# Synthetic Call Detail Records generator

Milita Songailaitė[a], Tomas Krilavičius[b]

[a]Vytauto Didžiojo universitetas, Mathematics and Statistics dept.
[b]Vytauto Didžiojo universitetas, Applied informatics dept.

**Abstract**

Researchers often have to deal with the problem of data availability and privacy restrictions. This is why synthetic data generation is usually a solution for the deficiency of research data and a way to provide the researchers with the data fitting specific situations. This paper proposes a system of synthetic Call Detail Records (CDR) data generation based on a real-life dataset. Our approach uses statistical analysis to extract the users' behaviour from the real-life dataset. The generator is tested by simulating several unique situations, which appear in the real world. Since this kind of situations can be hard to capture into the dataset, we believe that the proposed system could be useful in many telco research fields.

**Keywords**

CDR, Synthetic data generation, statistical analysis

## 1. Introduction

Telecommunications play a vital role in people's lives and provide opportunities for global communication. Due to the increased use of smart technologies, this industry has grown globally in recent years, and this growth is not expected to reduce for some time to come [1]. These companies' primary resource is customers, so a proper understanding of their behaviour and data management is one of the most critical factors for its success. However, most of this data is restricted by the user's rights and is not easily accessible for researchers and data analytics. To avoid these restrictions, we propose a method of synthetic telecommunications data generation, which will allow the researchers to generate their own data based on real-life situations or augment the already obtained dataset. Furthermore, synthetically generated data might provide the researchers with less frequent situations like some instances of fraud, anomalies or even breakdown of servers. This paper will mostly focus on the process of generating the CDR data and simulating some simple occurring problematic situations.

Call detail records (CDRs) are the records, which usually store the caller's and the receiver's numbers, the date and time when the call starts, the time the call connects, and the time the call ends [2]. CDR data is commonly used in the telecommunications industry as a primary source of information about the client. The records can be used to calculate the profitability of specific customers in the future, predict customers churn rate or notice the upcoming fraud case. The generator system described in this paper focuses on generating classical structure CDR files.

*IVUS2021: Information Society and University Studies 2021, April 23, 2021, Kaunas, Lithuania*

milita.songailaite@stud.vdu.lt (M. Songailaitė); tomas.krilavicius@vdu.lt (T. Krilavičius)

The generator rules are designed in such a way to fit real-world data with the possibility to alter specific parameters and create different situations for various testing purposes.

The rest of the paper is organised as follows. Related work is provided in Section 2. The generator system and generated results are discussed in Section 3. The generated simulations of real-world situations are provided in Section 4. The conclusions are given in Section 5.

## 2. Related work

In this paper, the analysed data covered two main problems: synthetic real-world data generation and the problems that arise for telecommunication companies and researchers when analysing or working with CDR data.

Any human-related data collecting can be very time consuming and restricted by various security measures. To overcome these difficulties, several synthetic data generation approaches were proposed. The system described in [3] presents a framework that generates web applications data from various layers, including networking, database queries and security checks. The authors generated nine synthetic datasets with normal and attack data using multiple different parameters. The data was used to simulate standard and malicious web application usage, therefore testing various security and reliability layers. Another field of synthetic data generation was introduced in [4]. The authors proposed a machine learning-based data generation method to generate home-based activity data for healthcare applications. The synthetic time series were generated using hidden Markov models and regression models, which were initially trained on real-life datasets. Lastly, [5] suggested a similar attempt to generate telecommunications data by imitating users behaviour described by statistical distributions. Our approach differs in that we simulate the data according to some particular situations occurring in real life. Similarly, as in [3], we define situations that might be useful to test for companies or researchers working in the telco field.

In order to better understand what problematic situations occur when working with CDR data, we analysed several typical areas of telco research. A number of research papers [6, 7, 8] investigated occurring telco fraud instances, which can be found in CDR data. The methods used to detect fraud included various clustering and machine learning algorithms. Although the proposed systems were mostly based on CDR data, the majority of them required additional information about users, like their payment history and geographical locations. Researchers in [9] described their approach to estimate poverty rates in Guatemala only by analysing CDR data. According to them, CDR data provides the potential to gather detailed and reliable estimates of poverty rates in real-time at a far lower cost than traditional surveys. Finally, papers [10, 11] analysed what potential information can be obtained from CDR data combined with the users' geolocation.

## 3. Generator overview

The goal of presented generator is to generate CDR data that is realistic enough to be used by researchers and telco companies. Moreover, we wish to simulate certain real-world situations which could be hard to capture even in manually gathered datasets. To achieve this, we created

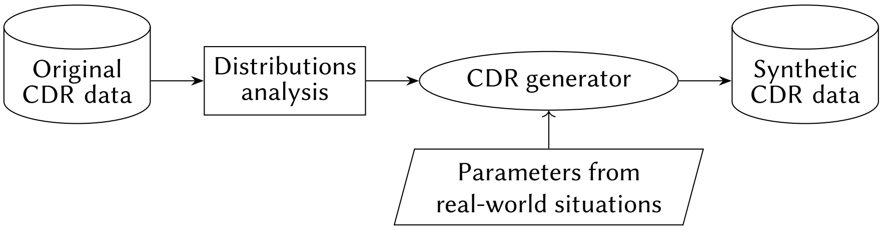

**Figure 1:** Workflow of the CDR generator

a synthetic CDR data generator system, which is depicted in figure 1. First, to capture real-world users' behaviour, the real data had to be collected. After this, we performed statistical analysis of the selected call record features, which our system will generate. Then, we carried out the analysis of possible problematic situations in telecommunications and implemented some of them in the generator system. Finally, the results were simulated according to the selected simulations rules.

## 3.1. Original CDR dataset

In order to capture typical users' calling behaviour, the analysis of real-world CDR data is needed. For this, we analysed data provided by a telecommunications company. The data consisted of over 20 thousand customers and six months of their activities. The real-life dataset had over 300 thousand call records collected in the year 2020. Moreover, it also contained failed (missed, denied or suspended for other reasons) calls records, which meant that we could precisely capture the call's likelihood of success. Each call record had its precise date, caller id, receiver id, destination name and call duration. Our generator implements most of these features; however, since the data will be simulated in one country locally, we decided to replace the call destination with the receiver's telecommunications company name. Also, to simulate the relations between different telco companies, each caller was assigned a company. The final feature vector is: *Caller's id, Caller's company, Receiver's id, Receiver's company, Timestamp, Duration s.*

## 3.2. Dataset feature analysis

Statistical analysis was used to extract features from the original dataset. The analysis consisted of several distinct parts: call durations, count and calling hour distributions analysis, most active weekdays analysis, finding the likelihood of the user making an international call and the probability of the call success. We used the distribution fit tests library *fitdistrplus* [12], provided by the statistical programming language R to tackle the distribution detection task. The *fitdistrplus* function estimates distribution parameters by maximising the likelihood function. After that, the results are displayed using Cullen and Frey (figure 2) graph, which depicts each distribution's skewness and kurtosis.

The blue dot on the graph represents skewness and kurtosis of the tested distribution. The closer the dot is to the symbol depicting particular distribution, the better this distribution fits

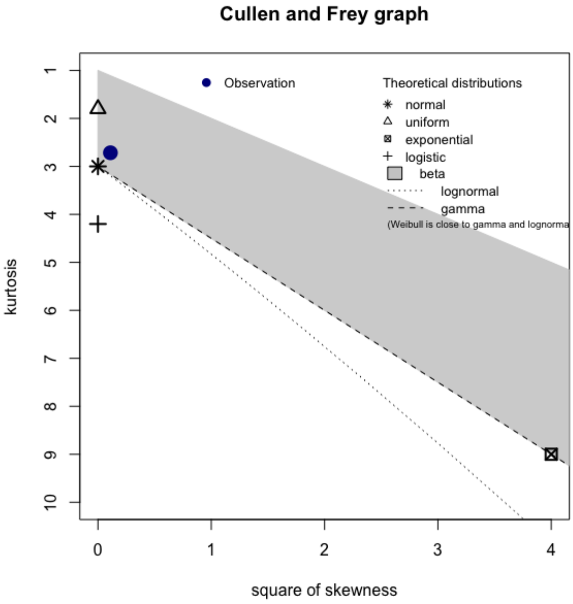

**Figure 2:** Calling hour Cullen and Frey graphs. In this case, the best-fitting distributions are normal, lognormal, gamma and Weibull's distributions

original data. To measure each distribution goodness-of-fit to the data, we considered four methods provided by the R library *fitdist*:

1. density plot, which shows the histogram of empirical data with fitted distribution's density function,
2. empirical and fitted distributions CFD plot,
3. quantile plot (Q-Q plot) depicting the empirical quantiles (y-axis) against the theoretical quantiles (x-axis)
4. probability plot (P-P plot) representing the empirical distribution evaluated at each data point (y-axis) against the fitted distribution function (x-axis).

