R Code in

Foundations and Applications of Statistics

An introduction to R is provided in Appendix A. R functions are introduced as needed throughout the main text and summarized at the end of each chapter. These end-of-chapter summary tables are provided here as a service to students and instructors who want an overview of what R functions are used where in the book. Feel free to copy and distribute these pages.
Chapter 1: Data

```r
x <- c(...)  # Concatenate arguments into a single vector and store in object x.
data(x)       # (Re)load the data set x.
str(x)        # Print a summary of the object x.
head(x,n=4)   # First four rows of the data frame x.
tail(x,n=4)   # Last four rows of the data frame x.
table(x)      # Table of the values in vector x.
xtabs(~x+y,data)  # Cross tabulation of x and y.
cut(x,breaks,right=TRUE)  # Divide up the range of x into intervals and code the values in x according to which interval they fall into.
require(fastR) # Load packages.
require(lattice)
require(Hmisc)
histogram(~x|z,data,...) # Histogram of x conditioned on z.
bwplot(x~z,data,...)    # Boxplot of x conditioned on z.
xyplot(y~x|z,data,...)  # Scatterplot of y by x conditioned on z.
stem(x)                # Stemplot of x.
sum(x); mean(x); median(x); var(x); sd(x); quantile(x) # Sum, mean, median, variance, standard deviation, quantiles of x.
summary(y~x,data,fun)  # Summarize y by computing the function fun on each group defined by x [Hmisc].
```

©Randall Pruim, 2011
Chapter 2: Discrete Distributions

choose(n,k)

\[ \binom{n}{k} = \frac{n!}{(n-k)!k!} \]

dbinom(x,size,prob)

P\{X = x\} for X \sim Binom(size,prob).
pbinom(q,size,prob)

P\{X \leq q\} for X \sim Binom(size,prob).
qbinom(p,size,prob)

Smallest x such that P\{X \leq x\} \geq p for X \sim Binom(size,prob).
rbinom(n,size,prob)

Simulate n random draws from a Binom(size,prob)-distribution.

dpois(...); ppois(...);
qpois(...); rpois(...);
dnbinom(...); pnbinom(...);
qnbinom(...); rnbinom(...);
dhyper(...); phyper(...);
qhyper(...); rhyper(...)

Similar to the functions above but for Poisson, negative binomial, and hypergeometric distributions.

rep(values,...)

Create a vector of repeated values.

sum(x,...); prod(x,...)

Compute the sum or product of values in the vector x.

binom.test(x,n,p,...)

Conduct a binomial test of \( H_0 : \pi = p \) from a data set with x successes in n tries.

fisher.test(

Conduct Fisher’s exact test with data summarized in the table

\[
\begin{array}{cc}
x & y \\
z & w \\
\end{array}
\]
Chapter 3: Continuous Distributions

\[ \text{dnorm}(x, \text{mean}, \text{sd}) \]  
pdf for \( X \sim \text{Norm}(\text{mean}, \text{sd}) \).

\[ \text{pnorm}(q, \text{mean}, \text{sd}) \]  
\( P(X \leq q) \) for \( X \sim \text{Norm}(\text{mean}, \text{sd}) \).

\[ \text{qnorm}(p, \text{mean}, \text{sd}) \]  
\( x \) such that \( P(X \leq x) = p \) for \( X \sim \text{Norm}(\text{mean}, \text{sd}) \).

\[ \text{rnorm}(n, \text{mean}, \text{sd}) \]  
Simulate \( n \) random draws from a \( \text{Norm}(\text{mean}, \text{sd}) \)-distribution.

\[ \text{dunif}(\ldots); \text{punif}(\ldots); \text{qunif}(\ldots); \text{runif}(\ldots); \]  
Similar to the functions above but for uniform, exponential, gamma, beta, and Weibull distributions.

