howto with Data in L-ift

Data team

# Setting up the system

In Lift we normally use R for the analysis of the data and for the data sampling process, this however does not limit us to just the language, we also use Python for process automation as well as excel for simple analysis and visualization. We also use other technologies such as tableau and powerBI depending a given project. This chapter simply gives a tutorial on how to set up the machine for data analysis which will include:

* *Installing R and R studio* To get R to work on a local machine you need to install base R and then install R studio. Base R: [Download BaseR](https://www.r-project.org/) R- studio: [Download R studio](https://www.rstudio.com/products/rstudio/download/#download)
* *Setting up github* Github is our go to back up system for our codes, we basically backup our work atleast once in a day but recommend doing so every hour. Lift has a github page where we have all our past projects and where each repository is given access to the right people who are working on that project. The link to the github profile is Github: [Access Github repo](https://github.com/liftransform). For more information on setting up your github page visit : Github: [Access Github documentation](https://docs.github.com/en)
* *downloading python and anaconda* Anaconda is the best python IDE and we use it for python, and to install it on your machine is relatively easy - you just need to visit and download the anaconda from its website and you get access to most data related python libraries and features. [Download Anaconda](https://docs.anaconda.com/anaconda/install/windows/)
* *Excel* Excel comes with the Microsoft office and therefore its installation is quite trivial. We use excel for simple analysis especially where the client wants to play around with the data and they arent technical, we create a pivot table based report.

# The L-ift theme & Report

Lift has a theme that sets the color and other aesthetics for its reports, visuals and graphs. To set this theme, the following code is used and can be copy pasted on to any project and then used. You will need to just clone the repository and then you can start your project analysis. Most of the work has already been done such as setting up the theme for the graphs, adding the logo to the reports etc. This is a work in progress and shall keep on being updated as we continue.

you can find the link to this report on this link: [Data Report template](https://github.com/liftransform/Data-report-template)

# Understanding data Used in L-ift

## Diary data

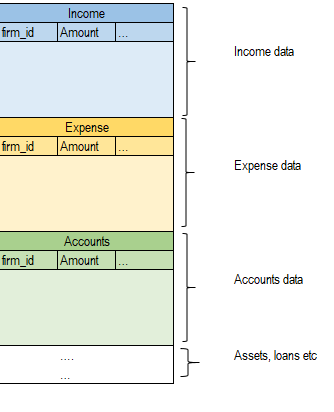
Diaries data is data that is collected over time similar to longitudanal data but with some aspect of self reporting by the respondent.

## data consolidation

Data comes from the finbit in seven main categories that can vary project to project:

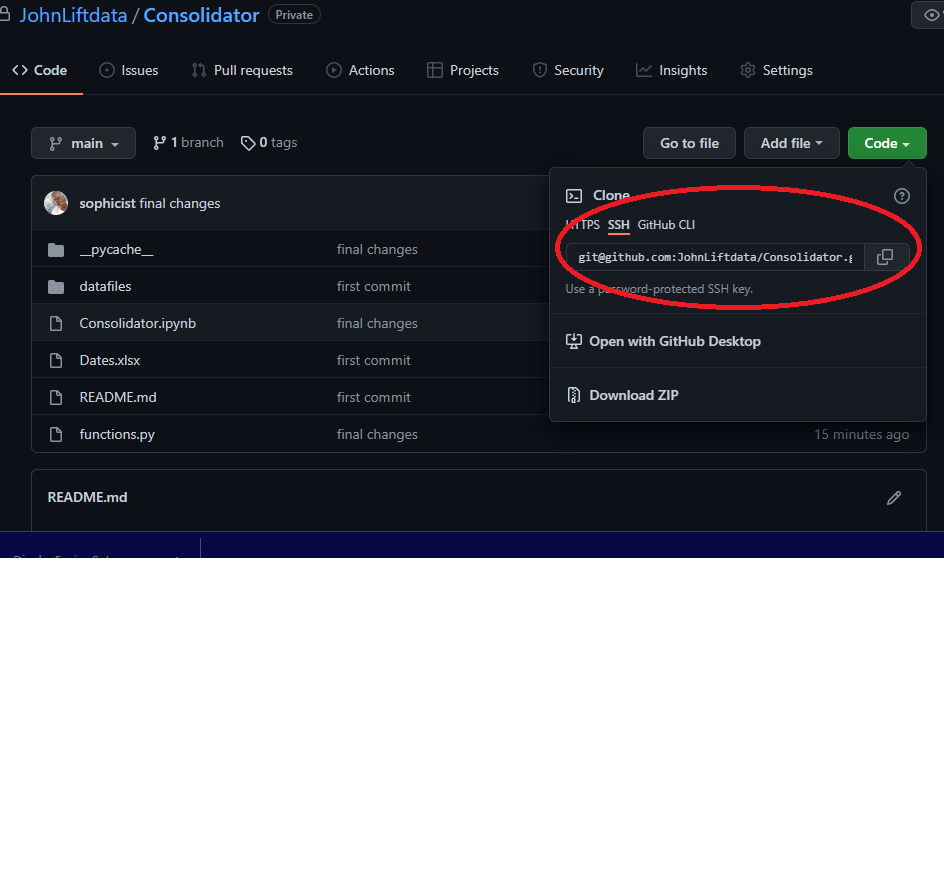
* Account report
* Asset report
* Employee report
* Expense report
* Income report
* Loan given report
* Loan taken report

To use these , we found an effective method is data consolidation where the data is added on top of each other after renaming the columns.

