**DATA SAMPLING:**

|  |  |  |
| --- | --- | --- |
| **Roll No** | **Name** | **Contribution** |
| CB.EN.U4CSE19236 | Praveen | Data sampling |
| CB.EN.U4CSE19241 | Sameekshaa | Data sampling |
| CB.EN.U4CSE19249 | Suvethaa | Data sampling |

**Abstract:**

Data sampling is a statistical analysis technique used to select, manipulate and analyze a representative subset of data points to identify patterns and trends in the larger data set being examine. Sampling can be particularly useful with data sets that are too large to efficiently analyze in full for example, in [big data analytics](https://searchbusinessanalytics.techtarget.com/definition/big-data-analytics) applications or surveys. Identifying and analyzing a representative sample is more efficient and cost-effective than surveying the entirety of the data or population.

An important consideration, though, is the size of the required data sample and the possibility of introducing a [sampling error](https://whatis.techtarget.com/definition/sampling-error). In some cases, a small sample can reveal the most important information about a data set. In others, using a larger sample can increase the likelihood of accurately representing the data as a whole, even though the increased size of the sample may impede ease of manipulation and interpretation.

**Dataset description**:

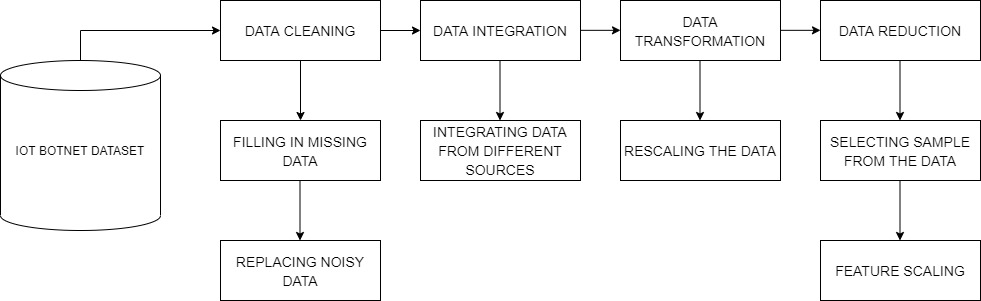
The IoT botnet dataset was created by the university of New South Wales by designing a realistic network environment. IoT botnet dataset divides traffic into normal traffic and botnet traffic based on the number of packets, transaction state, number of bytes in transaction and etc.

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **ATTRIBUTE TYPE** | **PURPOSE** |
| pkSeqID | numeric | Row identifier |
| stime | numeric | Gives recorded start time |
| flgs | categorical | Flow state flags seen in transaction |
| proto | categorical | Gives the textual representation of transaction protocol present in network flow |
| saddr |  | Gives source IP address |
| sport | numerical | Gives source port number |
| daddr |  | Gives destination IP address |
| dport | numeric | Gives destination port number |
| pkts | numeric | Gives the total number of pockets in the transaction |
| bytes | numeric | Gives the total number of bytes in the transaction |
| state | categorical | Gives the transaction state |
| ltime | numeric | Gives the recorded last time |
| seq | numeric | Gives argus sequence number |
| dur | numeric | Gives the record total duration |
| mean | numeric | Gives the average duration of aggregated records |
| stddev | numeric | Gives the standard deviation of aggregated records |
| smac |  |  |
| dmac |  |  |
| sum | numeric | Gives the total duration of aggregated records |
| min | numeric | Gives the minimum duration of aggregated records |
| max | numeric | Gives the maximum duration of aggregated records |
| soui |  |  |
| doui |  |  |
| sco |  |  |
| dco |  |  |
| spkts | numeric | Gives the source to destination packet count |
| dpkts | numeric | Gives the destination to source packet count |
| sbytes | numeric | Gives source to destination byte count |
| dbytes | numeric | Gives destination to source byte count |
| rate | numeric | Gives the total packets per second in transaction |
| srate | numeric | Gives source to destination packets per second count in transaction |
| drate | numeric | Gives destination to source packets per second count in transaction |
| attack | categorical | Gives whether the network experiences normal traffic or attack traffic |
| category | categorical | Gives traffic category |
| subcategory | categorical | Gives traffic subcategory |

**Class level details:**

This dataset contains 72 million observations with 14 variables. The columns are mentioned above table.

**Block diagram:**



**Data sampling:**

Data sampling is the method of extracting sample from the entire dataset.

Types of data sampling are probability random sampling and non-probability random sampling.

**Probability random sampling**:

In probability random sampling sample is selected in a random way in which each member of the population has a non-zero probability of getting selected in the sample.

Types of probability random sampling are random sampling, systematic sampling, stratified sampling and cluster sampling.

