Compare API and DEV Simple Means

Mitch Toland

3/26/2020

Introduction

This report compares the means calculated by Ceres with the value store by Ceres and with the value calculated in the Domino development environment. The average of the values match the stored mean value.

Caution

The means being calculated are of the nonzero values only. This may not be what people want because some datasets have many zeros. Thus the reported means will be higher than those including the zero-values.

First the needed software must loaded.

```
library(jsonlite)
library(readxl)
```


Next, filepathes are created and summary funcitons are defined. The input json files only have a stored mean value in subsites under local under means. The global arm of hte json-files have a slightly different structure. Therefore all of the comparisons are based on the local arm of hte json-files.

```
zdir <- "/repos/RCB4Cloud/Reference/Data/prod_out/p360/"
lstfls <- list.files("/repos/RCB4Cloud/Reference/Data/prod_out/p360/")
smFileId   <- c(which(substr(lstfls, 11, 12) == "S8"),
                which(substr(lstfls, 11, 12) == "S9"))
smbsFileId <-   which(substr(lstfls, 11, 12) == "SB")

SMsummaryFunc <- function(zi){
  smi                <- fromJSON(paste0(zdir,lstfls[zi]))
  smigsmmf           <- (as.data.frame(do.call(cbind,
                                               lapply(smi$modelOutputs$simpleMeans$means$global$factors
  smimosm_input_data <- smi$modelOutputs$simpleMeans$input$data[[1]][, 1:10]

  smilsmms  <-            smi$modelOutputs$simpleMeans$means$local$subsites[[1]]
  smilsmmsm <- as.numeric(smi$modelOutputs$simpleMeans$means$local$subsites[[1]]$value)
  smilsmmsf <-            smi$modelOutputs$simpleMeans$means$local$subsites[[1]]$factors
  #smilsmmsa <-            smi$modelOutputs$simpleMeans$means$local$subsites[[1]]$answers
  #smilsmmsc <-            smi$modelOutputs$simpleMeans$means$local$subsites[[1]]$controls
  zms <- c(mean(           smilsmmsm               , na.rm=TRUE),
           mean(           smimosm_input_data$value, na.rm=TRUE),
           mean(as.numeric(smilsmmsf[[1]]$value)   , na.rm=TRUE))
  return(list(means=zms, differences=diff(round(zms, 10))))
}

SMBSsummaryFunc <- function(zi){
  smbsi                <- fromJSON(paste0(zdir,lstfls[zi]))
  smbsigsmmf           <- (as.data.frame(do.call(cbind, lapply(smbsi$modelOutputs$simpleMeans$means$glo
  smbsimosm_input_data <- smbsi$modelOutputs$simpleMeans$input$data[[1]][, 1:10]
```

```
  smbsilsmms  <-            smbsi$modelOutputs$simpleMeans$means$local$subsites[[1]]
  smbsilsmmsm <- as.numeric(smbsi$modelOutputs$simpleMeans$means$local$subsites[[1]]$value)
  smbsilsmmsf <-            smbsi$modelOutputs$simpleMeans$means$local$subsites[[1]]$factors
  #smbsilsmmsa <-            smbsi$modelOutputs$simpleMeans$means$local$subsites[[1]]$answers
  #smbsilsmmsc <-            smbsi$modelOutputs$simpleMeans$means$local$subsites[[1]]$controls
  zms <- c(mean(           smbsilsmmsm               , na.rm=TRUE),
           mean(           smbsimosm_input_data$value, na.rm=TRUE),
           mean(as.numeric(smbsilsmmsf[[1]]$value    , na.rm=TRUE)))
  return(list(means=zms, differences=diff(round(zms, 10))))
}
```


Next, run the summary functions on both the global and local json-files.

```
smm        <- as.data.frame(do.call(rbind, lapply(smFileId, function(zi){SMsummaryFunc(zi)$means})))
names(smm) <- c("LocalMean", "InputDataMean", "FactorValueMean")
smd        <- as.data.frame(do.call(rbind, lapply(smFileId, function(zi){SMsummaryFunc(zi)$differences}))
names(smd) <- c("InputMean-LocalMean", "FactorMean_InputMean")

smbsm        <- as.data.frame(do.call(rbind, lapply(smbsFileId, function(zi){SMBSsummaryFunc(zi)$means}))
names(smbsm) <- c("LocalMean", "InputDataMean", "FactorValueMean")
smbsd        <- as.data.frame(do.call(rbind, lapply(smbsFileId, function(zi){SMBSsummaryFunc(zi)$differe
names(smbsd) <- c("InputMean-LocalMean", "FactorMean_InputMean")
```


Finally, the three mean values along with their respective differences are presented for each input data file. The results for the global datasets is given first followed by the results for the local datasets.

```
smm
```

```
##    LocalMean InputDataMean FactorValueMean
## 1        NaN      49.55834        49.55834
## 2        NaN      22.16667        22.16667
## 3        NaN      26.85714        26.85714
## 4        NaN      21.55556        21.55556
## 5        NaN      26.37500        26.37500
## 6        NaN      55.53351        55.53351
## 7        NaN      30.33333        30.33333
## 8        NaN      49.01666        49.01666
## 9        NaN      38.76928        38.76928
## 10       NaN      42.81705        42.81705
```

```
#smboth   <- cbind(smm, smd)
#smboth
```



```
smd
```

```
##    InputMean-LocalMean FactorMean_InputMean
## 1                  NaN                    0
## 2                  NaN                    0
## 3                  NaN                    0
## 4                  NaN                    0
## 5                  NaN                    0
## 6                  NaN                    0
```

```
## 7                  NaN                    0
## 8                  NaN                    0
## 9                  NaN                    0
## 10                 NaN                    0
```

```
#smboth   <- cbind(smm, smd)
#smboth
```



```
smbsm
```

```
##    LocalMean InputDataMean FactorValueMean
## 1   54.26471      54.26471        54.26471
## 2   45.44444      45.44444        45.44444
## 3   43.22857      43.22857        43.22857
## 4   54.62500      54.62500        54.62500
## 5   52.96875      52.96875        52.96875
## 6   44.16667      44.16667        44.16667
## 7   41.41667      41.41667        41.41667
## 8   57.63636      57.63636        57.63636
## 9   40.88889      40.88889        40.88889
## 10  54.40625      54.40625        54.40625
## 11  44.00000      44.00000        44.00000
## 12  56.88889      56.88889        56.88889
## 13  41.25000      41.25000        41.25000
## 14  53.94444      53.94444        53.94444
## 15  43.97143      43.97143        43.97143
## 16  44.88889      44.88889        44.88889
## 17  56.86111      56.86111        56.86111
## 18  56.47222      56.47222        56.47222
## 19  42.30556      42.30556        42.30556
```

```
#smbsboth <- cbind(smbsm, smbsd)
#smbsboth
```



```
smbsd
```

```
##    InputMean-LocalMean FactorMean_InputMean
## 1                    0                    0
## 2                    0                    0
## 3                    0                    0
## 4                    0                    0
## 5                    0                    0
## 6                    0                    0
## 7                    0                    0
## 8                    0                    0
## 9                    0                    0
## 10                   0                    0
## 11                   0                    0
## 12                   0                    0
## 13                   0                    0
## 14                   0                    0
## 15                   0                    0
## 16                   0                    0
## 17                   0                    0
```

```
## 18                   0                    0
## 19                   0                    0
```

```
#smbsboth <- cbind(smbsm, smbsd)
#smbsboth
```


Conclusion

Overlooking the exclusion of zero values, the means appear to be calculated correctly.