Package ‘clusterSEs’

October 12, 2022

Title Calculate Cluster-Robust p-Values and Confidence Intervals

Version 2.6.5

Description Calculate p-values and confidence intervals using cluster-adjusted t-statistics (based on Ibragimov and Muller (2010) <DOI:10.1198/jbes.2009.08046>, pairs cluster bootstrapped t-statistics, and wild cluster bootstrapped t-statistics (the latter two techniques based on Cameron, Gelbach, and Miller (2008) <DOI:10.1162/rest.90.3.414>. Procedures are included for use with GLM, ivreg, plm (pooling or fixed effects), and mlogit models.

Depends R (>= 3.3.0), AER, Formula, plm, stats

Imports sandwich, lmtest, mlogit (>= 1.1-0), utils, dfidx

License GPL (>= 2)

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Justin Esarey [aut, cre]

Maintainer Justin Esarey <justin@justinesarey.com>

Repository CRAN

Date/Publication 2021-04-05 20:10:02 UTC

R topics documented:

`cluster.bs.glm` *Pairs Cluster Bootstrapped p-Values For GLM*

Description

This software estimates p-values using pairs cluster bootstrapped t-statistics for GLM models (Cameron, Gelbach, and Miller 2008). The data set is repeatedly re-sampled by cluster, a model is estimated, and inference is based on the sampling distribution of the pivotal (t) statistic.

Usage

```
cluster.bs.glm(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  boot.reps = 1000,
  stratify = FALSE,
  cluster.se = TRUE,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
  seed = NULL
)
```

Arguments

`mod`	A model estimated using `glm`.
`dat`	The data set used to estimate `mod`.
`cluster`	A formula of the clustering variable.
`ci.level`	What confidence level should CIs reflect?
`boot.reps`	The number of bootstrap samples to draw.
`stratify`	Sample clusters only (= FALSE) or clusters and observations by cluster (= TRUE).
`cluster.se`	Use clustered standard errors (= TRUE) or ordinary SEs (= FALSE) for bootstrap replicates.
`report`	Should a table of results be printed to the console?
`prog.bar`	Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
`output.replicates`	Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)?
`seed`	Random number seed for replicability (default is NULL).

Value

A list with the elements

`p.values` A matrix of the estimated p-values.

`ci` A matrix of confidence intervals.

`replicates` Optional: A matrix of the coefficient estimates from each cluster bootstrap replicate.

Note

Code to estimate GLM clustered standard errors by Mahmood Arai: http://thetarzan.wordpress.com/2011/06/11/clustered-standard-errors-in-r/. Cluster SE degrees of freedom correction = (M/(M-1)) with M = the number of clusters.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

#####################################################################
# example one: predict whether respondent has a university degree
#####################################################################
require(effects)
data(WVS)
logit.model <- glm(degree ~ religion + gender + age, data=WVS, family=binomial(link="logit"))
summary(logit.model)

# compute pairs cluster bootstrapped p-values
clust.bs.p <- cluster.bs.glm(logit.model, WVS, ~ country, report = T)


#######################################
# example two: predict chicken weight
#######################################
rm(list=ls())
data(ChickWeight)

dum <- model.matrix(~ ChickWeight$Diet)
```

```
ChickWeight$Diet2 <- as.numeric(dum[,2])
ChickWeight$Diet3 <- as.numeric(dum[,3])
ChickWeight$Diet4 <- as.numeric(dum[,4])

weight.mod2 <- glm(formula = weight~Diet2+Diet3+Diet4+log(Time+1),data=ChickWeight)

# compute pairs cluster bootstrapped p-values
clust.bs.w <- cluster.bs.glm(weight.mod2, ChickWeight, ~ Chick, report = T)


####################################################################
# example three: murder rate by U.S. state, with interaction term
####################################################################
rm(list=ls())
require(datasets)

state.x77.dat <- data.frame(state.x77)
state.x77.dat$Region <- state.region
state.x77.dat$IncomeXHS <- state.x77.dat$Income * state.x77.dat$HS.Grad
income.mod <- glm( Murder ~ Income + HS.Grad + IncomeXHS , data=state.x77.dat)

# compute pairs cluster bootstrapped p-values
clust.bs.inc <- cluster.bs.glm(income.mod, state.x77.dat, ~ Region,
                               report = T, output.replicates=T, boot.reps=10000)

