Package ‘DiPs’

April 11, 2020

Type Package

Title Directional Penalties for Optimal Matching in Observational Studies

Version 0.6.1

Author Ruoqi Yu

Maintainer Ruoqi Yu <ruoqiyu@wharton.upenn.edu>

Description Improves the balance of optimal matching with near-fine balance by giving penalties on the unbalanced covariates with the unbalanced directions. Many directional penalties can also be viewed as Lagrange multipliers, pushing a matched sample in the direction of satisfying a linear constraint that would not be satisfied without penalization. Yu, R., and Rosenbaum, P. R. (2019). <doi:10.1111/biom.13098>.

License MIT+file LICENSE

Encoding UTF-8

LazyData true

Imports stats, liqueuer, plyr, mvnfast, methods

Suggests optmatch

Note One minimum cost flow problem may have several or many solutions that are equivalent in providing the same minimum total or mean cost. Minor differences between computers or implementations may have the minor consequence of altering which equivalent solution is produced.

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-11 05:00:09 UTC

R topics documented:

addcaliper ................................................................. 2
addDirectPenalty .......................................................... 3
addMagnitudePenalty .................................................... 4
check ....................................................................... 5
addcaliper

Add a caliper, that need not be symmetric, to a distance object.

Description

Imposes a caliper, that need not be symmetric, on p using a penalty function, adding the penalty to a distance matrix dmat and returning a new distance matrix.

This symmetric version of this function is discussed in Rosenbaum (2010).

Usage

addcaliper(dist, z, dx, rg, stdev = FALSE, penalty = 1000)

Arguments

dist A distance object with three components: d, start, end, typically created by maha_dense or maha_sparse. d[i] gives the distance between the (start[i])th treated and the (end[i]-sum(z))th control.

z A vector whose ith coordinate is 1 for a treated unit and is 0 for a control. Must have treated subjects (z=1) before controls (z=0).

dx A vector of with length(z)=length(dx) giving the variable used to define the caliper. For instance, dx might be the propensity score.

rg A vector with length(rg)=2 such that rg[1] <= 0 <= rg[2]. If the treated-minus-control difference in dx is < rg[1] or > rg[2], then penalty is added to the distance. If treated individuals have dx higher than controls, then you want to set rg[2] < -rg[1], so that you tolerate smaller positive differences and larger negative differences.

stdev If stdev = TRUE, rg is interpreted in units of an equally weighted pooled standard deviation; that is, rg is replaced by rg*sp where sp is sqrt((var(dx[z==1])+var(dx[z==0]))/2).

penalty The number added to a distance when the caliper is violated. A large penalty, like the default value of penalty = 1000, will try to enforce the caliper to the extent that this is feasible. Small penalties can slightly tilt a match in a desired direction.

Value

Returns a new distance object whose distance component d is updated by the sum of d and the penalties for caliper violations.
addDirectPenalty

Add a directional penalty to a distance object

Description
Add a directional penalty to a distance object.

Usage
addDirectPenalty(dist, z, dx, positive = TRUE, penalty = 1)

Arguments

- dist: A distance object with three components: d, start, end, typically created by maha_dense or maha_sparse. d[i] gives the distance between the (start[i])th treated and the (end[i]-sum(z))th control.
- z: A vector whose ith coordinate is 1 for a treated unit and 0 for a control. Must have treated subjects (z=1) before controls (z=0).
- dx: A vector of with length(z)=length(dx) giving the variable used to define the caliper. For instance, dx might be the propensity score.
- positive: If positive = TRUE, a treated-minus-control difference in dx that is positive is increased by penalty, but if positive = FALSE a a treated-minus-control difference in dx that is negative is increased by penalty. Zero differences are never penalized.
- penalty: The number added to a distance when the desired direction is violated.

Value
Returns a new distance matrix that is the sum of dmat and the penalties for direction violations.

