EPIX ANALYTICS
Sound answers for complex decisions

Suggested material to review prior to attending EpiX Analytics' training courses

The following are basic topics and knowledge that we suggest that participants study prior to attending our courses. Attending our courses with this knowledge ensures that the training is focused on key risk analysis issues and is not overly hampered by teaching the basics of using Windows, R, or the fundamentals of probability theory. Participants intending to skip modules must *also* have the equivalent knowledge of the modules they are skipping.

General background

It is helpful if participants:

- Have looked at risk analyses that have been done in their field
- Have an idea of how risk analysis could be useful to their work
- Have some prepared problems, with data if possible, that we can solve as a class exercise (more relevant for advanced courses)
- Come with a positive attitude to learn, to try, to be prepared to make mistakes, and to have fun!

Windows

Creating, loading, copying, moving, renaming and saving files (and directories where applicable)
Running programs, resizing windows, and general navigation around the windows environment.
Decompressing files, installing software.

Basic probability theory

Although this material will be covered during the class, participants should already have studied the information below, also explained in **sections 1-3 here**:

- Basic probability rules
- What are independent, dependent, and mutually exclusive events
- P(A and B) = P(A) * P(B) when events A, B are independent
- P(A and B) = P(A) * P(B given A) when events are dependent
- P(A and B) = 0 when events are mutually exclusive
- Graphical description of probabilities: Relative and cumulative plots of a probability distribution
- Event trees

R

- Opening and closing Tinn-R/R
- Setting the working directory using *setwd()* and viewing it with *getwd()*
- Basic plotting principles

Specialized R functions

The functions below are not a pre-requisite for our courses, but it would be useful to at least review the list prior to the course.

R in general

Functions are interpreted line-by -line. Always allocate a variable to a function using "=" or <-; Many functions take arrays (represented in curly brackets {}) as inputs. (…) is used when extra optional parameters are available (but not relevant to us). The list below is not comprehensive; just a reference to the most basic functions needed to perform simulations/risk analysis in R

Basic settings and manipulations

Set directory	setwd("C:/") *Sets directory to C drive. Replace C with desired directory*
Get directory	getwd() *Returns current working directory*
Loading packages	require("package name") or library("package name")
List of variables stored	ls()
Removing a variable	rm("variable name")
Object/data structure	str(x) *Returns the internal structure of object x*
Help files	*?fun opens help for function fun* or *?"*" for help with operators* (i.e. *)

Importing and creating external data
Data files need to be in the current working directory (see *setwd()* and *getwd()*)

Reading data

read.table("myfile.csv", sep=",", header=T) *will import a csv file with column headings*
read.csv("myfile.csv", header=TRUE) *same as above*
read.delim("myfile.txt", header=TRUE) *will import a tab-delimited file with column headings*
Reading from clipboard – *scan("clipboard"), read.table("clipboard"),* etc.

Writing data

write.table(mydf, "myfile.csv", sep=",", row.names=F) *will write mydf data frame to a csv file*
write.table(mydf,"myfile.txt", sep="\t", row.names=F) *same as above but in tab-delimited format*
write.table("*clipboard*","myfile.txt", sep="\t", row.names=F) *Writes table to clipboard*

Descriptive Statistics

Summary statistics	summary({x}) *Generic function. Will yield diff. results for non-data objects*
Mean	mean({x})
Standard deviation (sample)	sd({x})
Variance (sample)	var({x})
Maximum	max({x})
Minimum	min({x})
Median	median({x})
Quantiles/percentiles	quantile({x}, probs) *quantiles for probabilities in probs (default is quartiles)*
Rank	rank(x, ties.method, …) *ties controls how equal values ("ties") are treated*
Combinations	choose(n,x)

Basic plots for simulation (see also RiskFunctionsinR.r for examples and more advanced functions)
The *plot* function works differently depending on the objects it's applied to. For example, on a data.frame it creates a scatterplot matrix, and if using a table object, it creates a barplot.

Histogram	hist(x, breaks="fd") *Breaks sets bins. Can also take numbers*
Scatter plot	plot(x, y) *x and y are coordinates. y is often optional*
Line plot	plot(x, y, type="l")
Scatter with vertical lines	plot(x, y, type="h") *Useful to plot discrete distributions*
Overlaying on existing plot	plot(x, y, add=T) *also works for histograms and many others*
Adding to existing plot	points(x,y) or lines(x,y) *Adds points or lines to existing plot*

Distributions; the first letter of the function specifies whether to return the probability density or mass (d), cumulative (p), inverse (q) at x, or n random values (r). **(more are given in the course files). <u>Examples below only for random number generation (RNG)</u>**

n= number of samples (iterations), p= percentile, q= target value (quantile), x= target value

Binomial distribution	rbinom(n,size,prob)
Negative Binomial	rnbinom(n,size,prob, …)
Normal distribution	rnorm(n,μ,σ)
Poisson distribution	rpois(n,λ)
Hypergeometric distribution	rhyper(nn,n,m,k) *see file Hypergeometric.r and PowerPoint for details*
Uniform distribution	runif(n,min, max)

Basic simulation construction

For loops	for(i in e) {code} *i: loop index, e: values for i. code: code to be executed I times. Can be multiple lines*
Vectorized simulation	*Parameters in RNGs can take vectors. e.g. rpois(100, rgamma(100,shape,scale)) each sample from the Gamma distribution is used as an individual rate for each sample from the Poisson distribution*

Least Squares Regression (functions lm() and glm() are the most relevant)

General form	model=lm(y~x1+x2, data=dat)
Coefficients (m)	coefficients(model)
Intercept	coefficients(model)[1]
Standard error of y	summary(model)$sigma
Prediction (uncertainty only)	predict(model,newdata, interval = "confidence")
Prediction (uncertainty/variability)	predict(model,newdata, interval = "prediction")

Basic operations

Sum of an array	sum({x})
Logical if statement (v1)	ifelse(condition, value if true, value if false)
Logical if statement (v2)	if(condition) {value if true} else {value if false} *curly brackets are optional*
Logical AND statement	condition_1 & condition_2 & …
Logical OR	condition_1, condition_2…)
Count	length({x})
Dimensions (for df,matrix, arrays)	nrow(df); ncol(df); dim(df) *# rows, # columns, and both respectively*
Counts only specific values	length(x[x== criterion]) *See PowerPoint for details*
Lookup	match(value(s) to be matched, values(s) to be matched against, …)
Absolute value	abs(x)
Exponential base e	exp(x)
Log_e	log(x, base=exp(1)) *generic for any base*
Round off value	round(x,digits)
Round up value	roundup <- function(x) trunc(x+0.5)
Square root	sqrt(value)
Missing value	NA

Fitting distributions (using fitdistrplus package)

Loading package	require(fitdistrplus)
Installing package from web	install.packages('fitdistrplus') *If package is not previously installed*
Parametric fit using MLE	fitdist(x, distr, method="mle") *distr is name of distribution to fit*
Plotting fit	plot(x) *x is fitdist object*
Fit statistics	summary(x) *x is fitdist object*
Extracting AIC/BIC	x$aic or x$bic *x is fitdist object*