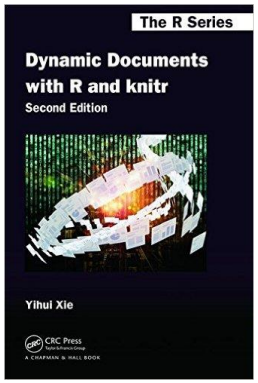
Open Source Software for Automating and Ensuring Quality in Official Statistics: An Example Using R and RStudio for Stratified Simple Random Sampling

Darryl V. Creel



Stratified Simple Random Sampling: Quality Control

Darryl V. Creel, RTI International

Wednesday, October 24, 2018

Quality, Reproducibility, Rigor, Standardization, and Transparency (QR^2ST). Quality is an important aspect Federal statistical information.

Quality

- : how good or bad something is
- : a characteristic or feature that someone or something has : something that can be noticed as a part of a person or thing
- : a high level of value or excellence

from www.merriam-webster.com

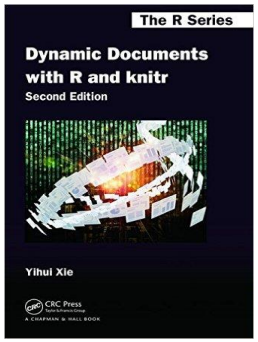
Planning for quality control should begin before the project starts: folder structure, naming conventions (folders, programs, functions, variables, data sets, etc.), automation, inputs/outputs, responsibilities, version control, testing, etc.

```
|----- ProjectName
|           |----- Computing
|           |----- Management
|           |----- Statistics
|           |           |----- A_Planning
|           |           |----- B_FrameDevelopment
|           |           |----- C_Sampling
|           |           |----- D_DataCollection
|           |           |----- E_DataProcessing
|           |           |           |----- EA_Weighting
|           |           |           |----- EB_Editing
|           |           |           |----- EC_Imputation
|           |           |----- F_Analysis
|           |           |----- G_Publication
|           |----- SubjectMatter
|           |----- SurveyMethodology
```

Programs in the sampling folder (C_Sampling).

```
|----- C_Sampling
|            |----- Ca_stratifiedSimpleRandomSampling_selection.Rmd
|            |----- Cb_stratifiedSimpleRandomSampling_qualityControl.Rmd
```

How can we increase quality (better), lower labor costs (cheaper), require less calendar time (faster), and document quality control processes?

