Stata & R Command Dictionary (Loosely Following Philipp Marek's STATA course, Univ. of Bremen, Summer 2015)

Torsten Heinrich*

August 12, 2015

Contents

*Some of the R code is from Claudius Gräbner. However, all remaining mistakes are mine.

1 Import dta File

R	Stata
`require(foreign)` `P<-read.dta("Session_1.dta")`	`use Session3.dta, clear`

2 Tabulate Properties

`table(P$bundesland)` `table(P$bundesland,P$gender)`	`tab bundesland` `tab bundesland gender`

3 Create Column/Property

`P$children_under14<-0`	`gen children_under14=0`

4 Rename Column

	`rename age_child8 age_child7`

5 Conditionally Modify Property

`P$children_under14 [P$age_child1<=14 & P$age_child1 >=0] <- 1`
`#or:`
`index<-P$age_child1<=14 & P$age_child1>=0`
`P$children_under14[index] <- P$children_under14 [ index]+1`
`#more complicated example:`
`for(x in names(P[6:12])){`
`P$children_under14[P[[x]]<=14&P[[x]]>=0]<-P$children_ under14[P[[x]]<=14&P[[x]]>=0]+1`
`}`
	`replace children_under14=children_under14+1 if age_child1<=14 & age_child1>0`
	`//more complicated example`
	`foreach x in age_child1 age_child2 age_child3 age_child4 age_child5 age_child6 age_child7 {`
	`replace children_under14=children_under14+1 if 'x' <=14 & 'x'>=0`
	`}`

6 Conditionally Drop Values

`P<-subset(P,P$hinc12>=0)`	`drop if hinc12<0`

7 Compute Global Variables, Apply Global Variables in Computations of Properties

```
P$ehs<-1+P$children_under14*.3+(P$hh_size-1-P$
    children_under14)*.5
index<-P$hinc12>=0
P$ehi[index]<-P$hinc12[index]/P$ehs[index]
index0<-Sess1$hinc12>=0
median(Sess1$ehi[index0])
P$povertyline[index0]<-median(P$ehi[index0])*.6
P$poor_hh<-0
index<-P$hinc12>=0 & P$ehi<P$povertyline
P$poor_hh<-0
index3<-P$hinc12>=0 & P$ehi<P$povertyline
P$poor_hh [ index3 ] <- 1
```

```
drop if hinc12<0
gen ehi=hinc12/equivalized_hhsize
egen median_ehi=median(ehi)
gen poverty=median_ehi*0.6
gen poor_hh=0
replace poor_hh=1 if ehi<poverty
```

8 Remove Outliers

```r
P$outlier<-0
P<-P[order(P$hinc12),]
H<-nrow(P)
P$h<-1:H
P$RP<-P$h/H
P$outlier[P$RP<0.025|P$RP>0.975]<-1
P<-subset(P,P$outlier==0)
#or:
size <- sum(!is.na(P$hinc12))
cutoffPercent <- 0.025
lowerLimit <- cutoffPercent * (size/100) + 1
upperLimit <- size - lowerLimit
P <- subset(P$hinc12, P$hinc12 > lowerLimit & P$
    hinc12 < upperLimit)
```

```stata
gen outlier=0
sort hinc12
egen H=sum(1)
gen h=1
replace h=h[_n-1]+1 in 2/l
gen RP=h/H
replace outlier=1 if RP<0.025 | RP>0.975
drop if outlier==1
```

9 Getting a Variable Type

```r
mode(P$bcbula)
typeof(P$bcbula)
levels(P$bcbula)
```

10 Working with Factor Level Variables as Integers

```
P$eastern <- factor(ifelse(as.numeric(P$bcbula %in%
   levels(P$bcbula)[15:19]) == 0, 1, 0))
#or:
P$eastern <- factor(ifelse(as.numeric(P$bcbula %in%
   levels(P$bcbula)[15:19]) == 0, 1, 0), labels = c("
   East exc Berlin", "West incl Berlin"))
```

```
gen East=0
replace East=1 if bcbula>=12 & bcbula <=16
```

11 t-Test

```
t.test(P$hinc12~P$eastern)
#or:
fit<-lm(P$hinc12~P$eastern)
summary(fit)
```

```
ttest hinc12, by Eastern
```

12 Discard First Dataset, Import Another File

```
#require(foreign)
P<-read.dta("Session3.dta")
```

```
use Session3.dta, clear
```

