An example of using the code to analyze a standardized dataset - Harding_30336784

Make sure you have all the packages installed before continue

```r
install.packages("sandwich")
install.packages("clubSandwich")
install.packages("WoodburyMatrix")
install.packages("dplyr")
install.packages("msm")
install.packages("lmerTest")
install.packages("splines")
install.packages("performance")
install.packages("stringr")
```

Load all functions

- vcovCRglmerMod.R is the robust variance estimator function for glmer
- fit_binary_CS.R: fit binary outcome data with cross-section (CS) design
- fit_binary_CO.R: fit binary outcome data with cohort (CO) design
- fit_gaussian_CS.R: fit gaussian outcome data with cross-section (CS) design
- fit_gaussian_CO.R: fit gaussian outcome data with cohort (CO) design
- fit_poisson_CS.R: fit poisson outcome data with cross-section (CS) design
- fit_poisson_CO.R: fit poisson outcome data with cohort (CO) design

```r
source("vcovCRglmerMod.R")
source("fit_binary_CS.R")
source("fit_binary_CO.R")
source("fit_gaussian_CS.R")
source("fit_gaussian_CO.R")
source("fit_poisson_CS.R")
source("fit_poisson_CO.R")
```

Load the standardized dataset

```r
# Harding_30336784: 2 poisson (no offset) and 1 binary, CS
# for now the loaded data has to be named as "data" in order to make the data processing automated
data <- read.csv("C:/Users/yongdong ouyang/Downloads/Harding_30336784.csv")
head(data)
```

```
##   id_cluster trt time time_on_trt out_poiss_1 out_poiss_2 out_bin_1
## 1          2   0    1           0           6          12         0
## 2          2   0    1           0           7          21         0
## 3          2   0    1           0           6           7         1
## 4          2   0    1           0          27          NA         1
## 5          2   0    1           0          13           7         1
## 6          2   0    1           0           7           7         1
```

Load the data_processing file to handle the imported dataset.

This function will do the following things:

- Format the standardized dataset: Return a list of datasets, with the number of datasets equals to the number of outcomes. Each dataset in the list will contain one specific outcome along with all the necessary variables for model fitting. For example, if we have three outcomes, this function will return a list of 3 datasets named as **data**. Each dataset contains one outcome with all other varibles we need for regression (e.g., cluster id, treatment indicator, exposure indicator etc)
- Provide some summary information, such as total number of outcomes, number of outcomes for each type

```r
source("Data_processing.R")

# check some summaries

# names and number of outcome columns (this gives us ideas about the outcome types)
outcome_columns # "out_poiss_1" "out_poiss_2" "out_bin_1"
```

```
## [1] "out_poiss_1" "out_poiss_2" "out_bin_1"
```

```r
length(outcome_columns) # 3
```

```
## [1] 3
```

```r
# Since there are poisson outcomes run the following code to see if there are offset columns
# ignore it if we don't have poisson outcomes
offset_columns # character(0)
```

```
## character(0)
```

```r
# We can further check how many continuous, binary or poisson outcomes we have:
length(gaussian_columns) #0
```

```
## [1] 0
```

```r
length(binary_columns) #1
```

```
## [1] 1
```

```r
length(poisson_columns) #2
```

```
## [1] 2
```

```r
# Check if we have individual ID column (a indicator whether we have CO or CS design)
length(ind_columns) # 0
```

```
## [1] 0
```

Upto here, we can quickly summarize that the Harding dataset has three outcomes (2 Poisson and 1 Binary), CS design, no offset for Poisson outcomes

Have a quick look of the processed dataset. Since we have three outcomes in total, the length of the data list is 3.

```r
# check the processed datasets
length(data) # 3
```

```
## [1] 3
```

```r
head(data[[1]]) # Outcome column is out_poiss_1
```

```
##   Cluster Treatment Period Exposure Outcome
## 1       2         0      1        0       6
## 2       2         0      1        0       7
## 3       2         0      1        0       6
## 4       2         0      1        0      27
## 5       2         0      1        0      13
## 6       2         0      1        0       7
```

```r
head(data[[2]]) # Outcome column is out_poiss_2
```

```
##   Cluster Treatment Period Exposure Outcome
## 1       2         0      1        0      12
## 2       2         0      1        0      21
## 3       2         0      1        0       7
## 4       2         0      1        0      NA
## 5       2         0      1        0       7
## 6       2         0      1        0       7
```

```r
head(data[[3]]) # Outcome column is out_bin_1
```

```
##   Cluster Treatment Period Exposure Outcome
## 1       2         0      1        0       0
## 2       2         0      1        0       0
## 3       2         0      1        0       1
## 4       2         0      1        0       1
## 5       2         0      1        0       1
## 6       2         0      1        0       1
```