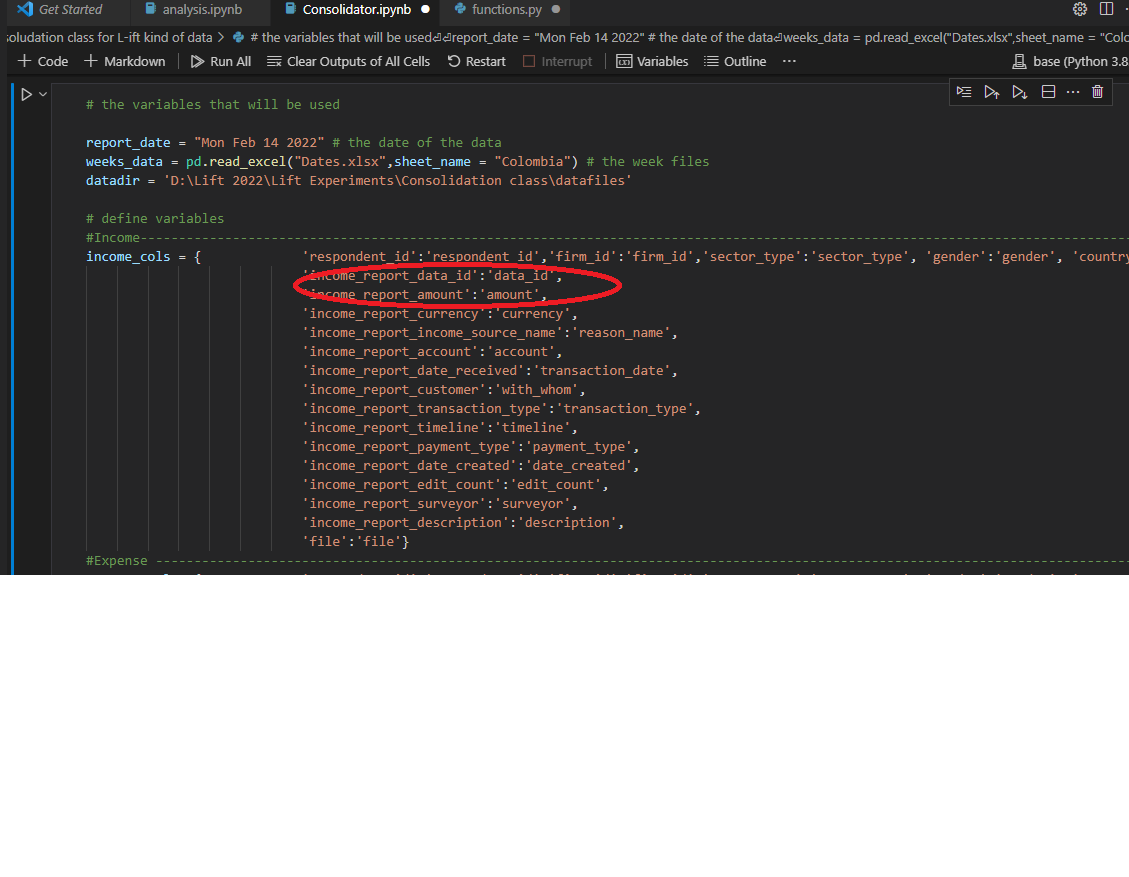


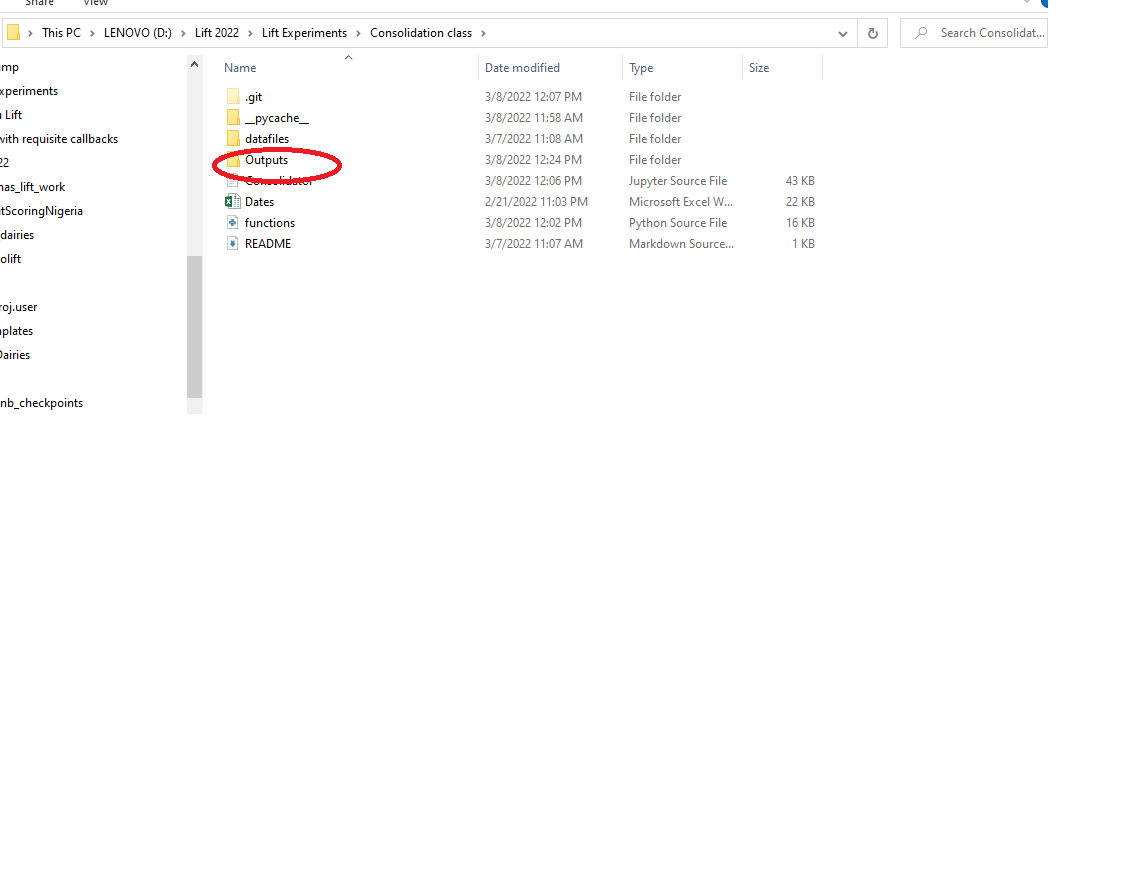
I have compiled the code for doing this on Github that uses python code. To use it you just need the following stepts:

1. clone the github repo https://github.com/liftransform/Consolidator.git after getting access rights from the repo admin.



1. locate the ‘define variables’ tab on the Consolidator file and change these as per the specific project. You need to makes sure you add all variable names that will be needed for the final data together with their renamed version. If a variable is needed but the name does not need renaming, just add it in the dictionary with the key-value as the current name.



1. run all the other cells and the data will be consolidated and added to the “outputs folder” 
2. Use the data for further analysis.

## Other types of data

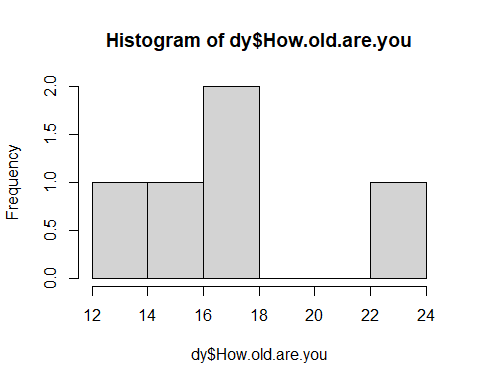
### Survey data

One of the most common data that we have in lift is survey data. We use the lifts own finbit app for collecting this data but sometimes we also use the Dooblo ( Survey2go) application for these surveys. Understanding survey data is important for L-ift and further understanding how to analyze survey data is likewise essential. There are 3 main types of questions on surveys that you may expect.

1. simple column answers

These can take the form of “yes” and “no”, continous data etc. They are the simplest to work with and can be analyzed as follows

library(tidyverse)  
library(knitr)  
dy = data.frame("How old are you" = c(23,18,17,16,12))  
hist(dy$How.old.are.you)



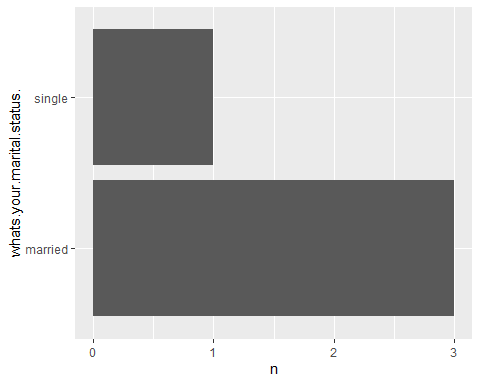
head(dy) %>% kable()

| How.old.are.you |
| --- |
| 23 |
| 18 |
| 17 |
| 16 |
| 12 |

dy = data.frame("whats your marital status?" = c("married","married","single","married"))  
dy %>% head() %>% kable()

| whats.your.marital.status. |
| --- |
| married |
| married |
| single |
| married |

dy %>% group\_by( whats.your.marital.status.) %>% tally() %>%   
 ggplot(aes(x = whats.your.marital.status.,y = n))+geom\_bar(stat = "identity")+coord\_flip()



1. Multiple answer questions

We sometimes ask questions that the respondent can have more than one answer to

* suppose we asked “which of the following sources of energy have you used in the last 1 week?”