# compute effect of income on murder rate, by percentage of HS graduates
# using conventional standard errors
HS.grad.vec <- seq(from=38, to=67, by=1)
me.income <- coefficients(income.mod)[2] + coefficients(income.mod)[4]*HS.grad.vec
plot(me.income ~ HS.grad.vec, type="l", ylim=c(-0.0125, 0.0125),
     xlab="% HS graduates", ylab="ME of income on murder rate")
se.income <- sqrt( vcov(income.mod)[2,2] + vcov(income.mod)[4,4]*(HS.grad.vec)^2 +
                   2*vcov(income.mod)[2,4]*HS.grad.vec )
ci.h <- me.income + qt(0.975, lower.tail=T, df=46) * se.income
ci.l <- me.income - qt(0.975, lower.tail=T, df=46) * se.income
lines(ci.h ~ HS.grad.vec, lty=2)
lines(ci.l ~ HS.grad.vec, lty=2)

# use pairs cluster bootstrap to compute CIs, including bootstrap bias-correction factor
# including bootstrap bias correction factor
# cluster on Region
################################################
# marginal effect replicates =
me.boot <- matrix(data = clust.bs.inc$replicates[,2], nrow=10000, ncol=30, byrow=F) +
           as.matrix(clust.bs.inc$replicates[,4]) %*% t(HS.grad.vec)
# compute bias-corrected MEs
me.income.bias.cor <- 2*me.income - apply(X=me.boot, FUN=mean, MARGIN=2)
# adjust bootstrap replicates for bias
me.boot.bias.cor <- me.boot + matrix(data = 2*(me.income -
                                     apply(X=me.boot, FUN=mean, MARGIN=2)),
                                     ncol=30, nrow=10000, byrow=T)
# compute pairs cluster bootstrap 95% CIs, including bias correction
me.boot.plot <- apply(X = me.boot.bias.cor, FUN=quantile, MARGIN=2, probs=c(0.025, 0.975))
```

```
# plot bootstrap bias-corrected marginal effects
lines(me.income.bias.cor ~ HS.grad.vec, lwd=2)
# plot 95% Cis
# a little lowess smoothing applied to compensate for discontinuities
# arising from shifting between replicates
lines(lowess(me.boot.plot[1,] ~ HS.grad.vec), lwd=2, lty=2)
lines(lowess(me.boot.plot[2,] ~ HS.grad.vec), lwd=2, lty=2)

# finishing touches to plot
legend(lty=c(1,2,1,2), lwd=c(1,1,2,2), "topleft",
       legend=c("Model Marginal Effect", "Conventional 95% CI",
                "BS Bias-Corrected Marginal Effect", "Cluster Bootstrap 95% CI"))


## End(Not run)
```

`cluster.bs.ivreg` *Pairs Cluster Bootstrapped p-Values For Regression With Instrumental Variables*

Description

This software estimates p-values using pairs cluster bootstrapped t-statistics for instrumental variables regression models (Cameron, Gelbach, and Miller 2008). The data set is repeatedly re-sampled by cluster, a model is estimated, and inference is based on the sampling distribution of the pivotal (t) statistic.

Usage