References

Examples

data("nh0506Homocysteine")
attach(nh0506Homocysteine)
X<-cbind(female,age,black,education,povertyr,bmi)
p<-glm(z~female+age+black+education+povertyr+bmi,family=binomial)$fitted.values
d<-cbind(nh0506Homocysteine,p)
detach(nh0506Homocysteine)
dist0<-maha_dense(d$z,X)
#symmetric caliper
dist1<-addcaliper(dist0, d$z, d$p, c(-.3,.3), stdev = TRUE, penalty = 1000)
head(dist1$d)
#asymmetric caliper
dist2<-addcaliper(dist0, d$z, d$p, c(-.5,.1), stdev = TRUE, penalty = 1000)
head(dist2$d)
References


Examples

data("nh0506Homocysteine")
attach(nh0506Homocysteine)
X<-cbind(female,age,black,education,povertyr,bmi)
p<-glm(z~female+age+black+education+povertyr+bmi,family=binomial)$fitted.values
d<-cbind(nh0506Homocysteine,p)
detach(nh0506Homocysteine)
dist<-maha_dense(d$z,X)
head(dist$d)
dist<-addDirectPenalty(dist, d$z, d$p, positive=TRUE, penalty = 1)
head(dist$d)

addMagnitudePenalty

Add a directional magnitude penalty to a distance matrix

Description

Adds a penalty to the distance component d of the distance object dist depending upon value of dx. The distance object dist has three components: d, start, end. d[i] gives the distance between the tth treated and the cth control, with t=start[i] and c=end[i]-sum(z). The value of dx for treated unit t, say dxt, is dx[z==1][t] and the value of dx for control c, say dx[c], is dx[z==0][c]. Then, d[i] is adjusted using dxt-dxc. If positive=TRUE, the adjustment changes d[i] to d[i] + multiplier*(max(0, (dxt-dxc)-hstick)). That is, a penalty is imposed if dxt exceeds dxc by more than hstick. If positive=FALSE, the adjustment changes d[i] to d[i] + multiplier*(max(0, (dxc-dxt)-hstick)).

Usage

addMagnitudePenalty(dist, z, dx, positive = TRUE, hstick = 0, multiplier = 2)

Arguments

dist A distance object with three components: d, start, end, typically created by maha_dense or maha_sparse. d[i] gives the distance between the (start[i])th treated and the (end[i]-sum(z))th control.

z A vector whose ith coordinate is 1 for a treated unit and is 0 for a control. Must have treated subjects (z=1) before controls (z=0).

dx A vector of with length(z)=length(dx) giving the variable used to define the caliper. For instance, dx might be the propensity score.

positive If positive = TRUE, a treated-minus-control difference di in dx that is positive is increased by a multiple of ldl, but if positive = FALSE a a treated-minus-control difference in dx that is negative is increased by a multiple of ldl.

hstick Hockey-stick value, which is a nonnegative number. See the description.

multiplier The magnitude added is multiplier*lidl/s where s is an equally weighted, pooled within group standard deviation of dx.
check

Value

Returns a new distance object whose distance component \( d \) is updated by the sum of \( d_{\text{mat}} \) and the penalties.

References


Examples

```r
## Not run:
library(MASS)
# data("nh0506Homocysteine")
attach(nh0506Homocysteine)
# Select covariates
X<-cbind(female, age, black, education, poverty, bmi)
# Propensity score
p<-glm(z~female+age+black+education+poverty+bmi,
      family=binomial)$fitted.values
d<-cbind(nh0506Homocysteine,p)
detach(nh0506Homocysteine)
dist<-maha_dense(d$z,X)
head(dist$d)
# Impose a penalty when a treated individual has a higher propensity
# score than a control
dist<-addMagnitudePenalty(dist, d$z, d$p, positive=TRUE, multiplier = 20)
head(dist$d)
## End(Not run)
```

---

check

Check standardized mean differences (SMDs) of the matched data set.

Description

The function is used to create a table of mean and SMDs to check the balance before and after matching.

Usage

```r
check(fdata, mdata, fz, mz)
```

Arguments

- **fdata** A full data frame with length(fz) rows and columns being variables that need to check SMDs. fdata and mdata must have the same variables with same column names in the same order.
mdata  A matched data frame with length(mz) rows and columns being variables that
need to check SMDs. fdata and mdata must have the same variables with same
column names in the same order.

fz  A vector whose ith coordinate is 1 for a treated unit and is 0 for a control for
subjects in the full data set.

mz  A vector whose ith coordinate is 1 for a treated unit and is 0 for a control for
subjects in the matched data set.

Value

A matrix with one row for each variable and five columns being the mean of treated group, mean
of matched control group, mean of full control group, SMD of matched control group and SMD of
full control group.

References


Examples

# To run this example, you must load the optmatch package.

data("nh0506Homocysteine")
attach(nh0506Homocysteine)
X<-cbind(female,age,black,education,poverty,bmi)
p<-glm(z~female+age+black+education+poverty+bmi,family=binomial)$fitted.values
d<-cbind(nh0506Homocysteine,p)
detach(nh0506Homocysteine)
dist<-maha_dense(d$z,X)
o<-match(d$z, dist, d)
matcheddata<-o$data
Xm<-subset(matcheddata, select=c('female','age','black','education','poverty','bmi','p'))
check(cbind(X,p),Xm,d$z,matcheddata$z)

edgenum  Computes the number of edges in the reduced bipartite graph.

