Summary of R commands for Statistics 100

Statistics 100 – Fall 2011
Professor Mark E. Glickman

The following is a summary of R commands we will be using throughout Statistics 100, and maybe a few extras we will not end up using. Please refer to the homework and course notes for examples of their usage, including the appropriate arguments of the commands. In the descriptions below, `fnc` is an arbitrary R command.

Reading, viewing, and assigning data in R:

`y = fnc(x)` − assigns the results of the function `fnc` evaluated at `x` to the variable `y`.

`file.choose()` − navigates to a data file on your computer.

`read.table(fname)` − reads data into R from file `fname`.

`read.csv(fname)` − reads data into R from a comma-separated value file `fname`

`data.frame(...)` − creates a data frame within R.

`View(x)` − view data frame `x` within R. Can also just type the name of the data frame at the prompt.

`help(fnc)` − help page for function "`fnc`".

Descriptive statistics:

`summary(x)` − data summary of `x`.

`mean(x)` − sample mean of `x`.

`sd(x)` − sample standard deviation of `x`.

`length(x)` − number of values in `x`.

`table(x)` − for categorical variable `x`, creates vector of counts of each unique category.

`cor(x,y)` − correlation between `x` and `y`.

`by(y,x,fnc)` − with categorical `x` and function `fnc`, carry out `fnc(y)` for each level of `x`.

Graphics:

`hist(x)` − histogram of data in `x`.

`stem(x)` − stem and leaf plot of data in `x`.

`plot(x,y)` − scatter plot of `y` against `x`.

`lines(supsmu(x,y))` − add smoother to existing scatter plot.

`boxplot(list(x1,x2,...))` − side-by-side boxplots of variables `x1`, `x2`, etc.

`boxplot(y ~ x)` − alternative method for boxplots if `y` is quantitative and `x` is categorical.

`barplot(x)` − barplot of `x` (where `x` contains the heights of the bars).

`abline(a,b)` − add the line $y = a + bx$ to an existing plot.

`abline(h=a)` − add a horizontal line at $y = a$ to an existing plot.

`abline(v=a)` − add a vertical line at $x = a$ to an existing plot.

`abline(model.fit)` − add a regression line based on the model `model.fit` to an existing plot.

`qqnorm(x)` − normal probability plot of data in `x`.

`qqline(x)` − adds a line to a normal probability plot passing through 1Q and 3Q

Probability distribution computations:

`dbinom(x, n, p)` − $\mathrm{P}(X = \mathtt{x})$ where $X \sim \mathrm{B}(\mathtt{n}, \mathtt{p})$

`pnorm(x, mean, sd)` − $\mathrm{P}(X < \mathtt{x})$ where $X \sim \mathrm{N}(\mathtt{mean}, \mathtt{sd})$

`qnorm(p, mean, sd)` − the value of x in $\mathtt{p} = \mathrm{P}(X < x)$, where $X \sim \mathrm{N}(\mathtt{mean}, \mathtt{sd})$

`pt(x, df)` − $\mathrm{P}(X < \mathtt{x})$ where $X \sim t(\mathtt{df})$

`qt(p, df)` − the value of x in $\mathtt{p} = \mathrm{P}(T < x)$, where $T \sim t(\mathtt{df})$

`pchisq(x, df)` − $\mathrm{P}(X^2 < \mathtt{x})$ where $X^2 \sim \chi^2(\mathtt{df})$

Random sampling (without replacement):

`sample(n)` − a random arrangement of the first `n` positive integers.

`sample(n, size)` − a random sample of `size` values from among the first `n` positive integers.

Statistical inference:

`t.test(x, mu)` − one-sample t-test or confidence interval with data in `x`, with null hypothesized value `mu`.

`t.test(x1, x2)` − two-sample t-test or confidence interval for difference in means with data in `x1` and `x2`

`t.test(y ~ x, data=data.df)` − alternative method for two-sample t-test; `y` is the quantitative response and `x` is binary categorical variable in data frame `data.df`.

`prop.test(x, n, p)` − one-sample z-test or confidence interval for a Binomial probability, with `x` successes in a sample size of `n`, and a hypothesized probability `p`.

`prop.test(x, n)` − two-sample z-test or confidence interval for difference in Binomial probabilities, with `x` containing two counts of successes, and `n` containing two sample sizes.

`mcnemar.test(x)` − McNemar's test for difference in Binomial probabilities with paired data, with `x` containing 2×2 data frame.

`aov(y ~ x, data=data.df)` − analysis of variance of response `y` on categorical variable `x` contained in data frame `data.df`.

`lm(y~x1+x2+x3+..., data=data.df)` − least-squares regression of `y` on `x1`, `x2`, etc., within data frame `data.df`.

`glm(y~x1+x2+x3+..., family=binomial, data=data.df)` − logistic regression of `y` on `x1`, `x2`, etc., within data frame `data.df`.

`summary(model.fit)` − summarize `model.fit`, the results of either analysis of variance, least-squares regression, or logistic regression.

`step(model.fit)` − stepwise variable selection for least-squares or logistic regressions, with largest model in `model.fit`.

`predict(model.fit, newdata=newdata.df)` − prediction of least-squares or logistic regression model in `model.fit` using data in `newdata.df`.

`fitted(model.fit)` − fitted values from `model.fit`.

`residuals(model.fit)` − residuals from `model.fit`.

`chisq.test(x, p)` − chi-squared goodness-of-fit test, with vector of counts in `x` and vector of probabilities in `p`.

`chisq.test(x)` − chi-squared test of independence, with counts in `x` as a data frame.