```
cluster.bs.ivreg(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  boot.reps = 1000,
  stratify = FALSE,
  cluster.se = TRUE,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
  seed = NULL
)
```

Arguments

`mod`	A model estimated using `ivreg`.
`dat`	The data set used to estimate `mod`.
`cluster`	A formula of the clustering variable.

`ci.level`	What confidence level should CIs reflect?
`boot.reps`	The number of bootstrap samples to draw.
`stratify`	Sample clusters only (= FALSE) or clusters and observations by cluster (= TRUE).
`cluster.se`	Use clustered standard errors (= TRUE) or ordinary SEs (= FALSE) for bootstrap replicates.
`report`	Should a table of results be printed to the console?
`prog.bar`	Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
`output.replicates`	Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)?
`seed`	Random number seed for replicability (default is NULL).

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals.
`replicates`	Optional: A matrix of the coefficient estimates from each cluster bootstrap replicate.

Note

Code to estimate clustered standard errors by Mahmood Arai: http://thetarzan.wordpress.com/2011/06/11/clustered-standard-errors-in-r/. Cluster SE degrees of freedom correction = (M/(M-1)) with M = the number of clusters.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

##############################################
# example one: predict cigarette consumption
```

```
##############################################
require(AER)
data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)
fm <- ivreg(log(packs) ~ log(rprice) + log(rincome) |
    log(rincome) + tdiff + I(tax/cpi), data = CigarettesSW)

# compute pairs cluster bootstrapped p-values
cluster.bs.c <- cluster.bs.ivreg(fm, dat = CigarettesSW, cluster = ~state, report = T)


################################################
# example two: pooled IV analysis of employment
################################################
require(plm)
require(AER)
data(EmplUK)
EmplUK$lag.wage <- lag(EmplUK$wage)
emp.iv <- ivreg(emp ~ wage + log(capital+1) | output + lag.wage + log(capital+1), data = EmplUK)

# compute cluster-adjusted p-values
cluster.bs.e <- cluster.bs.ivreg(mod = emp.iv, dat = EmplUK, cluster = ~firm)


## End(Not run)
```

`cluster.bs.mlogit` *Pairs Cluster Bootstrapped p-Values For mlogit*

Description

This software estimates p-values using pairs cluster bootstrapped t-statistics for multinomial logit models (Cameron, Gelbach, and Miller 2008). The data set is repeatedly re-sampled by cluster, a model is estimated, and inference is based on the sampling distribution of the pivotal (t) statistic.

Usage

```
cluster.bs.mlogit(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  boot.reps = 1000,
  cluster.se = TRUE,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
```

```
  seed = NULL
)
```

Arguments

- `mod` — A model estimated using `mlogit`.
- `dat` — The data set used to estimate `mod`.
- `cluster` — A formula of the clustering variable.
- `ci.level` — What confidence level should CIs reflect?
- `boot.reps` — The number of bootstrap samples to draw.
- `cluster.se` — Use clustered standard errors (= TRUE) or ordinary SEs (= FALSE) for bootstrap replicates.
- `report` — Should a table of results be printed to the console?
- `prog.bar` — Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
- `output.replicates` — Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)?
- `seed` — Random number seed for replicability (default is NULL).

Value

A list with the elements

- `p.values` — A matrix of the estimated p-values.
- `ci` — A matrix of confidence intervals.

Note

Code to estimate GLM clustered standard errors by Mahmood Arai: http://thetarzan.wordpress.com/2011/06/11/clustered-standard-errors-in-r/, although modified slightly to work for `mlogit` models. Cluster SE degrees of freedom correction = (M/(M-1)) with M = the number of clusters.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

#######################################
# example one: train ticket selection
#######################################
require(mlogit)
data("Train", package="mlogit")
Train$choiceid <- 1:nrow(Train)

Tr <- dfidx(Train, shape = "wide", varying = 4:11, sep = "_",
          choice = "choice", idx = list(c("choiceid", "id")),
          idnames = c(NA, "alt"))
Tr$price <- Tr$price/100 * 2.20371
Tr$time <- Tr$time/60

ml.Train <- mlogit(choice ~ price + time + change + comfort | -1, Tr)

# compute pairs cluster bootstrapped p-values
# note: few reps to speed up example
cluster.bs.tr <- cluster.bs.mlogit(ml.Train, Tr, ~ id, boot.reps=100)


##################################################################
# example two: predict type of heating system installed in house
##################################################################
require(mlogit)
data("Heating", package = "mlogit")
H <- Heating
H$region <- as.numeric(H$region)
H.ml <- dfidx(H, shape="wide", choice="depvar", varying=c(3:12),
         idx = list(c("idcase", "region")))
m <- mlogit(depvar~ic+oc, H.ml)

# compute pairs cluster bootstrapped p-values
cluster.bs.h <- cluster.bs.mlogit(m, H.ml, ~ region, boot.reps=1000)


## End(Not run)
```

`cluster.bs.plm` *Pairs Cluster Bootstrapped p-Values For PLM*

Description

This software estimates p-values using pairs cluster bootstrapped t-statistics for fixed effects panel linear models (Cameron, Gelbach, and Miller 2008). The data set is repeatedly re-sampled by cluster, a model is estimated, and inference is based on the sampling distribution of the pivotal (t) statistic.

Usage