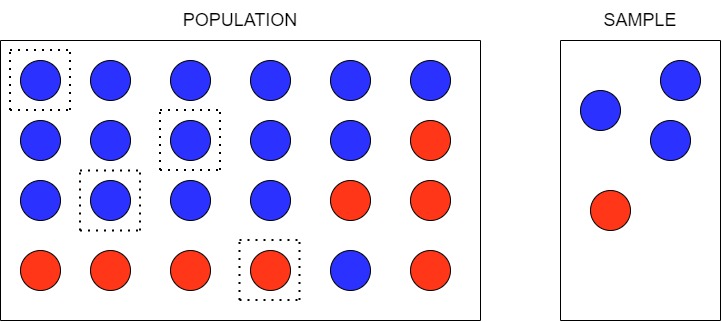
**Non-probability random sampling**:

In non-probability random sampling sample is selected in a non-random way.

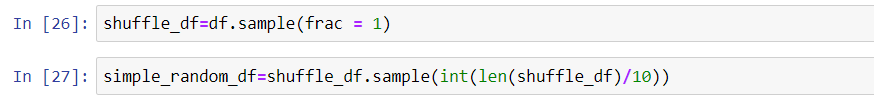
Types of non-probability random sampling are convenience sampling, judgement sampling, quota sampling and snowball sampling.

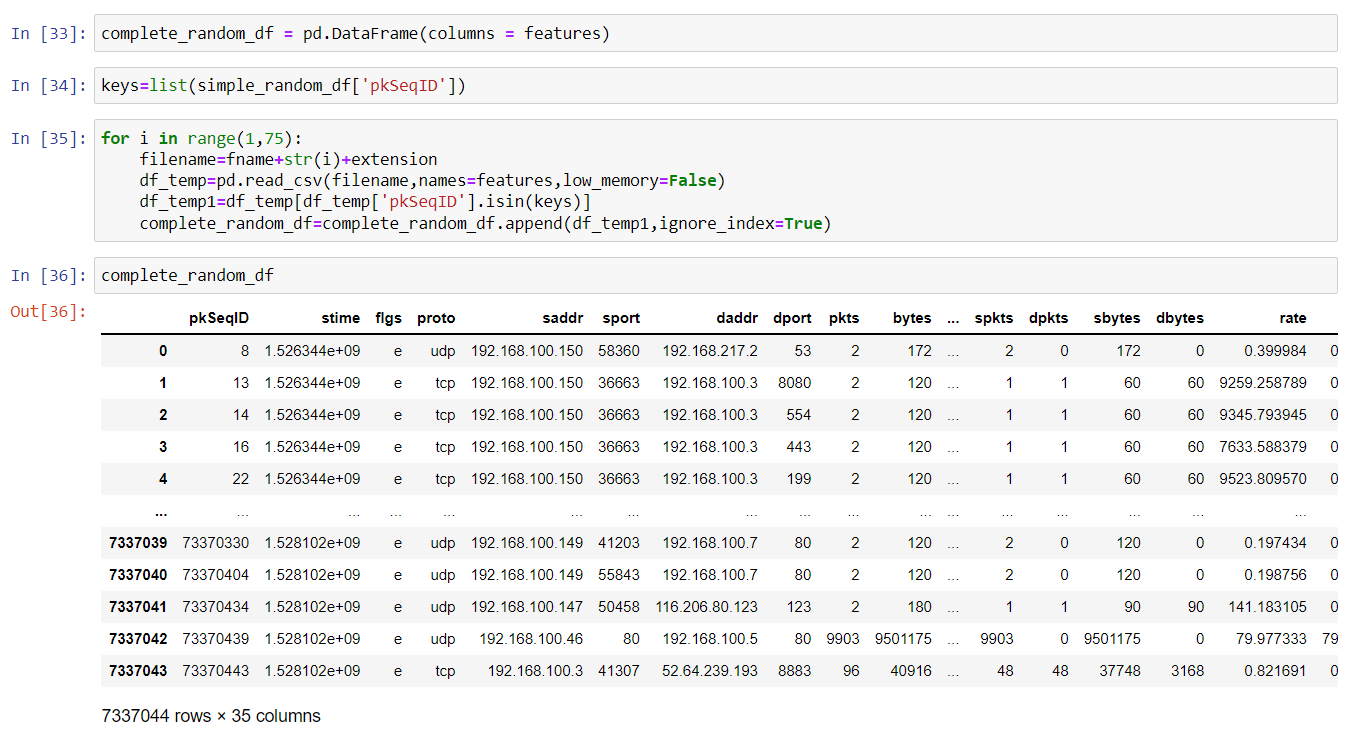
**Simple random sampling**:

In simple random sampling, each member in the population has the equal probability of getting selected.