\[ \text{dexp}(\ldots); \text{pexp}(\ldots); \text{qexp}(\ldots); \text{rexp}(\ldots); \]  

\[ \text{dgamma}(\ldots); \text{pgamma}(\ldots); \text{qgamma}(\ldots); \text{rgamma}(\ldots); \]  

\[ \text{dbeta}(\ldots); \text{pbeta}(\ldots); \text{qbeta}(\ldots); \text{rbeta}(\ldots); \]  

\[ \text{dweibull}(\ldots); \text{pweibull}(\ldots); \text{qweibull}(\ldots); \text{rweibull}(\ldots) \]  

\[ f <- \text{function}(\ldots) \{ \} \]
Define a function.

\[ \text{integrate}(f, \text{lower}, \text{upper}, \ldots) \]
Numerically approximate \( \int_{\text{lower}}^{\text{upper}} f(x) \, dx \).

\[ \text{adaptIntegrate}(f, \text{lowerLimit}, \text{upperLimit}, \text{tol}, \ldots) \]
Numerically approximate multivariate integrals [cubature].

\[ \text{fractions}(x, \ldots) \]  
Find a rational number near \( x \) [MASS].

\[ \text{sapply}(X, \text{FUN}) \]
Apply the function \( \text{FUN} \) to each element of the vector \( X \).

\[ \text{gamma}(x) \]
\( \Gamma(x) \)

\[ \text{density}(x, \text{bw}, \text{adjust}, \text{kernel}, \ldots) \]
Kernel density estimate.

\[ \text{densityplot}(x, \text{data}, \text{allow.multiple}, \text{bw}, \text{adjust}, \text{kernel}, \ldots) \]
Kernel density plot.

\[ \text{qnorm}(x, \ldots); \text{qqmath}(x, \ldots); \text{xqmath}(x, \ldots) \]  
Normal-quantile plot for \( x \). (Other distributions are also possible.)

\[ \text{data.frame}(\ldots); \]
Construct a new data frame.
## Chapter 4: Parameter Estimation and Testing

- `uniroot(f,interval,...)`  
  Numerically approximate a solution to $f(x) = 0$ with $x$ within the interval specified by `interval`.

- `sample(x,size,replace=FALSE)`  
  Select a sample of size `size` from `x`.

- `binom.test(x,n,p=0.50,...)`  
  Use binomial distributions to conduct a hypothesis test or construct a confidence interval for a proportion.

- `prop.test(x,n,p=0.50,...)`  
  Use normal approximations to the binomial distributions to conduct a hypothesis test or construct a confidence interval for a proportion.

- `t.test(x,...)`  
  $t$-tests and confidence intervals.

- `replicate(n,expr,...)`  
  Evaluate expression `expr` `n` times.

- `dt(x,df)`  
  Evaluate pdf for $t(df)$-distribution

- `pt(q,df)`  
  Evaluate cdf for $t(df)$-distribution

- `qt(p,df)`  
  Compute quantiles for $t(df)$-distribution

- `rt(n,df)`  
  Simulate `n` random draws from a $t(df)$-distribution.

- `dchisq(x,df); pchisq(q,df); qchisq(p,df); rchisq(n,df)`  
  Similar to the functions above but for $Chi^2(df)$-distributions.

- `df(x,df1,df2); pf(q,df1,df2); qf(p,df1,df2); rf(n,df1,df2)`  
  Similar to the functions above but for $F(df1,df2)$-distributions.