To analyze this data we need to melt it - melting data means transforming it from wide to long format and then analysis for this type of data becomes easier

library(reshape2)  
dy = data.frame("gender" = c("M","M","F","M","F"),  
 "Q\_1\_candle" = c(1,1,1,0,0),  
 "Q\_1\_kerosene" = c(0,1,0,0,1),  
 "Q\_1\_electricity" = c(1,1,1,1,0),  
 "Q\_1\_solar" = c(1,0,0,0,1))  
dy %>% head() %>% kable()

| gender | Q\_1\_candle | Q\_1\_kerosene | Q\_1\_electricity | Q\_1\_solar |
| --- | --- | --- | --- | --- |
| M | 1 | 0 | 1 | 1 |
| M | 1 | 1 | 1 | 0 |
| F | 1 | 0 | 1 | 0 |
| M | 0 | 0 | 1 | 0 |
| F | 0 | 1 | 0 | 1 |

# analysis  
dy\_melt = dy %>% melt(id\_vars = "gender")  
dy\_melt %>% head() %>% kable()

| gender | variable | value |
| --- | --- | --- |
| M | Q\_1\_candle | 1 |
| M | Q\_1\_candle | 1 |
| F | Q\_1\_candle | 1 |
| M | Q\_1\_candle | 0 |
| F | Q\_1\_candle | 0 |
| M | Q\_1\_kerosene | 0 |

# working with the data  
library(janitor)  
  
dy\_melt %>%mutate(value = factor(value)) %>% group\_by(variable,value) %>% tally() %>%   
 dcast(variable~value,value.var = "n") %>%   
 adorn\_percentages("row") %>% adorn\_totals("col") %>%adorn\_pct\_formatting() %>%   
 adorn\_title(placement ="top",col\_name = "Percentages",row\_name = "energy types" ) %>% kable()

|  | Percentages |  |  |
| --- | --- | --- | --- |
| energy types | 0 | 1 | Total |
| Q\_1\_candle | 40.0% | 60.0% | 100.0% |
| Q\_1\_kerosene | 60.0% | 40.0% | 100.0% |
| Q\_1\_electricity | 20.0% | 80.0% | 100.0% |
| Q\_1\_solar | 60.0% | 40.0% | 100.0% |

1. Loop questions

Loop questions are similar to the questions above but they are iterated over some other item like month, group etc. they are a followup to apreviously asked question.

q1. which group/s do you belong to: - group1 - group2 - group3 q2. whats the name of the group/s mentioned above

* group1 \_\_\_
* group2 \_\_\_
* group3 \_\_\_ q3. how much did you deposit in the group mentioned above
* January
  + group1 \_\_\_
  + group2 \_\_\_
  + group3 \_\_\_
* February
  + group1 \_\_\_
  + group2 \_\_\_
  + group3 \_\_\_ …etc

working on this type of data requires consideration and can only be done case by case hence no universal method that can be suggested.

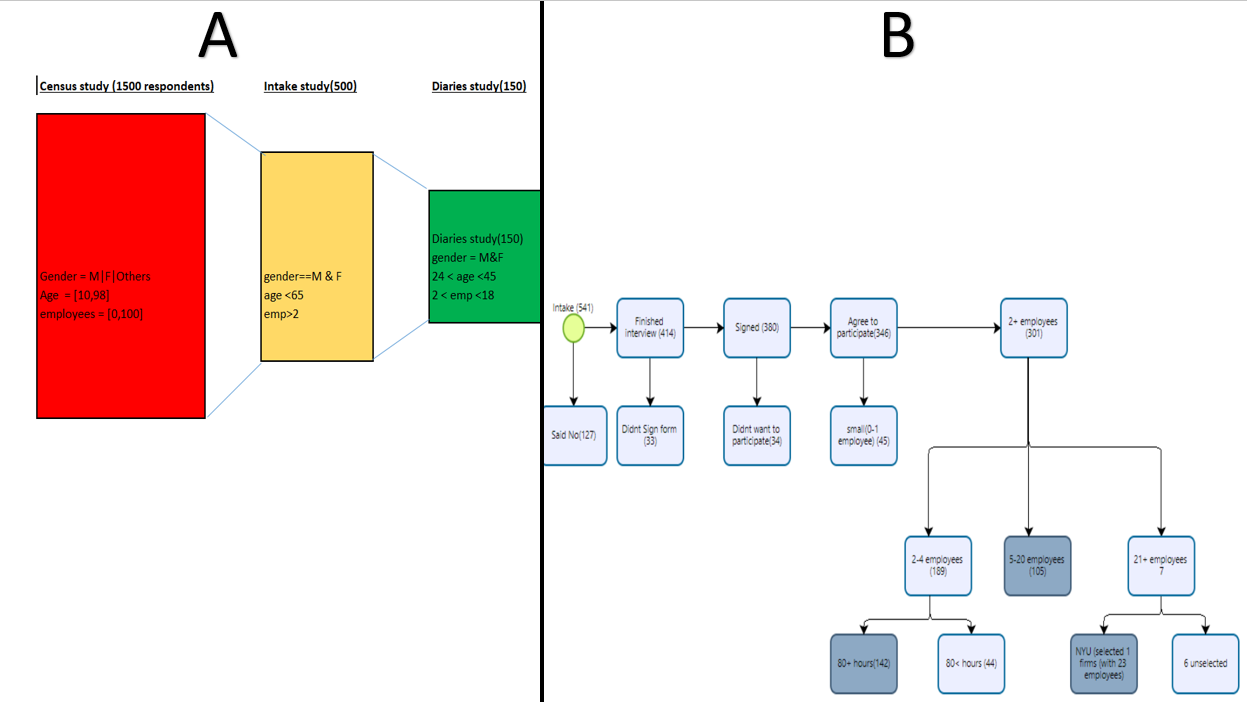
# Working with Diaries data

introduce - mention baseline, endline and midline

* time component
* inflows and outflows
* Trend analysis
* models suggested

## Data sampling

Data sampling is done in L-IFT to select a desired sample from a larger research done. We normally start with the Census study where we do an overall study, and select all respondents that may fit a particular criteria we are interested in. From this we select a sample that is more specific, using a criteria that is shared by the client. The sampling is like a funneling where we select a more specific sample over time until we decide on the final sample that now is studied in the diaries study.