13 OLS Regression

```r
lm(P$hourly_wage~P$age+P$education)
#or:
fit<-lm(P$hourly_wage~P$age+P$education+P$bundesland)
      #... save as fit
summary(fit)
```

```
reg hourly_wage age education
```

14 OLS Regression with Dummies

```r
lm(P$hourly_wage~P$age+P$education+P$bundesland)
```

```
reg hourly_wage age education i.bundesland
```

15 Breusch-Pagan Test

```r
#install.packages("lmtest")              #to do BP
   Test, lmtest must be installed (once)
library(lmtest)                                  #...
   and loaded
bp<-bptest(fit,studentize=FALSE)
                          #Breusch-Pagan Test
bp
#result: BP = 94.8439, df = 2, p-value < 2.2e-16
```

```
reg hourly_wage age education
estat hettest
// result:
//Breusch-Pagan / Cook-Weisberg test for
   heteroskedasticity
//         Ho: Constant variance
//       Variables: fitted values of hourly_wage
//
//       chi2(1)      =    87.03
//       Prob > chi2  =   0.0000
```

16 Studentized Breusch-Pagan Test

```
#install.packages("lmtest")              #to do BP
   Test, lmtest must be installed (once)
library(lmtest)                                  #...
   and loaded
bp<-bptest(fit)                                  #
   Breusch-Pagan Test
bp
#result: BP = 14.8384, df = 2, p-value = 0.0005996
```

17 Breusch-Pagan Test Manually

```r
index2_2<-!is.na(P$hourly_wage) #index ... could also
    be given like so:
P$resid_age_edu[index2_2]<-resid(fit2)  #compute
   residuals where all fit variables are defined
n<-sum(!is.na(P$resid_age_edu)) #number of not NA
   observations
ee<-sum(P$resid_age_edu[index2_2]^2)    #sum(e^2)
P$g[index2_2]<-P$resid_age_edu[index2_2]^2/(ee/n)-1
       #g=e^2/(sum(e^2)/n)-1
g<-as.matrix(P$g[index2_2])             #extract x
   matrix from P data frame
Z0<-as.matrix(P$resid_age_edu[index2_2]*0+1)
Z1<-as.matrix(P$age[index2_2])  #Z matrix 1st col
Z2<-as.matrix(P$education[index2_2])    #Z matrix 2nd
    col
Z_W1=cbind(Z0,Z1,Z2)    #Z matrix
LM_W1<-0.5*(t(g)%*%Z_W1%*%solve(t(Z_W1)%*%Z_W1)%*%t(Z
   _W1)%*%g)  #LM statistic
LM_W1           #print LM statistic
pchisq(LM_W1,ncol(Z))   #chi squared CDF at LM_W1
   with ncol(Z)=3 df
```

```
reg hourly_wage age education
predict e, residual
predict xb

//gen h=1 if e!=0
//egen n=sum(h)
//This does not work (h=1958 instead of h=1104) which
    is the reason why the test resulted in a test
   statistic of 626.6830431
//perhaps like so:
gen h=1 if e>0 | e<0
egen n=sum(h)
//egen n=sum(1) //or gen n[_N]
egen ee=sum(e^2)
gen g=e^2/(ee/n)-1
drop if g==.
//prepare mata
putmata g, replace
putmata Z=(age education), replace
//mata has now n x 1 vector g and n x 2 matrix Z
mata  //call mata
LM=.5*(g'*Z*cholinv(Z'*Z)*Z'*g)
LM  //show test statistic
c=chi2(cols(Z),LM)
c //show resulting probability
end //exit mata
```

18 Wald Test

```r
library("aod")
fit_base<-lm(P$hourly_wage~P$age+P$education)
#coef(fit_base) #coefficients vector
#vcov(fit_base) #var-covar matrix
wald.test(b=coef(fit_base),Sigma=vcov(fit_base),Terms
    = 2:3,df=1104)  #gives Chi-sq and F statistic
```

```
reg hourly_wage age education
test age=education  //gives Chi-sq statistic
```

