IMAV 2013 R Tutorial #1
Eric Feigelson (Penn State)
Valparaiso May 2013

```
# Start a session
setwd('/Users/e5f/Desktop/Valparaiso')   # set working directory
getwd()                                  # report working directory
system('pwd')                            # report working directory (from operating
system)
citation()                               # citation reference for publications using R


# Create and examine a vector
a.test <- c(33,44,57)                    # c = concatenate
class(a.test)                            # find class of an object
str(a.test)                                      # str = structure (important for complex
'list' objects)
mean(a.test) ; sd(a.test)                # mean and standard deviation  (; = <CR>)
median(a.test) ; mad(a.test)                     # median and Median Absolute Deviation
help(mad)                                        # help files should be consulted very often
a.test                                   # print object on console
plot(a.test)                             # plot object on device
help(device)                             # default graphics devices
write(file='test.txt', a.test)   # see also print() and format()
ls()                                             # list objects in working directory
rm(a.test)                                       # delete object

# Construct dataset of 120 Sloan quasar r & z band magnitudes
qso <- read.table('http://astrostatistics.psu.edu/datasets/SDSS_QSO.dat', head=T,
fill=T)
class(qso)                                       # data.frame ~ matrix plus column headings
dim(qso)                                         # dimension of data.frame
names(qso)                                       # column (variable) names
summary(qso)                                     # quartiles and mean
rmag <- qso[1:120,9]                             # filter on [rows,columns]
zmag <- qso[1:120,13]
zmag                                             # print vector on console


# Make a simple plot: univariate empirical distribution function

plot(ecdf(rmag))                                 # default plot for this class
help(par)                                        # myriad options to alter plot appearance

# Improve plot of r magnitudes
```

```
plot(ecdf(rmag), cex=0.3, pch=2)
plot(ecdf(rmag), cex=0.3, pch=2, verticals=T, color='darkgreen')
   #  see http://research.stowers-institute.org/efg/R/Color/Chart for color options
plot(ecdf(rmag), cex=0.3, pch=2, verticals=T, col='darkgreen', xlim=c(17,22),
xlab='Sloan magnitude', ylab='Cumulative distribution', main='')

# Add plot of z magnitudes and annotate

plot(ecdf(zmag), cex=0, verticals=T, col='darkred', add=T)
text(18.5,0.4,'z mag', col='darkred') ; text(19.5,0.3,'r mag', col='darkgreen')
legend(17.5,0.8,c('r','z'),lty=c(1,1),col=c('darkgreen','darkred'))

#  Dynamically adjust size of plot and copy from current graphics device to another
format

dev.copy2eps(file='test_edf.eps')

# Plot e.d.f. with confidence bands: first use of a CRAN package
# Browse http://cran.r-project.org/web/packages/sfsmisc

install.packages('sfsmisc')          # choose a CRAN mirror site and download
package
library('sfsmisc')                   # introduce package into this R session
ecdf.ksCI(rmag)                              # this package automatically generates a
plot
ecdf.ksCI(rmag, ci.col='blue', main='')


# Do the z_mag and r_mag distributions have the same shape?

help(wilcox.test)
wilcox.test(rmag,zmag, conf.int=T)                  # there is an offset between the
vectors
wilcox.test(rmag,zmag+0.249, conf.int=T)         # yes, their shapes are
indistinguishable


# Examine the two-dimensional relationship between r_mag and z_mag

plot(rmag, zmag, pch=20)                    # 2D scatter plot
library(MASS)                                # an important package pre-loaded
into R
help(kde2d)
sm_rz <- kde2d(rmag, zmag, n=50)                  # Normal kernel density estimator
(Gaussian smoother)
```

```
image(sm_rz, xlab='r (mag)', ylab='z (mag)')      # Display 2D smoothed image
contour(sm_rz, add=T)                                    # Add contours
points(data, pch=20, cex=0.5)                          # Add original data points
help(lsfit)  ;  help(abline)
abline(lsfit(rmag, zmag), lwd=3)                  # Add least squares linear regression
line

# Relax linear assumption. Try LOESS local regression technique with different
parameters

lines(loess.smooth(data[,1],data[,2]), col='lightblue', lwd=3)
lines(loess.smooth(data[,1],data[,2], span=0.3), col='lightblue', lwd=3)
lines(loess.smooth(data[,1],data[,2], span=0.3, family='gaussian'), col='lightblue',
lwd=3)


# 37 R functions are used in this tutorial:
# setwd, getwd, system, citation, c, class, str, mean, sd, median, mad,
# help, plot, write, ls, rm, read.table, dim, names, summary, plot,
# ecdf, text, legend, dev.copy2eps, install.packages, library, ecdf.ksCI,
# wilcox.test, kde2d, image, contour, points, lsfit, abline,
# loess.smooth, lines
```