GEN BUS 806 R COMMANDS

The following list of commands and information intends to assist you in getting familiar with the commands used in R common to the panel data analysis in GEN BUS 806

Useful Websites

http://www.r-project.org/	CRAN - Comprehensive R Archive Network.
http://biostat.mc.vanderbilt.edu/twiki/bin/view/Main/RS	A R website maintained by Frank Harrel with Vanderbilt University.
http://www.ku.edu/~pauljohn/R/Rtips.html	A useful tip sheet for beginning users maintained by Paul Johnson from Kansas University.
http://www.stat.math.ethz.ch/R-manual/	Links to R-manual which contains references for all the R commands
http://statcomp.ats.ucla.edu/splus/default.htm	UCLA stat-comp portal.

Introduction

R is an independent, open-source and free implementation of the S language. One of the great strength of S and R is the ability to adopt to new statistical methodology through libraries, many of them prepared by experts in applied statistics. Throughout this course, a few of the libraries for mixed effects analysis will be used extensively, i.e. library(nlme), library(lm), etc. Another strength of R is the ease with which well-designed publication-quality plots can be produced. Commands in S or R are either expressions or assignments. The # symbol marks the rest of the line as comments.

R operates in a data objects environment. These objects can be vectors, lists, arrays and data frames. Objects can be referred or entered into a R command by their assigned names. The most commonly used is data frame which can be thought of as a list of variables of the same length, but possibly of difference types (numeric, character or logical).

R primarily uses the command line interface. Thus whenever you are unclear as to what a command can accomplish for you, type "?command" will display a help file for the specific command. For example, "?read.table" will display the details about the "read.table" command.

Basics	**Description**	**Example Source**	**Example**
c()	The concatenate function can be used to combine columns of variables in a data frame. Note that the example shows a common way of referring variables in a data frame.	Chap1AnalysisR.txt	divorce[, c("DIVORCE", "AFDC")]
as.data.frame()	This function converts the specific object to a data frame.	Chapter4LotteryExplorationR.txt	mzip=as.data.frame(t(sapply(split(lottery[, c("NRETAIL", "PERPERHH", ...)], lottery$ZIP), mean)))
sapply(x, FUN)	Apply functions of mean etc. to lists or vectors. Please also note similar functions lapply(), tapply(). FUN can be mean, min, max, sd, median, etc.	See above	See above
data frame name <- data frame name[order(data frame$variable),]	This command offers a way to sort the data frame by a variable in ascending order. For other options of ordering, use "?order".	Chapter11yogurtR.txt	yogurt<-yogurt[order(yogurt$occasion),]
subset(data frame, criteria)	Used to create a subset of a data frame that meet with certain criterion.	Chapter2AnalysisR.txt	Medicare2 = subset(Medicare, STATE != 54 | YEAR != 2)

Reading Data In	Description	Example Source	Example
data frame name <- read.table(choose.files(), sep ="\t", quote = "",header=TRUE)	This command allows reading a tab separated text file in a table format and creates a data frame from it in R, header=TRUE will keep the column names when reading the data in.	For GEN BUS 806, data will be read in this format in all the chapters.	divorce = read.table(choose.files(), sep ="\t", quote = "",header=TRUE)

attach(data frame)
detach(data frame)
	By attaching a data frame variables can be referred simply by its names, eg. YEAR, instead of as Medicare$YEAR. Also if a data frame is attached, a copy is used and any subsequent changes will not be reflected in the data frame. When a data frame is detached the copy is normally discarded, but any changes made will be saved unless the argument save=F is set.		

