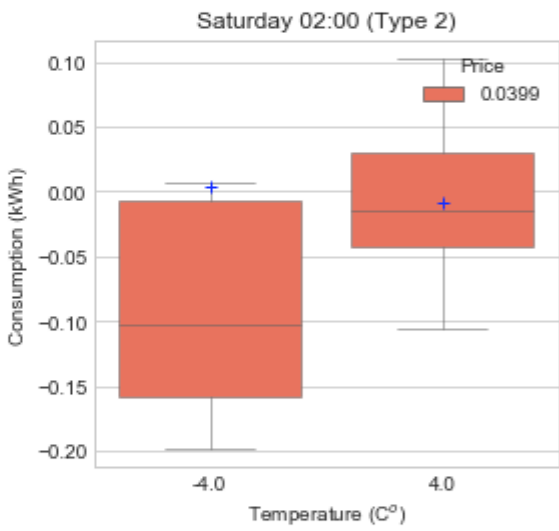

Saturday 01:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
-3.0
0.0
4.0
Temperature (C°)

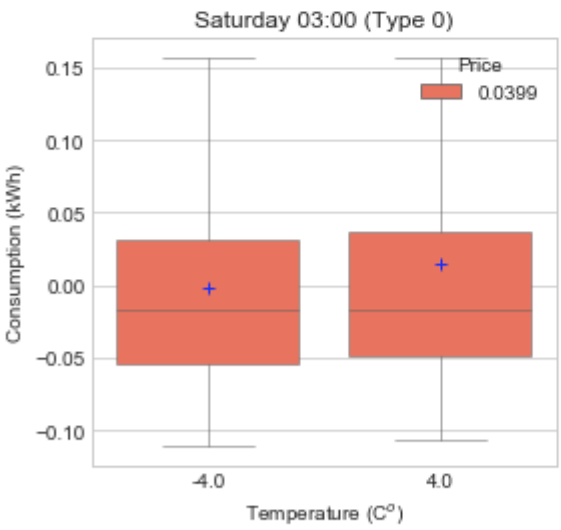

Saturday 01:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
-3.0
0.0
4.0
Temperature (C°)

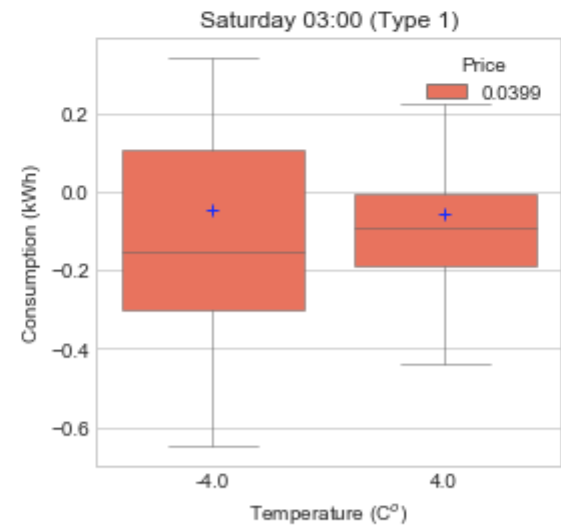

Saturday 01:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
-3.0
0.0
4.0
Temperature (C°)

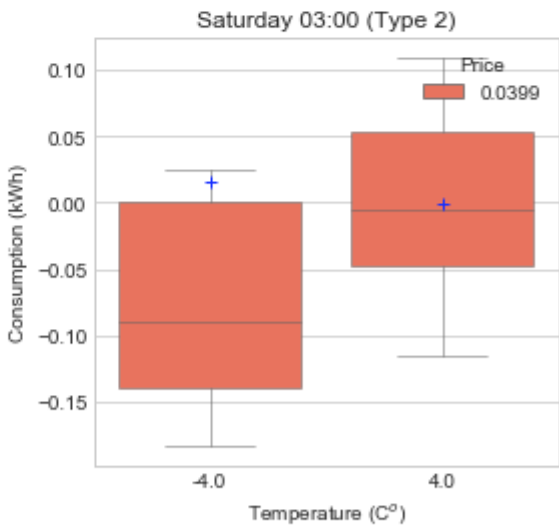

Saturday 02:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
-4.0
4.0
Temperature (C°)

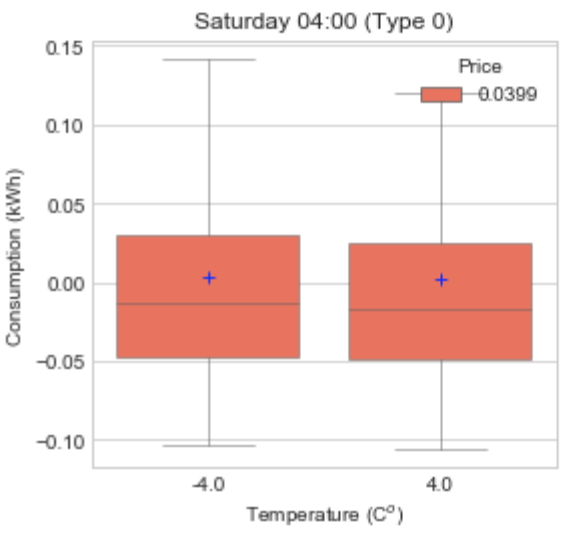

Saturday 02:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
4.0
Temperature (C°)

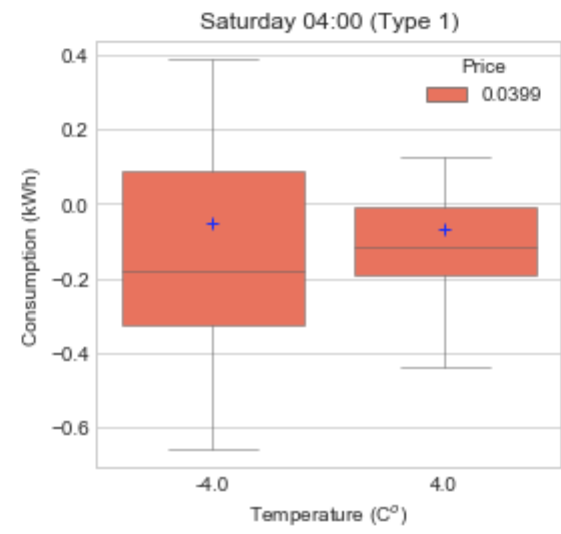

Saturday 02:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
-4.0
4.0
Temperature (C°)

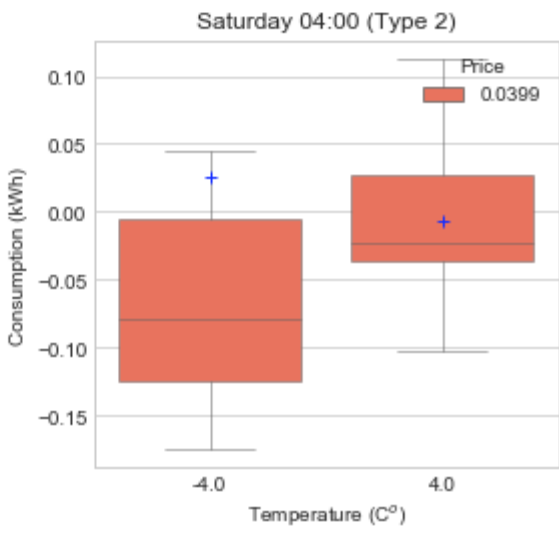

Saturday 03:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
-4.0
4.0
Temperature (C°)

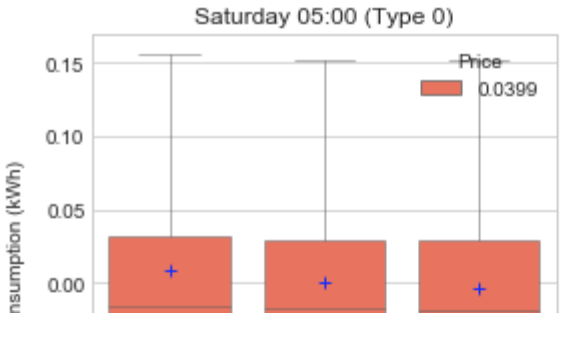

Saturday 03:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.2
0.0
−0.2
−0.4
−0.6
-4.0
4.0
Temperature (C°)

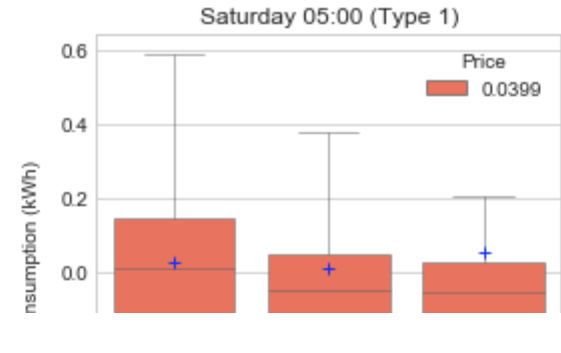

Saturday 03:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.10
0.05
0.00
−0.05
−0.10
−0.15
-4.0
4.0
Temperature (C°)

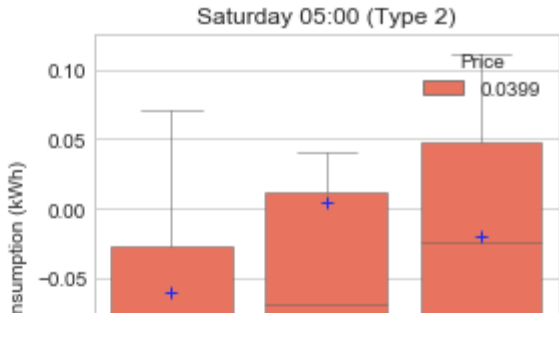

Saturday 04:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
-4.0
4.0
Temperature (C°)

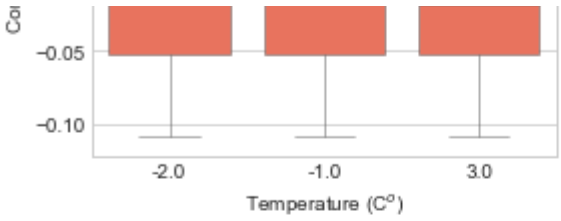

Saturday 04:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0
4.0
Temperature (C°)

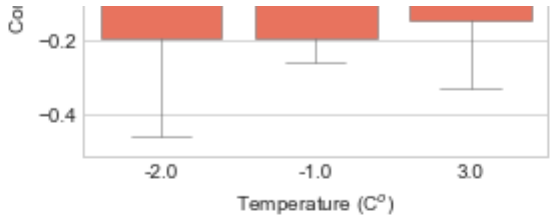

Saturday 04:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.10
0.05
0.00
−0.05
−0.10
−0.15
-4.0
4.0
Temperature (C°)

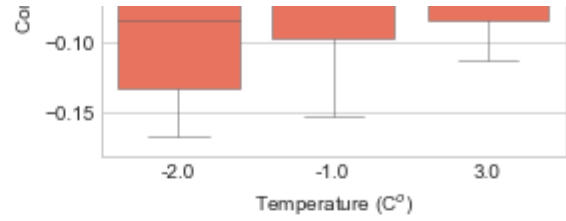

Saturday 05:00 (Type 0)
Price
0.0399
0.15
0.10
0.05
0.00

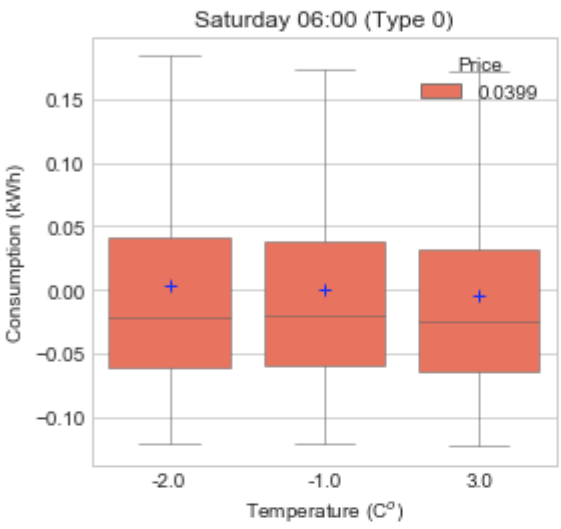

Saturday 05:00 (Type 1)
Price
0.0399
0.6
0.4
0.2
0.0

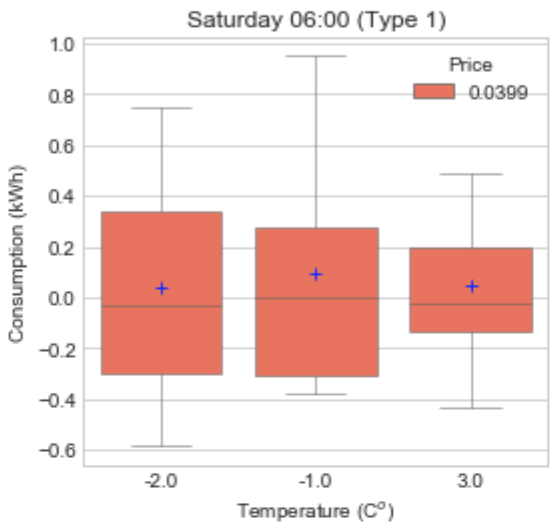

Saturday 05:00 (Type 2)
Price
0.0399
0.10
0.05
0.00
−0.05

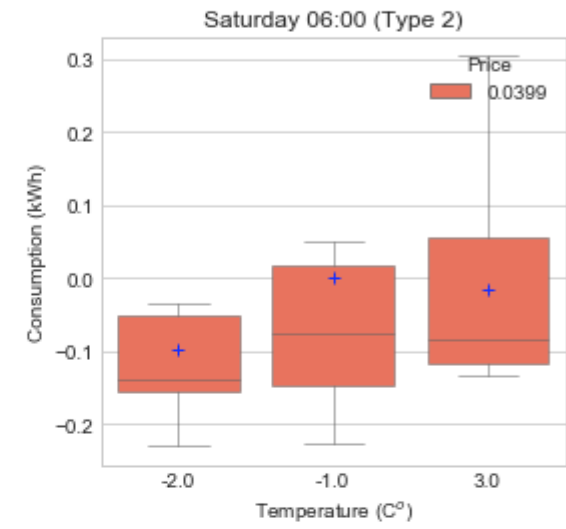

-0.05
-0.10
-2.0
-1.0
3.0
Temperature (C°)

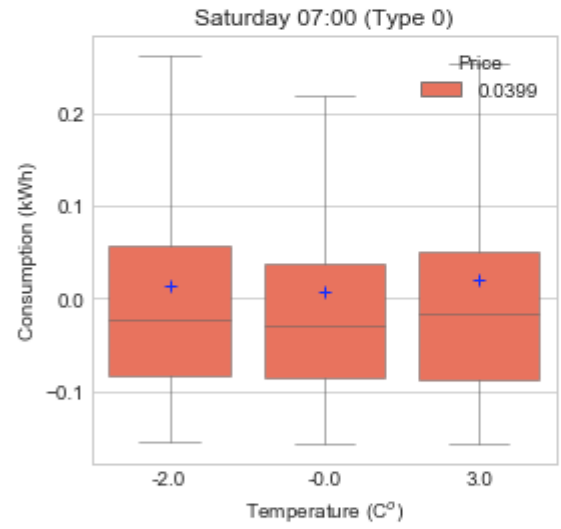

-0.2
-0.4
-2.0
-1.0
3.0
Temperature (C°)

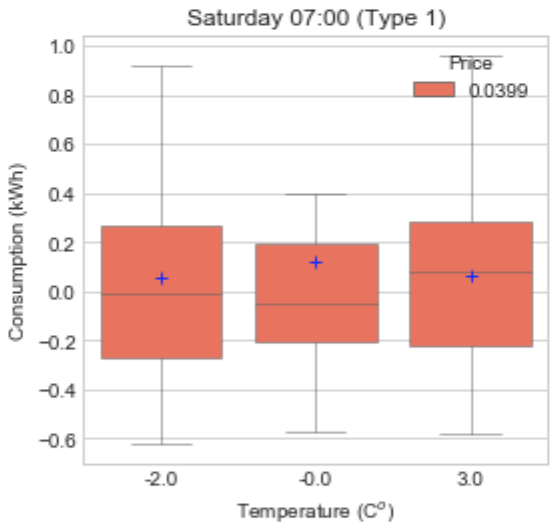

-0.10
-0.15
-2.0
-1.0
3.0
Temperature (C°)

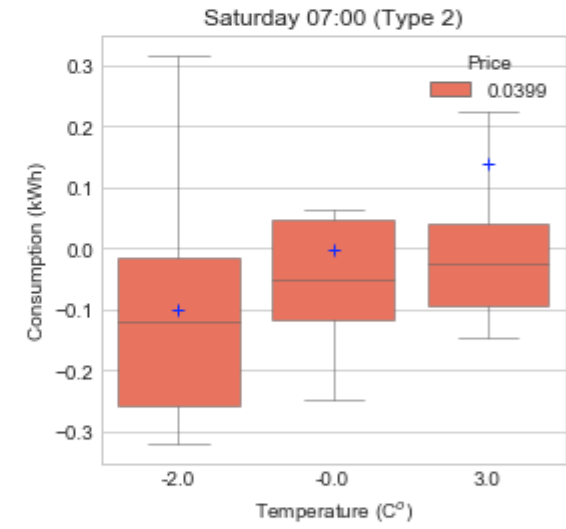

Saturday 06:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.15
0.10
0.05
0.00
−0.05
−0.10
-2.0
-1.0
3.0
Temperature (C°)

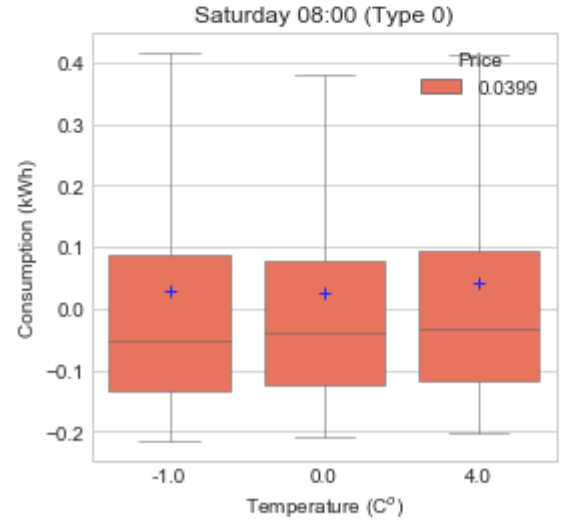

Saturday 06:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
-1.0
3.0
Temperature (C°)

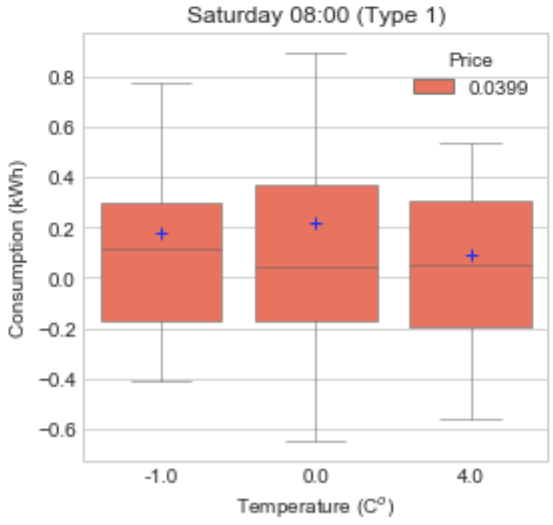

Saturday 06:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
-2.0
-1.0
3.0
Temperature (C°)

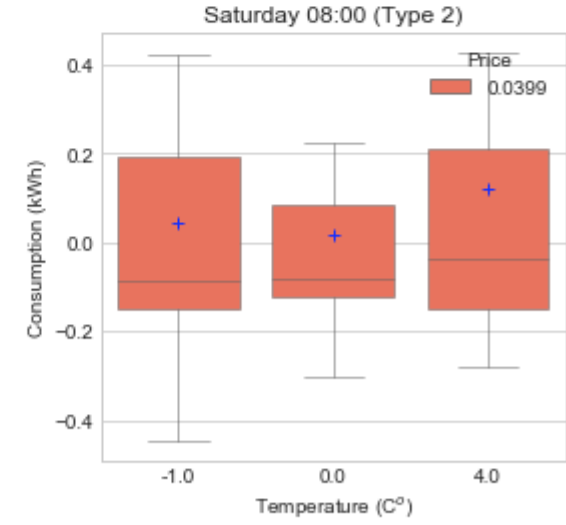

Saturday 07:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.2
0.1
0.0
−0.1
-2.0
-0.0
3.0
Temperature (C°)

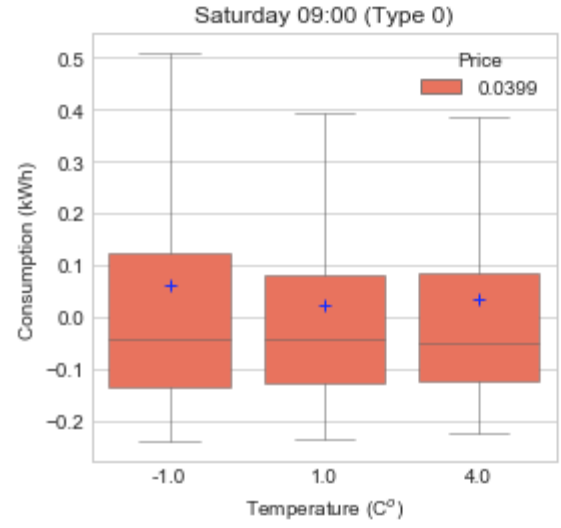

Saturday 07:00 (Type 1)
Price
0.0399
Consumption (kWh)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
-0.0
3.0
Temperature (C°)

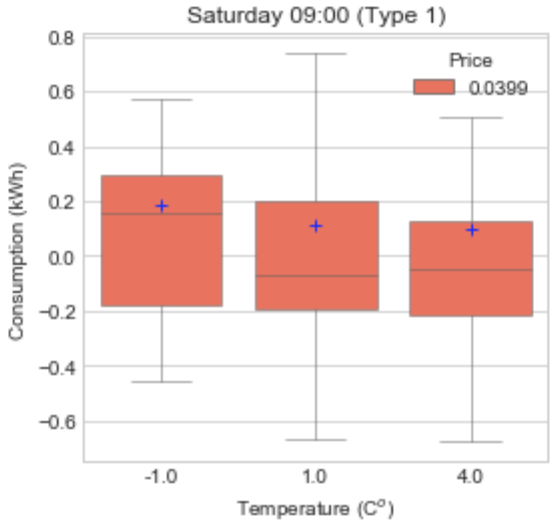

Saturday 07:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-2.0
-0.0
3.0
Temperature (C°)

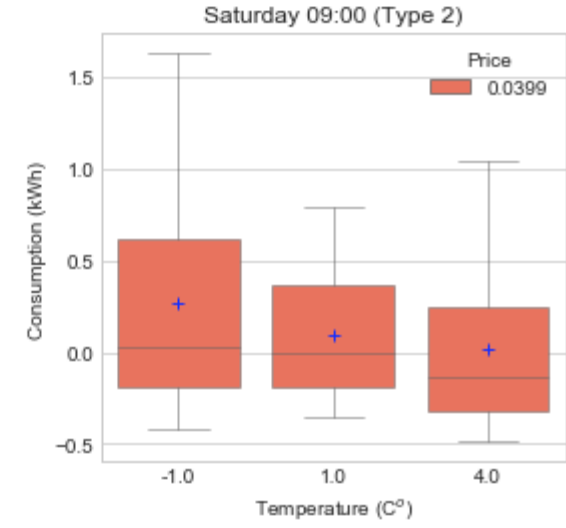

Saturday 08:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-1.0
0.0
4.0
Temperature (C°)

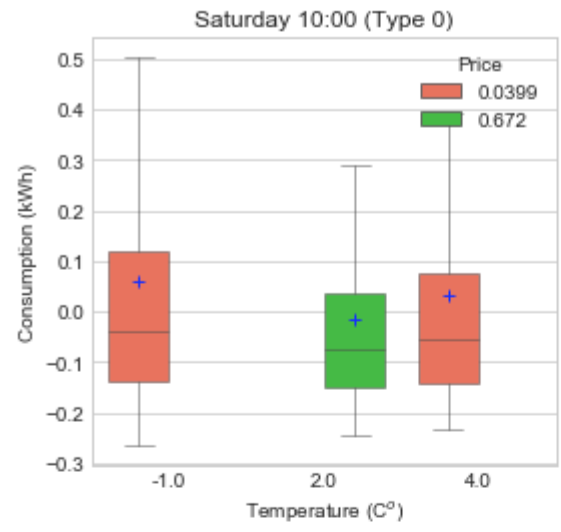

Saturday 08:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-1.0
0.0
4.0
Temperature (C°)

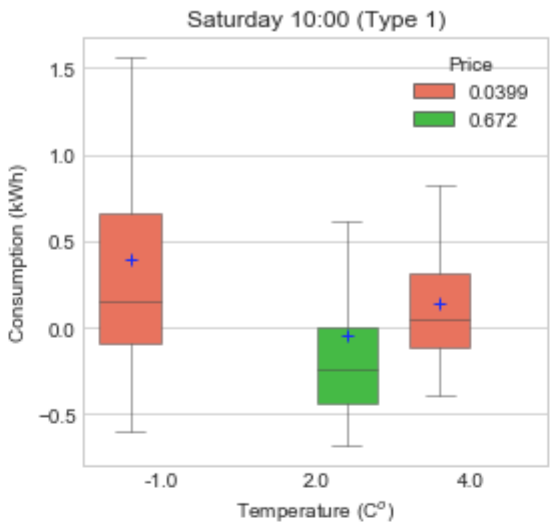

Saturday 08:00 (Type 2)
Price
0.0399
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
-1.0
0.0
4.0
Temperature (C°)

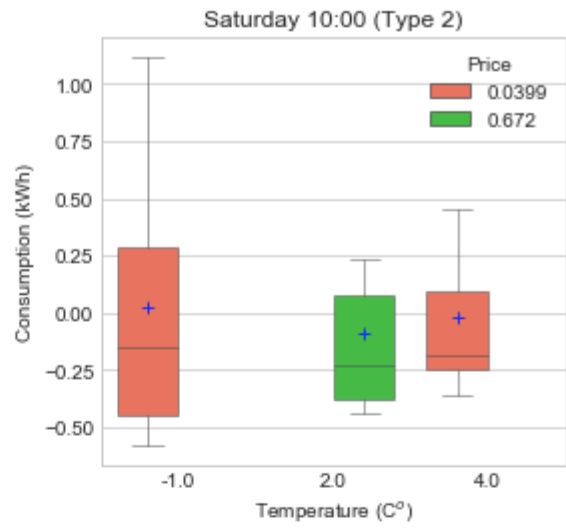

Saturday 09:00 (Type 0)
Price
0.0399
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-1.0
1.0
4.0
Temperature (C°)


Saturday 09:00 (Type 1)
Price
0.0399
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-1.0
1.0
4.0
Temperature (C°)

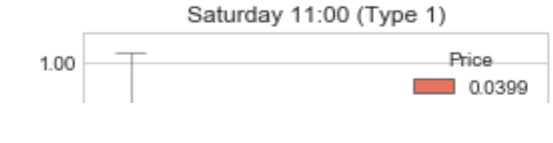

Saturday 09:00 (Type 2)
Price
0.0399
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-1.0
1.0
4.0
Temperature (C°)


Saturday 10:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-1.0
2.0
4.0
Temperature (C°)

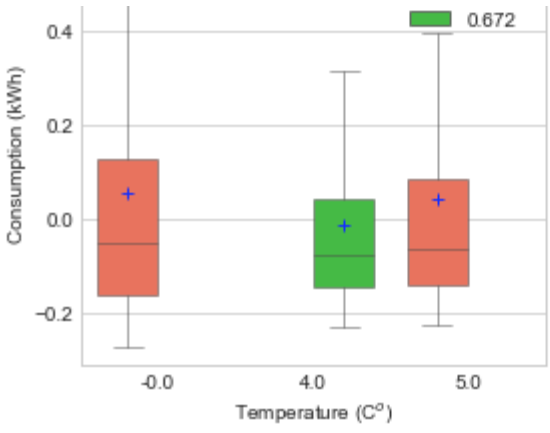

Saturday 10:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-1.0
2.0
4.0
Temperature (C°)

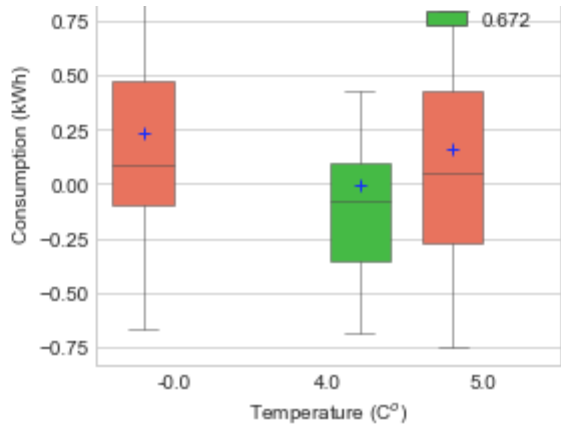

Saturday 10:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
-1.0
2.0
4.0
Temperature (C°)

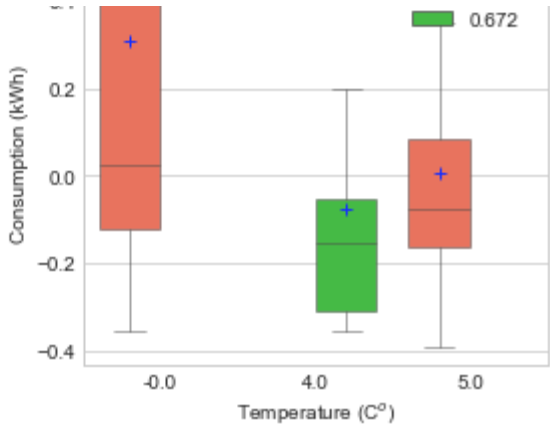

Saturday 11:00 (Type 0)
Price
0.0399

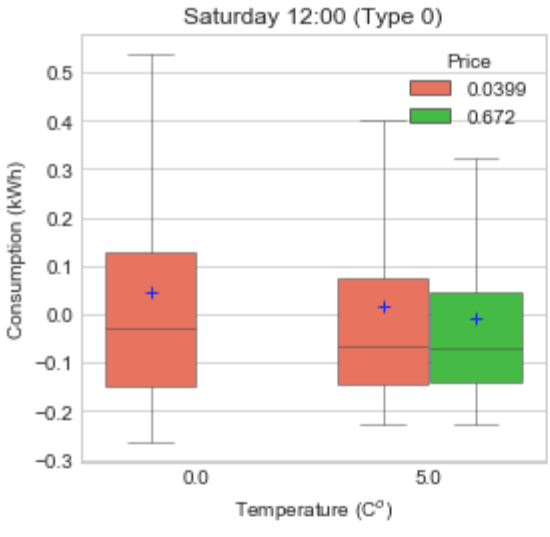

Saturday 11:00 (Type 1)
Price
0.0399
1.00

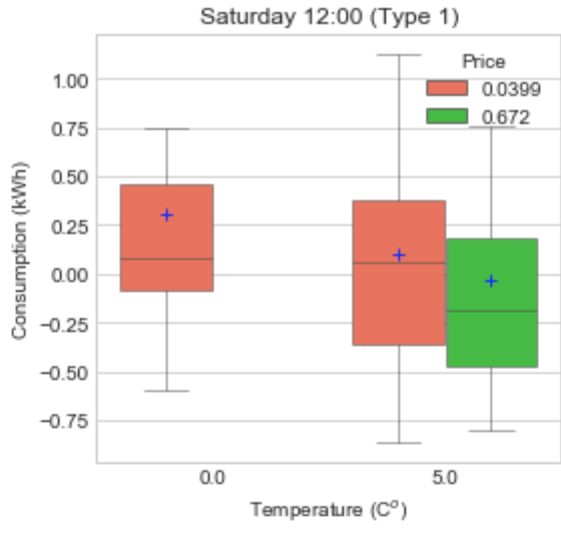

Saturday 11:00 (Type 2)
Price
0.0399
0.4

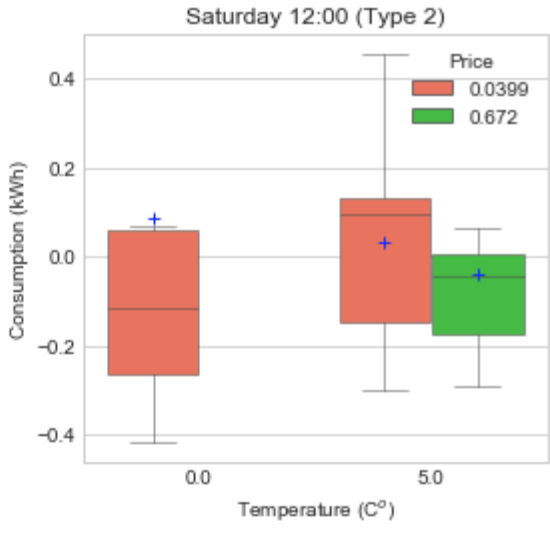

Price
0.672
Consumption (kWh)
Temperature (C°)
-0.0
4.0
5.0

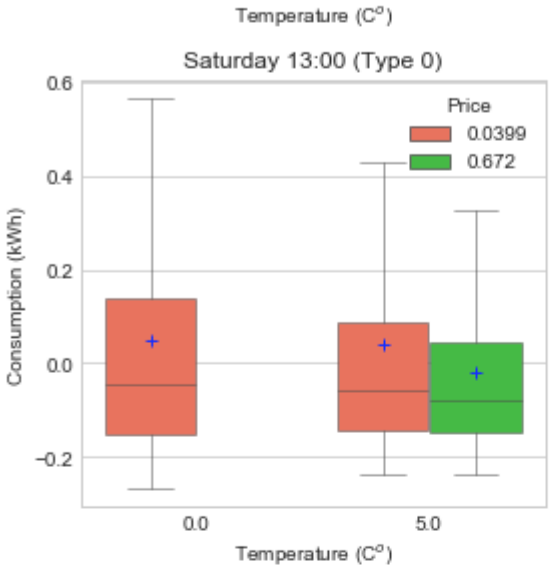

Price
0.672
Consumption (kWh)
Temperature (C°)
-0.0
4.0
5.0

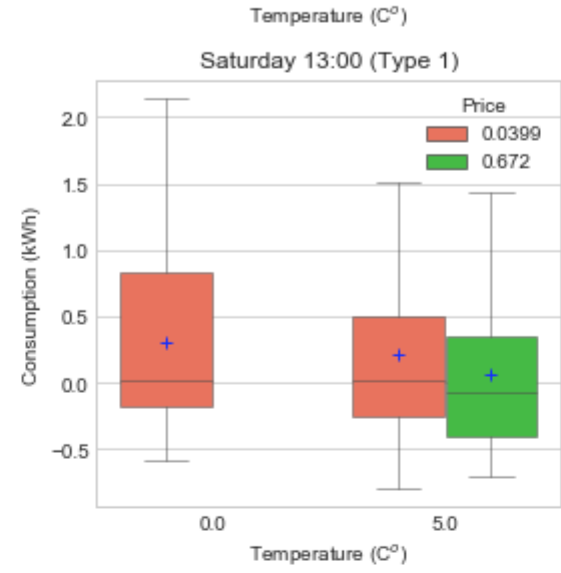

Price
0.672
Consumption (kWh)
Temperature (C°)
-0.0
4.0
5.0

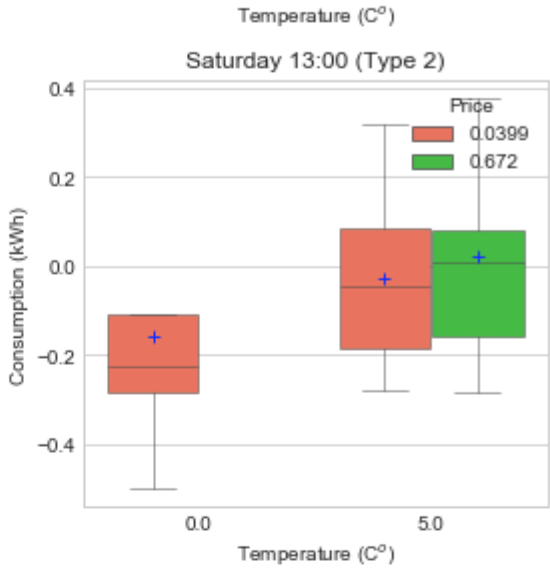

Saturday 12:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
0.0
5.0

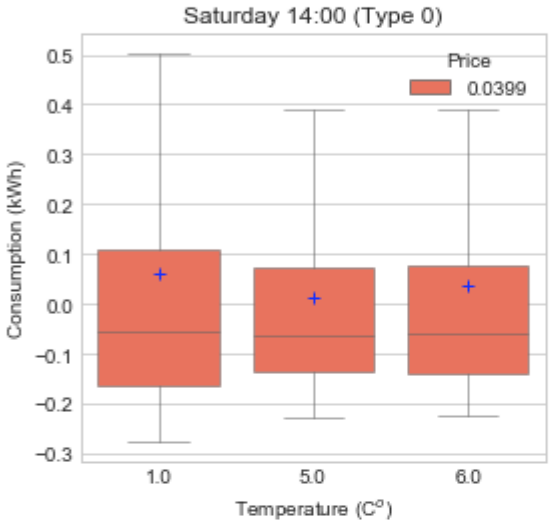

Saturday 12:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
0.0
5.0

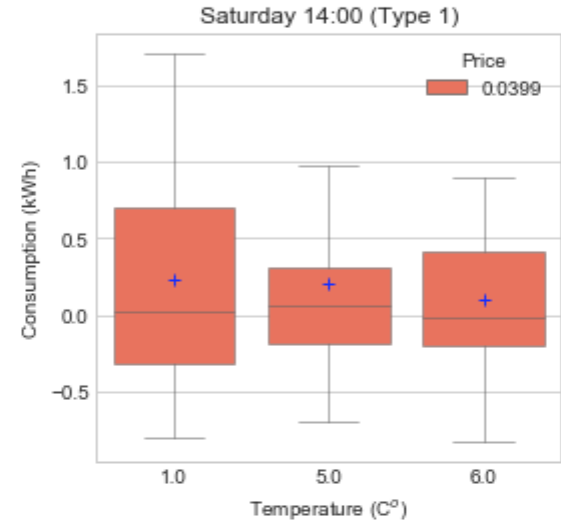

Saturday 12:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
0.0
5.0

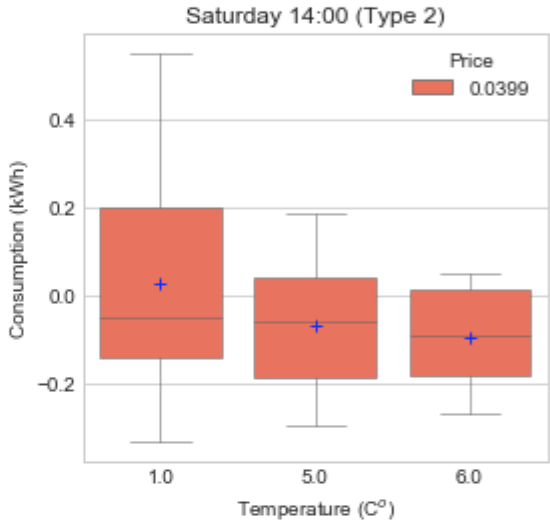

Saturday 13:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
0.0
5.0

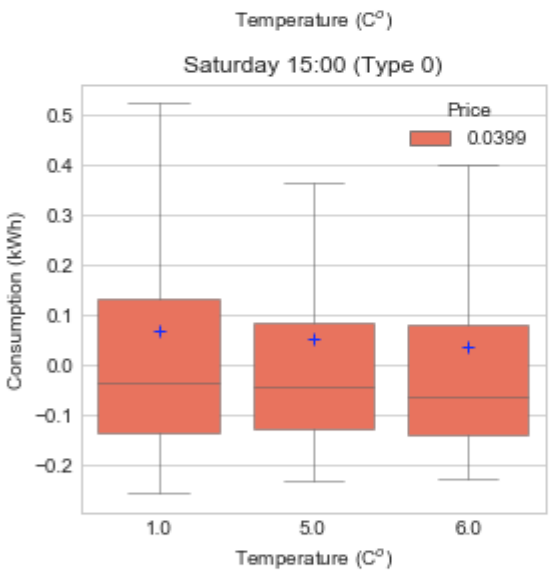

Saturday 13:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
0.0
5.0

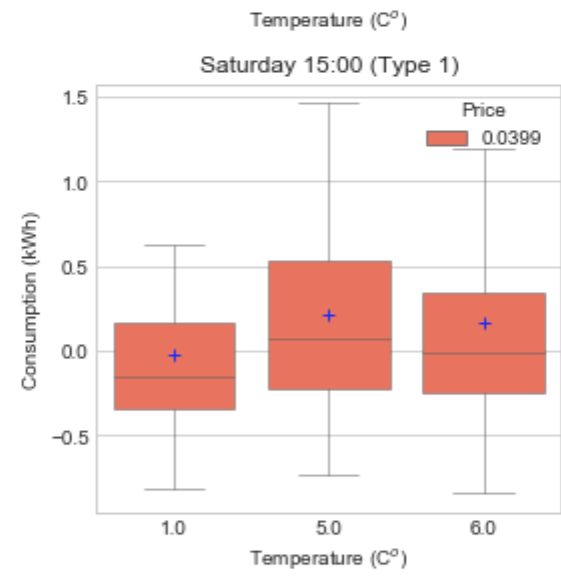

Saturday 13:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
0.0
5.0

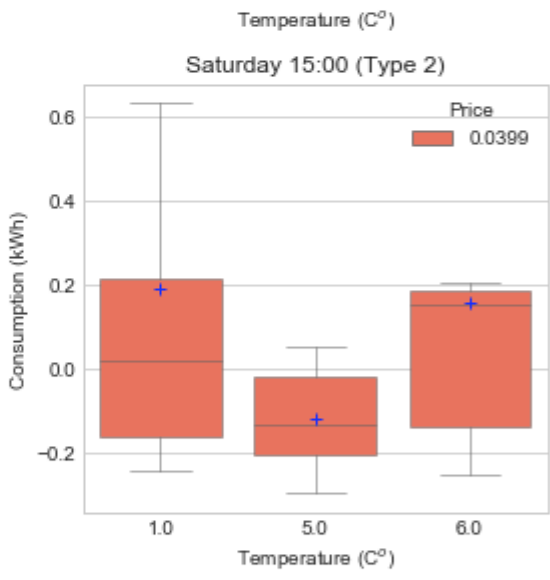

Saturday 14:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
1.0
5.0
6.0

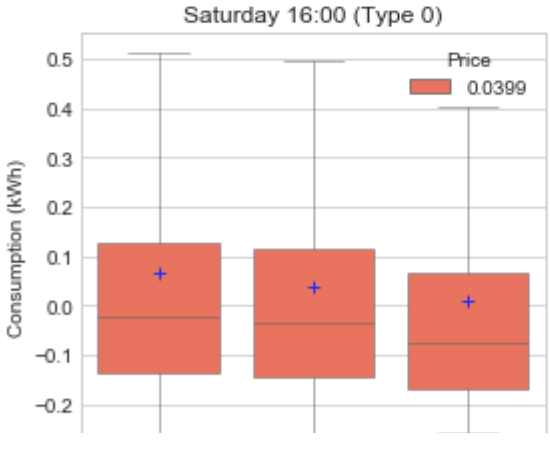

Saturday 14:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
1.0
5.0
6.0

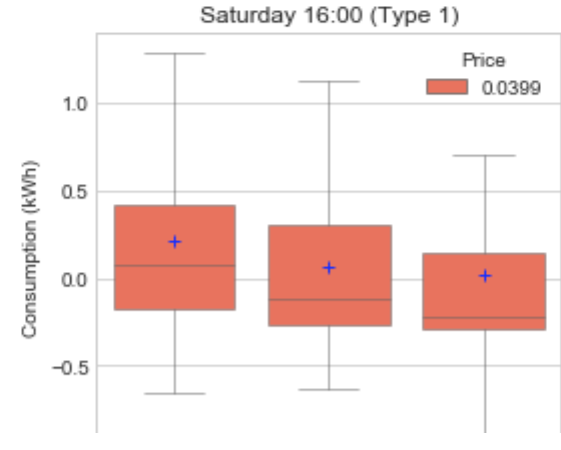

Saturday 14:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
1.0
5.0
6.0

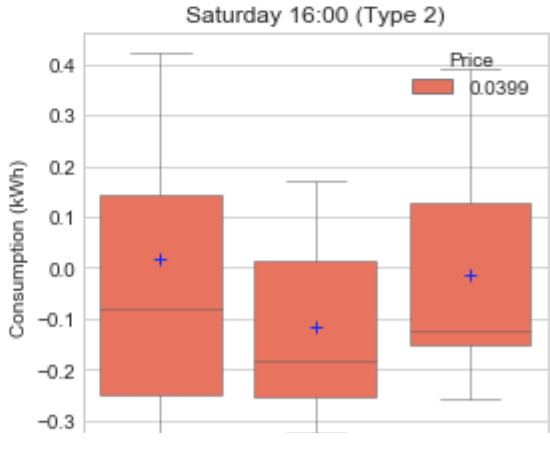

Saturday 15:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
1.0
5.0
6.0

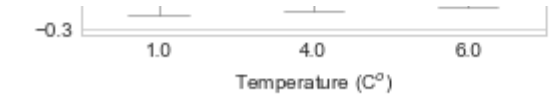

Saturday 15:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
1.0
5.0
6.0


Saturday 15:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
1.0
5.0
6.0


Saturday 16:00 (Type 0)
Price
0.0399
Consumption (kWh)

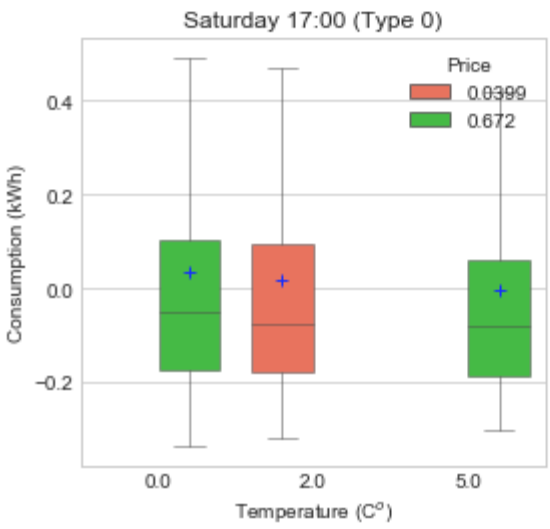

Saturday 16:00 (Type 1)
Price
0.0399
Consumption (kWh)

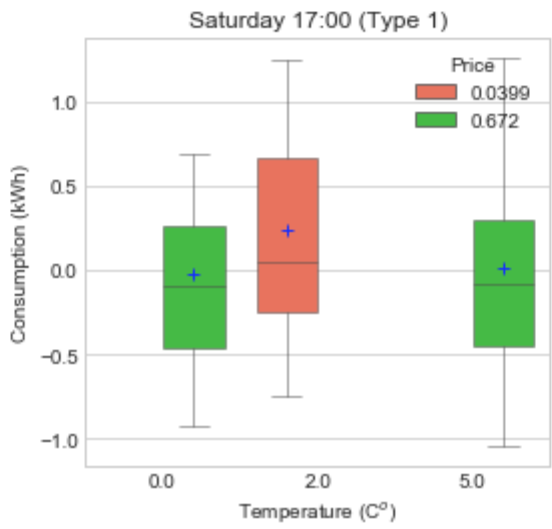

Saturday 16:00 (Type 2)
Price
0.0399
Consumption (kWh)

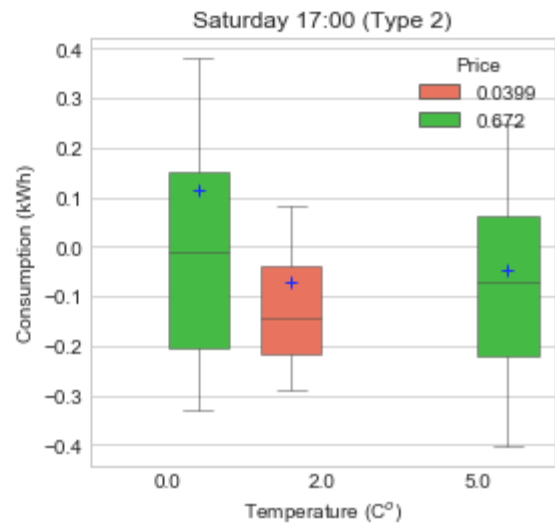

−0.3
1.0
4.0
6.0
Temperature (C°)

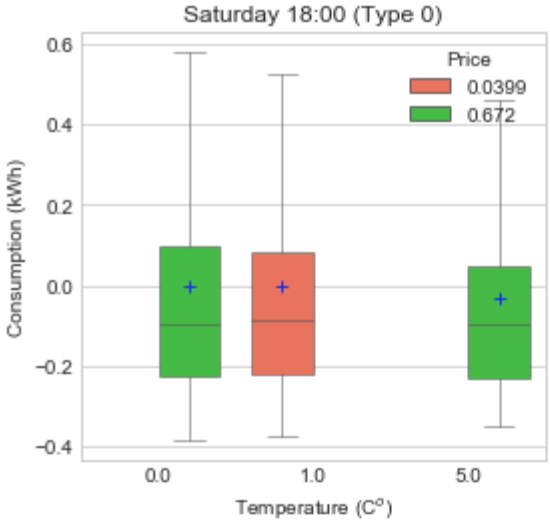

−1.0
1.0
4.0
6.0
Temperature (C°)

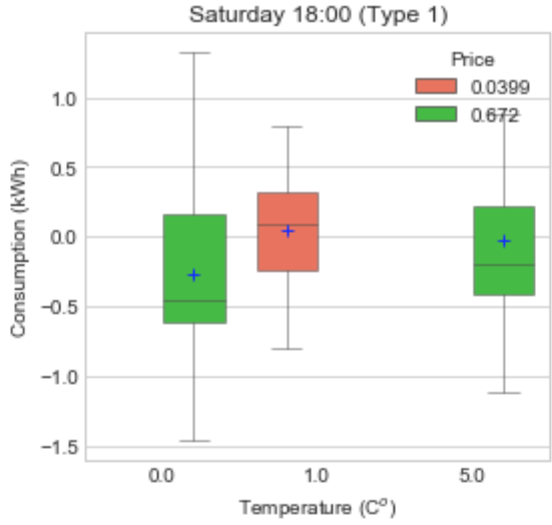

1.0
4.0
6.0
Temperature (C°)

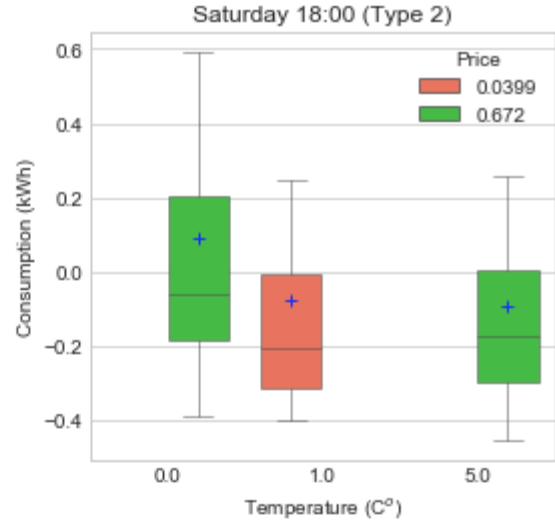

Saturday 17:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
0.0
2.0
5.0
Temperature (C°)

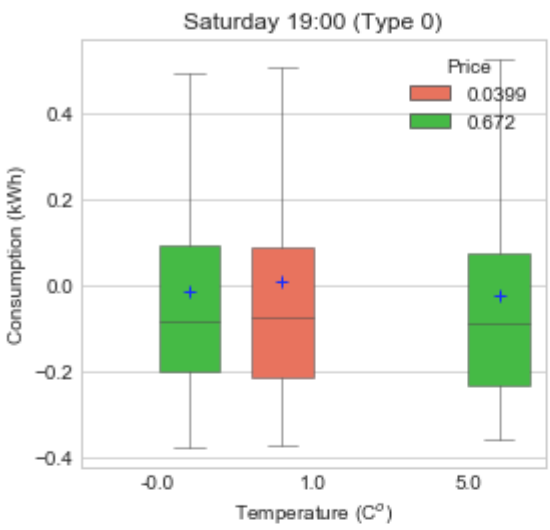

Saturday 17:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
0.0
2.0
5.0
Temperature (C°)

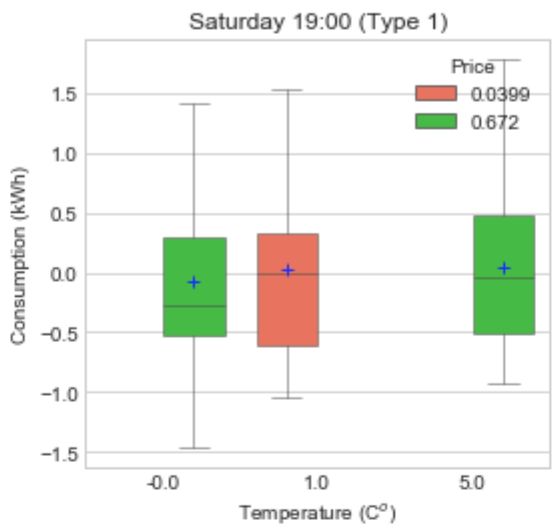

Saturday 17:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0.0
2.0
5.0
Temperature (C°)

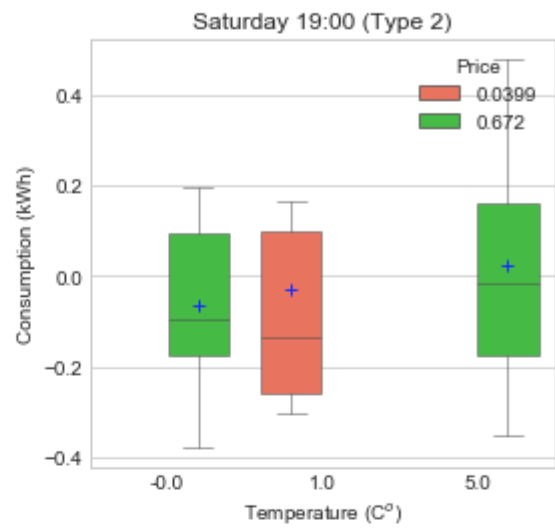

Saturday 18:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
1.0
5.0
Temperature (C°)

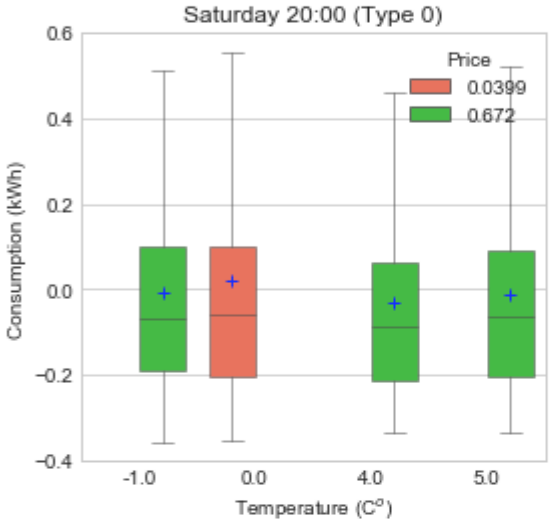

Saturday 18:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
1.0
5.0
Temperature (C°)

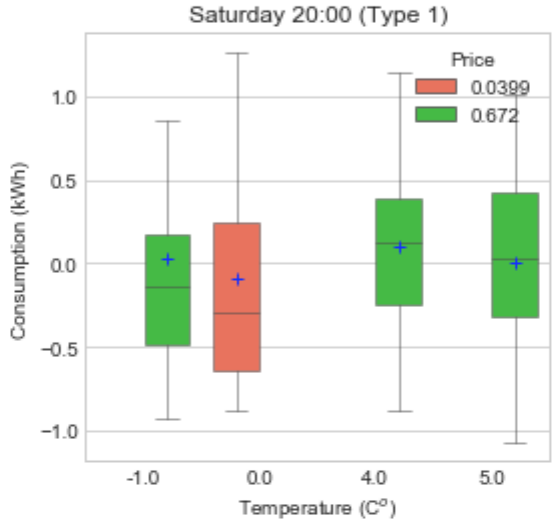

Saturday 18:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
1.0
5.0
Temperature (C°)

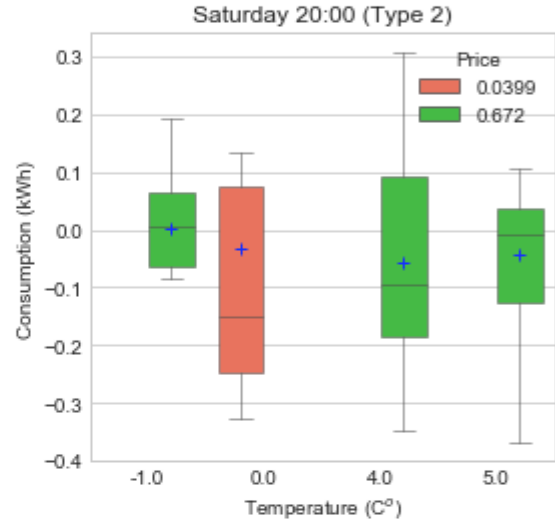

Saturday 19:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
-0.0
1.0
5.0
Temperature (C°)

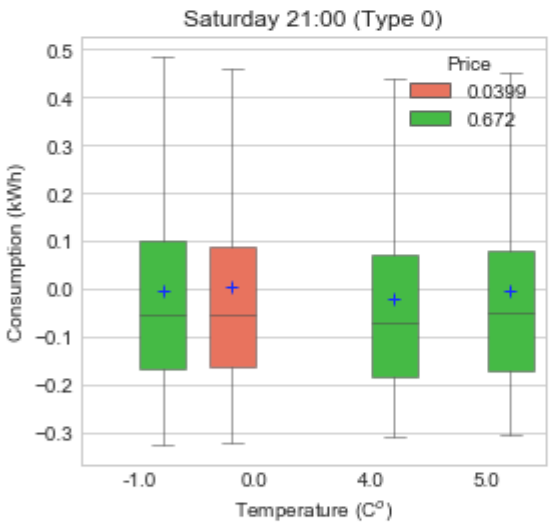

Saturday 19:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
-0.0
1.0
5.0
Temperature (C°)

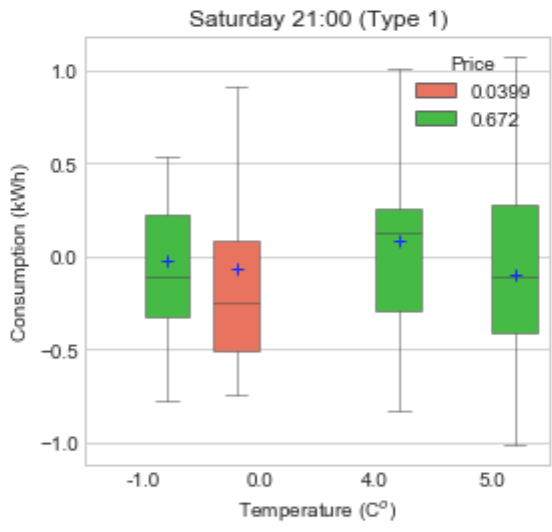

Saturday 19:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
-0.0
1.0
5.0
Temperature (C°)

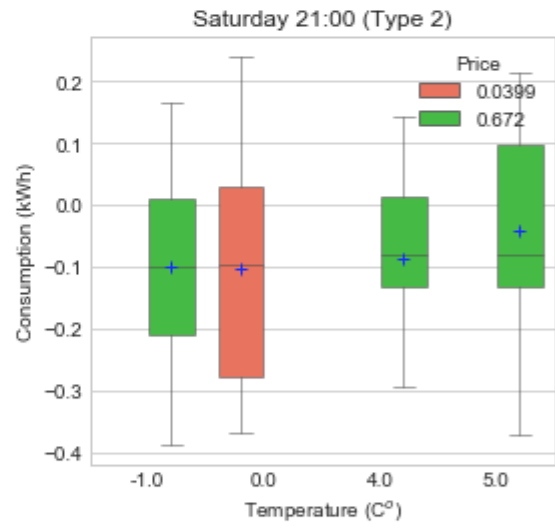

Saturday 20:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-1.0
0.0
4.0
5.0
Temperature (C°)

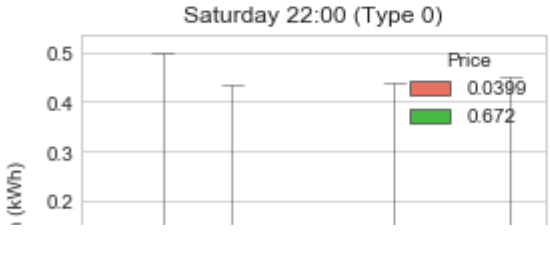

Saturday 20:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
0.0
4.0
5.0
Temperature (C°)

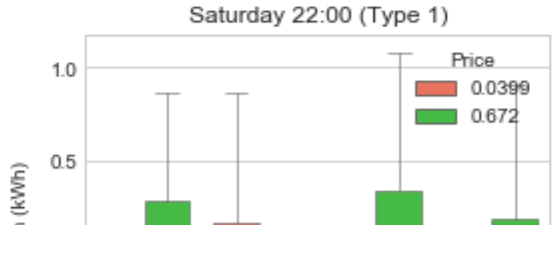

Saturday 20:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
-1.0
0.0
4.0
5.0
Temperature (C°)

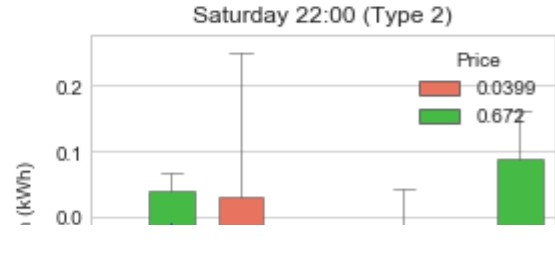

Saturday 21:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-1.0
0.0
4.0
5.0
Temperature (C°)

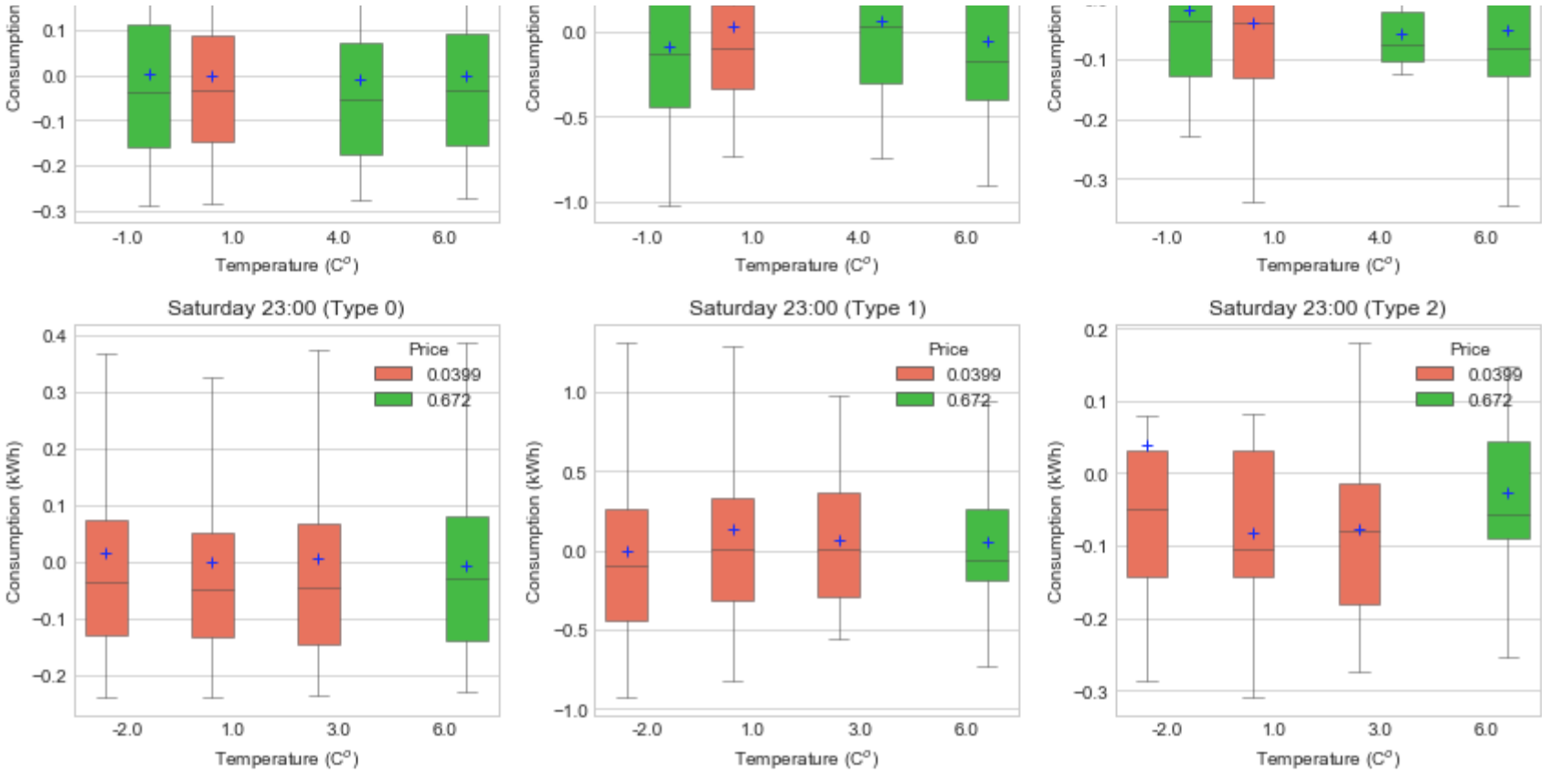

Saturday 21:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
0.0
4.0
5.0
Temperature (C°)