```
cluster.bs.plm(
  mod,
  dat,
  cluster = "group",
  ci.level = 0.95,
  boot.reps = 1000,
  cluster.se = TRUE,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
  seed = NULL
)
```

Arguments

`mod`	A "within" model estimated using `plm`.
`dat`	The data set used to estimate `mod`.
`cluster`	Clustering dimension ("group", the default, or "time").
`ci.level`	What confidence level should CIs reflect?
`boot.reps`	The number of bootstrap samples to draw.
`cluster.se`	Use clustered standard errors (= TRUE) or ordinary SEs (= FALSE) for bootstrap replicates.
`report`	Should a table of results be printed to the console?
`prog.bar`	Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
`output.replicates`	Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)?
`seed`	Random number seed for replicability (default is NULL).

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

# predict employment levels, cluster on group
require(plm)
data(EmplUK)

emp.1 <- plm(emp ~ wage + log(capital+1), data = EmplUK,
             model = "within", index=c("firm", "year"))
cluster.bs.plm(mod=emp.1, dat=EmplUK, cluster="group", ci.level = 0.95,
          boot.reps = 1000, cluster.se = TRUE, report = TRUE,
          prog.bar = TRUE)

# cluster on time

cluster.bs.plm(mod=emp.1, dat=EmplUK, cluster="time", ci.level = 0.95,
            boot.reps = 1000, cluster.se = TRUE, report = TRUE,
            prog.bar = TRUE)


## End(Not run)
```

`cluster.im.glm` *Cluster-Adjusted Confidence Intervals And p-Values For GLM*

Description

Computes p-values and confidence intervals for GLM models based on cluster-specific model estimation (Ibragimov and Muller 2010). A separate model is estimated in each cluster, and then p-values and confidence intervals are computed based on a t/normal distribution of the cluster-specific estimates.

Usage

```
cluster.im.glm(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  report = TRUE,
  drop = FALSE,
  truncate = FALSE,
  return.vcv = FALSE
)
```

Arguments

mod	A model estimated using glm.
dat	The data set used to estimate mod.
cluster	A formula of the clustering variable.
ci.level	What confidence level should CIs reflect?
report	Should a table of results be printed to the console?
drop	Should clusters within which a model cannot be estimated be dropped?
truncate	Should outlying cluster-specific beta estimates be excluded?
return.vcv	Should a VCV matrix and the means of cluster-specific coefficient estimates be returned?

Value

A list with the elements

p.values	A matrix of the estimated p-values.
ci	A matrix of confidence intervals.
vcv.hat	Optional: A cluster-level variance-covariance matrix for coefficient estimates.
beta.bar	Optional: A vector of means for cluster-specific coefficient estimates.

Note

Confidence intervals are centered on the cluster averaged estimate, which can diverge from original model estimates under several circumstances (e.g., if clusters have different numbers of observations). Consequently, confidence intervals may not be centered on original model estimates. If drop = TRUE, any cluster for which all coefficients cannot be estimated will be automatically dropped from the analysis. If truncate = TRUE, any cluster for which any coefficient is more than 6 times the interquartile range from the cross-cluster mean will also be dropped as an outlier.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Ibragimov, Rustam, and Ulrich K. Muller. 2010. "t-Statistic Based Correlation and Heterogeneity Robust Inference." *Journal of Business & Economic Statistics* 28(4): 453-468. <DOI:10.1198/jbes.2009.08046>.