©Randall Pruim, 2011
Chapter 5: Likelihood-Based Statistics

.nlm(f,p,x) Minimize $f$ starting from point $p$.
.nlmin(f,p,x) Minimize $f$ starting from point $p$ \texttt{[fastR wrapper for nlm()]}.  
.nlmax(f,p,x) Maximize $f$ starting from point $p$ \texttt{[fastR wrapper for nlm()]}.  
.summary(nlmax(f,p,x)) Summary output for \texttt{nlmax()}.  
.oldopt <- options(warn=-1) Turn off warnings and save previous options.  
options(oldopt) Revert to old options.  
.xhistogram(~x,data,...) Histogram with some extras \texttt{[fastR]}.  
.unisroot(f,interval,...) Numerically approximate a solution to $f(x) = 0$ for $x$ within the interval specified by \texttt{interval}.  
.nrow(x); ncol(x) The number of rows or columns in an object $x$.  
rbind(...) Bind together rowwise into a matrix.  
.cbind(...) Bind together columnwise into a matrix.  
.rownames(x) Access or set the row names of object $x$.  
.colnames(x) Access or set the column names of object $x$.  
.chisq.test(x,...) Perform a Pearson Chi-squared test; handles some simple goodness of fit testing and 2-way tables.  
.xchisq.test(x,...) Perform a Pearson Chi-squared test and display some extra information \texttt{[fastR]}.  
mosaic(...) Construct a mosaic plot \texttt{[vcd]}.  
.merge(x,y,...) Merge data frames $x$ and $y$.  
.BTm(...) Fit a Bradley-Terry model \texttt{[BradleyTerry2]}.  
.coef(model) Compute coefficients of a model.  
.logit(x), ilogit() Logit and inverse logit functions \texttt{[faraway]}.  

©Randall Pruim, 2011
Chapter 6: Introduction to Linear Models

```r
lm(y ∼ x,...)  # Fit a linear model.
glm(y ∼ x,
    family=binomial(link=logit),
    ...)  # Fit a logistic regression model.
glm(cbind(successes,failures) ∼ x,
    family=binomial(link=logit),
    ...)  # Fit a logistic regression model using tabulated data.
I(...)  # Inhibit interpretation in model formulas. Several arithmetic operators have special meanings in the context of a formula. Surrounding them with I() causes them to take on their usual arithmetic meaning.

summary(model)  # Print a numerical summary of a model (output from lm() or glm()).
anova(model)  # Print an ANOVA table.
plot(model,...)  # Generate some diagnostic plots.
xplot(model,...)  # Generate some diagnostic plots (using lattice plots) [fastR].
confint(model,...)  # Compute confidence intervals for parameters.
predict(model,...)  # Predict responses expressed as point estimate (default), confidence interval (interval='confidence'), or prediction interval (interval='prediction'). Use newdata=data.frame(...) to specify the explanatory variables for the predictions.
```

©Randall Pruim, 2011
Chapter 7: More Linear Models

- `factor(x)`
  - Convert x to a factor.

- `summary(formula, data,..., fun, method,...)`
  - Tabulate various summary statistics for a data set [Hmisc].

- `favstats(x)`
  - Compute some basic summary statistics for x [fastR].

- `project(y, v,...)`
  - Project y in the direction of v [fastR].

- `dot(x, y)`
  - Dot product of x and y [fastR].

- `vlength(x)`
  - Vector length (norm) of x [fastR].

- `model <- lm(y ~ x,...)`
  - Fit a linear model.

- `model <- glm(y ~ x,...)`
  - Fit a generalized linear model.

- `summary(model)`
  - Print summary of model.

- `resid(model)`
  - Residuals of model.

- `plot(model); xplot(model)`
  - Diagnostic plots for model.

- `anova(model)`
  - Print ANOVA table for model.

- `anova(model1, model2)`
  - Print ANOVA table for model comparison test of two nested models.

- `confint(model,...)`
  - Confidence intervals for model parameters.

- `predict(model,...)`
  - Confidence intervals and prediction intervals for the response variable.

- `glht(model,...)`
  - General linear hypothesis tests – p-values and confidence intervals for contrasts with multiple comparisons corrections as needed [multcomp].

- `mcp(...)`
  - Construct sets of contrast for use with `glht()` [multcomp].

- `step(model,...)`
  - Stepwise regression.

- `vif(model,...)`
  - Variance inflation factor [faraway].

- `aov(y~x,...)`
  - Alternative to `lm()` that stores and prints different information.

- `TukeyHSD(aov(y~x,...))`
  - Tukey Honest Significant Differences.

- `splom(...)`
  - Scatterplot matrix [lattice].

- `scatterplot.matrix(...)`
  - Scatterplot matrix [car].

©Randall Pruim, 2011