Saturday 21:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
-1.0
0.0
4.0
5.0
Temperature (C°)

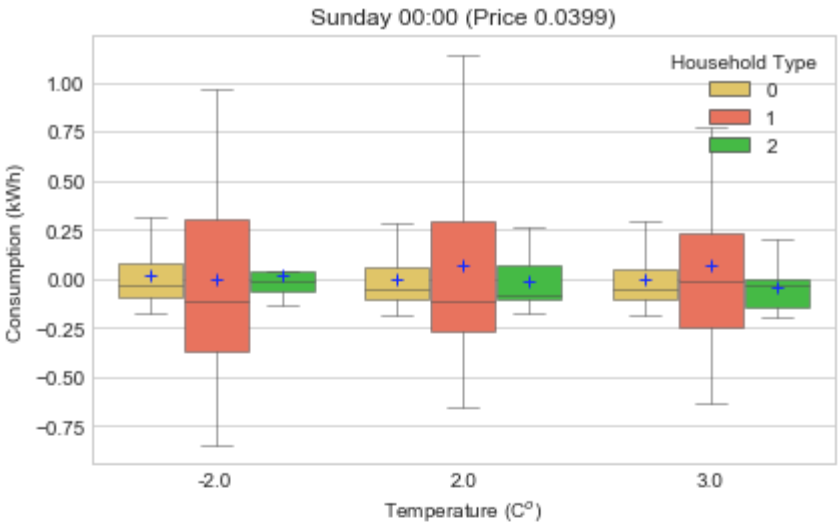

Saturday 22:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.5
0.4
0.3
0.2

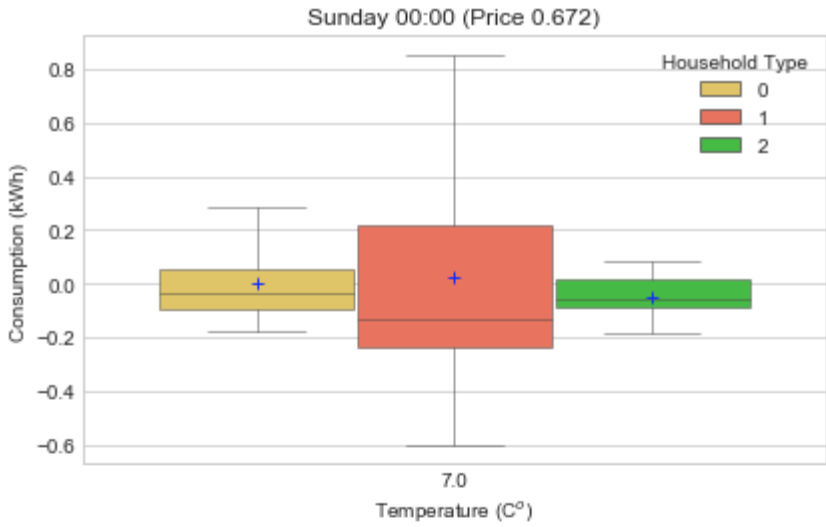

Saturday 22:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5

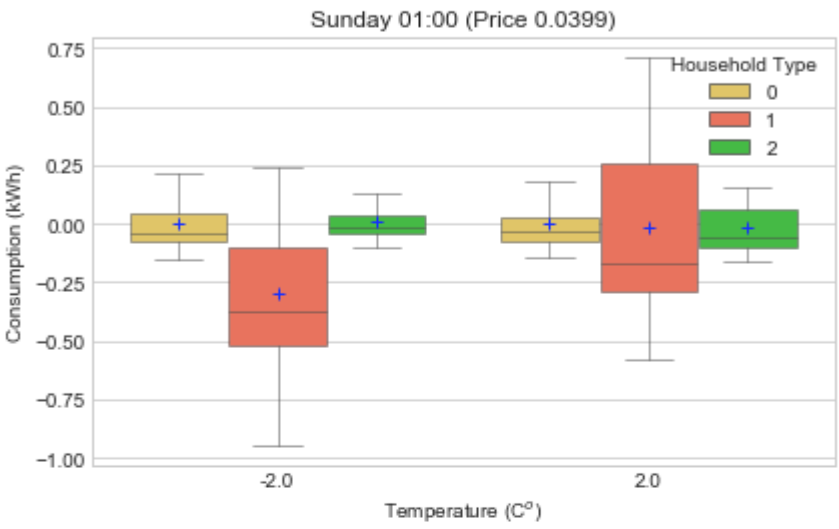

Saturday 22:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.2
0.1
0.0

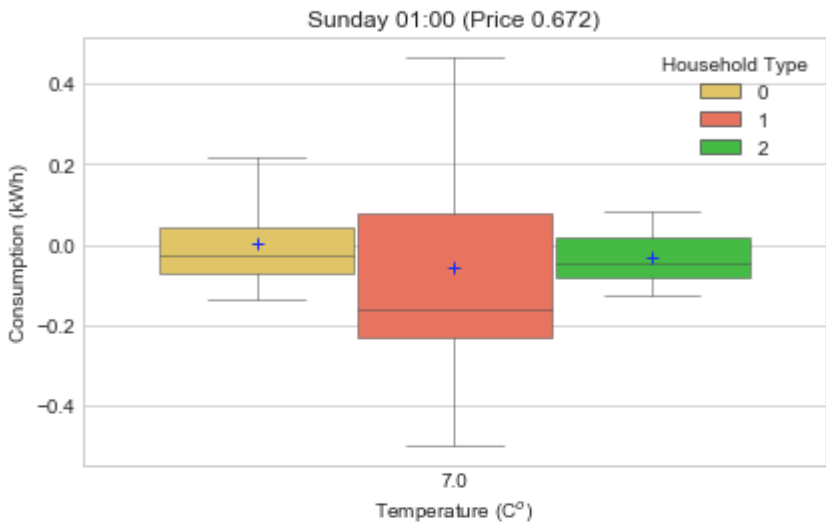

Saturday 23:00 (Type 0)
Saturday 23:00 (Type 1)
Saturday 23:00 (Type 2)
Consumption (kWh)
Temperature (C°)
Price
0.0399
0.672
-2.0
1.0
3.0
6.0

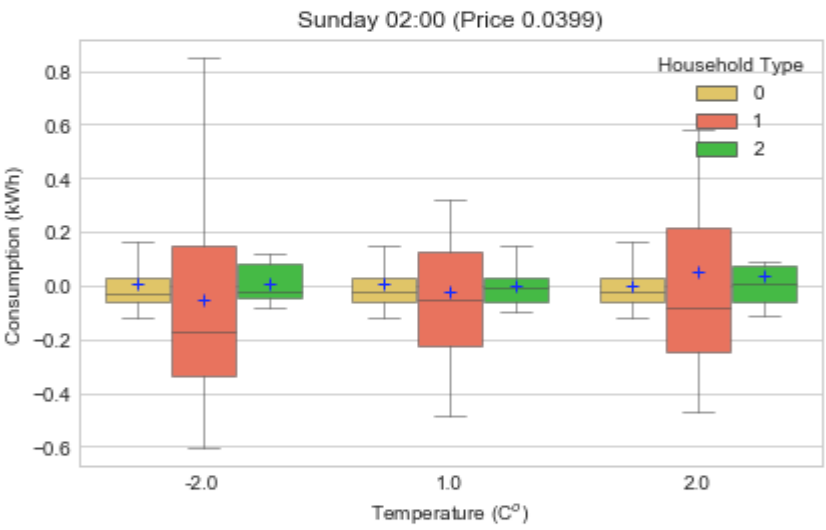
In [103]:
```
# Sunday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 6 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Price'] == prices[p])]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Price ' + str(prices[p]) +
')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day_hour, ax=ax_Ntou[-1
], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+",
"markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Price ' + str(prices[p]) +
')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Household Type')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

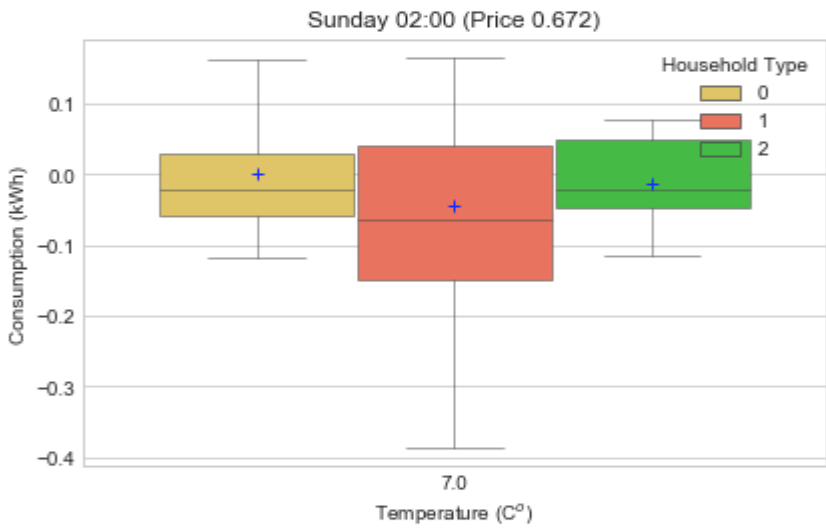

Sunday 00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-2.0
2.0
3.0
Temperature (C°)

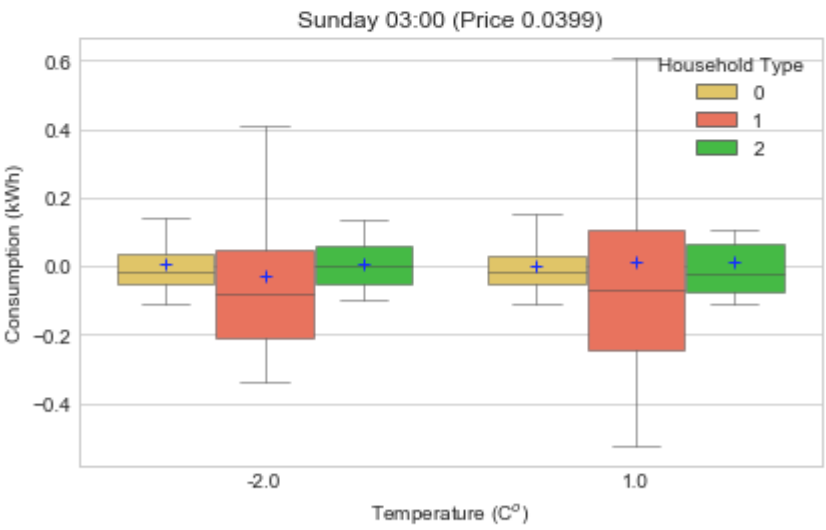

Sunday 00:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
7.0
Temperature (C°)

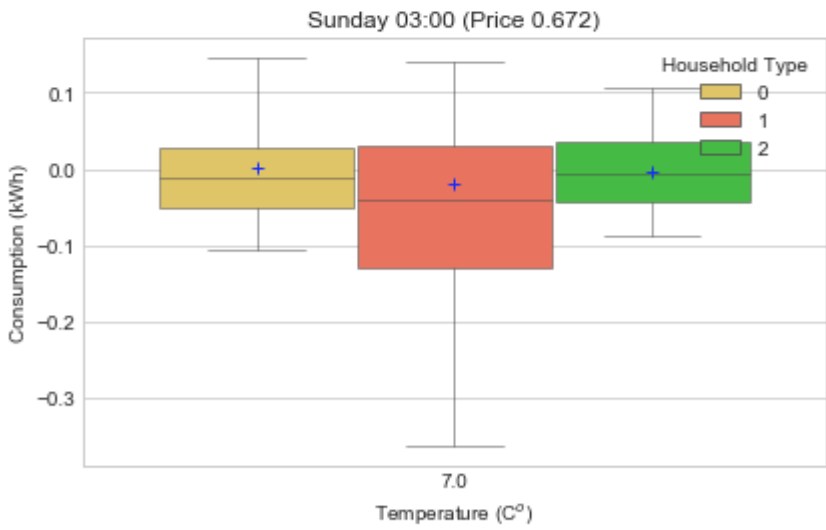

Sunday 01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-2.0
2.0
Temperature (C°)

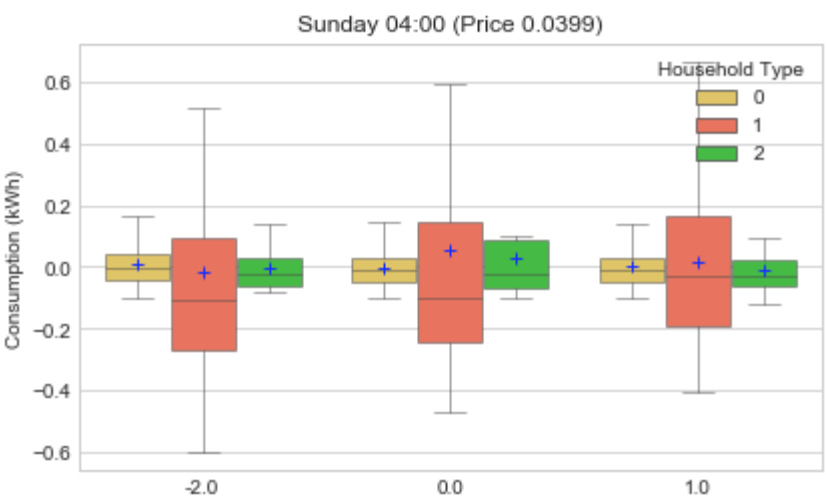

Sunday 01:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
7.0
Temperature (C°)

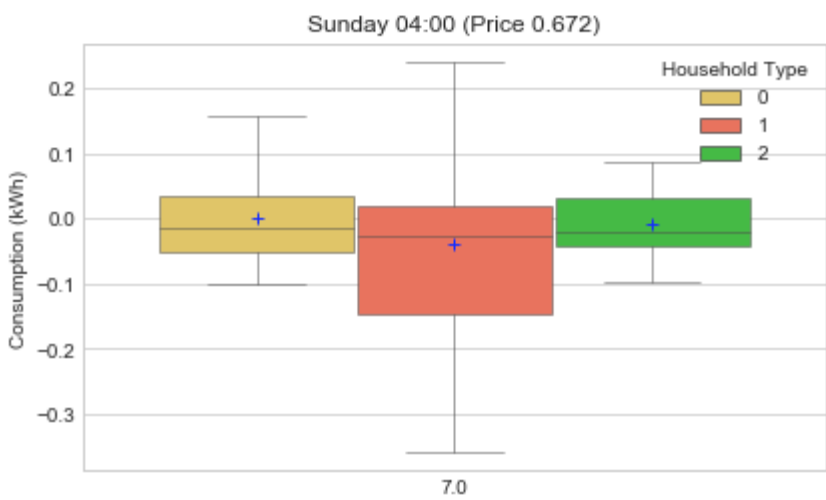

Sunday 02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
1.0
2.0
Temperature (C°)

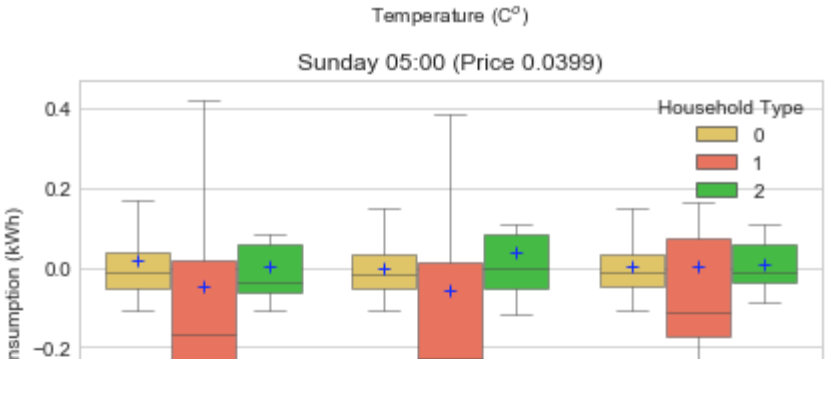

Sunday 02:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.1
0.0
−0.1
−0.2
−0.3
−0.4
7.0
Temperature (C°)

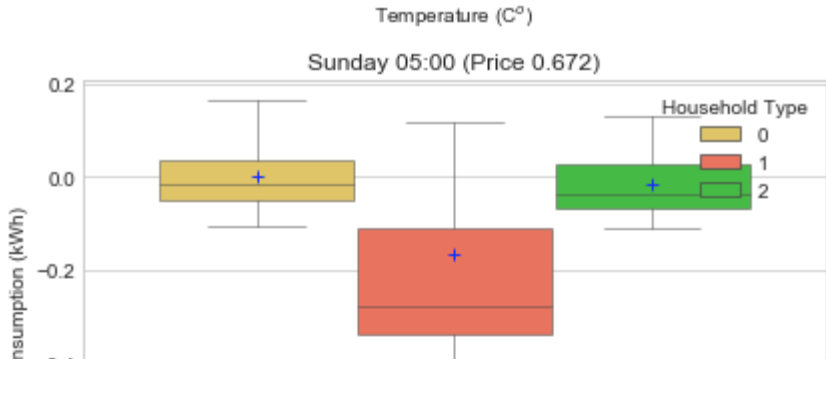

Sunday 03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0
1.0
Temperature (C°)

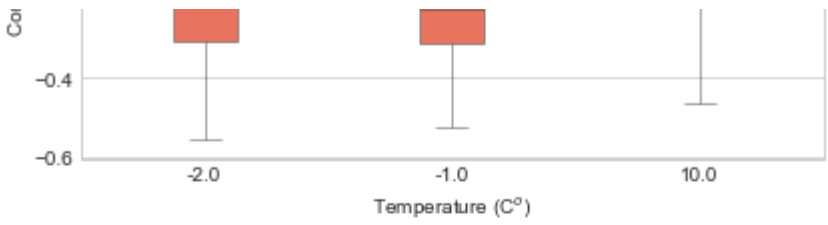

Sunday 03:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.1
0.0
−0.1
−0.2
−0.3
7.0
Temperature (C°)

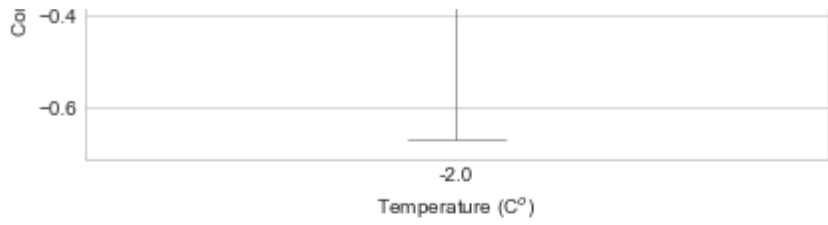

Sunday 04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
0.0
1.0
Temperature (C°)

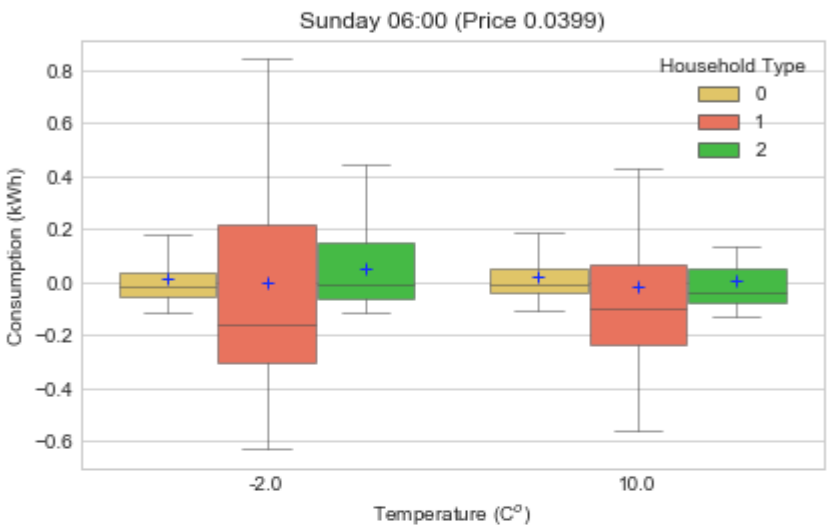

Sunday 04:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.2
0.1
0.0
−0.1
−0.2
−0.3
7.0
Temperature (C°)

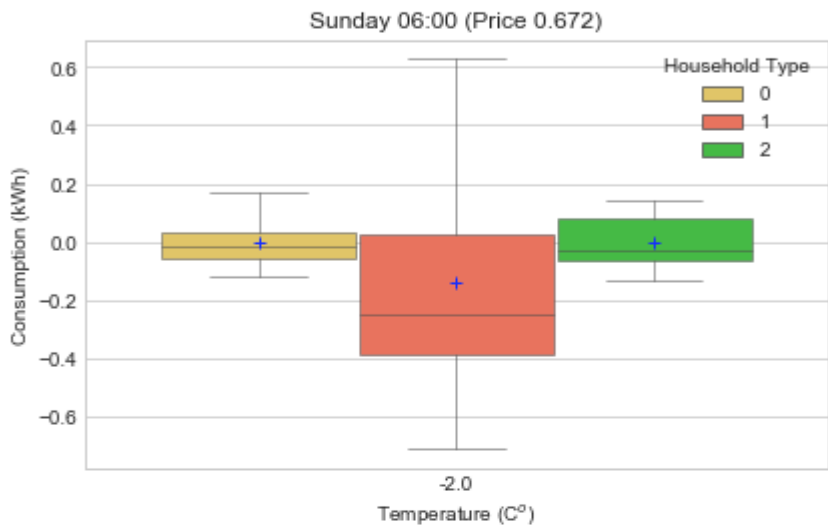

Sunday 05:00 (Price 0.0399)
Household Type
0
1
2
0.4
0.2
0.0
−0.2

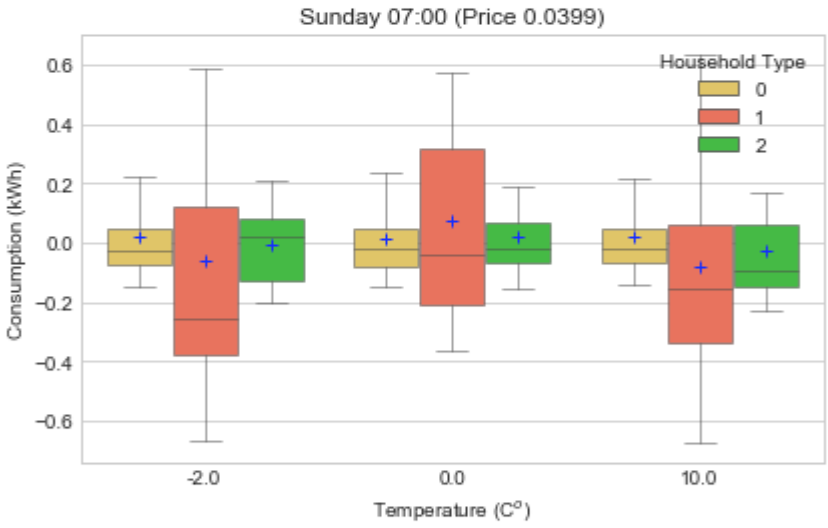

Sunday 05:00 (Price 0.672)
Household Type
0
1
2
0.2
0.0
−0.2

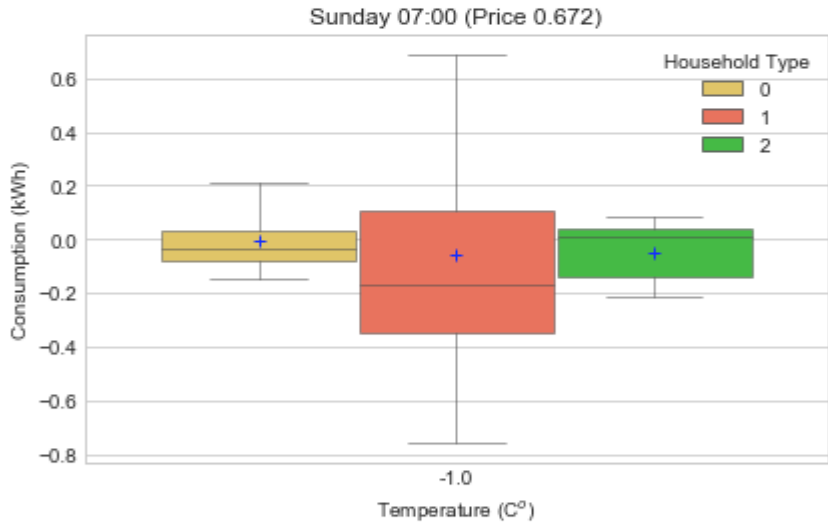

Co
−0.4
−0.6
-2.0
-1.0
10.0
Temperature (C°)

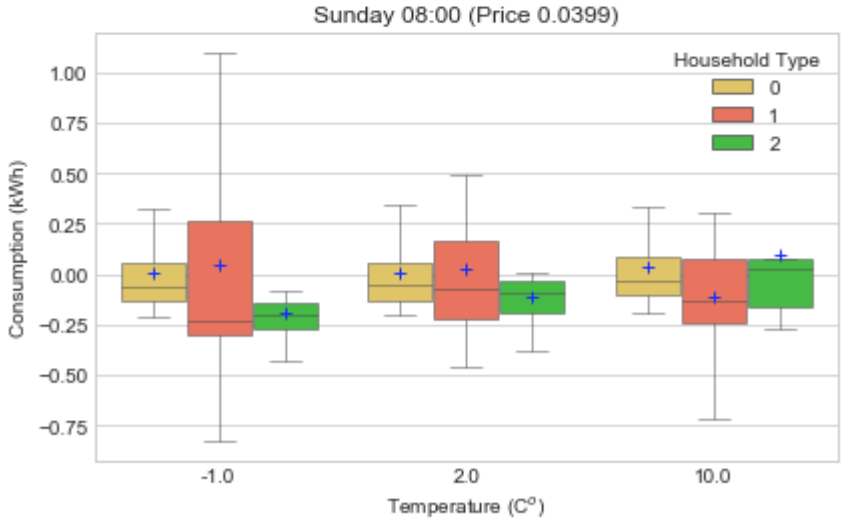

Co
−0.4
−0.6
-2.0
Temperature (C°)

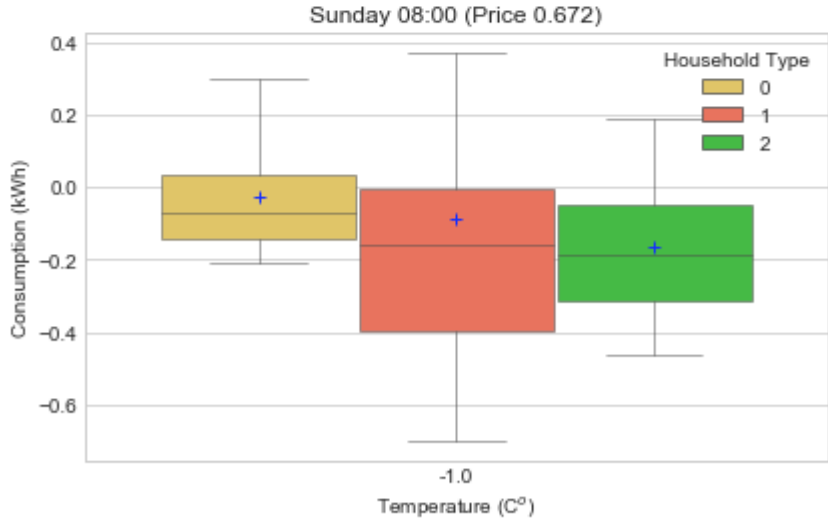

Sunday 06:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
10.0
Temperature (C°)

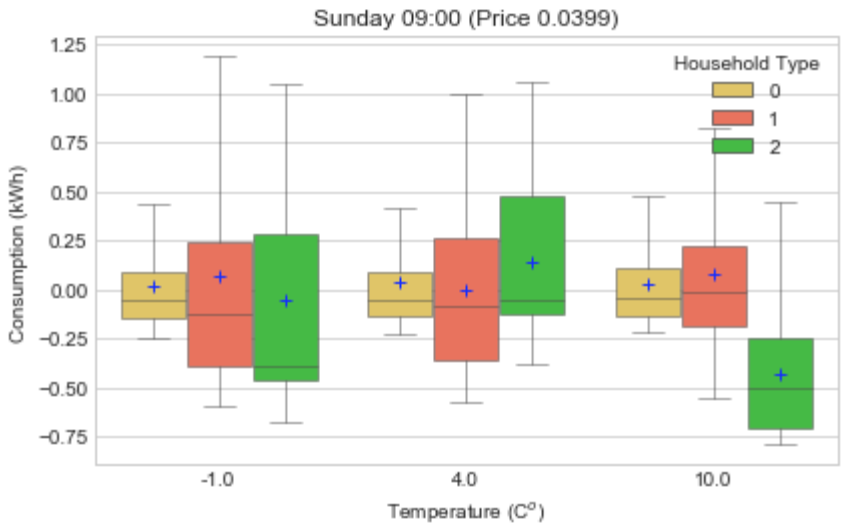

Sunday 06:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
Temperature (C°)

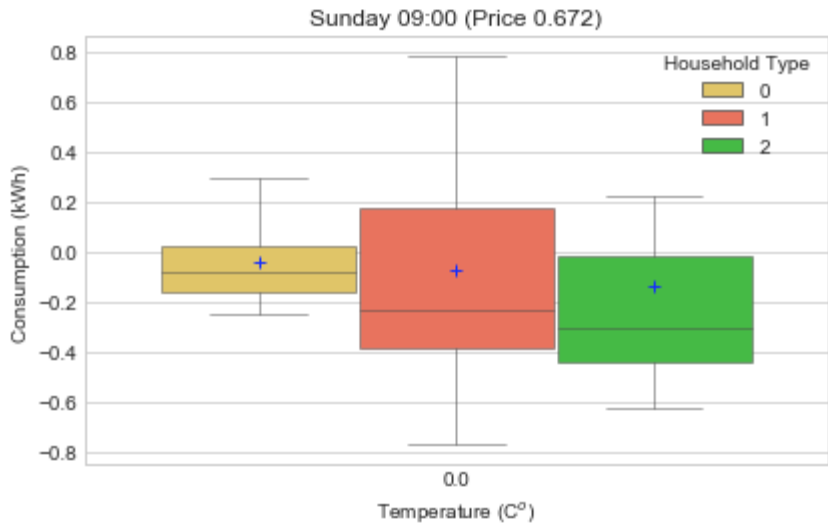

Sunday 07:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
0.0
10.0
Temperature (C°)

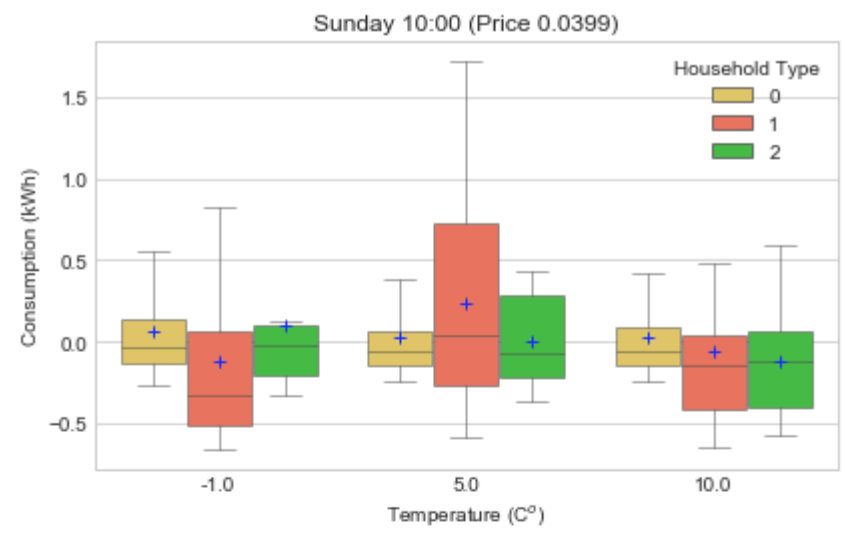

Sunday 07:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
-1.0
Temperature (C°)

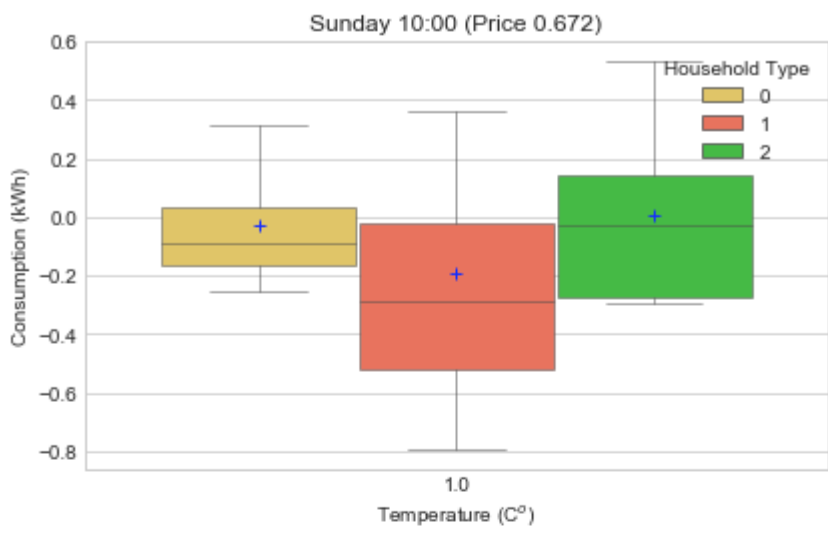

Sunday 08:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-1.0
2.0
10.0
Temperature (C°)


Sunday 08:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
-1.0
Temperature (C°)


Sunday 09:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-1.0
4.0
10.0
Temperature (C°)

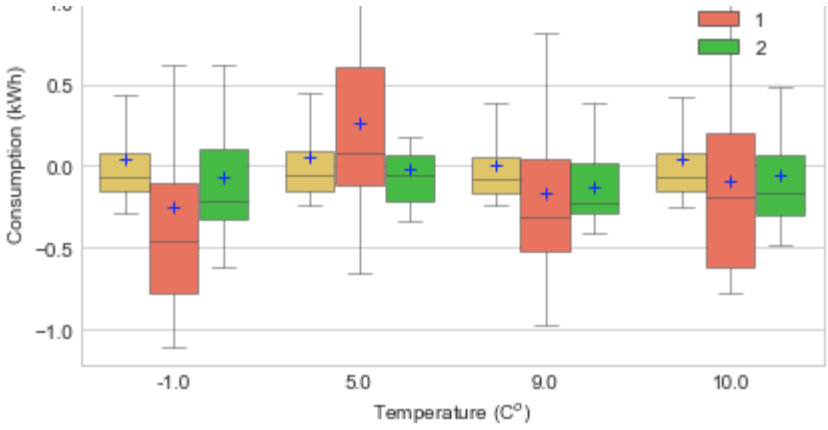

Sunday 09:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.0
Temperature (C°)

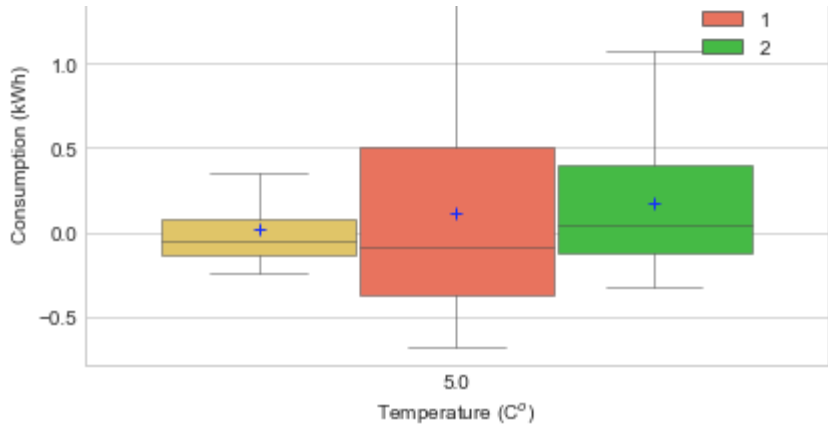

Sunday 10:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-1.0
5.0
10.0
Temperature (C°)

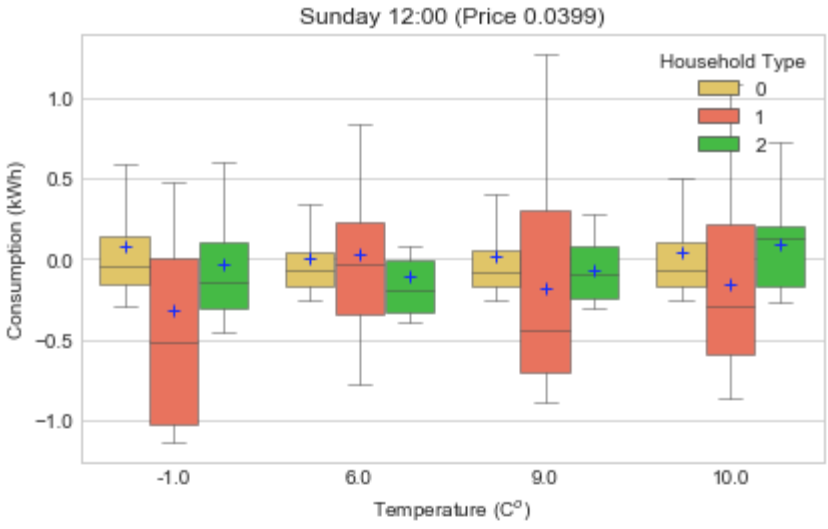

Sunday 10:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
1.0
Temperature (C°)

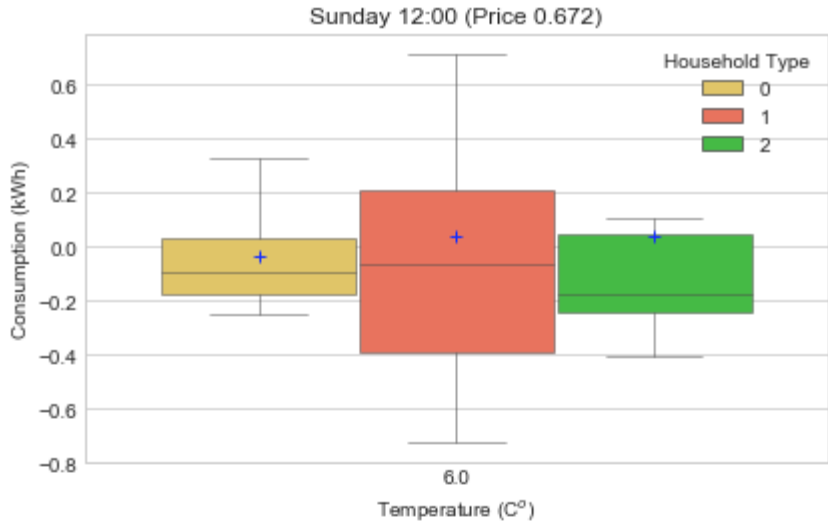

Sunday 11:00 (Price 0.0399)
Household Type
0
1.0

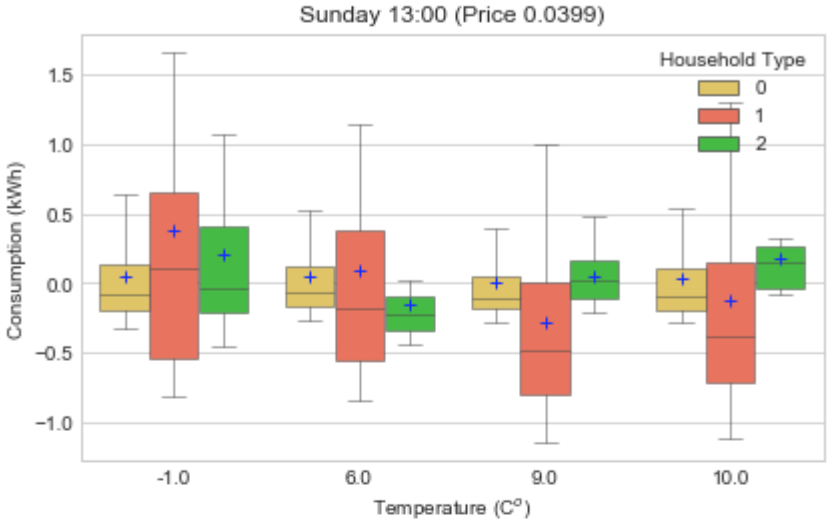

Sunday 11:00 (Price 0.672)
Household Type
0
1.5

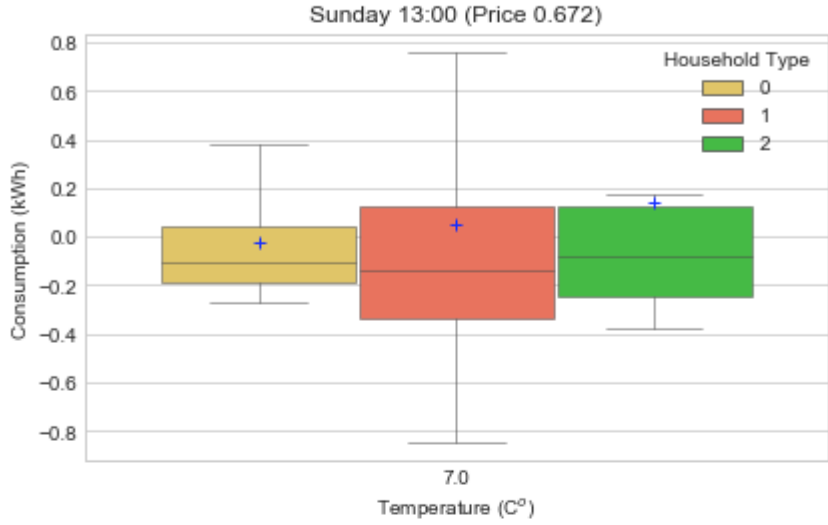

1
2
Consumption (kWh)
0.5
0.0
−0.5
−1.0
-1.0
5.0
9.0
10.0
Temperature (C°)

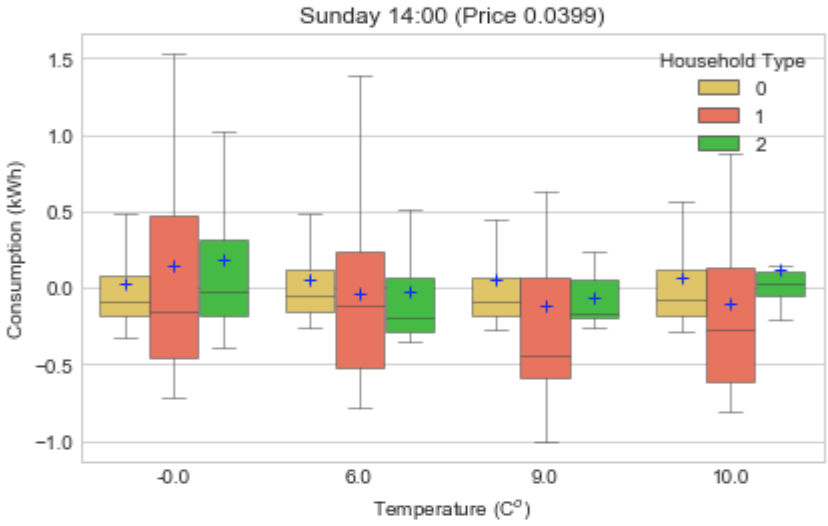

1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
5.0
Temperature (C°)

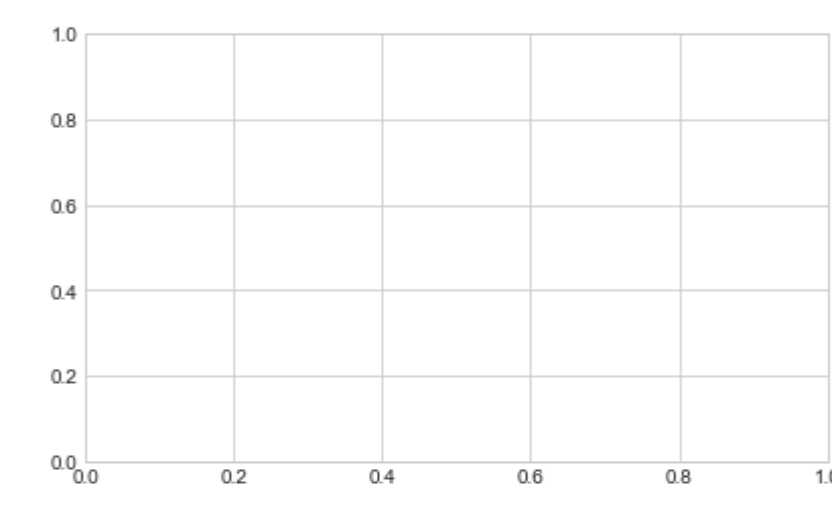

Sunday 12:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
6.0
9.0
10.0
Temperature (C°)

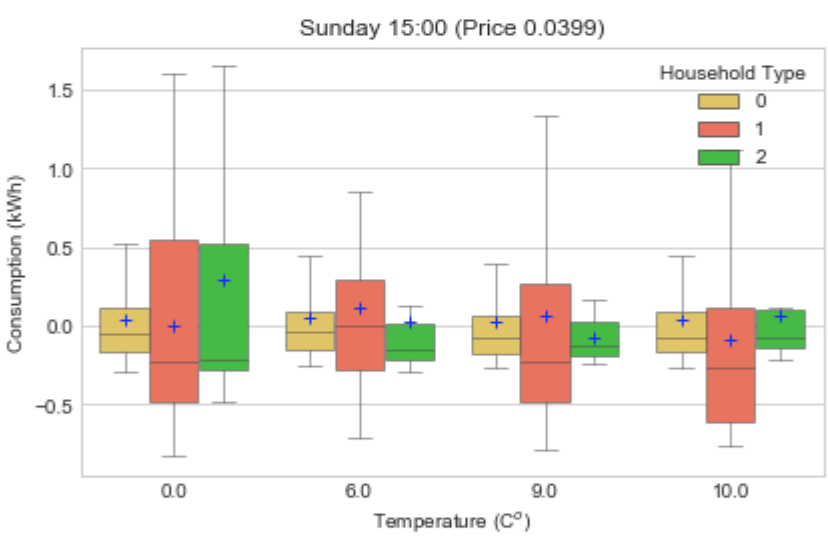

Sunday 12:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
6.0
Temperature (C°)

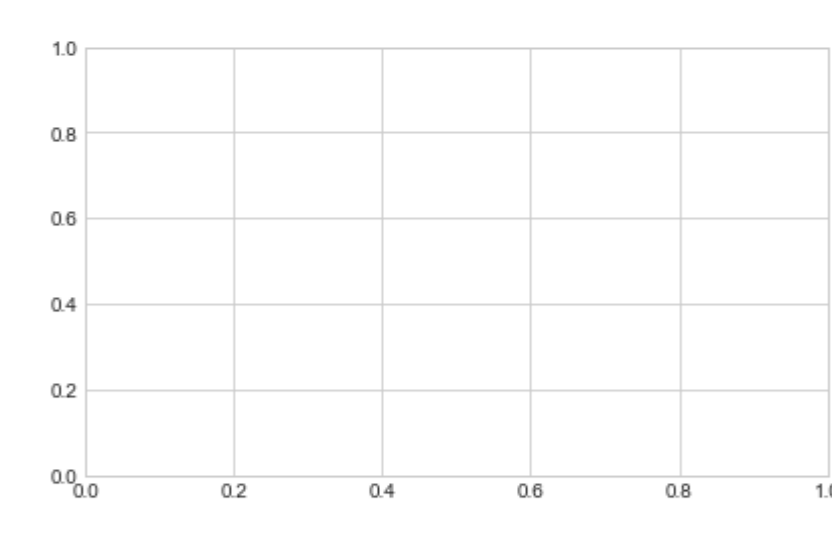

Sunday 13:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
6.0
9.0
10.0
Temperature (C°)

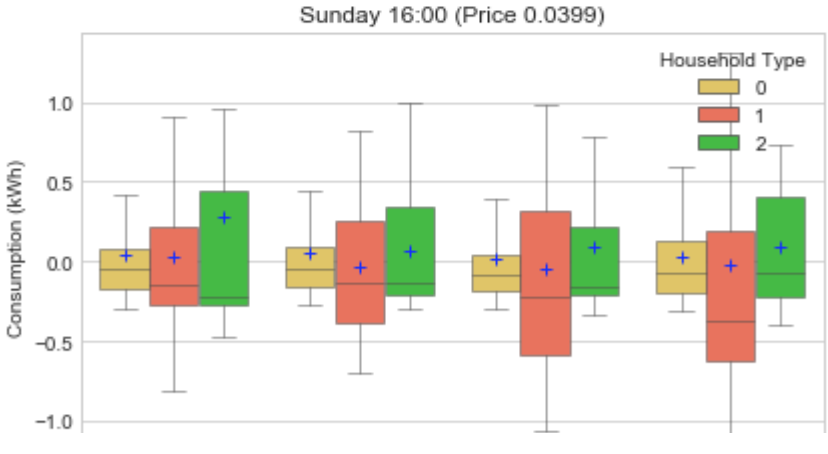

Sunday 13:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
7.0
Temperature (C°)

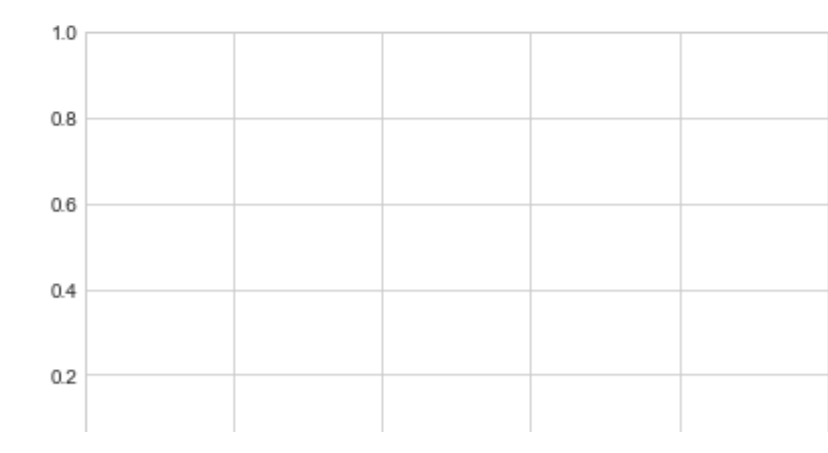

Sunday 14:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-0.0
6.0
9.0
10.0
Temperature (C°)


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Sunday 15:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
0.0
6.0
9.0
10.0
Temperature (C°)

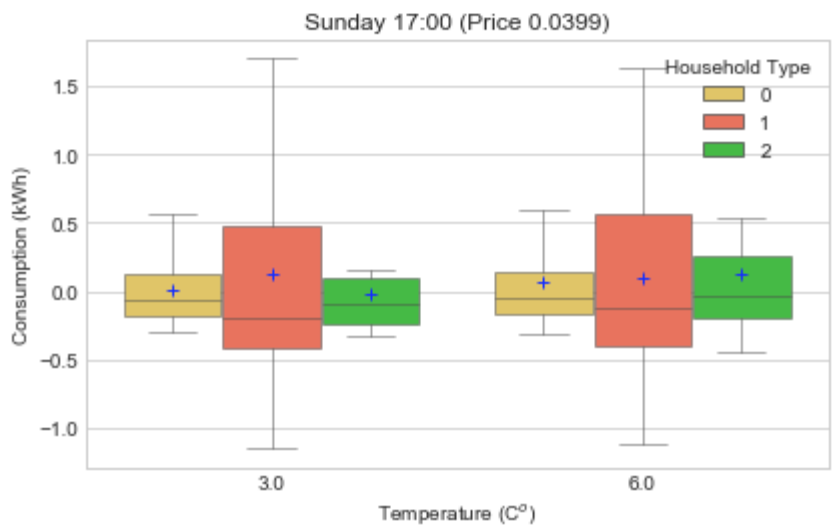

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

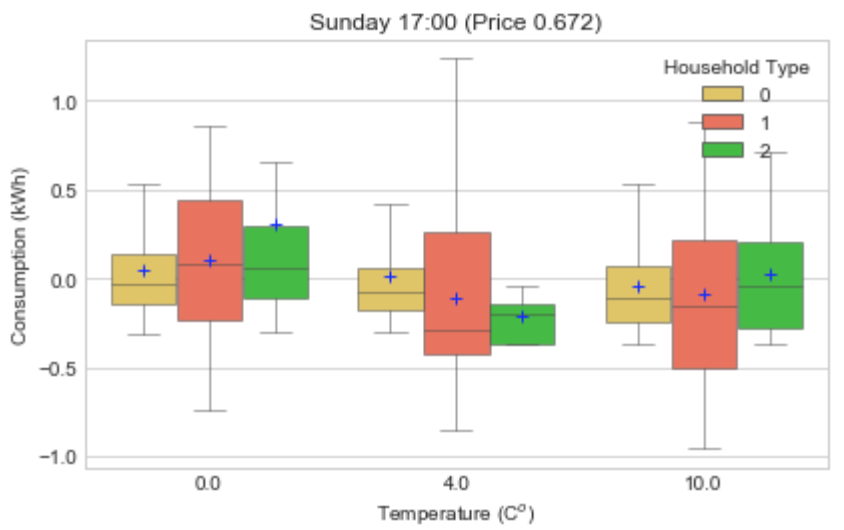

Sunday 16:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0

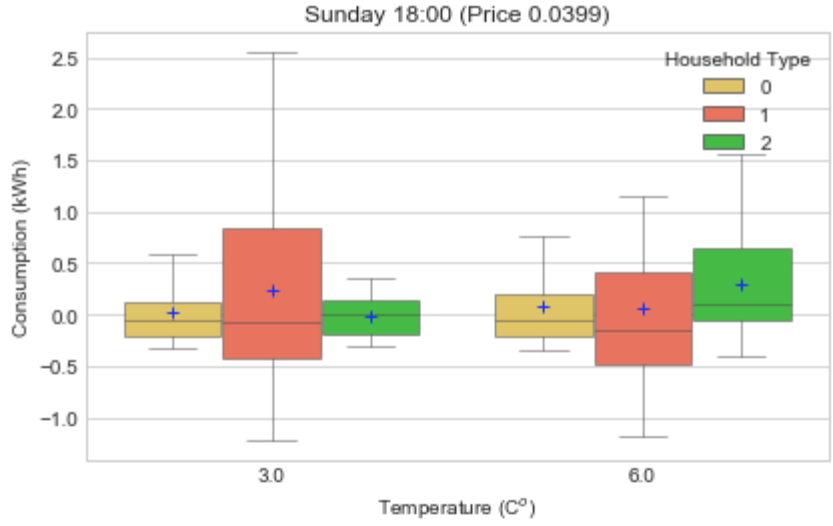

1.0
0.8
0.6
0.4
0.2

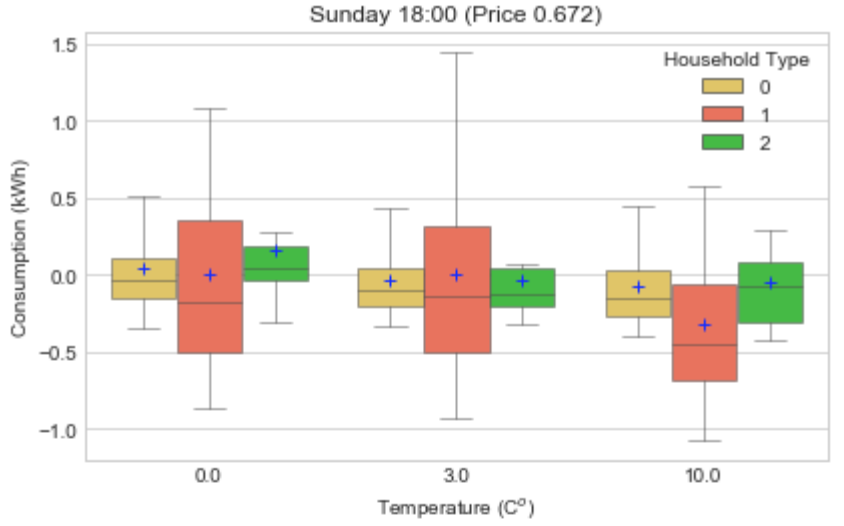

0.0
5.0
9.0
10.0
Temperature (C°)

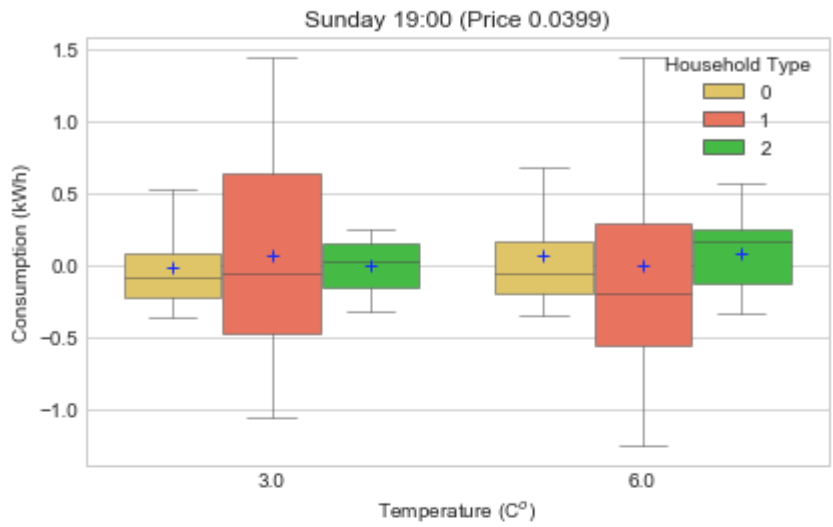

0.0
0.0
0.2
0.4
0.6
0.8
1.0

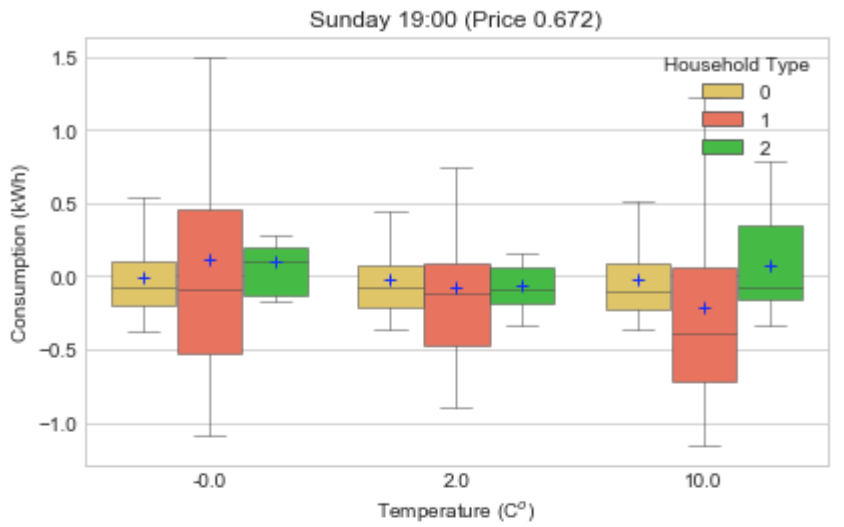

Sunday 17:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
3.0
6.0
Temperature (C°)

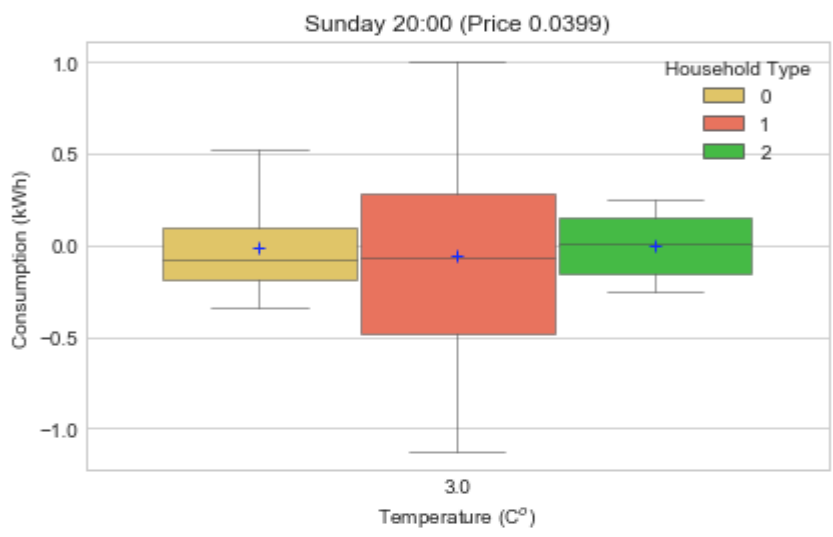

Sunday 17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
0.0
4.0
10.0
Temperature (C°)

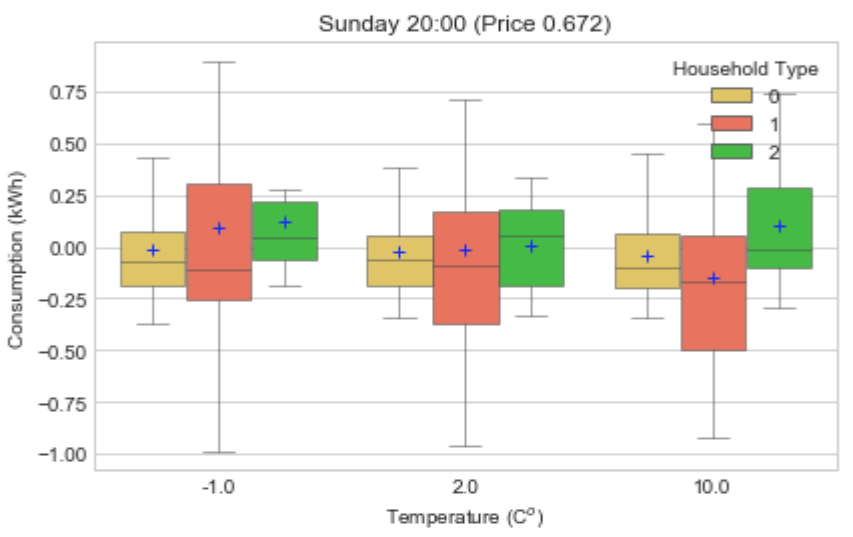

Sunday 18:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
3.0
6.0
Temperature (C°)

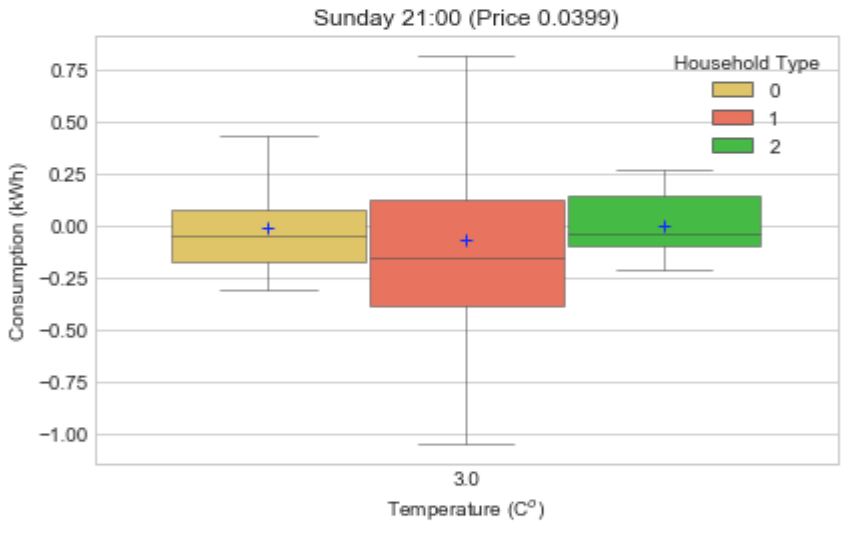

Sunday 18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
3.0
10.0
Temperature (C°)

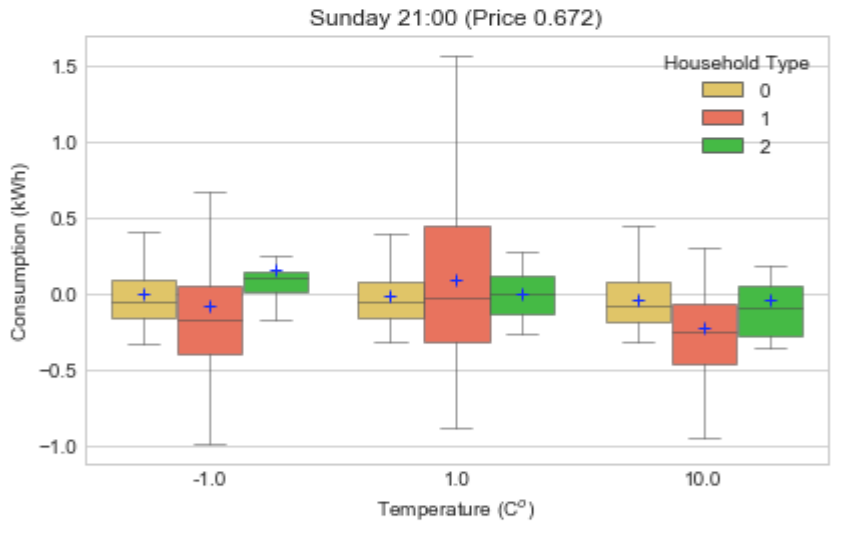

Sunday 19:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
3.0
6.0
Temperature (C°)

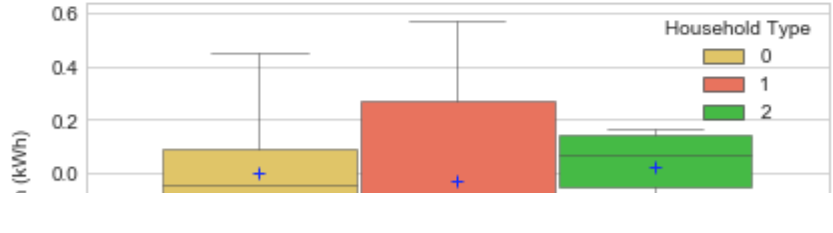

Sunday 19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-0.0
2.0
10.0
Temperature (C°)


Sunday 20:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
3.0
Temperature (C°)

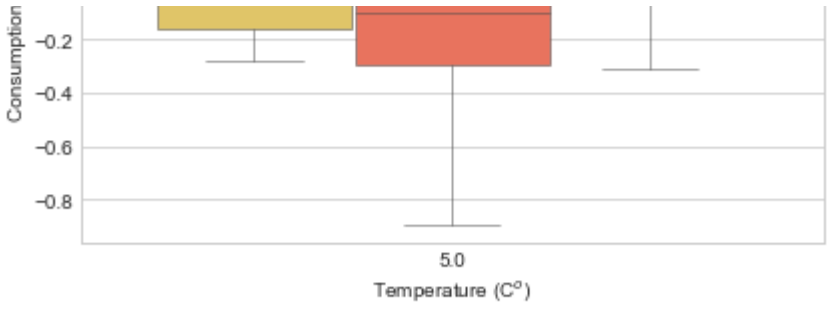

Sunday 20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
-1.0
2.0
10.0
Temperature (C°)

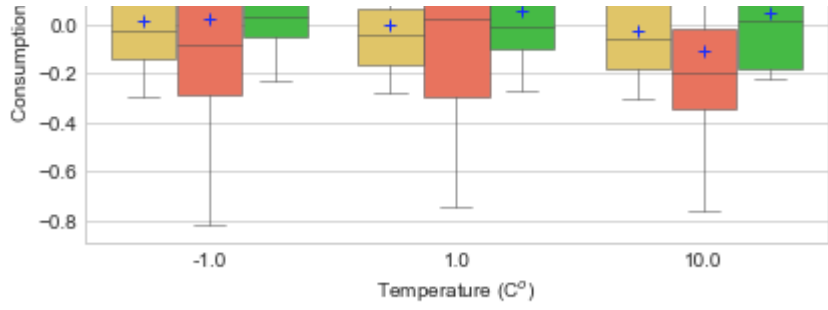

Sunday 21:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
3.0
Temperature (C°)

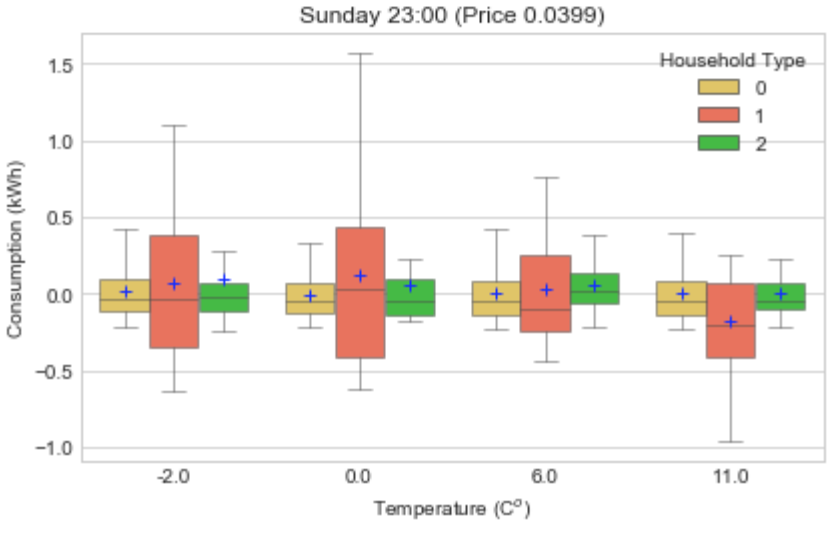

Sunday 21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
1.0
10.0
Temperature (C°)

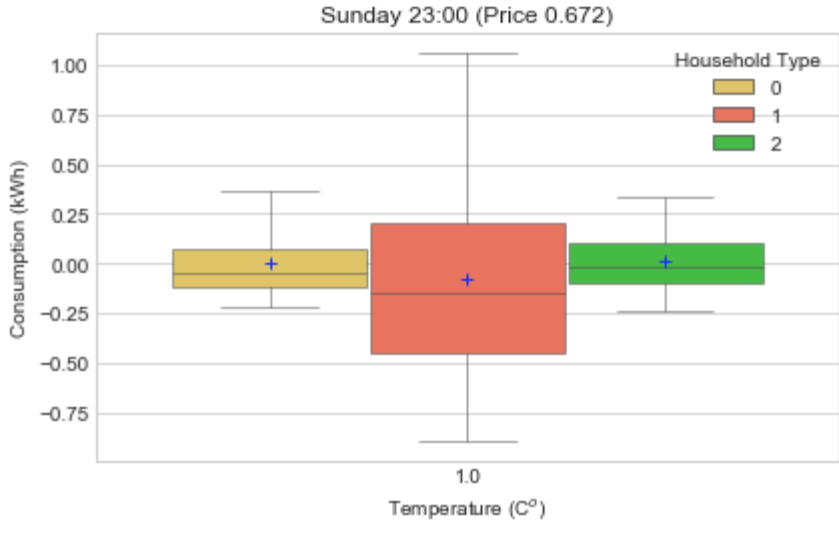

Sunday 22:00 (Price 0.0399)
Household Type
0
1
2
0.6
0.4
0.2
0.0


Sunday 22:00 (Price 0.672)
Household Type
0
1
2
0.8
0.6
0.4
0.2

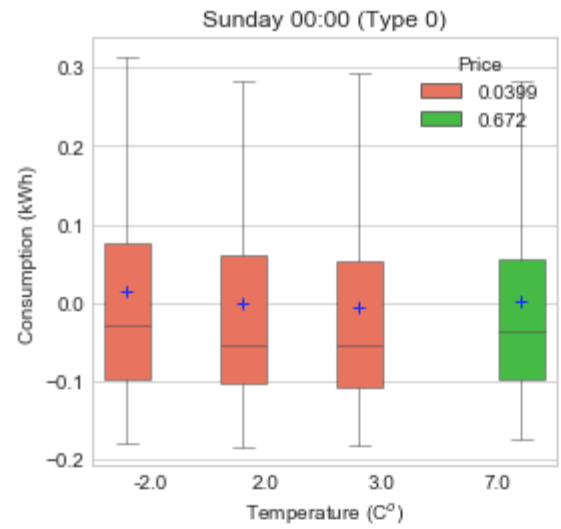

Consumption
−0.2
−0.4
−0.6
−0.8
5.0
Temperature (C°)

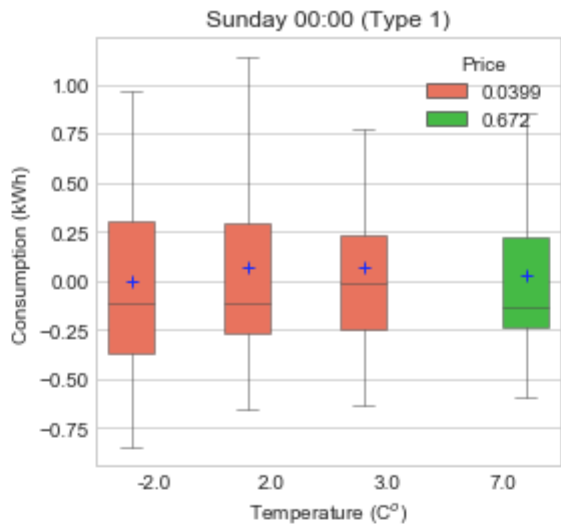

Consumption
0.0
−0.2
−0.4
−0.6
−0.8
-1.0
1.0
10.0
Temperature (C°)

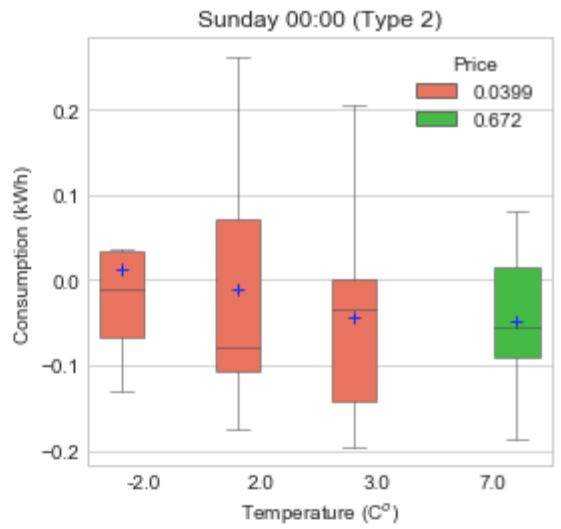

Sunday 23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-2.0
0.0
6.0
11.0
Temperature (C°)

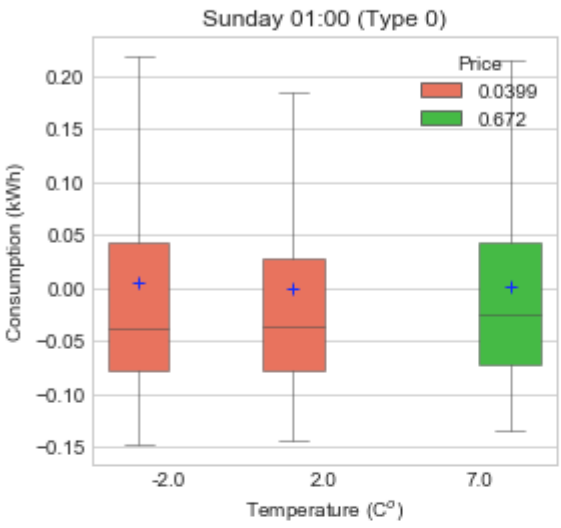

Sunday 23:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
1.0
Temperature (C°)

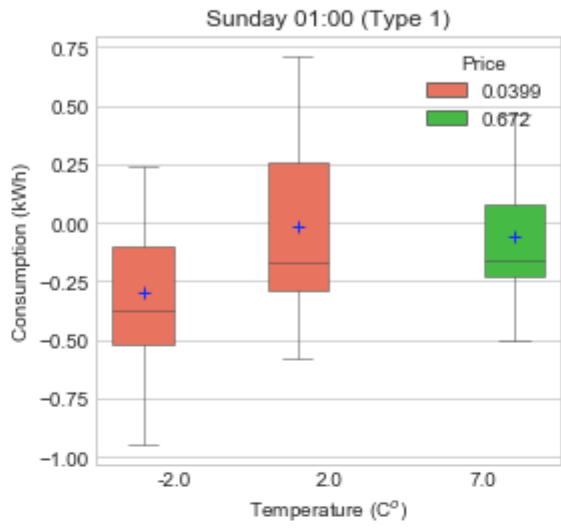
In [134]:
```
# price comparison
# Sunday hourly price responsiveness with different temperature and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
day_of_week = 6 # 0 is Monday, 6 is Sunday
df_day = df_all_res_long[df_all_res_long['Day of week'] == day_of_week]
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_day_hour = df_day[(df_day['Hour of day'] == i) & (df_day['Household type'] == g)]
        if df_day_hour.shape[0] >= 1:
            if i <= 9:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' 0' + str(i) + ':00 (Type ' + str(g) + ')' )
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
            else:
                sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day_hour, ax=ax_Ntou[-1], palett
e=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkc
d:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
                ax_Ntou[-1].set_title(weeks[day_of_week] + ' ' + str(i) + ':00 (Type ' + str(g) + ')')
                ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
                ax_Ntou[-1].set_ylabel('Consumption (kWh)')
                l = ax_Ntou[-1].legend()
                l.set_title('Price')
                for i,artist in enumerate(ax_Ntou[-1].artists):
                    artist.set_linewidth(0.5)
                    # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                    # Loop over them here, and use the same colour as above
                    for j in range(i*6,i*6+6):
                        line = ax_Ntou[-1].lines[j]
                        line.set_linewidth(0.5)
                ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