The diagram above shows the sampling process diagrammatically. Part A shows how the sample size decreases over time as we move from the census, Intake and finally the diary sample. Sometimes between the Intake and the Diary we have the baselien study but the logic remains the same. We start with a large overall study, we then use a criteria - usually key questions that were asked during the study, we then get more specific, filtering the data until we have the final sample.

Part B shows the internal logic while conducting the sampling. The Intake study starts off with a sample size of 541, we remove those that said “No” - dint want to participate further in the study. With now a sample size of 414,we look at those that didnt sign the consent form and we remain with 380. After removing those that didnt waht to participate and also those that we considered small firms (had less than 2 employees), we remain with 301 respondents. Between these we select those that are highlighted in black as follows:

* Those with 2-4 employees we selected only those whose employees worked over 80 hours a week.
* for those that had 5-20 employees we selected all of them
* for respondents with over 21 employees, we only selected one respondent that the client requested.

## Key considerations

* Sampling is often an optimization problem where some of the criteria are met while others are not, we try to satisfy as many criteria as we can.
* we have two types of criteria - strict and preferential. For strict they have to be met always, but for preferential we only have to try our level best to meet them. In the diagram above (B), signing a consent form, agreeing to participate etc was a strict criteria, whereas sampling firms with 2-4 where they had 80 hours was a preferential one. These can be relaxed when more sample is required.
* when you select a sample that meet a criteria and say you need a small group of that sample, we normally use random sampling withing that group.
* After sampling we create a sample list ( this includes and analysis of the metrics), we also include an unsampled dataset where we shall get additional sample upon request by the client or the country/ field managers.

more examples on this can be found here:

[SFD FIJI SAMPLE](https://github.com/liftransform/SmallFirmDiaries/tree/main/SFDsmall_firm_diaries/FIJI)

# Data Management

## data quality checks and cleaning

\*\* give code examples for each to ensure simplicity

1. Indicators: Each research has criteria that you use to proceed to the next stage, eg in the census, the NYU team must have had criteria for selecting the baseline sample, these variables or indicators are important to monitor daily and see their spread(eg willingness to participate, membership to something, size of the firm and so on), eg if it gender, and you find a particular surveyor has only women, this may indicate bias or less random selection. It may inform the next step where you ask the field team to focus on a particular indicator for balance. For this, we had a spreadsheet with pivots for each indicator and each day we monitor the spread of the main indicators and act accordingly.
2. Outliers: For continuous data ( numeric), we look for outliers ( these are figures that are too far apart from what the rest of the respondents are giving). They are not necessarily wrong, they are just worth taking a look into, if right they can form good case studies or exceptions to rule forms of studies but if wrong then they’ll be rectified early on. Mathematically an outlier can be defined as a point that is 2 standard deviations away from the mean. You can draw a boxplot and this will show them as isolated points, or you can calculate and flag the amounts.
3. Blanks: Blanks in the data are of special importance as they can mean that the question was asked to some of the respondents ( check if the way the question is formulated is correct), they can mean that the question was optional ( again see if it’s well coded), they can mean missing data, etc so they are worth investigation.
4. Completeness: For the other types of data its important to have criteria for what is acceptable for each column, based on theory or experience and to flag anything outside this scope eg what currencies are acceptable for a country, are some countries of origin possible based on the study area etc, are some amounts possible based on what we know about the respondents etc - this depends on some prior knowledge about the respondents and can be built over time. For example, we can have amounts that are not outliers for the entire data but are too large or small for some respondents given what we have been receiving from them, this can be identified after a few records have been received.
5. Balance & consistency: For diaries data, we look at the balance between the inflows ( income, loans received, savings withdrawn) and the outflows( loans given, expenses and savings added) and investigate firms that have imbalances, if, on the negative, they may be reporting too many expenses and fewer incomes may be due to fears that the data can be seen by the government and they get taxed for example etc. In excel creating pivots on this helps zero in on these cases very easily after the data is combined. For other types of data that are not diary-based like the baseline, check for the balance based on expectation eg for a respondent with this size of the firm, can they have x number of employees? etc
6. Duplicates: Sometimes the data is duplicated when the field team records a transaction twice or due to errors. You can identify some columns that are supposed to be unique eg if the same respondent has a similar transaction on the same day with the same amount, this should be investigated and flagged.
7. Open-ended question: there are some cases where the questions are open-ended eg the ones that say “Other specify”, these are hard to check for errors. If they are too many it means the question was not well designed hence the drop-down options are not capturing the respondent’s answers and detecting this early can help adjust and get better answers. These need to be checked one by one to check the consistency.
8. Contradiction: This is particularly important for loop or follow-up questions where someone reports something then something else on a follow-up question eg they state that they don’t have children then later report they have a family of size 7. These need to be also checked as they can indicate that the respondent didn’t understand some part of a question or something else. These also appear for different studies where the census someone says one thing and in the baseline, they say something else on the same question eg which sector the respondent operates in.