Code:

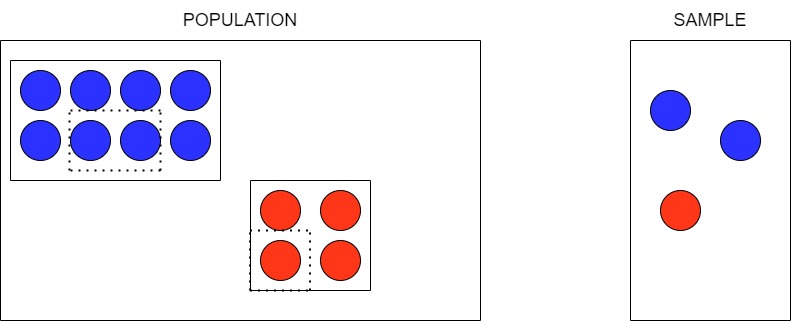




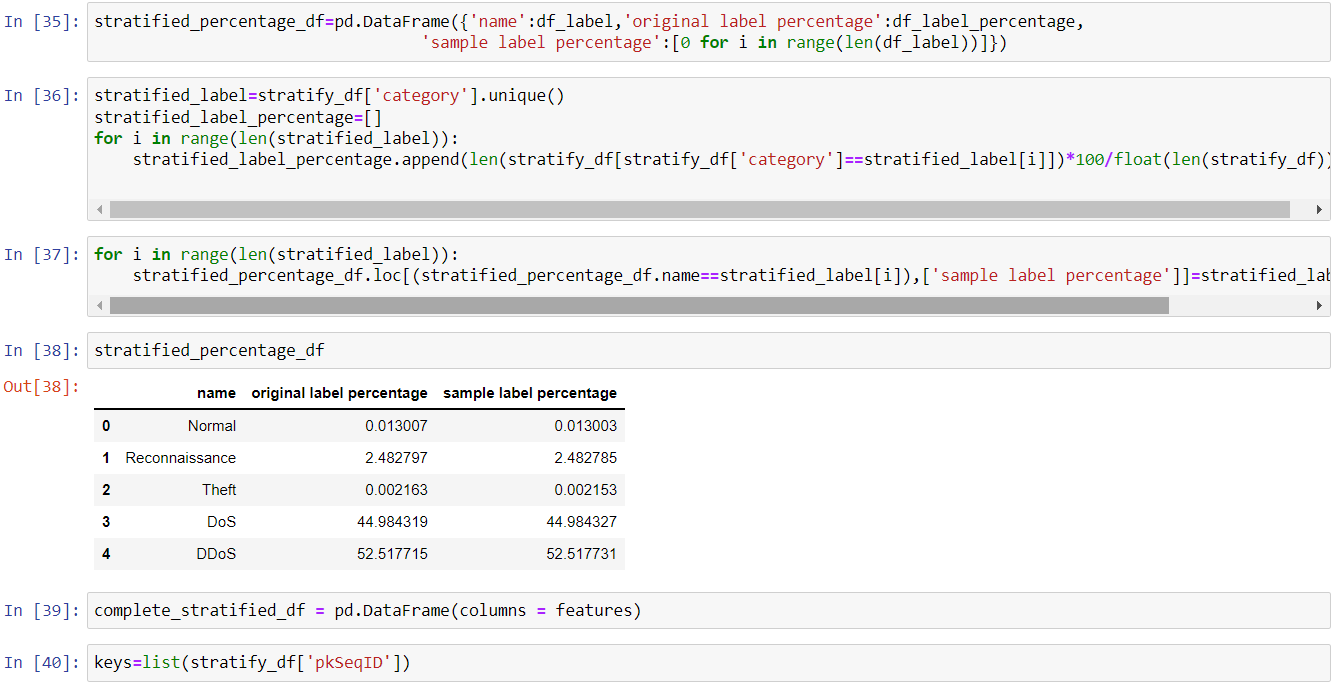
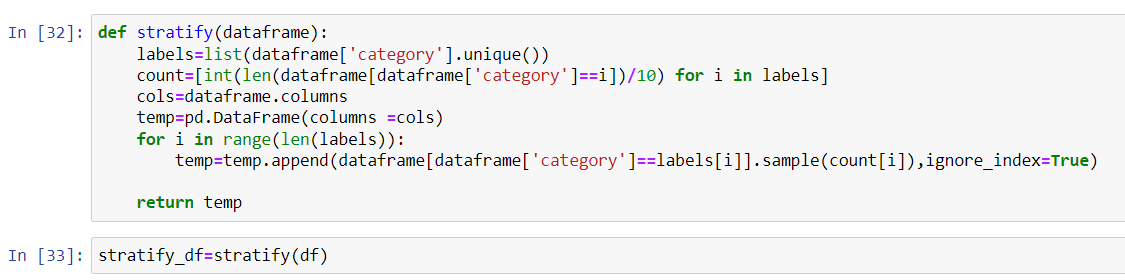