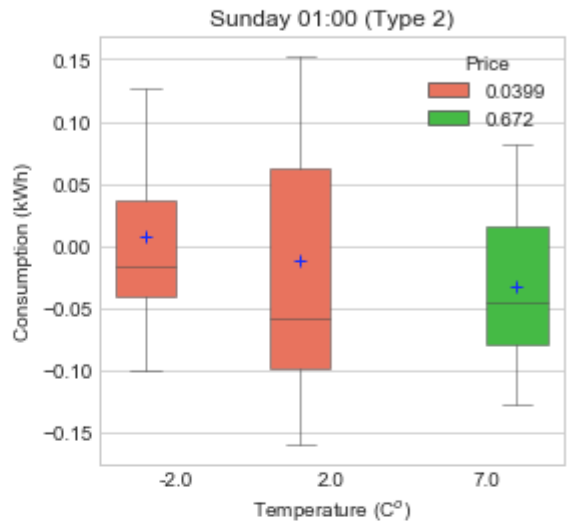

Sunday 00:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
2.0
3.0
7.0

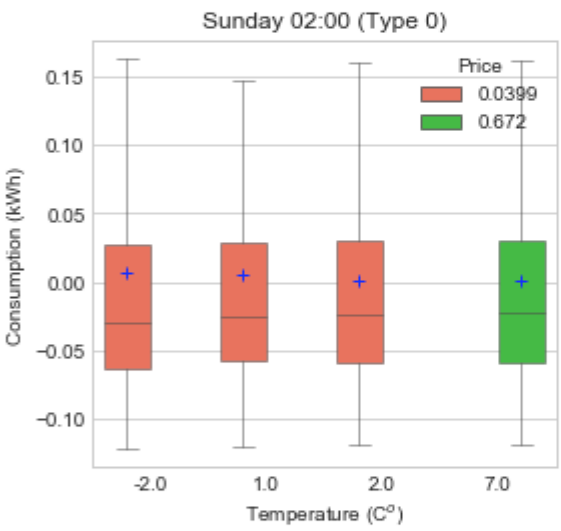

Sunday 00:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
2.0
3.0
7.0

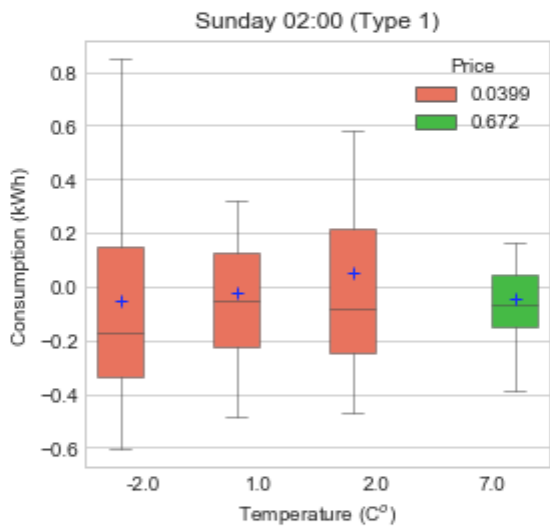

Sunday 00:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
2.0
3.0
7.0

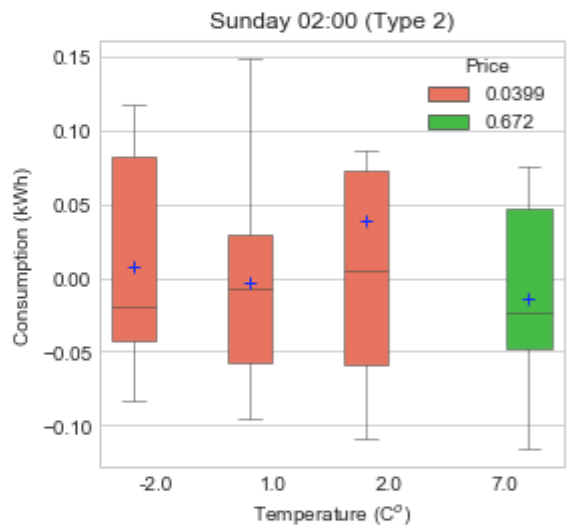

Sunday 01:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
2.0
3.0
7.0

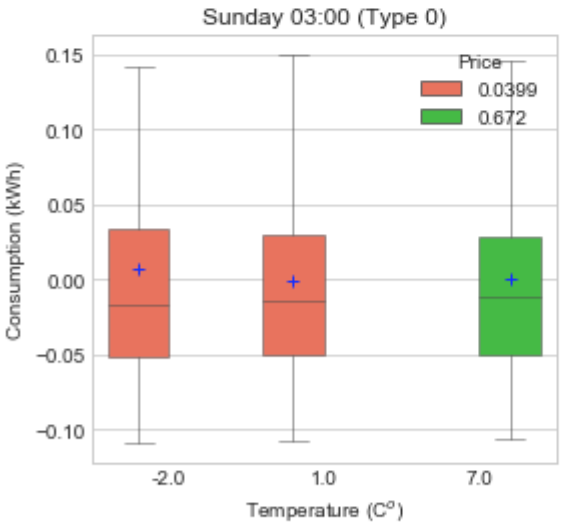

Sunday 01:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
2.0
7.0

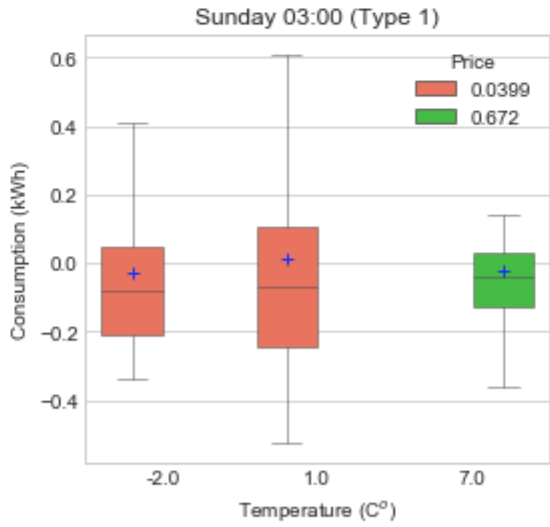

Sunday 01:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
2.0
7.0

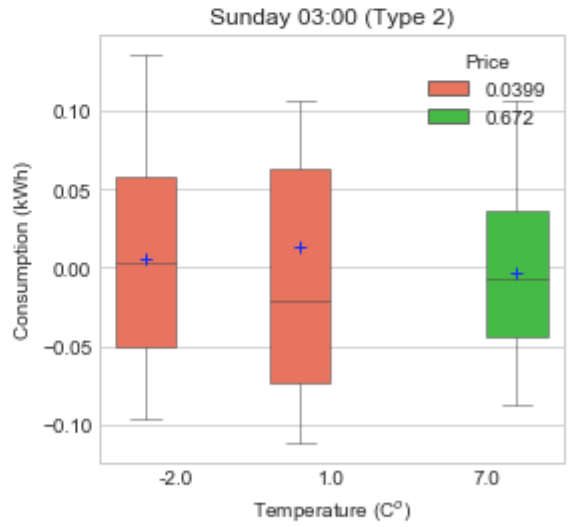

Sunday 02:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
1.0
2.0
7.0

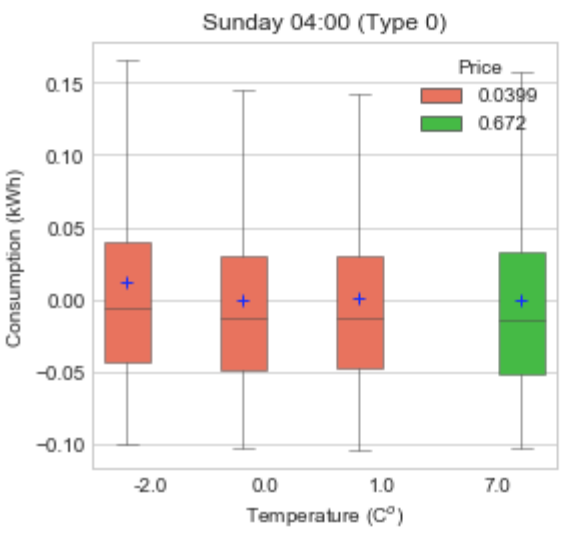

Sunday 02:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
1.0
2.0
7.0

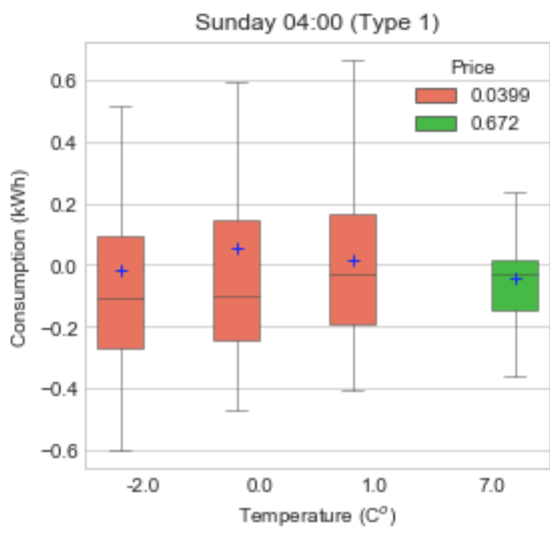

Sunday 02:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
1.0
2.0
7.0

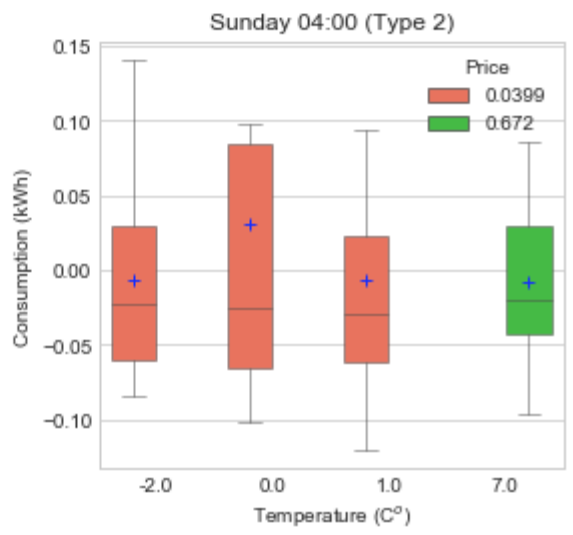

Sunday 03:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
1.0
7.0

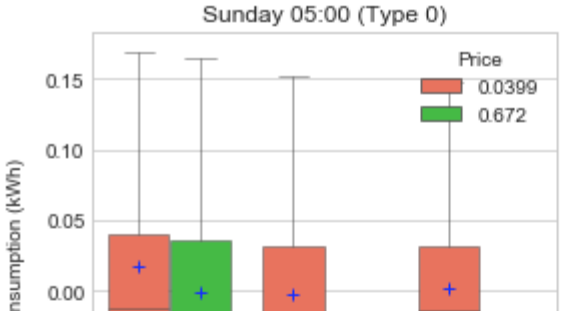

Sunday 03:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
1.0
7.0

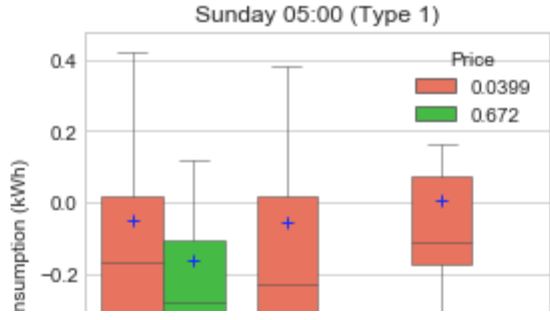

Sunday 03:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
1.0
7.0

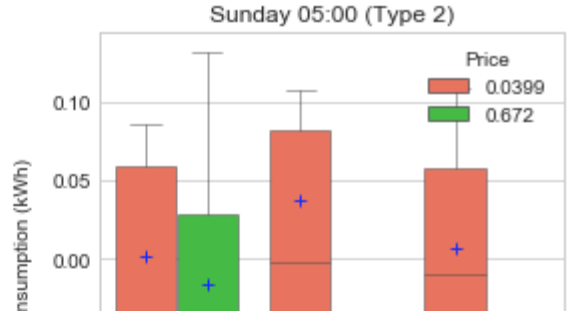

Sunday 04:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
0.0
1.0
7.0

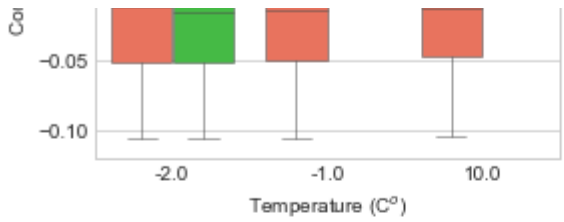

Sunday 04:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
0.0
1.0
7.0

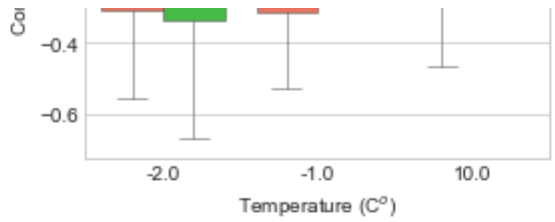

Sunday 04:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-2.0
0.0
1.0
7.0

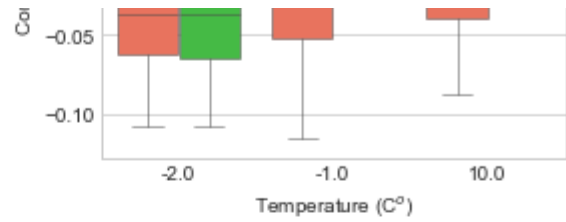

Sunday 05:00 (Type 0)
Price
0.0399
0.672
Temperature (C°)

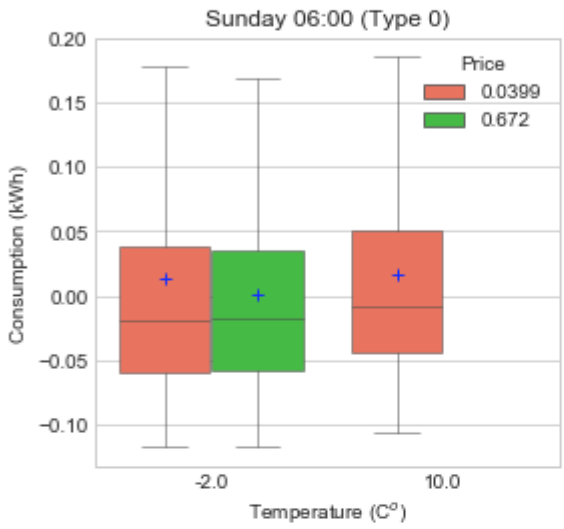

Sunday 05:00 (Type 1)
Price
0.0399
0.672
Temperature (C°)

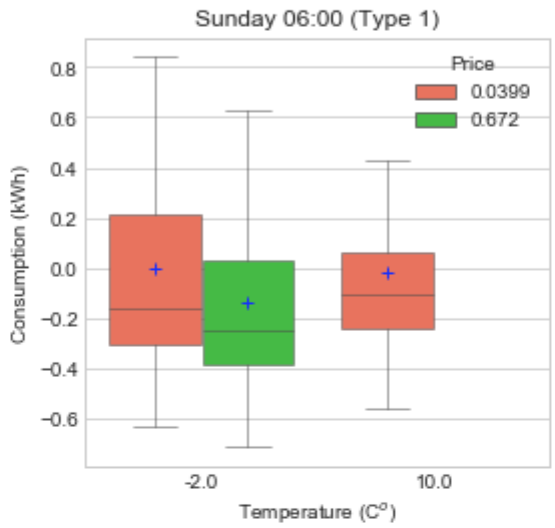

Sunday 05:00 (Type 2)
Price
0.0399
0.672
Temperature (C°)

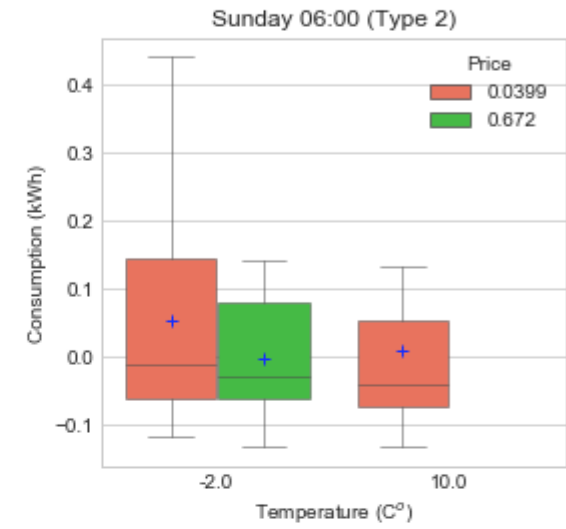

−0.05
−0.10
-2.0
-1.0
10.0
Temperature (C°)

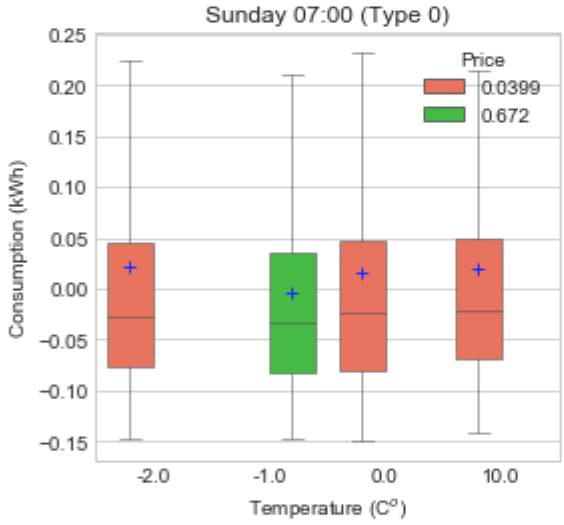

−0.4
−0.6
-2.0
-1.0
10.0
Temperature (C°)

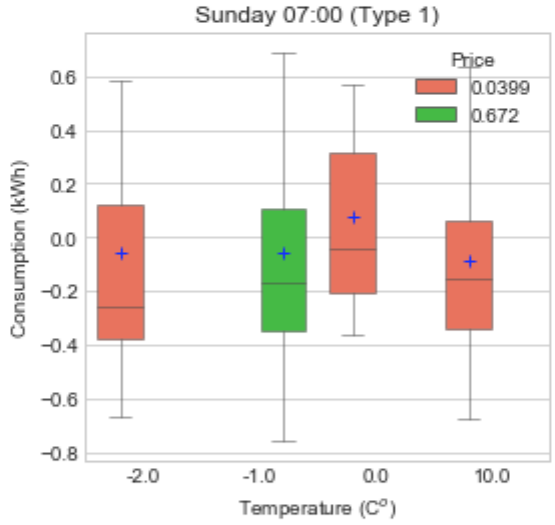

−0.05
−0.10
-2.0
-1.0
10.0
Temperature (C°)

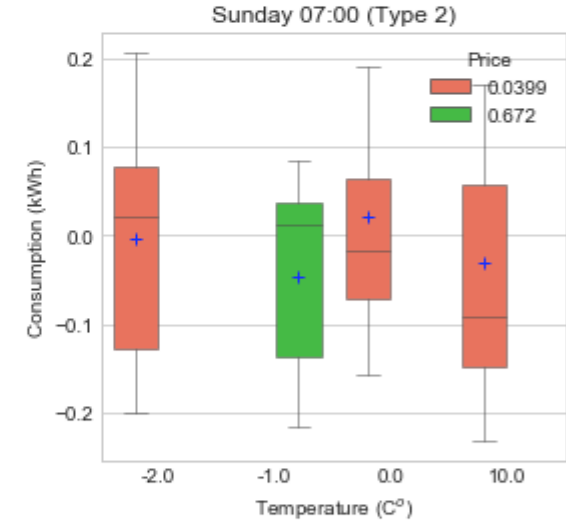

Sunday 06:00 (Type 0)
Consumption (kWh)
Price
0.0399
0.672
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
-2.0
10.0
Temperature (C°)

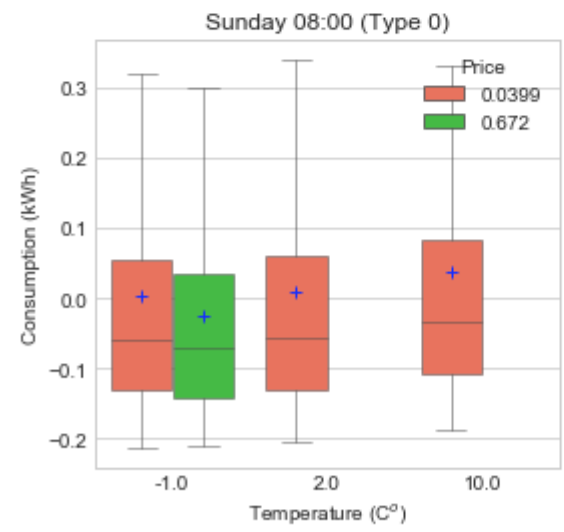

Sunday 06:00 (Type 1)
Consumption (kWh)
Price
0.0399
0.672
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-2.0
10.0
Temperature (C°)

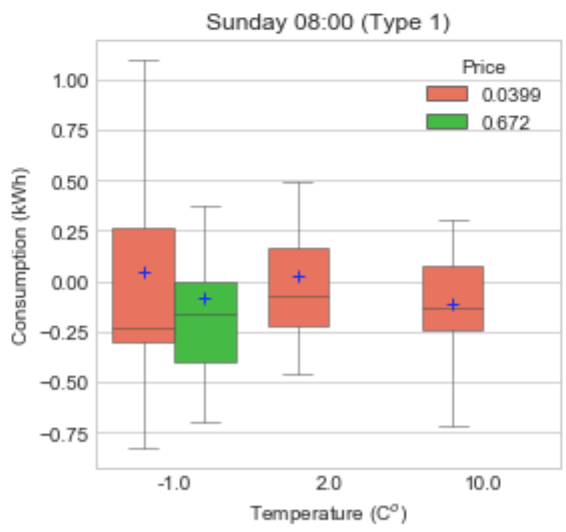

Sunday 06:00 (Type 2)
Consumption (kWh)
Price
0.0399
0.672
0.4
0.3
0.2
0.1
0.0
−0.1
-2.0
10.0
Temperature (C°)

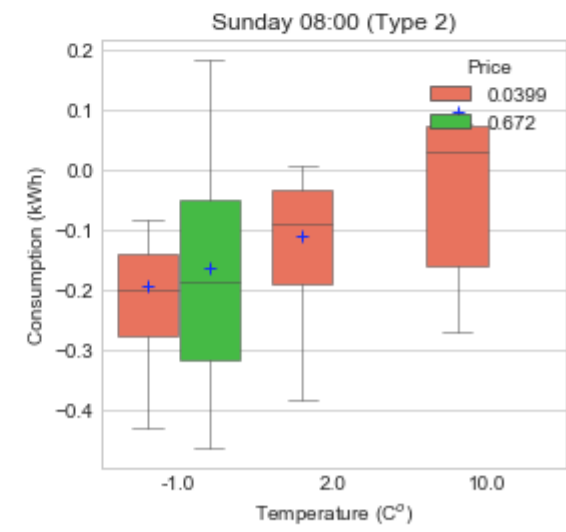

Sunday 07:00 (Type 0)
Consumption (kWh)
Price
0.0399
0.672
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
-2.0
-1.0
0.0
10.0
Temperature (C°)

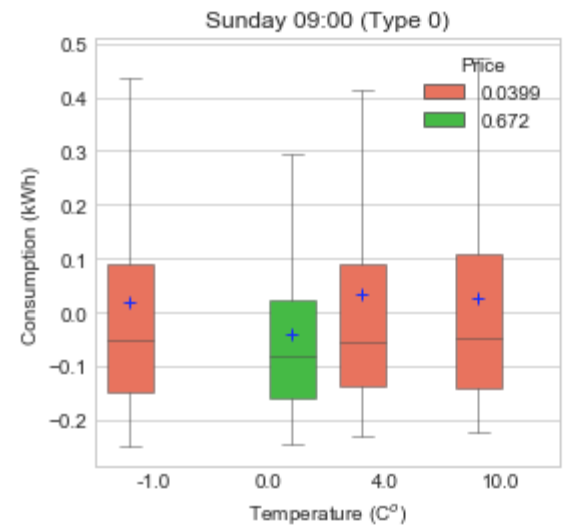

Sunday 07:00 (Type 1)
Consumption (kWh)
Price
0.0399
0.672
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
-2.0
-1.0
0.0
10.0
Temperature (C°)

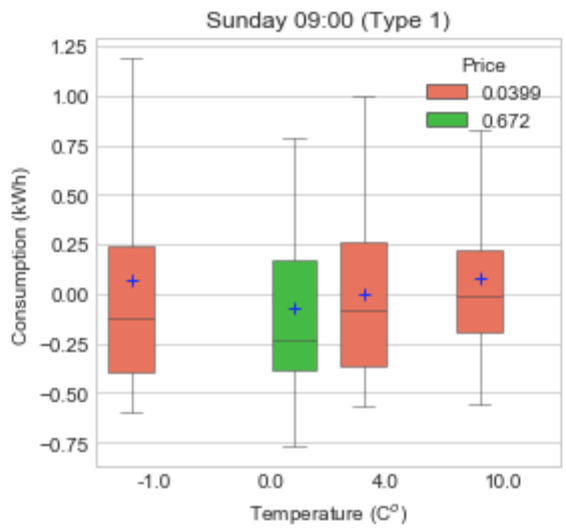

Sunday 07:00 (Type 2)
Consumption (kWh)
Price
0.0399
0.672
0.2
0.1
0.0
−0.1
−0.2
-2.0
-1.0
0.0
10.0
Temperature (C°)

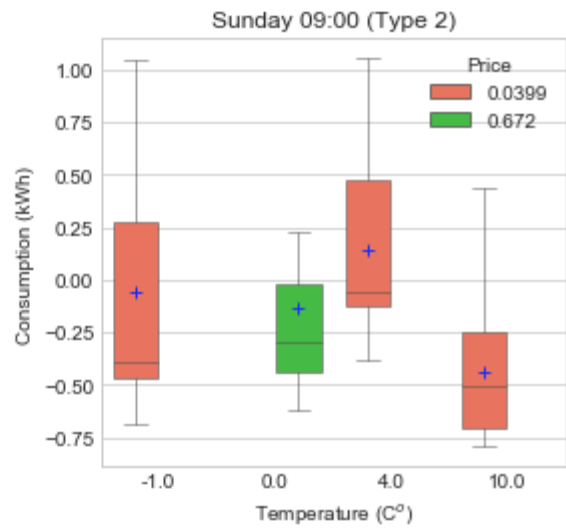

Sunday 08:00 (Type 0)
Consumption (kWh)
Price
0.0399
0.672
0.3
0.2
0.1
0.0
−0.1
−0.2
-1.0
2.0
10.0
Temperature (C°)

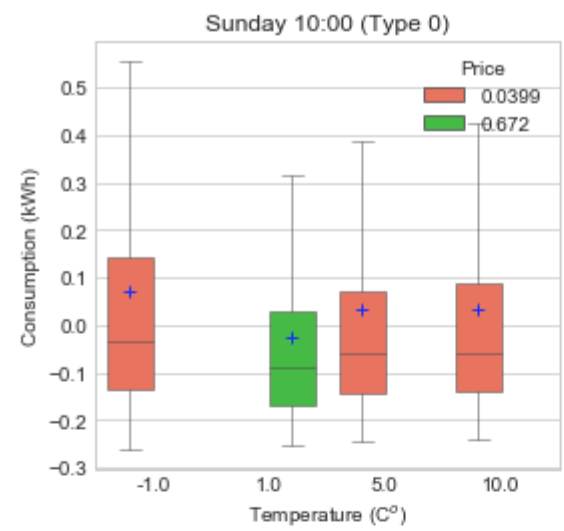

Sunday 08:00 (Type 1)
Consumption (kWh)
Price
0.0399
0.672
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-1.0
2.0
10.0
Temperature (C°)

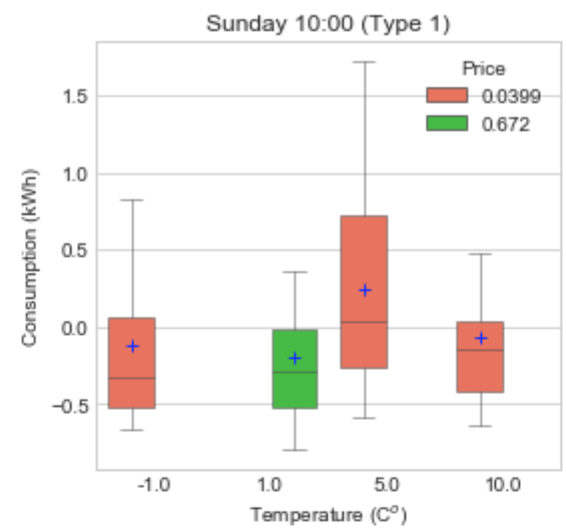

Sunday 08:00 (Type 2)
Consumption (kWh)
Price
0.0399
0.672
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
-1.0
2.0
10.0
Temperature (C°)

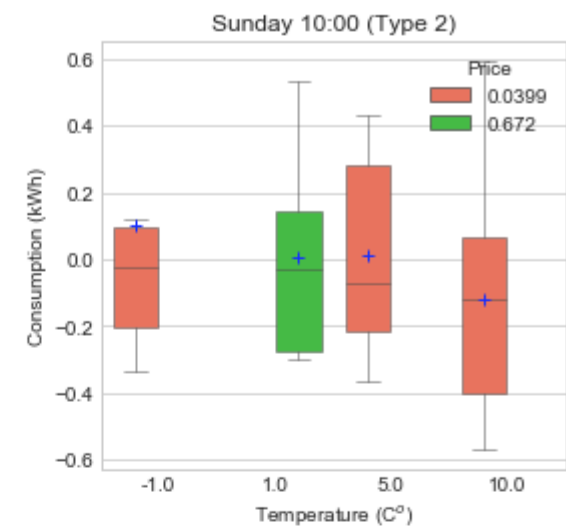

Sunday 09:00 (Type 0)
Consumption (kWh)
Price
0.0399
0.672
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
-1.0
0.0
4.0
10.0
Temperature (C°)


Sunday 09:00 (Type 1)
Consumption (kWh)
Price
0.0399
0.672
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-1.0
0.0
4.0
10.0
Temperature (C°)


Sunday 09:00 (Type 2)
Consumption (kWh)
Price
0.0399
0.672
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-1.0
0.0
4.0
10.0
Temperature (C°)


Sunday 10:00 (Type 0)
Consumption (kWh)
Price
0.0399
0.672
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-1.0
1.0
5.0
10.0
Temperature (C°)

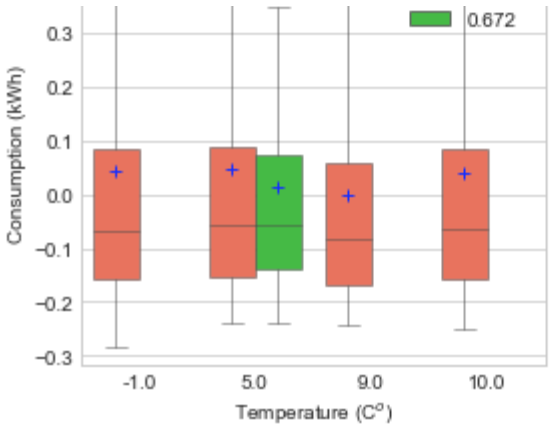

Sunday 10:00 (Type 1)
Consumption (kWh)
Price
0.0399
0.672
1.5
1.0
0.5
0.0
−0.5
-1.0
1.0
5.0
10.0
Temperature (C°)

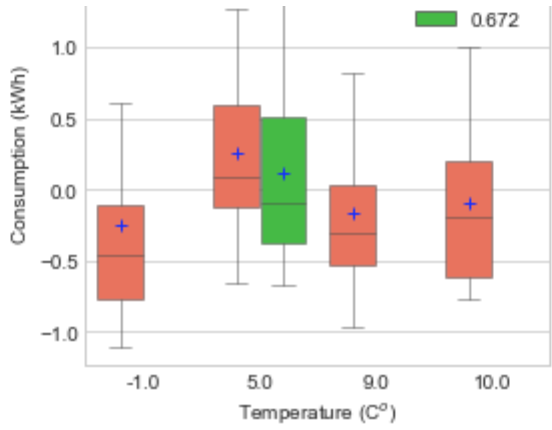

Sunday 10:00 (Type 2)
Consumption (kWh)
Price
0.0399
0.672
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-1.0
1.0
5.0
10.0
Temperature (C°)

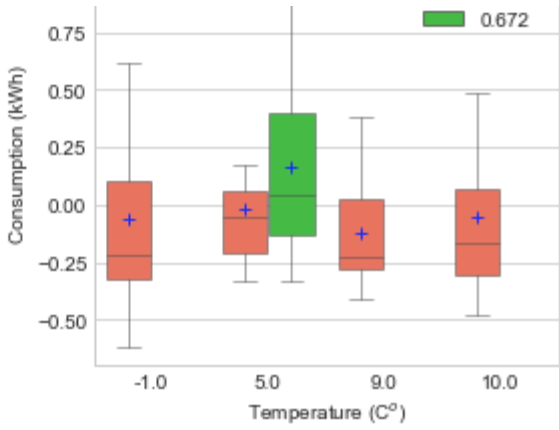

Sunday 11:00 (Type 0)
Consumption (kWh)
Price
0.0399
0.4

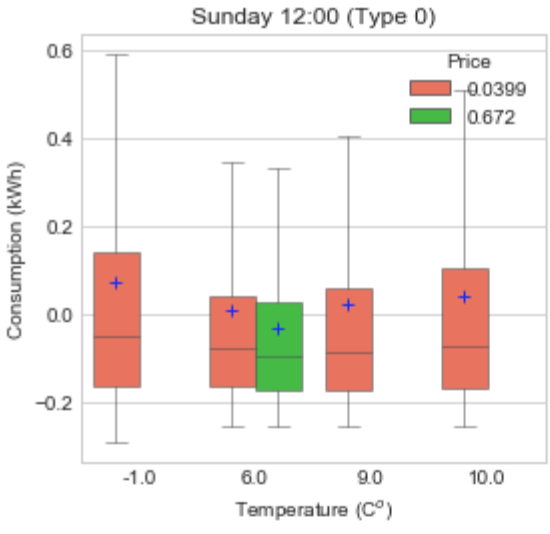

Sunday 11:00 (Type 1)
Consumption (kWh)
Price
0.0399
1.5

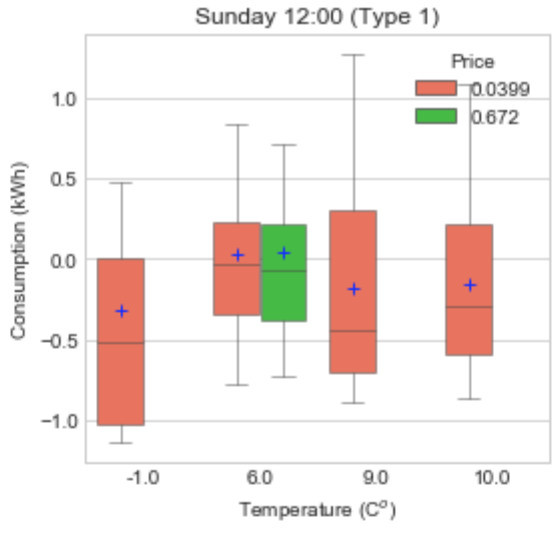

Sunday 11:00 (Type 2)
Consumption (kWh)
Price
0.0399
1.00

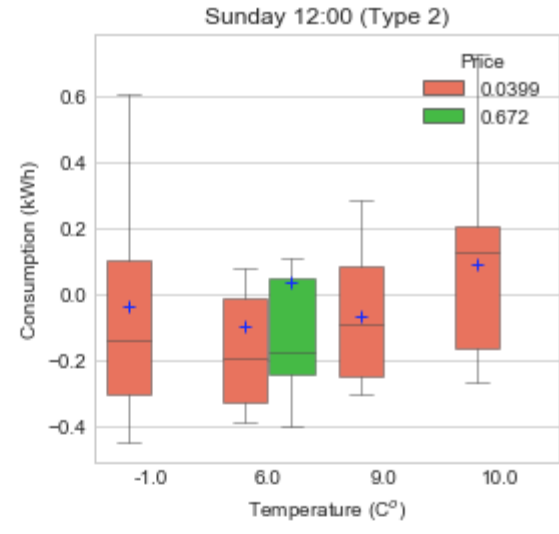

0.672
Consumption (kWh)
Temperature (C°)
-1.0
5.0
9.0
10.0

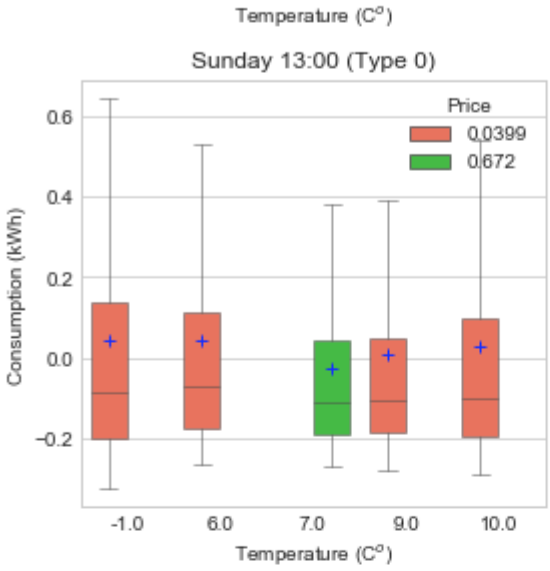

0.672
Consumption (kWh)
Temperature (C°)
-1.0
5.0
9.0
10.0

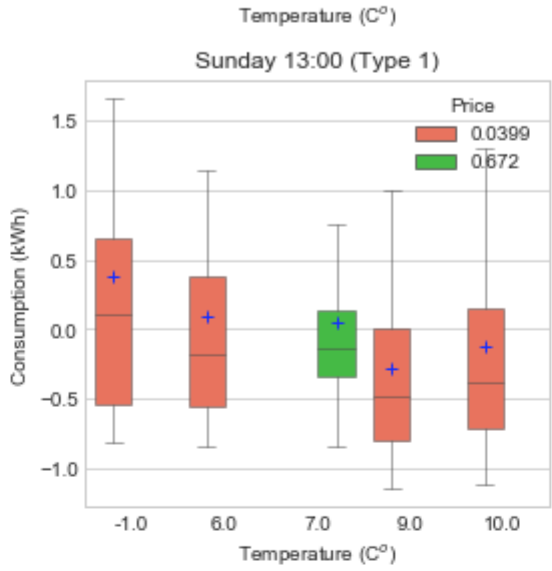

0.672
Consumption (kWh)
Temperature (C°)
-1.0
5.0
9.0
10.0

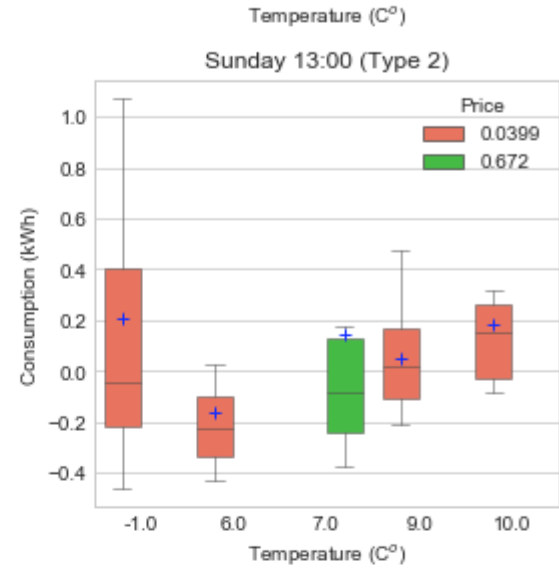

Sunday 12:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-1.0
6.0
9.0
10.0

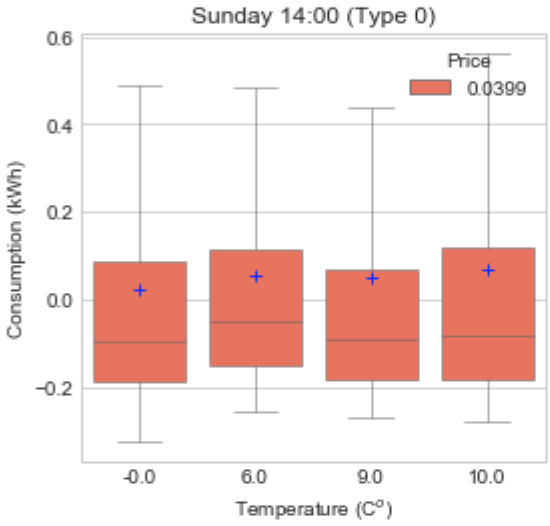

Sunday 12:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-1.0
6.0
9.0
10.0

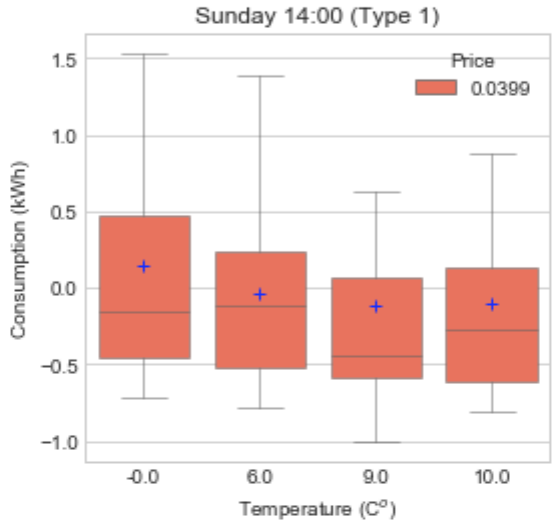

Sunday 12:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-1.0
6.0
9.0
10.0

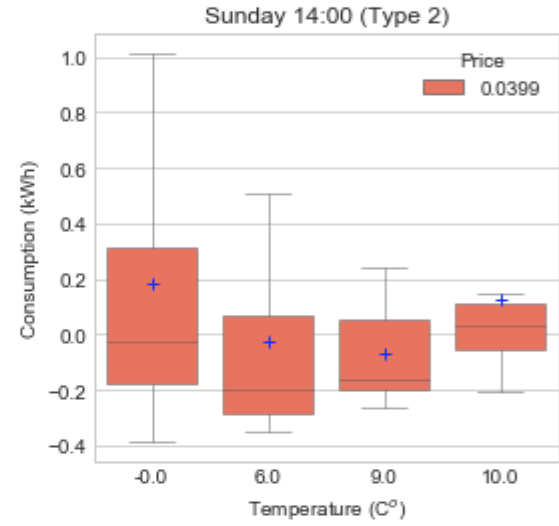

Sunday 13:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-1.0
6.0
7.0
9.0
10.0

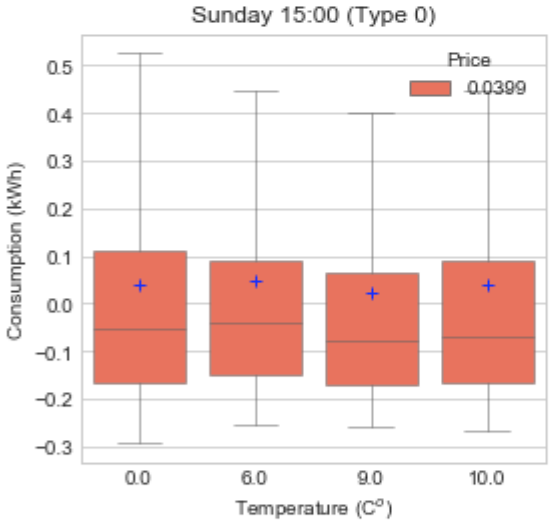

Sunday 13:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-1.0
6.0
7.0
9.0
10.0

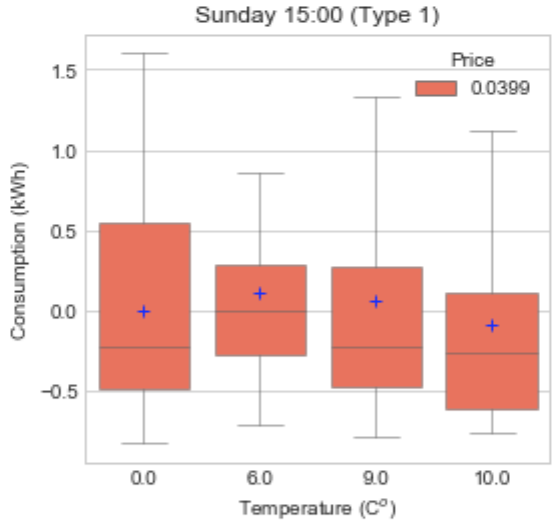

Sunday 13:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)
-1.0
6.0
7.0
9.0
10.0

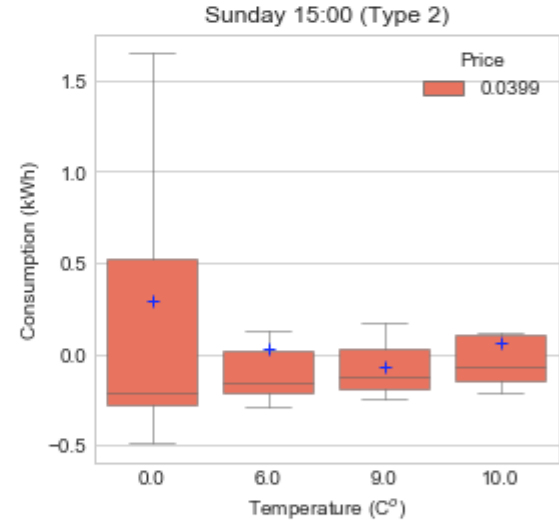

Sunday 14:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-0.0
6.0
9.0
10.0

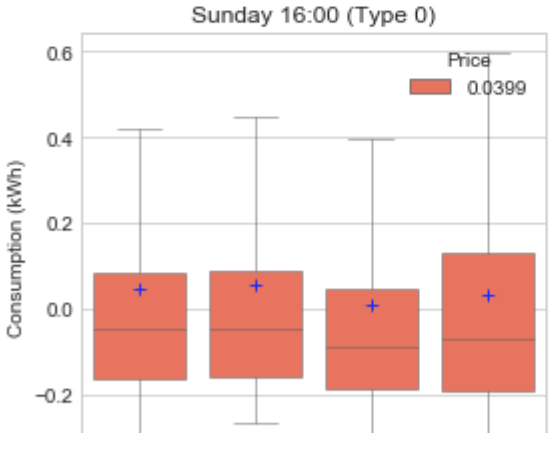

Sunday 14:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-0.0
6.0
9.0
10.0

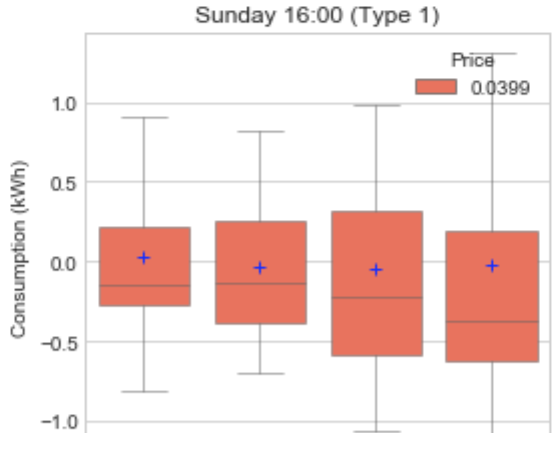

Sunday 14:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
-0.0
6.0
9.0
10.0

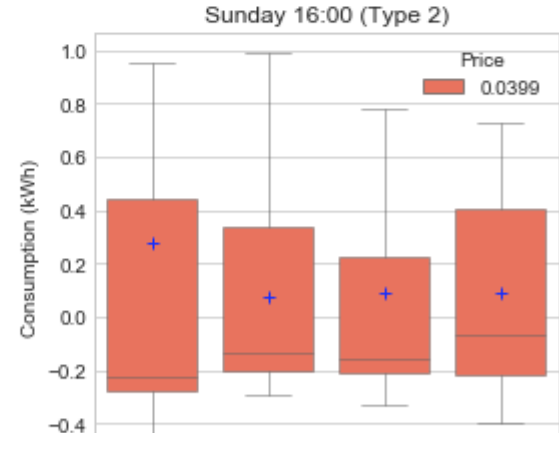

Sunday 15:00 (Type 0)
Price
0.0399
Consumption (kWh)
Temperature (C°)
0.0
6.0
9.0
10.0

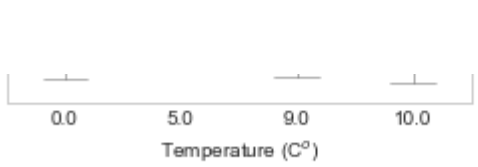

Sunday 15:00 (Type 1)
Price
0.0399
Consumption (kWh)
Temperature (C°)
0.0
6.0
9.0
10.0

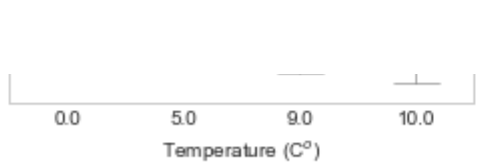

Sunday 15:00 (Type 2)
Price
0.0399
Consumption (kWh)
Temperature (C°)
0.0
6.0
9.0
10.0

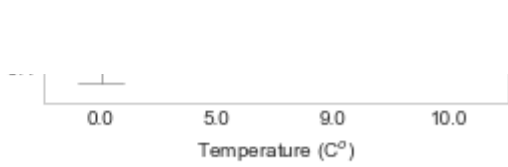

Sunday 16:00 (Type 0)
Price
0.0399
Consumption (kWh)