# Learning in the data team

## Skill required for the data team

The data team requires mostly skills in either R or python and then the ability to use Excel. Of course those with skills in R and python are at an advantage as some of the more complex automation may require skills in Python. R however can also be used but its more technically involved

## Excel

For excel mostly we shall need the ability to write simple functions as well as a mastery of pivot tables and pivot charts. This is because for the less technical clients who are interested in managing their own data we normally use Excel for the project and share frequent updates of the pivot tables with them.

## R

For R:

For most of these you can look at our older reports to get an idea of how most of these are done.

* simple data structures in R

there are several data structures in R but the most common are below

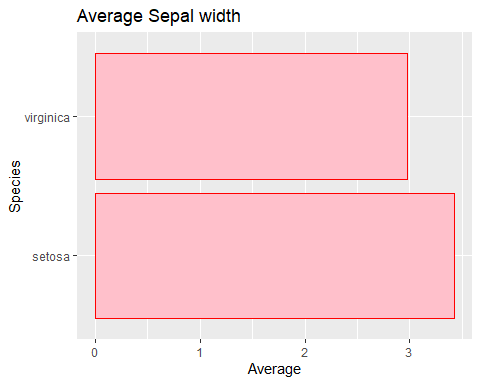
a <- 3 # using integers  
b <- 3.5 # using floats  
c <- c("names","areas","types") # vectors  
d <- list(1, "hello", c(2,3,1), FALSE, 3+4i, 6L) # lists  
e <- data.frame("names" = c("john","kamau","cosmas"),  
 "ages" = c(23,11,98))

* Learn DPLYR

Dplyr allows use of data piping in R, you can use the pipes to

1. select data
2. filter the data
3. group bys & summaries
4. visualise data
5. mutate the data

library(dplyr)  
library(ggplot2)  
iris %>%   
 select(Species,Sepal.Width) %>% # select 2 columns  
 filter(Species %in% c("setosa","virginica")) %>% # select 2 categories  
 group\_by(Species) %>% # group by the categorical column  
 summarise(Average = mean(`Sepal.Width`)) %>%   
 ggplot(aes(x = Species,y = Average))+geom\_bar(stat = "identity",color = "red",fill = "pink")+coord\_flip()+  
 labs(title = "Average Sepal width")



* GGPLOT

I have included a simple plot above, more examples will be added below.

* Janitor or equivalent table manipulation library

Janitor helps generate good tabular analytics. we use it to summarise data in a tabular format.

library(janitor)  
library(knitr)  
  
iris %>%   
 group\_by(Species) %>%   
 tally() %>%   
 adorn\_percentages("col") %>%   
 adorn\_totals() %>%   
 adorn\_pct\_formatting() %>% kable()

| Species | n |
| --- | --- |
| setosa | 33.3% |
| versicolor | 33.3% |
| virginica | 33.3% |
| Total | 100.0% |

* functions and for loops in R ( for heavier workload manipulation)

Dplyr alone will do most of the work for you in R especially with regard to L-IFT. Rarely you may need to used functions or for loops and below are simple implementations.

# for loop implementation  
names = c("John","cosmas")  
for (name in names) {  
 print(paste("Hi there:",name))  
   
}

## [1] "Hi there: John"  
## [1] "Hi there: cosmas"

# function implementation  
  
greeter <- function(name){  
 return(paste("Hi there:",name))  
}  
  
  
Map(greeter,names)

## $John  
## [1] "Hi there: John"  
##   
## $cosmas  
## [1] "Hi there: cosmas"

## Python

example notebook is shared on google colab: ( add link)

* Data structures in Python ( lists, tuples ,dictionaries,pandas dataframes)
* control flows ( for loops )
* functions
* list and dictionary comprehensions
* visualization libraries

There are numerous visualization libraries in python. The most common are: *matplotlib* which is the basic plotting library but also the most powerful. Built upon this you find *seaborn* which has additional function and can create more complex visualizations. For interactive plot you can use *plotly* as well as *bokeh* . One of the ways to use R’s *ggplot* in python is the use of *plotnine* library. For more on this and a demo you can see here: [Visualizations in python](https://www.kaggle.com/code/stemical/topic-2-visual-data-analysis-in-python/edit)

* use of OS
* Machine learning ( optional)