Description

Computes the number of edges in the reduced bipartite graph after applying the caliper and number
of nearest neighbors (constant). Equivalently, this is the number of candidate pairs for matching in
the observational study.

This function can provide users some idea of the required computation time. Smaller caliper and
constant removes more edges, hence accelerates computation, but risks infeasibility.
**edgenum**

**Usage**

```r
edgenum(z, p, caliper, constant=NULL, exact=NULL, ties.all=TRUE)
```

**Arguments**

- `z` A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
- `p` A vector of length(z)=length(p) giving the variable used to define the caliper. Typically, p is the propensity score or its rank.
- `caliper` If two individuals differ on p by more than caliper, we will not calculate the distance for this pair.
- `constant` If the number of pairs within a caliper is greater than constant, we will select the constant closest ones.
- `exact` If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.
- `ties.all` If ties.all is True, include all ties while choosing nearest neighbors. In this case, some treated may have more than constant controls. Otherwise, randomly select one or several controls to make sure there are not more than constant controls for each treated.

**Details**

A given choice of caliper and number of nearest neighbors (constant) removes candidate pairs, so there exists a corresponding reduced bipartite graph.

Smaller caliper and constant removes more edges from the original dense graph, hence the computation is faster. However, this risks infeasibility. A smallest caliper that permits a feasible match and its corresponding smallest number of nearest neighbors can be computed by functions optcal() and optconstant().

**Value**

Number of edges in the reduced bipartite graph with the constraints on caliper and number of nearest neighbors (constant).

**Examples**

```r
data(nh0506Homocysteine)
attach(nh0506Homocysteine)
p<-glm(z~female+age+black+education+poverty+bmi,family=binomial)$fitted.values
edgenum(z,p,0.2)
edgenum(z,p,0.2,10,exact=female)
detach(nh0506Homocysteine)
```
maha_dense

Creating a robust Mahalanobis distance for matching based on a dense network.

Description

Computes a robust Mahalanobis distance list for use in dense matching. In this case, we compute the distance for all possible pairs of treated and control.

This function and its use are discussed in Rosenbaum (2010). The robust Mahalanobis distance is described in Chapter 8 of Rosenbaum (2010).

Usage

maha_dense(z, X, exact=NULL,
nearexact=NULL, penalty=100)

Arguments

z A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
X A matrix with length(z) rows giving the covariates. X should be of full column rank.
exact If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.
nearexact If not NULL, then a vector of length length(z) giving variable that need to be exactly matched.
penalty The penalty for a mismatch on nearexact.

Details

The usual Mahalanobis distance works well for multivariate Normal covariates, but can exhibit odd behavior with typical covariates. Long tails or an outlier in a covariate can yield a large estimated variance, so the usual Mahalanobis distance pays little attention to large differences in this covariate. Rare binary covariates have a small variance, so a mismatch on a rare binary covariate is viewed by the usual Mahalanobis distance as extremely important. If you were matching for binary covariates indicating US state of residence, the usual Mahalanobis distance would regard a mismatch for Wyoming as much worse than a mismatch for California.

The robust Mahalanobis distance uses ranks of covariates rather than the covariates themselves, but the variances of the ranks are not adjusted for ties, so ties do not make a variable more important. Binary covariates are, of course, heavily tied.

Value

d A distance object for each pair of treated and control.
start The treated subject for each distance.
end The control subject for each distance.
maha_sparse

References

Examples

data("nh0506Homocysteine")
attach(nh0506Homocysteine)
X<-cbind(female,age,black,education,povertyr,bmi)
dist<-maha_dense(z,X)
head(dist$d)
detach(nh0506Homocysteine)

maha_sparse

Creates a robust Mahalanobis distance for matching based on a sparse network.

Description
Computes a robust Mahalanobis distance list for use in sparse matching. In this case, we will only calculate the distance for pairs within the caliper on p. If the caliper is too small, the matching may be infeasible. For the smallest caliper that keeps feasibility, refer to optcal() in package 'bigmatch'.

This function and its use are discussed in Rosenbaum (2010). It is preferred when the dataset is large. The robust Mahalanobis distance in described in Chapter 8 of Rosenbaum (2010).