Examples

```
## Not run:

######################################################################
# example one: predict whether respondent has a university degree
######################################################################

require(effects)
data(WVS)
logit.model <- glm(degree ~ religion + gender + age, data=WVS, family=binomial(link="logit"))
summary(logit.model)

# compute cluster-adjusted p-values
clust.im.p <- cluster.im.glm(logit.model, WVS, ~ country, report = T)

#############################################################################
# example two: linear model of whether respondent has a university degree
#              with interaction between gender and age + country FEs
#############################################################################

WVS$degree.n <- as.numeric(WVS$degree)
WVS$gender.n <- as.numeric(WVS$gender)
WVS$genderXage <- WVS$gender.n * WVS$age
lin.model <- glm(degree.n ~ gender.n + age + genderXage + religion + as.factor(country), data=WVS)

# compute marginal effect of male gender on probability of obtaining a university degree
# using conventional standard errors
age.vec <- seq(from=18, to=90, by=1)
me.age <- coefficients(lin.model)[2] + coefficients(lin.model)[4]*age.vec
plot(me.age ~ age.vec, type="l", ylim=c(-0.1, 0.1), xlab="age",
     ylab="ME of male gender on Pr(university degree)")
se.age <- sqrt( vcov(lin.model)[2,2] + vcov(lin.model)[4,4]*(age.vec)^2 +
                2*vcov(lin.model)[2,4]*age.vec)
ci.h <- me.age + qt(0.975, lower.tail=T, df=lin.model$df.residual) * se.age
ci.l <- me.age - qt(0.975, lower.tail=T, df=lin.model$df.residual) * se.age
lines(ci.h ~ age.vec, lty=2)
lines(ci.l ~ age.vec, lty=2)


# cluster on country, compute CIs for marginal effect of gender on degree attainment
# drop the FEs (absorbed into cluster-level coefficients)
lin.model.n <- glm(degree.n ~ gender.n + age + genderXage + religion, data=WVS)
clust.im.result <- cluster.im.glm(lin.model.n, WVS, ~ country, report = T, return.vcv = T)
# compute ME using average of cluster-level estimates (CIs center on this)
me.age.im <- clust.im.result$beta.bar[2] + clust.im.result$beta.bar[4]*age.vec
se.age.im <- sqrt( clust.im.result$vcv[2,2] + clust.im.result$vcv[4,4]*(age.vec)^2 +
                   2*clust.im.result$vcv[2,4]*age.vec)
# center the CIs on the ME using average of cluster-level estimates
# important: divide by sqrt(G) to convert SE of cluster-level estimates
#            into SE of the mean, where G = number of clusters
G <- length(unique(WVS$country))
ci.h.im <- me.age.im + qt(0.975, lower.tail=T, df=(G-1)) * se.age.im/sqrt(G)
```

```
ci.l.im <- me.age.im - qt(0.975, lower.tail=T, df=(G-1)) * se.age.im/sqrt(G)
plot(me.age.im ~ age.vec, type="l", ylim=c(-0.2, 0.2), xlab="age",
     ylab="ME of male gender on Pr(university degree)")
lines(ci.h.im ~ age.vec, lty=2)
lines(ci.l.im ~ age.vec, lty=2)
# for comparison, here's the ME estimate and CIs from the baseline model
lines(me.age ~ age.vec, lty=1, col="gray")
lines(ci.h ~ age.vec, lty=3, col="gray")
lines(ci.l ~ age.vec, lty=3, col="gray")


## End(Not run)
```

`cluster.im.ivreg` *Cluster-Adjusted Confidence Intervals And p-Values For GLM*

Description

Computes p-values and confidence intervals for GLM models based on cluster-specific model estimation (Ibragimov and Muller 2010). A separate model is estimated in each cluster, and then p-values and confidence intervals are computed based on a t/normal distribution of the cluster-specific estimates.

Usage

```
cluster.im.ivreg(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  report = TRUE,
  drop = FALSE,
  return.vcv = FALSE
)
```

Arguments

`mod`	A model estimated using `ivreg`.
`dat`	The data set used to estimate `mod`.
`cluster`	A formula of the clustering variable.
`ci.level`	What confidence level should CIs reflect?
`report`	Should a table of results be printed to the console?
`drop`	Should clusters within which a model cannot be estimated be dropped?
`return.vcv`	Should a VCV matrix and the means of cluster-specific coefficient estimates be returned?

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals.

Note

Confidence intervals are centered on the cluster averaged estimate, which can diverge from original model estimates under several circumstances (e.g., if clusters have different numbers of observations). Consequently, confidence intervals may not be centered on original model estimates. If drop = TRUE, any cluster for which all coefficients cannot be estimated will be automatically dropped from the analysis.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Ibragimov, Rustam, and Ulrich K. Muller. 2010. "t-Statistic Based Correlation and Heterogeneity Robust Inference." *Journal of Business & Economic Statistics* 28(4): 453-468. <DOI:10.1198/jbes.2009.08046>.