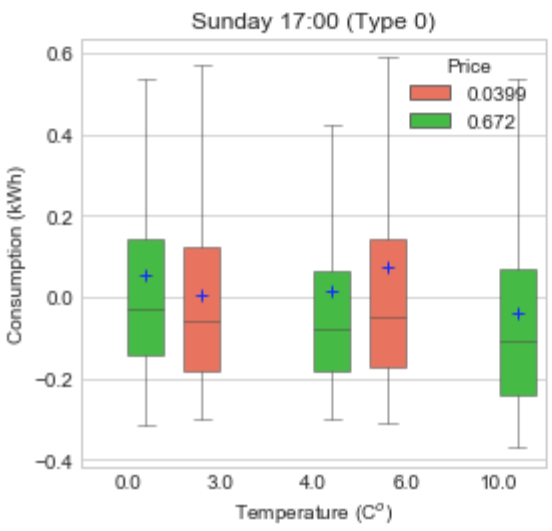

Sunday 16:00 (Type 1)
Price
0.0399
Consumption (kWh)

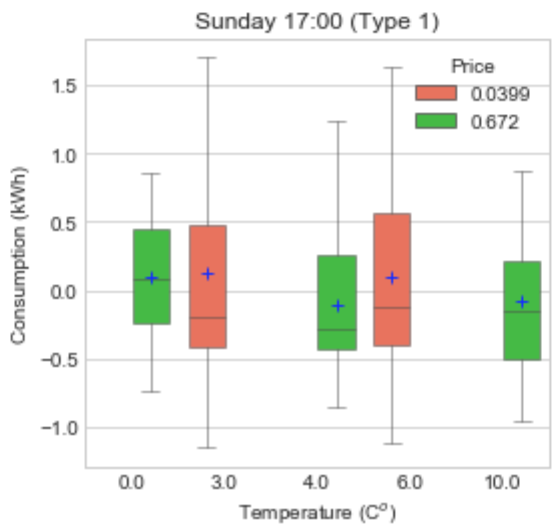

Sunday 16:00 (Type 2)
Price
0.0399
Consumption (kWh)

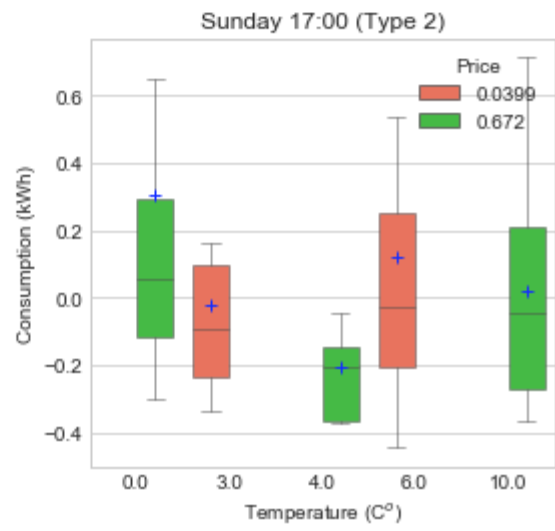

0.0
5.0
9.0
10.0
Temperature (C°)

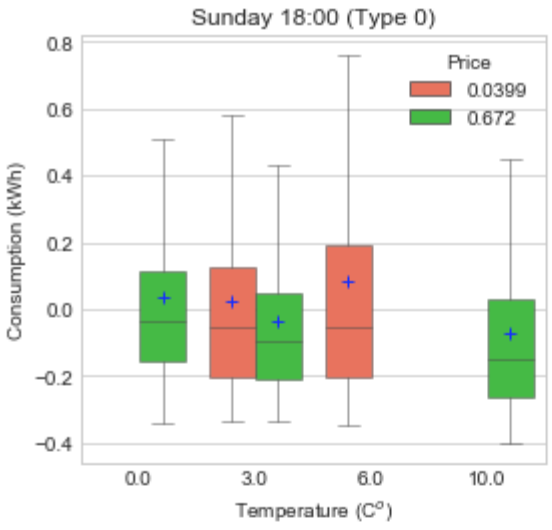

0.0
5.0
9.0
10.0
Temperature (C°)

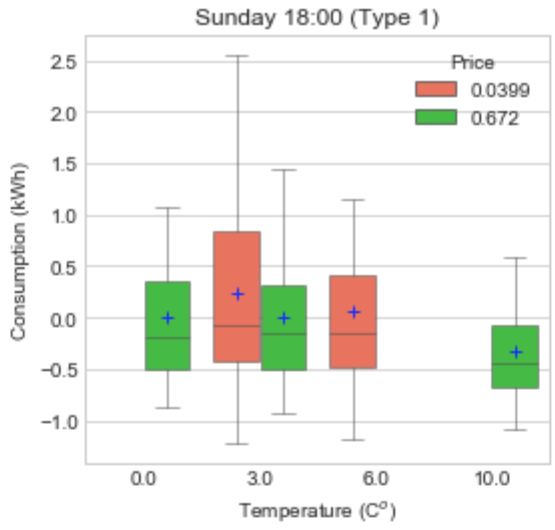

0.0
5.0
9.0
10.0
Temperature (C°)

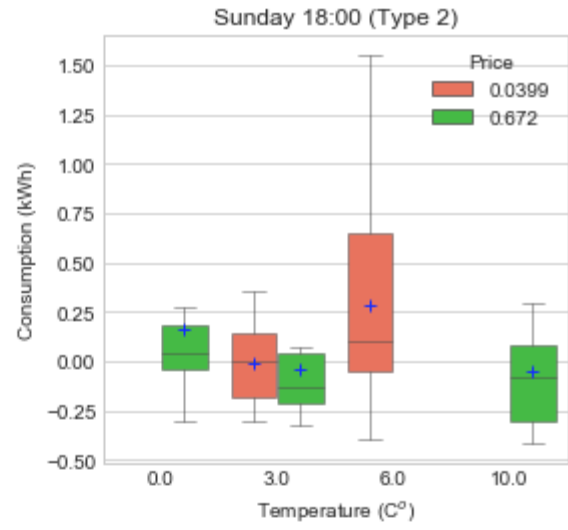

Sunday 17:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
3.0
4.0
6.0
10.0
Temperature (C°)

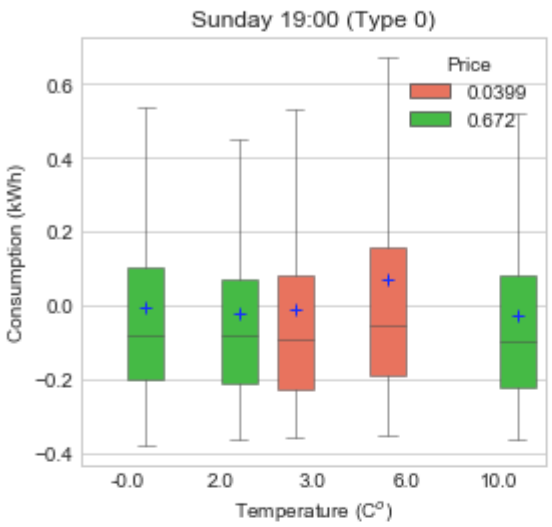

Sunday 17:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
3.0
4.0
6.0
10.0
Temperature (C°)

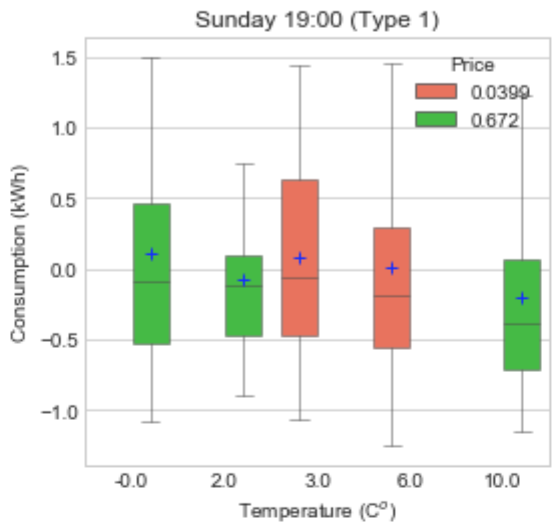

Sunday 17:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
3.0
4.0
6.0
10.0
Temperature (C°)

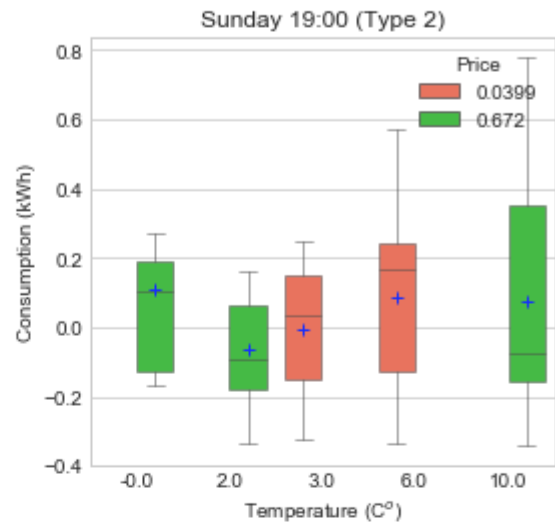

Sunday 18:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
3.0
6.0
10.0
Temperature (C°)

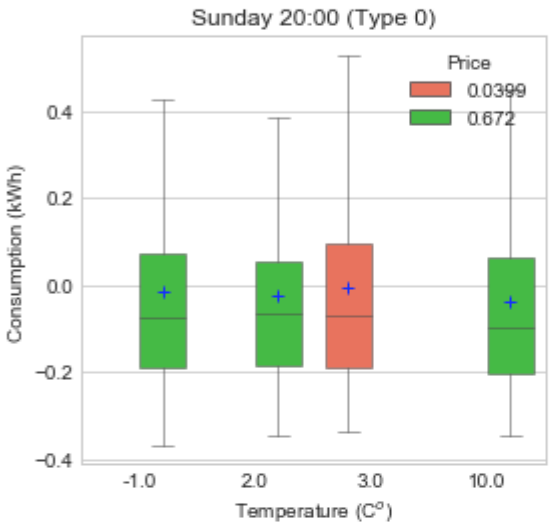

Sunday 18:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
3.0
6.0
10.0
Temperature (C°)

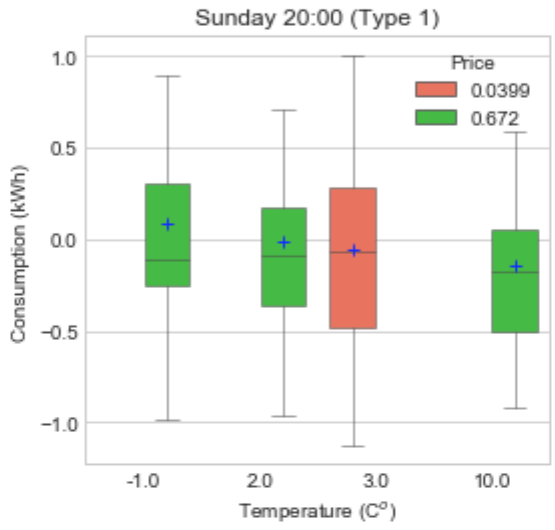

Sunday 18:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
0.0
3.0
6.0
10.0
Temperature (C°)

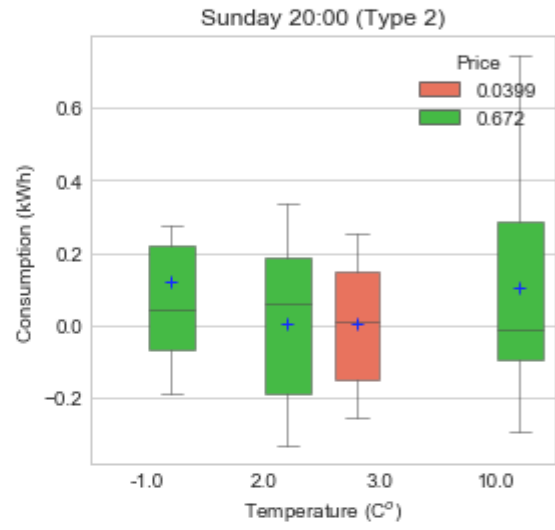

Sunday 19:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-0.0
2.0
3.0
6.0
10.0
Temperature (C°)

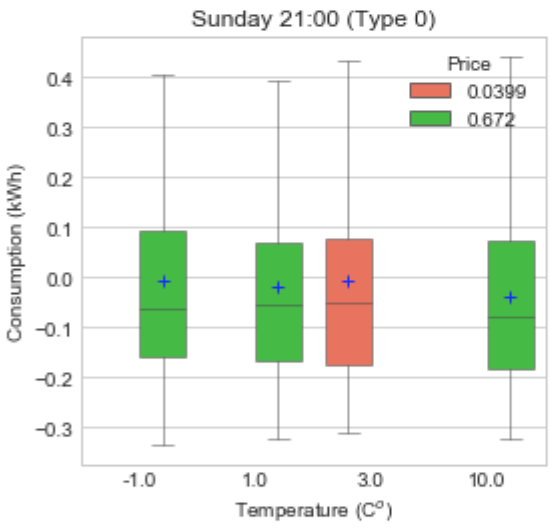

Sunday 19:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-0.0
2.0
3.0
6.0
10.0
Temperature (C°)

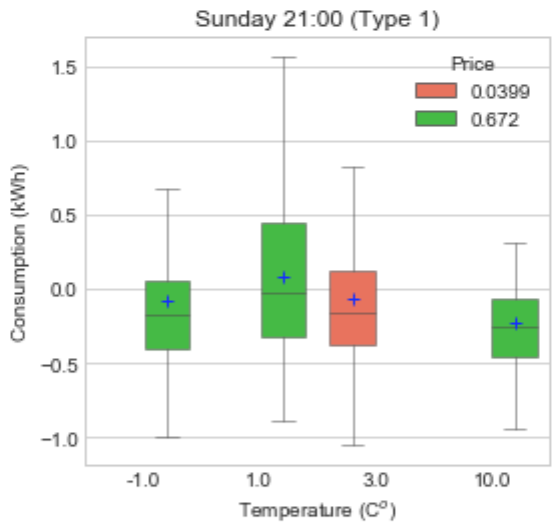

Sunday 19:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-0.0
2.0
3.0
6.0
10.0
Temperature (C°)

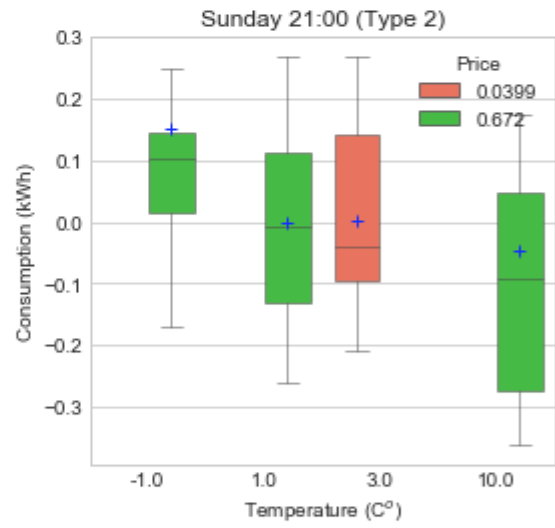

Sunday 20:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
-1.0
2.0
3.0
10.0
Temperature (C°)

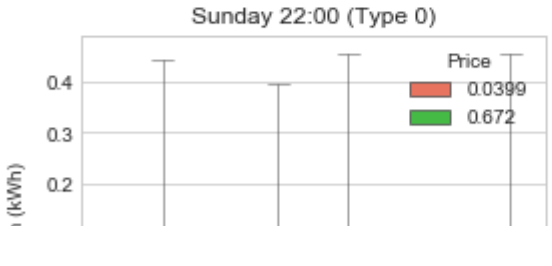

Sunday 20:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
2.0
3.0
10.0
Temperature (C°)

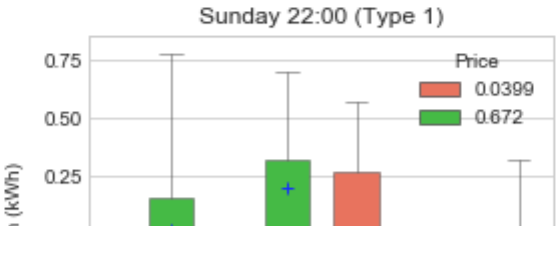

Sunday 20:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
-1.0
2.0
3.0
10.0
Temperature (C°)

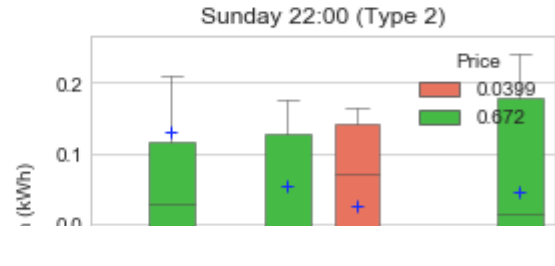

Sunday 21:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-1.0
1.0
3.0
10.0
Temperature (C°)

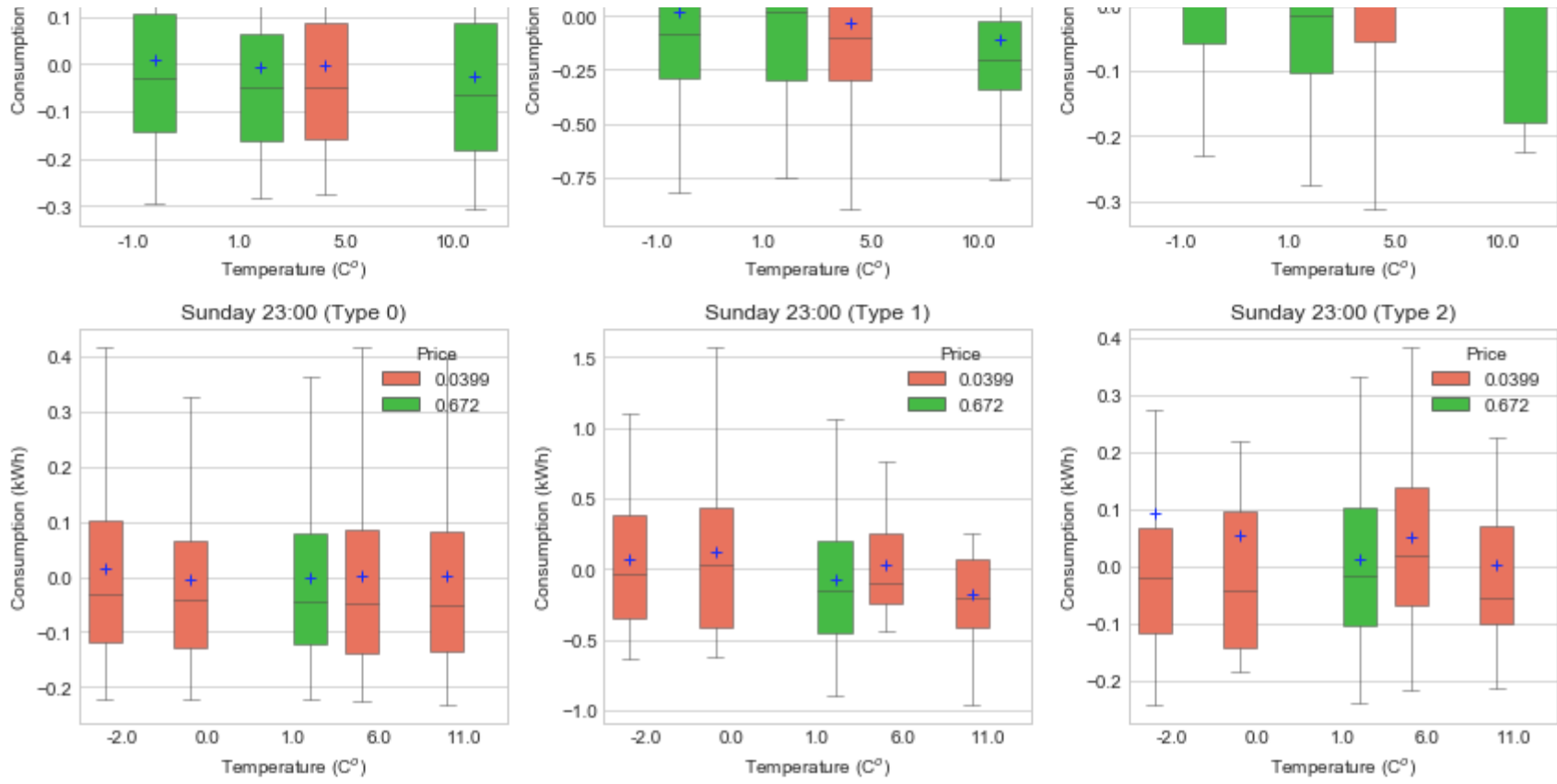

Sunday 21:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
1.0
3.0
10.0
Temperature (C°)


Sunday 21:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
-1.0
1.0
3.0
10.0
Temperature (C°)


Sunday 22:00 (Type 0)
Price
0.0399
0.672
0.4
0.3
0.2


Sunday 22:00 (Type 1)
Price
0.0399
0.672
0.75
0.50
0.25

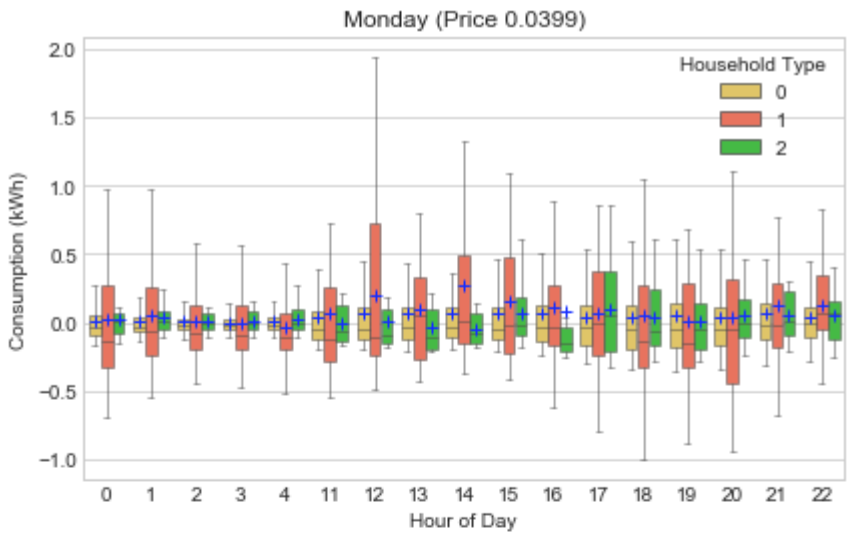

Sunday 22:00 (Type 2)
Price
0.0399
0.672
0.2
0.1

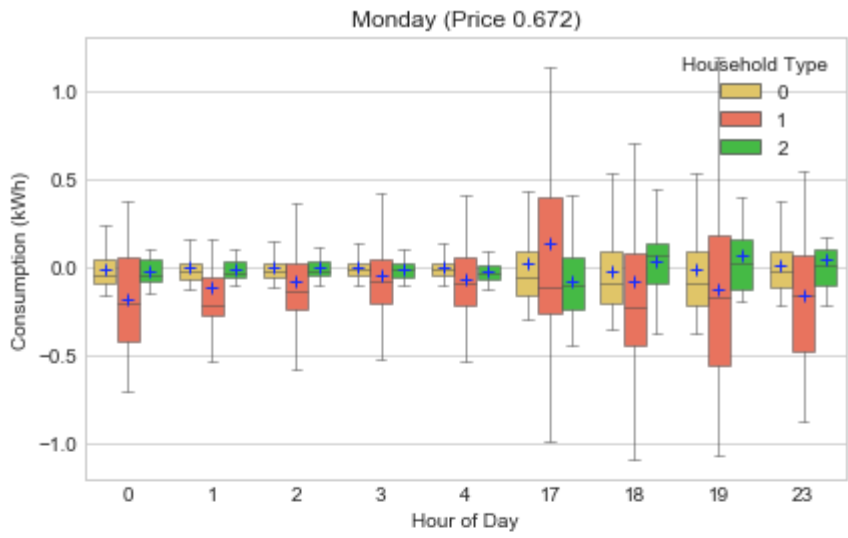


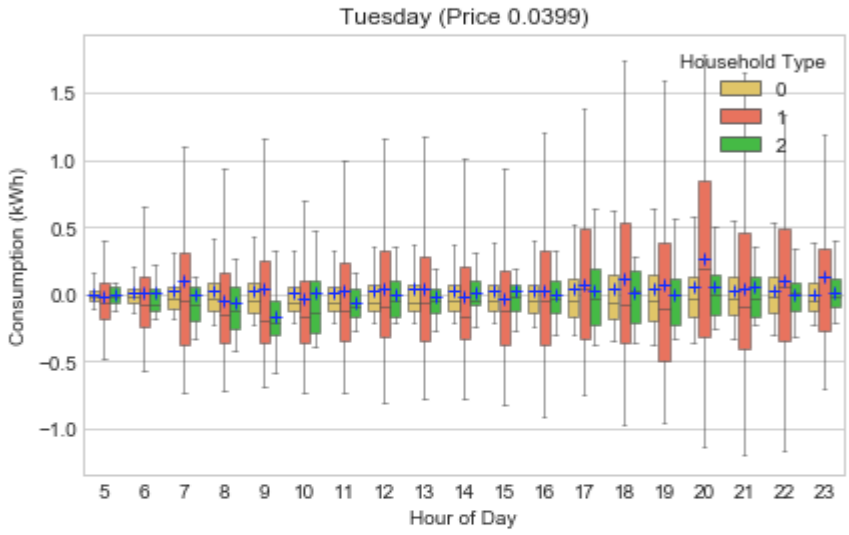
type2 may not be a good candidate for DR; type0 and type1 are good candidates but the program design must pay attention to the temperature impact on the demand responsiveness, we can expect with the same temperature, lower price would increase the consumption, higher price would decrease consumption, but this may not be true in different temperature situations. Conclusion: demand response plan must consider temperature impact, i.e., differentiated based on temperature situations instead of use a unified demand response plan for all temperatures.

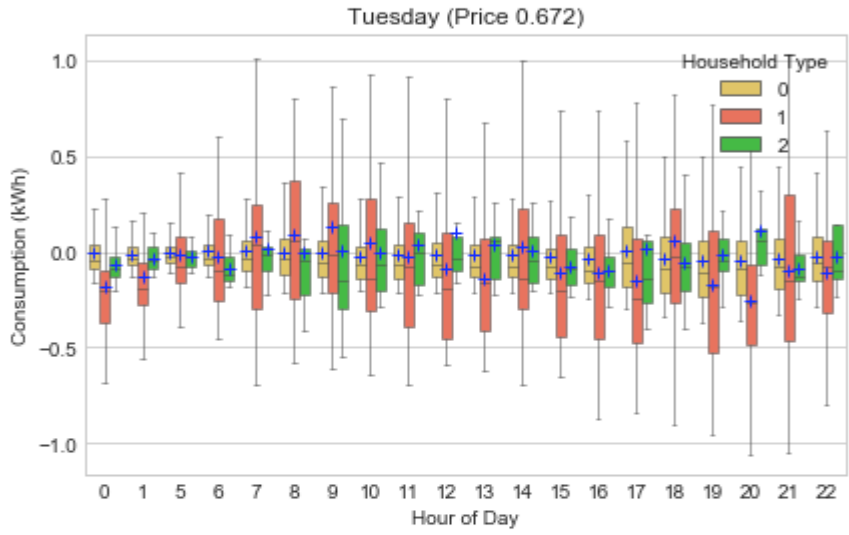
b) Day of week - hour : prices

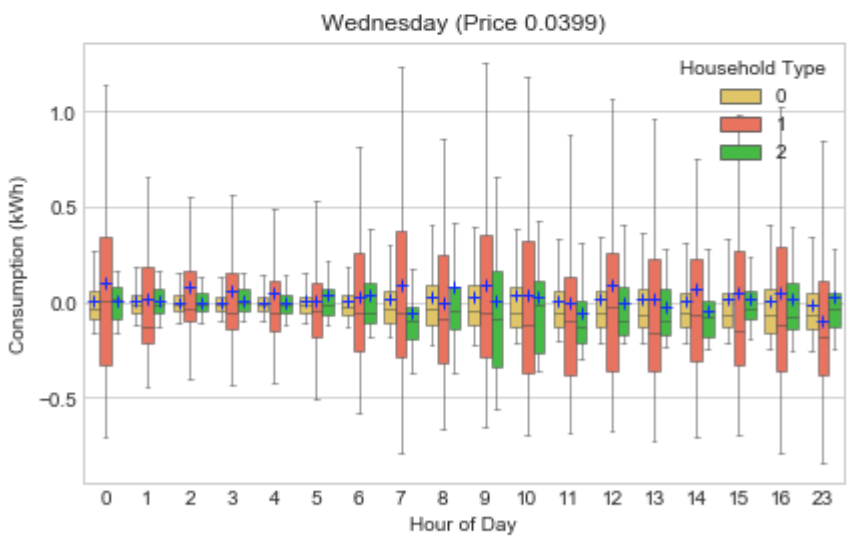
In [120]:
```
# hourly price responsiveness over days of week and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for p in range(2): # two price levels
    for day_of_week in range(7):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * day_of_week + (p + 1)))
        df_day = df_all_res_long[(df_all_res_long['Day of week'] == day_of_week) & (df_all_res_long['Price']
== prices[p])]
        if df_day.shape[0] >= 1:
            sns.boxplot(x="Hour of day", y="Consumption", hue="Household type", data=df_day, ax=ax_Ntou[-1],
palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","mar
kerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
            ax_Ntou[-1].set_title(weeks[day_of_week]  + ' (Price ' + str(prices[p]) + ')' )
            ax_Ntou[-1].set_xlabel(r'Hour of Day')
            ax_Ntou[-1].set_ylabel('Consumption (kWh)')
            l = ax_Ntou[-1].legend()
            l.set_title('Household Type')
            for i,artist in enumerate(ax_Ntou[-1].artists):
                artist.set_linewidth(0.5)
                # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                # Loop over them here, and use the same colour as above
                for j in range(i*6,i*6+6):
                    line = ax_Ntou[-1].lines[j]
                    line.set_linewidth(0.5)
            ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

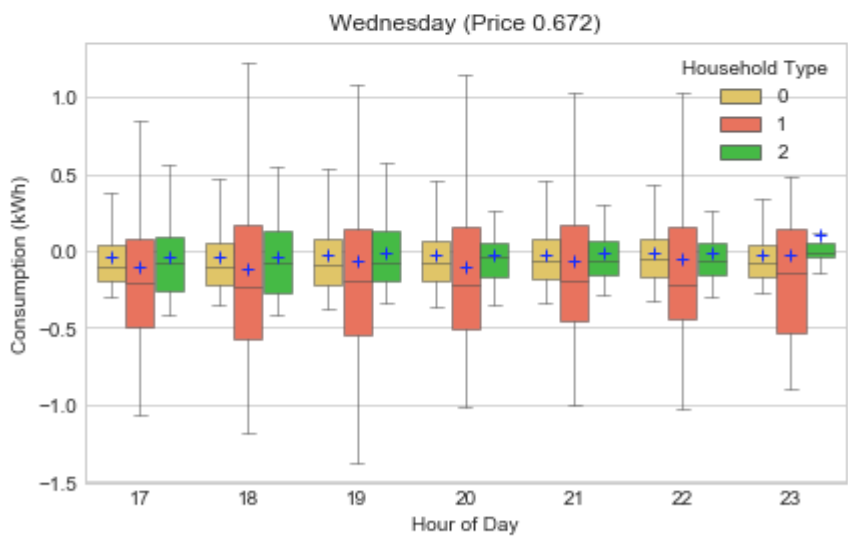

Monday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

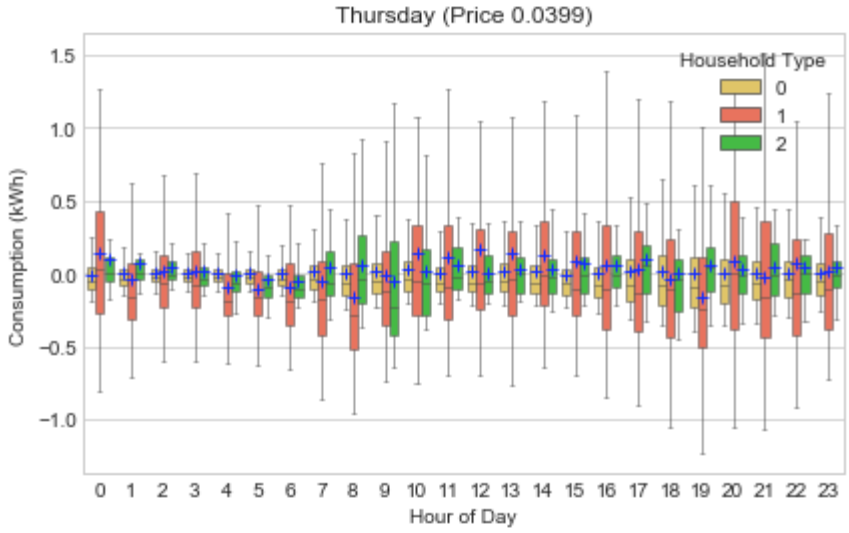

Monday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

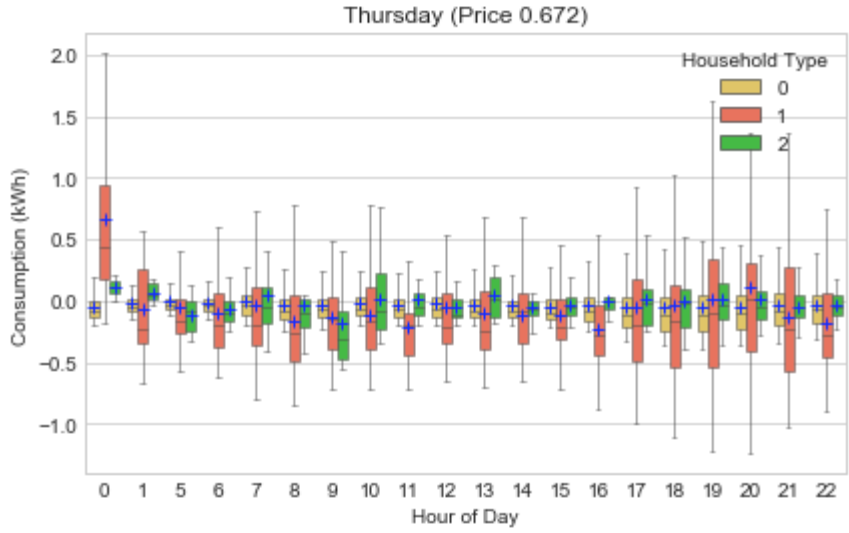

Tuesday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

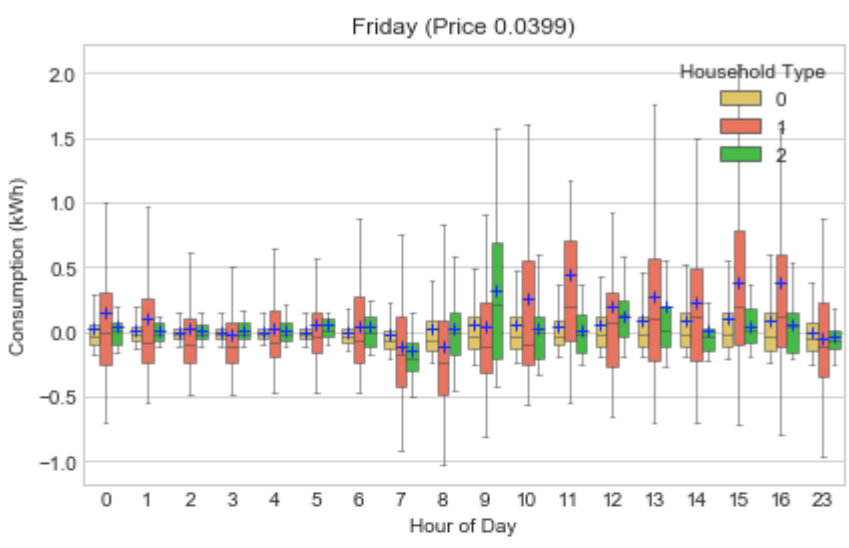

Tuesday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

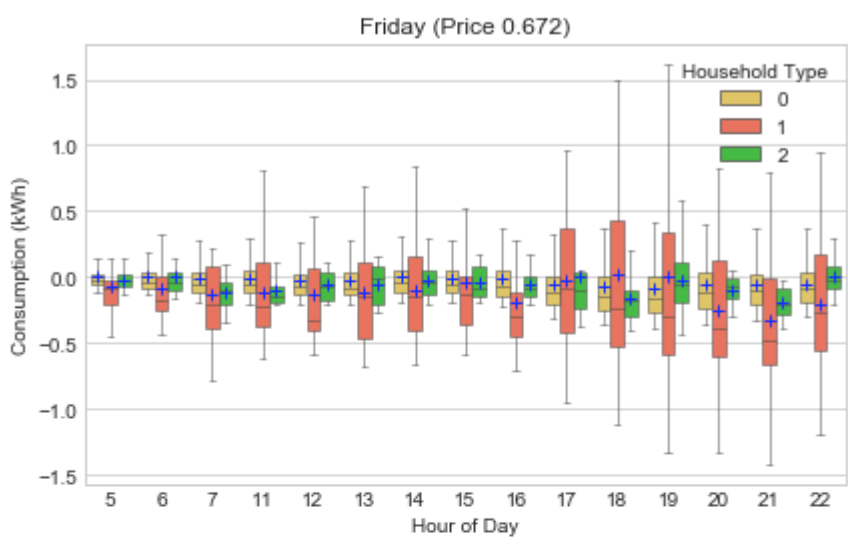

Wednesday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

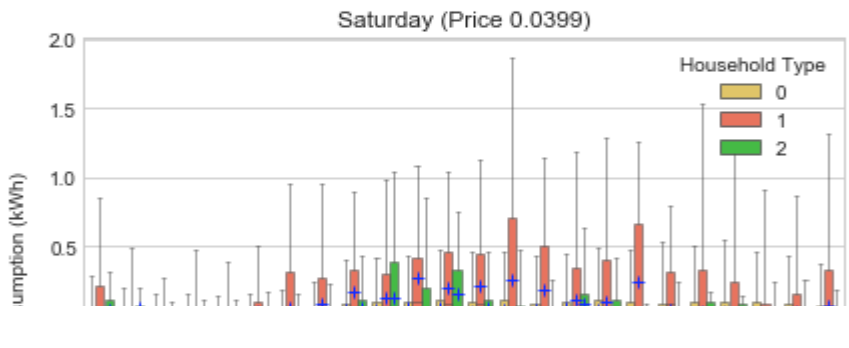

Wednesday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

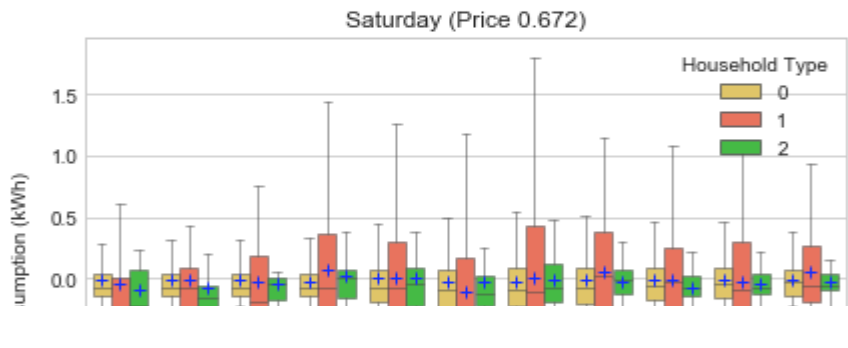

Thursday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

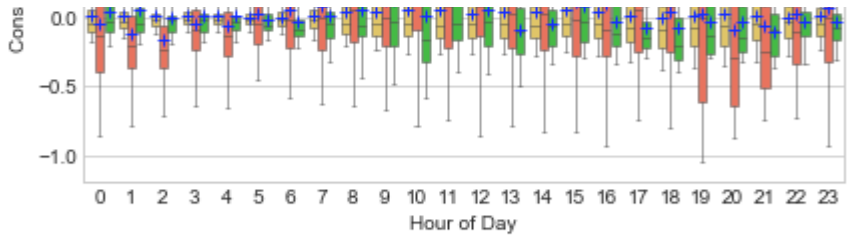

Thursday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

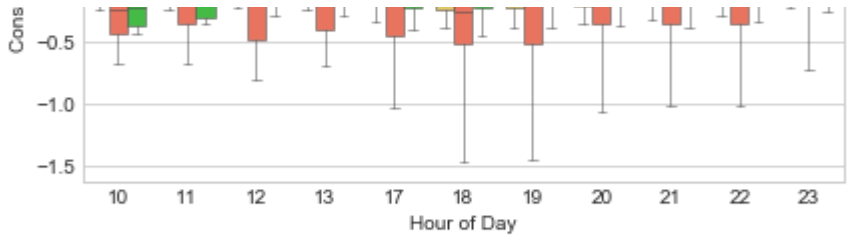

Friday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

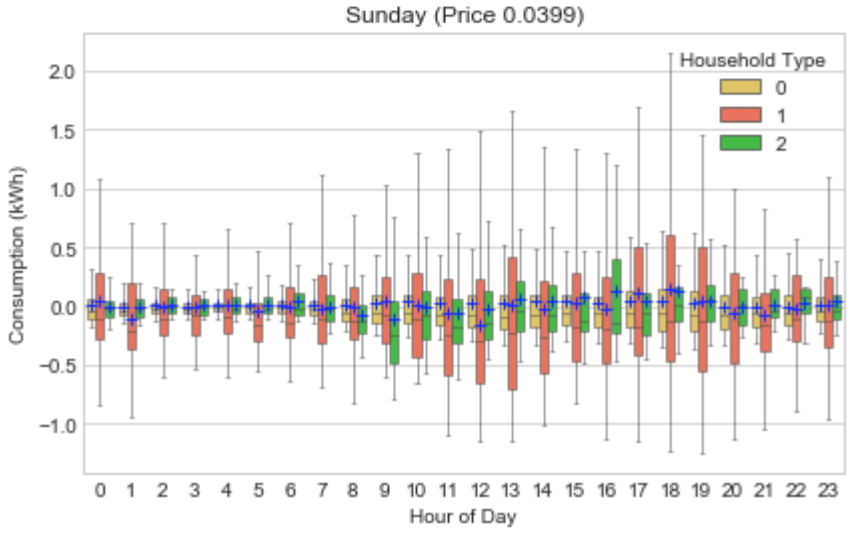

Friday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
Hour of Day

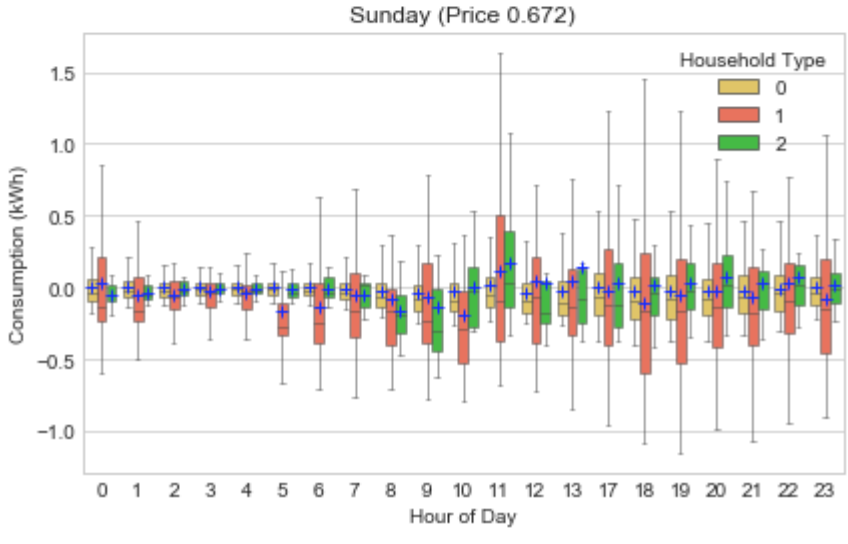

Saturday (Price 0.0399)
Household Type
0
1
2


Saturday (Price 0.672)
Household Type
0
1
2

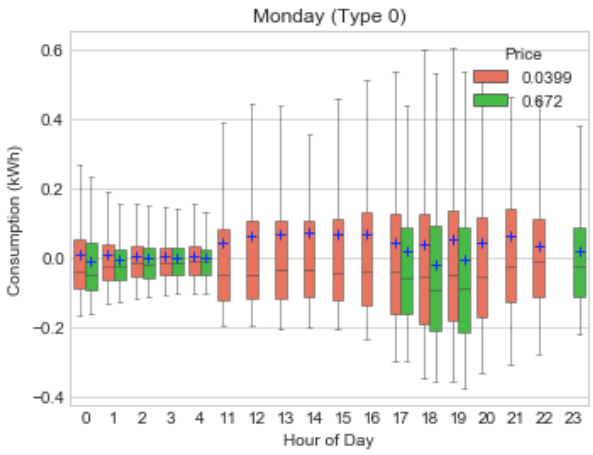

Cons
0.0
−0.5
−1.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day

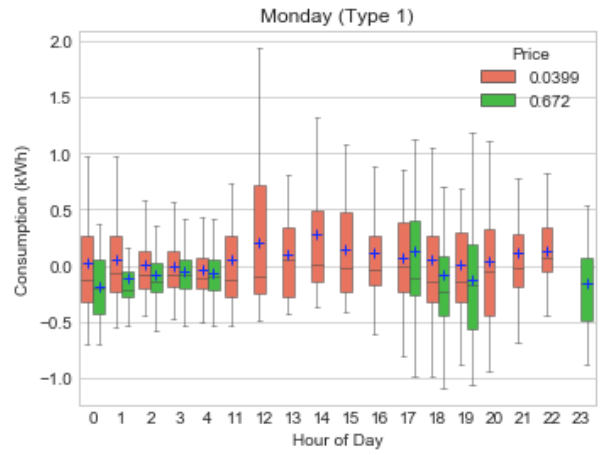

Cons
−0.5
−1.0
−1.5
10 11 12 13 17 18 19 20 21 22 23
Hour of Day

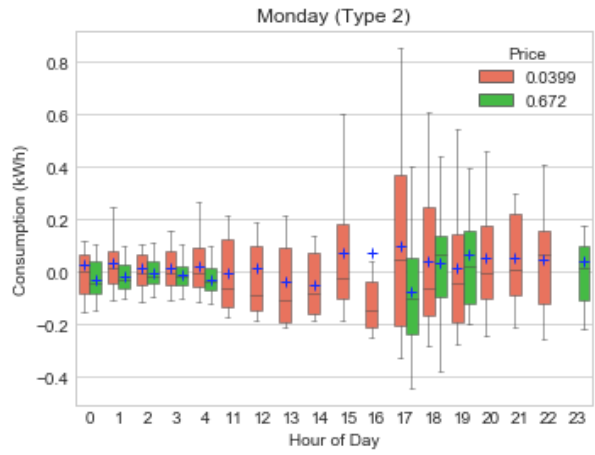

Sunday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day

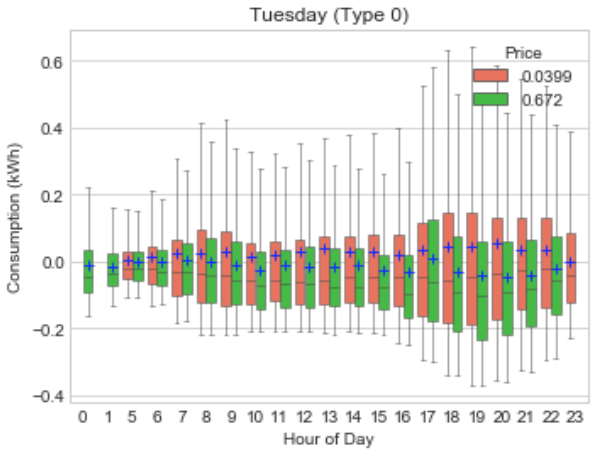

Sunday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 17 18 19 20 21 22 23
Hour of Day

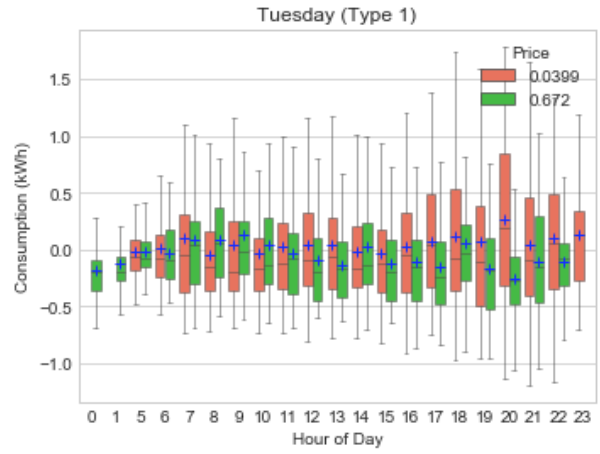
In [138]:
```
# price comparison
# hourly price responsiveness over days of week and prices
fig_all = plt.figure(figsize = (15,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for g in range(3): # three types
    for day_of_week in range(7):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3 * day_of_week + (g + 1)))
        df_day = df_all_res_long[(df_all_res_long['Day of week'] == day_of_week) & (df_all_res_long['Household type'] == g)]
        if df_day.shape[0] >= 1:
            sns.boxplot(x="Hour of day", y="Consumption", hue="Price", data=df_day, ax=ax_Ntou[-1], palette=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
            ax_Ntou[-1].set_title(weeks[day_of_week]  + ' (Type ' + str(g) + ')' )
            ax_Ntou[-1].set_xlabel(r'Hour of Day')
            ax_Ntou[-1].set_ylabel('Consumption (kWh)')
            l = ax_Ntou[-1].legend()
            l.set_title('Price')
            for i,artist in enumerate(ax_Ntou[-1].artists):
                artist.set_linewidth(0.5)
                # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                # Loop over them here, and use the same colour as above
                for j in range(i*6,i*6+6):
                    line = ax_Ntou[-1].lines[j]
                    line.set_linewidth(0.5)
            ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

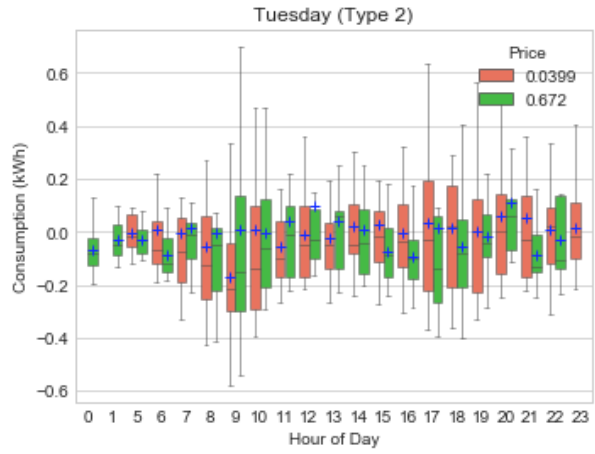

Monday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

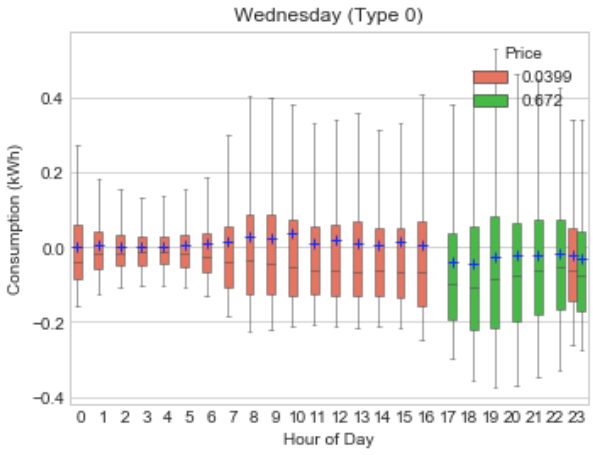

Monday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

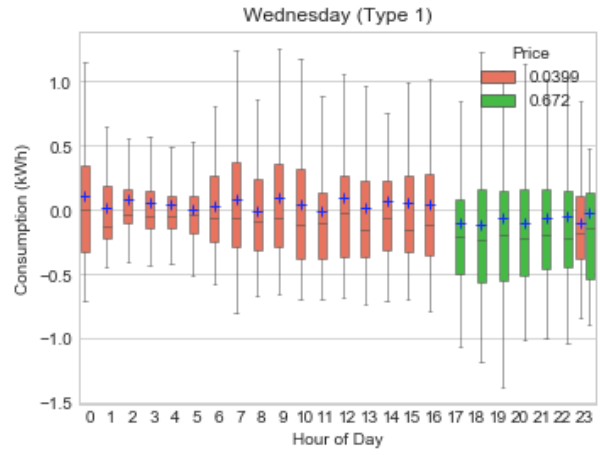

Monday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

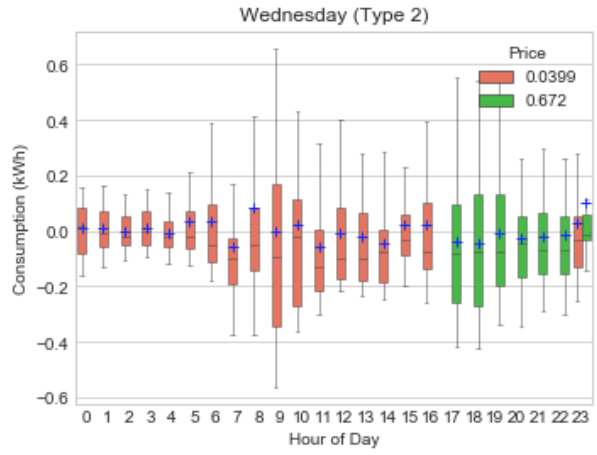

Tuesday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

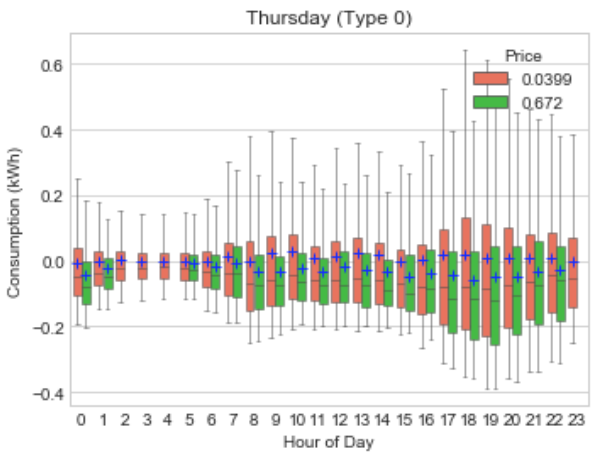

Tuesday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

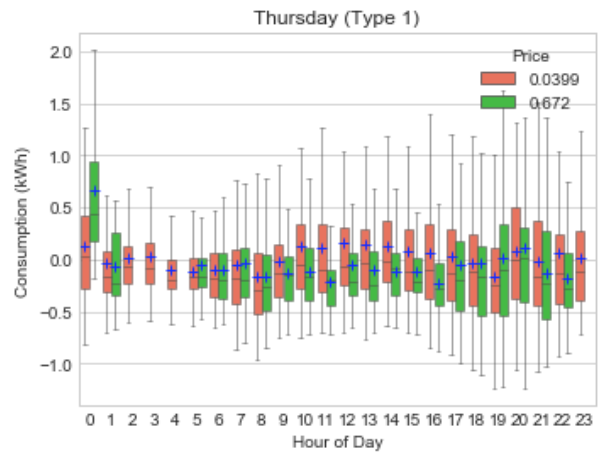

Tuesday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

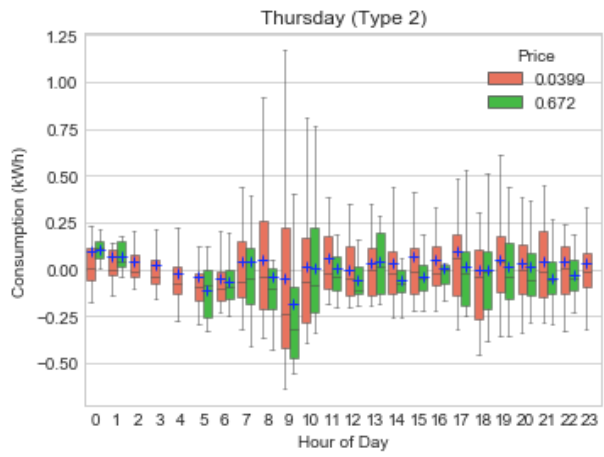

Wednesday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

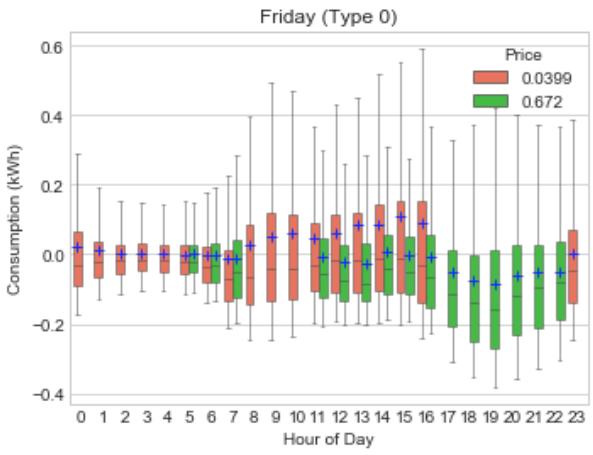

Wednesday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

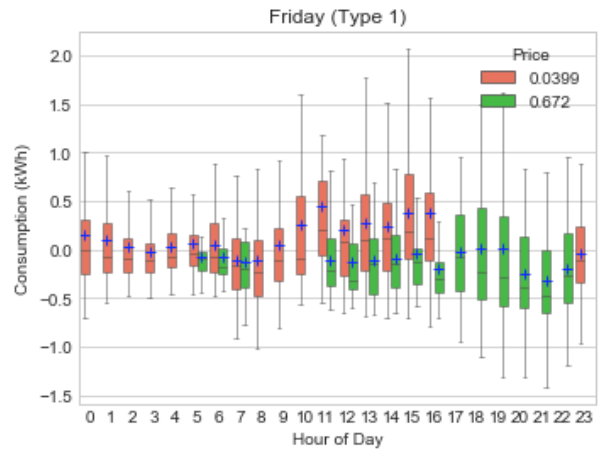

Wednesday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

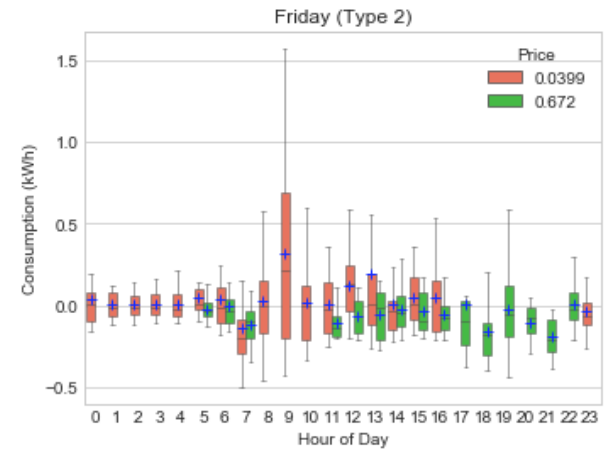

Thursday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

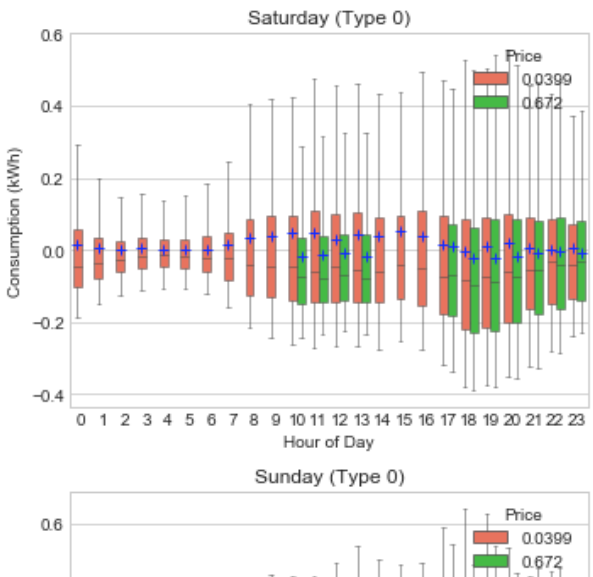

Thursday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

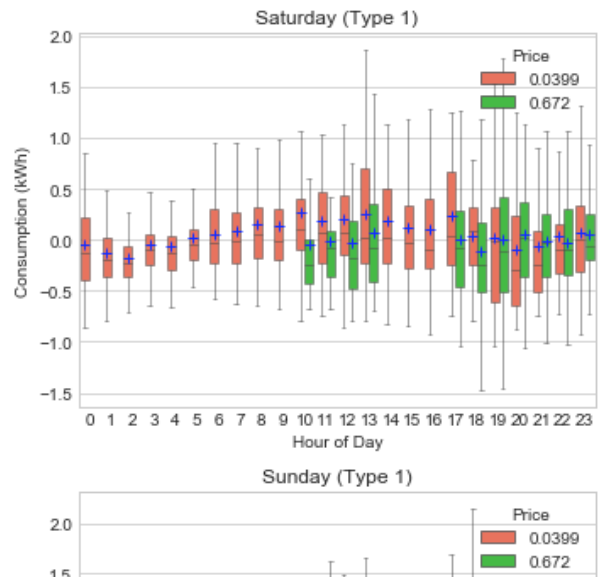

Thursday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

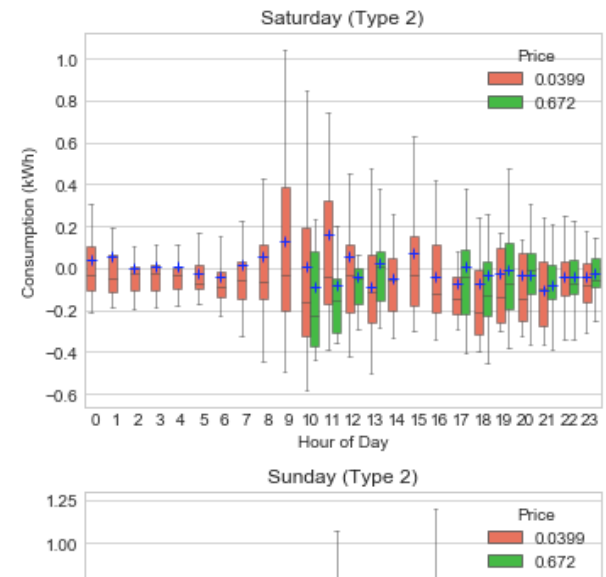

Friday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

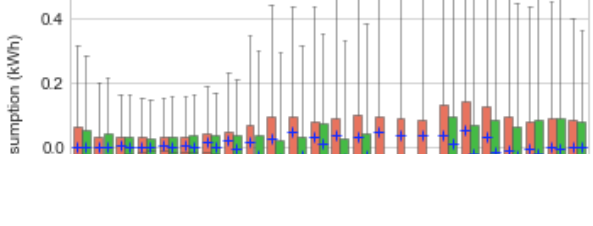

Friday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

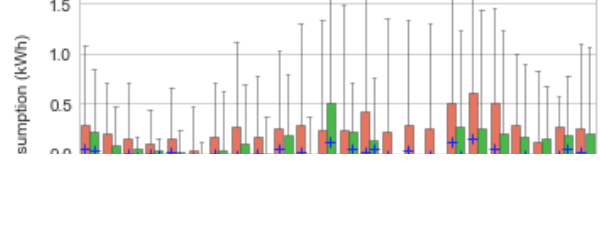

Friday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

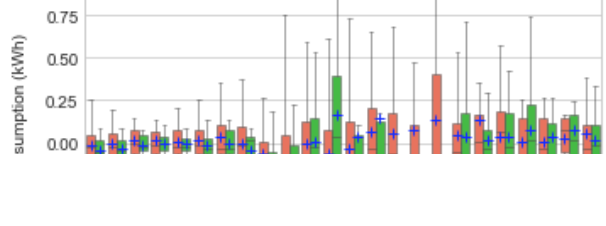

Saturday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day

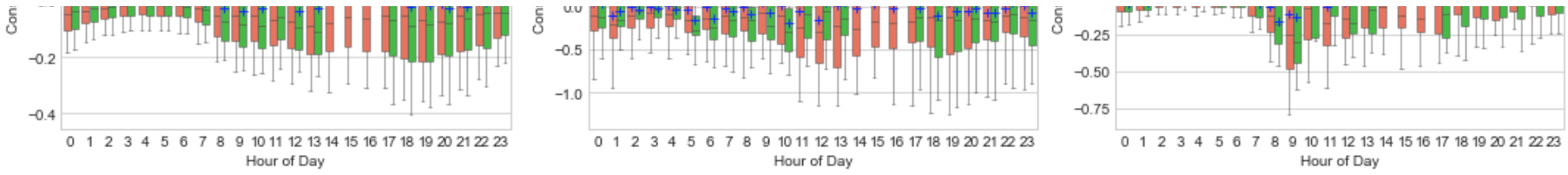

Saturday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day


Saturday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Hour of Day


Sunday (Type 0)
Price
0.0399
0.672
Hour of Day


Sunday (Type 1)
Price
0.0399
0.672
Hour of Day

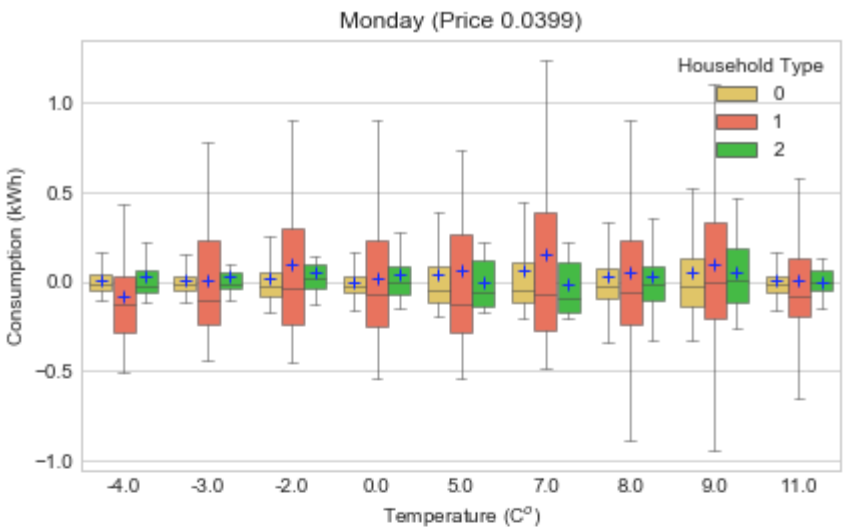

Sunday (Type 2)
Price
0.0399
0.672
Hour of Day

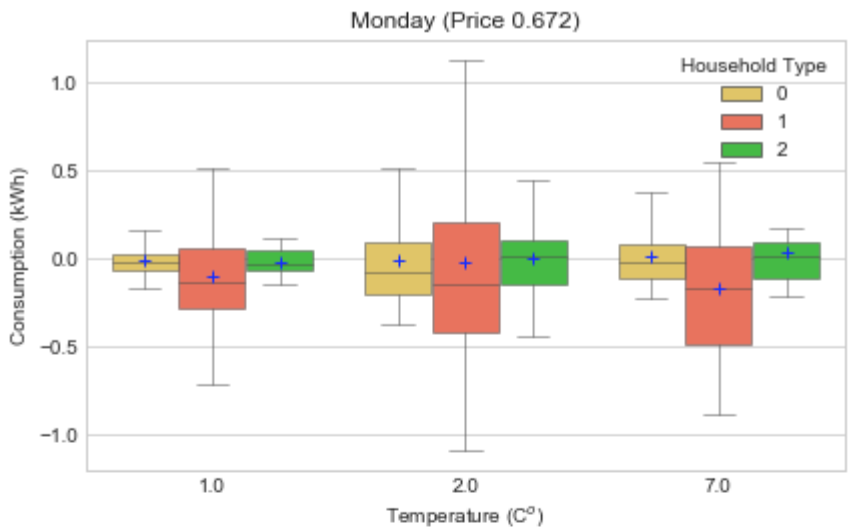


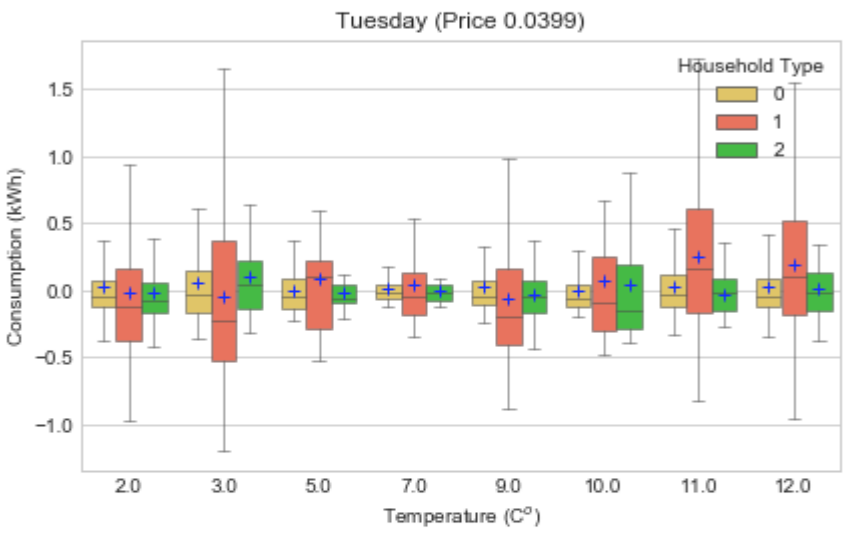
Friday night, Saturday night and Sunday night show an intersting pattern: on Friday night, people's price responsiveness increases, but on Saturday night, the price responstiveness become negative to low price meaning people simply use less energy even though it's a cheaper electricity price time period; Last on Sunday night, the price responsiveness recover to the balanced point.

