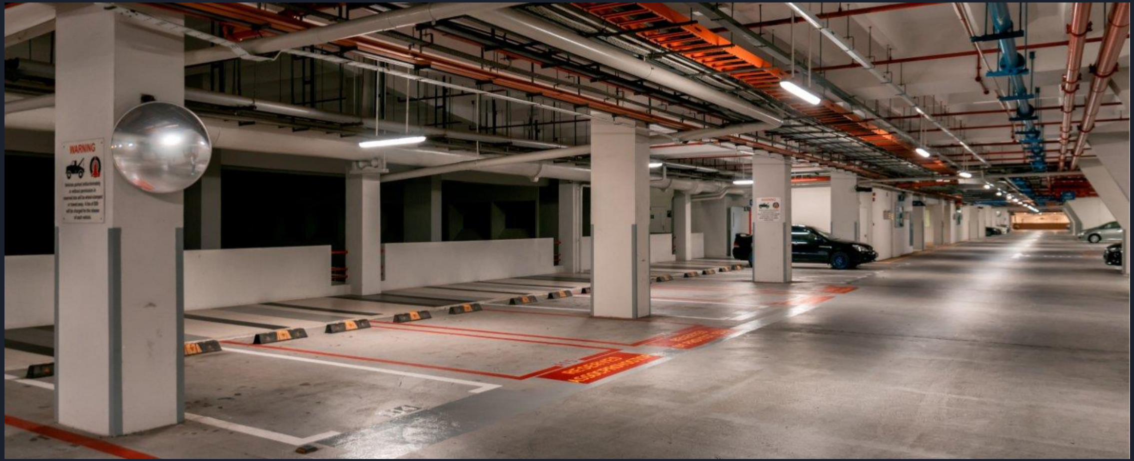
Predicting Carpark Availability

Garrick Goh
Ivan Lai
Ee Chern

Credit: sgCarMart.com

Motivation

- Going to a unfamiliar location? Afraid of lack of parking at peak times?
- Find out even before heading there!



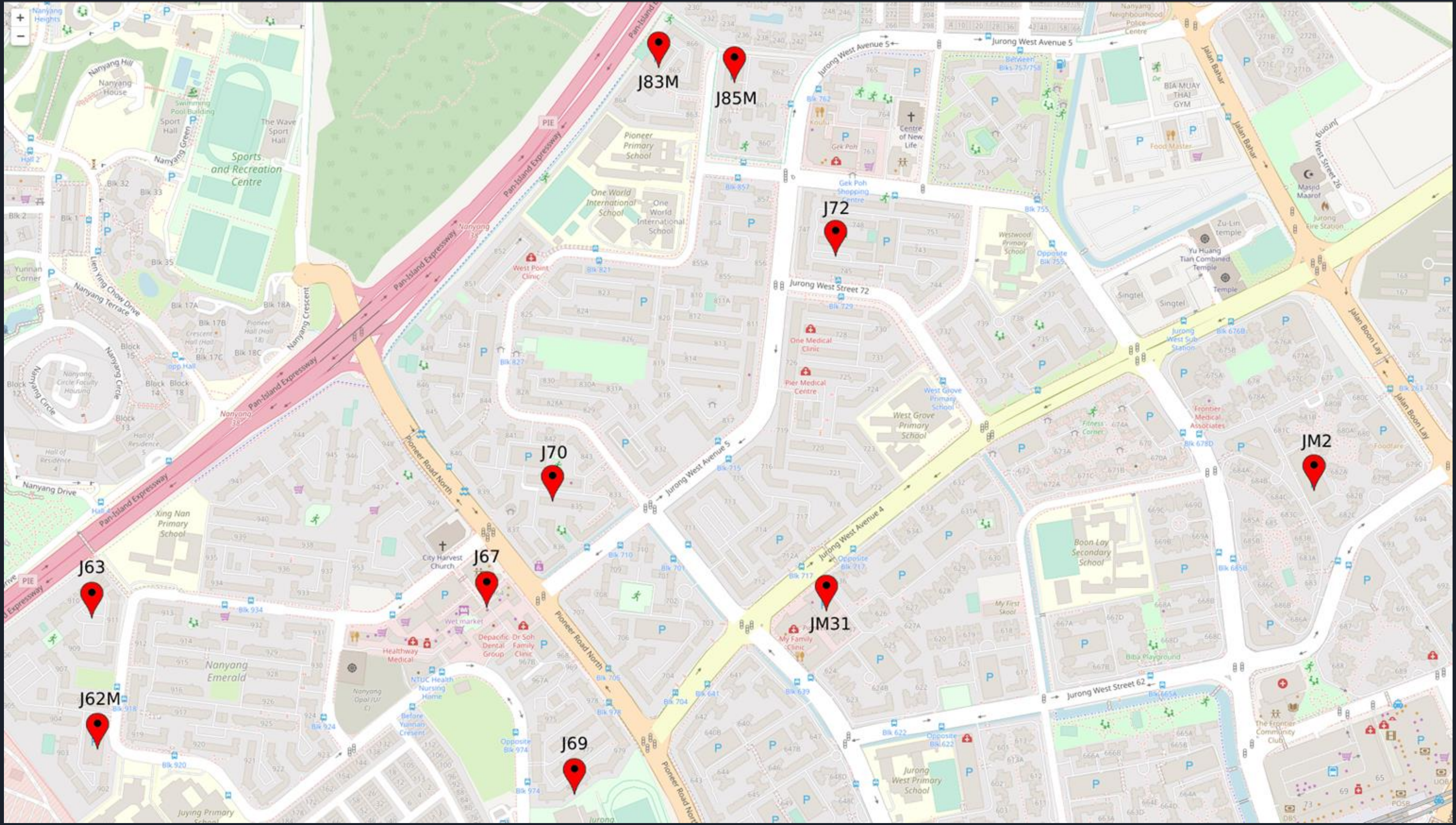
Credit: sgCarMart.com

Data Extraction

We were able to obtain car park availability data from the Government provided dataset (data.gov.sg)
Utilising a python script we scraped 4 years worth of carpark lots from 10 different carparks from
https://data.gov.sg/dataset/carpark-availability