Examples

```
## Not run:

# example: pooled IV analysis of employment
require(plm)
require(AER)
data(EmplUK)
EmplUK$lag.wage <- lag(EmplUK$wage)
emp.iv <- ivreg(emp ~ wage + log(capital+1) | output + lag.wage + log(capital+1), data = EmplUK)

# compute cluster-adjusted p-values
cluster.im.e <- cluster.im.ivreg(mod=emp.iv, dat=EmplUK, cluster = ~firm)


## End(Not run)
```

`cluster.im.mlogit` *Cluster-Adjusted Confidence Intervals And p-Values For mlogit*

Description

Computes p-values and confidence intervals for multinomial logit models based on cluster-specific model estimation (Ibragimov and Muller 2010). A separate model is estimated in each cluster, and then p-values and confidence intervals are computed based on a t/normal distribution of the cluster-specific estimates.

Usage

```
cluster.im.mlogit(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  report = TRUE,
  truncate = FALSE,
  return.vcv = FALSE
)
```

Arguments

`mod`	A model estimated using `mlogit`.
`dat`	The data set used to estimate `mod`.
`cluster`	A formula of the clustering variable.
`ci.level`	What confidence level should CIs reflect?
`report`	Should a table of results be printed to the console?
`truncate`	Should outlying cluster-specific beta estimates be excluded?
`return.vcv`	Should a VCV matrix and the means of cluster-specific coefficient estimates be returned?

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals.

Note

Confidence intervals are centered on the cluster averaged estimate, which can diverge from original model estimates under several circumstances (e.g., if clusters have different numbers of observations). Consequently, confidence intervals may not be centered on original model estimates. Any cluster for which all coefficients cannot be estimated will be automatically dropped from the analysis. If truncate = TRUE, any cluster for which any coefficient is more than 6 times the interquartile range from the cross-cluster mean will also be dropped as an outlier.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Ibragimov, Rustam, and Ulrich K. Muller. 2010. "t-Statistic Based Correlation and Heterogeneity Robust Inference." *Journal of Business & Economic Statistics* 28(4): 453-468. <DOI:10.1198/jbes.2009.08046>.

Examples

```
## Not run:

# example: predict type of heating system installed in house
require(mlogit)
data("Heating", package = "mlogit")
H <- Heating
H$region <- as.numeric(H$region)
H.ml <- dfidx(H, shape="wide", choice="depvar", varying=c(3:12),
         idx = list(c("idcase", "region")))
m <- mlogit(depvar~ic+oc, H.ml)

# compute cluster-adjusted p-values
cluster.im.h <- cluster.im.mlogit(m, H.ml, ~ region)


## End(Not run)
```

`cluster.wild.glm` *Wild Cluster Bootstrapped p-Values For Linear Family GLM*

Description

This software estimates p-values using wild cluster bootstrapped t-statistics for linear family GLM models (Cameron, Gelbach, and Miller 2008). Residuals are repeatedly re-sampled by cluster to form a pseudo-dependent variable, a model is estimated for each re-sampled data set, and inference is based on the sampling distribution of the pivotal (t) statistic. Users may choose whether to impose the null hypothesis for independent variables; the null is never imposed for the intercept or any model that includes factor variables, interactions, or polynomials (although manually specified versions of these can circumvent the restriction). Confidence intervals are only reported when the null hypothesis is *not* imposed.

Usage

```
cluster.wild.glm(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  impose.null = TRUE,
  boot.reps = 1000,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
  seed = NULL
)
```

Arguments

`mod`	A linear (identity link) model estimated using `glm`.
`dat`	The data set used to estimate `mod`.
`cluster`	A formula of the clustering variable.
`ci.level`	What confidence level should CIs reflect? (Note: only reported when `impose.null == FALSE`).
`impose.null`	Should we impose the null Ho?
`boot.reps`	The number of bootstrap samples to draw.
`report`	Should a table of results be printed to the console?
`prog.bar`	Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
`output.replicates`	Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)? Only available when impose.null = FALSE.
`seed`	Random number seed for replicability (default is NULL).

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals (if null not imposed).