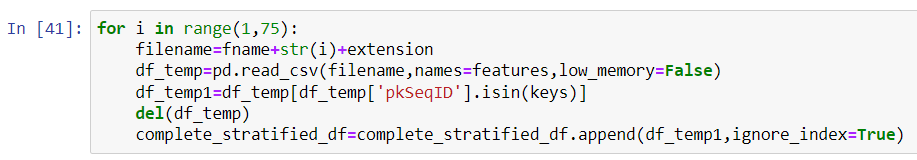
**Stratified random sampling:**

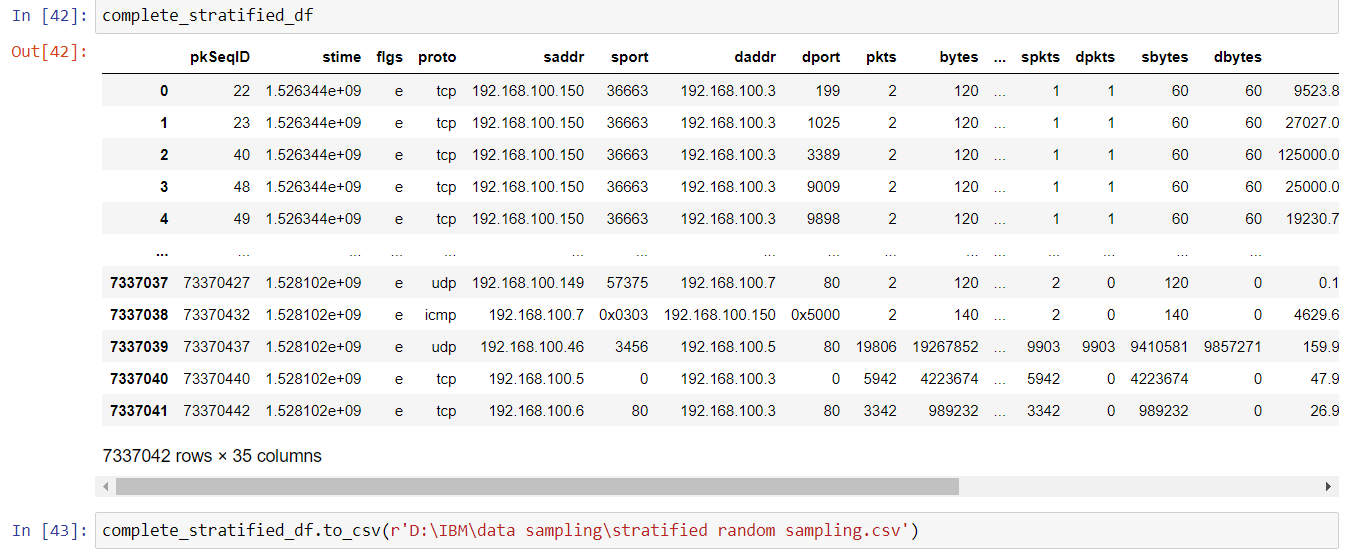
In stratified random sampling, dataset is divided into different strata and random sampling is used to select strata from each sample.



Code:

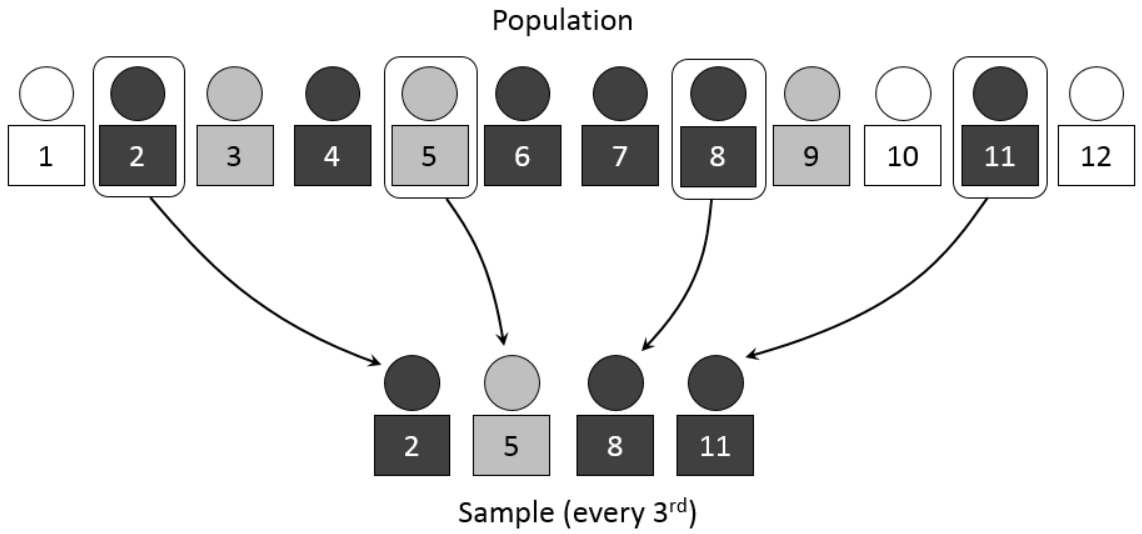




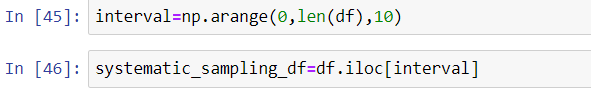


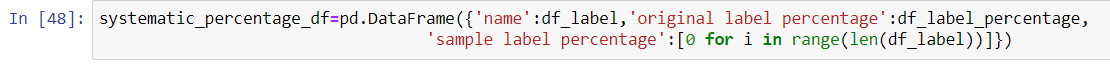
**Systematic Sampling**

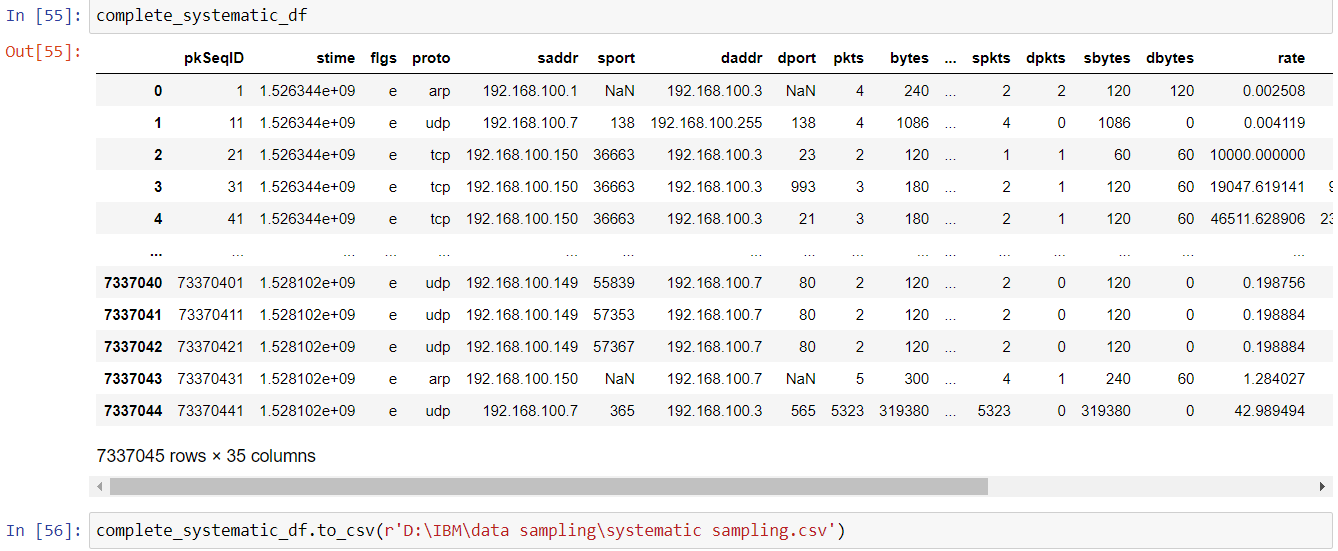
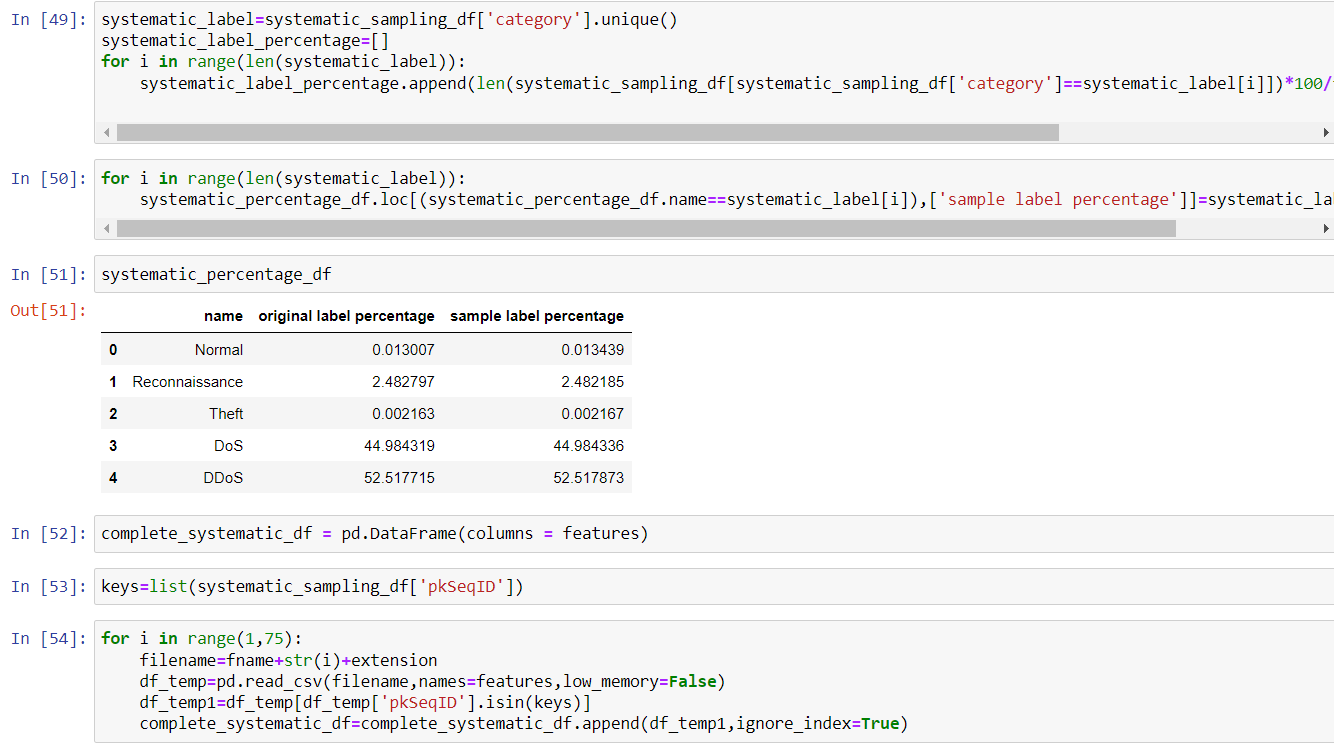
Systematic sampling is a type of probability sampling method in which sample members from a larger population are selected according to a random starting point but with a fixed, periodic interval.