19 2 and 3 Variable Wald Tests

```r
fit_mar<-lm(P$hourly_wage~P$education+P$age+P$german+
   P$married+P$bundesland)
#Wald test 3-way
wald.test(b=coef(fit_mar),Sigma=vcov(fit_mar),Terms
   =3:5,df=1958)
#Wald test 2-way
wald.test(b=coef(fit_mar),Sigma=vcov(fit_mar),Terms
   =3:4,df=1958)
wald.test(b=coef(fit_mar),Sigma=vcov(fit_mar),Terms
   =4:5,df=1958)
wald.test(b=coef(fit_mar),Sigma=vcov(fit_mar),Terms=c
   (3,5),df=1958)
P$age30<-30*P$age
fit_wtt30<-lm(P$hourly_wage~P$age30+P$education)
wald.test(b=coef(fit_wtt30),Sigma=vcov(fit_wtt30),
   Terms = 2:3,df=1101)
```

```
test age german married
test age german
test german married
test age married
test 30*age=education
```

20 Export to Excel

	`reg hourly_wage age education`
	`estimates store baseline`
	`reg hourly_wage age education german`
	`estimates store german`
	`reg hourly_wage age education german i.bundesland`
	`estimates store bundesland`
	`reg hourly_wage age education german married i.bundesland`
	`estimates store married`
	`estimates table baseline german bundesland married`
	`//export to MS excel (outreg2 is not installed in this computer lab), hence first install`
	`ssc install outreg2`
	`//save excel sheed analysis.xml with this table`
	`outreg2[baseline german bundesland married] using analysis, replace excel`

21 Export to LaTeX

```
fit_base<-lm(P$hourly_wage~P$age+P$education)
fit_ger<-lm(P$hourly_wage~P$age+P$education+P$german)
fit_bun<-lm(P$hourly_wage~P$age+P$education+P$german+
   P$bundesland)
fit_mar<-lm(P$hourly_wage~P$age+P$education+P$german+
   P$married+P$bundesland)
#install.packages("rockchalk")
library("rockchalk")
outreg(list(fit_base,fit_ger,fit_bun,fit_mar))
```

22 Install Packages

```
install.packages("rockchalk")
```

```
ssc install outreg2
```

23 Logit Regression

```
flogit<-glm(participation~age+german+high_edu+high_
   voc+mid_voc,data=P,family="binomial")
summary(flogit)
```

```
logit participation age german high_edu high_voc
   mid_voc
```

24 Show Marginal Effects

```r
#install.packages("mfx")
library("mfx")
marginaleffects<-mfx
marginaleffects
#or:
logitmfx(participation~age+german+high_edu+high_voc+
    mid_voc,data=P)
```

```
mfx
//or:
//ssc install margeff //install package margeff (only
    once)
margeff
```

25 Show Clasification of Observations

```r
estat classification
```

26 ROC Curves

```r
#install.packages("ROCR")
library("ROCR")
fitpreds = predict(flogit,newdata=P,type="response")
fitpred = prediction(fitpreds,P$participation)
fitperf = performance(fitpred,"tpr","fpr")
plot(fitperf,col="green",lwd=2,main="ROC␣Curve␣for␣
    Loit")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
dev.off()
```

27 Create and Save Plots

```
#Show figure
plot(fitperf,col="green",lwd=2,main="ROC Curve for
    Loit")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
#Now save the same figure as pdf
pdf("ROCcurve.pdf", useDingbats=FALSE)
plot(fitperf,col="green",lwd=2,main="ROC Curve for
    Loit")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
dev.off()
```

28 Likelihood Ratio Test

```
flogit1<-glm(participation~age+german+high_edu+high_
    voc+mid_voc,data=P,family="binomial")
flogit2<-glm(participation~age+german,data=P,family="
    binomial")
lrtest(flogit1,flogit2)
```

```
logit participation age german high_edu high_voc
    mid_voc
estimates store unrestricted
logit participation age german
estimates store restricted
lrtest unrestricted restricted
```

29 Tobit Regression

```r
P$weekly_working_hours[P$weekly_working_hours<0]<-0
        #sanitize data
library(AER)
ftobit<-tobit(participation~age+german+high_edu+high_
    voc+mid_voc,data=P,left=0)
summary(ftobit)
```

```
replace weekly_working_hours=0 if
    weekly_working_hours<0
tobit weekly_working_hours age german high_edu
    high_voc mid_voc, ll(0)
```