# Handy tips and tricks in L-IFT data analysis

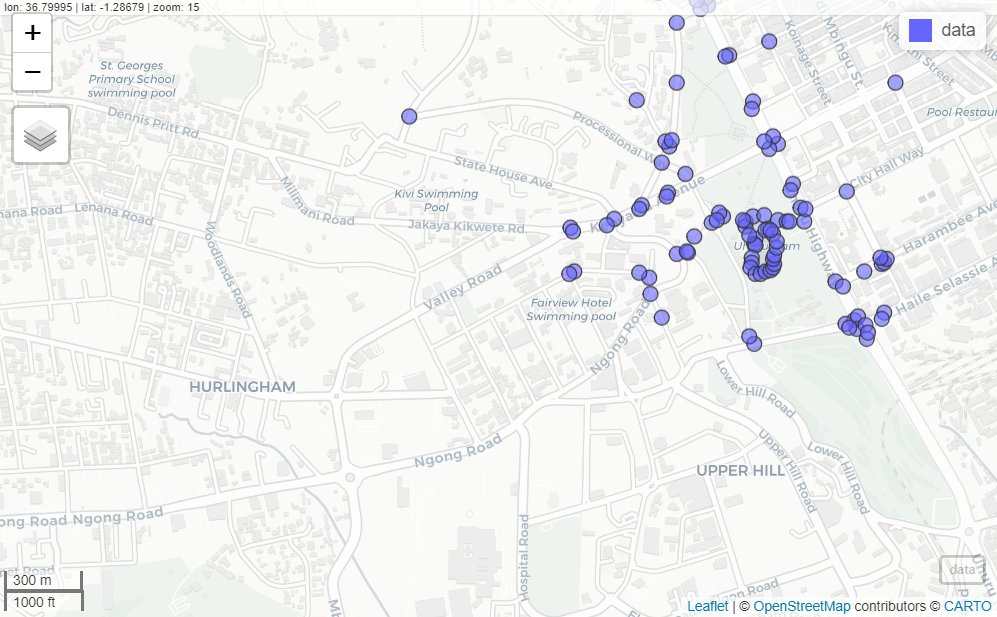
## 1. adding the theme to graphs

Adding the lift theme to your reports is easy, you simply download the L-IFT datareport template and the code is already there. for more reference , use the image below. You can customize the images as you see fit but the colors should remain, the others are optional though where possible just retain the theme as is.

## 2. Use of maps and geo data

In lift you may need to analyze geo - located data and this may require some general knowledge for handling such data. below is a simple library you can use for this although more advanced methods do exist, and can be used in its stead.

library(readr)  
library(tidyverse)  
library(tidyverse)  
library(mapview)  
library(sf)  
data <- read\_csv("data/nairobi\_locations.csv") %>% head(100)  
  
#mapview(data, xcol = "lon", ycol = "lat", crs = 4269, grid = FALSE)



## 3. automating report generation

We have had cases where we need automated report for each respondents, and this is what we called the automated report generation. There is a code that can be used to quickly generate these in R and all that is required is simply cloning the repository and then using it for the generation. These reports are in PDF but can be adapted to other types as required. [Automated reports](https://github.com/liftransform/Automated-reports-template)

## 4. backing up code on github

We Use github for data and files backups as well as data documentations. Our official github page is [“Github”](https://github.com/liftransform). Any new project will be created there by the lead and then a link will be shared as well as the access rights to this data. For more information on github use the following link to learn more. [Github documentation](https://docs.github.com/en)

## 5. data documentation

Why need data documentation: this will ensure all our data is understood by all our L-IFT staff and interpreted easily to the users. The documentation include what we have done, how it was done , when was done and why was done. The documentation method and format used in L-IFT include the README file- which contains the critical information about data file(s) organizations, codes used, software used including python ,R versions, Microsoft excel. Data file organization: Each study has its own folder with sub folders containing the different files. The files are for specific country. Example the small firm dairies folder with the sub folders are Ethiopia, Colombia , Kenya, Nigeria, Indonesia and Fiji. In each sub folder for instance Ethiopia have the files including the Census data files, Intake data files ,Dairy files, Surveys and the consolidated files.



## 6. data automation

One of the key areas where the data team can contribute is in data management. All the projects that have been done so far provide a rich source of data that can be mined and used for analysis and to inform management decisions. This can be done through data automation through generating data pipelines where data is ingested,cleaned and presented to an audience.