****

Code:

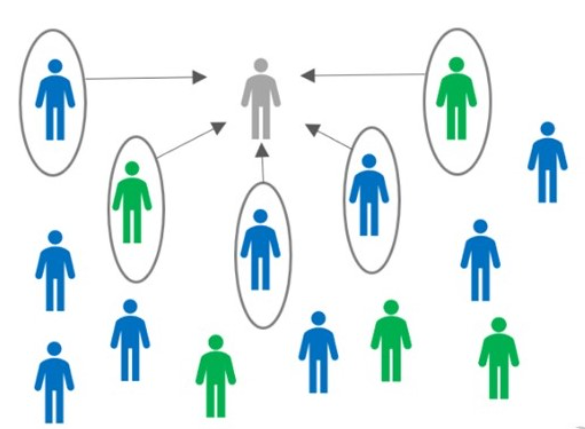
****

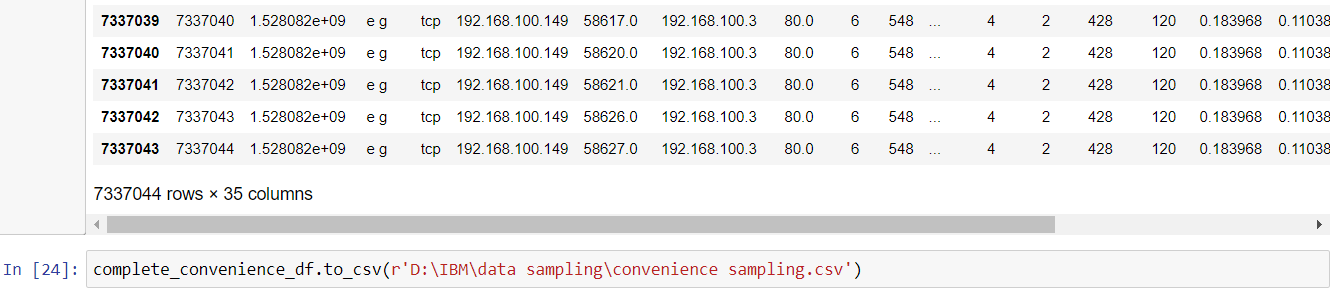
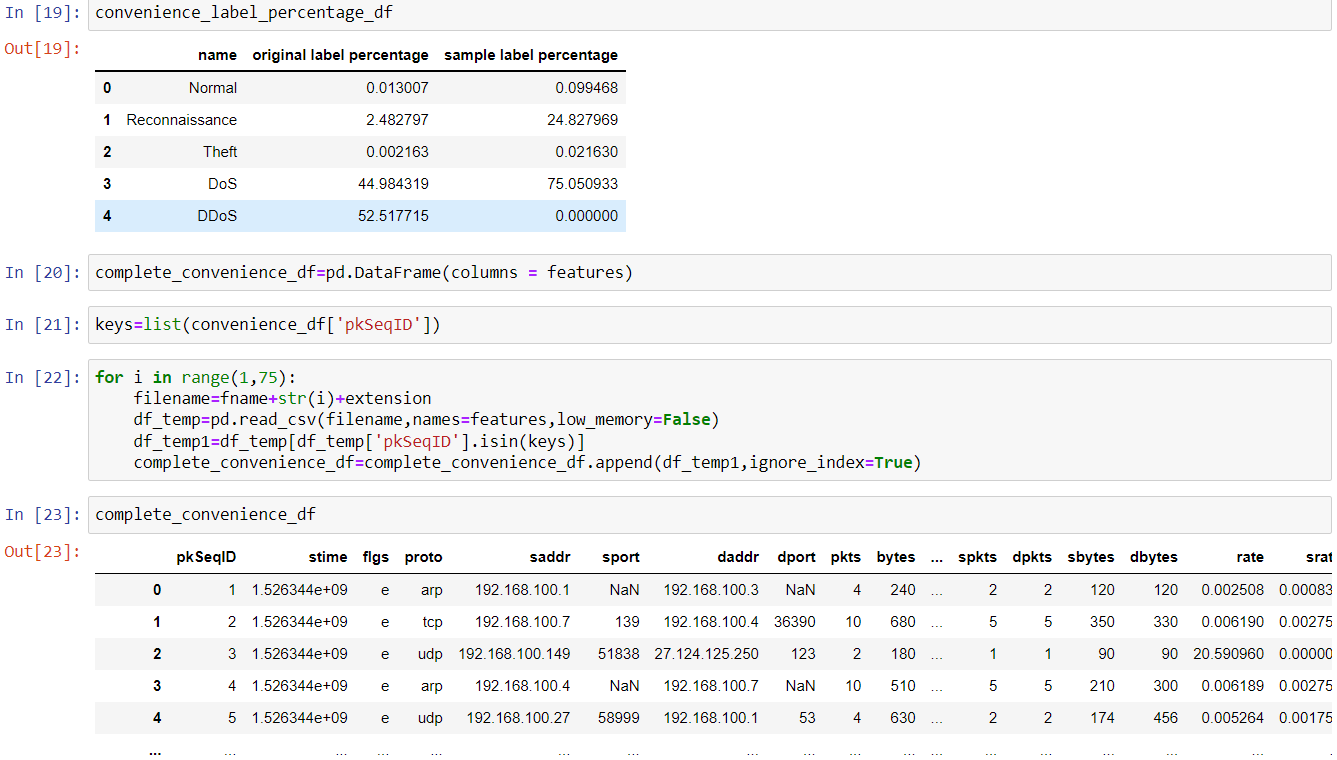
****

****

**Convenience Sampling**

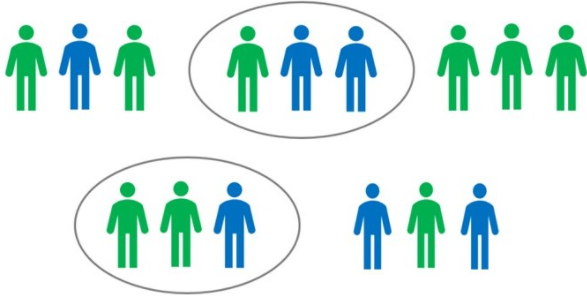
Convenience sampling is a type of non-probability sampling that involves the sample being drawn from that part of the population that is close to hand.