Figure 3 shows the resulting goodness-of-fit plots. Ideally, in each graph, the empirical parameters should match the fitted distribution parameters.

The rate of call success was calculated simply by finding the proportion of all successful and unsuccessful calls. It was then converted to the probability of call success. Lastly, the original data was not used to find the likelihood of making a local or an international call. This was because the data was obtained from an international telecommunications company; thus, the international calls rate would have been biased compared to the local calls.

### 3.3. Dataset features results

The methods described above were used to extract features from the original dataset. Based on the statistical analysis, we extracted the main features of each call record feature. The call

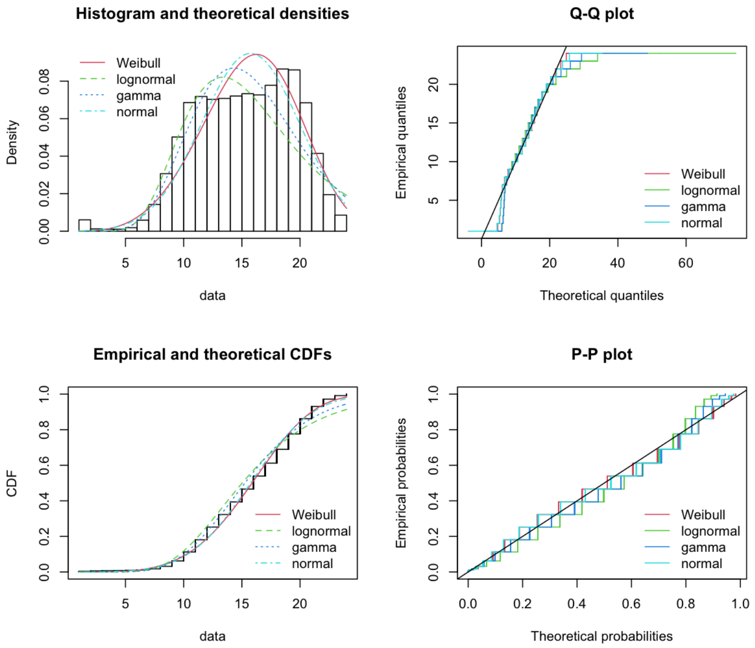

**Figure 3:** Calling hour fit goodness graphs: density plot (top left), quantile plot (top right), empirical and theoretical CDR plot (bottom left) and probability plot (bottom right)

**Table 1**
Estimated customer behaviour features distributions

| Feature | Distribution | Parameters | |
|---|---|---|---|
| Call duration | Weibull's | Shape: 0.61 | Scale: 413.62 |
| Call's count | Weibull's | Shape: 0.74 | Scale: 11.13 |
| Calling hour | Normal | Mean: 15.74 | Std: 4.21 |
| Most busy weekdays | Uniform | From: 1 | To: 7 |

durations ranged from 1 s (all unanswered calls were left out of the sample) up to 7000 s, and the median of call durations was 247 s. Each user's number of calls ranged from 1 to 233 calls, with a median of 5 calls per user. The most regular calling hours were 7 p.m. to 8 p.m.; moreover, the day's busiest period was from 10 a.m. to 9 p.m. Finally, the number of calls made during each weekday was similar; therefore, the weekdays' distribution is uniform. The estimated theoretical distributions of each feature are given in the table 1.

The percentage of successful calls was 70%; thus, the estimated probability of the call success was set to be $P(\text{callSuccess}) = 0.7$.

