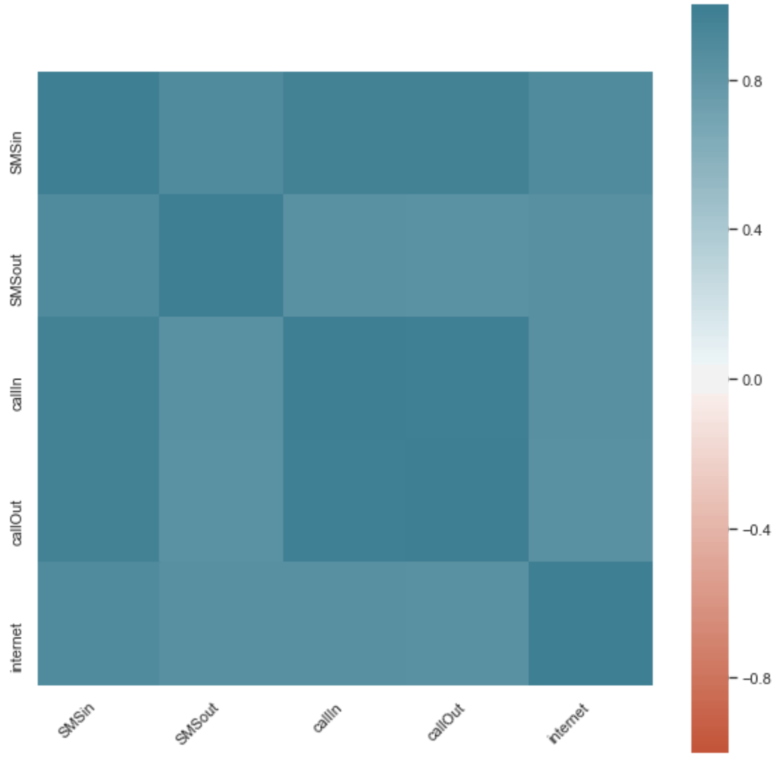
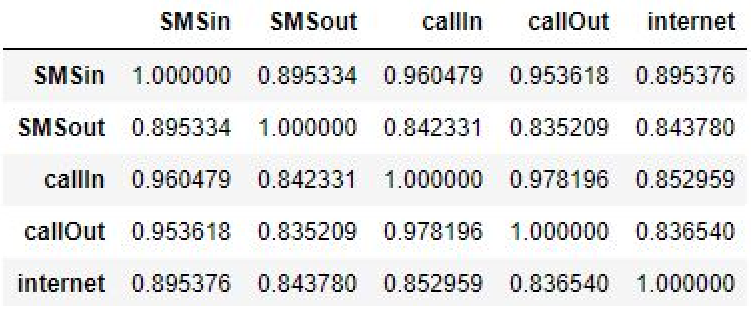
Capstone Project 1: Exploratory data analysis and inferential statistics

Correlation between variables:

Correlation matrix between the variables is generated below (Fig 1). Following variables have high correlation:

1. SMSIn and callIn
2. SMSIn and callOut
3. callIn and callOut



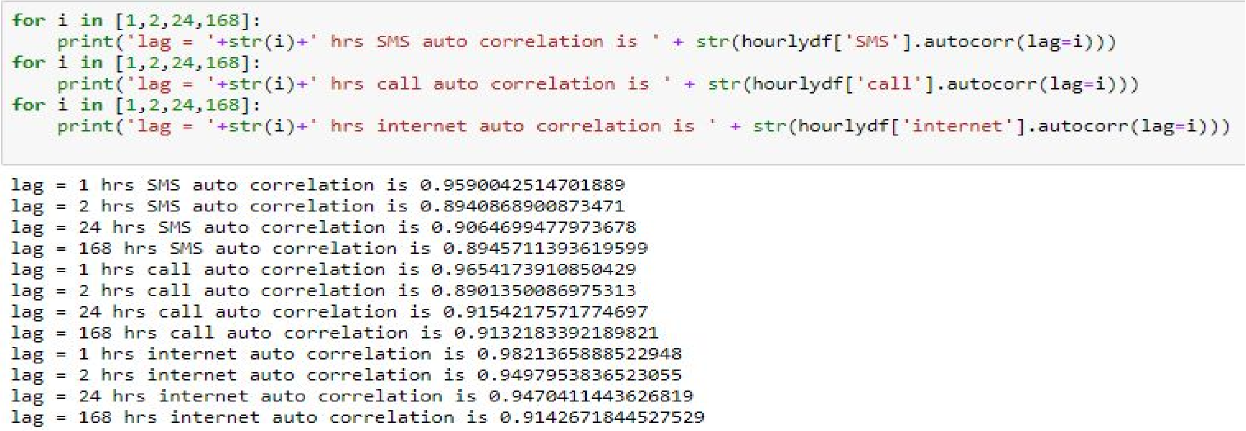


	SMSin	SMSout	callIn	callOut	internet
SMSin	1.000000	0.895334	0.960479	0.953618	0.895376
SMSout	0.895334	1.000000	0.842331	0.835209	0.843780
callIn	0.960479	0.842331	1.000000	0.978196	0.852959
callOut	0.953618	0.835209	0.978196	1.000000	0.836540
internet	0.895376	0.843780	0.852959	0.836540	1.000000

Fig1

Autocorrelation:

The data seems to be auto correlated. I’ve taken autocorrelation for different lag time periods, from one hour to 1 week (Fig 2).



```
for i in [1,2,24,168]:
    print('lag = '+str(i)+' hrs SMS auto correlation is ' + str(hourlydf['SMS'].autocorr(lag=i)))
for i in [1,2,24,168]:
    print('lag = '+str(i)+' hrs call auto correlation is ' + str(hourlydf['call'].autocorr(lag=i)))
for i in [1,2,24,168]:
    print('lag = '+str(i)+' hrs internet auto correlation is ' + str(hourlydf['internet'].autocorr(lag=i)))
```

```
lag = 1 hrs SMS auto correlation is 0.9590042514701889
lag = 2 hrs SMS auto correlation is 0.8940868900873471
lag = 24 hrs SMS auto correlation is 0.9064699477973678
lag = 168 hrs SMS auto correlation is 0.8945711393619599
lag = 1 hrs call auto correlation is 0.9654173910850429
lag = 2 hrs call auto correlation is 0.8901350086975313
lag = 24 hrs call auto correlation is 0.9154217571774697
lag = 168 hrs call auto correlation is 0.9132183392189821
lag = 1 hrs internet auto correlation is 0.9821365888522948
lag = 2 hrs internet auto correlation is 0.9497953836523055
lag = 24 hrs internet auto correlation is 0.9470411443626819
lag = 168 hrs internet auto correlation is 0.9142671844527529
```

Fig 2

It is observed that 1 hour correlation is high, but the data would not have moved much from the previous hour, for the same reason we cannot use this as we have to find a pattern. Other than 1 Hr the 24 Hrs lag has the highest correlation. (Fig 3)

