Live Data from Our Central Server

TestProject-Blast

This is an **R Notebook** demonstrating the Live Data Analysis from **Central Server**.

$$ \\[0.1in] $$

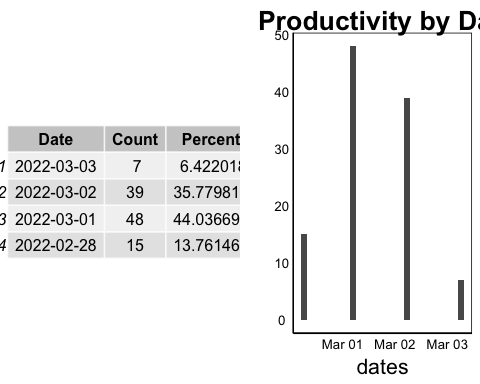
Here is the code-block to fetch data from the **Central Server** and counts total sample size!

loadpkg <- function(toLoad){  
 for(lib in toLoad){  
 if(! lib %in% installed.packages()[,1]) {  
 # install.packages(lib, repos='http://cran.rstudio.com/')  
 }  
 suppressMessages( library(lib, character.only=TRUE) )  
 }  
}  
  
  
loadpkg(c("dplyr", "ReporteRs", "readxl", "rio" , "openxlsx", "ruODK", "gridExtra", "stringr", "ggplot2"))  
  
source("http://bit.ly/theme\_pub") ## Theme set for ggplot2  
theme\_set(theme\_pub()) ## Theme set for ggplot2  
  
# Set project to "TestProject-Blast"   
project <- "https://central.pulsefe.org/v1/projects/17/forms/2225F110.svc"  
interviewer\_variable <- "int\_details\_interviewer\_name"  
  
  
# `ruODK` users only need default settings to their ODK Central:  
ru\_setup(url = "https://xxx.xxx.xxx",   
 un = "username",   
 pw = "password" ,   
 svc = project,  
 tz = "Asia/Calcutta")  
  
  
# File attachment download location  
loc <- fs::path("media")  
  
Sys.sleep(5) ## 5 Sec wait  
  
  
fq\_data <- ruODK::odata\_submission\_get(  
 table = fq\_svc$name[1],   
 local\_dir = loc,  
 wkt=TRUE)  
  
# Find total submissions  
n <- as.numeric(unlist(count(fq\_data)))

**Total Sample Size: 109**

Here is the code block that counts **date-wise productivity** and shows in a **Table** and **Pie Chart**

# Find productivity per day  
datesvector <- as.Date(fq\_data$todaydate, format="%Y/%m/%d")  
dates <- as.Date(unique(fq\_data$todaydate, format="%Y/%m/%d"))  
  
l <- length(dates)  
i <- 1  
datewisecount <- data.frame()  
  
while (i <= l) {  
 datewisecount\_i <- data.frame()  
 datei <- dates[i]  
 datecount <- sum(datesvector == datei)  
 datewisecount\_i <- data.frame(Date = datei, Count = datecount)  
 datewisecount <- rbind(datewisecount,datewisecount\_i)  
   
 rm(datewisecount\_i)  
 i <- i + 1  
}  
datewisecount$Percent <- datewisecount$Count\*100 / sum(datewisecount$Count) ## Add percent  
  
bar <- qplot(x= todaydate, data = fq\_data) ## Create bar chart  
bar <- bar + labs(colour = "Cylinders") + labs(title = "Productivity by Day") + labs(x = "dates")  
tbl <- tableGrob(datewisecount) ## Create Table  
  
grid.arrange(tbl, bar, nrow=1, newpage = FALSE) ## show table and bar in one row

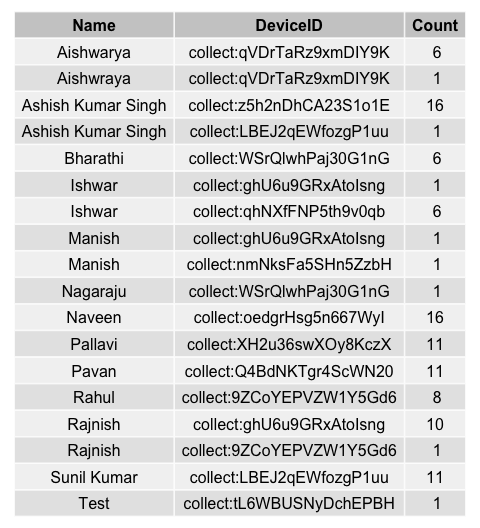


$$ \\[0.1in] $$

**Productivity Per Device**

The next code block finds **productivity per device**

# Find productivity per device ID  
devicevector <- (paste0(str\_to\_title(str\_trim(fq\_data$int\_details\_interviewer\_name)), "-" , fq\_data$deviceid))  
deviceunique <- unique(paste0(str\_to\_title(str\_trim(fq\_data$int\_details\_interviewer\_name)),"-",fq\_data$deviceid))  
  
l <- length(deviceunique)  
i <- 1  
devicewisecount <- data.frame()  
  
while (i <= l) {  
 devicewisecount\_i <- data.frame()  
 devicei <- deviceunique[i]  
 #interviewer <- filter(inter\_df, deviceid == devicei)  
 devicecount <- sum(devicevector == devicei)  
 device <- unlist(strsplit(devicei,"-"))  
 devicewisecount\_i <- data.frame(Name = device[1], DeviceID = device[2], Count = devicecount)  
 devicewisecount <- rbind(devicewisecount,devicewisecount\_i)  
 rm(devicewisecount\_i)  
 i <- i + 1  
}  
devicewisecount <- devicewisecount[order(devicewisecount$Name),]  
  
  
tbl\_device <- tableGrob(devicewisecount, rows=NULL) ## Create Table  
grid.arrange(tbl\_device, nrow=1, newpage = TRUE) ## show table and bar in one row



$$ \\[0.1in] $$

**Analysis**

source("/Users/debjitbiswas/shiny-projectstatus/crosstab.r") ## Load crosstab function from "http://pcwww.liv.ac.uk/~william/R/crosstab.r"