Note

Code to estimate GLM clustered standard errors by Mahmood Arai: http://thetarzan.wordpress.com/2011/06/11/clustered-standard-errors-in-r/. Cluster SE degrees of freedom correction = (M/(M-1)) with M = the number of clusters.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

#########################################
# example one: predict chicken weight
#########################################

# predict chick weight using diet, do not impose the null hypothesis
# because of factor variable "Diet"
data(ChickWeight)
weight.mod <- glm(formula = weight~Diet,data=ChickWeight)
cluster.wd.w.1 <-cluster.wild.glm(weight.mod, dat = ChickWeight,cluster = ~Chick, boot.reps = 1000)

# impose null
dum <- model.matrix(~ ChickWeight$Diet)
ChickWeight$Diet2 <- as.numeric(dum[,2])
ChickWeight$Diet3 <- as.numeric(dum[,3])
ChickWeight$Diet4 <- as.numeric(dum[,4])

weight.mod2 <- glm(formula = weight~Diet2+Diet3+Diet4,data=ChickWeight)
cluster.wd.w.2 <-cluster.wild.glm(weight.mod2, dat = ChickWeight,cluster = ~Chick, boot.reps = 1000)

############################################################################
# example two: linear model of whether respondent has a university degree
#              with interaction between gender and age + country FEs
############################################################################

require(effects)
data(WVS)

WVS$degree.n <- as.numeric(WVS$degree)
WVS$gender.n <- as.numeric(WVS$gender)
WVS$genderXage <- WVS$gender.n * WVS$age
lin.model <- glm(degree.n ~ gender.n + age + genderXage + religion, data=WVS)

# compute marginal effect of male gender on probability of obtaining a university degree
# using conventional standard errors
age.vec <- seq(from=18, to=90, by=1)
```

```
me.age <- coefficients(lin.model)[2] + coefficients(lin.model)[4]*age.vec
plot(me.age ~ age.vec, type="l", ylim=c(-0.1, 0.1), xlab="age",
     ylab="ME of male gender on Pr(university degree)")
se.age <- sqrt( vcov(lin.model)[2,2] + vcov(lin.model)[4,4]*(age.vec)^2 +
                2*vcov(lin.model)[2,4]*age.vec)
ci.h <- me.age + qt(0.975, lower.tail=T, df=lin.model$df.residual) * se.age
ci.l <- me.age - qt(0.975, lower.tail=T, df=lin.model$df.residual) * se.age
lines(ci.h ~ age.vec, lty=2)
lines(ci.l ~ age.vec, lty=2)


# cluster on country, compute CIs for marginal effect of gender on degree attainment
clust.wild.result <- cluster.wild.glm(lin.model, WVS, ~ country,
                                      impose.null = F, report = T,
                                      output.replicates=T)
replicates <- clust.wild.result$replicates
me.boot <- matrix(data=NA, nrow=dim(replicates)[1], ncol=length(age.vec))
for(i in 1:dim(replicates)[1]){
  me.boot[i,] <- replicates[i,"gender.n"] + replicates[i,"genderXage"]*age.vec
}
ci.wild <- apply(FUN=quantile, X=me.boot, MARGIN=2, probs=c(0.025, 0.975))

# a little lowess smoothing applied to compensate for discontinuities
# arising from shifting between replicates
lines(lowess(ci.wild[1,] ~ age.vec), lty=3)
lines(lowess(ci.wild[2,] ~ age.vec), lty=3)

# finishing touches to plot
legend(lty=c(1,2,3), "topleft",
       legend=c("Model Marginal Effect", "Conventional 95% CI",
                "Wild BS 95% CI"))


## End(Not run)
```

`cluster.wild.ivreg` *Wild Cluster Bootstrapped p-Values For For Regression With Instrumental Variables*

Description

This software estimates p-values using wild cluster bootstrapped t-statistics for instrumental variables regression models (Cameron, Gelbach, and Miller 2008). Residuals are repeatedly re-sampled by cluster to form a pseudo-dependent variable, a model is estimated for each re-sampled data set, and inference is based on the sampling distribution of the pivotal (t) statistic. Users may choose whether to impose the null hypothesis for independent variables; the null is never imposed for the intercept or any model that includes factor variables, interactions, or polynomials (although manually specified versions of these can circumvent the restriction). Confidence intervals are only reported when the null hypothesis is *not* imposed.

Usage