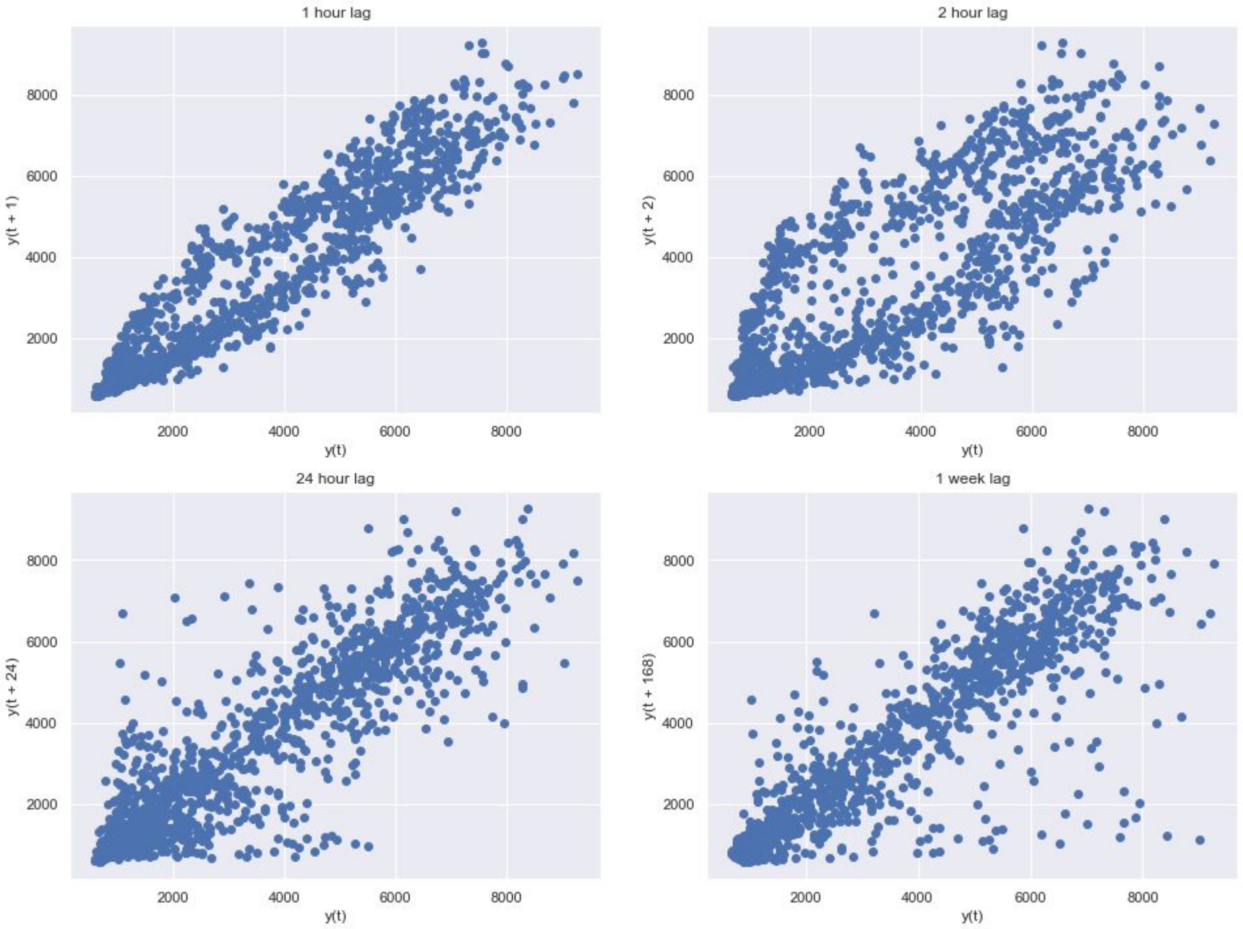

1 hour lag
y(t + 1)
y(t)
2 hour lag
y(t + 2)
y(t)
24 hour lag
y(t + 24)
y(t)
1 week lag
y(t + 168)
y(t)
2000
4000
6000
8000

Fig 3

Fitting the data to a model:

The data seems to be following a sine distribution over a 24 hour period. A simple sin function is defined to fit the model of the data (Fig 4)