carpark_number	update_datetime	timestamp	total_lots	lots_available
J70	2018-01-01T00:28:19	2018-01-01T00:29:28+08:00	151	0
J85M	2018-01-01T00:28:12	2018-01-01T00:29:28+08:00	217	187
J72	2018-01-01T00:28:14	2018-01-01T00:29:28+08:00	255	0
J69	2018-01-01T00:28:17	2018-01-01T00:29:28+08:00	222	70
JM31	2018-01-01T00:28:11	2018-01-01T00:29:28+08:00	180	112
J62M	2018-01-01T00:27:44	2018-01-01T00:29:28+08:00	222	168
J63	2018-01-01T00:27:44	2018-01-01T00:29:28+08:00	153	21
J83M	2018-01-01T00:28:17	2018-01-01T00:29:28+08:00	163	123
J67	2018-01-01T00:28:01	2018-01-01T00:29:28+08:00	24	0
JM2	2018-01-01T00:28:15	2018-01-01T00:29:28+08:00	300	0
J70	2018-01-01T01:28:19	2018-01-01T01:29:27+08:00	151	0
J85M	2018-01-01T01:28:13	2018-01-01T01:29:27+08:00	217	176
J72	2018-01-01T01:28:16	2018-01-01T01:29:27+08:00	255	0
J69	2018-01-01T01:28:19	2018-01-01T01:29:27+08:00	222	57
JM31	2018-01-01T01:28:11	2018-01-01T01:29:27+08:00	180	118
J62M	2018-01-01T01:27:45	2018-01-01T01:29:27+08:00	222	165
J63	2018-01-01T01:27:45	2018-01-01T01:29:27+08:00	153	6

Data Extraction



Data Cleaning

carpark_1_2018.csv	carpark_4_2018.csv	carpark_7_2018.csv	carpark_10_2018.csv
carpark_1_2019.csv	carpark_4_2019.csv	carpark_7_2019.csv	carpark_10_2019.csv
carpark_1_2020.csv	carpark_4_2020.csv	carpark_7_2020.csv	carpark_10_2020.csv
carpark_1_2021.csv	carpark_4_2021.csv	carpark_7_2021.csv	carpark_10_2021.csv
carpark_2_2018.csv	carpark_5_2018.csv	carpark_8_2018.csv	carpark_11_2018.csv
carpark_2_2019.csv	carpark_5_2019.csv	carpark_8_2019.csv	carpark_11_2019.csv
carpark_2_2020.csv	carpark_5_2020.csv	carpark_8_2020.csv	carpark_11_2020.csv
carpark_2_2021.csv	carpark_5_2021.csv	carpark_8_2021.csv	carpark_11_2021.csv
carpark_3_2018.csv	carpark_6_2018.csv	carpark_9_2018.csv	carpark_12_2018.csv
carpark_3_2019.csv	carpark_6_2019.csv	carpark_9_2019.csv	carpark_12_2019.csv
carpark_3_2020.csv	carpark_6_2020.csv	carpark_9_2020.csv	carpark_12_2020.csv
carpark_3_2021.csv	carpark_6_2021.csv	carpark_9_2021.csv	carpark_12_2021.csv

Data Cleaning

rjchow fix: updated 2022 public holidays (#11) ... 59cc1bb on Oct 25, 2021 24 commits

api	fix: updated 2022 public holidays (#11)	6 months ago
csv	fix: updated 2022 public holidays (#11)	6 months ago
.gitignore	organised stuff into an api	3 years ago
LICENSE	Initial commit	5 years ago
README.md	updated readme with api url	3 years ago
package-lock.json	chore: update csvtojson (#10)	6 months ago
package.json	chore: update csvtojson (#10)	6 months ago

README.md

singapore_public_holidays

Singapore Public Holidays

Dates are sourced from http://www.mom.gov.sg/employment-practices/public-holidays

You can access this via HTTP GET at https://rjchow.github.io/singapore_public_holidays/api/**<year>**/data.json

e.g for 2016: https://rjchow.github.io/singapore_public_holidays/api/2016/data.json

Please submit a pull request if you would like to contribute previous years!

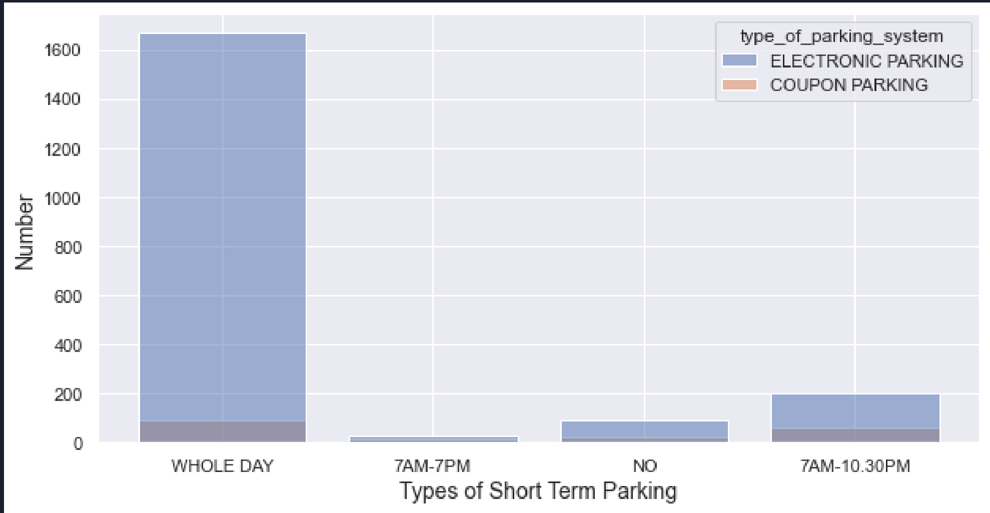
Data Cleaning

	carpark_number	update_datetime	total_lots	lots_available
0	J70	2018-01-01 00:28:19	151	0
1	J85M	2018-01-01 00:28:12	217	187
2	J72	2018-01-01 00:28:14	255	0
3	J69	2018-01-01 00:28:17	222	70
4	JM31	2018-01-01 00:28:11	180	112

	carpark_number	update_datetime	total_lots	lots_available	hour_delta	day	hour	carpark_index	holiday
0	J70	2018-01-01 00:00:00	151	0	0.0	0	0	0	1
1	J70	2018-01-01 01:00:00	151	0	1.0	0	1	0	0
2	J70	2018-01-01 02:00:00	151	0	2.0	0	2	0	0
3	J70	2018-01-01 03:00:00	151	0	3.0	0	3	0	0
4	J70	2018-01-01 04:00:00	151	0	4.0	0	4	0	0

EDA -Exploratory Data Analysis

- In Singapore there are electronic and coupon parking.
- The car parks we chose are all electronic parking, with half of them being surface car-parks and the other half being multi-storey car parks.