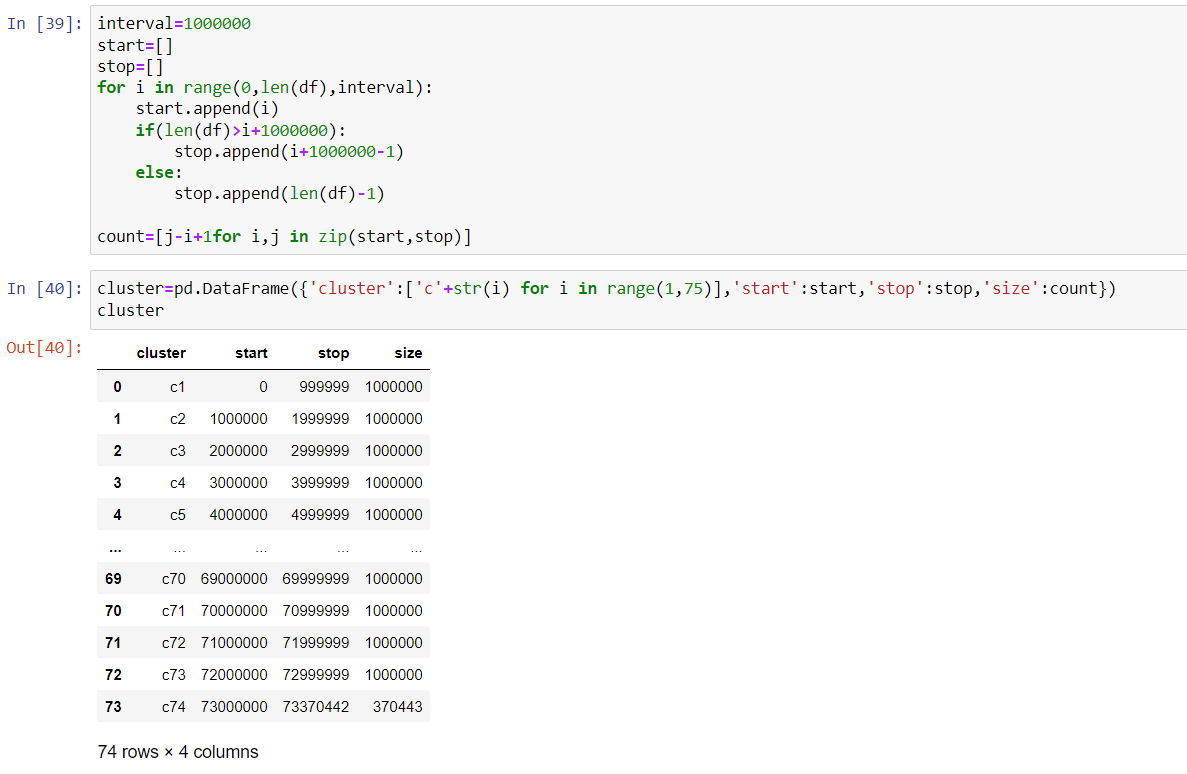


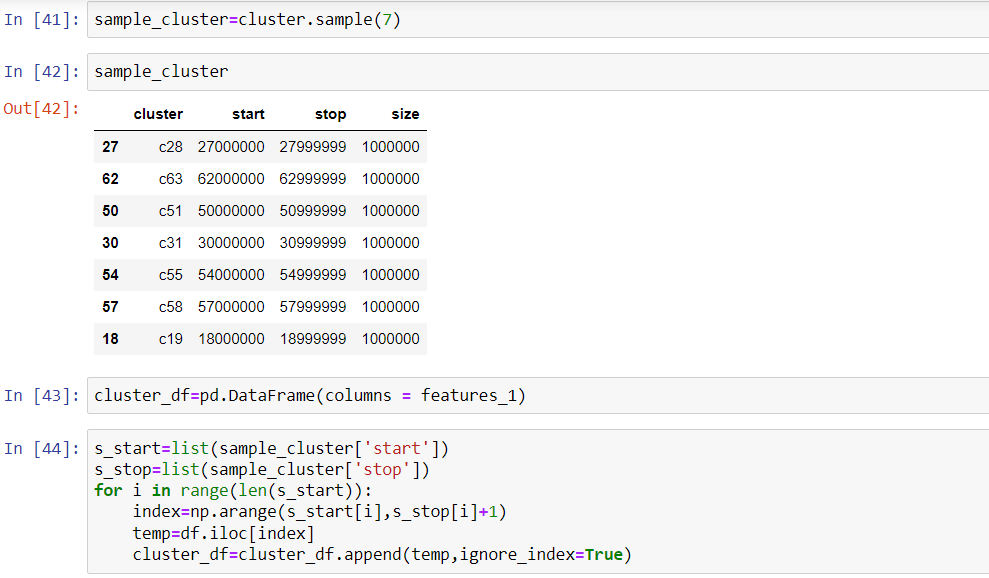
**Cluster Sampling**

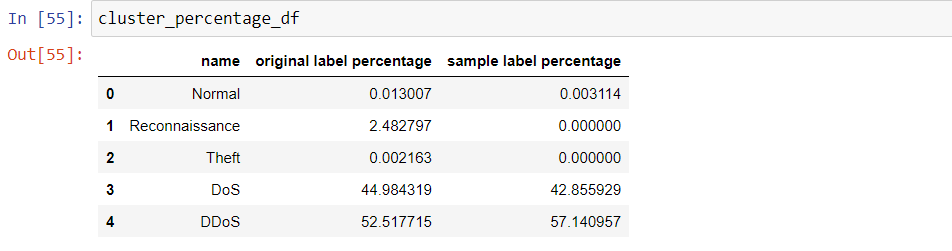
Cluster sampling is a probability sampling technique where the population is divided into multiple groups (clusters) and random groups are selected with a simple random or systematic random sampling technique for data collection and data analysis.



Code:







****

**Additional:**

**Explore:** **Check if any library is existing for sampling in python**

**csample is** a library which is used for sampling in python

It can be installed using the command:

**csample** provides pseudo-random sampling methods applicable when the size of population is unknown :

* We can use hash based sampling to fix sampling rate (data unsized)
* We can use reservoir sampling to fix sample size

csample provides two sampling functions: pip install csample

**sample\_tuple()**

This function expects tuples instead of strings as a content of iterable.

We have imported this function with the help of some dataset:



**sample\_line()**

 This function accepts iterable type containing strings.

We have imported this function with the help of some dataset:

