# Package ‘GPSCDF’

January 20, 2025

**Type** Package

**Title** Generalized Propensity Score Cumulative Distribution Function

**Version** 0.1.1

**Description** Implements the generalized propensity score cumulative distribution function proposed by Greene (2017) <https://digitalcommons.library.tmc.edu/dissertations/AAI10681743/>. A single scalar balancing score is calculated for any generalized propensity score vector with three or more treatments. This balancing score is used for propensity score matching and stratification in outcome analyses when analyzing either ordinal or multinomial treatments.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr (>= 0.7.6) , nbpMatching (>= 1.5.1), nnet (>= 7.3-12), MASS (>= 7.3-49), stats (>= 3.4), survival (>= 2.41-3), utils (>= 3.4)

**RoxygenNote** 6.1.1

**Suggests** testthat

**NeedsCompilation** no

**Author** Derek W. Brown [aut, cre],
Thomas J. Greene [aut],
Stacia M. DeSantis [aut]

**Maintainer** Derek W. Brown <derek9@gwu.edu>

**Repository** CRAN

**Date/Publication** 2019-03-05 15:20:18 UTC

## Contents

## GPSCDF *Generalized Propensity Score Cumulative Distribution Function (GPS-CDF)*

### Description

GPSCDF takes in a generalized propensity score (GPS) object with length >2 and returns the GPS-CDF balancing score.

### Usage

```
GPSCDF(pscores = NULL, data = NULL, trt = NULL, stratify = FALSE,
  nstrat = 5, optimal = FALSE, greedy = FALSE, ordinal = FALSE,
  multinomial = FALSE, caliper = NULL)
```

### Arguments

| | |
|---|---|
| pscores | The object containing the treatment ordered generalized propensity scores for each subject. |
| data | An optional data frame to attach the calculated balancing score. The data frame will also be used in stratification and matching. |
| trt | An optional object containing the treatment variable. |
| stratify | Option to produce strata based on the power parameter (ppar). Default is FALSE. |
| nstrat | An optional parameter for the number of strata to be created when stratify is set to TRUE. Default is 5 strata. |
| optimal | Option to perform optimal matching of subjects based on the power parameter (ppar). Default is FALSE. |
| greedy | Option to perform greedy matching of subjects based on the power parameter (ppar). Default is FALSE. |
| ordinal | Specifies ordinal treatment groups for matching. Subjects are matched based on the ratio of the squared difference of power parameters for two subjects, ppar_i and ppar_j, in the numerator and the squared difference in observed treatment received, trt_i and trt_j, in the denominator: (ppar_i-ppar_j)^2/(trt_i-trt_j)^2. Default is FALSE. |
| multinomial | Specifies multinomial treatment groups for matching. Subjects are matched based on the absolute difference of power parameters for two subjects, ppar_i and ppar_j, who received different treatments: \|ppar_i - ppar_j\|. Default is FALSE. |
| caliper | An optional parameter for the caliper value used when performing greedy matching. Used when greedy is set to TRUE. Default is .25*sd(ppar). |

## Details

The `GPSCDF` method is used to conduct propensity score matching and stratification for both ordinal and multinomial treatments. The method directly maps any GPS vector (with length >2) to a single scalar value that can be used to produce either average treatment effect (ATE) or average treatment effect among the treated (ATT) estimates. For the `K` multinomial treatments setting, the balance achieved from each `K!` ordering of the GPS should be assessed to find the optimal ordering of the GPS vector (see Examples for more details).

## Value

`ppar` The power parameter scalar balancing score to be used in outcome analyses through stratification or matching.

`data` The user defined dataset with power parameter (ppar), strata, and/or optimal matching variables attached.

`nstrat` The number of strata used for stratification.

`strata` The strata produced based on the calculated power parameter (`ppar`).

`optmatch` The optimal matches produced based on the calculated power parameter (`ppar`).

`optdistance` The average absolute total distance of power parameters (`ppars`) for optimally matched pairs.

`caliper` The caliper value used for greedy matching.

`grddata` The user defined dataset with greedy matching variable attached.

`grdmatch` The greedy matches produced based on the calculated power parameter (`ppar`).

`grdydistance` The average absolute total distance of power parameters (`ppars`) for greedy matched pairs.

## Author(s)

Derek W. Brown, Thomas J. Greene, Stacia M. DeSantis

## References

Greene, TJ. (2017). Utilizing Propensity Score Methods for Ordinal Treatments and Prehospital Trauma Studies. Texas Medical Center Dissertations (via ProQuest).