Average Carpark Lots

- Over the years of 2018-2021
- Shows the average total lots, lots available and used space.

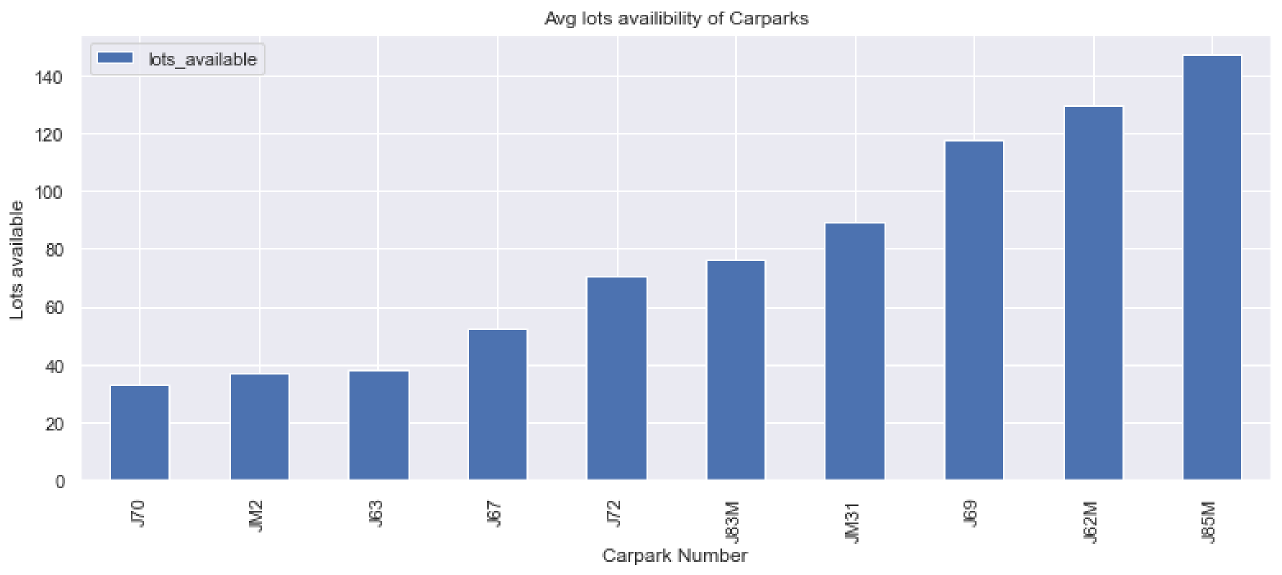
carpark_number	total_lots
J83M	139.304242
J70	148.648423
J63	151.878465
J67	154.565135
JM31	185.006758
J62M	211.485545
J69	222.424479
J72	237.445274
J85M	247.339099
JM2	300.000000

carpark_number	lots_available
J70	33.365602
JM2	36.831565
J63	37.999359
J67	52.153077
J72	70.633910
J83M	76.343251
JM31	89.236564
J69	117.675593
J62M	129.794451
J85M	147.042587

carpark_number	used_space
J62M	38.133694
J85M	38.381228
J69	47.184195
J83M	49.974366
JM31	52.142414
J67	66.534002
J72	71.010221
J63	75.094435
J70	77.612034
JM2	87.722809

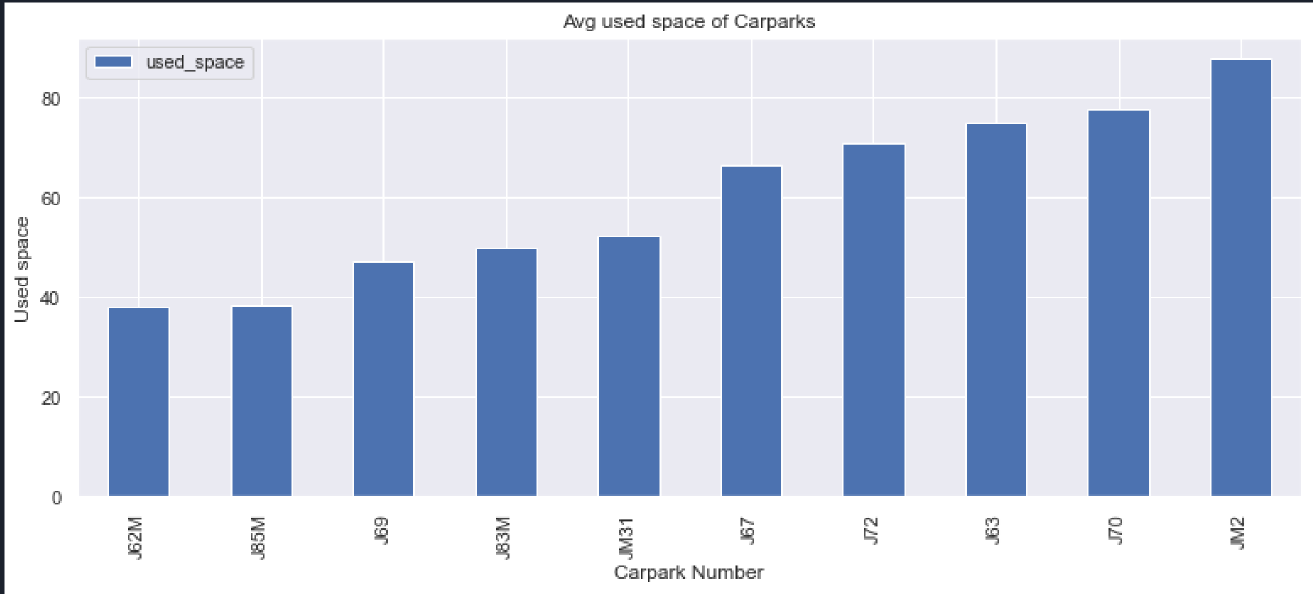
Average lots availability of Car Parks

- Graph shows the average lots available for the 10 car parks.
- J85M most lots available