30 Ordered Logit Regression

```r
#install.packages("MASS")
library("MASS")
P$newvar<-P$participation
index<-(P$weekly_working_hours<=0|P$weekly_working_
    hours>=35)&P$participation==1
P$newvar[index]<-2
polr(as.factor(newvar)~East+onechild+twochild+
    threechild+hh_inc+P$B5_CON+P$B5_OPE+B5_EXT+B5_AGR+
    P$B5_EMO_Stab,data=P)
library("mfx")
mfx
```

```
gen newvar=participation
replace newvar=2 if weekly_working_hours<=0 and
    participation=1
replace newvar=2 if weekly_working_hours>=35
ologit newvar East onechild twochild threechild
    hh_inc B5_OPE B5_EXT B5_AGR B5_EMO_Stab
mfx
```

31 Conditional Logit

```r
P<-read.dta("Session5cl.dta") #load new dataset
as.numeric(P$bundesland)
BLI<-read.dta("Regional/bl_info.dta")
P$BL_ID<-as.numeric(P$bundesland)-3
BLI$BL_ID<-as.numeric(as.factor(BLI$bundesland))
P$ID<-1:nrow(P)
nobs<-nrow(P)
P$ID2<-P$ID #create a new first column as the below
   expansion will insist on changing this column
P<-P[,c(11,1:10)]
P<-P[rep(seq_len(nrow(P)), each=16),] #expansion (
   repeat each observation 16 times)
P$alt<-rep(1:16,nobs) #create strata (the dummies for
    the bundesland one for each bundesland for each
   observation)
P$choice<-0 #create dummy for bundesland actually
   chosen
P$choice[P$BL_ID==P$alt]<-1
names(P)[names(P)=="migration"]<-"year" #rename
   migration year column
P<-merge(P,BLI,by=c("BL_ID","year"))  #merge BLI into
    dataset
P$GDP_PC<-P$GDP/P$bevoelkerung_bl
P<-P[with(P,order(ID,alt)),]  #order dataset
#... continued next page
```

```
use Session5/Session5cl.dta, clear
//ssc install sdecode
decode bundesland, gen(h)
gen hh=strpos(h,"]")
//replace bundesland=substr(bundesland,hh+2,.)
replace h=substr(h,hh+2,.)
replace hh=strpos(h,"]")
replace h=substr(h,hh+2,.)
drop hh
ren h bundeland
replace bundesland=subinstr(bundesland,"ue","ü")
gen i=[_n]
egen alt=group(bundesland)
egen hh=max(alt)
expand hh
save help.dta, replace
keep alt bundesland
duplicates drop bundesland, force
save help_bl.dta,replace
use help.dta,clear
ren bundesland BL
sort i
replace alt=1
replace alt=alt[_n-1]+1 if i==i[_n-1]
merge m:1 alt using help_bl.dta
sort i alt
//... continued next page
```

```r
#... continued from last page
#install.packages("survival")
library("survival")
clogit(choice~GDP_PC+bevoelkerung_bl+strata(alt),data
   =P)
#fails with:
####Error in fitter(X, Y, strats, offset, init,
   control, weights = weights,  :
####  NA/NaN/Inf in foreign function call (arg 5)
####In addition: Warning message:
####In fitter(X, Y, strats, offset, init, control,
   weights = weights,  :
####  Ran out of iterations and did not converge
```

```
//... continued from last page
gen choice=0
replace choice=1 if bundesland==BL
ren migration year
keep i alt choice year bundesland
merge m:1 bundesland year using Session5\Regional\
   bl_info.dta
keep if _merge==3
sort i alt
gen gdp_pc=GDP/bevoelkerung
asclogit choice gdp_pc bevoelkerung case(i) alt(alt)
asclogit choice gdp_pc bevoelkerung case(i) alt(alt)
   noconst
```

32 Merging Datasets

```r
P<-merge(P,BLI,by=c("BL_ID","year"))  #merge BLI into
    dataset
```

```
merge m:1 bundesland year using Session5\Regional\
   bl_info.dta
keep if _merge==3
```

33 Ordering Data

```r
P<-P[with(P,order(ID,alt)),]
```

```
sort i alt
```

34 Save Datasets

```r
save(P,file="Pfile.Rda")
##load this later with
#load("Pfile.Rda")
##or: save as stata native format (dta) (and load as
   shown above)
write.dta(write.dta(P, "Pfile.dta")
```

```
save help_bl.dta,replace
//load as (as also shown above)
use help.dta,clear
```