## 3.4. System design

The generator was implemented using Python programming language. There are five distinct classes: local operator, international operator, local customer, international customer and CDR

**Table 2**
The sample of generated CDR file

| Caller's id | Caller's company | Receiver's id | Receiver's company | Timestamp | Duration s. |
|---|---|---|---|---|---|
| 42511 | Kompanija1 | 28736 | Kompanija1 | 04-12-20 00:46 | 196 |
| 45839 | Kompanija1 | 50378 | Kompanija1 | 04-12-20 05:25 | 6269 |
| 48650 | Kompanija2 | 48231 | Kompanija1 | 04-12-20 05:52 | 1783 |
| 48231 | Kompanija1 | 38147 | Kompanija1 | 04-12-20 09:48 | 1368 |
| 27938 | Kompanija1 | 41790 | Kompanija2 | 04-12-20 09:58 | 225 |

record. The parameters for each of these classes are provided in a .json format configuration file. That file contains basic information about the generation: total customer number, total operator number, generation time interval and max number of local and international friends that one user can acquire.

After the parameters are adjusted, the generator starts creating objects accordingly to the given five classes. First, it creates operators with the market share parameter. The more market the operator holds, the more customers it has. Market shares allocation is implemented in such a way that one company could not have more than 70% of all the customers. Next, the international and local operators and customers are created. Each customer is assigned a local operator. Since we decided to generate only local companies' CDR data, a customer cannot be assigned an international operator. After all the customers had been assigned an operator, the generator starts building a network of people. Each customer can have a number of friends and acquaintances. Both numbers are chosen randomly from the interval from 1 to max number of friends/acquaintances. After that, a customer can make a call to a friend, acquaintance or the person not in his network with the probabilities: $P(\text{callFriend}) = 0.5$, $P(\text{callAcquaintance}) = 0.3$ and $P(\text{callOther}) = 0.2$. Finally, the rest of the call record features are generated randomly according to the estimated distributions.

## 3.5. Generation results

The data generation experiment was performed with selected parameters: 10 000 customers, three operators, max friends number: 5, max acquaintances number: 10, start date: 2020-12-04. A sample of generated CDR results is given in the table 2.

When analysing generated data, we mainly focused on the features' samples from estimated distributions 4. Each of the four chosen call record features followed the distributions of real-world data almost precisely. However, there were some inadequacies noticed when analysing calling hour and weekdays. Even though the busiest calling hours in the collected data were 7 p.m. and 8 p.m., the model almost always generated most calls during the mid-day period (from 2 p.m. to 7 p.m.). The generated weekdays sample is close to being distributed uniformly; however, most of the time, the end of the week is busier than other weekdays.

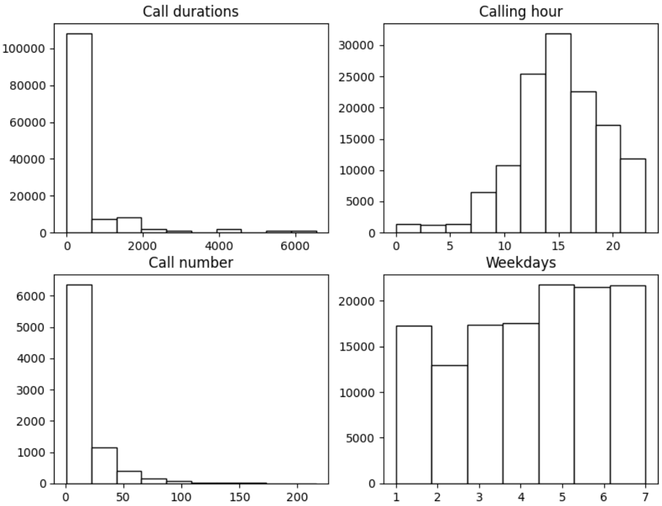

**Figure 4:** Generated data call duration, calling hour, call number and weekday distributions

# 4. Real-world situations simulations

To better understand what problems might occur in telecommunications, we asked the telco experts to consult us on the topic. According to them, there are four most common situations which are considered problematic in the telco field:

1. Increased load in the whole area. The best example of a situation like that would be new years eve. The call number in the CDR data is increased for a short time period.
2. Increased load at one certain place. These situations arise in the places of crowd events, for instance, a concert. In this case, one telecommunications station has to allocate more calls then it normally should. Therefore, it starts crashing, and the calls fail.
3. A natural disaster, for example, an earthquake. In such a case, there will be bigger traffic to that location; moreover, due to possible high numbers of casualties, many calls might also fail.
4. Cell tower failure. The consequences of this type of situation are similar to those caused by natural disasters.

Only two situations were chosen to test the generator: increased load in the whole area and the cell tower failure. The other situations were not included since they required the knowledge of an accurate user location, which the CDR data does not provide.