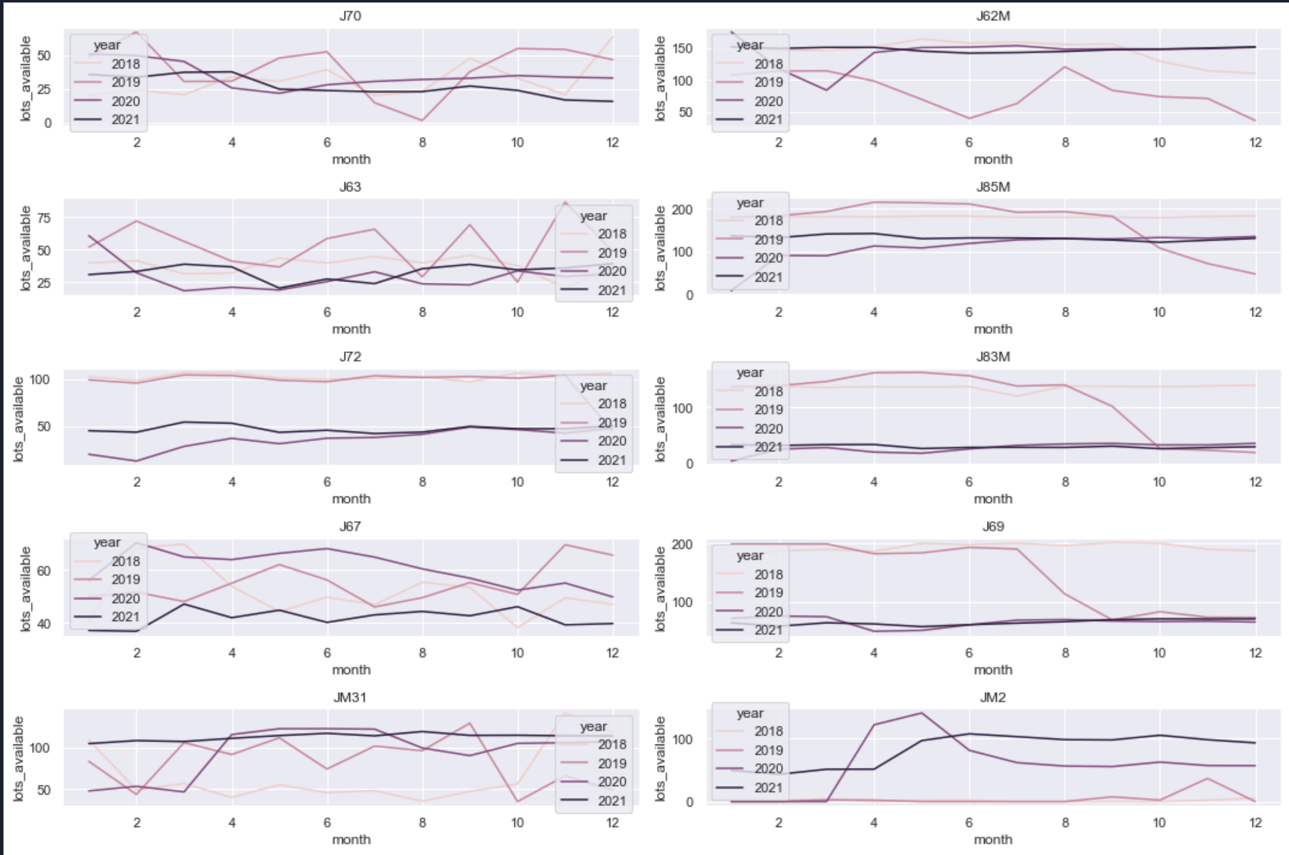
Average Used Space of Car Parks

- Graph shows the average used space for the 10 car parks.
- JM2 used the most



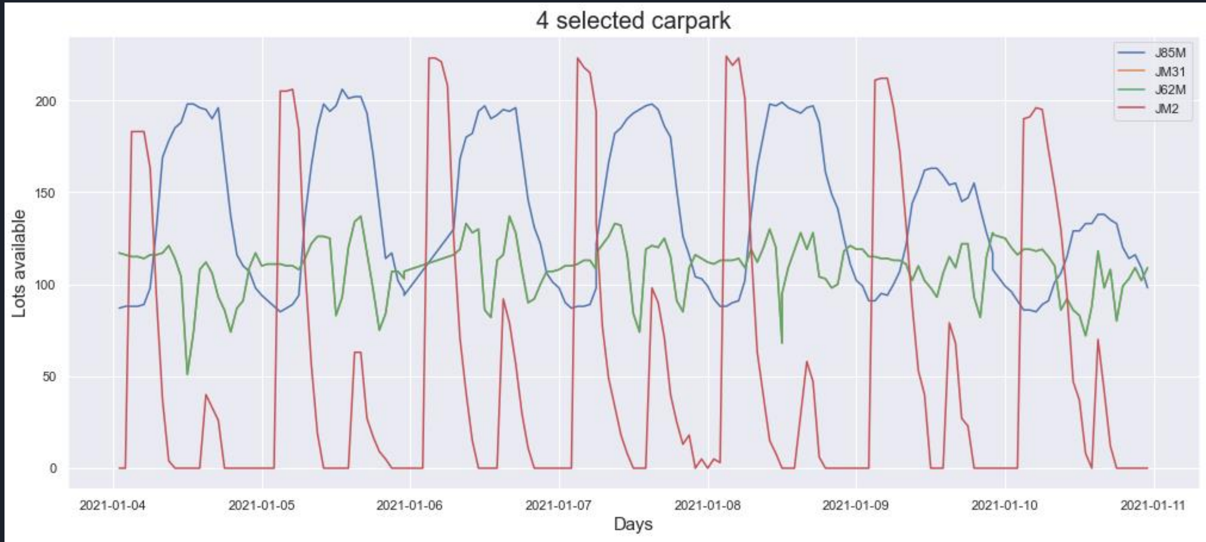
Car Parks over the years

- Graph representation over 2018-2021
- Fluctuations for 2018 and 2019.
- 2021 seems the most stable