35 Multinomial Logit

```r
P<-read.dta("session5.dta")
library(nnet)
mfit1<-multinom(P$party_preferenc~P$hinc+P$East+P$
   high_edu+P$high_voc+P$mid_voc+P$age+P$female+P$
   Kinder)
summary(mfit1)
```

```
use Session5\Session5.dta ,clear
encode party, gen(party)
sum party*
mlogit party hinc East high_edu high_voc mid_voc age
   female Kinder
estimates store unrestricted  //store for LR test
   below
```

36 Wald Test for Multinomial Logit

```
#wald.test of library aod will not work on this as it
    does not expect multiple sets of coefficients etc
    (from the multiple regression...)
#... however according to http://www.ats.ucla.edu/
   stat/r/dae/mlogit.htm we can use the z statistic z
   =b/std(b) to evaluate the quality (according to
   the website, this is actually a Wald test - is it
   ?)
z <- summary(mfit1)$coefficients/summary(mfit1)$
   standard.errors
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
//Wald test
test age
```

37 Likelihood Ratio Test for 2 Multinomial Logit Models

```
mfit2<-multinom(P$party_preferenc~P$hinc+P$East+P$
   high_edu+P$high_voc+P$mid_voc+P$female+P$Kinder)
lrtest(mfit1,mfit2)
```

```
//LR test
mlogit party hinc East high_edu high_voc mid_voc
   female Kinder
estimates store restricted
lrtest restricted unrestricted
```

38 Poisson Regression

```
fit1<-glm(doctor_visits~private_insurance+education+
    chronical_complaints,family="poisson",data=P)
summary(fit1)
```

```
poisson doctor_visits private_insurance education
    chronical_complaints
```

39 Marginal Effects in Poisson Regression

```
library("mfx")
fit1_mfx<-poissonmfx(doctor_visits~private_insurance+
    education+chronical_complaints,data=P)
fit1_mfx
```

```
mfx
```

40 Negative Binomial Regression

```
library(MASS)
fit2_negbin<-glm.nb(doctor_visits~private_insurance+
    chronical_complaints+age+age2+german+stress+
    disabled+weekly_working_hours+high_edu+high_voc+
    mid_voc,data=P)
summary(fit2_negbin)
```

```
nbreg doctor_visits private_insurance
    chronical_complaints age age2 german stress
    disabled weekly_working_hours high_edu high_voc
    mid_voc
```

41 Predict Number of Events From Poisson Fit

```
P$lambda[!is.na(P$doctor_visits)]<-predict(fit1)
```

```
predict lambda, n
```

42 Test for Overdispersion

```
P$lambda[!is.na(P$doctor_visits)]<-predict(fit1)
P<-subset(P,!is.na(P$doctor_visits))
P$z=((P$doctor_visits-P$lambda)^2+P$doctor_visits)/P$
    lambda
summary(zfit<-lm(P$z~0+P$lambda))
```

```
predict lambda, n
//tab lambda
sum lambda
gen z=((doctor_visits-lambda)^2-doctor_visits)/lambda
reg z lambda, noconst noheader
```

43 Force (OLS) Regression Without Intercept

```
zfit<-lm(P$z~0+P$lambda)
```

```
reg z lambda, noconst noheader
```

44 Get Some Basic Descriptive Statistics

```
summary(P$doctor_visits)
```

```
tabstat doctor_visits, stat (mean sd median p75)
```

45 Get Exact Quantiles

```r
quantile(P$doctor_visits, probs=c(.05,.95))
```

46 Heckman Two Step Estimation

```r
P<-read.dta("Session3.dta")
P$lnwage<-log(P$hourly_wage)
#install.packages("sampleSelection")
library("sampleSelection")
heckfit<-heckit(outcome=lnwage~age+high_edu+high_voc+
    mid_voc+labor_experience+german+bundesland,
    selection=participation~age+high_edu+high_voc+mid_
    voc+onechild+twochild+threechild+el_ort+hh_inc+
    bundesland,data=P,method="2step")
summary(heckfit)
```

```
use Session5\Session3.dta, clear
gen lnwage=ln(hourly_wage)
heckman lnwage age high_edu high_voc mid_voc
    labor_experience german i.bundesland, select(
    participation=age high_edu high_voc mid_voc
    onechild twochild threechild el_ort hh_inc i.
    bundesland) twostep
```