Model fiting

```
test1 <- fit.data.poisson.CS(data = data[[1]], rve_type = "classic", ss_correct = T, offset = NULL)
```

For out_poiss_1 (this is the first element of the data list)

```
## IT model1 converged successfully.
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00354043 (tol = 0.002, component 1)
```

```
## ETI model2 converged successfully.
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## TEH model1 converged successfully.
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00465713 (tol = 0.002, component 1)
```

```
## NCS model2 converged successfully.
```

```
names(test1)
```

```
## [1] "IT"          "ETI"         "NCS"         "TEH"         "convergence"
```

```
names(test1$IT)
```

```
## [1] "est"     "vcov"    "rve"     "summary"
```

```
names(test1$ETI)
```

```
## [1] "est"     "vcov"    "rve"     "summary" "lrtest"
```

```
names(test1$NCS)
```

```
## [1] "est"     "vcov"    "rve"     "summary"
```

```
names(test1$TEH)
```

```
## [1] "est"
```

```
names(test1$convergence)
```

```
## [1] "IT"  "ETI" "TEH" "NCS"
```

```r
test1$convergence
```

```
##   IT ETI TEH NCS
## 1  1   2   1   2
```

The outputs (e.g., test1) contains the following items:

- test1 is list contains following items:
  - **IT**: information about immediate treatment effect model
  - **ETI**: information about exposure time indicator model
  - **NCS**: information about natural cubic spline model
  - **TEH**: information about treatment heterogeneity exposure time model
  - **convergence**: information about which model was converged, following most complicated to least complicated (for TEH model, the random effect for exposure time is always in the model)
  - model 1:
    - random intercept + random cluster-period + random individual effect (for a CO design)
    - random intercept + random cluster-period (for a CS design)
  - model 2:
    - random intercept + random cluster-period (for a CO design)
    - random intercept (for a CS design)
  - model 3:
    - random intercept (for a CO design)
    - NA for CS design
- Within each model information we have:
  - **est**: estimation table include the fixed effect estimates, SE, p-value etcs
  - **vcov**: variance-covariance matrix
  - **rve**: variance-covariance matrix with RVE
  - **summary**: average treatment effect estimates, SE and CI with/without RVE (CIs have (or do not have small sample corrections based on the input of "ss_correct"))

```r
test2 <- fit.data.poisson.CS(data = data[[2]], rve_type = "classic", ss_correct = T, offset = NULL)
```

out_poiss_2 (this is the second element of the data list)

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0026963 (tol = 0.002, component 1)
```

```
## IT model1 converged successfully.
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00256926 (tol = 0.002, component 1)
```

```
## ETI model2 converged successfully.
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00339482 (tol = 0.002, component 1)
## TEH model1 converged successfully.

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00607882 (tol = 0.002, component 1)

## NCS model2 converged successfully.
```

```
test3 <- fit.data.binary.CS(data = data[[3]], rve_type = "classic", ss_correct = T)
```

out__bin__1 (this is the third element of the data list)

```
## boundary (singular) fit: see help('isSingular')

## IT model1 converged successfully.

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0100283 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00597997 (tol = 0.002, component 1)

## ETI Both models failed to converge.

## boundary (singular) fit: see help('isSingular')

## TEH model1 converged successfully.

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00827484 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00321053 (tol = 0.002, component 1)

## NCS Both models failed to converge.
```

Sometimes, none of the models we tried under a specific type were converged. For example, here all models we tried under ETI and NCS models did not converge. In this case, nothing will be returned.

```
test3$ETI
```

```
## $summary
## data frame with 0 columns and 0 rows
```

```
test3$NCS
```

```
## $summary
## data frame with 0 columns and 0 rows
```