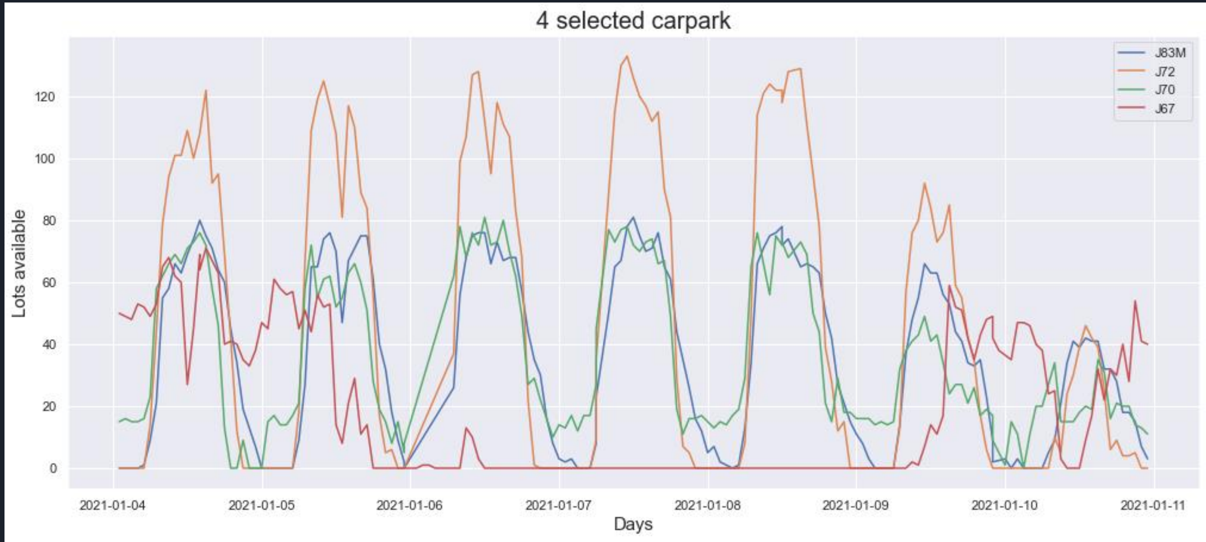
Analysing First Week of January 2021

- Graph shows 4 selected car parks
- J85M, JM31, J62M, JM2



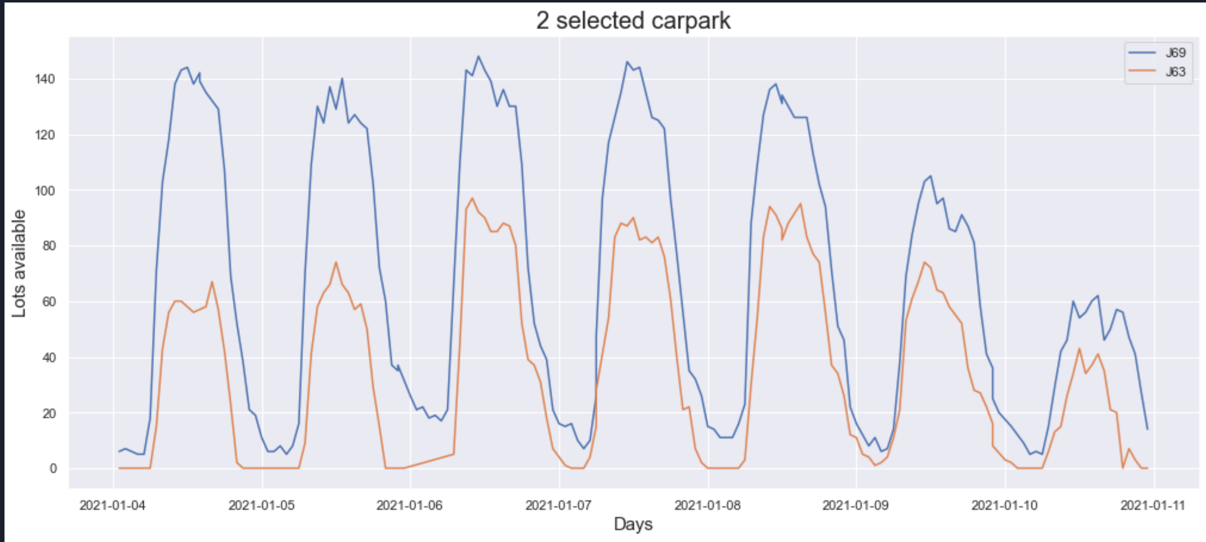
Analysing First Week of January 2021

- Graph shows 4 selected car parks
- J83M, J72, J70, J67



Analysing First Week of January 2021

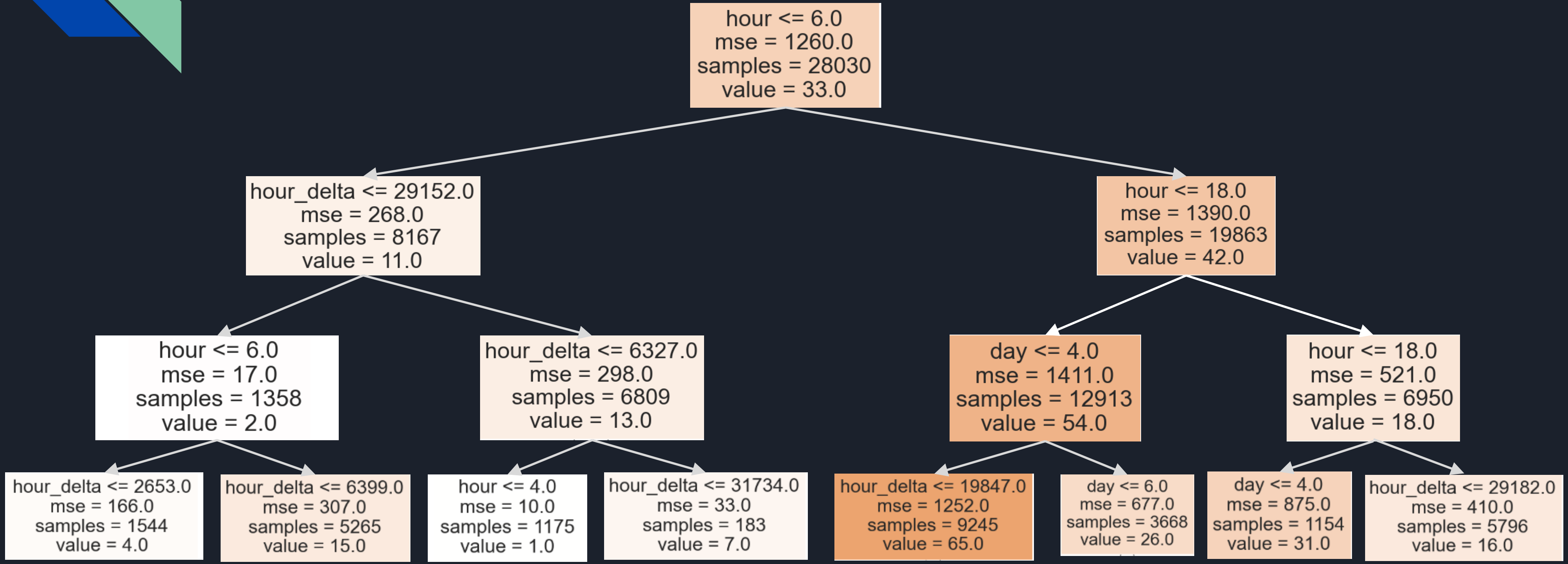
- Graph shows 2 selected car parks
- J69, J63



ML Model 1 - Decision Tree Regressor

Using depth of 4 to train Decision Tree Regressor using Predictors[Day, Hour, Hour_Delta]

ML Model 1 - Decision Tree Regressor

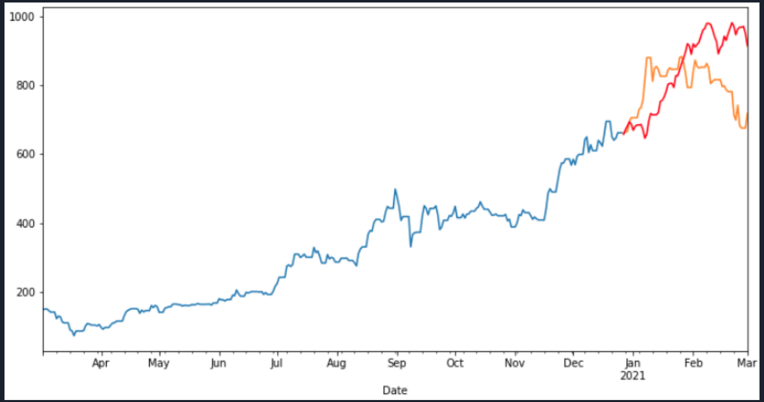


ML Model 2
Autoregressive Integrated Moving Average(ARIMA)

Autoregressive Integrated Moving Average
Combines 2 Models: Autoregressive & Moving Average to forecast results

Utilises 3 variables in the model + seasonality
p: Forecasting values based on past data - aka Lags
d: Differencing Value, when data is non-stationary
q: Forecasting based on past errors - aka error Lags

Seasonality: Frequency of Data
Eg. Hourly, Daily, Monthly, Yearly
24 7 12 1



Credit: Taha Binhuraib

ML Model 2
Autoregressive Integrated Moving Average(ARIMA)

Utilising pmdarima’s auto_arima to find the best fit model for our dataset

Using forward chaining to test and train our dataset further

Akaike's Information Criterion (AIC):
Usually used to determine predictors for regression, used in ARIMA to determine order p, q and d for ARIMA model

AIC lower = Better

```
Performing stepwise search to minimize aic
 ARIMA(1,0,1)(0,0,1)[24] intercept   : AIC=inf, Time=10.31 sec
 ARIMA(0,0,0)(0,0,0)[24] intercept   : AIC=73184.998, Time=0.09 sec