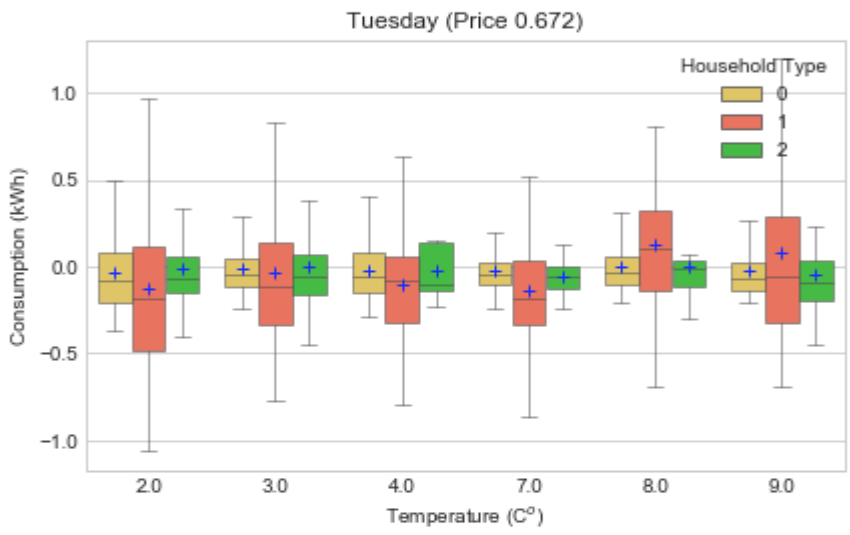
c) Day of week - temperature : prices

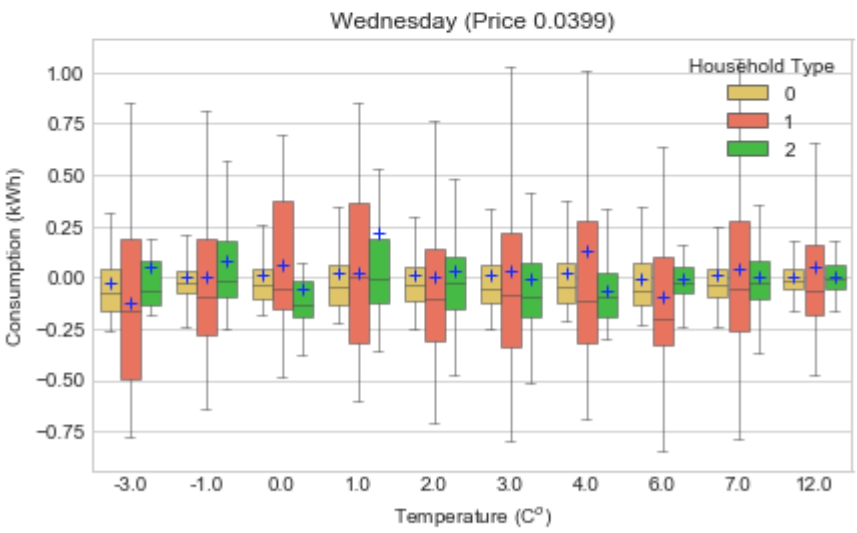
In [118]:
```
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for p in range(2): # two price levels
    for day_of_week in range(7):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * day_of_week + (p + 1)))
        df_day = df_all_res_long[(df_all_res_long['Day of week'] == day_of_week) & (df_all_res_long['Price']
== prices[p])]
        if df_day.shape[0] >= 1:
            sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_day, ax=ax_Ntou[-1], palett
e=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfac
ecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
            ax_Ntou[-1].set_title(weeks[day_of_week]  + ' (Price ' + str(prices[p]) + ')' )
            ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
            ax_Ntou[-1].set_ylabel('Consumption (kWh)')
            l = ax_Ntou[-1].legend()
            l.set_title('Household Type')
            for i,artist in enumerate(ax_Ntou[-1].artists):
                artist.set_linewidth(0.5)
                # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                # Loop over them here, and use the same colour as above
                for j in range(i*6,i*6+6):
                    line = ax_Ntou[-1].lines[j]
                    line.set_linewidth(0.5)
            ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

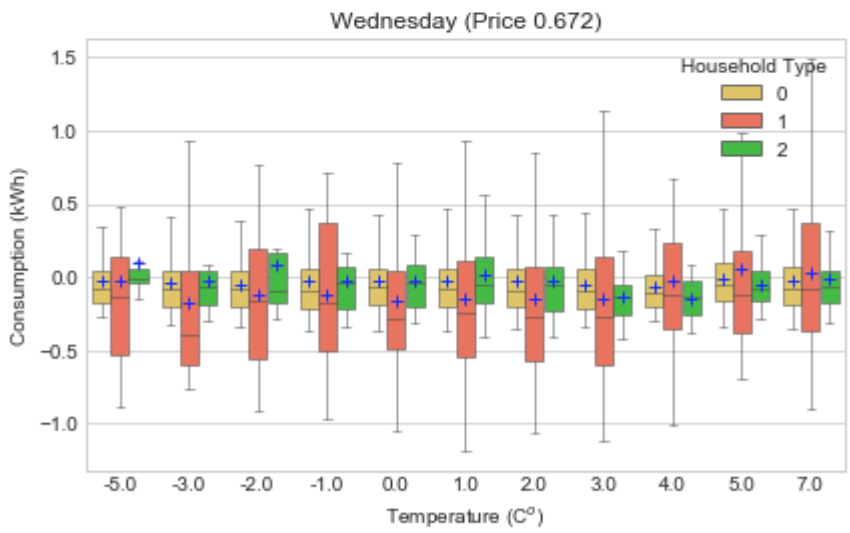

Monday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-4.0
-3.0
-2.0
0.0
5.0
7.0
8.0
9.0
11.0
Temperature (C°)

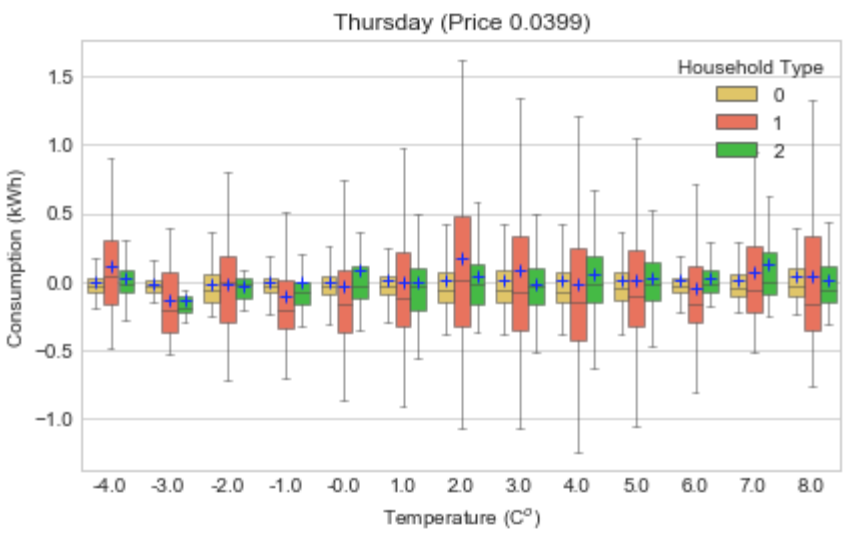

Monday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
1.0
2.0
7.0
Temperature (C°)

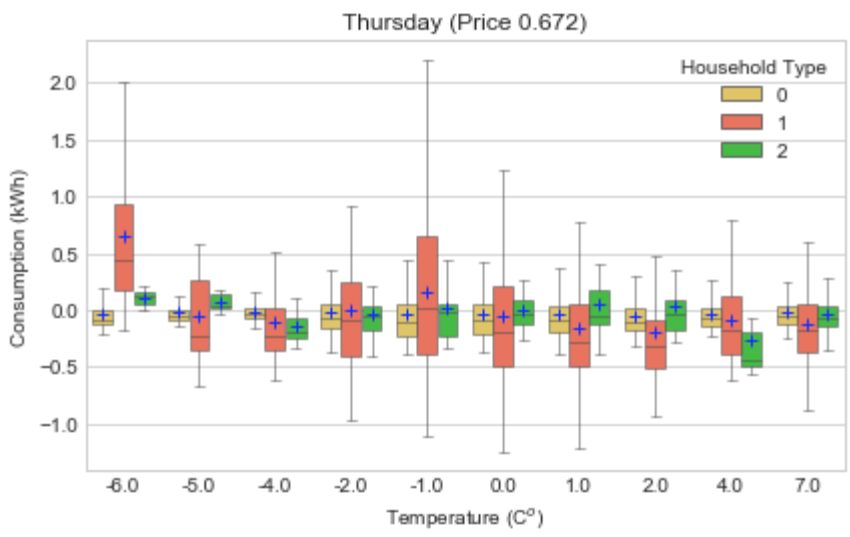

Tuesday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
2.0
3.0
5.0
7.0
9.0
10.0
11.0
12.0
Temperature (C°)

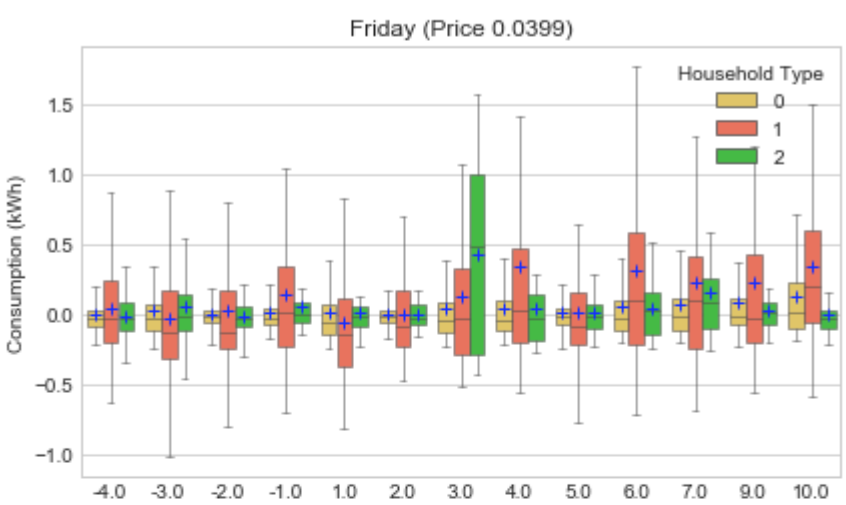

Tuesday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
2.0
3.0
4.0
7.0
8.0
9.0
Temperature (C°)

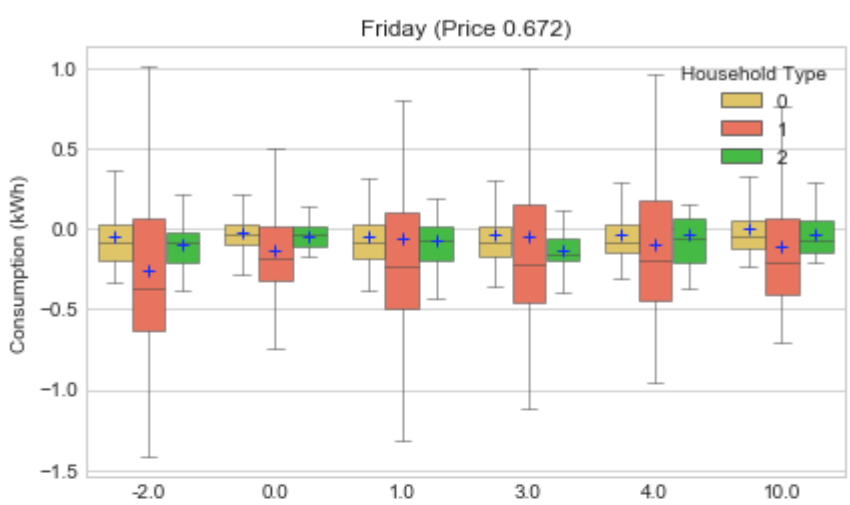

Wednesday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-3.0
-1.0
0.0
1.0
2.0
3.0
4.0
6.0
7.0
12.0
Temperature (C°)

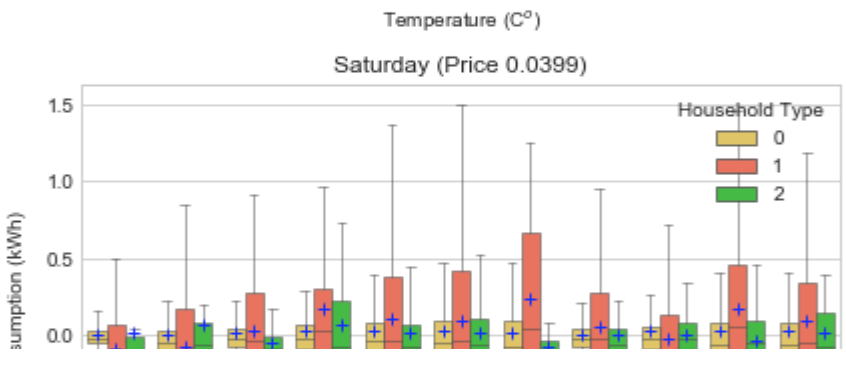

Wednesday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-5.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
7.0
Temperature (C°)

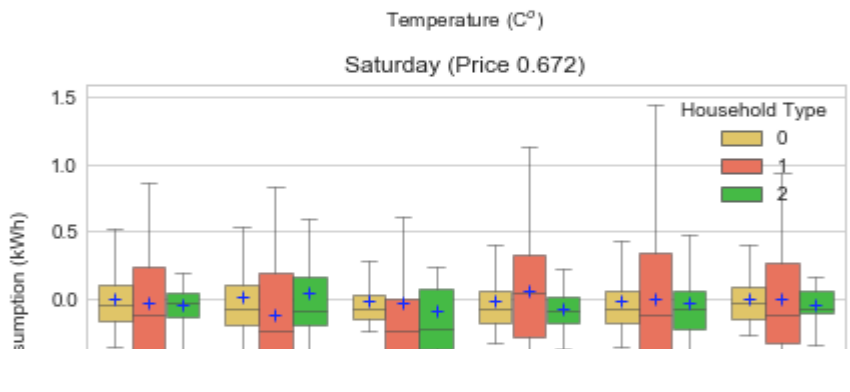

Thursday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-4.0
-3.0
-2.0
-1.0
-0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
Temperature (C°)

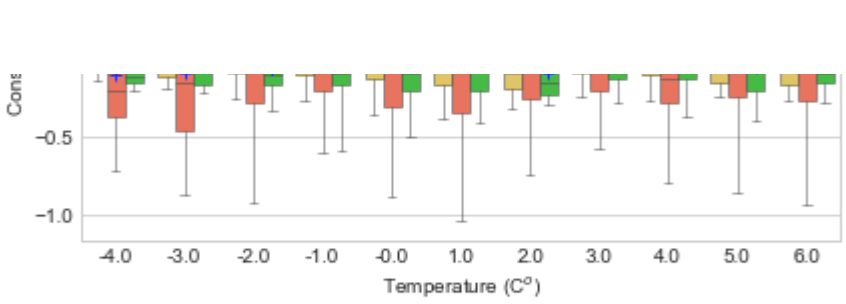

Thursday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
-6.0
-5.0
-4.0
-2.0
-1.0
0.0
1.0
2.0
4.0
7.0
Temperature (C°)

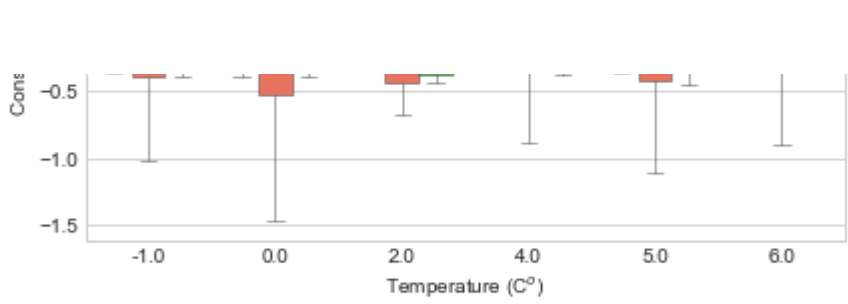

Friday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-4.0
-3.0
-2.0
-1.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
9.0
10.0
Temperature (C°)

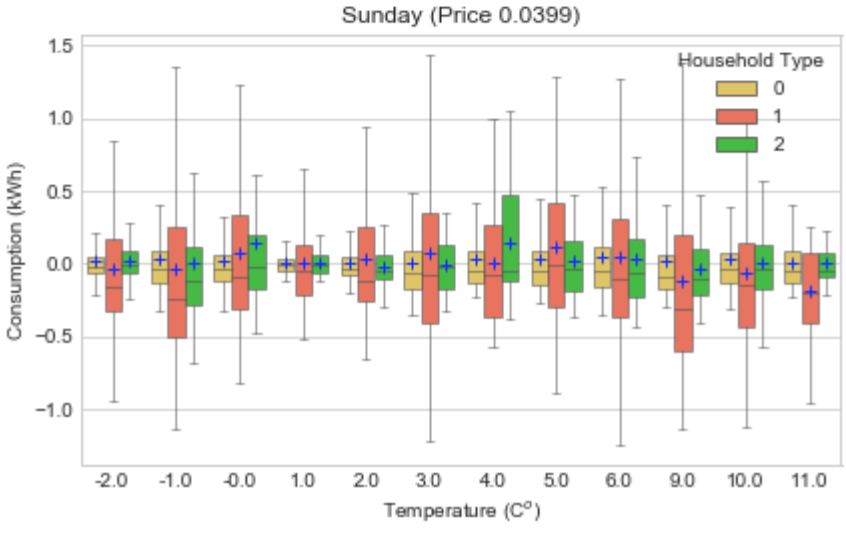

Friday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
−1.5
-2.0
0.0
1.0
3.0
4.0
10.0
Temperature (C°)

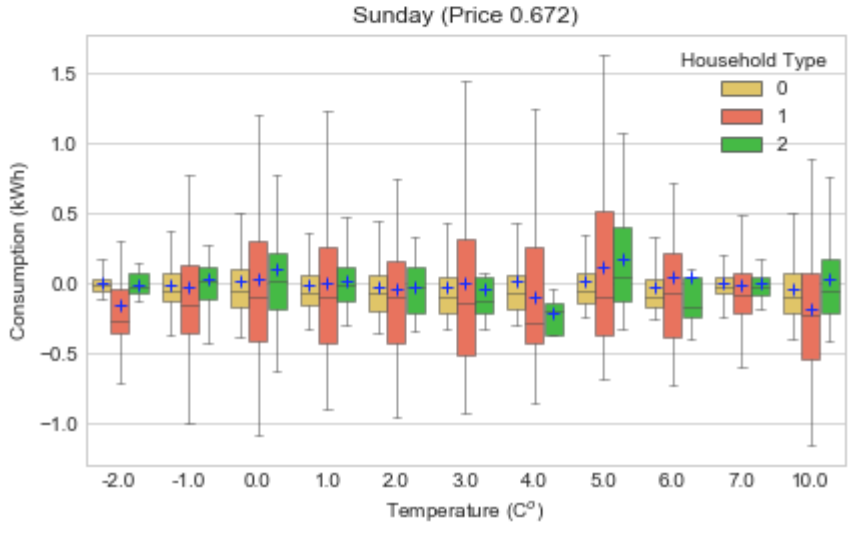

Saturday (Price 0.0399)
Household Type
0
1
2
1.5
1.0
0.5
0.0
Temperature (C°)


Saturday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
Temperature (C°)

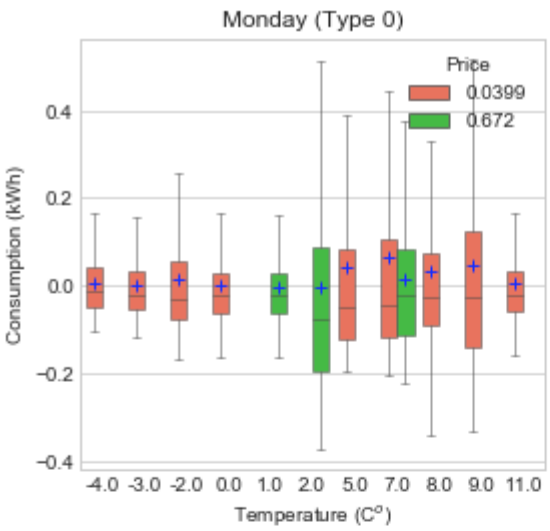

Cons
−0.5
−1.0
-4.0
-3.0
-2.0
-1.0
-0.0
1.0
2.0
3.0
4.0
5.0
6.0
Temperature (C°)

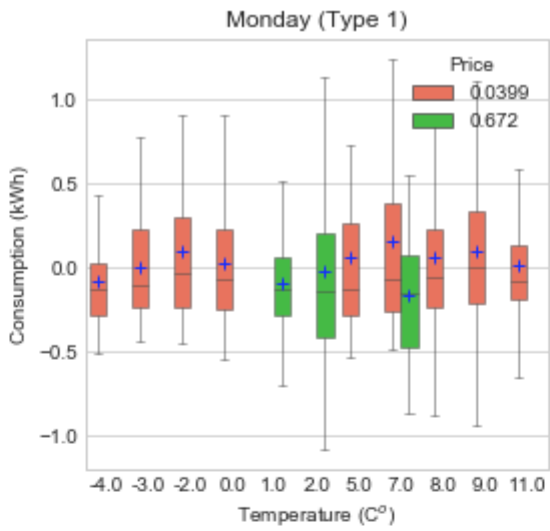

Cons
−0.5
−1.0
−1.5
-1.0
0.0
2.0
4.0
5.0
6.0
Temperature (C°)

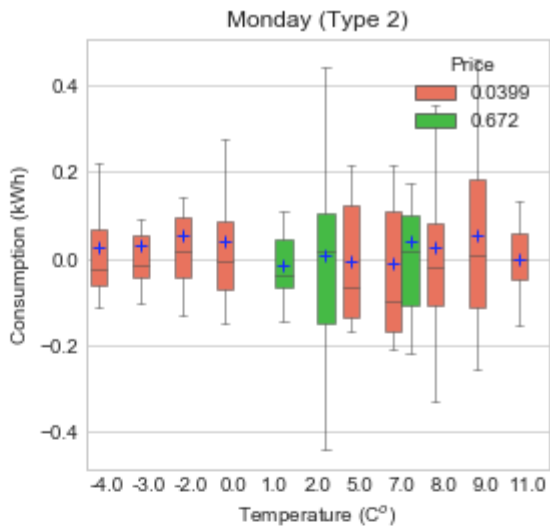

Sunday (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-2.0
-1.0
-0.0
1.0
2.0
3.0
4.0
5.0
6.0
9.0
10.0
11.0
Temperature (C°)

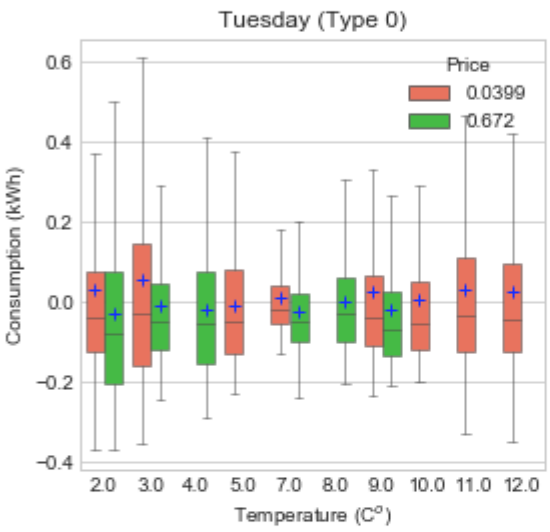

Sunday (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
10.0
Temperature (C°)

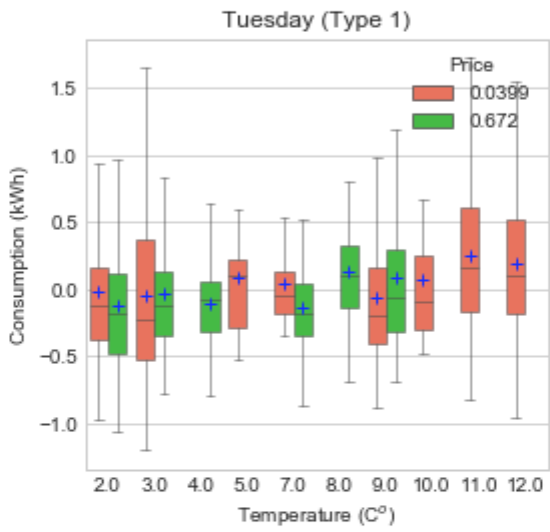
In [140]:
```
# price comparision
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
weeks = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for g in range(3): # three types
    for day_of_week in range(7):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3 * day_of_week + (g + 1)))
        df_day = df_all_res_long[(df_all_res_long['Day of week'] == day_of_week) & (df_all_res_long['Household type'] == g)]
        if df_day.shape[0] >= 1:
            sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_day, ax=ax_Ntou[-1], palette=['tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
            ax_Ntou[-1].set_title(weeks[day_of_week]  + ' (Type ' + str(g) + ')' )
            ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
            ax_Ntou[-1].set_ylabel('Consumption (kWh)')
            l = ax_Ntou[-1].legend()
            l.set_title('Price')
            for i,artist in enumerate(ax_Ntou[-1].artists):
                artist.set_linewidth(0.5)
                # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                # Loop over them here, and use the same colour as above
                for j in range(i*6,i*6+6):
                    line = ax_Ntou[-1].lines[j]
                    line.set_linewidth(0.5)
            ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

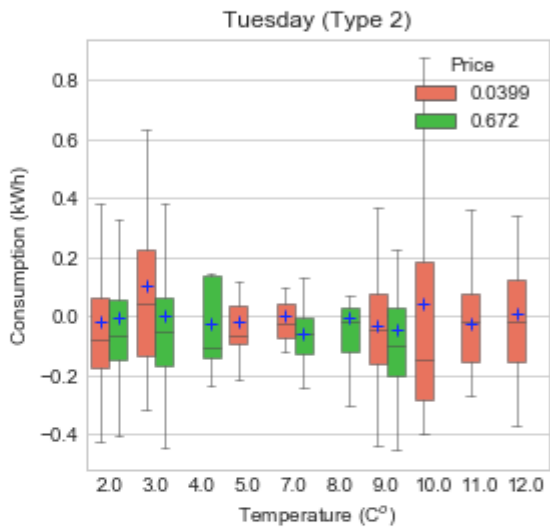

Monday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

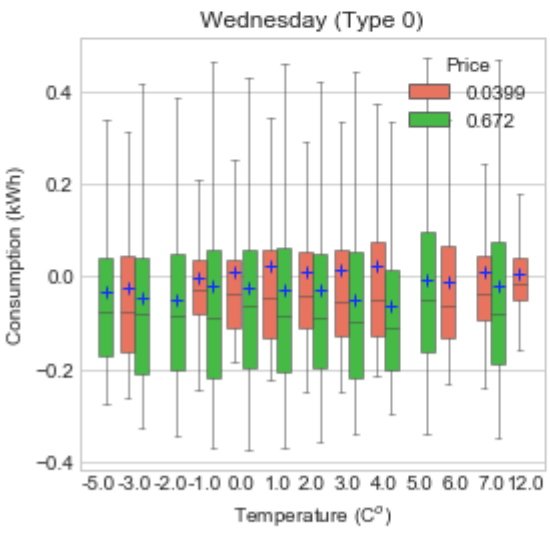

Monday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

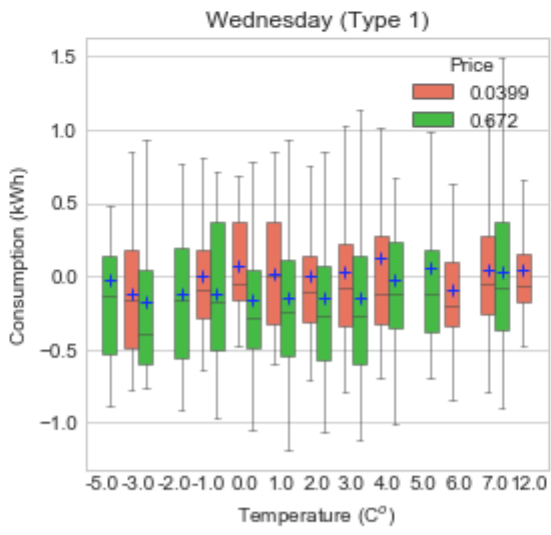

Monday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

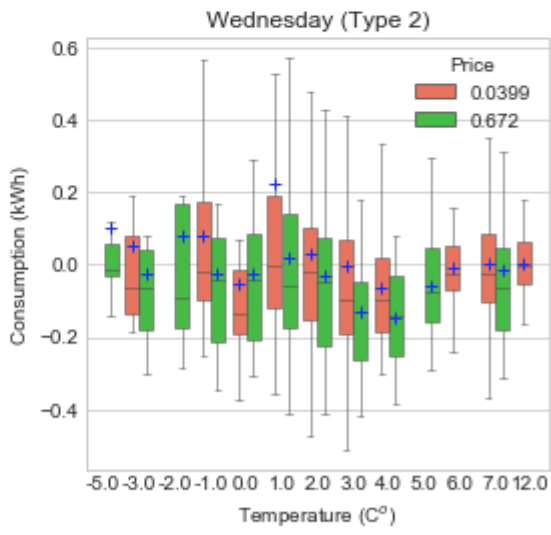

Tuesday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

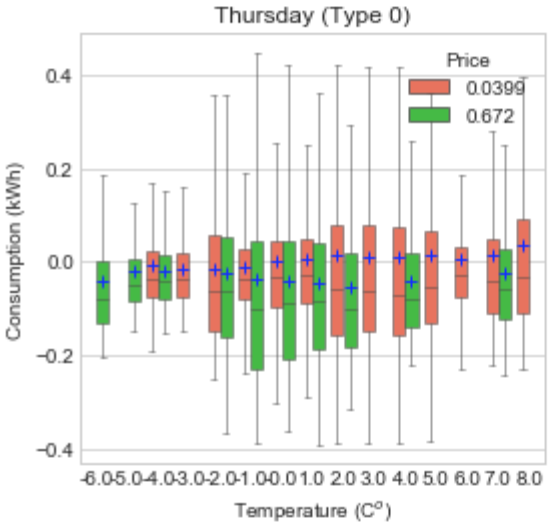

Tuesday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

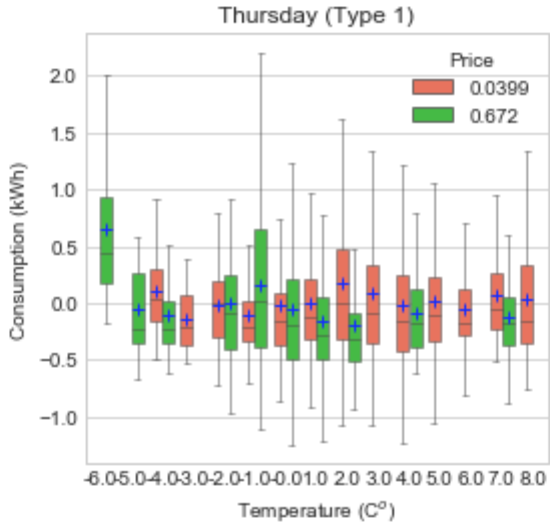

Tuesday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

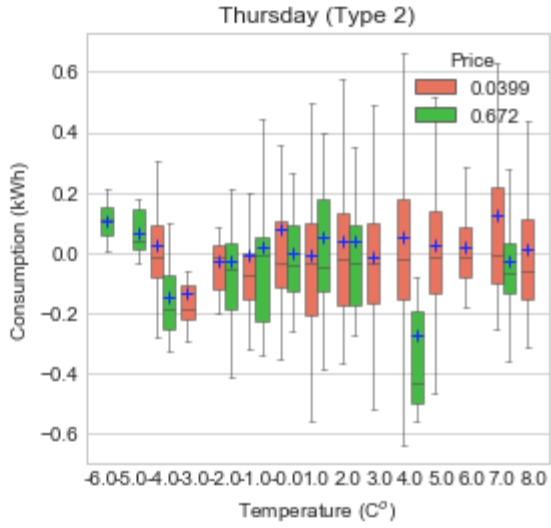

Wednesday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

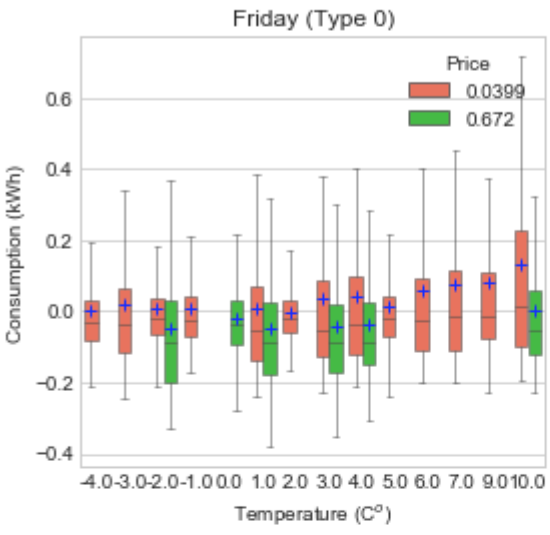

Wednesday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

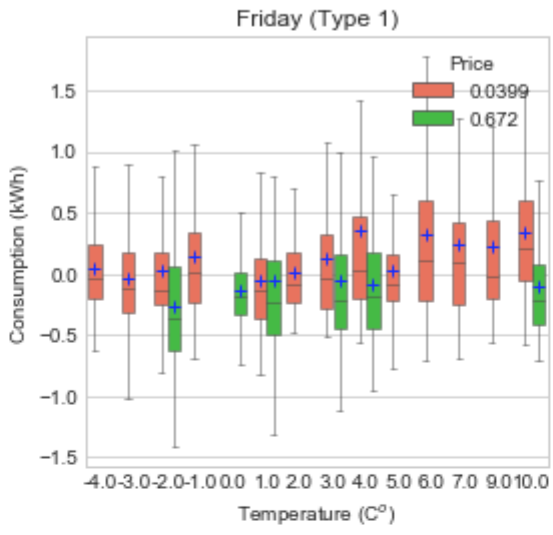

Wednesday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

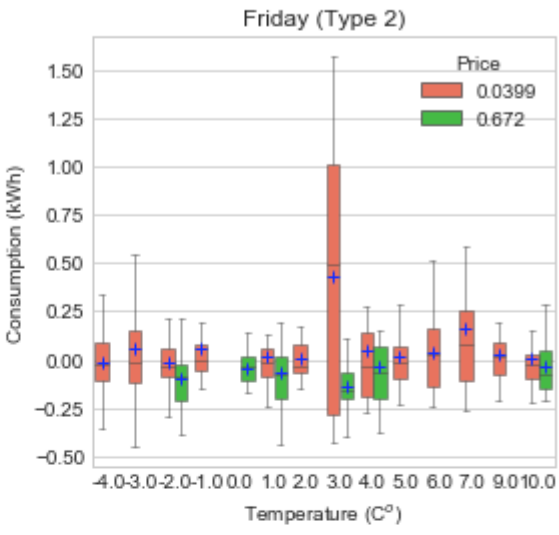

Thursday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

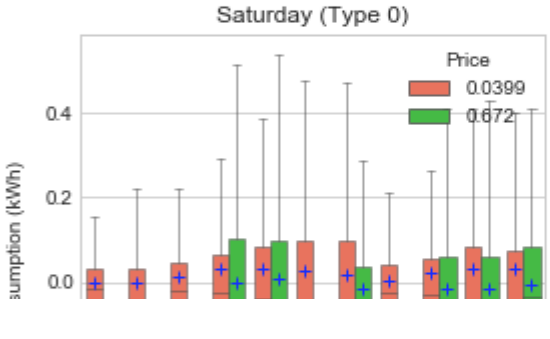

Thursday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

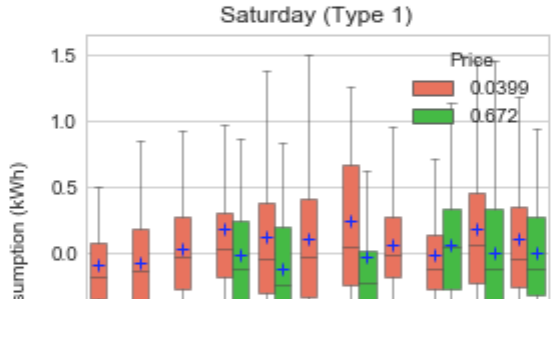

Thursday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

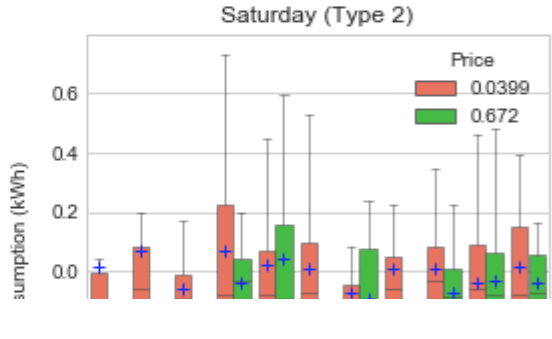

Friday (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

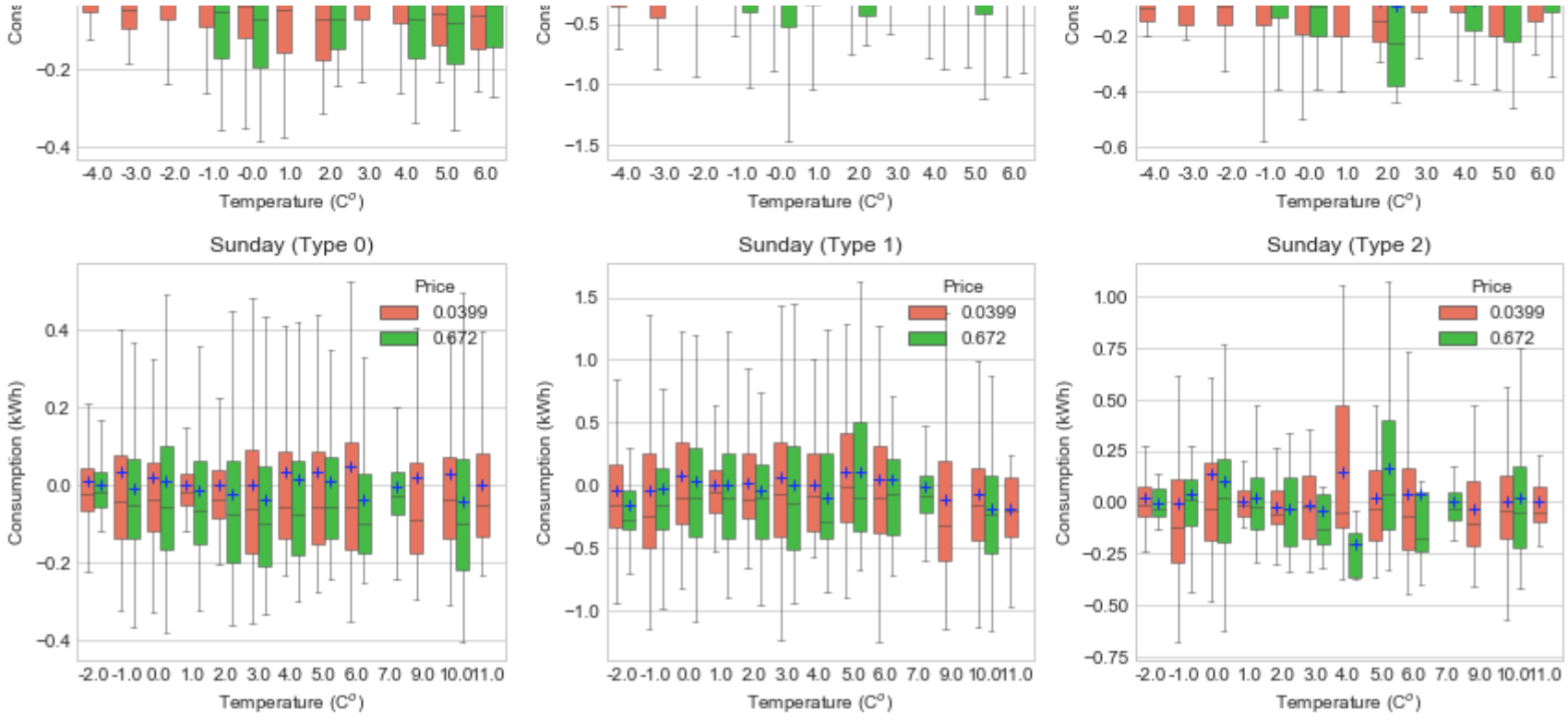

Friday (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)


Friday (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)


Saturday (Type 0)
Price
0.0399
0.672


Saturday (Type 1)
Price
0.0399
0.672

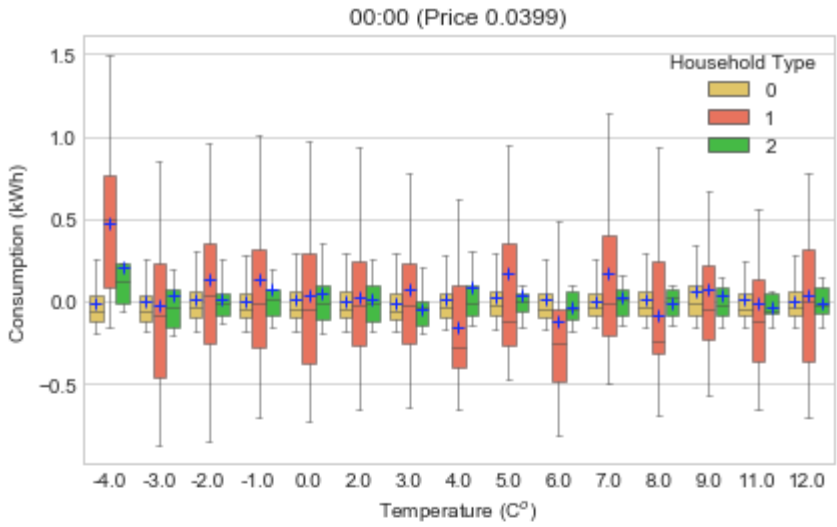

Saturday (Type 2)
Price
0.0399
0.672

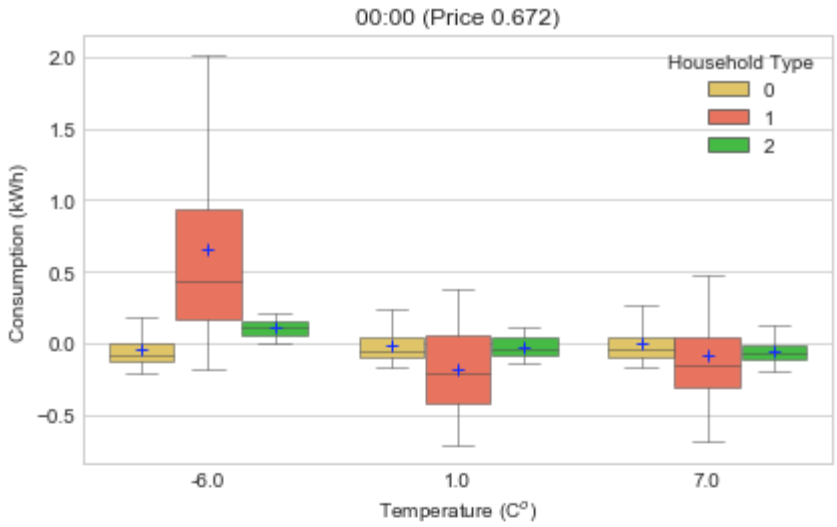


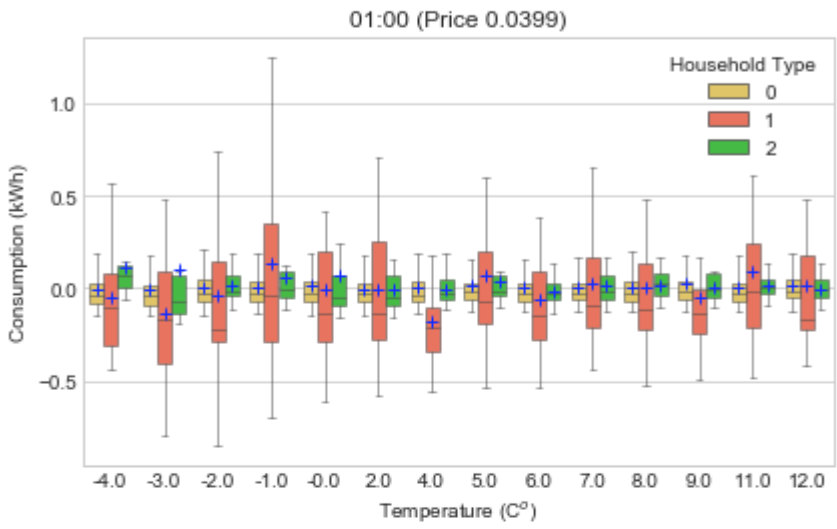
Household type 1 has a wider range of price responsiveness, we could may be focus more on studying of these type of users.