Summary Statistics			
names("data frame name")	Shows variable names of a data frame.	Chap1AnalysisR.txt	names(divorce)
str("data frame name")	Shows the structure of a data frame including number of observations and number of variables, variable names, format, etc. It offers a convenient way to check whether the data was imported properly.	Chap1AnalysisR.txt	str(divorce)
summary(data frame name$variable name) or summary(data frame name[, c("variable name 1", "variable name 2"…)])	summary provides statistics on minimum, maximum, 1st quartile, median, 3rd quantile, mean, and number of missing observations. The "$" operator extracts a column from a data frame. data frame name[, c("variable name 1", "variable name 2"…] extracts several columns out of a data frame.	Chap1AnalysisR.txt	summary(divorce[, c("DIVORCE", "AFDC")])
gsummary(data frame name[, c("variable name 1", "variable name 2", ...)], groups=data frame$grouping variable, FUN=mean)	gsummary provides mean, standard deviation, minimum, maximum summary statistics by a grouping variable. This command together with "groupedData" provide suitable ways to analyze multilevel data or longitudinal data.	Chap3AnalysisR.txt For groupedData example see Chap2AnalysisR.txt	gsummary(taxprep[, c("MS", "HH", "AGE", "EMP", "PREP")], groups=taxprep$TIME, FUN=mean)

sd(data frame name[,c("variable name", ...)], na.rm=TRUE)
var(data frame name[,c("variable name", ...)], na.rm=TRUE)
	Calculates standard deviation, or variance of variables, with missing values removed.	Chap1AnalysisR.txt	sd(divorce[,c("DIVORCE", "AFDC")], na.rm=TRUE)

cor(data frame$variable 1,data frame$variable 2, use="pairwise.complete.obs")	Calculates correlation using observations when pairs of variables' observations are complete.	Chap1AnalysisR.txt	cor(divorce$DIVORCE, divorce$AFDC, use="pairwise.complete.obs")
table(data frame$variable)	Creates a frequency table for binary variables.	Chap10AnalysisR.txt	table(tfiling$CAPS)
xtabs(~x1+x2, data=…)	Creates a cross - classifying frequency table for binary variables.	Chap9AnalysisR.txt	xtabs(~taxprep$PREP+taxprep$EMP, data=taxprep)

Create & Replace Variables	**Description**	**Example Source**	**Example**
In R creating a new object or replacing an old object are done in the same way.			
data frame$variable name<-expression of the variable	When an existing object or variable names are used on the left side of the expression, the content of the object will simply be written over by the new expression.	Chap2AnalysisR.txt	Medicare$NUM.DCHG=Medicare$NUM.DCHG/1000
data frame$variable name<-factor(data frame$variable name)	Creates a categorical variable out of an existing variable. In R using a factor indicates to many of the statistical functions that this is a categorical variable so it is treated specially.	Chap2AnalysisR.txt	Medicare$FSTATE = factor(Medicare$STATE)
Graphics			
lset(col.whitebg())	Set the background of the plot to be white.	Chap2AnalysisR.txt	lset(col.whitebg())
boxplot(y~x, ...)	Box plot of y vs. x	Chap2AnalysisR.txt	boxplot (CCPD ~ YEAR, xlab="YEAR", ylab="CCPD")
plot(y~x, ...)	Generally produces scatter plot. In panel data analysis this command can be used to do multiple times series plot.	Chap2AnalysisR.txt	plot(CCPD ~ YEAR, data = Medicare, xaxt="n", yaxt="n", ylab="", xlab="") for (i in Medicare$STATE) { lines(CCPD ~ YEAR, data = subset(Medicare, STATE == i)) }

plot(groupedData…)	A unique type of plot available in R is the trellis plot. It usually requires first grouping a data frame by a factor variable with different levels. "layout=c(18, 3)" controls number of columns and number of rows for the panels in the plot. The panels are ordered by increasing maximum response.	Chap2AnalysisR.txt	library(nlme)
			GrpMedicare = groupedData(CCPD ~ YEAR|FSTATE, data=Medicare)
			plot(GrpMedicare, xlab="YEAR", ylab="CCPD", scale = list(x=list(draw=FALSE)),layout=c(18,3))