 ARIMA(1,0,0)(1,0,0)[24] intercept   : AIC=inf, Time=13.65 sec
 ARIMA(0,0,1)(0,0,1)[24] intercept   : AIC=inf, Time=7.49 sec
 ARIMA(0,0,0)(0,0,0)[24]             : AIC=93176.299, Time=0.05 sec
 ARIMA(0,0,0)(1,0,0)[24] intercept   : AIC=inf, Time=7.94 sec
 ARIMA(0,0,0)(0,0,1)[24] intercept   : AIC=inf, Time=3.12 sec
 ARIMA(0,0,0)(1,0,1)[24] intercept   : AIC=inf, Time=9.52 sec
 ARIMA(1,0,0)(0,0,0)[24] intercept   : AIC=63890.157, Time=0.25 sec
 ARIMA(1,0,0)(0,0,1)[24] intercept   : AIC=inf, Time=7.50 sec
 ARIMA(1,0,0)(1,0,1)[24] intercept   : AIC=inf, Time=20.59 sec
 ARIMA(2,0,0)(0,0,0)[24] intercept   : AIC=63694.275, Time=0.32 sec
 ARIMA(2,0,0)(1,0,0)[24] intercept   : AIC=inf, Time=23.92 sec
 ARIMA(2,0,0)(0,0,1)[24] intercept   : AIC=inf, Time=5.64 sec
 ARIMA(2,0,0)(1,0,1)[24] intercept   : AIC=inf, Time=24.47 sec
 ARIMA(3,0,0)(0,0,0)[24] intercept   : AIC=63520.464, Time=0.46 sec
 ARIMA(3,0,0)(1,0,0)[24] intercept   : AIC=inf, Time=40.09 sec
 ARIMA(3,0,0)(0,0,1)[24] intercept   : AIC=inf, Time=11.40 sec
 ARIMA(3,0,0)(1,0,1)[24] intercept   : AIC=48283.044, Time=32.81 sec
 ARIMA(3,0,0)(2,0,1)[24] intercept   : AIC=63488.359, Time=37.77 sec
 ARIMA(3,0,0)(1,0,2)[24] intercept   : AIC=52127.644, Time=101.04 sec
 ARIMA(3,0,0)(0,0,2)[24] intercept   : AIC=inf, Time=54.24 sec
 ARIMA(3,0,0)(2,0,0)[24] intercept   : AIC=inf, Time=330.26 sec
 ARIMA(3,0,0)(2,0,2)[24] intercept   : AIC=inf, Time=125.75 sec
 ARIMA(3,0,1)(1,0,1)[24] intercept   : AIC=inf, Time=38.97 sec
 ARIMA(2,0,1)(1,0,1)[24] intercept   : AIC=inf, Time=36.43 sec
 ARIMA(3,0,0)(1,0,1)[24]             : AIC=inf, Time=9.37 sec

Best model:  ARIMA(3,0,0)(1,0,1)[24] intercept
Total fit time: 953.444 seconds
```