Usage
maha_sparse(z,X,p=rep(1,length(z)),caliper=1,stdev=FALSE,constant=NULL,ncontrol=1,
exact=NULL,nearexact=NULL,penalty=100,subX=NULL,ties.all=TRUE)

Arguments
z  A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
X  A matrix with length(z) rows giving the covariates. X should be of full column rank.
p  A vector of length(z)=length(p) giving the variable used to define the caliper. Typically, p is the propensity score or its rank.
caliper  If two individuals differ on p by more than caliper, we will not calculate the distance for this pair. If caliper is a positive number, then a symmetric caliper is applied. If caliper is a vector of a negative number and a positive number, then an asymmetric caliper is applied.
stdev  If stdev = TRUE, caliper is interpreted in units of an equally weighted pooled standard deviation; that is, caliper is replaced by caliper*sp where sp is sqrt((var(dx[z==1])+var(dx[z==0]))/2).
constant  If the number of pairs within a caliper is greater than constant, we will select the constant closest ones.
ncontrol A positive integer giving the number of controls to be matched to each treated subject. If ncontrol is too large, the match will be infeasible.

exact If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.

nearexact If not NULL, then a vector of length length(z) giving variable that need to be exactly matched.

penalty The penalty for a mismatch on nearexact.

subX If a subset matching is required, the variable that the subset matching is based on. That is, for each level of subX, extra treated will be discarded in order to have the number of matched treated subjects being the minimum size of treated and control groups. If exact matching on a variable x is desired and discarding extra treated is fine if there are more treated than controls for a certain level k, set exact=x, subX=x.

ties.all If ties.all is True, include all ties while choosing nearest neighbors. In this case, some treated may have more than constant controls. Otherwise, randomly select one or several controls to make sure there are not more than constant controls for each treated.

Details
The usual Mahalanobis distance works well for multivariate Normal covariates, but can exhibit odd behavior with typical covariates. Long tails or an outlier in a covariate can yield a large estimated variance, so the usual Mahalanobis distance pays little attention to large differences in this covariate. Rare binary covariates have a small variance, so a mismatch on a rare binary covariate is viewed by the usual Mahalanobis distance as extremely important. If you were matching for binary covariates indicating US state of residence, the usual Mahalanobis distance would regard a mismatch for Wyoming as much worse than a mismatch for California.

The robust Mahalanobis distance uses ranks of covariates rather than the covariates themselves, but the variances of the ranks are not adjusted for ties, so ties do not make a variable more important. Binary covariates are, of course, heavily tied.

Value
d A distance list for each pair within the caliper distance and constant constraint.

start The treated subject for each distance.

end The control subject for each distance.

References

Examples
data("nh0506Homocysteine")
attach(nh0506Homocysteine)
X<-cbind(female,age,black,education,poverty,bmi)
p<-glm(z~female+age+black+education+poverty+bmi,family=binomial)$fitted.values
d<-cbind(nh0506Homocysteine,p)
detach(nh0506Homocysteine)

#apply symmetric caliper 0.15 on propensity score
dist1<-maha_sparse(d$z,X,p,0.15)
length(dist1$d)

#apply asymmetric caliper c(-0.2,0.1) on propensity score
dist2<-maha_sparse(d$z,X,p,c(-0.2,0.1))
length(dist2$d)

match

Minimum-distance near-fine matching.

Description

The program finds an optimal near-fine match with a given caliper on p, plus directional penalties on dx to offset the distribution imbalances. That is, it finds a near-fine match that minimizes the penalized Mahalanobis distance. In order to avoid the distortion of the original distribution by large penalties, it has the option of apply asymmetric calipers on those covariates.

Usage

match(z, dist, dat, p = rep(1, length(z)), exact = NULL, fine = rep(1, length(z)), ncontrol = 1, penalty = round(max(dist$d)*1000), s.cost = 100, subX = NULL)