```
cluster.wild.ivreg(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  impose.null = TRUE,
  boot.reps = 1000,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
  seed = NULL
)
```

Arguments

`mod`	A linear (identity link) model estimated using `ivreg`.
`dat`	The data set used to estimate `mod`.
`cluster`	A formula of the clustering variable.
`ci.level`	What confidence level should CIs reflect? (Note: only reported when `impose.null == FALSE`).
`impose.null`	Should we impose the null Ho?
`boot.reps`	The number of bootstrap samples to draw.
`report`	Should a table of results be printed to the console?
`prog.bar`	Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
`output.replicates`	Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)? Only available when impose.null = FALSE.
`seed`	Random number seed for replicability (default is NULL).

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals (if null not imposed).

Note

Code to estimate clustered standard errors by Mahmood Arai: http://thetarzan.wordpress.com/2011/06/11/clustered-standard-errors-in-r/. Cluster SE degrees of freedom correction = (M/(M-1)) with M = the number of clusters.

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

##############################################
# example one: predict cigarette consumption
##############################################
require(AER)
data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)
fm <- ivreg(log(packs) ~ log(rprice) + log(rincome) |
    log(rincome) + tdiff + I(tax/cpi), data = CigarettesSW)

# compute cluster-adjusted p-values
cluster.wd.c <- cluster.wild.ivreg(fm, dat=CigarettesSW, cluster = ~state, report = T)


#################################################
# example two: pooled IV analysis of employment
#################################################
require(plm)
require(AER)
data(EmplUK)
EmplUK$lag.wage <- lag(EmplUK$wage)
emp.iv <- ivreg(emp ~ wage + log(capital+1) | output + lag.wage + log(capital+1), data = EmplUK)

# compute cluster-adjusted p-values
cluster.wd.e <- cluster.wild.ivreg(mod=emp.iv, dat=EmplUK, cluster = ~firm)


## End(Not run)
```

`cluster.wild.plm` *Wild Cluster Bootstrapped p-Values For PLM*

Description

This software estimates p-values using wild cluster bootstrapped t-statistics for fixed effects panel linear models (Cameron, Gelbach, and Miller 2008). Residuals are repeatedly re-sampled by cluster to form a pseudo-dependent variable, a model is estimated for each re-sampled data set, and inference is based on the sampling distribution of the pivotal (t) statistic. The null is never imposed for PLM models.

Usage

```
cluster.wild.plm(
  mod,
  dat,
  cluster,
  ci.level = 0.95,
  boot.reps = 1000,
  report = TRUE,
  prog.bar = TRUE,
  output.replicates = FALSE,
  seed = NULL
)
```

Arguments

`mod`	A "within" model estimated using `plm`.
`dat`	The data set used to estimate `mod`.
`cluster`	Clustering dimension ("group", the default, or "time").
`ci.level`	What confidence level should CIs reflect? (Note: only reported when `impose.null == FALSE`).
`boot.reps`	The number of bootstrap samples to draw.
`report`	Should a table of results be printed to the console?
`prog.bar`	Show a progress bar of the bootstrap (= TRUE) or not (= FALSE).
`output.replicates`	Should the cluster bootstrap coefficient replicates be output (= TRUE) or not (= FALSE)?
`seed`	Random number seed for replicability (default is NULL).

Value

A list with the elements

`p.values`	A matrix of the estimated p-values.
`ci`	A matrix of confidence intervals (if null not imposed).

Author(s)

Justin Esarey

References

Esarey, Justin, and Andrew Menger. 2017. "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods* forthcoming: 1-35. <URL:http://jee3.web.rice.edu/cluster-paper.pdf>.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90(3): 414-427. <DOI:10.1162/rest.90.3.414>.

Examples

```
## Not run:

# predict employment levels, cluster on group
require(plm)
data(EmplUK)

emp.1 <- plm(emp ~ wage + log(capital+1), data = EmplUK, model = "within",
             index=c("firm", "year"))
cluster.wild.plm(mod=emp.1, dat=EmplUK, cluster="group", ci.level = 0.95,
        boot.reps = 1000, report = TRUE, prog.bar = TRUE)

# cluster on time
cluster.wild.plm(mod=emp.1, dat=EmplUK, cluster="time", ci.level = 0.95,
            boot.reps = 1000, report = TRUE, prog.bar = TRUE)


## End(Not run)
```

Index