ML Model 3 - Random Forest Regression

ENSEMBLE LEARNING

REDUCE INDIVIDUAL ERRORS

Verification Techniques: K-Folds

Split 1 | Split 2 | Split 3 | Split 4 | Split 5 | Split 6 | Split 7 | Split 8 | Split 9 | Split 10

```
cv10_score = cross_val_score(model_dtreg, trainX, trainY, scoring="r2", cv=10)

pred_score = cross_val_score(model_dtreg,  testX, testY, scoring="r2", cv=10)


dtreg_append = {'carpark':carpark, 'trainCV': np.mean(cv10_score), 'testCV': np.mean(pred_score)}
dtreg_CV = dtreg_CV.append(dtreg_append,ignore_index=True)
```

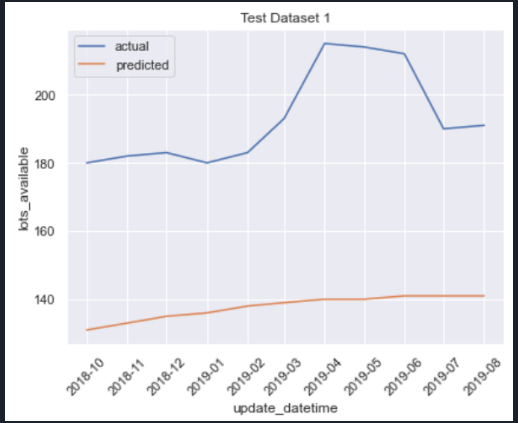
```
Random Forest Regression:
Cross-validation 10-Folds Average
```

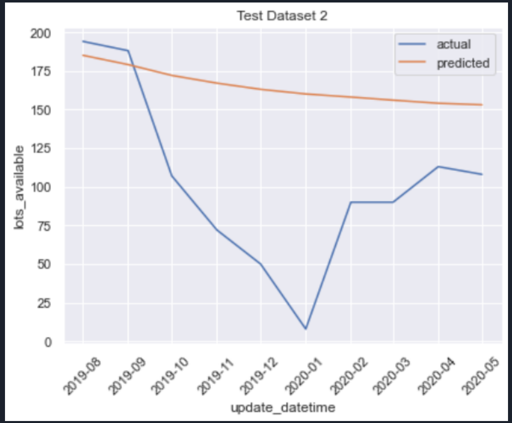
	carpark	trainCV	testCV
0	J70	0.656	0.613
1	J85M	0.931	0.602
2	J72	0.872	0.681
3	J69	0.734	0.453
4	JM31	0.903	-0.182
5	J62M	0.942	0.178
6	J63	0.803	0.607
7	J83M	0.971	0.656
8	J67	0.578	-0.390
9	JM2	0.893	0.211

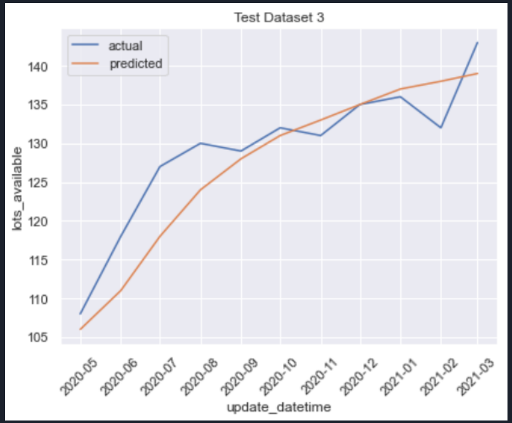
Verification Techniques: Forward-Chaining