Arguments

z A vector whose ith coordinate is 1 for a treated unit and is 0 for a control.
dist A distance object with three components: d, start, end, typically created by maha_dense or maha_sparse. d[i] gives the distance between the (start[i])th treated and the (end[i]-sum(z))th control.
dat A data frame with length(z) rows. If the match is feasible, the matched portion of dat is returned with additional columns that define the match.
p A vector of length(z)=length(p) giving the variable used to define the caliper. Typically, p is the propensity score or its rank. If the dense match is performed, use the default p=rep(1,length(z)).
exact If not NULL, then a vector of length(z)=length(p) giving variable that need to be exactly matched.
fine A vector of with length(z)=length(fine) giving variable that need to be nearly-finely balanced.
ncontrol A positive integer giving the number of controls to be matched to each treated subject. If ncontrol is too large, the match will be infeasible.
penalty A numeric penalty imposed for each violation of fine balance.
s.cost The scaling factor for cost of the each pair of treated and control while rounding the cost.
subX If a subset matching is required, the variable that the subset matching is based on. That is, for each level of subX, extra treated will be discarded in order to have the number of matched treated subjects being the minimum size of treated and control groups. If exact matching on a variable x is desired and discarding extra treated is fine if there are more treated than controls for a certain level k, set exact=x, subX=x.

Details

The match minimizes the total distance between treated subjects and their matched controls subject to a near-fine balance constraint imposed as a penalty on imbalances. Another set of directional penalties on dx can be imposed in order to offset the distribution imbalances. In order to avoid the case of matching far pairs to get close means, the user can the option of apply asymmetric calipers on covariates in dx. We add a larger penalty for pairs outside the asymmetric caliper to avoid infeasibility issue. But a match may be infeasible if the caliper on p is too small. In this case, increase the caliper, or find the smallest caliper that gives a feasible matching by using optcal() in package 'bigmatch'.

For discussion of networks for fine-balance, see Rosenbaum (1989, Section 3) and Rosenbaum (2010). For near-fine balannce balance, see Yang et al. (2012).

You MUST install and load the optmatch package to use nearfine.

Value

If the match is infeasible, a warning is issued. Otherwise, a list of results is returned.

A match may be infeasible if the caliper or constant on p is too small, or ncontrol is too large, or if exact matching for exact is impossible.

feasible Indicator of whether matching is feasible or not.

data The matched sample, selected rows of dat.

x A vector of indicators of whether each treated-control pair is included in the matched sample.

References


Examples

```r
# To run this example, you must load the optmatch package.

data("nh0506Homocysteine")
attach(nh0506Homocysteine)
X<-cbind(female,age,black,education,povertyr,bmi)
p<-glm(z~female+age+black+education+povertyr+bmi,family=binomial)$fitted.values
d<-cbind(nh0506Homocysteine,p)
detach(nh0506Homocysteine)
dist<-maha_dense(d$z,X)
dist$d<-dist$d+1000*as.numeric(dist$d>7)
dist<-addcaliper(dist, d$z, d$p, c(-.5,.15), stdev=TRUE, penalty=1000)
dist<-addMagnitudePenalty(dist, d$z, d$p, positive=TRUE, multiplier=20)
dist<-addDirectPenalty(dist, d$z, d$p, positive=TRUE, penalty=1)
dist<-addDirectPenalty(dist, d$z, d$black, positive=FALSE, penalty=2)
dist<-addDirectPenalty(dist, d$z, d$bmi, positive=FALSE, penalty=2)
dist<-addDirectPenalty(dist, d$z, d$female, positive=FALSE, penalty=4)
o<-match(d$z, dist, d, fine=d$education, ncontrol=2)
md<-o$data
head(md)
```

---

**net**

*Optimal near-fine match from a distance matrix.*

### Description

The function creates the network for optimal near-fine matching to be passed via callrelax to the Fortran code for Bertsekas and Tseng’s (1988) Relax IV.

Of limited interest to most users; function netfine() would typically be called by some other functions.

### Usage

```r
net(z, dist, ncontrol=1, fine=rep(1,length(z)),
penalty=round(max(dist$d)*100), s.cost=100, subX=NULL)
```

### Arguments

- **z**: A vector whose $i$th coordinate is 1 for a treated unit and is 0 for a control.
- **dist**: A distance list with the starting node (treated subject), ending node (control), the extra distance between them based on directional penalty.
- **ncontrol**: A positive integer giving the number of controls to be matched to each treated subject.
- **fine**: A vector of with length(z)=length(fine) giving the nominal levels that are to be nearly-finely balanced.
- **penalty**: A numeric penalty imposed for each violation of fine balance.
s.cost
The scaling factor for cost of the each pair of treated and control while rounding the cost.

subX
If a subset matching is required, the variable that the subset matching is based on. That is, for each level of subX, extra treated will be discarded in order to have the number of matched treated subjects being the minimum size of treated and control groups. If exact matching on a variable x is desired and discarding extra treated is fine if there are more treated than controls for a certain level k, set exact=x, subX=x.