## Examples

```
### Example: Create data example
N<- 100

set.seed(18201) # make sure data is repeatable
Sigma <- matrix(.2,4,4)
diag(Sigma) <- 1
data<-matrix(0, nrow=N, ncol=6,dimnames=list(c(1:N),
      c("Y","trt",paste("X",c(1:4),sep=""))))
data[,3:6]<-matrix(MASS::mvrnorm(N, mu=rep(0, 4), Sigma,
      empirical = FALSE) , nrow=N, ncol = 4)
```

```
dat<-as.data.frame(data)


#Create Treatment Variable
tlogits<-matrix(0,nrow=N,ncol=2)
tprobs<-matrix(0,nrow=N,ncol=3)

alphas<-c(0.25, 0.3)
strongbetas<-c(0.7, 0.4)
modbetas<-c(0.2, 0.3)

for(j in 1:2){
  tlogits[,j]<- alphas[j] + strongbetas[j]*dat$X1 + strongbetas[j]*dat$X2+
                modbetas[j]*dat$X3 + modbetas[j]*dat$X4
}

for(j in 1:2){
  tprobs[,j]<- exp(tlogits[,j])/(1 + exp(tlogits[,1]) + exp(tlogits[,2]))
  tprobs[,3]<- 1/(1 + exp(tlogits[,1]) + exp(tlogits[,2]))
}

set.seed(91187)
for(j in 1:N){
  data[j,2]<-sample(c(1:3),size=1,prob=tprobs[j,])
}


#Create Outcome Variable
ylogits<-matrix(0,nrow=N,ncol=1,dimnames=list(c(1:N),c(”Logit(P(Y=1))”)))
yprobs<-matrix(0,nrow=N,ncol=2,dimnames=list(c(1:N),c(”P(Y=0)”,”P(Y=1)”)))

for(j in 1:N){
  ylogits[j,1]<- -1.1 + 0.7*data[j,2] + 0.6*dat$X1[j] + 0.6*dat$X2[j] +
                 0.4*dat$X3[j] + 0.4*dat$X4[j]

  yprobs[j,2]<- 1/(1+exp(-ylogits[j,1]))

  yprobs[j,1]<- 1-yprobs[j,2]
}

set.seed(91187)
for(j in 1:N){
  data[j,1]<-sample(c(0,1),size=1,prob=yprobs[j,])
}

dat<-as.data.frame(data)


### Example: Using GPSCDF

#Create the generalized propensity score (GPS) vector using any parametric or
#nonparametric model
```

```
glm<- nnet::multinom(as.factor(trt)~ X1+ X2+ X3+ X4, data=dat)
probab<- round(predict(glm, newdata=dat, type="probs"),digits=8)
gps<-cbind(probab[,1],probab[,2],1-probab[,1]-probab[,2])


#Create scalar balancing power parameter
fit<-GPSCDF(pscores=gps)

## Not run:
  fit$ppar

## End(Not run)


#Attach scalar balancing power parameter to user defined data set
fit2<-GPSCDF(pscores=gps, data=dat)

## Not run:
  fit2$ppar
  fit2$data

## End(Not run)


### Example: Ordinal Treatment

#Stratification
fit3<-GPSCDF(pscores=gps, data=dat, stratify=TRUE, nstrat=5)

## Not run:
  fit3$ppar
  fit3$data
  fit3$nstrat
  fit3$strata

  library(survival)
  model1<-survival::clogit(Y~as.factor(trt)+X1+X2+X3+X4+strata(strata),
                           data=fit3$data)
  summary(model1)

## End(Not run)


#Optimal Matching
fit4<- GPSCDF(pscores=gps, data=dat, trt=dat$trt, optimal=TRUE, ordinal=TRUE)

## Not run:
  fit4$ppar
  fit4$data
  fit4$optmatch
  fit4$optdistance
  library(survival)
  model2<-survival::clogit(Y~as.factor(trt)+X1+X2+X3+X4+strata(optmatch),
                           data=fit4$data)
  summary(model2)

## End(Not run)


#Greedy Matching
fit5<- GPSCDF(pscores=gps, data=dat, trt=dat$trt, greedy=TRUE, ordinal=TRUE)

## Not run:
  fit5$ppar
  fit5$data
  fit5$caliper
  fit5$grddata
  fit5$grdmatch
  fit5$grdydistance

  library(survival)
  model3<-survival::clogit(Y~as.factor(trt)+X1+X2+X3+X4+strata(grdmatch),
                           data=fit5$grddata)
  summary(model3)

## End(Not run)


### Example: Multinomial Treatment

#Create all K! orderings of the GPS vector
gps1<-cbind(gps[,1],gps[,2],gps[,3])
gps2<-cbind(gps[,1],gps[,3],gps[,2])
gps3<-cbind(gps[,2],gps[,1],gps[,3])
gps4<-cbind(gps[,2],gps[,3],gps[,1])
gps5<-cbind(gps[,3],gps[,1],gps[,2])
gps6<-cbind(gps[,3],gps[,2],gps[,1])

gpsarry<-array(c(gps1, gps2, gps3, gps4, gps5, gps6), dim=c(N,3,6))


#Create scalar balancing power parameters for each ordering of the GPS vector
fit6<- matrix(0,nrow=N,ncol=6,dimnames=list(c(1:N),c(”ppar1”,”ppar2”,”ppar3”,
              ”ppar4”,”ppar5”,”ppar6”)))

## Not run:
for(i in 1:6){
  fit6[,i]<-GPSCDF(pscores=gpsarry[,,i])$ppar
}

  fit6

#Perform analyses (similar to ordinal examples) using each K! ordering of the
#GPS vector. Select ordering which achieves optimal covariate balance
```

```
#(i.e. minimal standardized mean difference).

## End(Not run)
```

# Index