

# Call Center – Working Hours

Miroslav Novak, miroslav.novak@gmail.com

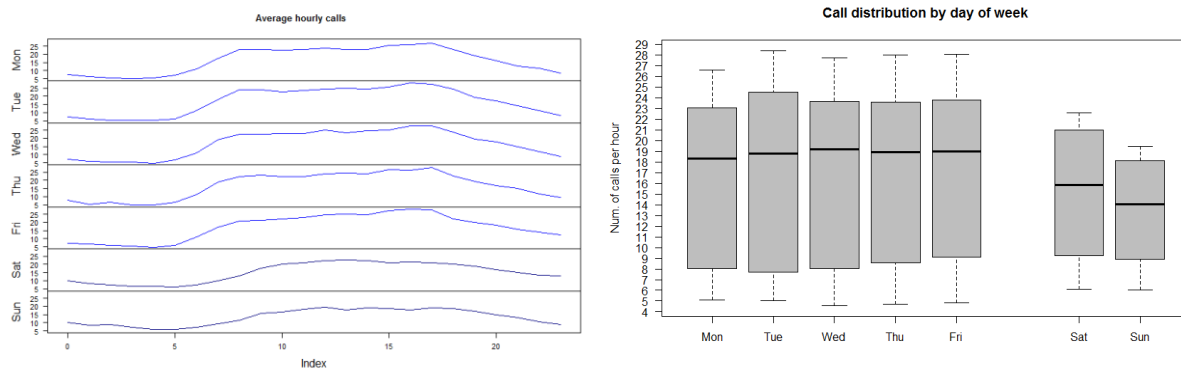
## Abstract

*This document describes a solution for M/D/c problem. The solution was found by using a discrete event simulation which employs prediction of peaks based on weather forecast data. The solution can be easily generalized to provide answer to more realistic variations of the problem with more constraints (e.g. compliancy with working hour regulations). The simulation can provide quick validation for more complex predictive algorithms (using richer data sets, finer granularity ...).*

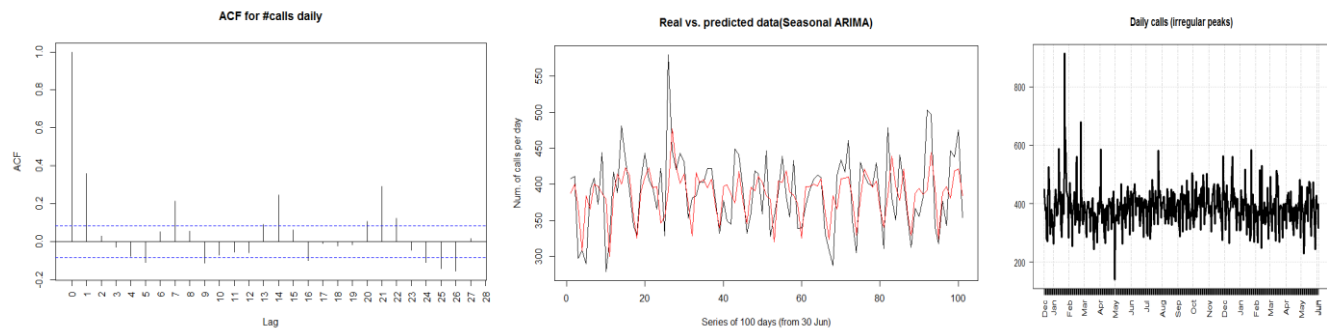
## Data set analysis

Provided data set was used to propose call center working hours. The data manifests several patterns:

- A similar trend for working days (Mon-Fri). The number of call starts growing around 8 a.m. and declines at 7 p.m. There are more call during working day than during weekends.
- For weekends, the peak is shifted (there is less of activity on early weekend mornings).