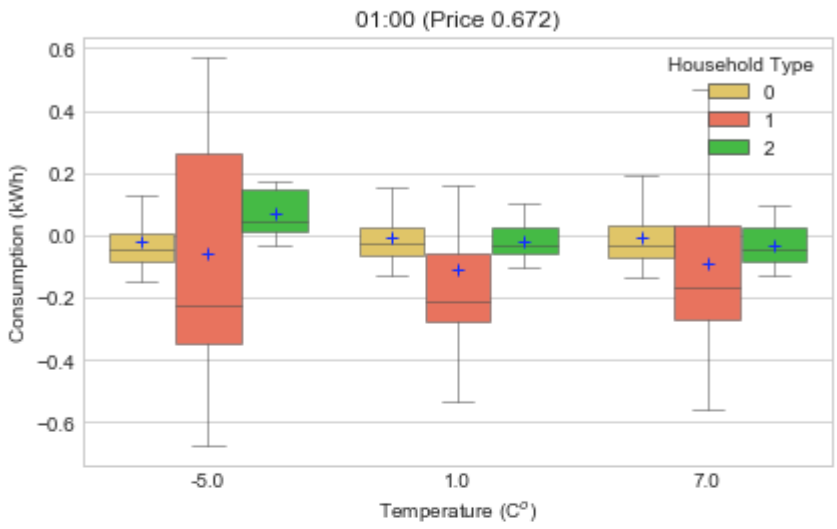
d) Hour of day - temperature : prices

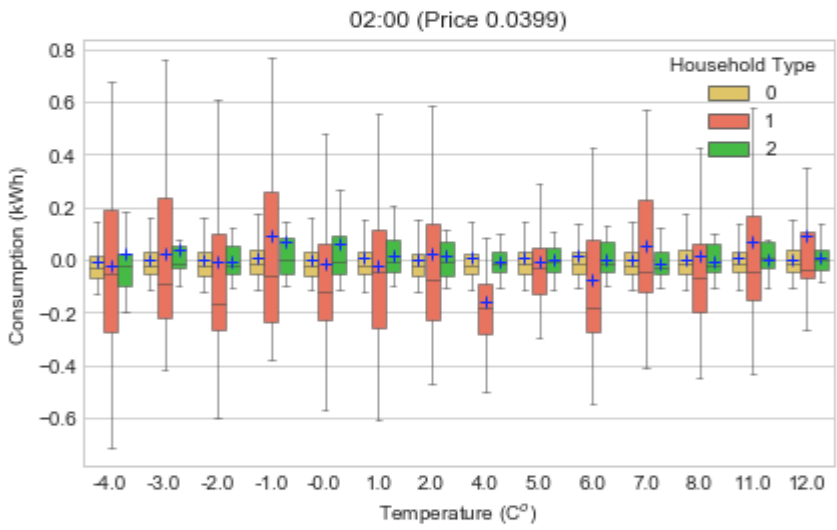
In [106]:
```
# hourly price responsiveness over temperatures and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for p in range(2): # two price levels
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 2, 2 * i + (p + 1)))
        df_hour = df_all_res_long[(df_all_res_long['Hour of day'] == i) & (df_all_res_long['Price'] == prices
[p])]
#         min_consumption = df_hour['Consumption'].min()
#         max_consumption = df_hour['Consumption'].max()
        if df_hour.shape[0] >= 1:
            sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_hour, ax=ax_Ntou[-1], palet
te=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfa
cecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
            if i < 9:
                ax_Ntou[-1].set_title('0' + str(i) + ':00' + ' (Price ' + str(prices[p]) + ')' )
            else:
                ax_Ntou[-1].set_title(str(i) + ':00' + ' (Price ' + str(prices[p]) + ')' )
            ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
            ax_Ntou[-1].set_ylabel('Consumption (kWh)')
            l = ax_Ntou[-1].legend()
            l.set_title('Household Type')
            for i,artist in enumerate(ax_Ntou[-1].artists):
                artist.set_linewidth(0.5)
                # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                # Loop over them here, and use the same colour as above
                for j in range(i*6,i*6+6):
                    line = ax_Ntou[-1].lines[j]
                    line.set_linewidth(0.5)
            ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

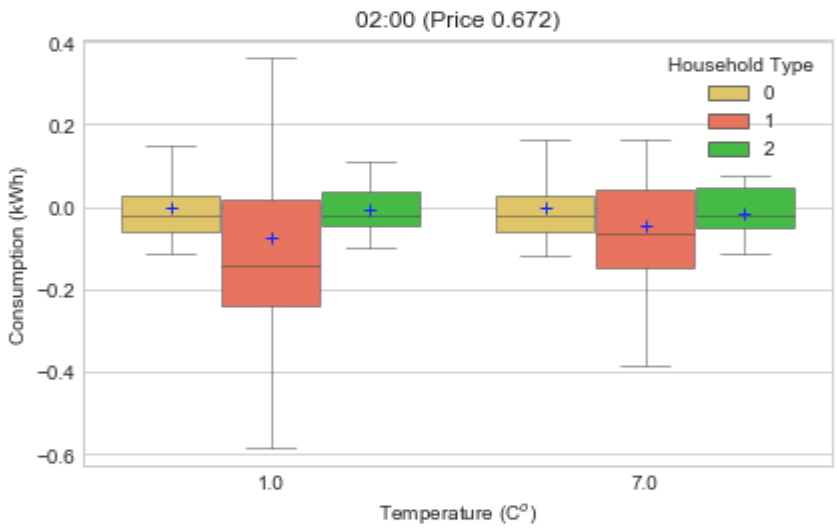

00:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
-4.0 -3.0 -2.0 -1.0 0.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 11.0 12.0
Temperature (C°)

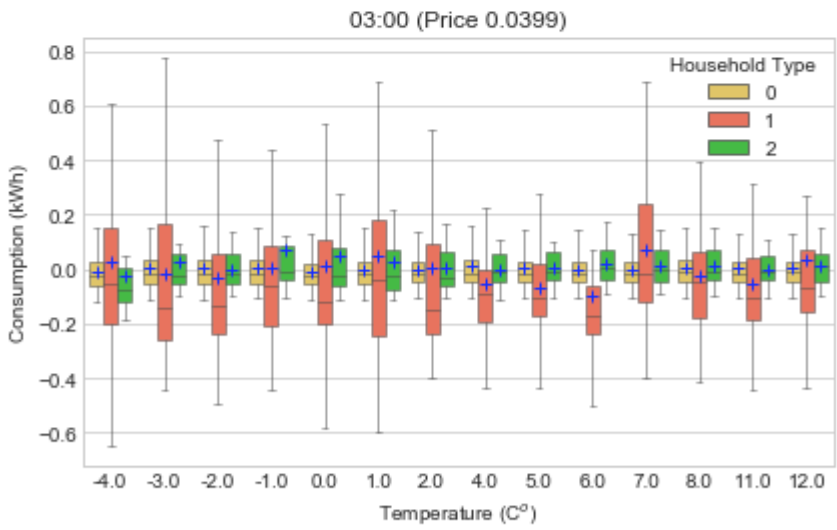

00:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
-6.0
1.0
7.0
Temperature (C°)

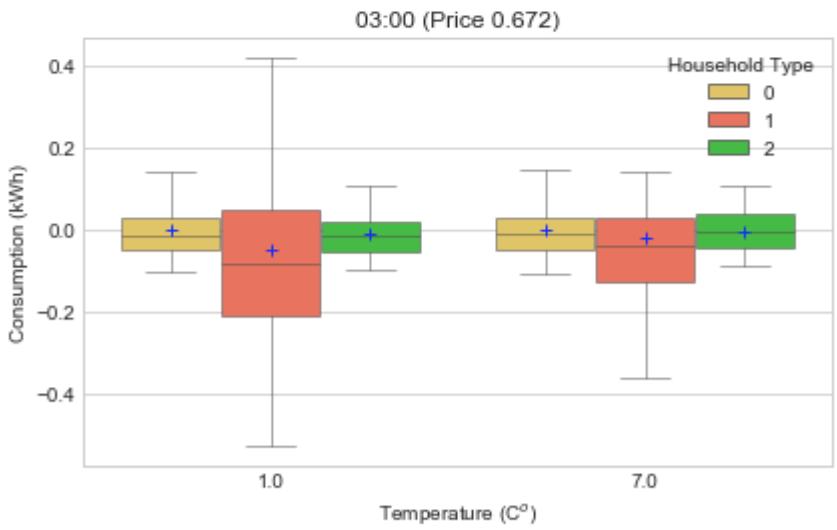

01:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
-4.0 -3.0 -2.0 -1.0 -0.0 2.0 4.0 5.0 6.0 7.0 8.0 9.0 11.0 12.0
Temperature (C°)

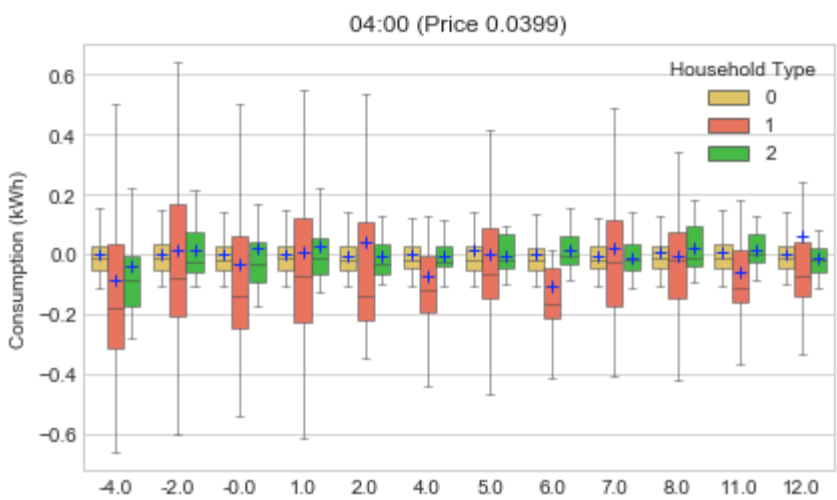

01:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-5.0
1.0
7.0
Temperature (C°)

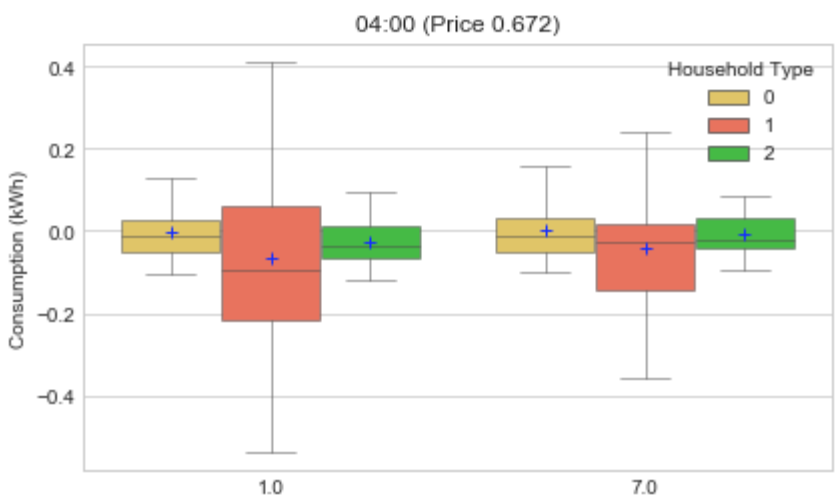

02:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0 -3.0 -2.0 -1.0 -0.0 1.0 2.0 4.0 5.0 6.0 7.0 8.0 11.0 12.0
Temperature (C°)


02:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
1.0
7.0
Temperature (C°)


03:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0 -3.0 -2.0 -1.0 0.0 1.0 2.0 4.0 5.0 6.0 7.0 8.0 11.0 12.0
Temperature (C°)

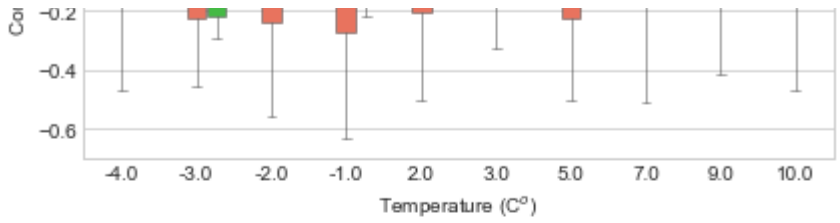

03:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
1.0
7.0
Temperature (C°)

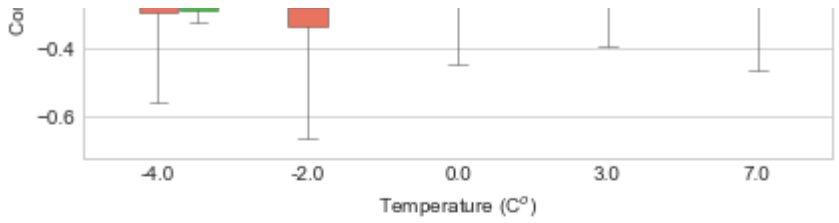

04:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
-4.0 -2.0 -0.0 1.0 2.0 4.0 5.0 6.0 7.0 8.0 11.0 12.0
Temperature (C°)

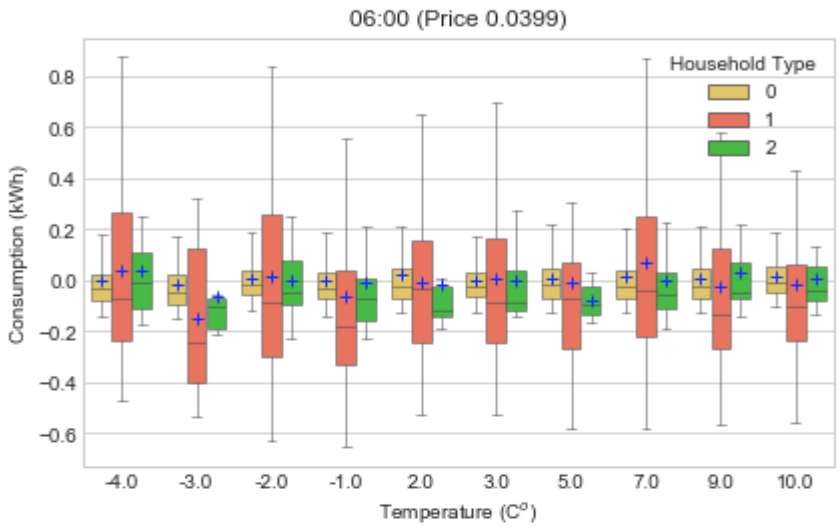

04:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
1.0
7.0
Temperature (C°)

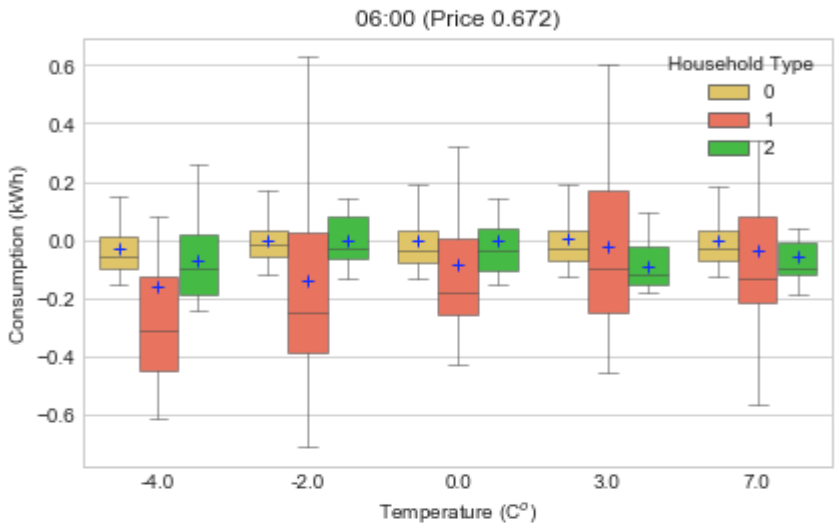
05:00 (Price 0.0399)
Household Type
0
1
2
0.6
0.4
0.2
0.0

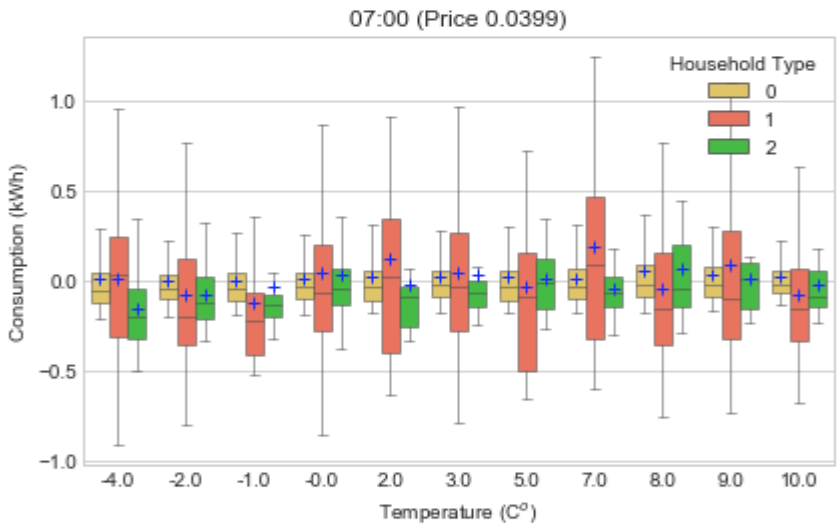
05:00 (Price 0.672)
Household Type
0
1
2
0.4
0.2
0.0
−0.2

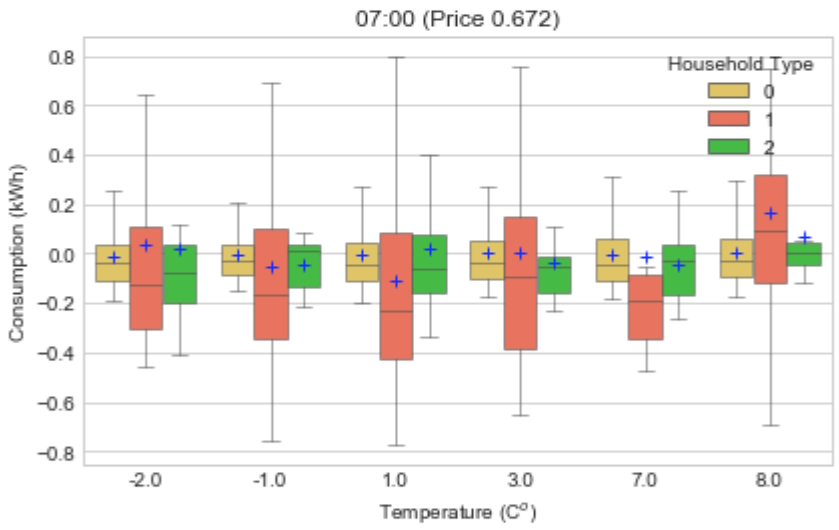

-0.2
-0.4
-0.6
-4.0
-3.0
-2.0
-1.0
2.0
3.0
5.0
7.0
9.0
10.0
Temperature (C°)

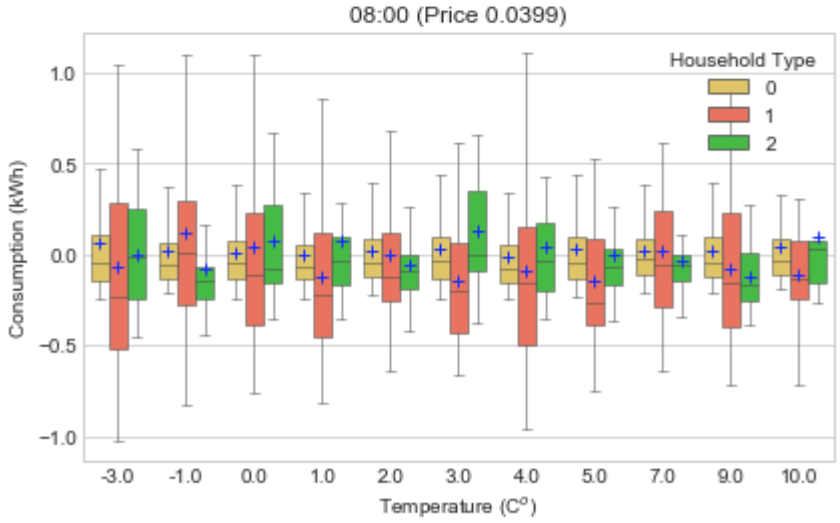

-0.4
-0.6
-4.0
-2.0
0.0
3.0
7.0
Temperature (C°)

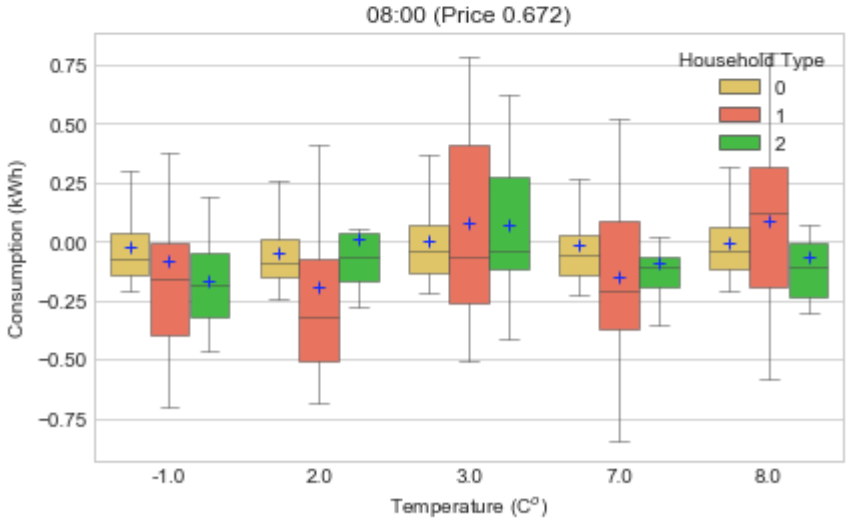

06:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-4.0
-3.0
-2.0
-1.0
2.0
3.0
5.0
7.0
9.0
10.0
Temperature (C°)

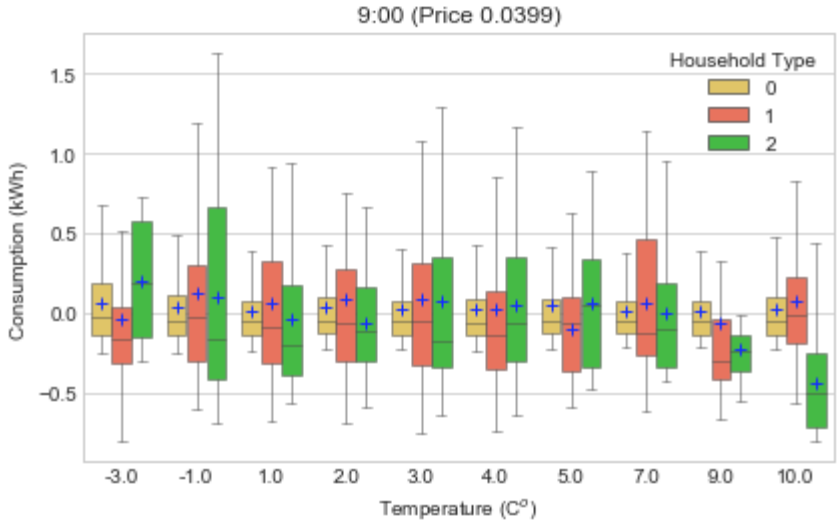

06:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-4.0
-2.0
0.0
3.0
7.0
Temperature (C°)

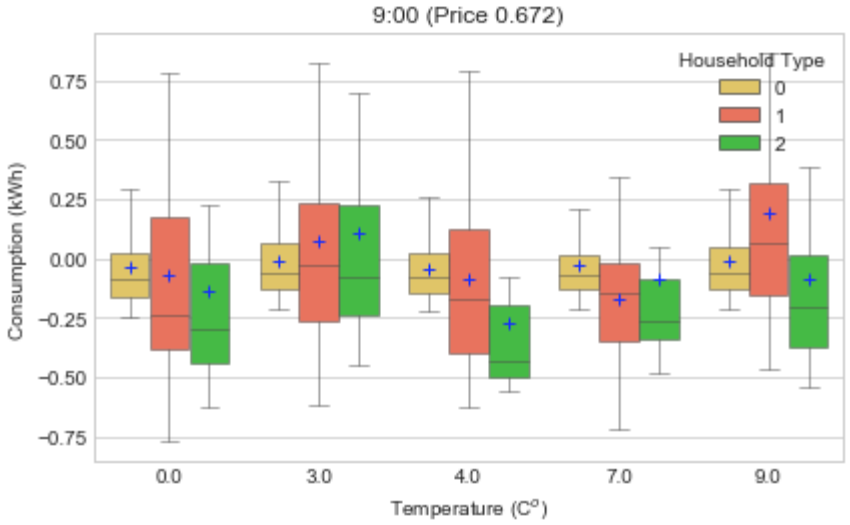

07:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
-0.5
-1.0
-4.0
-2.0
-1.0
-0.0
2.0
3.0
5.0
7.0
8.0
9.0
10.0
Temperature (C°)

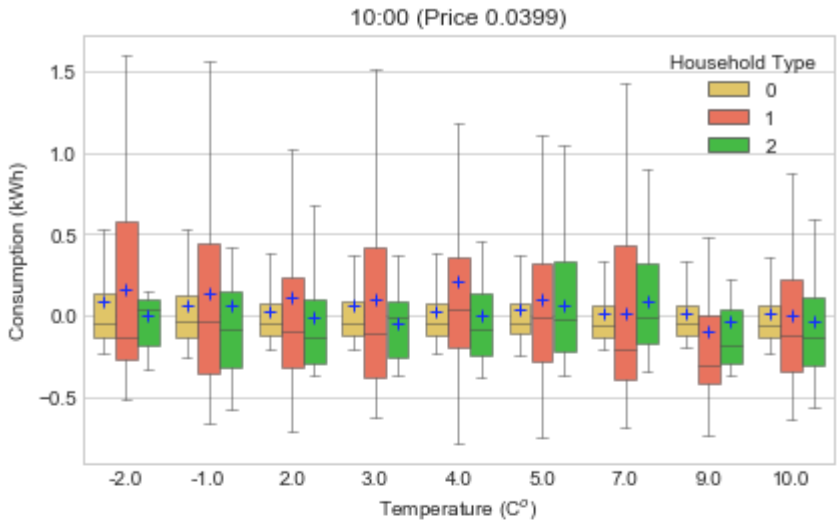

07:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-2.0
-1.0
1.0
3.0
7.0
8.0
Temperature (C°)

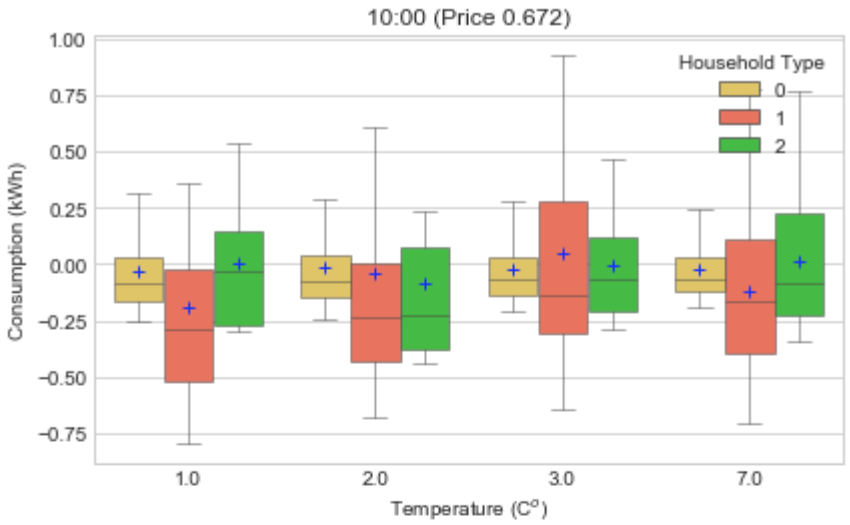

08:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0
0.5
0.0
-0.5
-1.0
-3.0
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
7.0
9.0
10.0
Temperature (C°)


08:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.0
2.0
3.0
7.0
8.0
Temperature (C°)


9:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
-0.5
-3.0
-1.0
1.0
2.0
3.0
4.0
5.0
7.0
9.0
10.0
Temperature (C°)

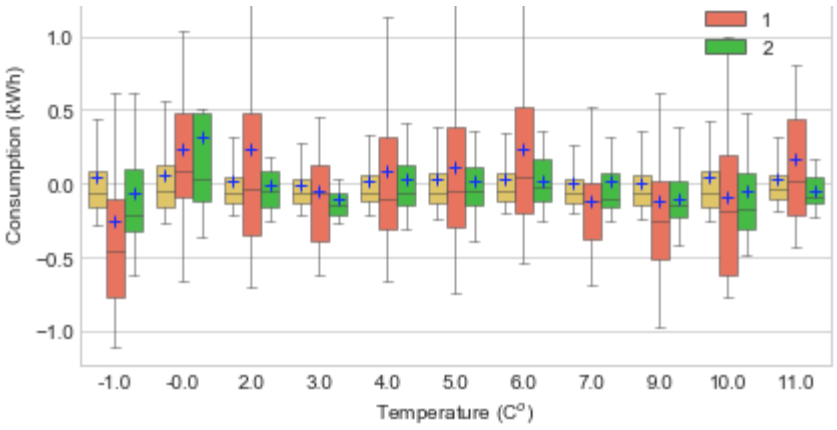

9:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
0.0
3.0
4.0
7.0
9.0
Temperature (C°)

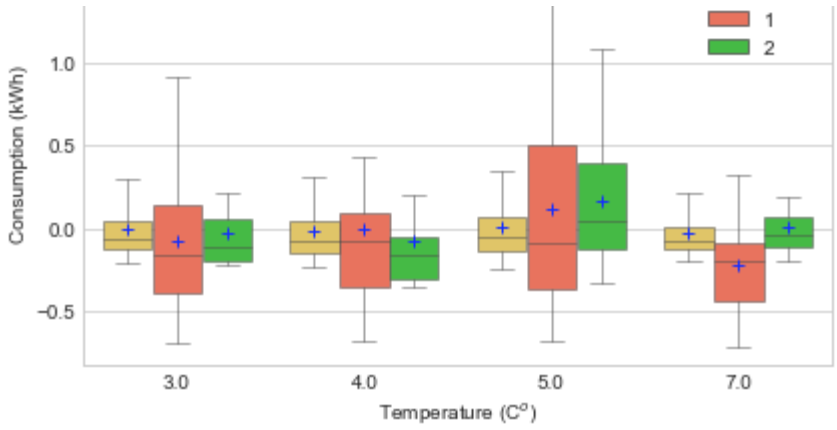

10:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
-0.5
-2.0
-1.0
2.0
3.0
4.0
5.0
7.0
9.0
10.0
Temperature (C°)

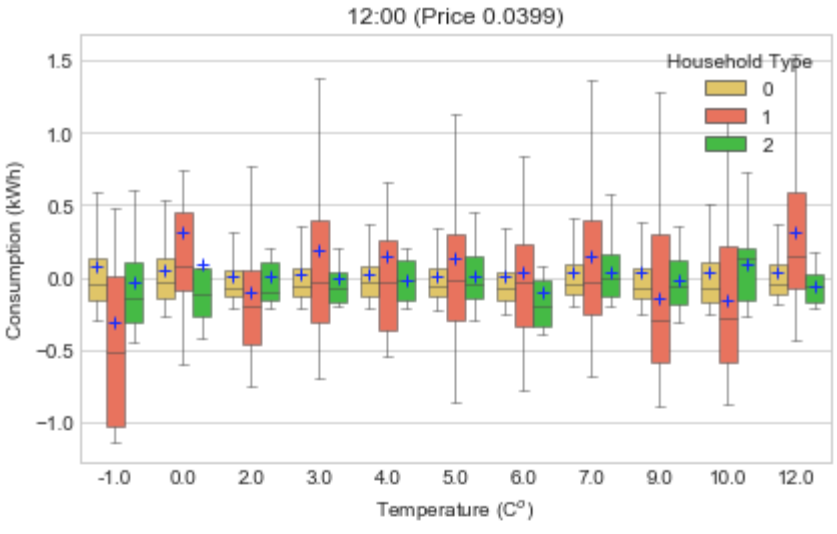

10:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
1.0
2.0
3.0
7.0
Temperature (C°)

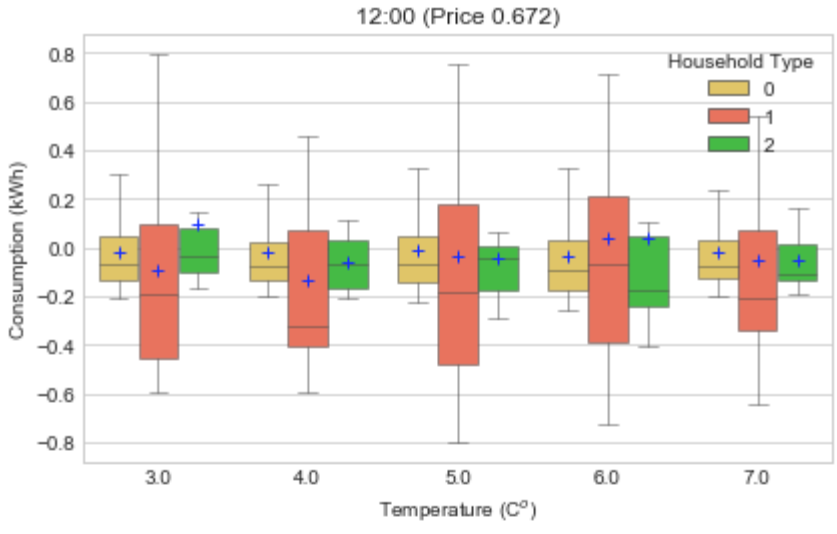
11:00 (Price 0.0399)
Household Type
0
1.5

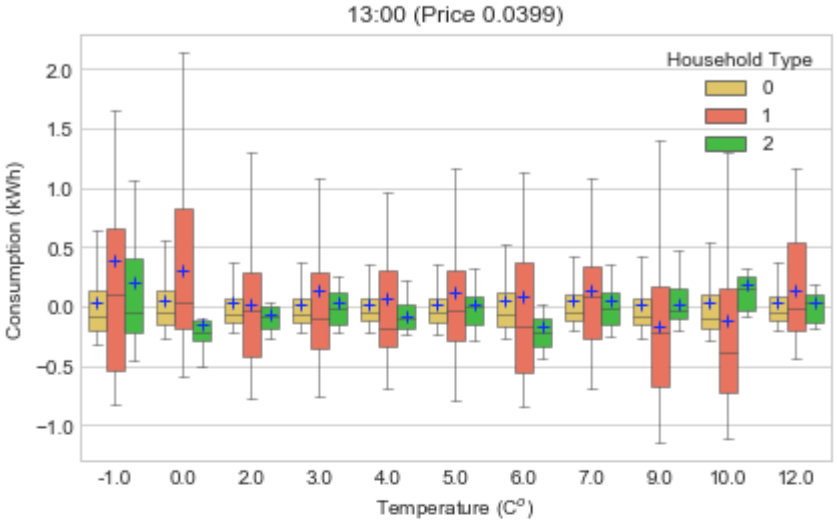
11:00 (Price 0.672)
Household Type
0
1.5

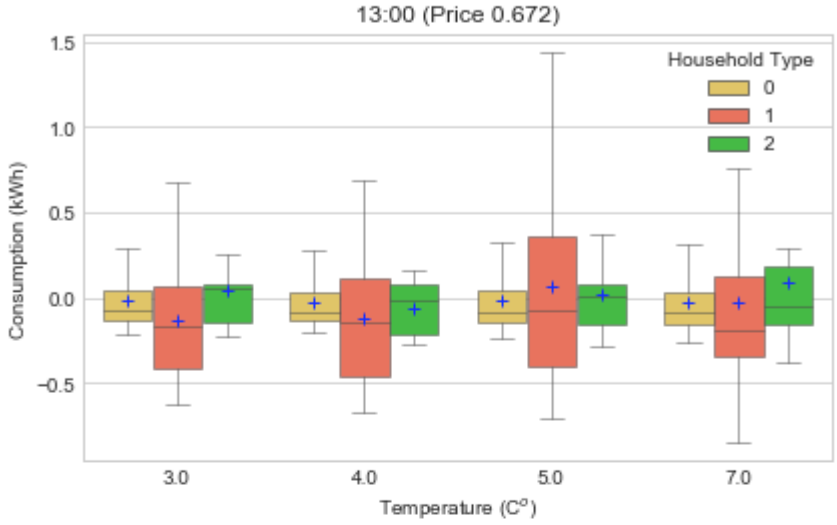

1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
−1.0
-1.0
-0.0
2.0
3.0
4.0
5.0
6.0
7.0
9.0
10.0
11.0
Temperature (C°)

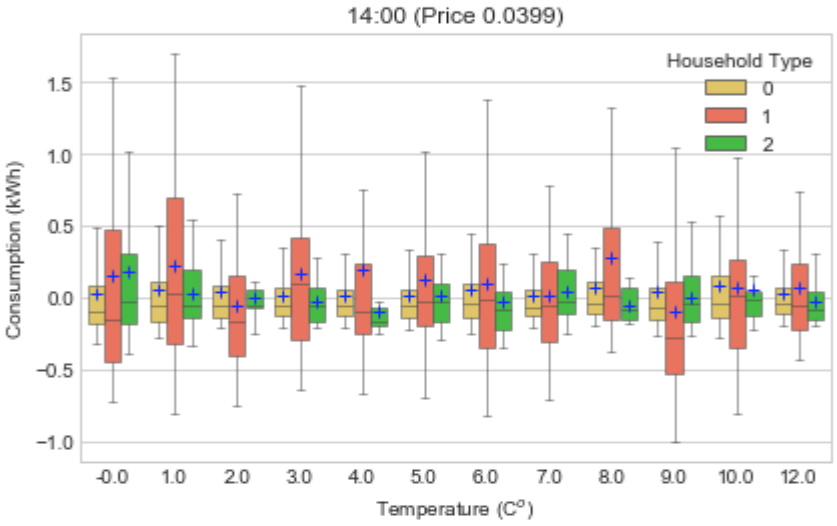

1
2
Consumption (kWh)
1.0
0.5
0.0
−0.5
3.0
4.0
5.0
7.0
Temperature (C°)

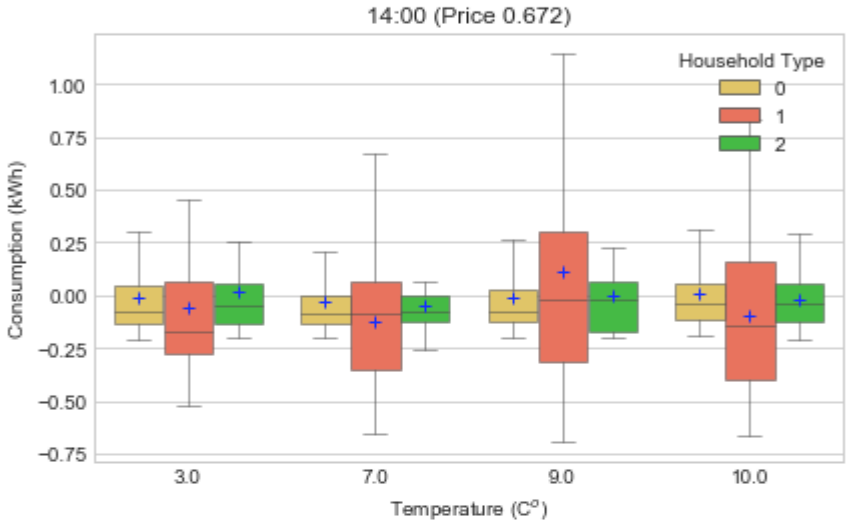

12:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
0.0
2.0
3.0
4.0
5.0
6.0
7.0
9.0
10.0
12.0
Temperature (C°)

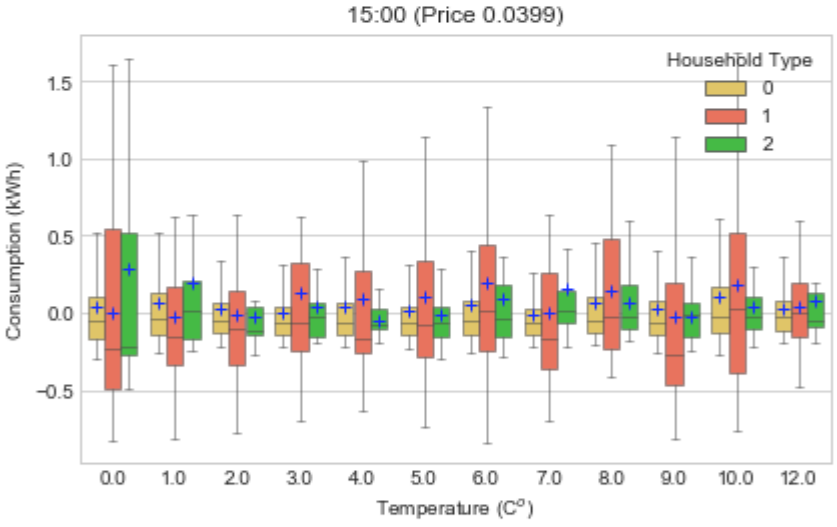

12:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
3.0
4.0
5.0
6.0
7.0
Temperature (C°)

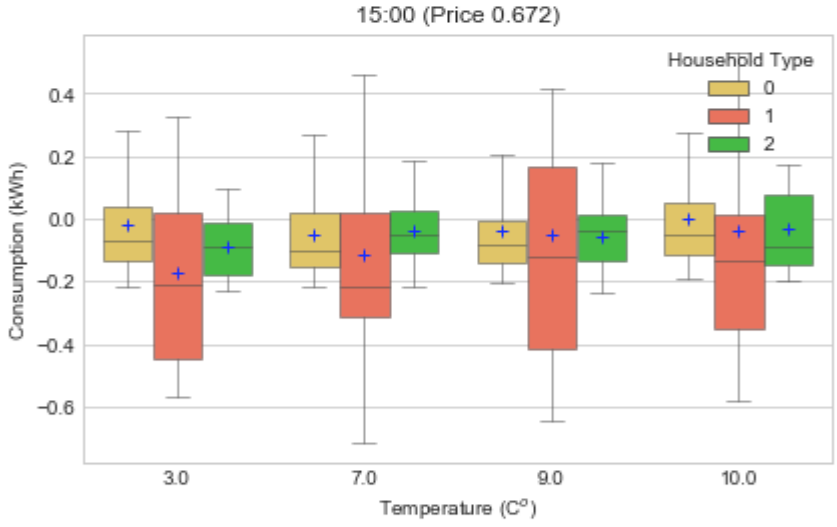

13:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
-1.0
0.0
2.0
3.0
4.0
5.0
6.0
7.0
9.0
10.0
12.0
Temperature (C°)

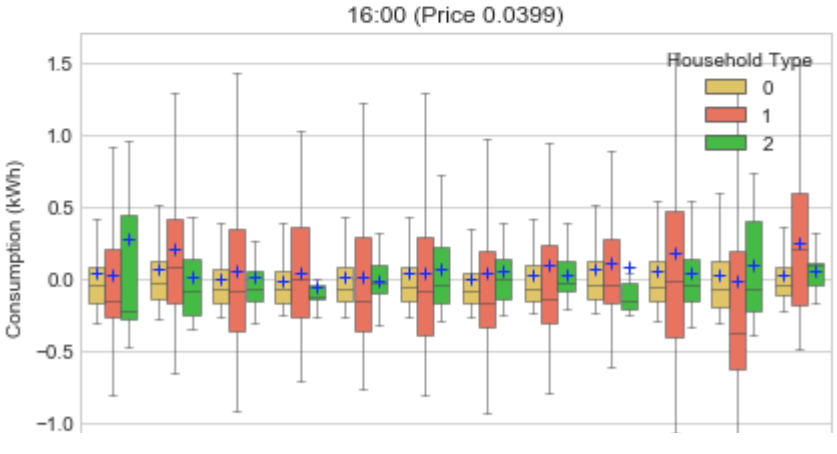

13:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
3.0
4.0
5.0
7.0
Temperature (C°)

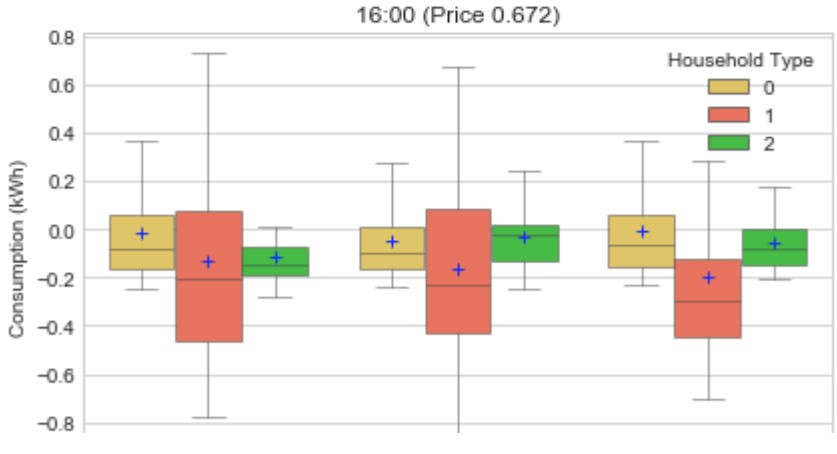

14:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
12.0
Temperature (C°)

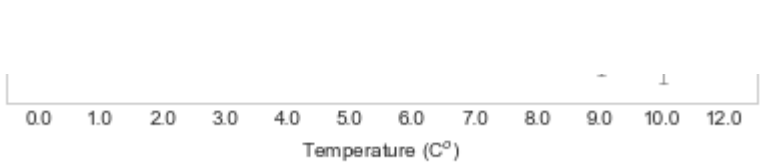

14:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
3.0
7.0
9.0
10.0
Temperature (C°)

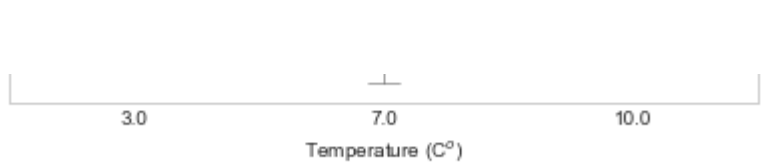

15:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
12.0
Temperature (C°)

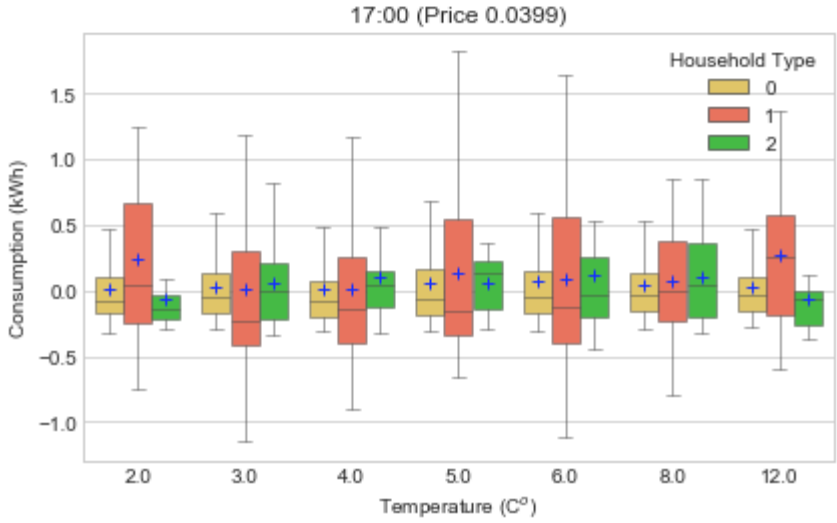

15:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.4
0.2
0.0
−0.2
−0.4
−0.6
3.0
7.0
9.0
10.0
Temperature (C°)

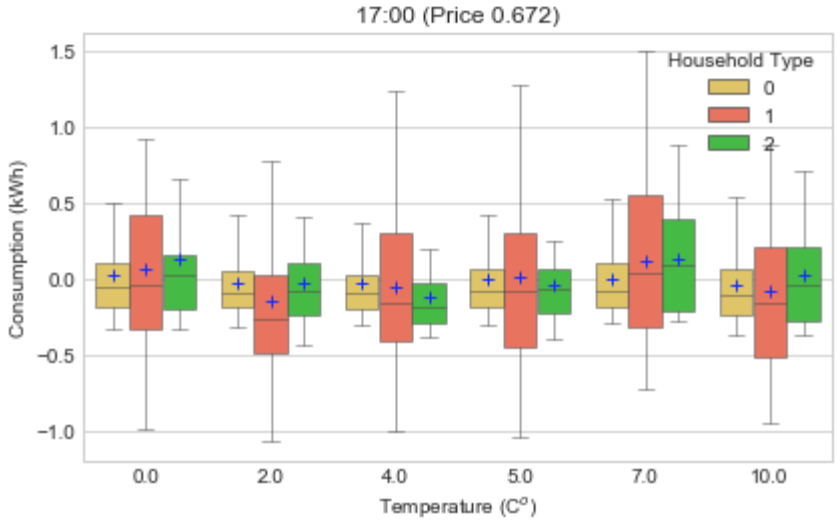

16:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0

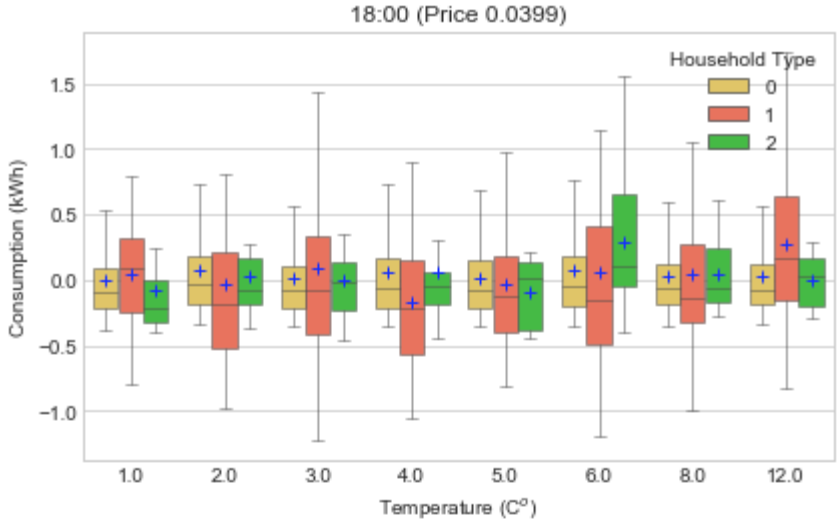

16:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8

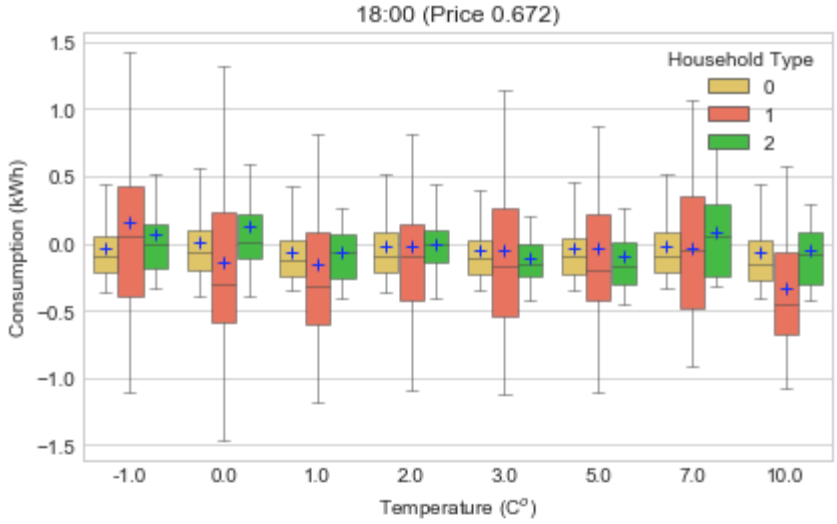

0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 12.0
Temperature (C°)

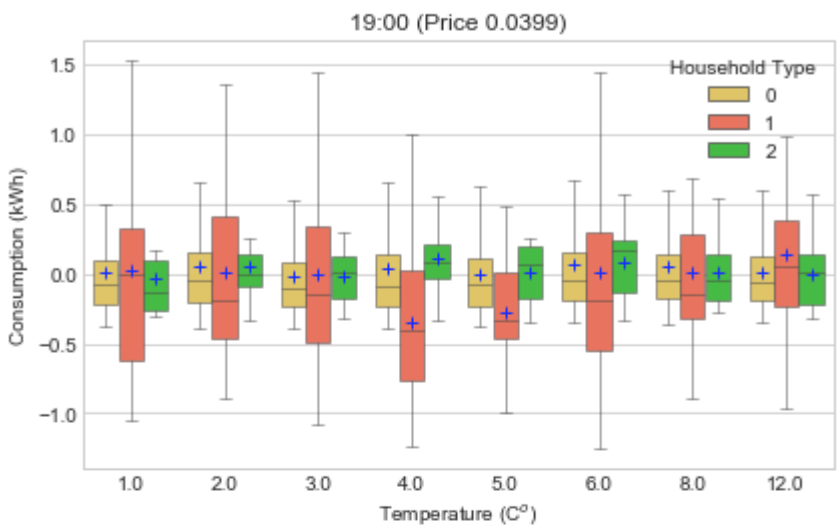

3.0 7.0 10.0
Temperature (C°)

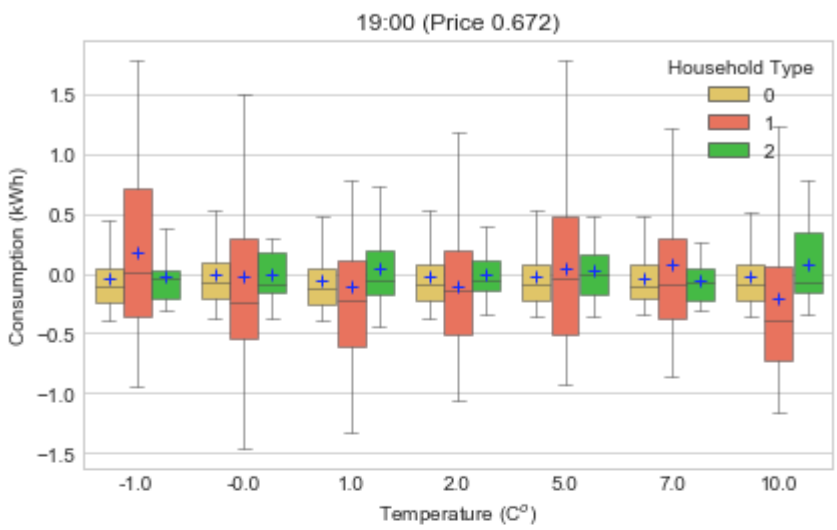

17:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
2.0 3.0 4.0 5.0 6.0 8.0 12.0
Temperature (C°)

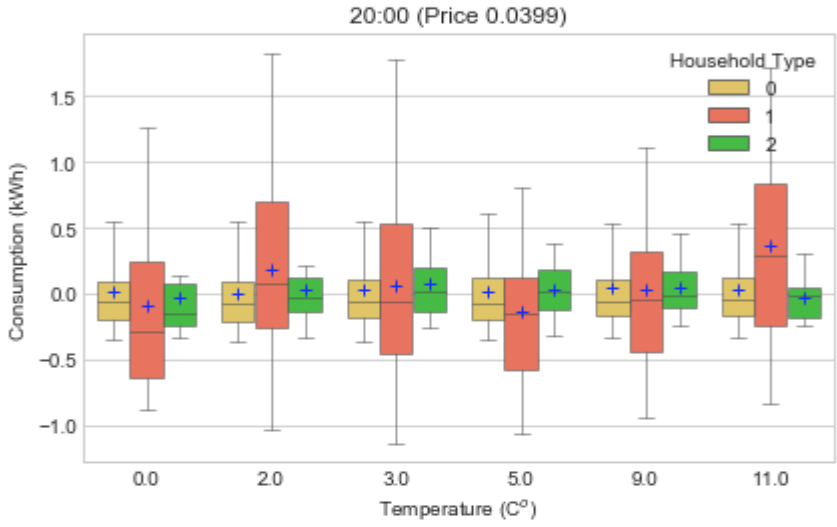

17:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
0.0 2.0 4.0 5.0 7.0 10.0
Temperature (C°)

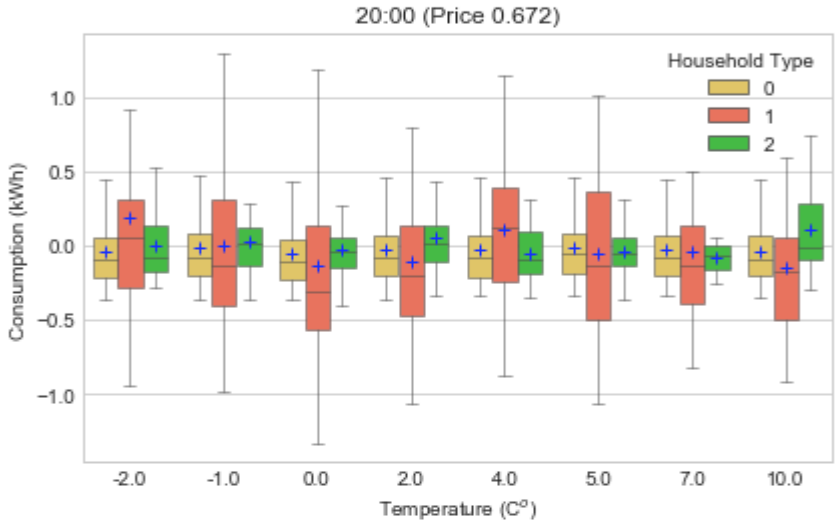

18:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0 2.0 3.0 4.0 5.0 6.0 8.0 12.0
Temperature (C°)

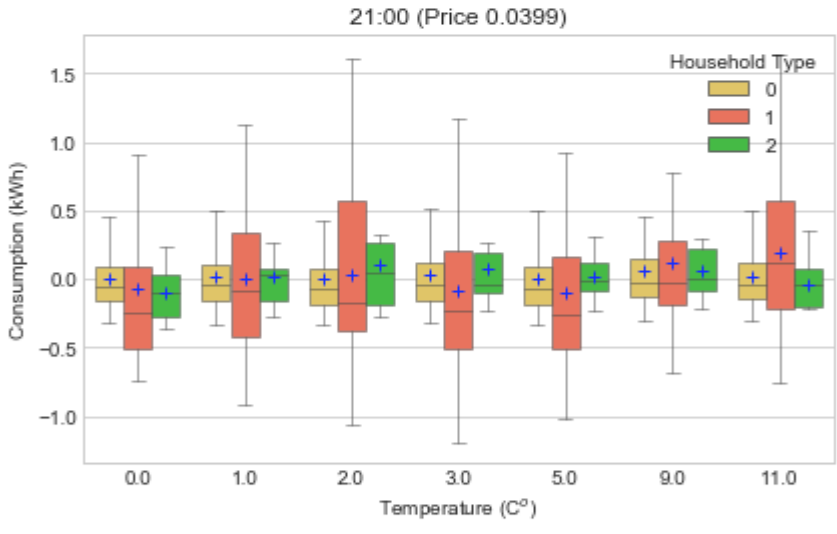

18:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
-1.0 0.0 1.0 2.0 3.0 5.0 7.0 10.0
Temperature (C°)

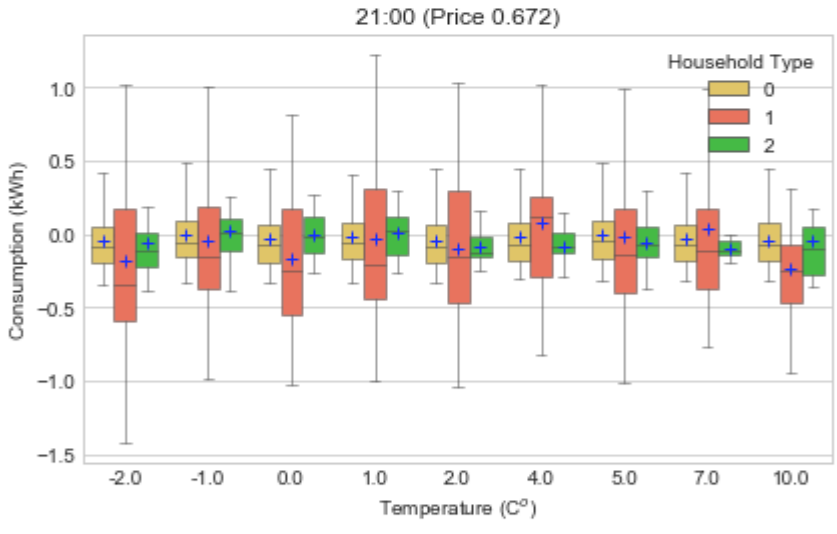

19:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.0 2.0 3.0 4.0 5.0 6.0 8.0 12.0
Temperature (C°)

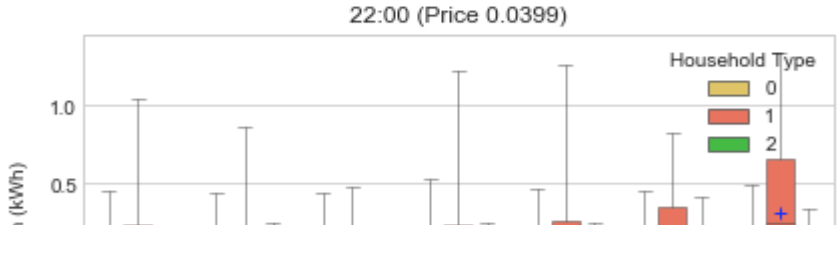

19:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
-1.0 -0.0 1.0 2.0 5.0 7.0 10.0
Temperature (C°)

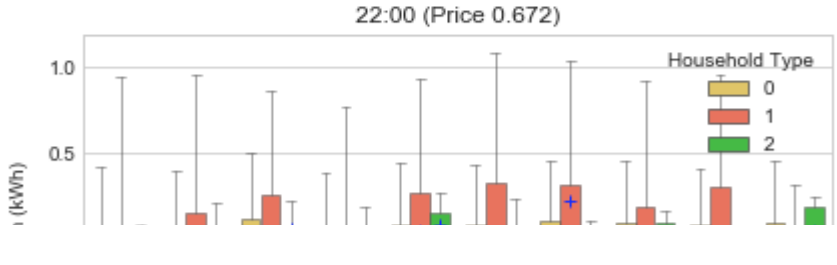

20:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.0 2.0 3.0 5.0 9.0 11.0
Temperature (C°)

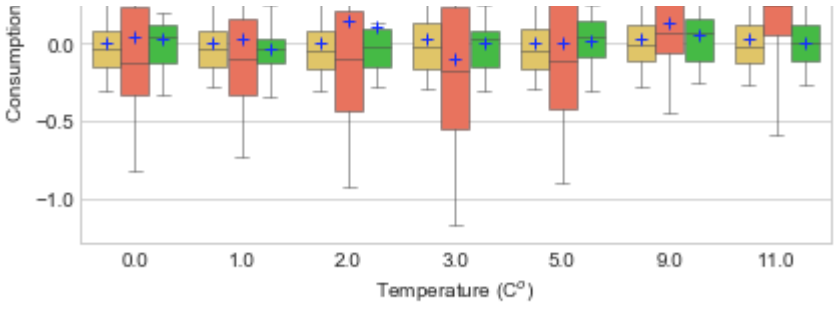

20:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
-2.0 -1.0 0.0 2.0 4.0 5.0 7.0 10.0
Temperature (C°)

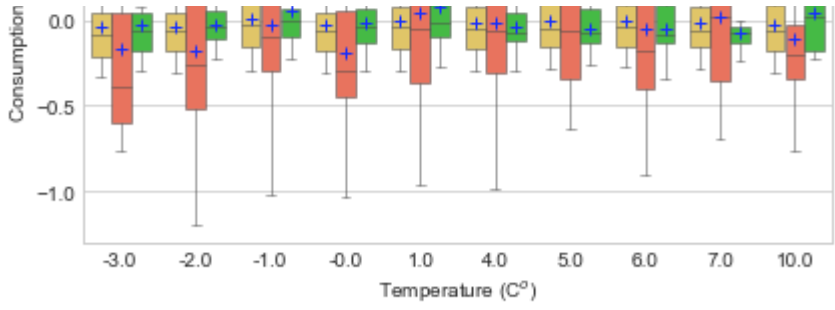

21:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
0.0 1.0 2.0 3.0 5.0 9.0 11.0
Temperature (C°)

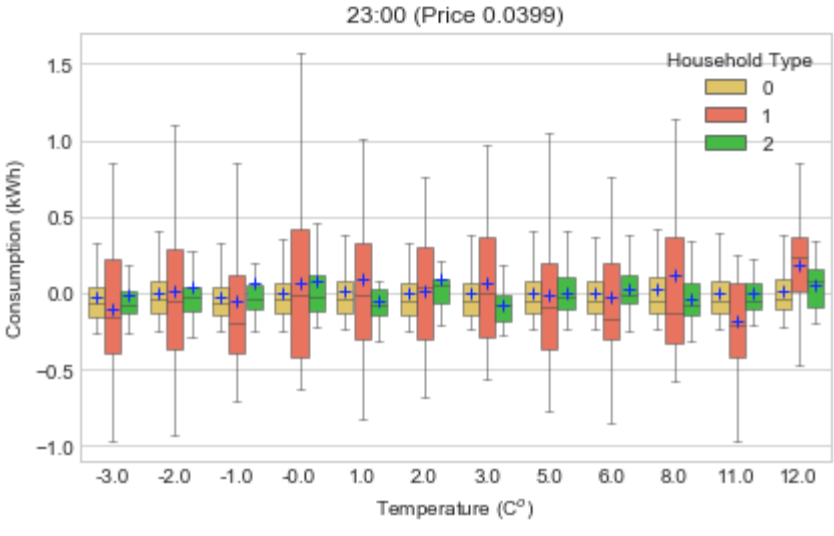

21:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
-2.0 -1.0 0.0 1.0 2.0 4.0 5.0 7.0 10.0
Temperature (C°)

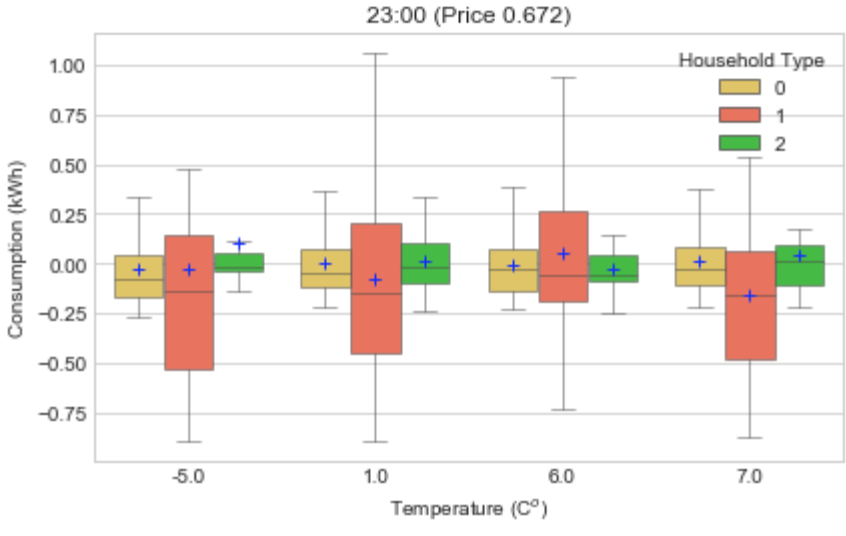

22:00 (Price 0.0399)
Household Type
0
1
2


22:00 (Price 0.672)
Household Type
0
1
2

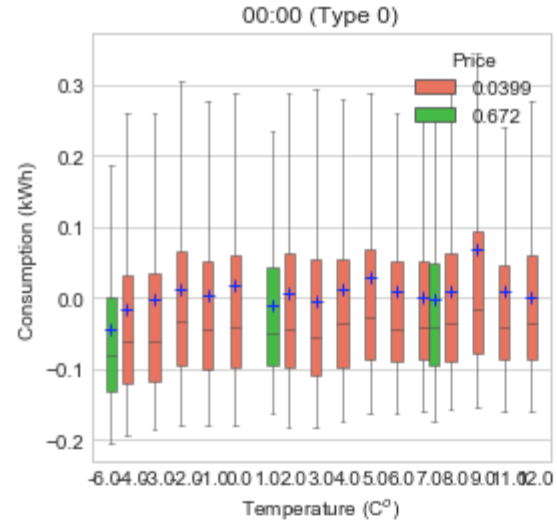

Consumption
0.0
−0.5
−1.0
0.0
1.0
2.0
3.0
5.0
9.0
11.0
Temperature (C°)

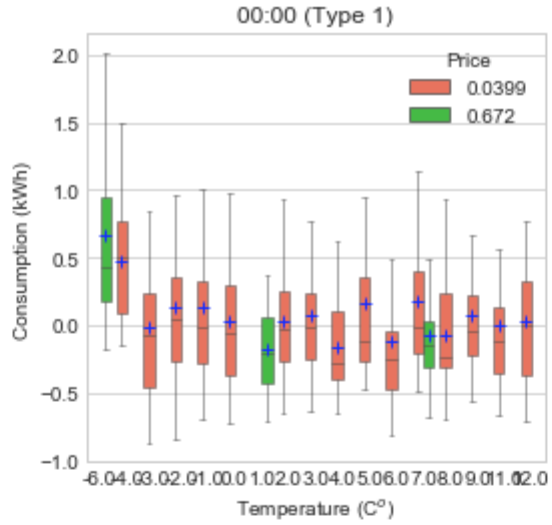

Consumption
0.0
−0.5
−1.0
-3.0
-2.0
-1.0
-0.0
1.0
4.0
5.0
6.0
7.0
10.0
Temperature (C°)

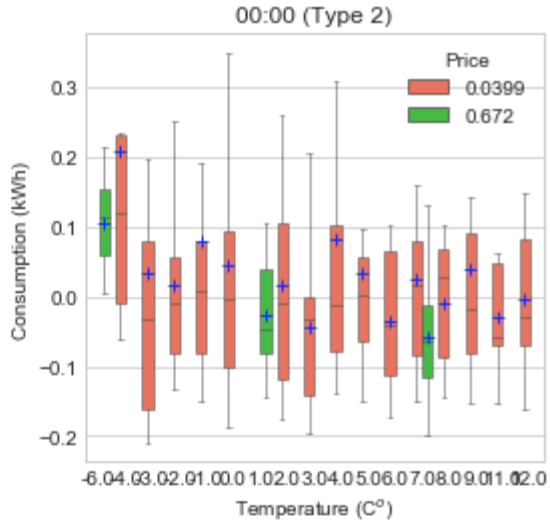

23:00 (Price 0.0399)
Household Type
0
1
2
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-3.0
-2.0
-1.0
-0.0
1.0
2.0
3.0
5.0
6.0
8.0
11.0
12.0
Temperature (C°)

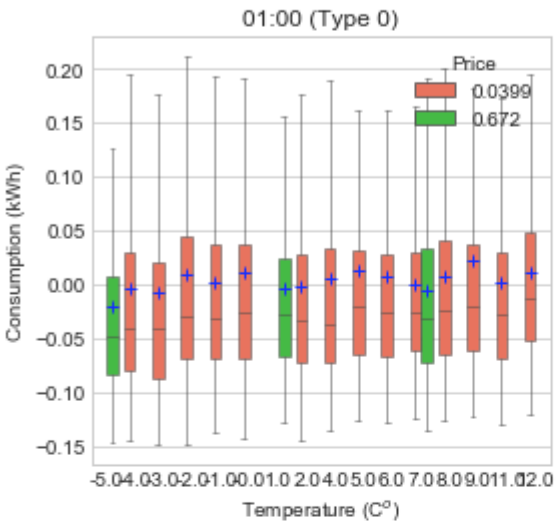

23:00 (Price 0.672)
Household Type
0
1
2
Consumption (kWh)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
-5.0
1.0
6.0
7.0
Temperature (C°)

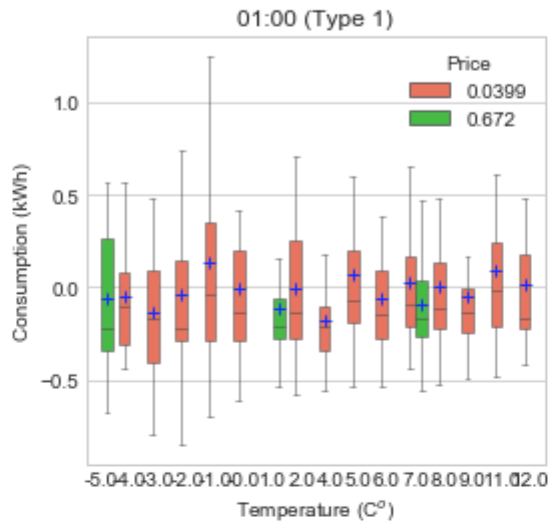
In [141]:
```
# price comparision
# hourly price responsiveness over temperatures and prices
fig_all = plt.figure(figsize = (12,90))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for g in range(3): # three types
    for i in range(24):
        ax_Ntou.append(fig_all.add_subplot(24, 3, 3* i + (g + 1)))
        df_hour = df_all_res_long[(df_all_res_long['Hour of day'] == i) & (df_all_res_long['Household type']
== g)]
#         min_consumption = df_hour['Consumption'].min()
#         max_consumption = df_hour['Consumption'].max()
        if df_hour.shape[0] >= 1:
            sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_hour, ax=ax_Ntou[-1], palette=['toma
to','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:vivid b
lue", "markeredgecolor":"xkcd:vivid blue"})
            if i < 9:
                ax_Ntou[-1].set_title('0' + str(i) + ':00' + ' (Type ' + str(g) + ')' )
            else:
                ax_Ntou[-1].set_title(str(i) + ':00' + ' (Type ' + str(g) + ')' )
            ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
            ax_Ntou[-1].set_ylabel('Consumption (kWh)')
            l = ax_Ntou[-1].legend()
            l.set_title('Price')
            for i,artist in enumerate(ax_Ntou[-1].artists):
                artist.set_linewidth(0.5)
                # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
                # Loop over them here, and use the same colour as above
                for j in range(i*6,i*6+6):
                    line = ax_Ntou[-1].lines[j]
                    line.set_linewidth(0.5)
            ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```

