Reid Minto

Unicef Work Plan:

**Overall Timeline:**

**2/1/20 – 10/15/20**

**Weeks 1 – 36**

**Week 1:**

**2/1/20 – 2/7/20**

**Subject matter understanding**

Subject matter understanding – gain knowledge / context of processes / procedures that need to be accomplished for project. May sound trivial – but coming from a different domain I find this to be the largest hurdle of the entire project, since I’m confident that once this hurdle is crossed the actual R coding / scripting routines etc will be smooth sailing.

**Weeks 2-8:**

**2/8/20 – 3/15/20**

**Write R draft scripts for Chapters 00 – 08**

After full understanding of the processes, write and deliver the R scripts for the first 9 chapters, using the SPSS syntax translated into R where appropriate, and rebuilding / customizing other parts where appropriate.

In order to just produce more than “the same routines but in R”, we want the code to be generalizable / standardized such that the code framework can fit any geographical region by switching R functional parameters, i.e. if we think of the analysis as a function that takes an input data set , puts it through the “function” of R code to produce the report output,

Input Data 🡪 R 🡪 Output, my plan is to build a framework in R that is able to capture the analysis routines that all regions have in common, then abstract out the differences into coded parameters that are functional inputs the various teams can choose to switch between.

In other words - the scripts to produce a general framework that gives teams the ability to choose / specify their specific region / group and have the code run accordingly. This is how we’ll achieve the flexibility for any group / region to use the R scripts for their specific purpose.

**Weeks 9 – 12:**

**3/16/20 – 4/15/20**

**Review the draft scripts for Chapters 00 – 08**

This will include ironing out any inefficiencies, making sure the data flows from raw input through the R script process and produces final output in appropriate report format. These 4 weeks will include improving code efficiency, commenting, thoroughly documenting the process of using the scripts, and making any adjustments suggested by the team.

My goal is to document the code and process such that anyone with extremely minimal experience can follow the documentation and use the R code such that they are able to input the data through the code and obtain the appropriate output.

**Weeks 13 – 20**

**4/16/20 – 6/15/20**

**Write R Draft Scripts for Ch09 – annex E of SFR**

Similar process to writing the first draft scripts, work with team to understand processes and translate the data input 🡪 analysis work into R scripts.

**Weeks 21 – 24**

**6/16/20 – 7/15/20**

**Review of draft scripts and finalization**

Similarly, revise scripts and improve upon inefficiencies, working with the project team to make sure drafts meet every need.

**Weeks 25 – 32**

**7/16/20 – 9/15/20**

**Write 4 Statistical Snapshots**

Produce visual representations of the results – graphs, figures, tables and boxes. Use R’s extensive functionality in these areas and work closely with team to produce specifically formatted / standardized representations. Use e.g. ggplot2 for graphics, pandoc / dataTable / etc. that the team can choose from R’s extensive report production capabilities.

**Weeks 33 – 37**

**9/16/20 – 10 / 14 / 20**

**Review of R scripts for finalization**

Review R scripts from statistical snapshots and project scripts in its entirety. During this period I will ensure that not only the deliverables are satisfactory and R scripts reviewed, but that I have thoroughly documented my entire process and code in such a way that someone with minimal or no R experience can follow along the documentation and use the scripts / routines / functions to produce the report outputs.

**Strategy and Additional Deliverables:**

Depending on time / approval of deliverables reviewed, an ambitious goal would be to produce a data product such as a Shiny web application where users could interact with graphical representations of the data, and even upload their input data, click a “Go” / action button, where the app would run through the scripts behind the scenes and produce the report that could then be downloaded onto their computer. Examples of shiny applications can be seen here:

<https://shiny.rstudio.com/gallery/>

A less ambitious goal that I expect to deliver *at a minimum* is improving the code efficiency for running through these routines for the users. So instead of users having to import their data and run through hundreds of lines of code to produce the results, or click a button to run through the code, while writing the draft R scripts I will identify common tasks that are repeated throughout the chapters / entirety of work, and wrap these tasks up into R functions which can be stored into a “Unicef R library”, which can be source / called by any user. This is one of the advantages of R over SAS / SPSS, aside from open source framework is the functional programming capability – so the R scripts will still be viewable / editable and fully owned by Unicef, but you’re able to provide users of these scripts R functions which hide a lot of the mess / clutter normally accompanying code scripts, and work with functions.

I’ll provide more concrete examples as we go – but for example if we find calculating infant mortality rates to be a common task that is done in a standard defined way, instead of running through an algebraic formula in a script every time we calculate infant mortality, we’ll have a set infant mortality function in R where the user just specifies their dataset variables necessary in the calculation and the function spits out the rate (where you can still edit / view what function is doing).

This idea can be rolled up even higher into a “calculate Ch01” with specified input data, so users aren’t spending their time running through scripts and back and forth between datasets and script processes etc.