## 4.1. Increased load in the whole area simulation

For this experiment, we chose to simulate the CDR workflow during New Year's Eve. The main change in this situation is the increased call amount made by every user. The increment for

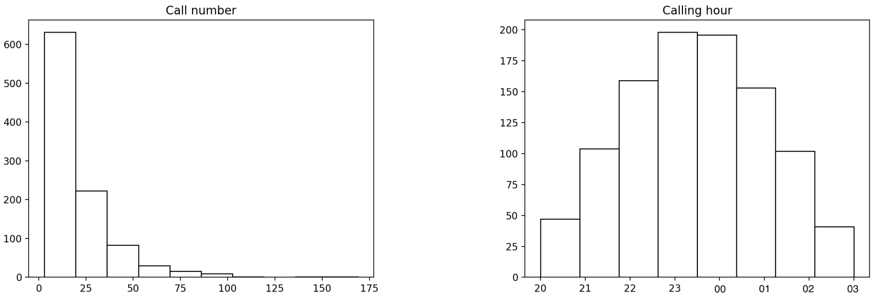

**Figure 5:** Distributions of event number of calls (left) and calling hours (right).

each user was chosen randomly using the normal distribution. The parameters for the normal distribution were chosen according to the equations (1, 2). Let $N_{\text{friends}}$ and $N_{\text{acquaintances}}$ be the number of friends and acquaintances respectively for every user. Then the parameters of event calls number distribution would be:

$$\text{mean} = \frac{N_{\text{friends}} + N_{\text{acquaintances}}}{2} \quad \text{std} = 1, \tag{1}$$

$$\min = 0 \quad \max = N_{\text{friends}} + N_{\text{acquaintances}} \tag{2}$$

The call record timestamp was also generated using the normal distribution with the parameters shown in the equations (3, 4), where $t_{\text{event}}$ is the event duration. The day and calling minute was chosen uniformly during the event period. The distribution parameters and the rest of the variables were determined by a series of experiments. However, the actual generator has the possibility to adjust the model for it to fit certain other situations by fixing the parameters in the input files.

$$\text{mean} = \frac{t_{\text{event}}}{2}, \quad \text{std} = \frac{t_{\text{event}}}{4}, \tag{3}$$

$$\min = 0, \quad \max = t_{\text{event}} \tag{4}$$

The simulated event started at 8 p.m. and ended at 4 a.m. Figure 5 shows the resulting distributions of the newly generated features for 1 000 users. The majority of the users made less than 25 calls during the whole event, and most calls were made at around midnight.

### 4.2. Failed station simulation

The failed station simulation was created in the following steps. First, the company responsible for the failed station is chosen at random. Since we do not generate user locations, we have to assume that a certain number of users will be in the failed station range. To simulate that, we introduce a parameter representing the customers' percentage, one or more failed stations has to service. This parameter will be the probability that the customer is in the station range.

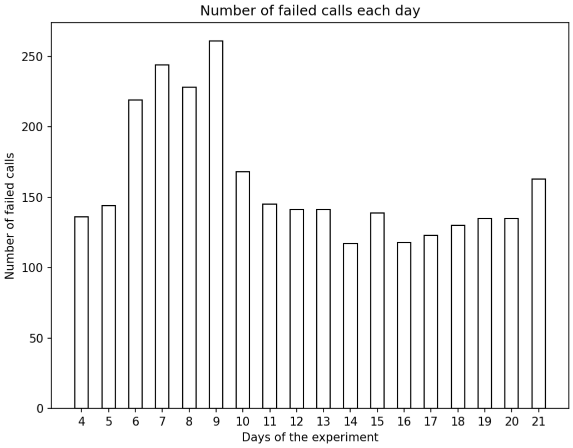

**Figure 6:** Failed call amount during the normal and event days. The event occurs from 2020-12-06 to 2020-12-10. The increase of the failed calls is noticeable during the event days (between 6th and 10th)

After that, we start simulating the failed calls. At the given time period, with the specified probability, we select calls from the CDR list and change their duration to 0s.

The simulated situation had the parameters: *station_size* = 0.3, *start_date* = 2020-12-06, *end_date* = 2020-12-10. The size of the failed area was chosen so large for a better demonstration. We calculated the percentage of failed calls during normal days and the event period to analyse the results. During the four days that event lasted, the number of failed calls increased by 70-90%. Figure 6 shows the failed calls' histogram during normal days and the event period.