**Frequency & Percentage by Selected Center (d\_selected\_centre)**

crosstab(fq\_data, row.vars = "d\_selected\_centre", type = "f")

## d\_selected\_centre Count  
## 1 57  
## 2 52  
## Sum 109

crosstab(fq\_data, row.vars = "d\_selected\_centre", type = "r")

## d\_selected\_centre %  
## 1 52.29  
## 2 47.71  
## Sum 100.00

**Frequency by data Point Q1 (d\_q1) & Q3 (d\_q3)**

crosstab(fq\_data, row.vars = "d\_q1", type = "f")

## d\_q1 Count  
## 1 8  
## 10 2  
## 2 30  
## 3 11  
## 4 10  
## 5 5  
## 7 1  
## 8 27  
## 9 15  
## Sum 109

crosstab(fq\_data, row.vars = "d\_q3", type = "f")

## d\_q3 Count  
## 1 50  
## 2 12  
## 3 1  
## 4 35  
## 5 10  
## 6 1  
## Sum 109

**Crosstab Selected Center with data Point Q3 (d\_q3)**

crosstab(fq\_data, row.vars = "d\_selected\_centre", col.vars = "d\_q3", type = "f")

## d\_q3 1 2 3 4 5 6 Sum  
## d\_selected\_centre   
## 1 34 1 1 18 2 1 57  
## 2 16 11 0 17 8 0 52  
## Sum 50 12 1 35 10 1 109

**Statistical Summary of a Numeric Field**

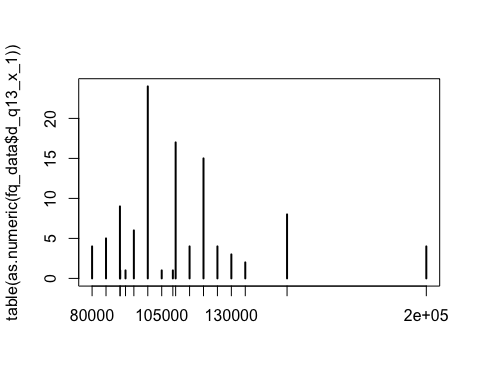
summary(as.numeric(fq\_data$d\_q13\_x\_1))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 80000 100000 110000 112028 120000 200000

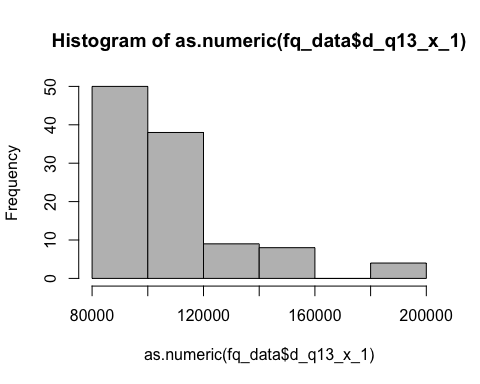
table(as.numeric(fq\_data$d\_q13\_x\_1))

##   
## 80000 85000 90000 90100 92000 95000 1e+05 105000 109000 110000 115000   
## 4 5 9 1 1 6 24 1 1 17 4   
## 120000 125000 130000 135000 150000 2e+05   
## 15 4 3 2 8 4

plot(table(as.numeric(fq\_data$d\_q13\_x\_1)))

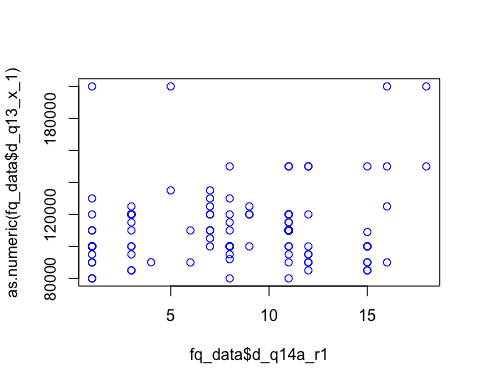


hist.default(as.numeric(fq\_data$d\_q13\_x\_1),col='gray')



**Bivariate statistical analysis** | **Scatter Plot** X = d\_q14a\_r1 ~ Y = d\_q13\_x\_1

plot(as.numeric(fq\_data$d\_q13\_x\_1)~fq\_data$d\_q14a\_r1,col='blue')



**T Test**

t.test(as.numeric(fq\_data$d\_q14a\_r1))

##   
## One Sample t-test  
##   
## data: as.numeric(fq\_data$d\_q14a\_r1)  
## t = 17.189, df = 108, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 7.077457 8.922543  
## sample estimates:  
## mean of x   
## 8