Details
The network contains a bipartite graph for treated and control subjects plus additional nodes for fine balance categories, plus additional nodes accept needed deviations from fine balance yielding near-fine balance.

For discussion of fine-balance, see Rosenbaum (1989, Section 3) and Rosenbaum (2010). For near-fine balance balance, see Yang et al. (2012).

Value
A network for optimal near-fine matching.

References


---

**nh0506Homocysteine**

**Homocysteine and Smoking**

**Description**
NHANES 2005-2006 data on smoking and homocysteine levels in adults.

**Usage**
data("nh0506Homocysteine")
Format

A data frame with 2759 observations on the following 11 variables.

x  Row number, 1 to 2759
SEQN  NHANES identification number
z  Smoking status, 1 = daily smoker, 0 = never smoker
female  1 = female, 0 = male
age  Age in years, >=20, capped at 85
black  1=black race, 0=other
education  Level of education
povertyr  Ratio of family income to the poverty level, capped at 5 times poverty
bmi  BMI or body-mass-index
cigsperday30  Cigarettes smoked per day, 0 for never smokers
cotinine  Blood cotinine level, a biomarker of recent exposure to tobacco
homocysteine  Level of homocysteine

Details

The following code constructed the data as used here. Attention is confined to adults, excluding children. Also, people who have smoked in the past, but do not now smoke daily, are excluded. A moderate number of individuals with missing povertyr, cotinine or homocysteine were excluded.

library(foreign) DEMO<-read.xport("DEMO_D.XPT") HCY<-read.xport("HCY_D.XPT") SMQ<-read.xport("SMQ_D.XPT") BMX<-read.xport("BMX_D.XPT") COT<-read.xport("COT_D.XPT")
d<-merge(DEMO,HCY,by="SEQN",all.x=TRUE) d<-merge(d,SMQ,by="SEQN",all.x=TRUE) d<-merge(d,BMX,by="SEQN",all.x=TRUE) rm(DEMO,HCY,SMQ,COT,BMX)
d$SEQN age<-$d$RIDAGEYR race<-$d$RIDRETH1 black<-$1*(race==4) hispanic<-$1*((race==1)|(race==2))
female<-$1*(d$RIAGENDR==2) education<-$d$DMDEDUC2 education[education>6]<-NA povertyr<-$d$INDFMPIR homocysteine<-$d$LBXHCY bmi<-$d$BMXBMI cotinine<-$d$LBXCOT cigs100life<-$d$SMQ020 cigs100life[cigs100life>3]<-NA cigs100life<-$cigs100life==1)*1 smokenow<-$1*(d$SMQ040<2.5)
smokenow[cigs100life==0]<-$0 dsM641 cigsdays30<-$d$SSMD641 cigsdays30[cigsdays30>32]<-NA cigsdays30[smokenow==0]<-$cigsdays30<-$0 cigsperday30<-$d$SSMD650 cigsperday30[cigsdperday30>100]<-$NA cigsperday30[smokenow==0]<-$cigs100life<-$0 dailysmoker<-$1*(cigs100life==1)&($cigsdays30==30)&($smokenow==1)&($cigsdays30>32)) neversmoker<-$1*((cigs100life==0)&($smokenow==0)) z<-$dailysmoker z[(neversmoker==0)&($daily smoker==0)]]<-(-999) ds<-$data.frame(SEQN,female,age,black,education,povertyr,bmi, homocysteine,cotinine) daily smoker,neversmoker,z] use<-$age==20 ds1<-$use] use<-$complete.cases(ds1) ds1$z[ds1$z==(-999]<-$NA ds2<-$ds1[use&ls.na(ds1$z),] rm(SEQN,female,age,black,education,povertyr, homocysteine,cotinine, cigs100life,smokewon,cigsdays30,cigdsperday30, dailysmoker,neversmoker,z,use,bmi)
ds2<-$ds2[order(1-ds2$z),] attach(ds2) nh0506Homocysteine<-$data.frame(SEQN,z,female,age,black,education,povertyr,bmi,write.csv(nh0506Homocysteine,"nh0506Homocysteine.csv")

Source

From the NHANES web page, for NHANES 2005-2006
References


Examples

```r
data(nh0506Homocysteine)
summary(nh0506Homocysteine)
```
Index

*Topic datasets
  nh0506Homocysteine, 14

addcaliper, 2
addDirectPenalty, 3
addMagnitudePenalty, 4

check, 5

edgenum, 6

maha_dense, 8
maha_sparse, 9
match, 11

net, 13

nh0506Homocysteine, 14