Even though only one company's tower failed, this failure affected every company in the country. Whilst the companies with broken tower failed calls increased by 60%, failed calls in other companies had also increased by 30%. This phenomenon occurs because the customers from other companies might also call to the failed tower area. Whereas the people in the failed cell tower's range has no connection, the call can not reach its destination and fails.

## 5. Conclusions

This paper proposes an idea of abstract synthetic data generation tool for call records generation. The system can be based on any telco operator data and generate simulations according to the prefered parameters. The concrete system described in this paper was based on the real-life telecommunications dataset, which was analysed using statistical analysis. We used distribution fit tests to extract monthly call number, call duration, calling hour and most popular weekdays from the dataset. The parameters were then implemented into the generator, and the CDR data was simulated according to three distinct situations: 0 - normal data flow, 1 - increased load in the whole area, 2 - cell tower failure. The motivation for this kind of tool arose from the need for distinct situations when analysing telecommunications data. Our generator focuses on simulating situations, which could be detected only from CDR data without additional information about users or telecommunication services providers. The generator's source code and the instructions for others to use and expand the system are provided in the GitHub repository: https://github.com/CARD-AI/CDR-Generator.

## References

[1] *Global Telecom Services Market Size & Share Report, 2020-2027.* 2020. URL: https://www.grandviewresearch.com/industry-analysis/global-telecom-services-market (visited on 12/29/2020).

[2] *Cisco Unified Communications Manager CDR Analysis and Reporting Administration Guide for Cisco Unified Communications Manager Release 6.0(1).* Tech. rep. 1981. URL: http://www.cisco.com.

[3] Nathaniel Boggs et al. *Synthetic Data Generation and Defense in Depth Measurement of Web Applications.* Tech. rep. 2014. URL: http://heartbleed.com/.

[4] Jessamyn Dahmen and Diane Cook. "SynSys: A Synthetic Data Generation System for Healthcare Applications". In: *Sensors* 19.5 (Mar. 2019), p. 1181. ISSN: 1424-8220. DOI: 10.3390/s19051181. URL: https://www.mdpi.com/1424-8220/19/5/1181.

[5] A. Murtic et al. "SNA-based artificial call detail records generator". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings.* Institute of Electrical and Electronics Engineers Inc., June 2018, pp. 1226–1230. ISBN: 9789532330977. DOI: 10.23919/MIPRO.2018.8400222.

[6] Alae Chouiekh and El Hassane Ibn El Haj. "ConvNets for fraud detection analysis". In: *Procedia Computer Science.* Vol. 127. Elsevier B.V., Jan. 2018, pp. 133–138. DOI: 10.1016/j.procs.2018.01.107.

[7] Kashif Sultan, Hazrat Ali, and Zhongshan Zhang. *Call Detail Records Driven Anomaly Detection and Traffic Prediction in Mobile Cellular Networks.* Tech. rep. 2018. URL: https://orcid.org/0000-0003-3058-5794.

[8] Mhair Kashir and Sajid Bashir. "Machine learning techniques for SIM box fraud detection". In: *2019 International Conference on Communication Technologies, ComTech 2019.* Institute of Electrical and Electronics Engineers Inc., Mar. 2019, pp. 4–8. ISBN: 9781538651063. DOI: 10.1109/COMTECH.2019.8737828.

[9] Marco Hernandez et al. *Estimating Poverty Using Cell Phone Data Evidence from Guatemala.* Tech. rep. 2017. URL: https://openknowledge.worldbank.org/handle/10986/26136.

[10] Yuxiao Dong et al. "Inferring Unusual Crowd Events from Mobile Phone Call Detail Records". In: (2015). DOI: 10.1007/978-3-319-23525-7. URL: http://www.d4d.orange.com.

[11] Nai Chun Chen et al. *DATA MINING TOURISM PATTERNS Call Detail Records as Complementary Tools for Urban Decision Making.* Tech. rep. 2017.

[12] Marie Laure Delignette-Muller. *fitdistrplus: An R Package for Fitting Distributions.* Tech. rep. 2014. URL: https://cran.r-project.org/web/packages/fitdistrplus/vignettes/paper2JSS.pdf.