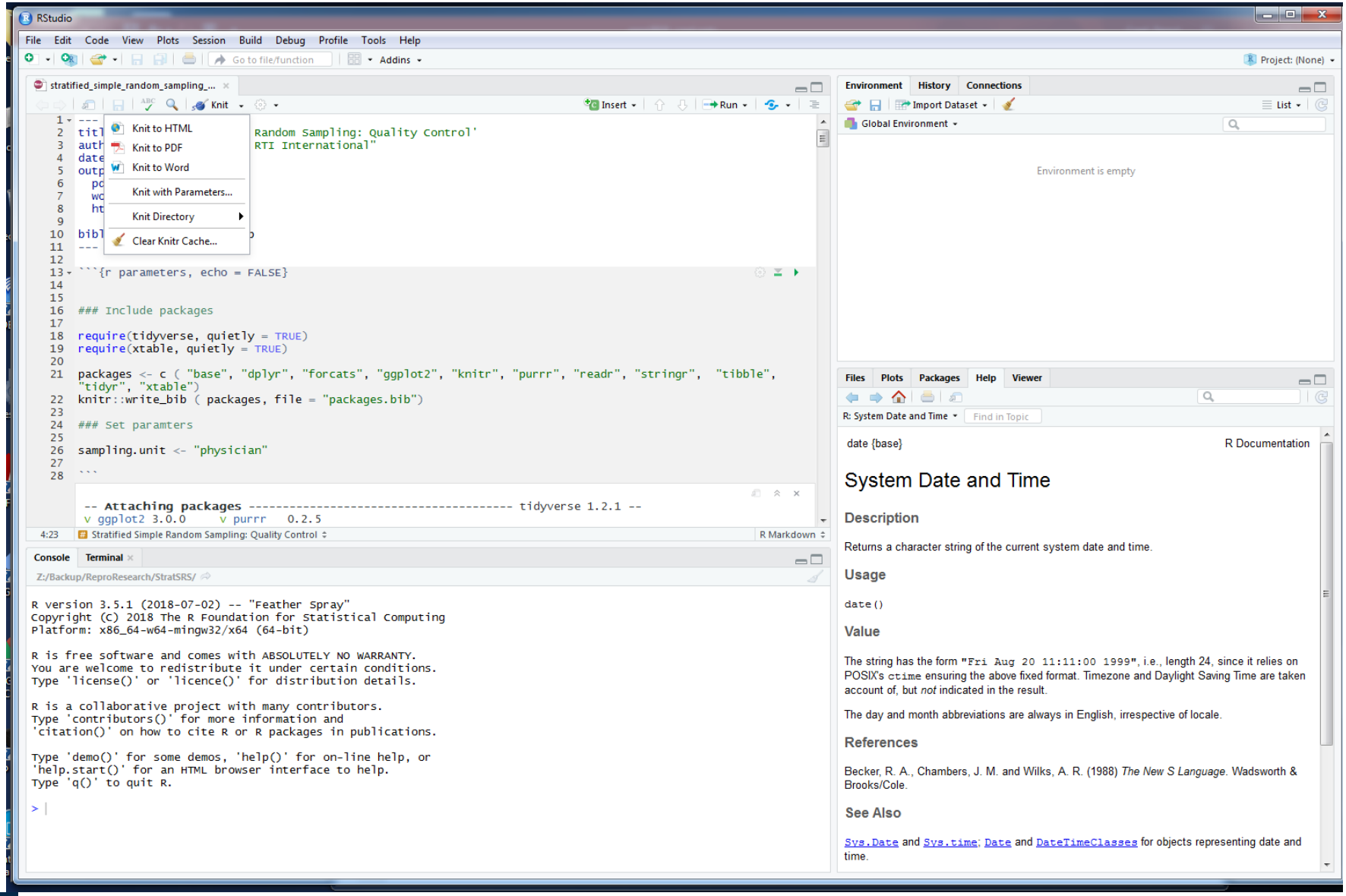


Stratified Simple Random Sampling: Quality Control

Darryl V. Creel, RTI International

Wednesday, October 24, 2018

RStudio, an integrated development environment for R.



Literate programming weaves the text and program code together.

In a sampling stratum, the sample size should equal the sum of the probabilities of selection. For the h^{th} sampling stratum, the sample size, n_{h}, should equal the sum of the probabilities of selection, p_{hi}. That is, in the h^{th} sampling stratum, the check to ensure that the probability of selection was calculated correctly is

$$
n_{h}=\sum_{i=1}^{N_h} p_{hi}.
$$

```{r probSelSum, type = "latex", results = "asis", echo = FALSE}

psSumPop <- sampSize %>% select(samplingStratum, sampleSize)

psSumSamp <- frame %>% select(samplingStratum, probabilityOfSelection) %>% group_by(samplingStratum) %>% summarize(psSum = sum(probabilityOfSelection))

psSumCheck <- full_join(psSumPop, psSumSamp, by = "samplingStratum") %>% mutate(diff = round(psSum - sampleSize))

psSumDiff <- psSumCheck %>% filter(diff != 0 | is.na(diff))

if (dim(psSumDiff)[1] == 0) {
  cat("Pass: All sampling strata have the sum of the probabilities of selection equal to the sample size.\n")
} else {
  cat("Fail: At least one samling stratum does not have the sum of the probabilities of selection equal to the sample size.\n")
  psSumDiff.xt <- xtable(psSumDiff)
  caption(psSumDiff.xt) <- "Sum Probabilites of Selection not Equal Sample Size"
  print(psSumDiff.xt, include.rownames = FALSE, caption.placement = "top", comment = FALSE)
}

```

Using the pipe, %>%, in R. It comes from the **magrittr** package by Stefan Milton.

```
sampSize <- read_csv("sampleSize_01.csv") %>%
mutate(posPop = sampleSize/populationCount, dw =
populationCount/sampleSize)
```

```
sampSize <- read_csv("sampleSize_01.csv")
sampSize$posPop <-
sampSize$sampleSize/sampSize$populationCount
sampSize$dw <-
sampSize$populationCount/sampSize$sampleSize
```

Using the pipe, %>%, in R. It comes from the **magrittr** package by Stefan Milton.

```
psSumSamp <- frame %>% select(samplingStratum, probabilityOfSelection) %>% group_by(samplingStratum) %>% summarize(psSum = sum(probabilityOfSelection))
```

```
psSumSamp2 <- tapply(X = frame$probabilityOfSelection, INDEX = frame$samplingStratum, FUN = sum)*
```

* Not quite right does not have information when samplingStratum is missing

RStudio, incorporating LaTeX

In a sampling stratum, the sample size should equal the sum of the probabilities of selection. For the h^{th} sampling stratum, the sample size, n_{h}, should equal the sum of the probabilities of selection, p_{hi}. That is, in the h^{th} sampling stratum, the check to ensure that the probability of selection was calculated correctly is

$$
n_{h}=\sum_{i=1}^{N_h} p_{hi}.
$$

RStudio, incorporating LaTeX.

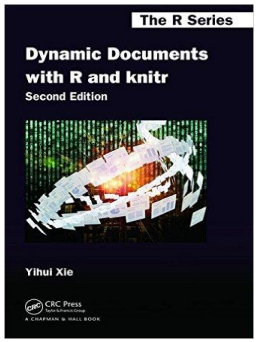
In a sampling stratum, the sample size should equal the sum of the sample indicators. For the h^{th} sampling stratum, the sample size, n_h, should equal the sum of the sample indicators, s_{hi}. That is, in the h^{th} sampling stratum, the check to ensure that the sample indicators were calculated correctly is

$$n_h = \sum_{i=1}^{N_h} s_{hi} \, .$$

Rstudio includes the table derived from the code.

```
# A tibble: 2 x 4
  samplingStratum sampleSize psSum  diff
            <int>      <int> <dbl> <dbl>
1               2         20  NA      NA
2              NA         NA   0.3    NA
```

How can we increase quality (better), lower labor costs (cheaper), require less calendar time (faster), and document quality control processes? RStudio and knitr.



Stratified Simple Random Sampling: Quality Control

Darryl V. Creel, RTI International

dcreel@rti.org

(301) 770-8229