Response Surface Analysis with Clustered Data

James Rigby

October 29, 2018

## Overview

The resp\_surf() function was written to solve a problem frequently encountered in psychological research testing fit-hypotheses: clustered data. Failing to address clustered data results in incorrect estimates of standard errors which causes inflated type I and type II erros when conducting hypothesis testing (Bliese & Hanges, 2004). This function automates the methodology outlined in Edwards & Parry (1993) and allows for the user to indicate a clustering variable that will be accounted for using cluster robust standard errors as implemented by the sandwich package (see ?vcovCL and Cameron et al, 2011 for more information)

The companion function plot\_surf() provides quick and easy wrapper that plots objects created by the resp\_surf() fucntion.

## List of Arguments

### resp\_surf()

1. data: a data frame that contains all variables of interest. Note that quadratic terms will be created for you, although the user must center data at the appropriate values.
2. dep\_var: The quoted name of your dependent variable
3. fit\_var: A character vector of the two variables you are interested in examining congruence. Must be length 2.
4. control: A character vector that contains the names of control variables of interest.
5. robust: A logical value. If TRUE robust standard error should be used. If a cluster is not provided, a simple sandwich covariance is used instead
6. cluster: a character value that specifies the name of the clustering variable

### plot\_surf()

1. obj: an object created by resp\_surf()
2. max.x: the maximum value of your first fit\_var to be plotted. Order is determined by the order you entered the variables in resp\_surf()
3. min.x: the minimum value of your first fit\_var to be plotted.
4. max.y: the maximum value of your second fit\_var to be plotted.
5. min.y: the minimum value of your second fit\_var to be plotted.
6. inc: the increments it should be plotted. Smaller increments results in more grid lines.
7. phi: the vertical rotation of the graph.
8. theta: the horizontal rotation of the graph.
9. llables: Whether lines of interest should include labels (experimental)
10. …: takes any argument that persp() can take

## Example 1: resp\_surf() with No Cluster

library(tidyverse)

## -- Attaching packages -------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.6  
## v tidyr 0.8.1 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ----------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(sandwich) #Make sure to install and download this package.   
  
#Loading the functions. Works when resp\_surf.R and plot\_surf.R are saved in the working directory  
source("resp\_surf.R")  
source("plot\_surf.R")  
  
#Reading in the data  
precept\_data<-read.csv("TCH Final Dataset.csv")  
  
#Some of the tenure variables were entered as character - fixing the two obs and converting to integer  
precept\_data$yrs\_nurse<-as.character(precept\_data$yrs\_nurse)  
precept\_data$yrs\_nurse[c(18,20)]<-c(1.5, 20)  
precept\_data$yrs\_nurse<-as.integer(precept\_data$yrs\_nurse)  
  
# Centering Fit Variables at Scale Center ---------------------------------  
#This step is still necessary. resp\_surf() will not do this for you, but it will warn you if it notices that things aren't centered  
precept\_data<-precept\_data%>%  
 mutate\_at(vars(DH, dhp, AH, ahp), funs(c=(.-4)))%>%  
 mutate\_at(vars(yrs\_nurse), funs(c = .-mean(yrs\_nurse, na.rm =TRUE)))  
  
#Model 1: No clustering  
irb\_DH\_c<-resp\_surf(dep\_var = "irb", fit\_var = c("DH\_c", "dhp\_c"), control = c("yrs\_nurse"), data = precept\_data)  
  
#Table illustrating differences and congruence  
irb\_DH\_c$dif\_tab

##   
## DH\_c < dhp\_c DH\_c congruent to dhp\_c DH\_c > dhp\_c   
## 0.2427184 0.3883495 0.3009709

#Results generated by selected options. When robust = FALSE results will equal lm() output  
irb\_DH\_c$results

## Estimates Standard\_Error T\_Test P\_value  
## (Intercept) 4.885331463 0.172972750 28.2433589 0.0000  
## DH\_c 0.273112649 0.138565219 1.9710044 0.0521  
## dhp\_c 0.117629344 0.085811523 1.3707873 0.1742  
## DH\_c\_sq 0.026820612 0.039210973 0.6840078 0.4959  
## dhp\_c\_sq 0.012013802 0.019198723 0.6257605 0.5332  
## yrs\_nurse 0.001436717 0.005594031 0.2568304 0.7980  
## DH\_c:dhp\_c 0.086578848 0.042695601 2.0278166 0.0458

# Data frame containing linear and quadratic tests for lines of interest   
irb\_DH\_c$loi

## Line\_of\_Interest Parameter Estimate Standard\_Error T\_Test  
## 1 Line of Congruence Linear 0.39074199 0.20502517 1.9058245  
## 2 Line of Congruence Quadratic 0.12541326 0.07035606 1.7825509  
## 3 Line of Incongruence Linear 0.15548331 0.10532119 1.4762776  
## 4 Line of Incongruence Quadratic -0.04774443 0.05172769 -0.9229957  
## P\_value  
## 1 0.06017709  
## 2 0.07836066  
## 3 0.14369941  
## 4 0.35871738

# Calculations for stationary point  
irb\_DH\_c$stat\_pnt

## X. Value  
## dhp\_c X Coordinates -0.5835275  
## DH\_c Y Coordinates -2.7929630

# Calculations for principle axes  
irb\_DH\_c$princ\_axis

## Axis Param Estimate  
## 1 First Intercept -2.3007590  
## 2 First Slope 0.8434976  
## 3 Second Intercept -3.4847581  
## 4 Second Slope -1.1855398

# Raw output from lm(). Used in majority of behind the scenes calculations  
irb\_DH\_c$model

##   
## Call:  
## lm(formula = equation, data = d)  
##   
## Coefficients:  
## (Intercept) DH\_c dhp\_c DH\_c\_sq dhp\_c\_sq   
## 4.885331 0.273113 0.117629 0.026821 0.012014   
## yrs\_nurse DH\_c:dhp\_c   
## 0.001437 0.086579

# Equation that was used to fit model  
irb\_DH\_c$equation

## irb ~ DH\_c + dhp\_c + DH\_c\_sq + dhp\_c\_sq + DH\_c:dhp\_c + yrs\_nurse  
## <environment: 0x00000000186a2228>

## Example 2: Cluster Robust Standard Errors

# Format is identical with two changes: robust = TRUE, cluster = "preceptor\_name"  
irb\_DH\_c<-resp\_surf(dep\_var = "irb", fit\_var = c("DH\_c", "dhp\_c"), data = precept\_data, control = c("yrs\_nurse"), robust = TRUE, cluster = "preceptor\_name" )  
  
#Standard errors have changed as the method has changed. Format of the output remains the same.   
irb\_DH\_c$results

## Estimates Standard\_Error T\_Test P\_value  
## (Intercept) 4.885331463 0.10573511 46.2034948 0.0000  
## DH\_c 0.273112649 0.08738328 3.1254565 0.0025  
## dhp\_c 0.117629344 0.04933568 2.3842650 0.0194  
## DH\_c\_sq 0.026820612 0.02896562 0.9259465 0.3572  
## dhp\_c\_sq 0.012013802 0.01464242 0.8204791 0.4143  
## yrs\_nurse 0.001436717 0.00668075 0.2150533 0.8303  
## DH\_c:dhp\_c 0.086578848 0.03673711 2.3567138 0.0208

## resp\_surf(): Plotting Examples

plot\_surf(irb\_DH\_c, max.x = 3, min.x = -3, max.y = 3, min.y = -3, inc = 1, ticktype = "detailed")