Implemented for ARIMA model

Split 1 | Split 2 | Split 3 | Split 4 | Split 5





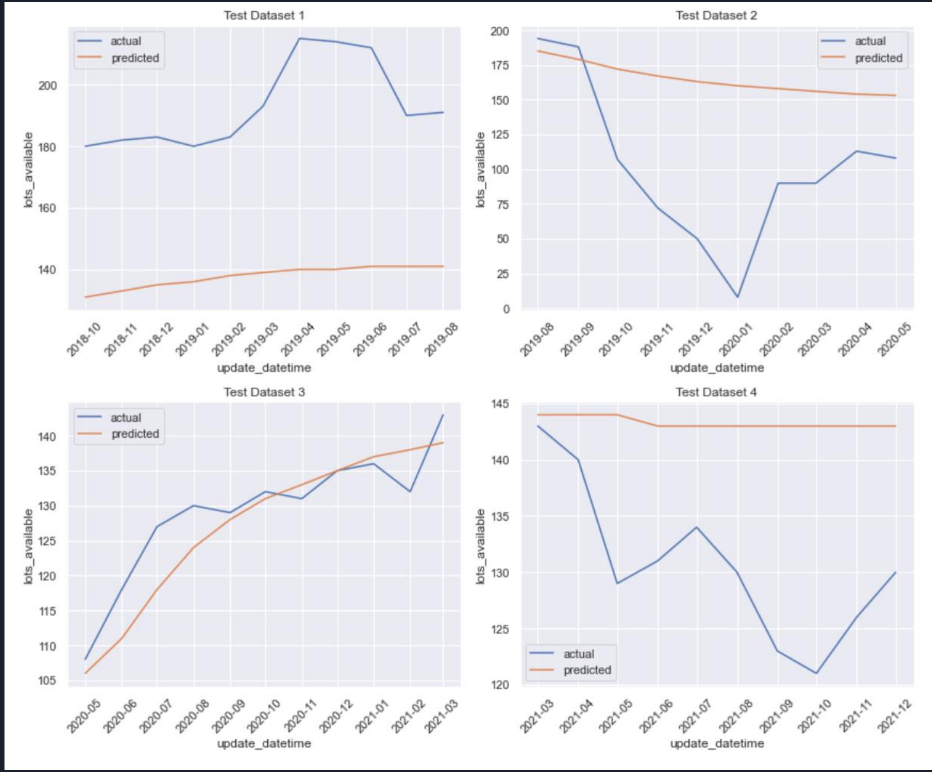


Comparisons

	Decision Tree Regression	ARIMA	Random Forest Regression
Advantages	• Fast results • Clear decision path that can be traced • Determines predictors that heavily affect outcome	• Constantly improving with each CV set • With auto_arima find best fit values for model	• Run efficiently on large datasets • Able to estimate and maintain accuracy over missing data periods
Disadvantages	• Works poorly with large dataset • Heavily inaccurate as hours are grouped together and given single value	• Works poorly with large dataset • Slow and long process to create model • Unable to forecast results	• Can be overfitted when fitted with noisy regression task • Can be bias towards categorical data

Comparisons Forward Chaining

ARIMA



Comparisons (10-Fold Cross Validation)

Decision Tree Regressor

```
Decision Tree:
Cross-validation 10-Folds Average
```

	carpark	trainCV	testCV
0	J70	0.316	0.509
1	J85M	0.740	0.344
2	J72	0.716	0.631
3	J69	0.709	0.499
4	JM31	0.476	-0.409
5	J62M	0.658	0.058
6	J63	0.400	0.602
7	J83M	0.920	0.623
8	J67	0.053	-0.058
9	JM2	0.731	-2.646

Random Forest Regression

```
Random Forest Regression:
Cross-validation 10-Folds Average
```

	carpark	trainCV	testCV
0	J70	0.656	0.613
1	J85M	0.931	0.602
2	J72	0.872	0.681
3	J69	0.734	0.453
4	JM31	0.903	-0.182
5	J62M	0.942	0.178
6	J63	0.803	0.607
7	J83M	0.971	0.656
8	J67	0.578	-0.390
9	JM2	0.893	0.211

Conclusion

Random Forest Regression

Random Forest Regression:
Cross-validation 10-Folds Average

	carpark	trainCV	testCV
0	J70	0.656	0.613
1	J85M	0.931	0.602
2	J72	0.872	0.681
3	J69	0.734	0.453
4	JM31	0.903	-0.182
5	J62M	0.942	0.178
6	J63	0.803	0.607
7	J83M	0.971	0.656
8	J67	0.578	-0.390
9	JM2	0.893	0.211

Conclusion, a forecast on 5th Jan 2022 Available Lots

	J70	J85M	J72	J69	JM31	J62M	J63	J83M	J67	JM2
0	0.020000	92.004000	4.207133	25.390000	113.297333	135.057517	0.850500	0.000000	9.557786	149.908000
1	0.020000	88.667333	4.177133	23.878000	112.317833	134.514017	0.265848	0.000000	7.993119	151.582000
2	0.020000	86.982910	4.131133	26.034200	112.203000	134.314683	0.219181	0.000000	5.620452	153.518000
3	0.026000	86.421176	4.221800	26.026500	111.979167	134.804183	0.251281	0.000000	3.625952	224.134971
4	0.020000	86.678010	4.431267	26.446500	111.791167	134.969683	1.086667	0.004000	3.474286	223.455976
5	6.712000	87.801267	6.970400	30.668000	111.724667	135.062767	7.312000	0.000000	1.712833	223.444743
6	5.580000	92.340000	12.412000	37.086000	111.357667	135.235767	10.266000	0.030000	1.524833	208.070000
7	29.502000	110.310000	31.458000	69.318000	112.063833	140.176000	25.510000	11.720000	1.360833	186.898000
8	41.496333	137.083000	77.156000	106.657500	110.750500	149.819333	52.947167	38.848500	5.150000	108.797000
9	54.142833	163.094000	106.693600	123.818000	115.851833	161.792000	76.440000	53.602000	7.328000	103.846000
10	53.238167	171.295400	115.482600	134.169000	116.001833	165.517500	79.654000	53.030095	7.712000	85.362000
11	47.971386	168.155500	116.575667	136.924100	101.270926	161.778200	81.813471	53.563829	8.362000	59.363167
12	17.626000	167.944500	116.106000	136.826600	82.428000	161.324800	81.732500	52.990667	11.600000	34.100000
13	21.086000	167.812500	113.661333	131.702400	98.062000	161.169600	78.413000	53.188000	19.164000	25.032000
14	5.842000	166.910000	110.846333	132.745000	112.100000	161.121500	77.172000	53.452000	3.136000	18.958000
15	56.588000	166.658000	111.787000	130.411000	120.528000	161.342000	75.902000	53.498000	2.802000	87.116000
16	14.961500	167.912000	102.375467	126.303000	123.853000	160.616000	68.142500	50.555500	2.714000	76.654667
17	0.546400	167.264267	89.302900	119.194000	119.914733	161.749667	69.026867	50.120000	2.538000	46.002000
18	0.000000	160.140000	72.920000	98.688000	95.500000	163.562000	52.412000	42.936000	19.620000	25.304000
19	0.000000	141.848000	45.756000	82.960000	93.490000	158.556000	40.162000	31.112000	1.212000	15.462000
20	0.010000	125.926000	37.612000	68.486000	95.890000	149.478000	25.608000	21.654000	1.086000	4.692000
21	4.756000	125.764000	32.588000	55.314000	107.828000	149.552000	11.164000	16.324000	1.076000	158.539600
22	0.000000	110.272600	25.736333	42.200667	113.181729	141.546100	6.764000	3.313500	0.628000	160.076500
23	0.000000	97.139693	18.328948	27.632634	113.409662	138.353276	1.056000	0.000000	0.621000	161.348538

	J70	J85M	J72	J69	JM31	J62M	J63	J83M	J67	JM2
19	0.000000	141.848000	45.756000	82.960000	93.490000	158.556000	40.162000	31.112000	1.212000	15.462000