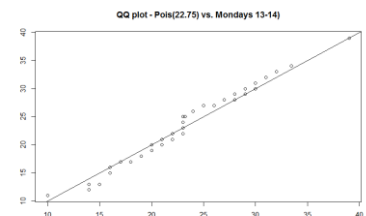


- The weekly correlation is nicely revealed by autocorrelation function (ACF). There are peaks at lag of 7, 14, 21 days.
- Seasonal ARIMA model (1,0,1)x(1,0,1) fits the time series of daily calls quite well. It does not provide explanation for the peaks above 500 calls a day (discussed later).



- Distribution of calls in a “stable” period matches nicely Poisson distribution (*rate of call varies over time but is “stable” for certain periods – as in the example below*).
- For longer time period, calls do not follow Poisson distribution (*see analyze\_data.R at [6]*)

*Note: All R scripts generating the charts and transforming the data are available within [6] together with complete source code.*



## Theory & prior solutions

Well-known queuing theory, coined by Erlang [1], provides apparatus for calculations of probabilities and expected values of the queue length and waiting times. According to the standard notation [2], our problem is described as M/D/c, assuming that each call takes constant amount of time (2 minutes). At first, it was solved by Erlang and later studied and simplified by various mathematicians [3], [4]. If it was required to generalize and assume service times with exponential distribution, the model would be M/M/c.

Alternatively, discrete event simulations are used to find an optimal solution to a queuing problem. Simulations have several major advantages in comparison to analytical models:

- It is easy to incorporate additional constraints which are typical for real-world applications [5].
- Queuing models provide little help with irregular peaks and assume independence of events which is violated e.g. during severe weather events.
- Queuing models provided limited answer to more complex KPIs (e.g. 80% of customers served within 20 seconds).

## Static call center capacity

Using the simulation (*callcenter.py* & *simulation\_basic.py*), performance of a simple working hour schedule was tested (note that data was cut off to the whole weeks for simplicity, see *filter\_data\_whole\_weeks\_only.R* at [6]).

1 assistant on duty <b>24x7</b> (Mon-Sun)	<b>Mon-Fri, 1 extra 8-20h</b>	<b>Sat-Sun, 1 extra 10-20h</b>
---	-------------------------------	--------------------------------

With this schedule, the average waiting time is **21s** and QoS (% of customer waiting less than 20s) is **86%**. However, there are 20 calls with wait time above 83 minutes. Longest waiting times would be experienced on 24<sup>th</sup> Feb 2016.

Another disadvantage is that the utilization of the call center is low: **36%**.

## Extra capacity based on severe weather warnings

In order not to worsen the utilization but address extreme peaks, an extra capacity predictor was built. Data from Storm Prediction Center [7] was used. Textual data was converted (*weather\_extract\_features.R*) to numeric features used by the predictor. Gradient boosting regressor (*train\_weather.py*) was used to predict number of calls for severe weather warnings. Because of low amount of training data, 4 least contributing features were removed (tornados, count of affected states, year & day).

The predictor (*prediction.py*) was used to assess number of extra employees. The simulation using the predictor (*simulation.py*) lowers avg waiting times to **15s** and improves QoS to **86.4%** while not worsening the utilization.

## Future work

The implemented predictor is quite naïve and could be significantly improved. The root cause of extreme peaks should be identified, supporting data sets collected and the need for extra call center employees should be predicted with hour granularity.

## References

- [1] Erlang, A.K. (1909, 1920), "The theory of probabilities and telephone conversations" and "Telephone waiting times"
- [2] Kendall, D. G. (1953). "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain"
- [3] Crommelin, C.D. (1932), "Delay probability formulas when the holding times are constant"
- [4] Franx, G. J. (2001). "A simple solution for the M/D/c waiting time distribution"
- [5] Linda Green (2006), *QUEUEING THEORY AND MODELING*
- [6] All resources related to the solution (R scripts, Python implementation, images ...) – [Google Drive](#)
- [7] Storm Prediction Center – archived events ([2016 & 2017, PA & NJ](#))