47 Create Maps Using Shapefiles

```r
P<-read.dta("Regional/gdp_bundesland.dta")
P$region<-toupper(P$bundesland)
NN<-read.dta("Regional/nuts_name.dta")
P<-merge(P,NN,by="region")
P$NUTS_ID<-P$nuts
P<-subset(P,P$year==2009)
NC<-read.dta("Regional/nutsdatabase_cleaned.dta")
P<-merge(P,NC,by="NUTS_ID")
SHF<-read.dta("Regional/nutscoordinates.dta") #read
    shapefile
SHF_DF <- data.frame(SHF) #transform to data frame
SHF_DF$GDPpc <- P$GDPpc[match(SHF_DF$X_ID,P$id)]  #
    copy GDPpc into shape data for convenience
SHF_DF<-subset(SHF_DF,!is.na(SHF_DF$GDPpc)) #retain
    only regions with GDPpc data
#install.packages("mapproj")
#install.packages("ggplot2")
library(ggplot2)
ggplot(SHF_DF, aes(x=X_X, y=X_Y, group=X_ID, fill=
    GDPpc)) + geom_polygon(color='black') + coord_map
    () + theme_classic() + scale_fill_gradient(low = "
    #0000FF", high ="#FF0000") #plot with gradient
    between blue and red
#image is saved as Rplots.pdf
```

```
use "Session 6\Regional\gdp_bundesland.dta", clear
//prepare data
ren bundesland region
replace region=upper(region)
replace region=subinstr(region,"ü","Ü",.)
merge m:m region using "Session 6\Regional\nuts_name.
    dta"
keep if _merge==3
keep if year==2009
ren nuts NUTS_ID
drop _merge
merge 1:1 NUTS_ID using "Session 6\Regional\
    nutsdatabase_cleaned.dta"
keep if _merge==3
//create map
//ssc install spmap
spmap GDPpc using "Session 6\Regional\nutscoordinates
    .dta", id(id)
```

48 Fixed Effects Panel Regression

```r
P<-read.dta("Session6.dta")
#install.packages("plm")
#library("plm")
FE<-plm(P$hourly_wage ~ P$bilzeit + P$age + P$expft +
   P$exppt, index=c("persnr","year"), data=P, model="
   within")
summary(FE)
fixef(FE)                #print the fixed effects
```

```stata
use "Session 6\Session6.dta", clear
xtset persnr year
xtreg hourly_wage bilzeit age expft exppt, fe
estimates store FE
```

49 Random Effects Panel Regression

```r
RE<-plm(P$hourly_wage ~ P$bilzeit + P$age + P$expft +
   P$exppt, index=c("persnr","year"), data=P, model="
   random")
#or (gives same result):
panelset<-plm.data(P,index=c("persnr","year"))
also_RE<-plm(hourly_wage~bilzeit+age+expft+exppt,data
   =panelset,model="random")
summary(RE)
summary(also_RE)
```

```stata
xtreg hourly_wage bilzeit age expft exppt, re
estimates store RE
```

50 F-Test Against OLS for Fixed Effects Model

```
OLS=lm(hourly_wage~bilzeit+age+expft+exppt,data=P)
pFtest(FE,OLS)  #standard F-test against OLS, use FE
   if p<0.05
```

51 Hausman Test for Fixed vs. Random Effects Model

```
phtest(FE,RE)   #Hausman test FE agains RE, do not
   use RE if p<0.05
```

```
hausman FE RE, sigmamore
```

52 Arellano-Bond Approach

```
AB_GMM<-pgmm(hourly_wage ~ lag(hourly_wage,1) + age +
    expft + exppt | lag(hourly_wage, 2:9),data=
   panelset,effect="individual",transformation="d")
summary(AB_GMM)
```

```
//something with xtrbond??
```

53 Mundlak Approach

```r
BZ_MEAN<-aggregate(P$bilzeit,by=list(P$persnr),mean)
names(BZ_MEAN)<-c("persnr","bz_mean")
PM<-merge(panelset,BZ_MEAN,by=c("persnr"))
MRE<-plm(hourly_wage~bilzeit+age+expft+exppt+bz_mean,
    data=PM,model="random")
summary(MRE)
```

```
bysort persnr: egen bz_mean=mean(bilzeit)
xtreg hourly_wage bilzeit age expft exppt bz_mean, re
```