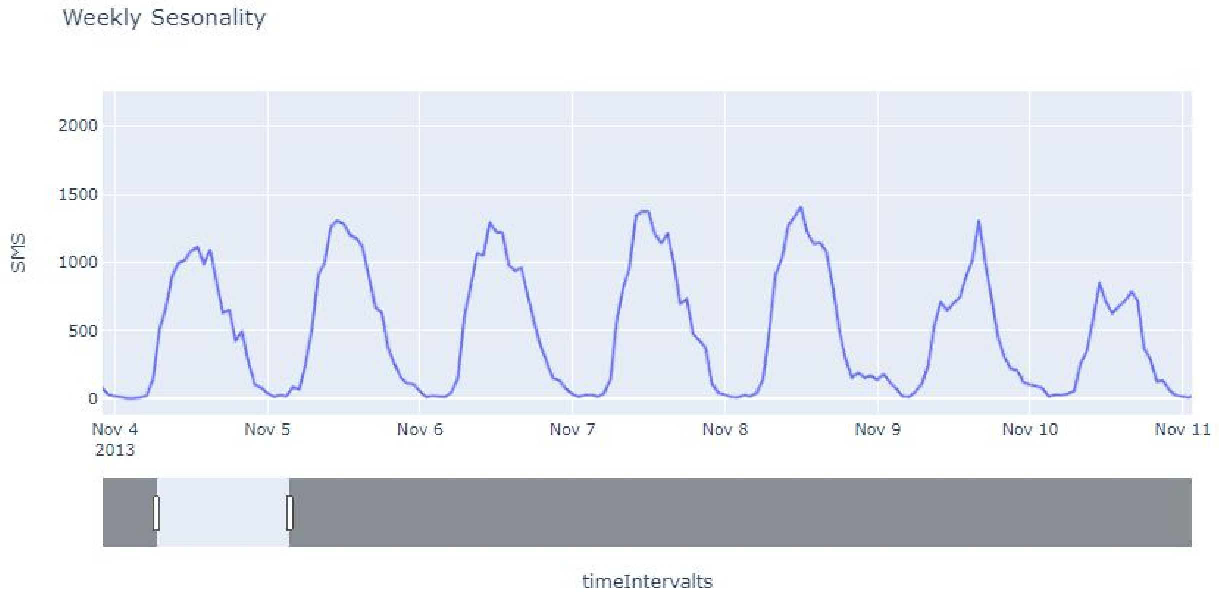


Fig 4

Curved fit of the data is extracted using the scip.stats.optimize.curve_fit module. (Fig 5)

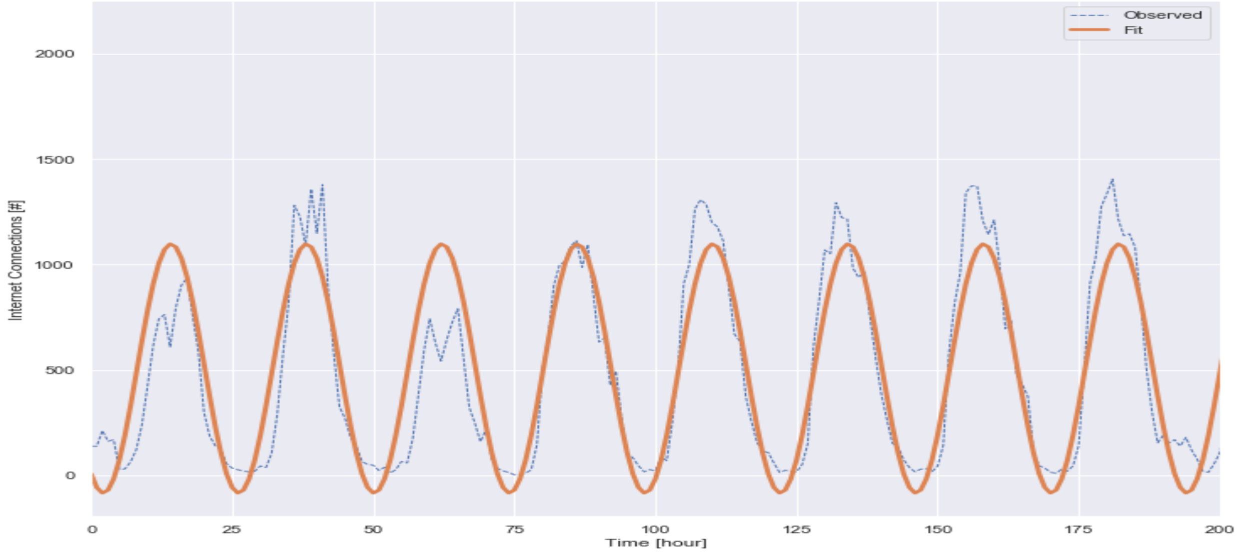
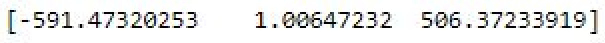


Fig 5

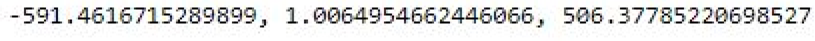
The optimized parameters of the model is as below:



```
[-591.47320253    1.00647232   506.37233919]
```

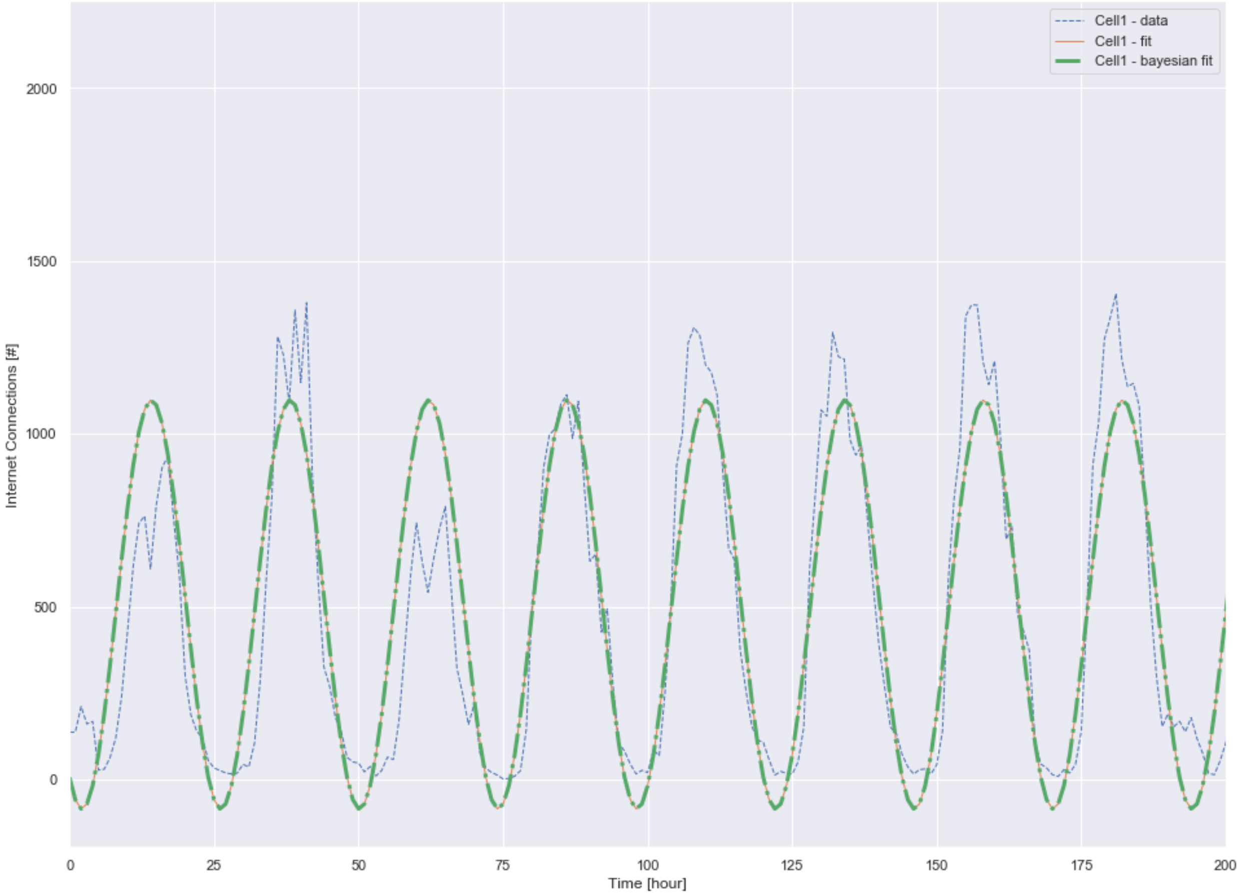
Bayesian Inference to verify the model parameters:

The optimized parameters derived from the curved_fit model are used to simulate Bayesian Inference to further optimize the parameter. The sine function defined earlier is passed as a parameter along with the other parameters and simulated for 100,000 times. The resulting parameters are as below:



```
-591.4616715289899, 1.0064954662446066, 506.37785220698527
```

The observed data, curve fit model and Bayesian fit model are plotted below. The curve fit model and Bayesian inference both seem to be very close.



Conclusion: The parameters derived from scipy stats curve fit and Bayesian inference are almost the same, we can infer that the parameters to represent the sin distribution is accurate based on the available data.