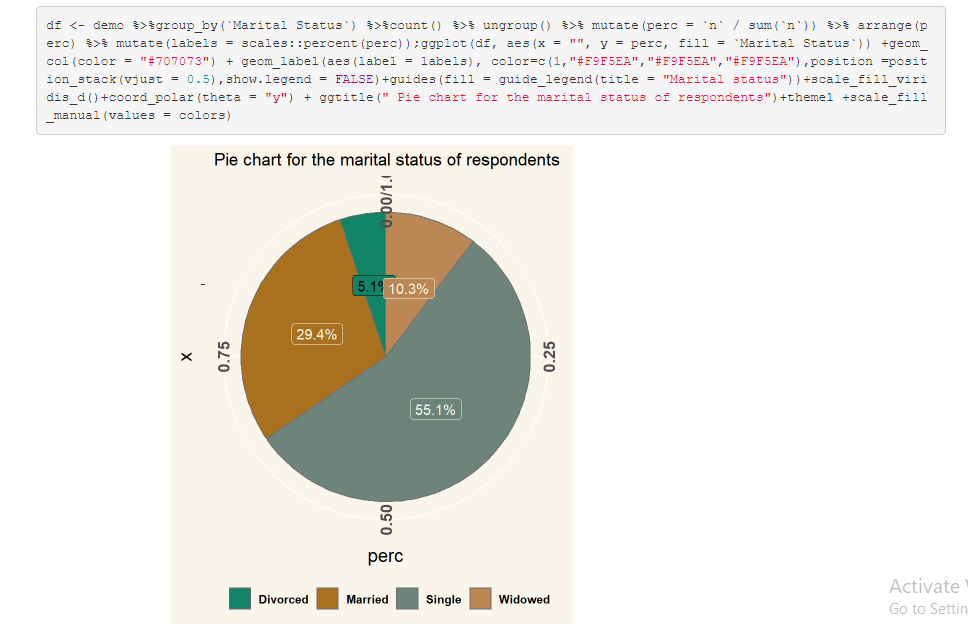
## 7. Daily experimentation

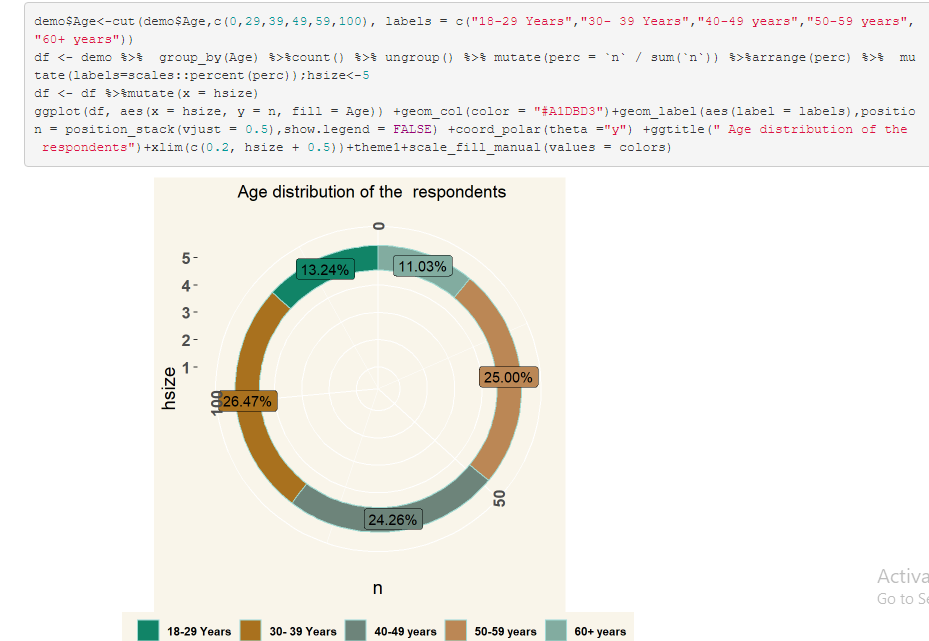
Every individual is allowed one hour for practicing data analysis and experiments using the L-IFT data or any other data for sharpening skills and being updated on the current tools for analysis in the industry of data.

## 8. Main types of plots used

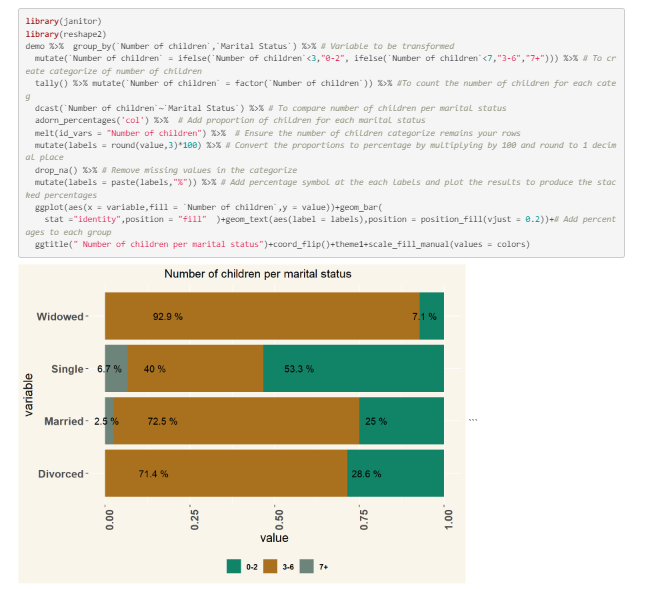
In L-IFT were use the pie charts, bar graphs and line graphs

### Pie chart

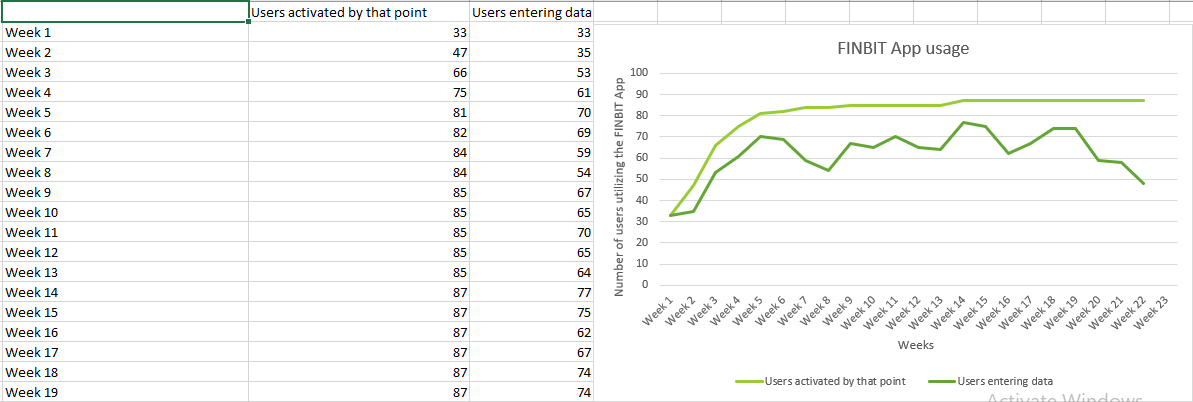
 ### Doughnut plot



### Stacked plot



### Line plot (Excel)



### Simple bar chart (Excel)