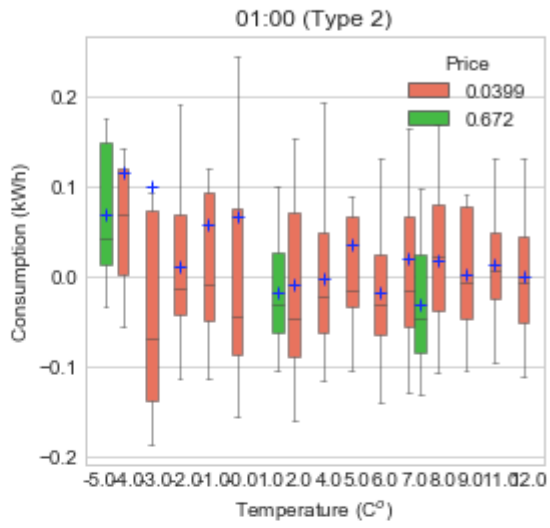

00:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

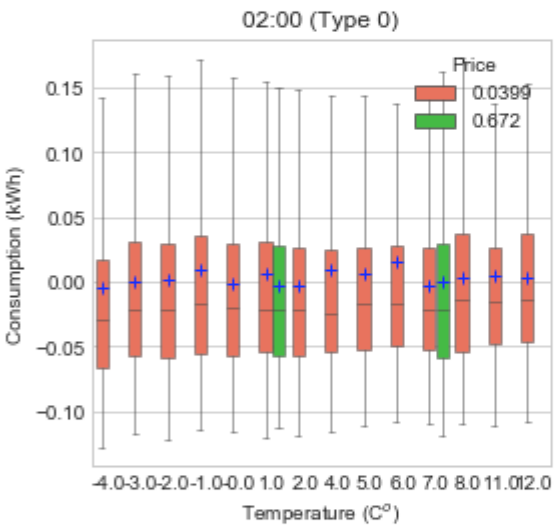

00:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

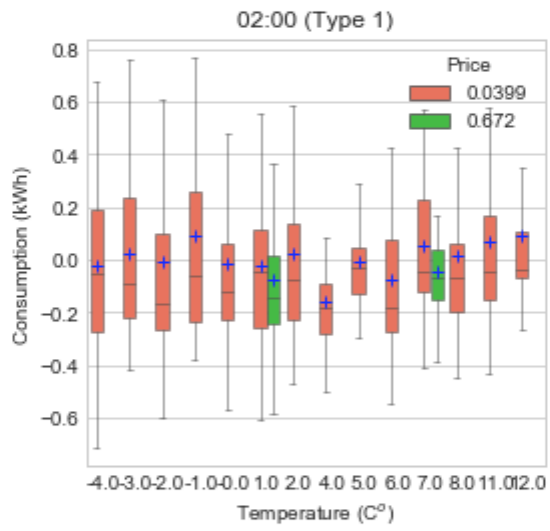

00:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

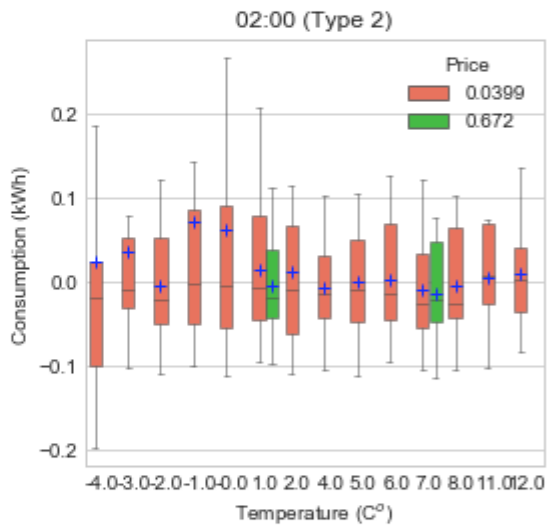

01:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

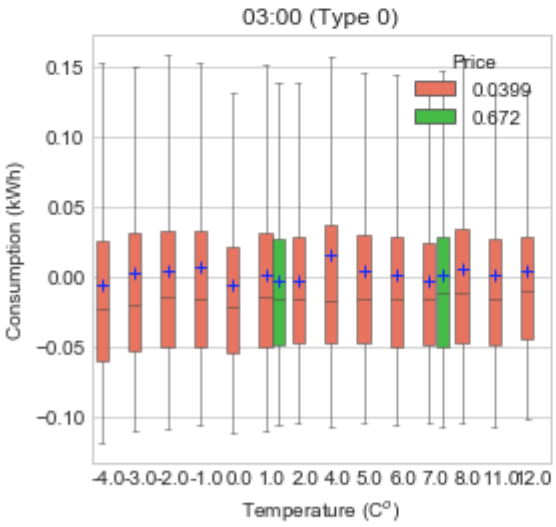

01:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

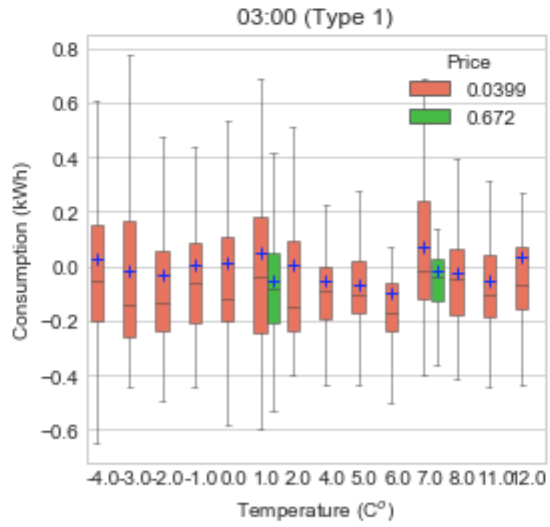

01:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

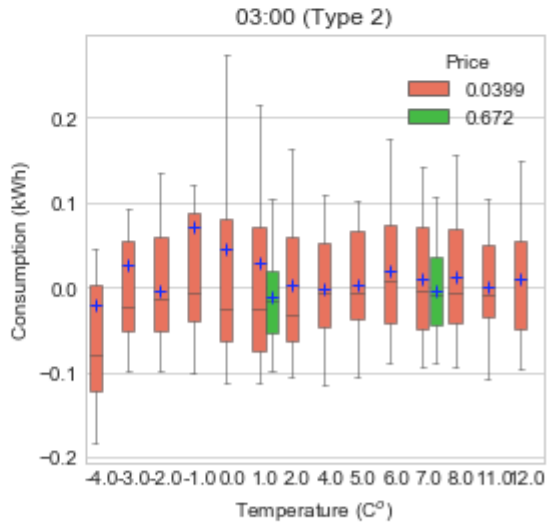

02:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

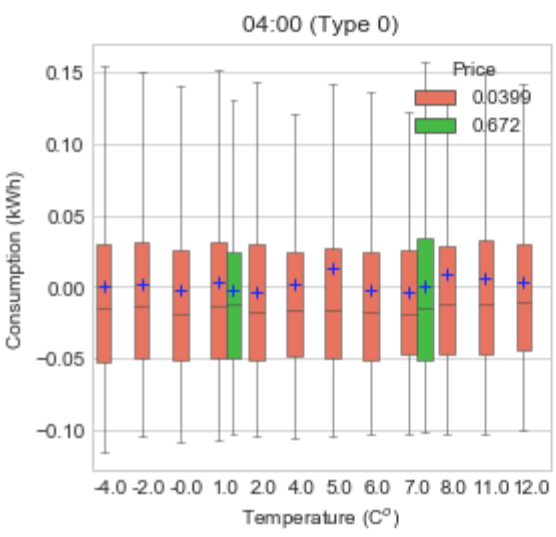

02:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

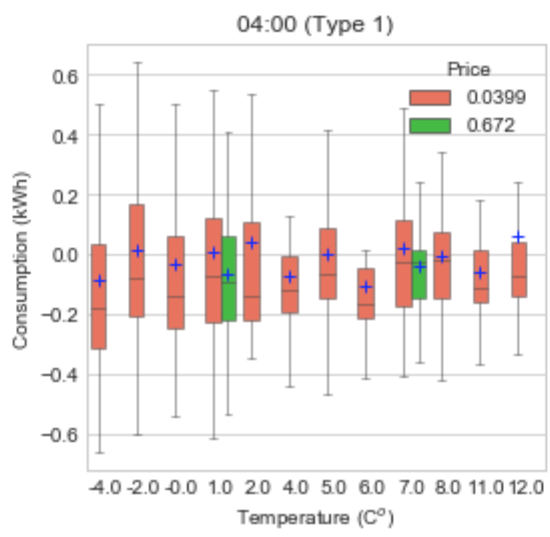

02:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

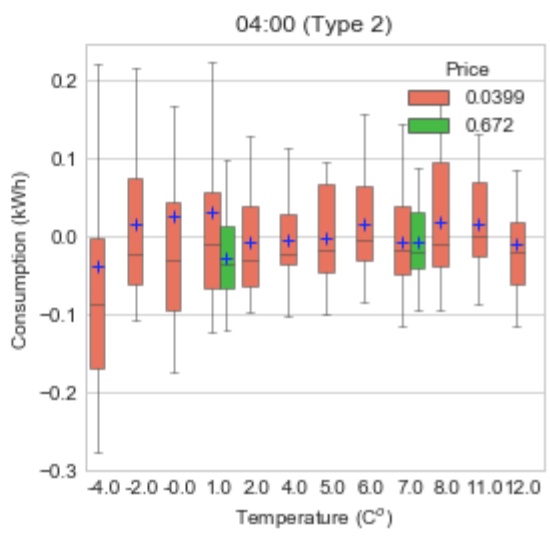

03:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

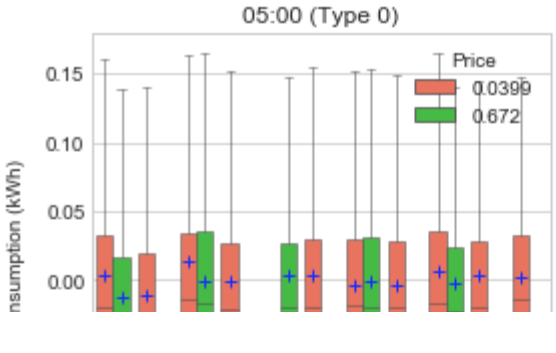

03:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

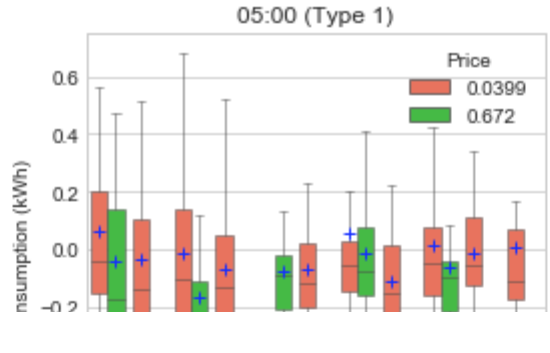

03:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

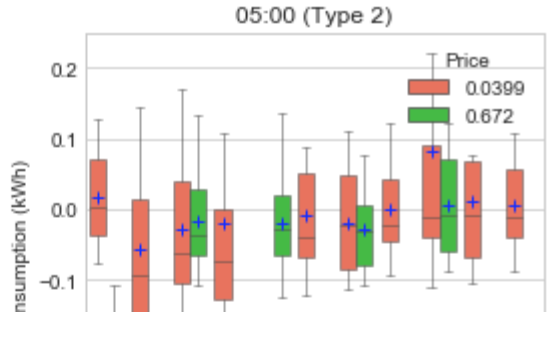

04:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

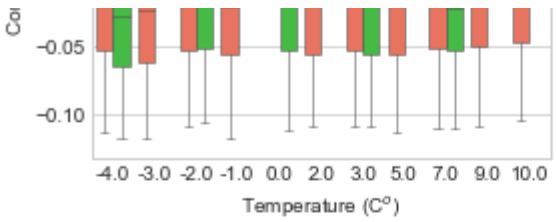

04:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

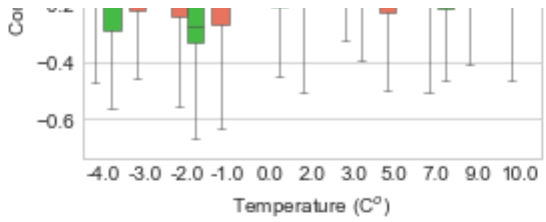

04:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

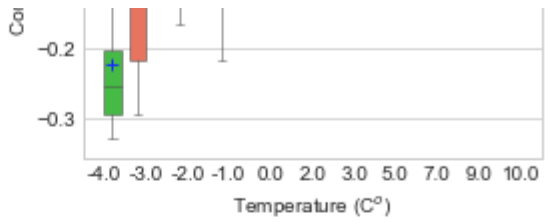

05:00 (Type 0)
Price
0.0399
0.672
Temperature (C°)

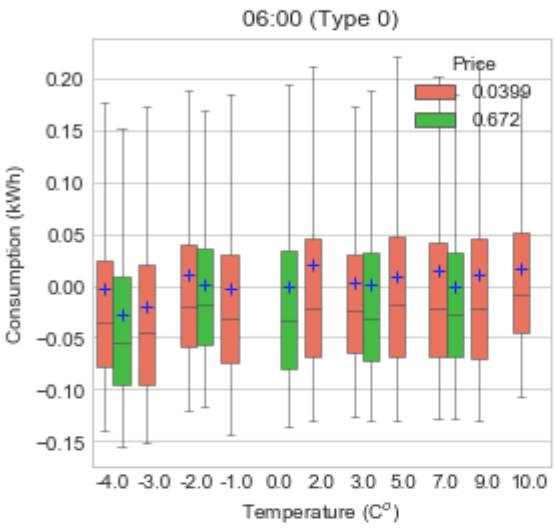

05:00 (Type 1)
Price
0.0399
0.672
Temperature (C°)

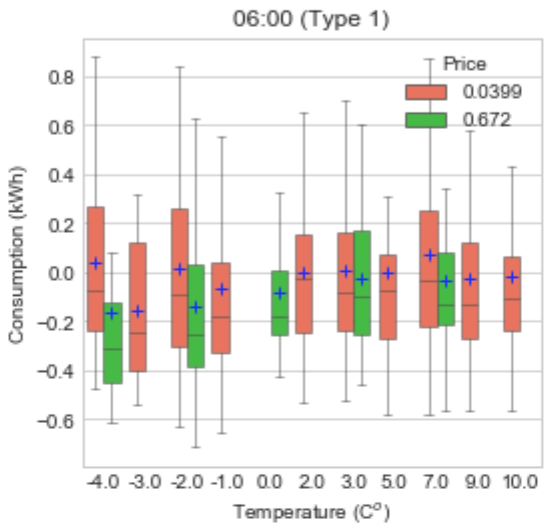

05:00 (Type 2)
Price
0.0399
0.672
Temperature (C°)

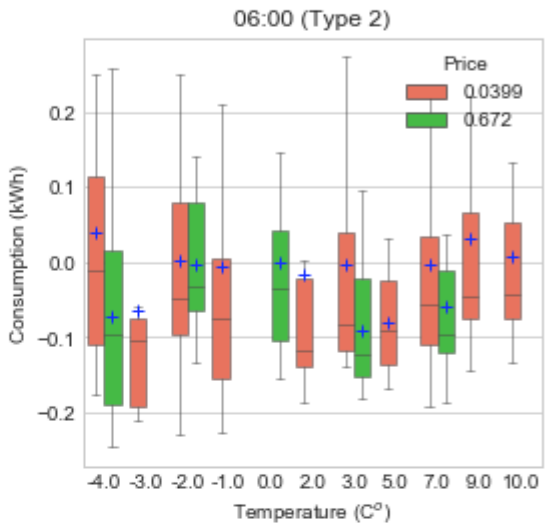

Temperature (C°)

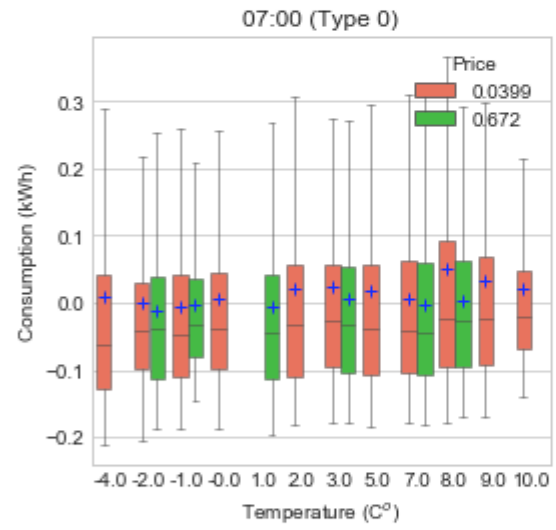

Temperature (C°)

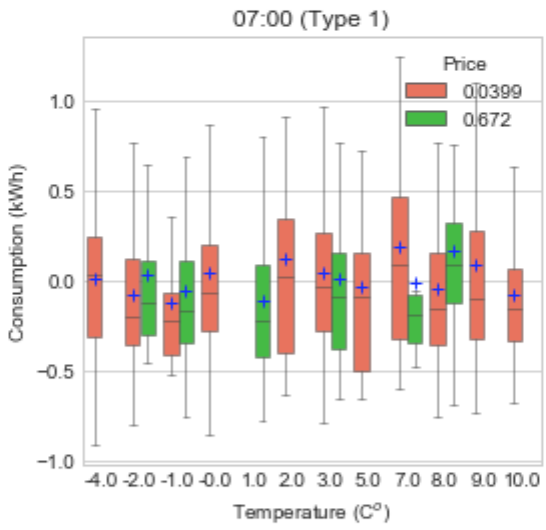

Temperature (C°)

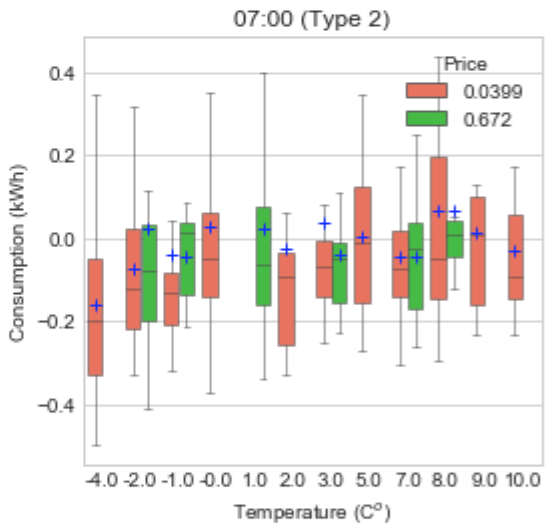

06:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

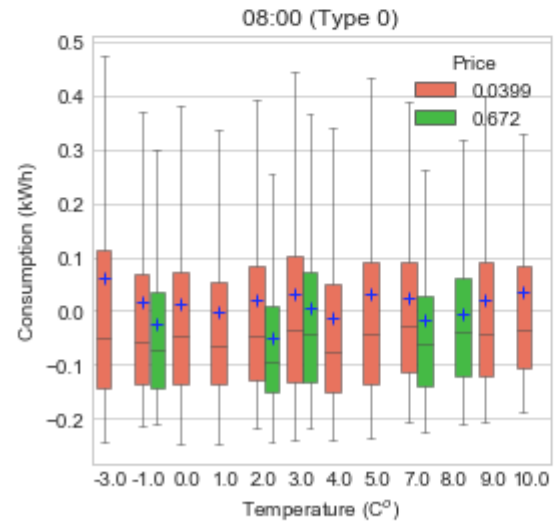

06:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

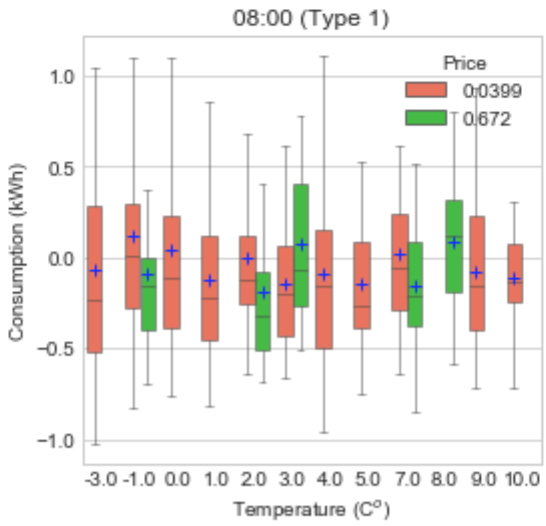

06:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

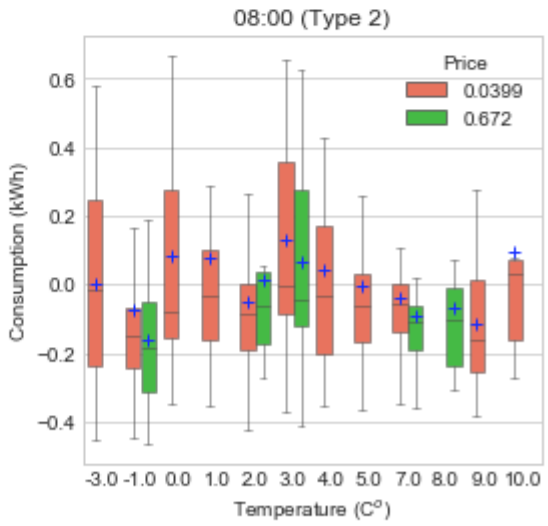

07:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

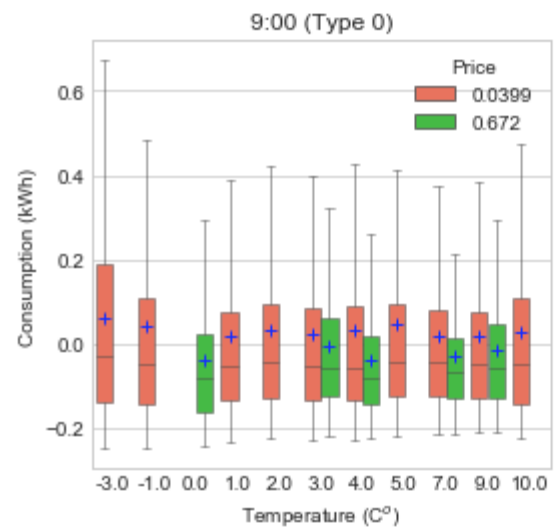

07:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

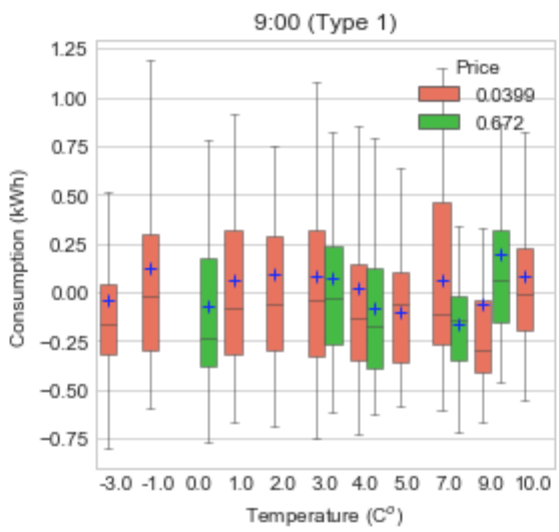

07:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

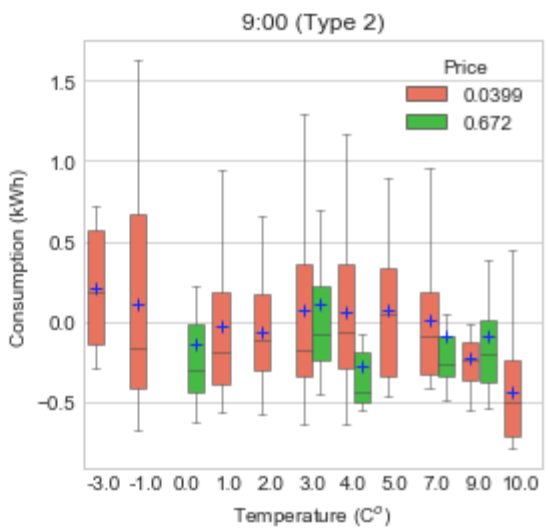

08:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

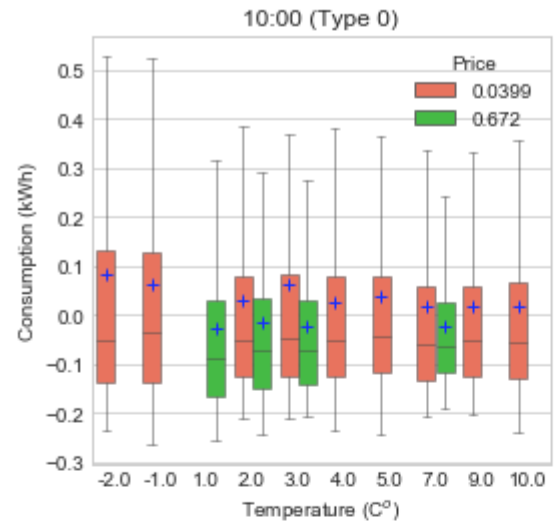

08:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

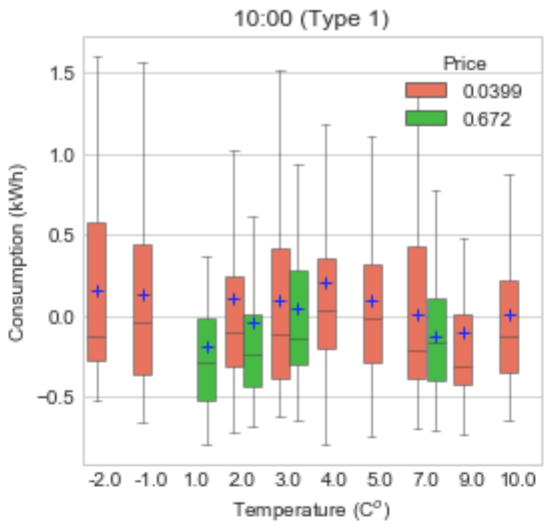

08:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

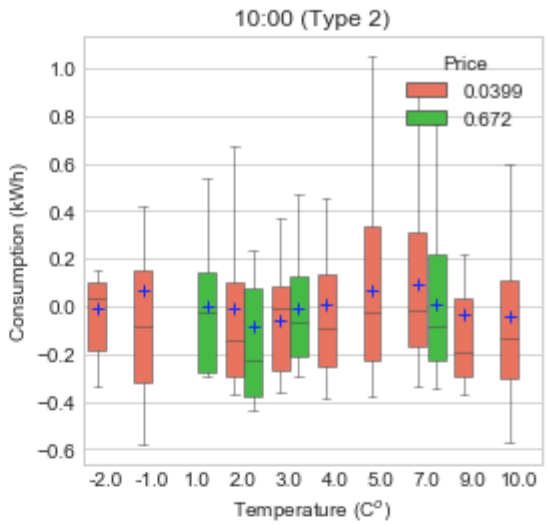

9:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

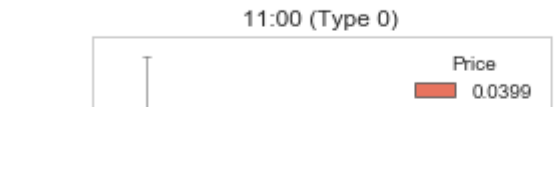

9:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

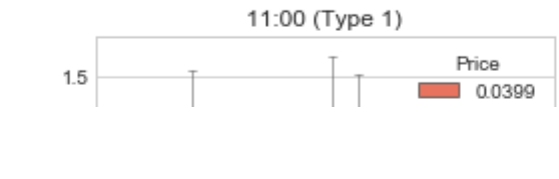

9:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

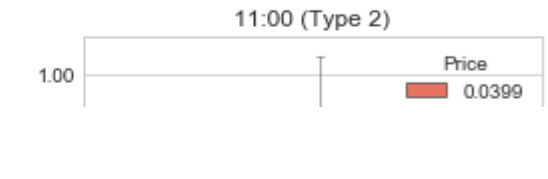

10:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

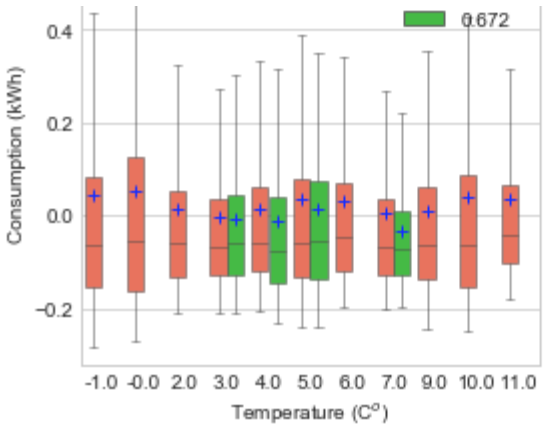

10:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

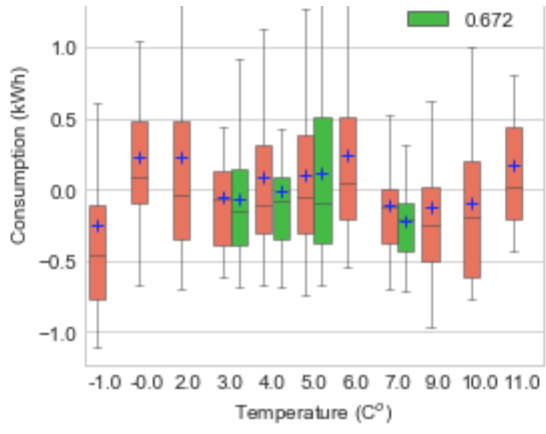

10:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

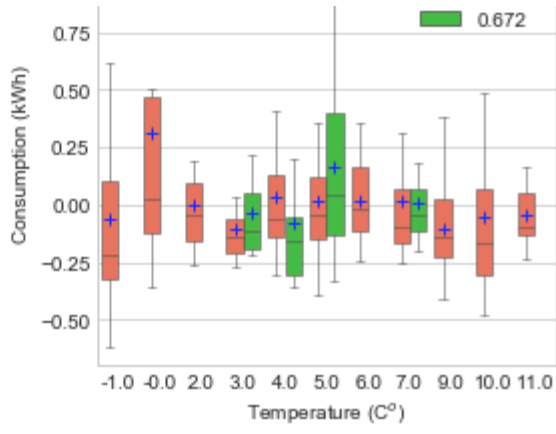

11:00 (Type 0)
Price
0.0399

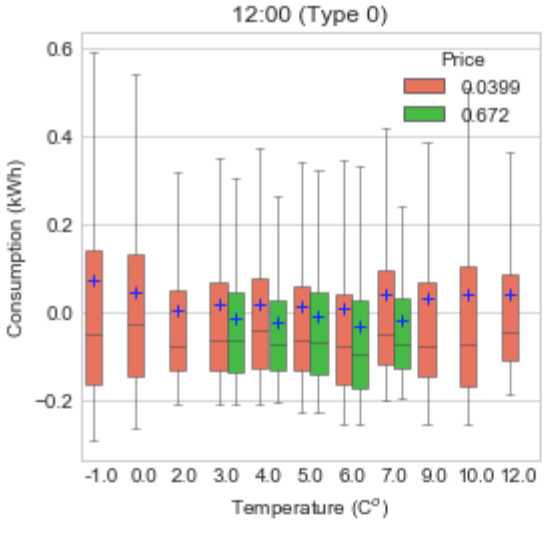

11:00 (Type 1)
Price
0.0399

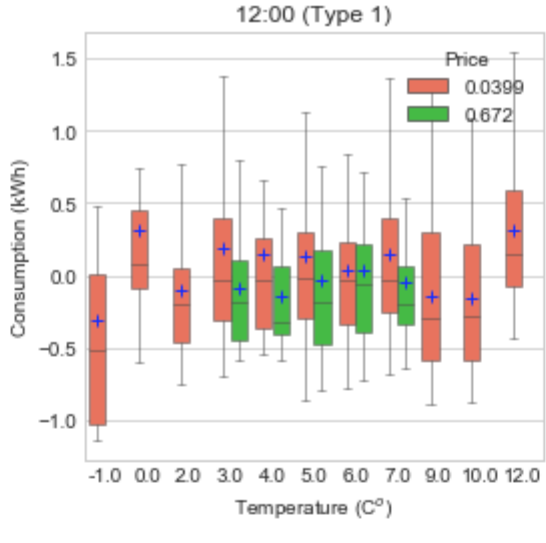

11:00 (Type 2)
Price
0.0399

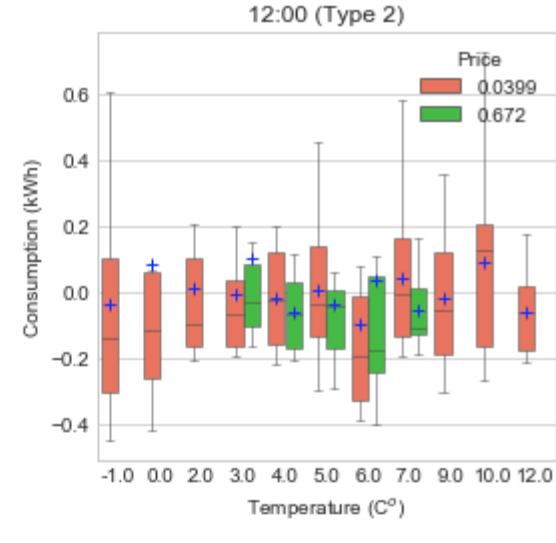

0.672
Consumption (kWh)
Temperature (C°)

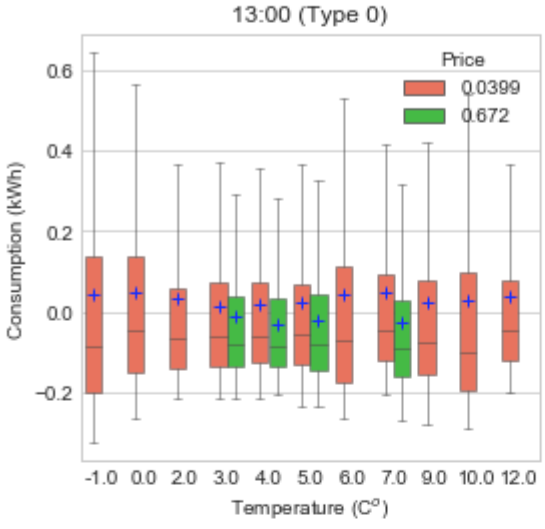

0.672
Consumption (kWh)
Temperature (C°)

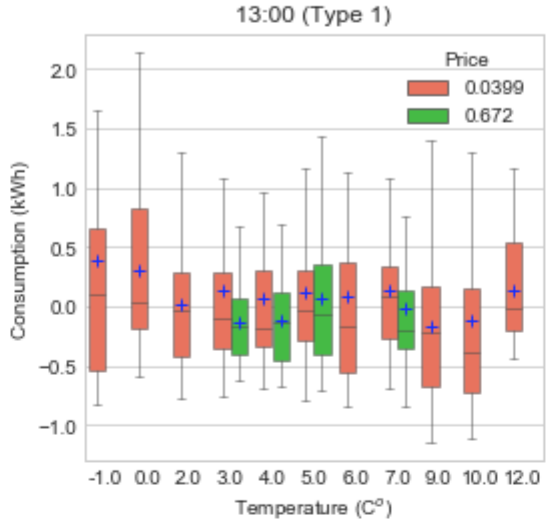

0.672
Consumption (kWh)
Temperature (C°)

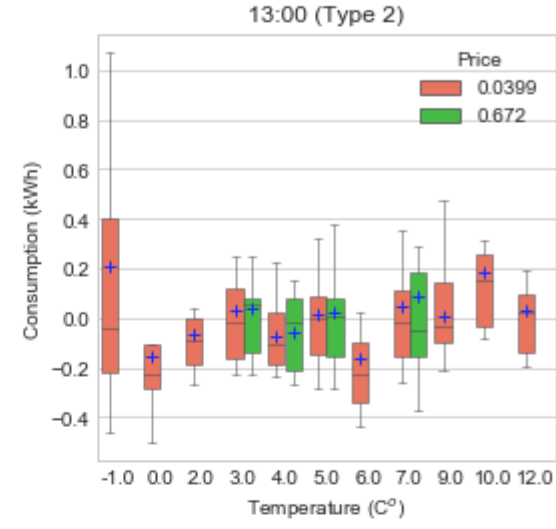

12:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

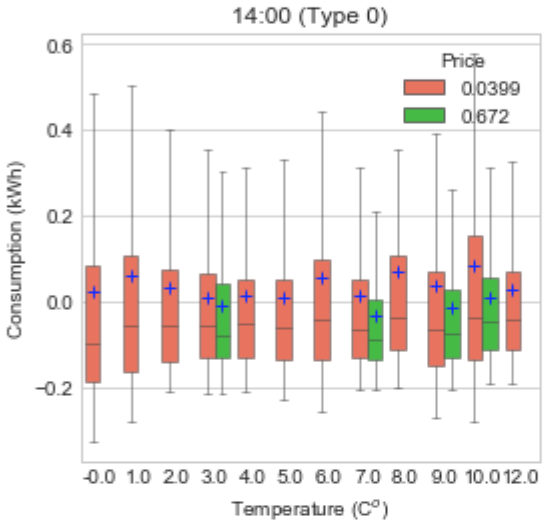

12:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

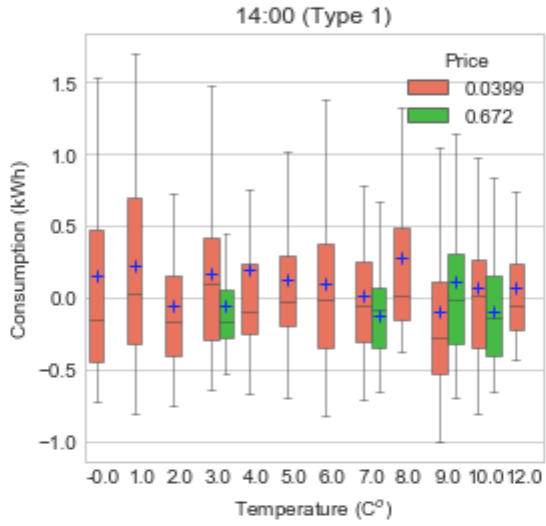

12:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

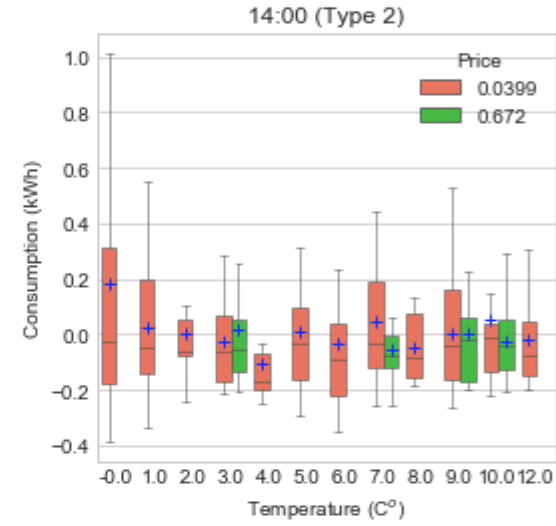

13:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

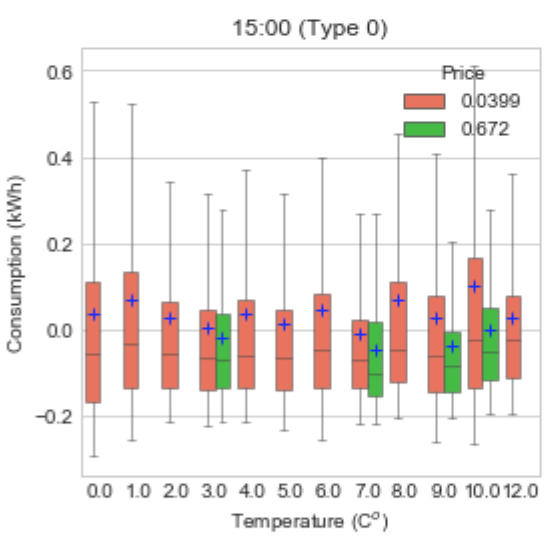

13:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

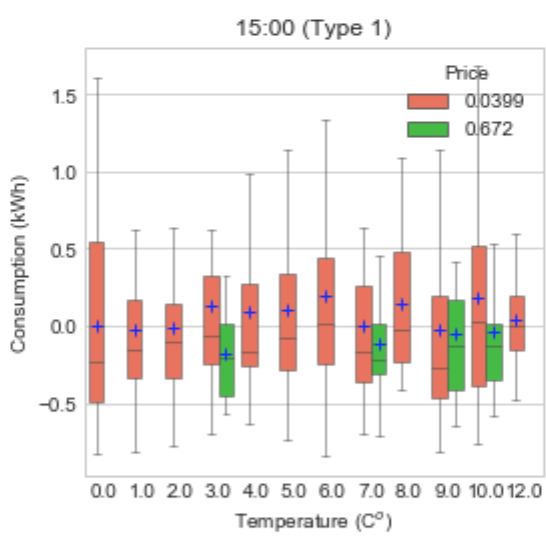

13:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

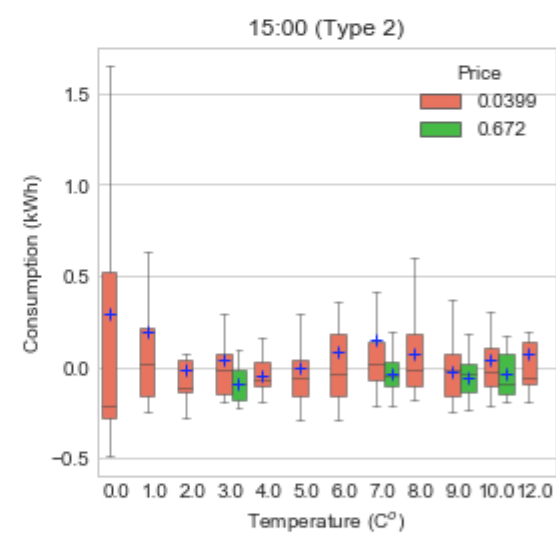

14:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

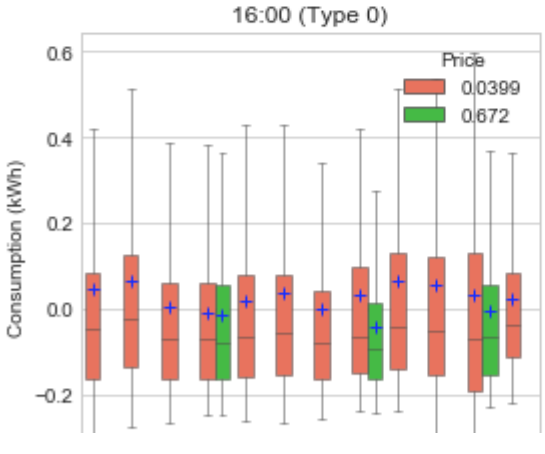

14:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

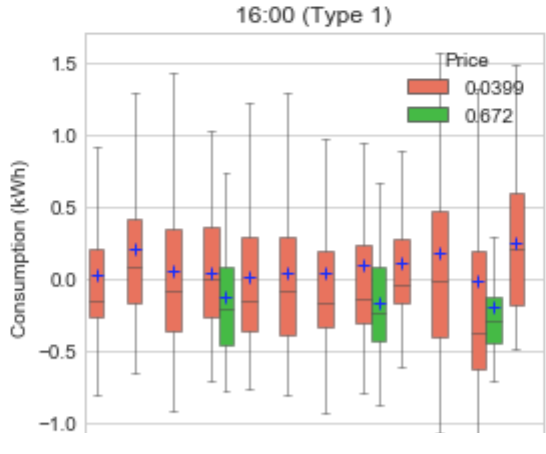

14:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

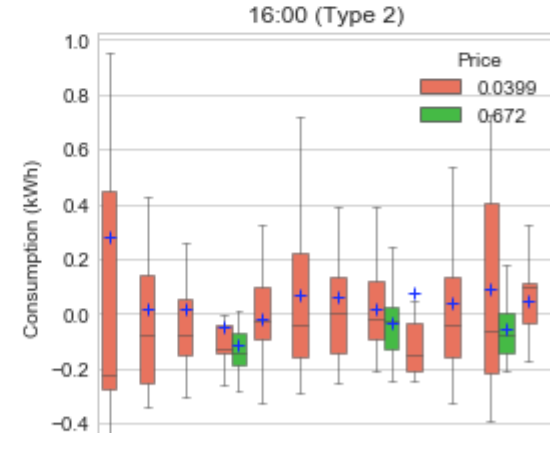

15:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

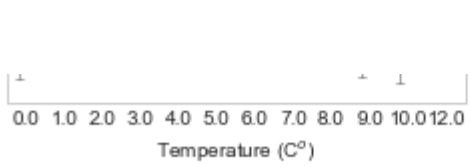

15:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

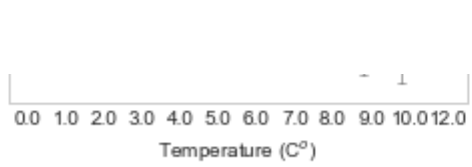

15:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
Temperature (C°)

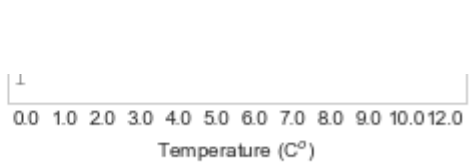

16:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)

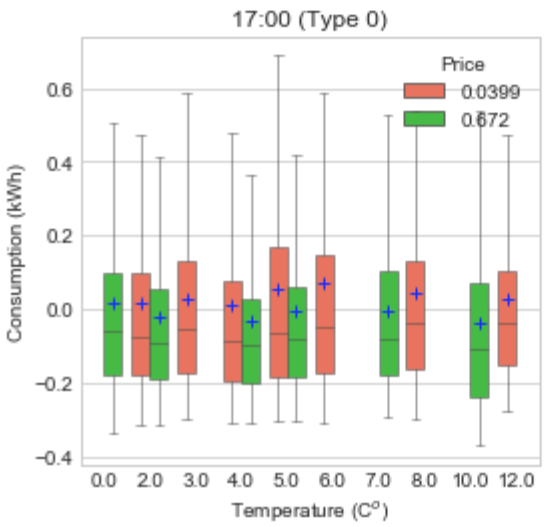

16:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)

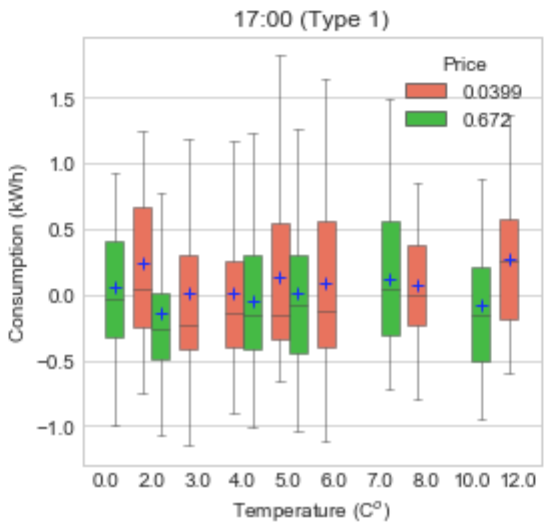

16:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)

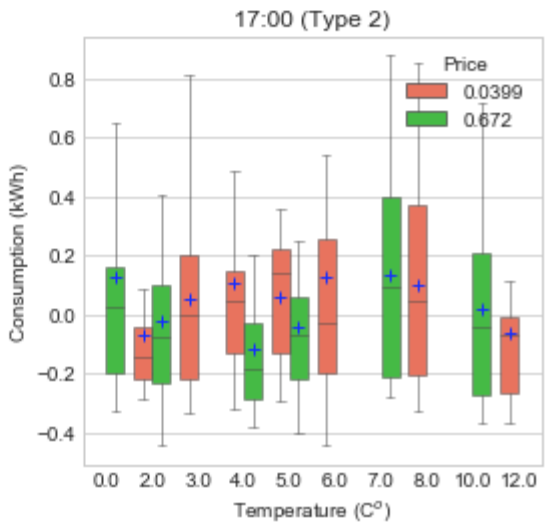

0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 12.0
Temperature (C°)

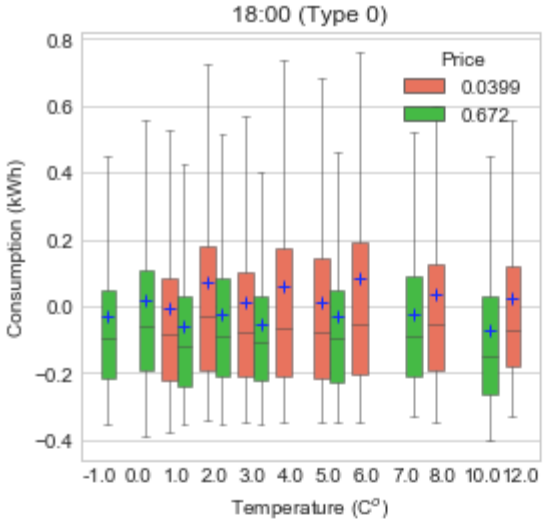

0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 12.0
Temperature (C°)

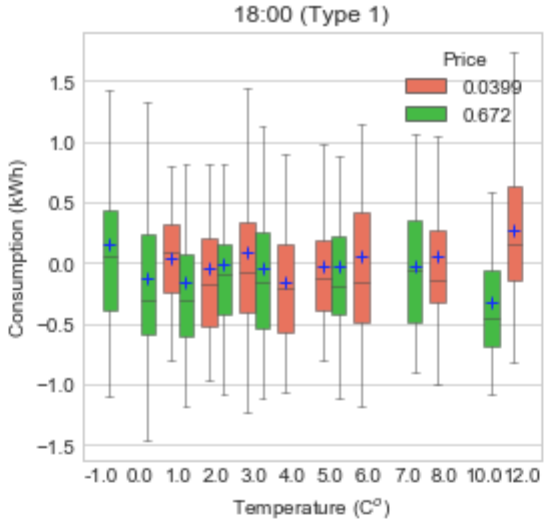

0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 12.0
Temperature (C°)

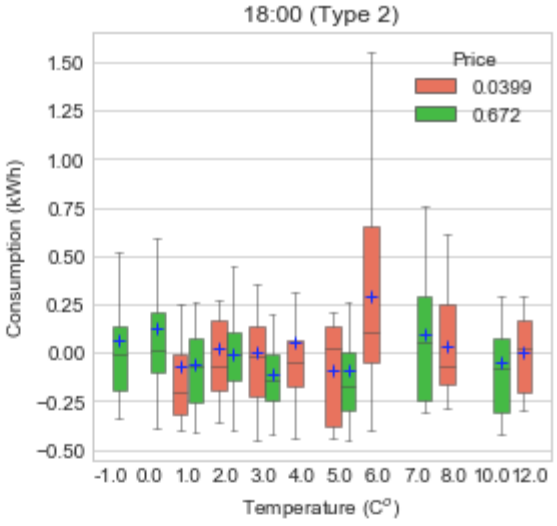

17:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

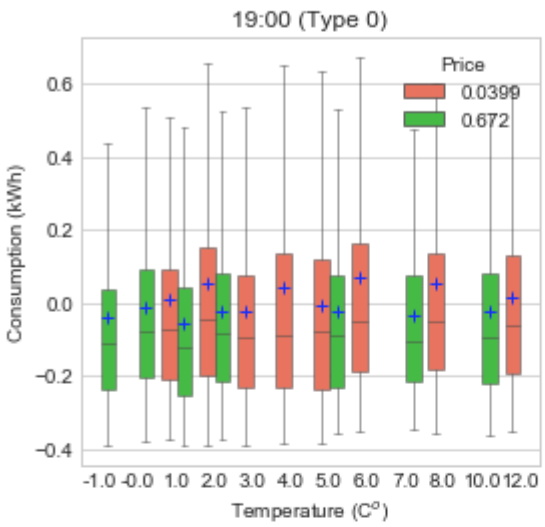

17:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

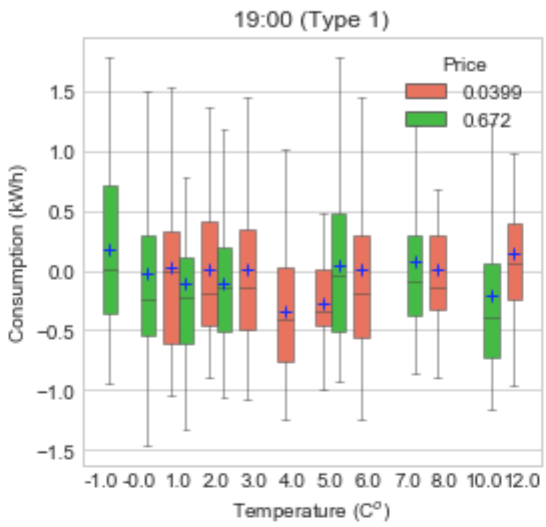

17:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

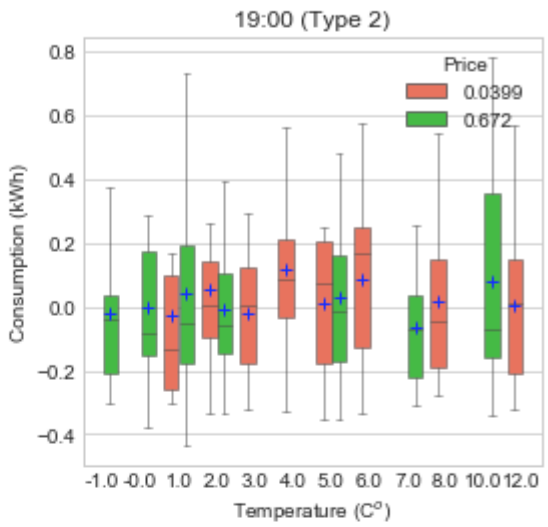

18:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-1.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

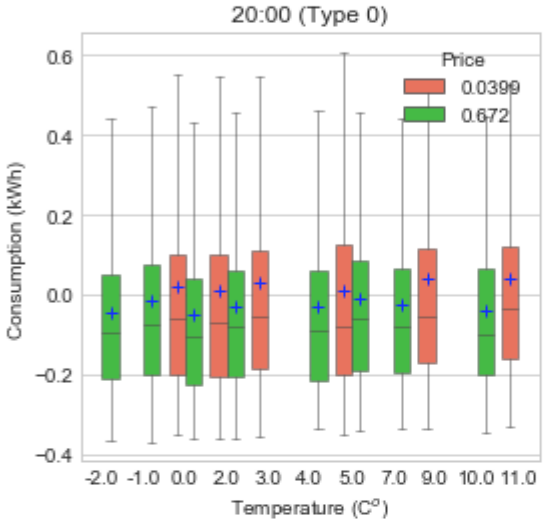

18:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
-1.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

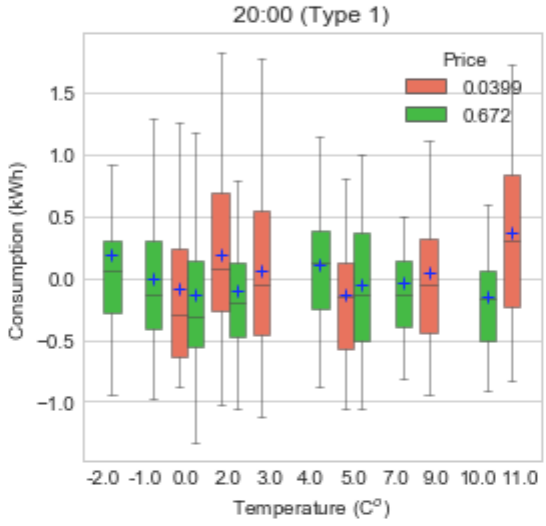

18:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
-1.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

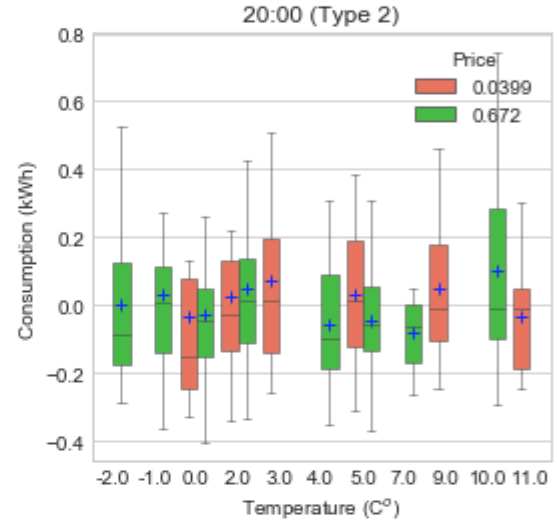

19:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-1.0 -0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

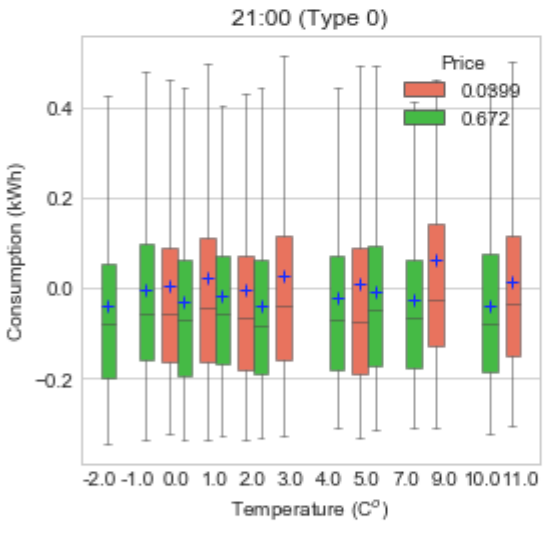

19:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
-1.0 -0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

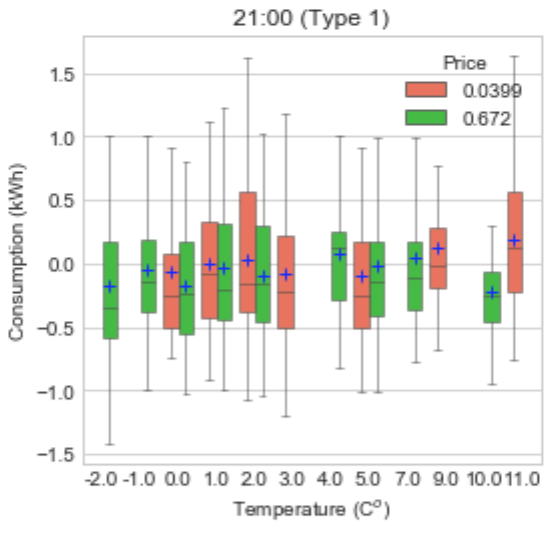

19:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-1.0 -0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0
Temperature (C°)

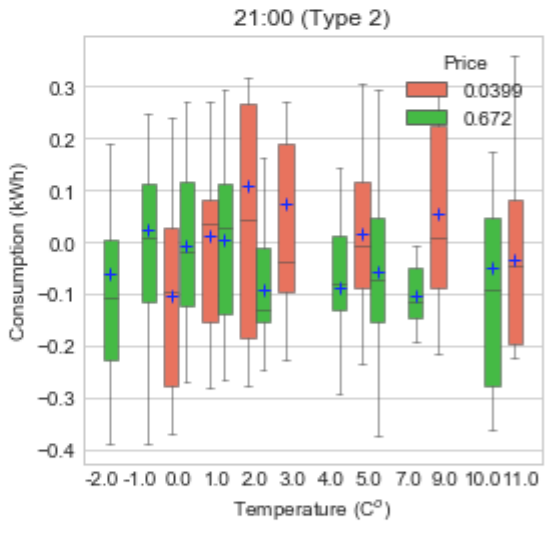

20:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0 -1.0 0.0 2.0 3.0 4.0 5.0 7.0 9.0 10.0 11.0
Temperature (C°)


20:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
-2.0 -1.0 0.0 2.0 3.0 4.0 5.0 7.0 9.0 10.0 11.0
Temperature (C°)


20:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
-2.0 -1.0 0.0 2.0 3.0 4.0 5.0 7.0 9.0 10.0 11.0
Temperature (C°)


21:00 (Type 0)
Price
0.0399
0.672
Consumption (kWh)
0.4
0.2
0.0
−0.2
-2.0 -1.0 0.0 1.0 2.0 3.0 4.0 5.0 7.0 9.0 10.0 11.0
Temperature (C°)

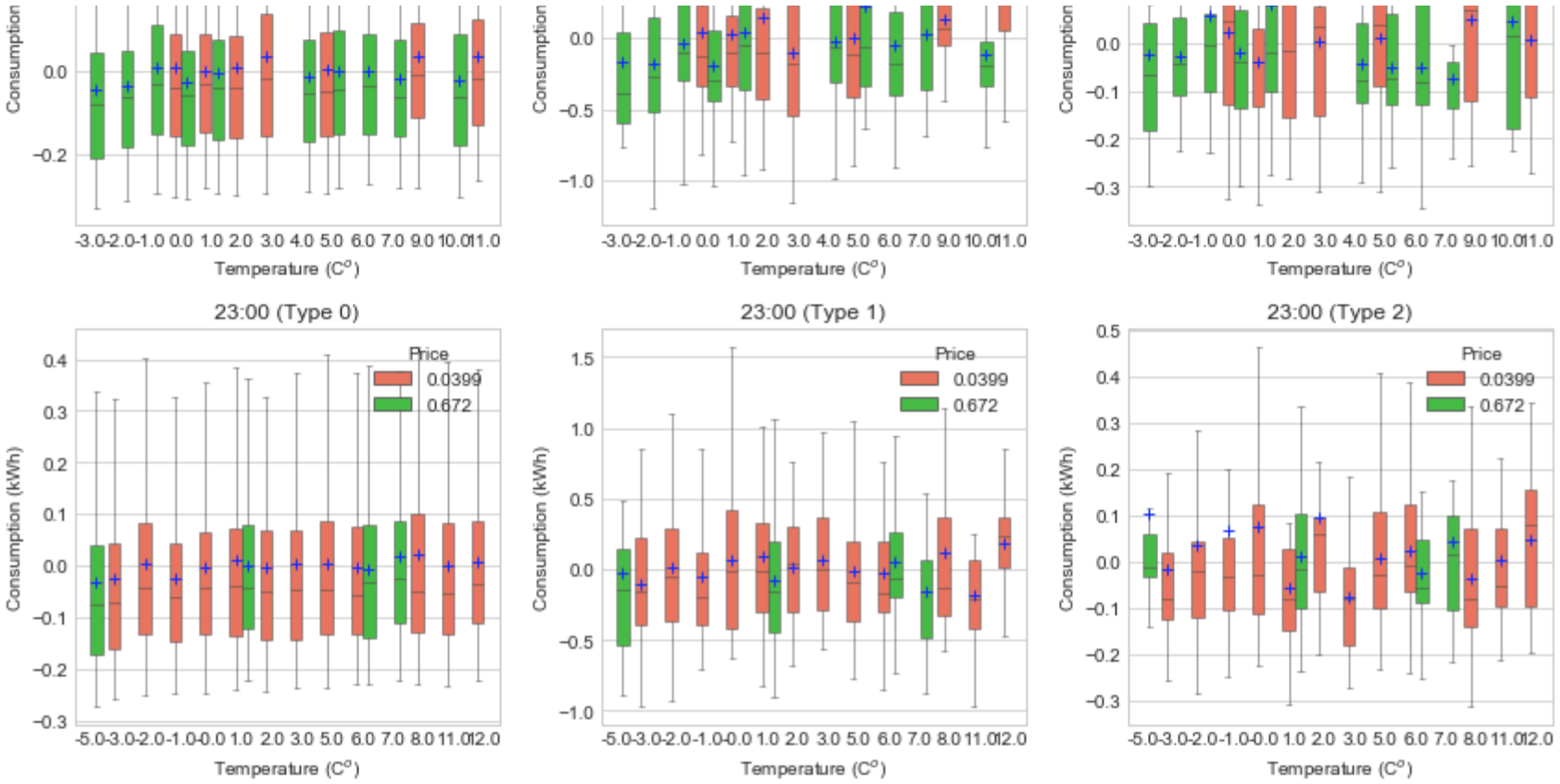

21:00 (Type 1)
Price
0.0399
0.672
Consumption (kWh)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
-2.0 -1.0 0.0 1.0 2.0 3.0 4.0 5.0 7.0 9.0 10.0 11.0
Temperature (C°)


21:00 (Type 2)
Price
0.0399
0.672
Consumption (kWh)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
-2.0 -1.0 0.0 1.0 2.0 3.0 4.0 5.0 7.0 9.0 10.0 11.0
Temperature (C°)

22:00 (Type 0)
Price
0.0399
0.672
0.4
0.2

22:00 (Type 1)
Price
0.0399
0.672
1.0
0.5

22:00 (Type 2)
Price
0.0399
0.672
0.4
0.3
0.2
0.1



When design DR plans, also need to pay attention on the day of weeks effects, otherwise if only use temperature, it could lead to oppositie direction of responsiveness.

See 22:00 low price picture, it's clear that when temperature is higher the reponsiveness to low price increases.