One - Way Fixed Effects Model			
lm(y~x+factor variable for subject -1…, data=data frame name)	One way fixed effects model specifies different intercepts for different subjects. In R this accomplished by using a categorical variable for the subjects, which is the factorized subject variable.	Chap2AnalysisR.txt	Medicare.lm = lm(CCPD ~ NUM.DCHG + Yr31 + YEAR + AVE.DAYS + FSTATE - 1, data=Medicare2)
Fixed Effects Model with Autocorrelated Error			
gls(y~x+factor variable, data=data frame name, random~1|subject, correlation=corAR1(form=...))	Different from one -way fixed effects model, with AR1 autocorrelation, the GLS estimator is used.	Chap4LotteryInsampleR.txt	lme(LNSALES~MEDSCHYR+POPULATN, data=Lottery2, random=~1|ZIP, correlation=corAR1(form=~TIME|ZIP))

One - Way Random Effects Model	**Description**	**Example Source**	**Example**
lme(y~x, data=data frame name, random~1|subject)	lme() is the command for estimating random effects model. Two methods are allowed including "ML" and "REML". The default is "REML," the restricted maximum likelihood.	Chap3Analysis.do	lme(LNTAX~MS+HH+..., data=taxprep, random=~1|SUBJECT, method="ML")
Random Effects Model with Autocorrelated Error			
lme(y~x, data=data frame name, random~1|subject, correlation=corAR1(form=...))	lme() can also accommodate AR(1) correlation in the error component.	Chap4LotteryInsampleR.txt	lme(LNSALES~MEDSCHYR+POPULATN, data=Lottery2, random=~1|ZIP, correlation=corAR1(form=~TIME|ZIP))
Binary Dependent Variables			
glm(y~x..., binomial(link=logit), data=...)	Fits a homogeneous model.	Chap9AnalysisR.txt	glm(PREP~LNTPI+MR+EMP, binomial(link=logit), data=taxprep)
lrm(y~x+factor subject variable, data=...)	Fits a one way fixed effects logistic model.	Chap9AnalysisR.txt	lrm(PREP~LNTPI+MR+EMP+facsub, data=taxprep)
GLMM(y~x, random=~1|SUBJECT, family=binomial(link=logit), data=data frame name)	Fits a generalized linear mixed effects model via penalized likelihood.	Chap9AnalysisR.txt	GLMM(PREP~LNTPI+MR+EMP, random=~1|SUBJECT, family=binomial(link=logit), data=taxprep)
gee(y ~ x, id=SUBJECT, data=data frame name, family=binomial(link=logit), corstr="exchangeable")	Marginal models and GEE.	Chap9AnalysisR.txt	gee(PREP ~ LNTPI+MR+EMP, id=SUBJECT, data=taxprep, family=binomial(link=logit), corstr="exchangeable")
Poisson Dependent Variables			
glm(y~x..., family=poisson(link=log), data=..)	Fits a homogeneous model.	Chap10AnalysisR.txt	tfilinghom<-glm(NUMFILE ~ POPLAWYR+..., data=tfiling, family=poisson(link="log"), offset=LNPOP)
glm(y~x+factor subject variable -1..., family=poisson(link=log), data=..)	Fits a one way fixed effects logistic model.	Chap10AnalysisR.txt	glm(NUMFILE ~STATEFAC+...-1, data=tfiling, family=poisson(link="log"), offset=LNPOP)
GLMM(y~x, random=~1|SUBJECT, family=poisson(link=log), data=data frame name)	Fits a generalized linear mixed effects model estimated via penalized likelihood.	Chap10AnalysisR.txt	GLMM(NUMFILE ~ offset(LNPOP)+POPLAWYR+..., random=~1|STATE, family=poisson(link=log),data=tfiling)

Poisson Dependent Variables	**Description**	**Example Source**	**Example**
gee(y ~ x, id=SUBJECT, data=data frame name, family=poisson(link=log), corstr="independence")	Marginal models and GEE.	Chap10AnalysisR.txt	gee(NUMFILE ~ offset(LNPOP)+POPLAWYR+..., id=STATE, data=tfiling, family=poisson(link="log"), corstr="independence")