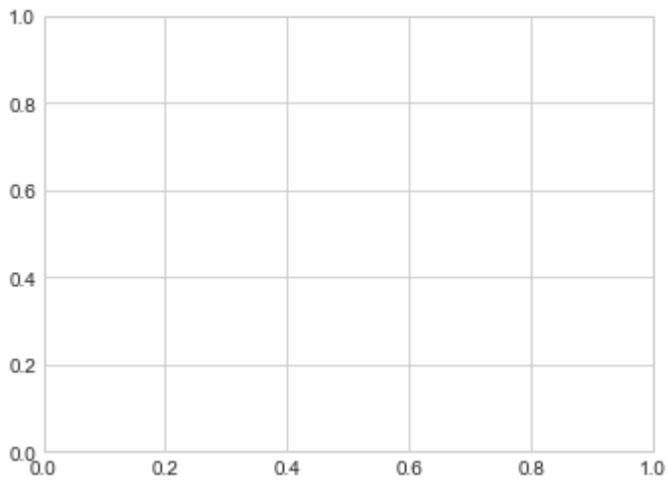
When temperature is low enough, high price won't stop people consume energy

e) Day of week: prices

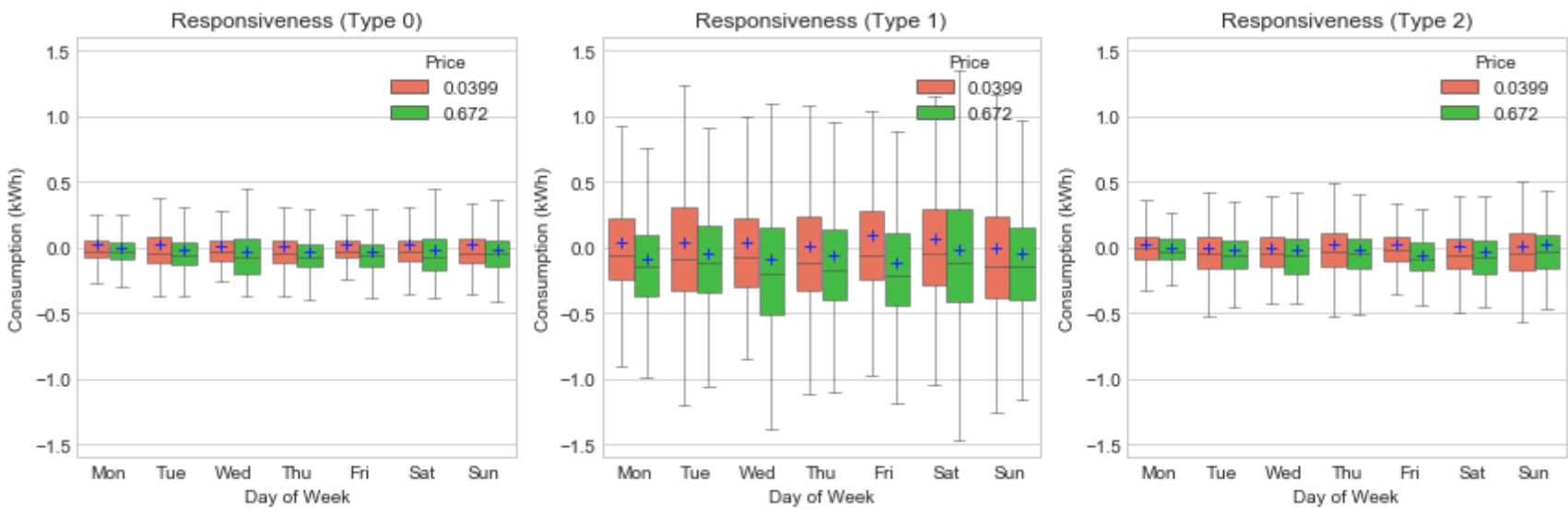
In [15]:
```
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,4))
ax_Ntou = [] # store subplot objects
weeks = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for p in range(2): # two price levels
    ax_Ntou.append(fig_all.add_subplot(1, 2, (p + 1)))
    df_day = df_all_res_long[df_all_res_long['Price'] == prices[p]]
    if df_day.shape[0] >= 1:
        sns.boxplot(x="Day of week", y="Consumption", hue="Household type", data=df_day, ax=ax_Ntou[-1], pale
tte=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerf
acecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
        ax_Ntou[-1].set_title('Responsiveness'  + ' (Price ' + str(prices[p]) + ')' )
        ax_Ntou[-1].set_xlabel(r'Day of Week')
        ax_Ntou[-1].set_xticklabels(weeks)
        ax_Ntou[-1].set_ylabel('Consumption (kWh)')
        l = ax_Ntou[-1].legend()
        l.set_title('Household Type')
        for i,artist in enumerate(ax_Ntou[-1].artists):
            artist.set_linewidth(0.5)
            # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
            # Loop over them here, and use the same colour as above
            for j in range(i*6,i*6+6):
                line = ax_Ntou[-1].lines[j]
                line.set_linewidth(0.5)
        ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-15-1cb817ecf84a> in <module>()
      7 for p in range(2): # two price levels
      8     ax_Ntou.append(fig_all.add_subplot(1, 2, (p + 1)))
----> 9     df_day = df_all_res_long[df_all_res_long['Price'] == prices[p]]
     10     if df_day.shape[0] >= 1:
     11         sns.boxplot(x="Day of week", y="Consumption", hue="Household type", data=df_day, ax=ax_Ntou[
-1], palette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":
"+","markerfacecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})

NameError: name 'df_all_res_long' is not defined
```



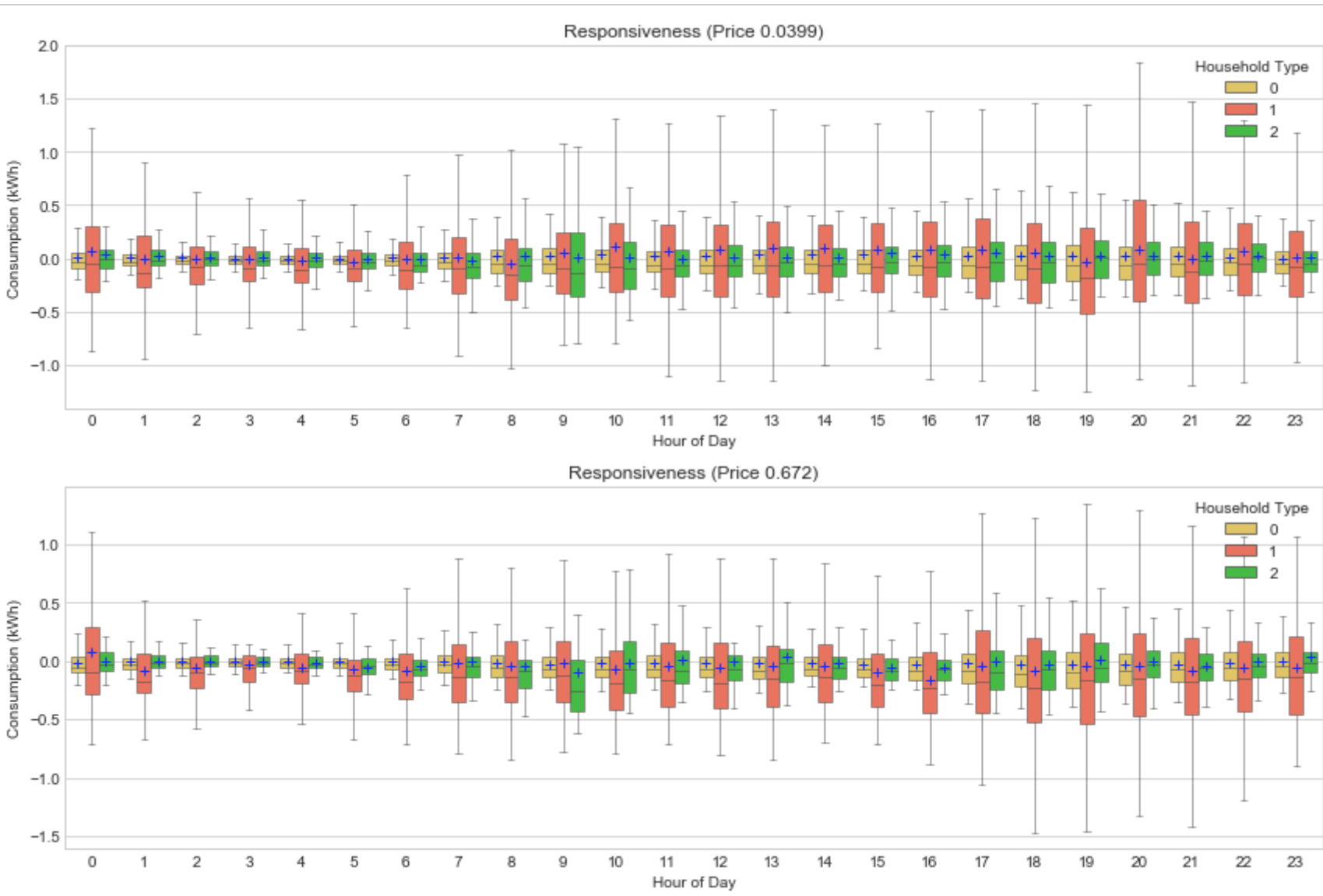
In [145]:
```
# price comparison
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,4))
ax_Ntou = [] # store subplot objects
weeks = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for g in range(3): # three types
    ax_Ntou.append(fig_all.add_subplot(1, 3, (g + 1)))
    df_day = df_all_res_long[df_all_res_long['Household type'] == g]
    if df_day.shape[0] >= 1:
        sns.boxplot(x="Day of week", y="Consumption", hue="Price", data=df_day, ax=ax_Ntou[-1], palette=['tom
ato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:vivid
 blue", "markeredgecolor":"xkcd:vivid blue"})
        ax_Ntou[-1].set_title('Responsiveness'  + ' (Type ' + str(g) + ')' )
        ax_Ntou[-1].set_xlabel(r'Day of Week')
        ax_Ntou[-1].set_xticklabels(weeks)
        ax_Ntou[-1].set_ylabel('Consumption (kWh)')
#         ax_Ntou[-1].set(ylim = (-1.6, 1.6))
        l = ax_Ntou[-1].legend()
        l.set_title('Price')
        for i,artist in enumerate(ax_Ntou[-1].artists):
            artist.set_linewidth(0.5)
            # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
            # Loop over them here, and use the same colour as above
            for j in range(i*6,i*6+6):
                line = ax_Ntou[-1].lines[j]
                line.set_linewidth(0.5)
        ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```



By comparing the median change from 0 in two pictures, we find people are more sensitive to the high price than low price, because in high price the median has obivious drops.

f) Hour of day: prices

In [148]:
```
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,8))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for p in range(2): # two price levels
    ax_Ntou.append(fig_all.add_subplot(2, 1, (p + 1)))
    df_hour = df_all_res_long[df_all_res_long['Price'] == prices[p]]
    if df_hour.shape[0] >= 1:
        sns.boxplot(x="Hour of day", y="Consumption", hue="Household type", data=df_hour, ax=ax_Ntou[-1], pal
ette=['xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","marker
facecolor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
        ax_Ntou[-1].set_title('Responsiveness'  + ' (Price ' + str(prices[p]) + ')' )
        ax_Ntou[-1].set_xlabel(r'Hour of Day')
        ax_Ntou[-1].set_ylabel('Consumption (kWh)')
        l = ax_Ntou[-1].legend()
#         ax_Ntou[-1].yaxis.grid(True)
#         ax_Ntou[-1].xaxis.grid(True)
        l.set_title('Household Type')
        for i,artist in enumerate(ax_Ntou[-1].artists):
            artist.set_linewidth(0.5)
            # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
            # Loop over them here, and use the same colour as above
            for j in range(i*6,i*6+6):
                line = ax_Ntou[-1].lines[j]
                line.set_linewidth(0.5)
        ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```



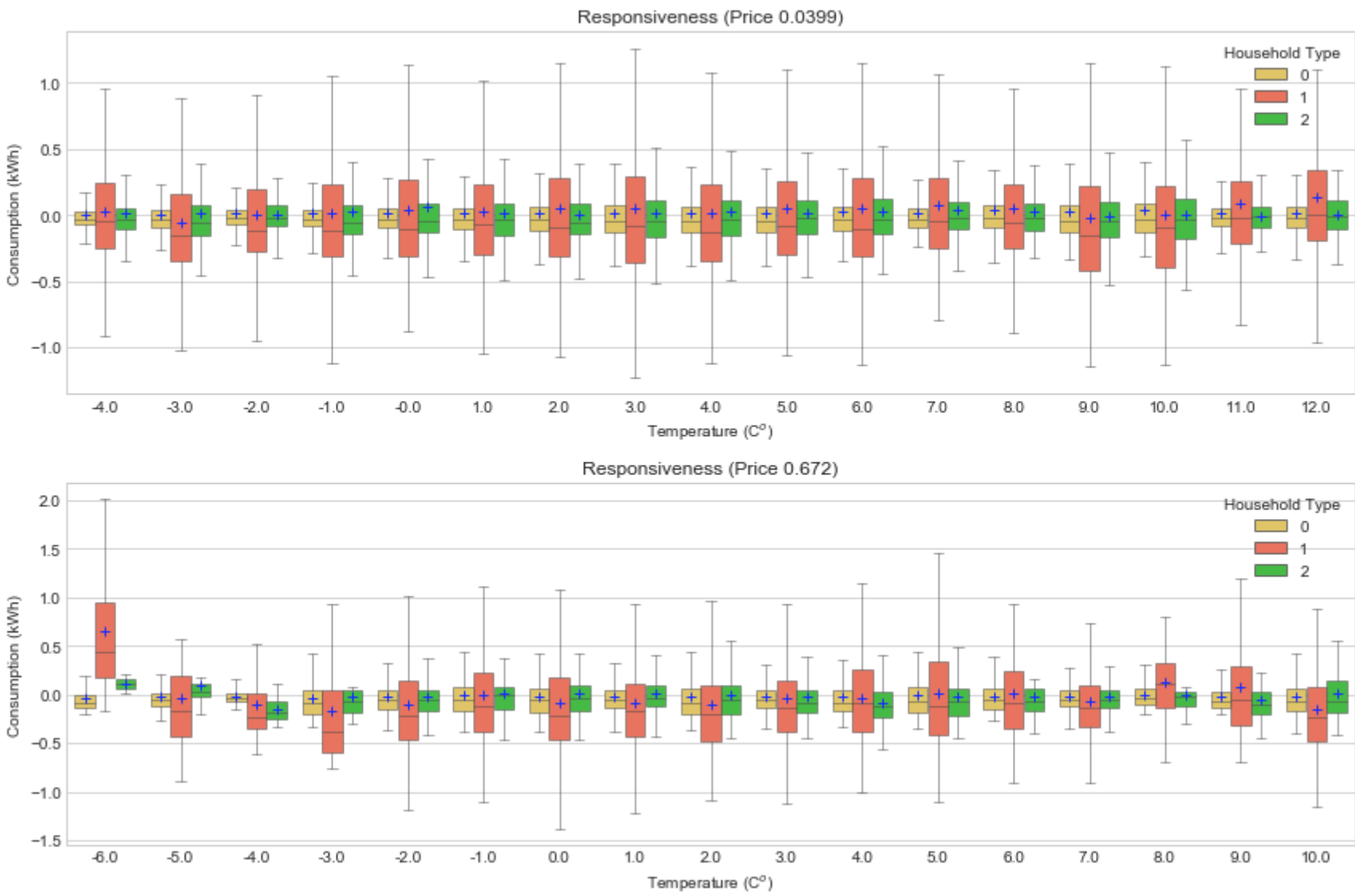
In [150]:
```
# price comparison
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,12))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for g in range(3): # three types
    ax_Ntou.append(fig_all.add_subplot(3, 1, (g + 1)))
    df_hour = df_all_res_long[df_all_res_long['Household type'] == g]
    if df_hour.shape[0] >= 1:
        sns.boxplot(x="Hour of day", y="Consumption", hue="Price", data=df_hour, ax=ax_Ntou[-1], palette=['to
mato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:vivid
blue", "markeredgecolor":"xkcd:vivid blue"})
        ax_Ntou[-1].set_title('Responsiveness'  + ' (Type ' + str(g) + ')' )
        ax_Ntou[-1].set_xlabel(r'Hour of Day')
        ax_Ntou[-1].set_ylabel('Consumption (kWh)')
        l = ax_Ntou[-1].legend()
#         ax_Ntou[-1].yaxis.grid(True)
#         ax_Ntou[-1].xaxis.grid(True)
        l.set_title('Price')
        for i,artist in enumerate(ax_Ntou[-1].artists):
            artist.set_linewidth(0.5)
            # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
            # Loop over them here, and use the same colour as above
            for j in range(i*6,i*6+6):
                line = ax_Ntou[-1].lines[j]
                line.set_linewidth(0.5)
        ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```


In a day, for low price, the max price responsivenss happens around 1pm (noon) and 8pm (evening); for high price the max responsiveness happens around 1pm and 6-7pm (evening).

Midnight doesn't have too much responsiveness potential. Also high price has more impact on the median behvaior for all household groups

g) Temperature: prices

In [109]:
```
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,8))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for p in range(2): # two price levels
    ax_Ntou.append(fig_all.add_subplot(2, 1, (p + 1)))
    df_temp = df_all_res_long[df_all_res_long['Price'] == prices[p]]
    if df_temp.shape[0] >= 1:
        sns.boxplot(x="TempC", y="Consumption", hue="Household type", data=df_temp, ax=ax_Ntou[-1], palette=[
'xkcd:maize','tomato','limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfaceco
lor":"xkcd:vivid blue", "markeredgecolor":"xkcd:vivid blue"})
        ax_Ntou[-1].set_title('Responsiveness'  + ' (Price ' + str(prices[p]) + ')' )
        ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
        ax_Ntou[-1].set_ylabel('Consumption (kWh)')
#         ax_Ntou[-1].set(ylim = (-1.6, 1.6))
#         ax_Ntou[-1].yaxis.grid(True)
#         ax_Ntou[-1].xaxis.grid(True)
        l = ax_Ntou[-1].legend()
        l.set_title('Household Type')
        for i,artist in enumerate(ax_Ntou[-1].artists):
            artist.set_linewidth(0.5)
            # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
            # Loop over them here, and use the same colour as above
            for j in range(i*6,i*6+6):
                line = ax_Ntou[-1].lines[j]
                line.set_linewidth(0.5)
        ax_Ntou[-1].legend(loc = 'upper right').set_title('Household Type')
plt.tight_layout()
```



In [152]:
```
# price comparison
# price responsiveness under different tempertures over days of week and prices
fig_all = plt.figure(figsize = (12,12))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.6720]
sns.set_style("whitegrid")
for g in range(3): # three types
    ax_Ntou.append(fig_all.add_subplot(3, 1, (g + 1)))
    df_temp = df_all_res_long[df_all_res_long['Household type'] == g]
    if df_temp.shape[0] >= 1:
        sns.boxplot(x="TempC", y="Consumption", hue="Price", data=df_temp, ax=ax_Ntou[-1], palette=['tomato',
'limegreen'], showfliers = False, showmeans=True, meanprops={"marker":"+","markerfacecolor":"xkcd:vivid blue"
, "markeredgecolor":"xkcd:vivid blue"})
        ax_Ntou[-1].set_title('Responsiveness'  + ' (Type ' + str(g) + ')' )
        ax_Ntou[-1].set_xlabel(r'Temperature (C$^o$)')
        ax_Ntou[-1].set_ylabel('Consumption (kWh)')
#         ax_Ntou[-1].set(ylim = (-1.6, 1.6))
#         ax_Ntou[-1].yaxis.grid(True)
#         ax_Ntou[-1].xaxis.grid(True)
        l = ax_Ntou[-1].legend()
        l.set_title('Price')
        for i,artist in enumerate(ax_Ntou[-1].artists):
            artist.set_linewidth(0.5)
            # Each box has 6 associated Line2D objects (to make the whiskers, fliers, etc.)
            # Loop over them here, and use the same colour as above
            for j in range(i*6,i*6+6):
                line = ax_Ntou[-1].lines[j]
                line.set_linewidth(0.5)
        ax_Ntou[-1].legend(loc = 'upper right').set_title('Price')
plt.tight_layout()
```


Still higher price shows more impacts on behavior's change. Also, when temperate is lower to -6 celcius degree, the household type 1 increases their energy usage a significant amount no matter the high price.

Again the last picture verify our point that temperature is a necessary concern of DR plan. See the value when temperature is at -6 degree, without considering the comparision under same temperature level, we probably feel that people won't be affected by the price when they are at lower temperature. But the more accurate understanding should be we don't have similar data for low price with same temperature, so it could be price also play a role under such case, i.e., price may also affect people's behavior under such low temperature, it's just we don't have enough experients and evidence, which doesn't mean we can ignore it or rush to a uncertain conclusion.

Additional info

Consumption variance of different type of households over 24 hours (Cold Season).

In [62]:
```
# type 0 event time consumption prediction
pred_result = []
for i in range(df_type0.shape[0]):
    pred_result.append(predict(df_regression_0, df_type0['Day of week'].iloc[i], df_type0['Hour of day'].iloc
[i], df_type0['TempC'].iloc[i]))
df_type0['Predicted consumption'] = pred_result
# price responsiveness elementwise
df_type0_con = df_type0.copy() # store the consumption not the response, just for this additional test
# df_type0_con.iloc[:,1:houseGroupDf.counts.iloc[0] + 1] = df_type0.iloc[:,1:houseGroupDf.counts.iloc[0] +
 1].sub(df_type0['Predicted consumption'], axis = 0)

# type 1 event time consumption prediction
pred_result = []
for i in range(df_type1.shape[0]):
    pred_result.append(predict(df_regression_1, df_type1['Day of week'].iloc[i], df_type1['Hour of day'].iloc
[i], df_type1['TempC'].iloc[i]))
df_type1['Predicted consumption'] = pred_result

# price responsiveness elementwise
df_type1_con = df_type1.copy()
# df_type1_con.iloc[:,1:houseGroupDf.counts.iloc[1] + 1] = df_type1.iloc[:,1:houseGroupDf.counts.iloc[1] +
 1].sub(df_type1['Predicted consumption'], axis = 0)

# type 2 event time consumption prediction
pred_result = []
for i in range(df_type2.shape[0]):
    pred_result.append(predict(df_regression_2, df_type2['Day of week'].iloc[i], df_type2['Hour of day'].iloc
[i], df_type2['TempC'].iloc[i]))
df_type2['Predicted consumption'] = pred_result

# price responsiveness elementwise
df_type2_con = df_type2.copy()
# df_type2_con.iloc[:,1:houseGroupDf.counts.iloc[2] + 1] = df_type2.iloc[:,1:houseGroupDf.counts.iloc[2] +
 1].sub(df_type2['Predicted consumption'], axis = 0)
```

In [42]:
```
df_type0
```

Out[42]:

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024	TempC	TempF	Price	Eve
0	2013-01-04 14:00:00	0.102	0.239	0.192	0.148	0.018	0.592	0.253	0.069	0.087	...	0.013	0.132	10.333333	51.000000	0.0399	L3
1	2013-01-04 15:00:00	0.097	1.016	0.644	0.152	0.030	1.029	0.928	0.062	0.361	...	0.032	0.087	10.000000	51.000000	0.0399	L3
2	2013-01-04 16:00:00	1.361	0.314	0.544	0.595	0.026	0.783	0.737	0.056	0.634	...	0.225	0.186	9.333333	49.333333	0.0399	L3
3	2013-01-07 23:00:00	0.314	0.344	0.299	0.256	0.026	0.569	0.119	0.097	0.525	...	0.126	0.467	7.000000	44.000000	0.6720	H3
4	2013-01-08 00:00:00	0.098	0.229	0.277	0.269	0.024	0.208	0.103	0.070	0.174	...	0.105	0.149	7.000000	44.000000	0.6720	H3
5	2013-01-08 01:00:00	0.099	0.167	0.193	0.174	0.032	0.152	0.104	0.067	0.098	...	0.132	0.139	6.666667	43.333333	0.6720	H3
6	2013-01-10 02:00:00	0.097	0.121	0.841	0.174	0.023	0.099	0.100	0.072	0.072	...	0.031	0.228	1.333333	34.333333	0.0399	L3
7	2013-01-10 03:00:00	0.095	0.114	0.493	0.163	0.028	0.093	0.089	0.070	0.091	...	0.014	0.197	1.000000	34.000000	0.0399	L3
8	2013-01-10 04:00:00	0.220	0.074	0.306	0.201	0.020	0.195	0.107	0.070	0.055	...	0.057	0.184	1.000000	34.333333	0.0399	L3
9	2013-01-11 11:00:00	0.105	0.199	0.217	0.322	0.020	0.105	0.156	0.071	0.101	...	0.177	0.157	2.666667	36.666667	0.6720	H3
10	2013-01-11 12:00:00	0.060	0.178	0.167	0.392	0.121	0.169	0.254	0.082	0.103	...	0.030	0.128	4.000000	39.000000	0.6720	H3
11	2013-01-11 13:00:00	0.103	0.201	0.136	0.549	0.072	0.106	0.169	0.082	0.055	...	0.018	0.351	4.000000	39.000000	0.6720	H3
12	2013-01-13 05:00:00	0.237	0.089	0.167	0.144	0.019	0.134	0.091	0.071	0.049	...	0.014	0.147	-1.666667	29.333333	0.6720	H6
13	2013-01-13 06:00:00	0.090	0.155	0.150	0.402	0.029	0.094	0.115	0.074	0.087	...	0.055	0.205	-2.000000	29.000000	0.6720	H6
14	2013-01-13 07:00:00	0.068	0.078	0.186	0.448	0.017	0.235	0.079	0.075	0.278	...	0.014	0.174	-1.333333	30.000000	0.6720	H6
15	2013-01-13 08:00:00	0.100	0.467	0.291	0.264	0.032	0.262	0.127	0.323	0.464	...	0.014	0.367	-0.666667	31.000000	0.6720	H6
16	2013-01-13 09:00:00	0.100	0.355	0.192	0.257	0.017	0.243	0.183	1.567	0.305	...	0.057	0.339	0.000000	32.000000	0.6720	H6
17	2013-01-13 10:00:00	0.072	0.377	0.195	0.324	0.029	0.185	0.283	0.284	0.304	...	0.013	0.165	0.666667	33.000000	0.6720	H6
18	2013-01-16 23:00:00	0.101	0.498	0.462	0.419	0.027	0.407	0.145	0.112	0.760	...	0.211	0.393	-5.333333	22.333333	0.6720	H3
19	2013-01-17 00:00:00	0.088	0.114	0.334	0.317	0.022	0.101	0.100	0.064	0.413	...	0.330	0.265	-6.000000	21.000000	0.6720	H3

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024	TempC	TempF	Price	Eve
20	2013-01-17 01:00:00	0.076	0.118	0.151	0.243	0.017	0.122	0.084	0.069	0.315	...	0.160	0.233	-5.333333	22.000000	0.6720	H3
21	2013-01-19 05:00:00	0.080	0.078	0.188	0.384	0.028	0.097	0.103	0.054	0.107	...	0.014	0.161	-1.666667	29.333333	0.0399	CM
22	2013-01-19 06:00:00	0.095	0.108	0.137	0.356	0.019	0.135	0.102	0.076	0.050	...	0.038	0.205	-2.000000	29.000000	0.0399	CM
23	2013-01-19 07:00:00	0.094	0.104	0.171	0.314	0.017	0.205	0.088	0.056	0.056	...	0.040	0.189	-1.666667	29.666667	0.0399	CM
24	2013-01-19 08:00:00	0.069	0.182	0.354	0.405	0.033	0.177	0.154	0.948	0.064	...	0.014	0.174	-1.333333	30.333333	0.0399	CM
25	2013-01-19 09:00:00	0.130	0.393	0.402	0.389	0.018	0.316	0.227	0.308	0.067	...	0.016	0.399	-1.000000	31.000000	0.0399	CM
26	2013-01-19 10:00:00	0.957	0.494	0.305	0.237	0.017	0.157	1.200	0.461	0.197	...	0.069	1.241	-0.666667	31.333333	0.0399	CM
27	2013-01-19 11:00:00	0.599	0.095	0.363	0.382	0.031	0.443	0.225	0.394	0.258	...	0.165	0.311	-0.333333	31.666667	0.0399	CM
28	2013-01-19 12:00:00	0.343	0.272	0.352	0.347	0.105	1.334	0.199	1.006	0.174	...	0.120	0.210	0.000000	32.000000	0.0399	CM
29	2013-01-19 13:00:00	0.109	0.351	0.357	0.322	0.025	0.202	0.594	0.471	0.119	...	0.055	0.245	0.333333	32.333333	0.0399	CM
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
549	2013-12-19 14:00:00	0.114	0.308	0.334	0.296	0.019	0.417	0.126	0.079	0.033	...	0.056	0.220	5.000000	42.000000	0.0399	L24
550	2013-12-19 15:00:00	0.120	0.172	0.310	0.392	0.030	0.266	0.239	0.098	0.047	...	0.039	0.187	5.000000	42.000000	0.0399	L24
551	2013-12-19 16:00:00	0.162	0.142	0.415	0.475	0.020	0.372	0.301	0.759	0.074	...	0.809	0.801	5.000000	42.000000	0.0399	L24
552	2013-12-19 17:00:00	0.538	0.549	0.430	0.780	0.017	0.459	0.353	0.112	0.040	...	0.191	0.934	5.000000	42.000000	0.0399	L24
553	2013-12-19 18:00:00	0.557	0.551	0.439	0.633	0.031	0.526	0.668	0.080	0.052	...	0.161	0.209	5.000000	42.000000	0.0399	L24
554	2013-12-19 19:00:00	0.667	0.615	0.439	0.526	0.021	0.635	0.943	0.072	0.215	...	0.167	0.183	5.000000	42.000000	0.0399	L24
555	2013-12-19 20:00:00	0.505	0.403	0.510	0.443	0.019	0.524	0.295	0.269	0.480	...	0.154	0.198	5.000000	42.000000	0.0399	L24
556	2013-12-19 21:00:00	0.477	0.357	0.437	0.367	0.031	0.515	0.244	0.134	0.181	...	0.220	0.184	5.000000	42.000000	0.0399	L24
557	2013-12-19 22:00:00	0.418	0.459	0.440	0.316	0.022	0.947	0.237	0.157	0.238	...	0.092	0.547	5.000000	41.333333	0.0399	L24
558	2013-12-19 23:00:00	0.276	0.459	0.413	0.253	0.018	0.574	0.229	0.169	0.307	...	0.131	0.283	5.000000	40.666667	0.0399	L24

	GMT	D0000	D0001	D0005	D0007	D0010	D0012	D0013	D0016	D0018	...	D1023	D1024	TempC	TempF	Price	Eve
559	2013-12-20 00:00:00	0.097	0.224	0.403	0.255	0.026	0.157	0.192	0.124	0.589	...	0.148	0.209	5.000000	40.000000	0.0399	L24
560	2013-12-20 01:00:00	0.093	0.094	0.346	0.172	0.027	0.121	0.099	0.071	0.089	...	0.101	0.365	5.000000	40.000000	0.0399	L24
561	2013-12-20 02:00:00	0.102	0.085	0.273	0.193	0.017	0.115	0.110	0.076	0.071	...	0.107	0.180	5.000000	40.000000	0.0399	L24
562	2013-12-20 03:00:00	0.119	0.084	0.217	0.221	0.020	0.135	0.104	0.082	0.048	...	0.046	0.252	5.000000	40.000000	0.0399	L24
563	2013-12-20 04:00:00	0.369	0.103	0.161	0.142	0.029	0.100	0.108	0.073	0.076	...	0.059	0.222	5.000000	40.000000	0.0399	L24
564	2013-12-26 08:00:00	0.109	0.260	0.175	0.354	0.029	0.165	0.083	0.085	0.119	...	0.026	0.151	4.000000	40.000000	0.0399	L3
565	2013-12-26 09:00:00	0.284	0.317	0.168	0.310	0.020	0.124	0.602	0.126	0.145	...	0.016	0.267	4.000000	40.000000	0.0399	L3
566	2013-12-26 10:00:00	0.147	0.367	0.167	0.329	0.016	0.120	0.328	0.286	0.302	...	0.066	0.151	4.000000	40.000000	0.0399	L3
567	2013-12-29 17:00:00	0.945	0.457	0.425	0.445	0.019	0.137	0.379	0.083	0.494	...	0.056	0.541	3.000000	38.000000	0.0399	L12
568	2013-12-29 18:00:00	0.686	0.357	0.349	0.513	0.030	0.115	1.118	0.084	0.301	...	0.104	0.299	3.000000	38.000000	0.0399	L12
569	2013-12-29 19:00:00	0.682	0.274	0.220	1.195	0.021	0.238	0.770	0.086	0.361	...	0.110	0.470	3.000000	38.000000	0.0399	L12
570	2013-12-29 20:00:00	0.689	0.312	0.272	0.417	0.019	0.210	0.303	0.084	0.287	...	0.194	0.282	3.000000	38.000000	0.0399	L12
571	2013-12-29 21:00:00	0.620	0.328	0.215	0.363	0.017	0.116	0.338	0.242	0.646	...	0.177	0.297	3.000000	38.000000	0.0399	L12
572	2013-12-29 22:00:00	0.514	0.295	0.178	0.347	0.019	0.113	0.315	0.291	0.494	...	0.140	0.270	4.666667	40.666667	0.0399	L12
573	2013-12-29 23:00:00	0.443	0.316	0.199	0.253	0.021	0.113	0.175	0.455	0.877	...	0.149	0.489	6.333333	43.333333	0.0399	L12
574	2013-12-30 00:00:00	0.293	0.361	0.180	0.288	0.030	0.100	0.127	0.180	1.252	...	0.077	0.191	8.000000	46.000000	0.0399	L12
575	2013-12-30 01:00:00	0.294	0.108	0.182	0.228	0.021	0.110	0.162	0.159	0.616	...	0.073	0.212	8.000000	46.000000	0.0399	L12
576	2013-12-30 02:00:00	0.142	0.086	0.359	0.148	0.019	0.101	0.142	0.080	0.075	...	0.031	0.201	8.000000	46.000000	0.0399	L12
577	2013-12-30 03:00:00	0.111	0.065	0.234	0.141	0.019	0.087	0.121	0.081	0.100	...	0.030	0.203	8.000000	46.000000	0.0399	L12
578	2013-12-30 04:00:00	0.542	0.056	0.188	0.139	0.018	0.120	0.103	0.082	0.095	...	0.047	0.151	8.000000	46.000000	0.0399	L12

579 rows × 419 columns

In [63]:
```
# select each user ID and form a seperate dataframe
# first change all the user ID to "Consumption"
new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[0]):
    new_col.append('Consumption')
new_col = new_col + list(df_type0_con.columns)[-8:] # the new column names
df_type0_con.columns = new_col
df_all_con_long = pd.DataFrame()
for i in range(1, houseGroupDf.counts.iloc[0] + 1):
    df_all_con_long = df_all_con_long.append(df_type0_con.iloc[:,[0,i,-8,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[1]): # for renaming the second type
    new_col.append('Consumption')
new_col = new_col + list(df_type1_con.columns)[-8:] # the new column names
df_type1_con.columns = new_col
for i in range(1, houseGroupDf.counts.iloc[1] + 1):
    df_all_con_long = df_all_con_long.append(df_type1_con.iloc[:,[0,i,-8,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[2]): # for renaming the third type
    new_col.append('Consumption')
new_col = new_col + list(df_type2_con.columns)[-8:] # the new column names
df_type2_con.columns = new_col
for i in range(1, houseGroupDf.counts.iloc[2] + 1):
    df_all_con_long = df_all_con_long.append(df_type2_con.iloc[:,[0,i,-8,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

# for clearer visulization, round the temperatures
df_all_con_long['TempC'] = df_all_con_long.TempC.round(0)
df_all_con_long['TempF'] = df_all_con_long.TempF.round(0)
df_all_con_long # ready for the price responsiveness analysis
```

Out[63]:

	GMT	Consumption	TempC	TempF	Price	Event_tags	Day of week	Hour of day	Household type	Predicted consumption
0	2013-01-04 14:00:00	0.102	10.0	51.0	0.0399	L3	4	14	0	0.186761
1	2013-01-04 15:00:00	0.097	10.0	51.0	0.0399	L3	4	15	0	0.192933
2	2013-01-04 16:00:00	1.361	9.0	49.0	0.0399	L3	4	16	0	0.228681
3	2013-01-07 23:00:00	0.314	7.0	44.0	0.6720	H3	0	23	0	0.221413
4	2013-01-08 00:00:00	0.098	7.0	44.0	0.6720	H3	1	0	0	0.164445
5	2013-01-08 01:00:00	0.099	7.0	43.0	0.6720	H3	1	1	0	0.132424
6	2013-01-10 02:00:00	0.097	1.0	34.0	0.0399	L3	3	2	0	0.111933
7	2013-01-10 03:00:00	0.095	1.0	34.0	0.0399	L3	3	3	0	0.110324
8	2013-01-10 04:00:00	0.220	1.0	34.0	0.0399	L3	3	4	0	0.106748
9	2013-01-11 11:00:00	0.105	3.0	37.0	0.6720	H3	4	11	0	0.205935
10	2013-01-11 12:00:00	0.060	4.0	39.0	0.6720	H3	4	12	0	0.204464
11	2013-01-11 13:00:00	0.103	4.0	39.0	0.6720	H3	4	13	0	0.206057
12	2013-01-13 05:00:00	0.237	-2.0	29.0	0.6720	H6	6	5	0	0.105996
13	2013-01-13 06:00:00	0.090	-2.0	29.0	0.6720	H6	6	6	0	0.116613
14	2013-01-13 07:00:00	0.068	-1.0	30.0	0.6720	H6	6	7	0	0.148359
15	2013-01-13 08:00:00	0.100	-1.0	31.0	0.6720	H6	6	8	0	0.212905
16	2013-01-13 09:00:00	0.100	0.0	32.0	0.6720	H6	6	9	0	0.246042
17	2013-01-13 10:00:00	0.072	1.0	33.0	0.6720	H6	6	10	0	0.257176
18	2013-01-16 23:00:00	0.101	-5.0	22.0	0.6720	H3	2	23	0	0.273620
19	2013-01-17 00:00:00	0.088	-6.0	21.0	0.6720	H3	3	0	0	0.205406
20	2013-01-17 01:00:00	0.076	-5.0	22.0	0.6720	H3	3	1	0	0.147150
21	2013-01-19 05:00:00	0.080	-2.0	29.0	0.0399	CM	5	5	0	0.107990
22	2013-01-19 06:00:00	0.095	-2.0	29.0	0.0399	CM	5	6	0	0.119402
23	2013-01-19 07:00:00	0.094	-2.0	30.0	0.0399	CM	5	7	0	0.154056
24	2013-01-19 08:00:00	0.069	-1.0	30.0	0.0399	CM	5	8	0	0.216660
25	2013-01-19 09:00:00	0.130	-1.0	31.0	0.0399	CM	5	9	0	0.245253
26	2013-01-19 10:00:00	0.957	-1.0	31.0	0.0399	CM	5	10	0	0.268189
27	2013-01-19 11:00:00	0.599	-0.0	32.0	0.0399	CM	5	11	0	0.273999
28	2013-01-19 12:00:00	0.343	0.0	32.0	0.0399	CM	5	12	0	0.268615

	GMT	Consumption	TempC	TempF	Price	Event_tags	Day of week	Hour of day	Household type	Predicted consumption
29	2013-01-19 13:00:00	0.109	0.0	32.0	0.0399	CM	5	13	0	0.268550
...	...	...	...	...	...	...	...	...	...	...
258783	2013-12-19 14:00:00	0.239	5.0	42.0	0.0399	L24	3	14	2	0.212958
258784	2013-12-19 15:00:00	1.027	5.0	42.0	0.0399	L24	3	15	2	0.274775
258785	2013-12-19 16:00:00	1.347	5.0	42.0	0.0399	L24	3	16	2	0.309605
258786	2013-12-19 17:00:00	2.114	5.0	42.0	0.0399	L24	3	17	2	0.883250
258787	2013-12-19 18:00:00	0.565	5.0	42.0	0.0399	L24	3	18	2	0.516592
258788	2013-12-19 19:00:00	0.242	5.0	42.0	0.0399	L24	3	19	2	0.380476
258789	2013-12-19 20:00:00	0.276	5.0	42.0	0.0399	L24	3	20	2	0.364367
258790	2013-12-19 21:00:00	0.295	5.0	42.0	0.0399	L24	3	21	2	0.324095
258791	2013-12-19 22:00:00	0.256	5.0	41.0	0.0399	L24	3	22	2	0.299453
258792	2013-12-19 23:00:00	0.271	5.0	41.0	0.0399	L24	3	23	2	0.191650
258793	2013-12-20 00:00:00	0.183	5.0	40.0	0.0399	L24	4	0	2	0.125740
258794	2013-12-20 01:00:00	0.090	5.0	40.0	0.0399	L24	4	1	2	0.101436
258795	2013-12-20 02:00:00	0.115	5.0	40.0	0.0399	L24	4	2	2	0.097334
258796	2013-12-20 03:00:00	0.161	5.0	40.0	0.0399	L24	4	3	2	0.103238
258797	2013-12-20 04:00:00	0.089	5.0	40.0	0.0399	L24	4	4	2	0.101526
258798	2013-12-26 08:00:00	0.230	4.0	40.0	0.0399	L3	3	8	2	0.291602
258799	2013-12-26 09:00:00	0.157	4.0	40.0	0.0399	L3	3	9	2	0.253392
258800	2013-12-26 10:00:00	0.085	4.0	40.0	0.0399	L3	3	10	2	0.235750
258801	2013-12-29 17:00:00	0.343	3.0	38.0	0.0399	L12	6	17	2	0.598094
258802	2013-12-29 18:00:00	0.288	3.0	38.0	0.0399	L12	6	18	2	0.452974
258803	2013-12-29 19:00:00	0.273	3.0	38.0	0.0399	L12	6	19	2	0.342955
258804	2013-12-29 20:00:00	NaN	3.0	38.0	0.0399	L12	6	20	2	0.343809
258805	2013-12-29 21:00:00	0.290	3.0	38.0	0.0399	L12	6	21	2	0.312450
258806	2013-12-29 22:00:00	0.311	5.0	41.0	0.0399	L12	6	22	2	0.275382
258807	2013-12-29 23:00:00	0.208	6.0	43.0	0.0399	L12	6	23	2	0.174786
258808	2013-12-30 00:00:00	0.132	8.0	46.0	0.0399	L12	0	0	2	0.115437
258809	2013-12-30 01:00:00	0.138	8.0	46.0	0.0399	L12	0	1	2	0.101805
258810	2013-12-30 02:00:00	0.088	8.0	46.0	0.0399	L12	0	2	2	0.103848

	GMT	Consumption	TempC	TempF	Price	Event_tags	Day of week	Hour of day	Household type	Predicted consumption
258811	2013-12-30 03:00:00	0.116	8.0	46.0	0.0399	L12	0	3	2	0.101531
258812	2013-12-30 04:00:00	0.153	8.0	46.0	0.0399	L12	0	4	2	0.103520

258813 rows × 10 columns

In [74]:
```
 # variance of type 0, 1, 2
var_type0_high = []
var_type1_high = []
var_type2_high = []
var_type0_low = []
var_type1_low = []
var_type2_low = []
for i in range(24):
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 0) & (df_all_con_long['Price'] == 0.6720)]
    var_type0_high.append(df_hour_con_long['Consumption'].std())
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 0) & (df_all_con_long['Price'] == 0.0399)]
    var_type0_low.append(df_hour_con_long['Consumption'].std())
for i in range(24):
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 1) & (df_all_con_long['Price'] == 0.6720)]
    var_type1_high.append(df_hour_con_long['Consumption'].std())
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 1) & (df_all_con_long['Price'] == 0.0399)]
    var_type1_low.append(df_hour_con_long['Consumption'].std())
for i in range(24):
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 2) & (df_all_con_long['Price'] == 0.6720)]
    var_type2_high.append(df_hour_con_long['Consumption'].std())
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 2) & (df_all_con_long['Price'] == 0.0399)]
    var_type2_low.append(df_hour_con_long['Consumption'].std())
```

Now we get the standard deviation of all type housholds in different pricing over 24 hours.

In [77]:
```
# non-event data
# Type 0 household
# build a new dataframe with columns: GMT, user ids in the group
df_type0_nf = df_tou1h_nf_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[0]].
House)]
df_type0_nf = pd.merge(df_type0_nf, df_wea1h_nf_cold, on = 'GMT') # add weather
df_type0_nf['Price'] = 0.1176
df_type0_nf['Day of week'] = pd.to_datetime(df_type0_nf.GMT).dt.dayofweek # add day of week
df_type0_nf['Hour of day'] = pd.to_datetime(df_type0_nf.GMT).dt.hour # add hour of day
df_type0_nf['Household type'] = 0

# Type 1 household
# build a new dataframe with columns: GMT, user ids in the group
df_type1_nf = df_tou1h_nf_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[1]].
House)]
df_type1_nf = pd.merge(df_type1_nf, df_wea1h_nf_cold, on = 'GMT') # add weather
df_type1_nf['Price'] = 0.1176
df_type1_nf['Day of week'] = pd.to_datetime(df_type1_nf.GMT).dt.dayofweek # add day of week
df_type1_nf['Hour of day'] = pd.to_datetime(df_type1_nf.GMT).dt.hour # add hour of day
df_type1_nf['Household type'] = 1

# Type 2 household
# build a new dataframe with columns: GMT, user ids in the group
df_type2_nf = df_tou1h_nf_cold[['GMT'] + list(houseLableDf[houseLableDf.Label == houseGroupDf.Label.iloc[2]].
House)]
df_type2_nf = pd.merge(df_type2_nf, df_wea1h_nf_cold, on = 'GMT') # add weather
df_type2_nf['Price'] = 0.1176
df_type2_nf['Day of week'] = pd.to_datetime(df_type2_nf.GMT).dt.dayofweek # add day of week
df_type2_nf['Hour of day'] = pd.to_datetime(df_type2_nf.GMT).dt.hour # add hour of day
df_type2_nf['Household type'] = 2
```

In [78]:
```
# add dummy column to use code in the next cell
# type 0 event time consumption prediction
df_type0_nf['Predicted consumption'] = 0
# price responsiveness elementwise
df_type0_con_nf = df_type0_nf.copy() # store the consumption not the response, just for this additional test
# df_type0_con.iloc[:,1:houseGroupDf.counts.iloc[0] + 1] = df_type0.iloc[:,1:houseGroupDf.counts.iloc[0] +
 1].sub(df_type0['Predicted consumption'], axis = 0)

# type 1 event time consumption prediction
df_type1_nf['Predicted consumption'] = 0
# price responsiveness elementwise
df_type1_con_nf = df_type1_nf.copy()
# df_type1_nf_con.iloc[:,1:houseGroupDf.counts.iloc[1] + 1] = df_type1_nf.iloc[:,1:houseGroupDf.counts.iloc
[1] + 1].sub(df_type1_nf['Predicted consumption'], axis = 0)

# type 2 event time consumption prediction
df_type2_nf['Predicted consumption'] = 0
# price responsiveness elementwise
df_type2_con_nf = df_type2_nf.copy()
# df_type2_con.iloc[:,1:houseGroupDf.counts.iloc[2] + 1] = df_type2.iloc[:,1:houseGroupDf.counts.iloc[2] +
 1].sub(df_type2['Predicted consumption'], axis = 0)
```

In [79]:
```
# select each user ID and form a seperate dataframe
# first change all the user ID to "Consumption"
new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[0]):
    new_col.append('Consumption')
new_col = new_col + list(df_type0_con_nf.columns)[-7:] # the new column names
df_type0_con_nf.columns = new_col
df_all_con_long = pd.DataFrame()
for i in range(1, houseGroupDf.counts.iloc[0] + 1):
    df_all_con_long = df_all_con_long.append(df_type0_con_nf.iloc[:,[0,i,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[1]): # for renaming the second type
    new_col.append('Consumption')
new_col = new_col + list(df_type1_con_nf.columns)[-7:] # the new column names
df_type1_con_nf.columns = new_col
for i in range(1, houseGroupDf.counts.iloc[1] + 1):
    df_all_con_long = df_all_con_long.append(df_type1_con_nf.iloc[:,[0,i,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

new_col = ['GMT']
for i in range(0, houseGroupDf.counts.iloc[2]): # for renaming the third type
    new_col.append('Consumption')
new_col = new_col + list(df_type2_con_nf.columns)[-7:] # the new column names
df_type2_con_nf.columns = new_col
for i in range(1, houseGroupDf.counts.iloc[2] + 1):
    df_all_con_long = df_all_con_long.append(df_type2_con_nf.iloc[:,[0,i,-7,-6,-5,-4,-3,-2,-1]], ignore_index
= True)

# for clearer visulization, round the temperatures
df_all_con_long['TempC'] = df_all_con_long.TempC.round(0)
df_all_con_long['TempF'] = df_all_con_long.TempF.round(0)
df_all_con_long # ready for the price responsiveness analysis
```

Out[79]:

	GMT	Consumption	TempC	TempF	Price	Day of week	Hour of day	Household type	Predicted consumption
0	2013-01-01 00:00:00	1.447	7.0	45.0	0.1176	1	0	0	0
1	2013-01-01 01:00:00	0.336	7.0	44.0	0.1176	1	1	0	0
2	2013-01-01 02:00:00	0.244	6.0	44.0	0.1176	1	2	0	0
3	2013-01-01 03:00:00	0.213	6.0	43.0	0.1176	1	3	0	0
4	2013-01-01 04:00:00	0.210	5.0	41.0	0.1176	1	4	0	0
5	2013-01-01 05:00:00	0.198	5.0	40.0	0.1176	1	5	0	0
6	2013-01-01 06:00:00	0.202	4.0	38.0	0.1176	1	6	0	0
7	2013-01-01 07:00:00	0.167	4.0	38.0	0.1176	1	7	0	0
8	2013-01-01 08:00:00	0.360	3.0	38.0	0.1176	1	8	0	0
9	2013-01-01 09:00:00	0.178	3.0	38.0	0.1176	1	9	0	0
10	2013-01-01 10:00:00	0.202	4.0	40.0	0.1176	1	10	0	0
11	2013-01-01 11:00:00	0.188	5.0	41.0	0.1176	1	11	0	0
12	2013-01-01 12:00:00	0.154	6.0	43.0	0.1176	1	12	0	0
13	2013-01-01 13:00:00	0.429	6.0	43.0	0.1176	1	13	0	0
14	2013-01-01 14:00:00	0.494	6.0	43.0	0.1176	1	14	0	0
15	2013-01-01 15:00:00	0.190	6.0	43.0	0.1176	1	15	0	0
16	2013-01-01 16:00:00	0.892	5.0	42.0	0.1176	1	16	0	0
17	2013-01-01 17:00:00	0.411	5.0	40.0	0.1176	1	17	0	0
18	2013-01-01 18:00:00	0.503	4.0	39.0	0.1176	1	18	0	0
19	2013-01-01 19:00:00	0.400	4.0	38.0	0.1176	1	19	0	0
20	2013-01-01 20:00:00	0.723	3.0	38.0	0.1176	1	20	0	0
21	2013-01-01 21:00:00	1.056	3.0	37.0	0.1176	1	21	0	0
22	2013-01-01 22:00:00	0.624	3.0	37.0	0.1176	1	22	0	0
23	2013-01-01 23:00:00	0.323	2.0	36.0	0.1176	1	23	0	0
24	2013-01-02 00:00:00	0.358	2.0	36.0	0.1176	2	0	0	0
25	2013-01-02 01:00:00	0.181	2.0	36.0	0.1176	2	1	0	0
26	2013-01-02 02:00:00	0.092	2.0	36.0	0.1176	2	2	0	0
27	2013-01-02 03:00:00	0.097	2.0	36.0	0.1176	2	3	0	0
28	2013-01-02 04:00:00	0.324	2.0	36.0	0.1176	2	4	0	0

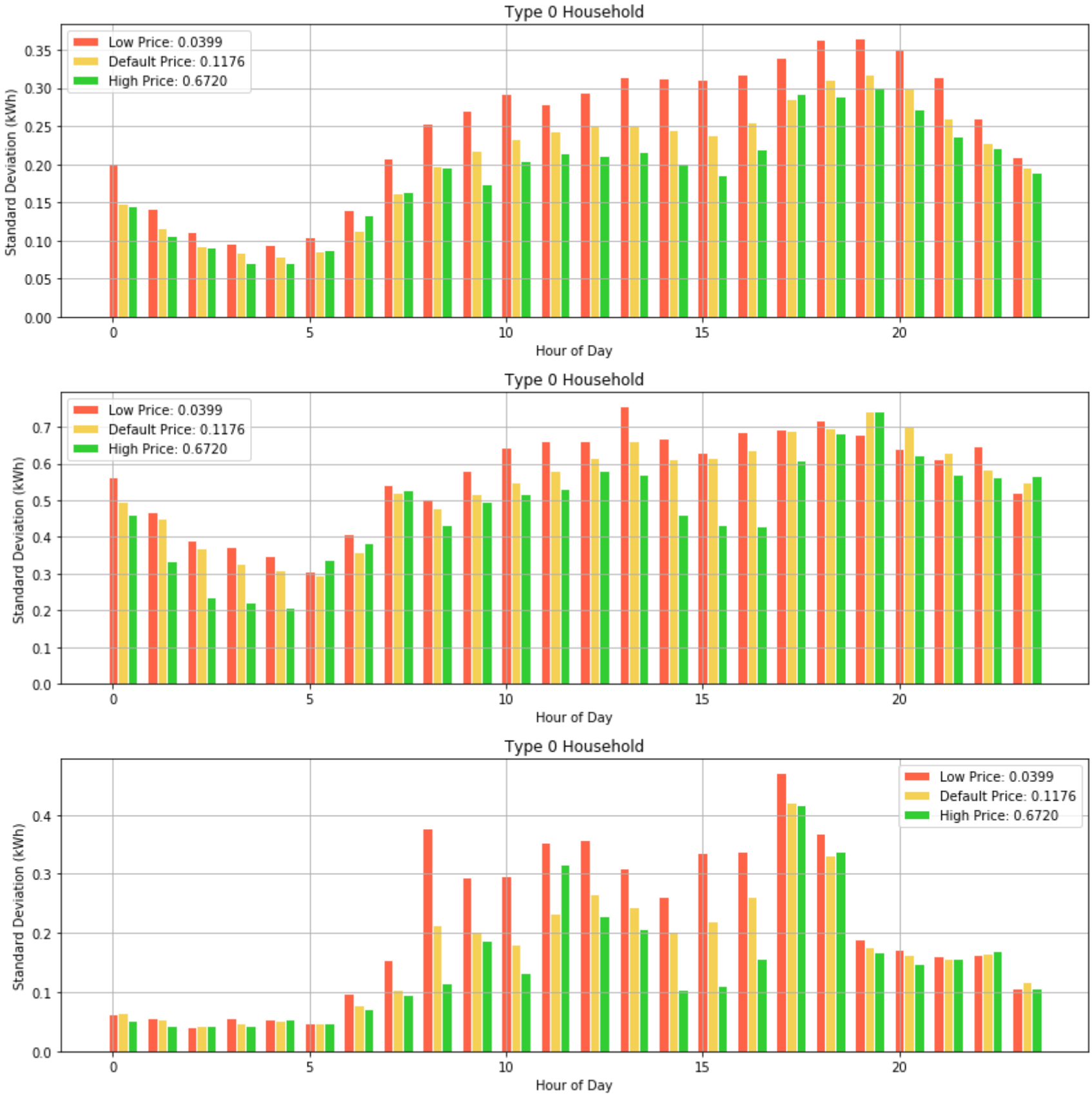
	GMT	Consumption	TempC	TempF	Price	Day of week	Hour of day	Household type	Predicted consumption
29	2013-01-02 05:00:00	0.095	2.0	35.0	0.1176	2	5	0	0
...	...	...	...	...	...	...	...	...	...
890394	2013-12-28 18:00:00	0.312	6.0	42.0	0.1176	5	18	2	0
890395	2013-12-28 19:00:00	0.231	6.0	42.0	0.1176	5	19	2	0
890396	2013-12-28 20:00:00	0.262	6.0	42.0	0.1176	5	20	2	0
890397	2013-12-28 21:00:00	0.304	6.0	42.0	0.1176	5	21	2	0
890398	2013-12-28 22:00:00	0.258	5.0	41.0	0.1176	5	22	2	0
890399	2013-12-28 23:00:00	0.243	4.0	39.0	0.1176	5	23	2	0
890400	2013-12-31 00:00:00	0.152	9.0	48.0	0.1176	1	0	2	0
890401	2013-12-31 01:00:00	0.123	9.0	48.0	0.1176	1	1	2	0
890402	2013-12-31 02:00:00	0.176	9.0	48.0	0.1176	1	2	2	0
890403	2013-12-31 03:00:00	0.100	9.0	48.0	0.1176	1	3	2	0
890404	2013-12-31 04:00:00	0.101	9.0	48.0	0.1176	1	4	2	0
890405	2013-12-31 05:00:00	0.184	9.0	48.0	0.1176	1	5	2	0
890406	2013-12-31 06:00:00	0.103	9.0	48.0	0.1176	1	6	2	0
890407	2013-12-31 07:00:00	0.136	9.0	48.0	0.1176	1	7	2	0
890408	2013-12-31 08:00:00	0.346	9.0	48.0	0.1176	1	8	2	0
890409	2013-12-31 09:00:00	0.385	9.0	48.0	0.1176	1	9	2	0
890410	2013-12-31 10:00:00	0.229	9.0	48.0	0.1176	1	10	2	0
890411	2013-12-31 11:00:00	0.277	9.0	48.0	0.1176	1	11	2	0
890412	2013-12-31 12:00:00	0.217	9.0	48.0	0.1176	1	12	2	0
890413	2013-12-31 13:00:00	0.175	9.0	48.0	0.1176	1	13	2	0
890414	2013-12-31 14:00:00	0.269	9.0	48.0	0.1176	1	14	2	0
890415	2013-12-31 15:00:00	0.266	9.0	48.0	0.1176	1	15	2	0
890416	2013-12-31 16:00:00	0.230	9.0	48.0	0.1176	1	16	2	0
890417	2013-12-31 17:00:00	1.330	9.0	48.0	0.1176	1	17	2	0
890418	2013-12-31 18:00:00	0.336	9.0	48.0	0.1176	1	18	2	0
890419	2013-12-31 19:00:00	0.234	9.0	48.0	0.1176	1	19	2	0
890420	2013-12-31 20:00:00	0.263	9.0	48.0	0.1176	1	20	2	0
890421	2013-12-31 21:00:00	0.311	9.0	48.0	0.1176	1	21	2	0

	GMT	Consumption	TempC	TempF	Price	Day of week	Hour of day	Household type	Predicted consumption
890422	2013-12-31 22:00:00	0.268	9.0	48.0	0.1176	1	22	2	0
890423	2013-12-31 23:00:00	0.290	9.0	48.0	0.1176	1	23	2	0

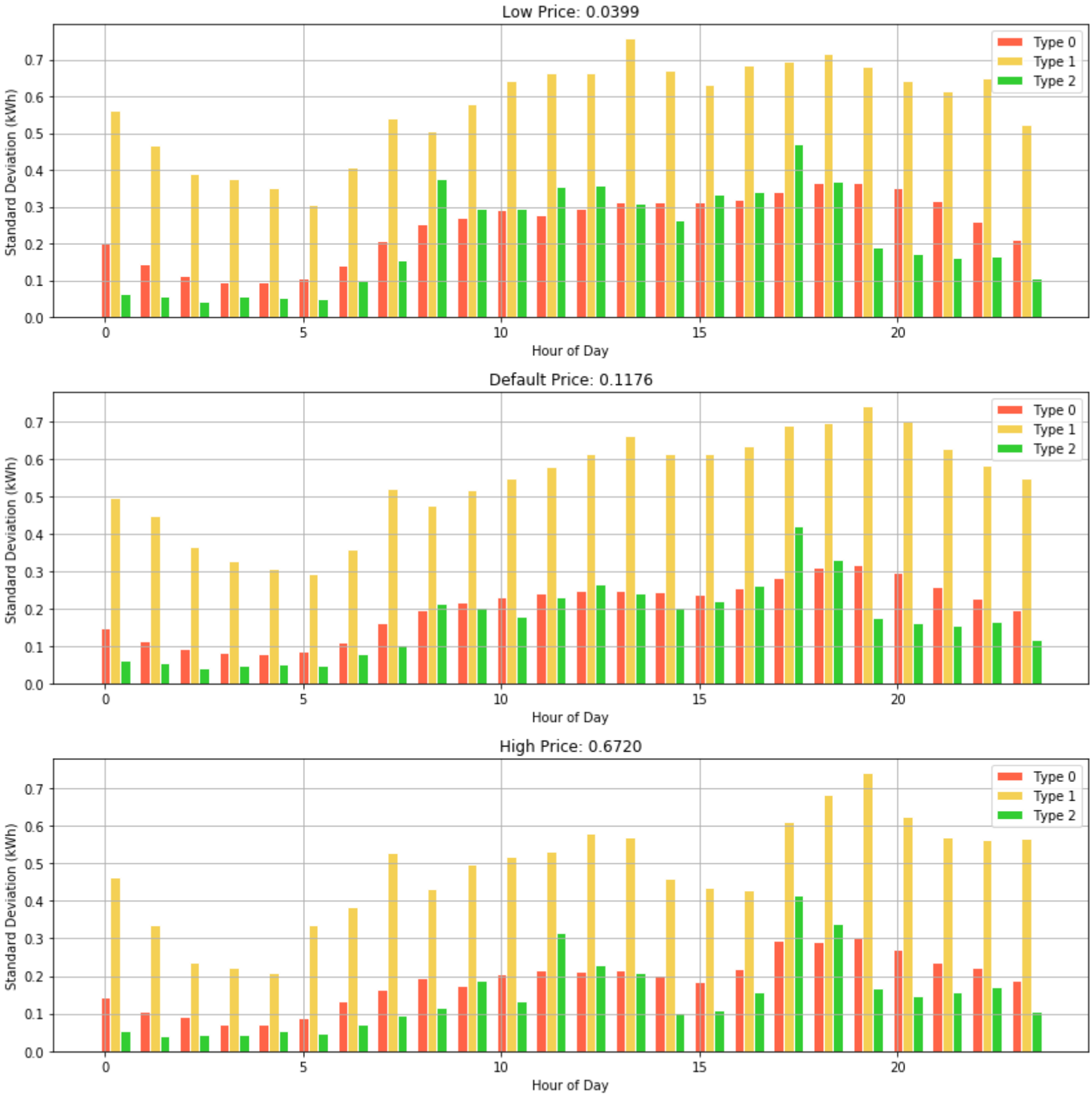
890424 rows × 9 columns

In [80]:
```
 # variance of type 0, 1, 2
var_type0_default = []
var_type1_default = []
var_type2_default = []
for i in range(24):
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 0) & (df_all_con_long['Price'] == 0.1176)]
    var_type0_default.append(df_hour_con_long['Consumption'].std())
for i in range(24):
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 1) & (df_all_con_long['Price'] == 0.1176)]
    var_type1_default.append(df_hour_con_long['Consumption'].std())
for i in range(24):
    df_hour_con_long = df_all_con_long[(df_all_con_long['Hour of day'] == i) & (df_all_con_long['Household ty
pe'] == 2) & (df_all_con_long['Price'] == 0.1176)]
    var_type2_default.append(df_hour_con_long['Consumption'].std())
```

In [102]:
```
# plot the grouped bar charts based on price, for 3 household types
fig_all = plt.figure(figsize = (12,12))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.1176, 0.6720]
# set width of bar
barWidth = 0.25
# Set position of bar on X axis
r1 = np.arange(len(var_type0_high))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]
for g in range(3): # three types
    ax_Ntou.append(fig_all.add_subplot(3, 1, (g + 1)))
    # Make the plot
    if g == 0:
        ax_Ntou[-1].bar(r1, var_type0_low, color='tomato', width=barWidth, edgecolor='white', label='Low Pric
e: 0.0399')
        ax_Ntou[-1].bar(r2, var_type0_default, color='xkcd:maize', width=barWidth, edgecolor='white', label=
'Default Price: 0.1176')
        ax_Ntou[-1].bar(r3, var_type0_high, color='limegreen', width=barWidth, edgecolor='white', label='High
Price: 0.6720')
    if g == 1:
        ax_Ntou[-1].bar(r1, var_type1_low, color='tomato', width=barWidth, edgecolor='white', label='Low Pric
e: 0.0399')
        ax_Ntou[-1].bar(r2, var_type1_default, color='xkcd:maize', width=barWidth, edgecolor='white', label=
'Default Price: 0.1176')
        ax_Ntou[-1].bar(r3, var_type1_high, color='limegreen', width=barWidth, edgecolor='white', label='High
Price: 0.6720')
    if g == 2:
        ax_Ntou[-1].bar(r1, var_type2_low, color='tomato', width=barWidth, edgecolor='white', label='Low Pric
e: 0.0399')
        ax_Ntou[-1].bar(r2, var_type2_default, color='xkcd:maize', width=barWidth, edgecolor='white', label=
'Default Price: 0.1176')
        ax_Ntou[-1].bar(r3, var_type2_high, color='limegreen', width=barWidth, edgecolor='white', label='High
Price: 0.6720')
    ax_Ntou[-1].grid()
    ax_Ntou[-1].set_title('Responsiveness'  + ' (Type ' + str(g) + ')' )
    ax_Ntou[-1].set_xlabel('Hour of Day')
    ax_Ntou[-1].set_ylabel('Standard Deviation (kWh)')
    ax_Ntou[-1].legend()
    ax_Ntou[-1].set_title('Type 0 Household')
plt.tight_layout()
```


Type 0 Household
Low Price: 0.0399
Default Price: 0.1176
High Price: 0.6720
Standard Deviation (kWh)
Hour of Day
Type 0 Household
Low Price: 0.0399
Default Price: 0.1176
High Price: 0.6720
Standard Deviation (kWh)
Hour of Day
Type 0 Household
Low Price: 0.0399
Default Price: 0.1176
High Price: 0.6720
Standard Deviation (kWh)
Hour of Day

In [105]:
```
# plot the grouped bar charts based on price, for 3 household types
fig_all = plt.figure(figsize = (12,12))
ax_Ntou = [] # store subplot objects
prices = [0.0399, 0.1176, 0.6720]
# set width of bar
barWidth = 0.25
# Set position of bar on X axis
r1 = np.arange(len(var_type0_high))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]
for g in range(3): # three types
    ax_Ntou.append(fig_all.add_subplot(3, 1, (g + 1)))
    # Make the plot
    if g == 0:
        ax_Ntou[-1].bar(r1, var_type0_low, color='tomato', width=barWidth, edgecolor='white', label='Type 0')
        ax_Ntou[-1].bar(r2, var_type1_low, color='xkcd:maize', width=barWidth, edgecolor='white', label='Type
1')
        ax_Ntou[-1].bar(r3, var_type2_low, color='limegreen', width=barWidth, edgecolor='white', label='Type
 2')
        ax_Ntou[-1].set_title('Low Price: 0.0399')
    if g == 1:
        ax_Ntou[-1].bar(r1, var_type0_default, color='tomato', width=barWidth, edgecolor='white', label='Type
0')
        ax_Ntou[-1].bar(r2, var_type1_default, color='xkcd:maize', width=barWidth, edgecolor='white', label=
'Type 1')
        ax_Ntou[-1].bar(r3, var_type2_default, color='limegreen', width=barWidth, edgecolor='white', label='T
ype 2')
        ax_Ntou[-1].set_title('Default Price: 0.1176')
    if g == 2:
        ax_Ntou[-1].bar(r1, var_type0_high, color='tomato', width=barWidth, edgecolor='white', label='Type 0'
)
        ax_Ntou[-1].bar(r2, var_type1_high, color='xkcd:maize', width=barWidth, edgecolor='white', label='Typ
e 1')
        ax_Ntou[-1].bar(r3, var_type2_high, color='limegreen', width=barWidth, edgecolor='white', label='Type
2')
        ax_Ntou[-1].set_title('High Price: 0.6720')
    ax_Ntou[-1].grid()
    ax_Ntou[-1].set_xlabel('Hour of Day')
    ax_Ntou[-1].set_ylabel('Standard Deviation (kWh)')
    ax_Ntou[-1].legend()
plt.tight_layout()
```


Low Price: 0.0399
Standard Deviation (kWh)
Hour of Day
Type 0
Type 1
Type 2
Default Price: 0.1176
Standard Deviation (kWh)
Hour of Day
Type 0
Type 1
Type 2
High Price: 0.6720
Standard Deviation (kWh)
Hour of Day
Type 0
Type 1
Type 2