CDT Module 1 - R Review

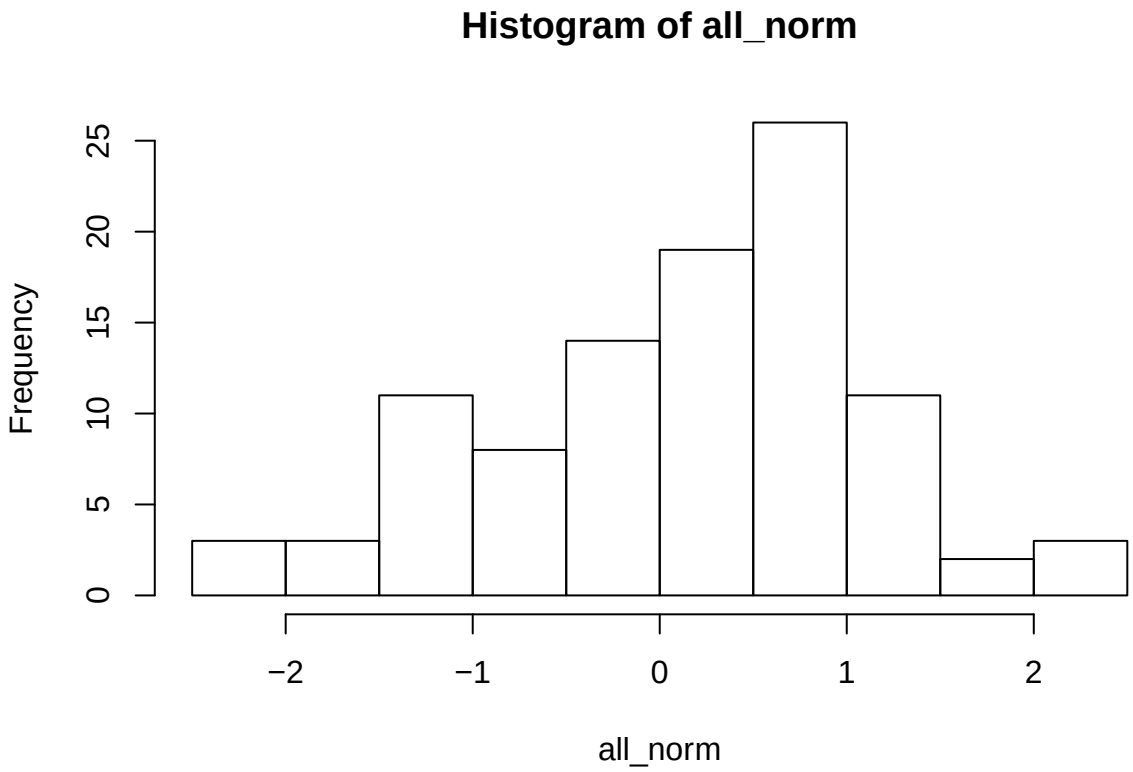
Valerie Bradley

9/19/2018

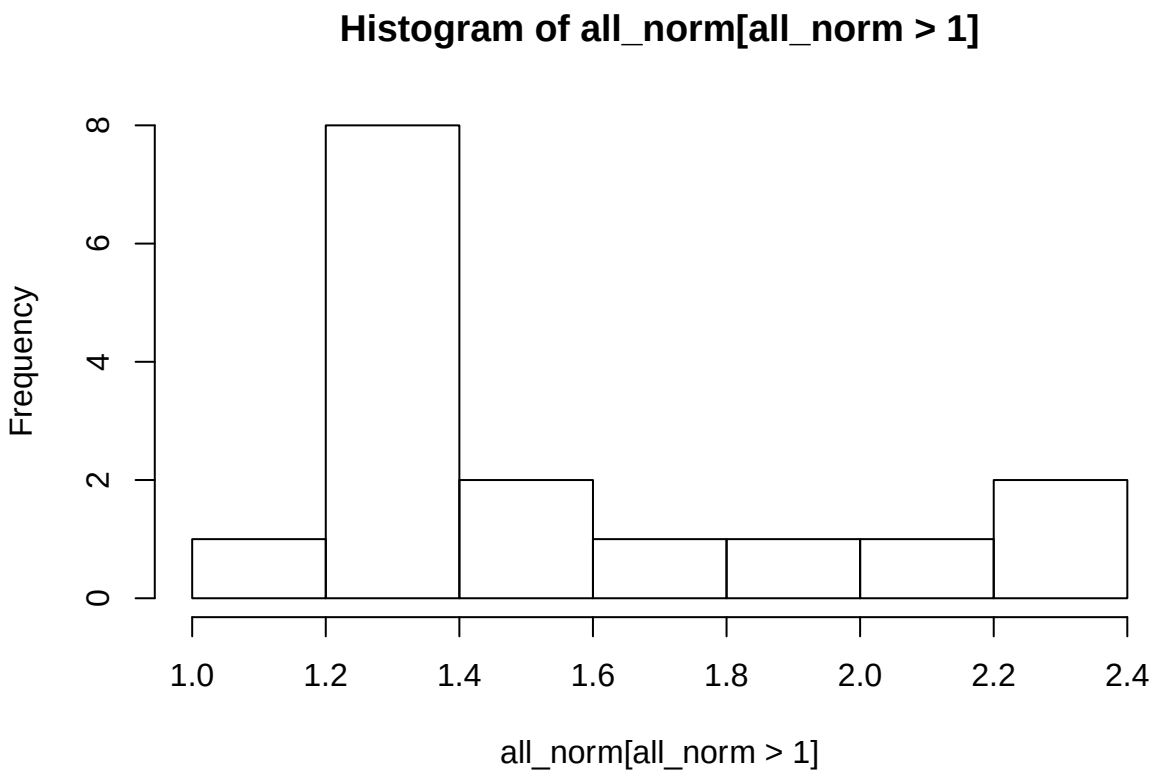
1. Vectors

a. Generate 100 standard normal random variables, and keep only the ones which are greater than 1. Don't use a loop!

```
all_norm = rnorm(100)
hist(all_norm)
```



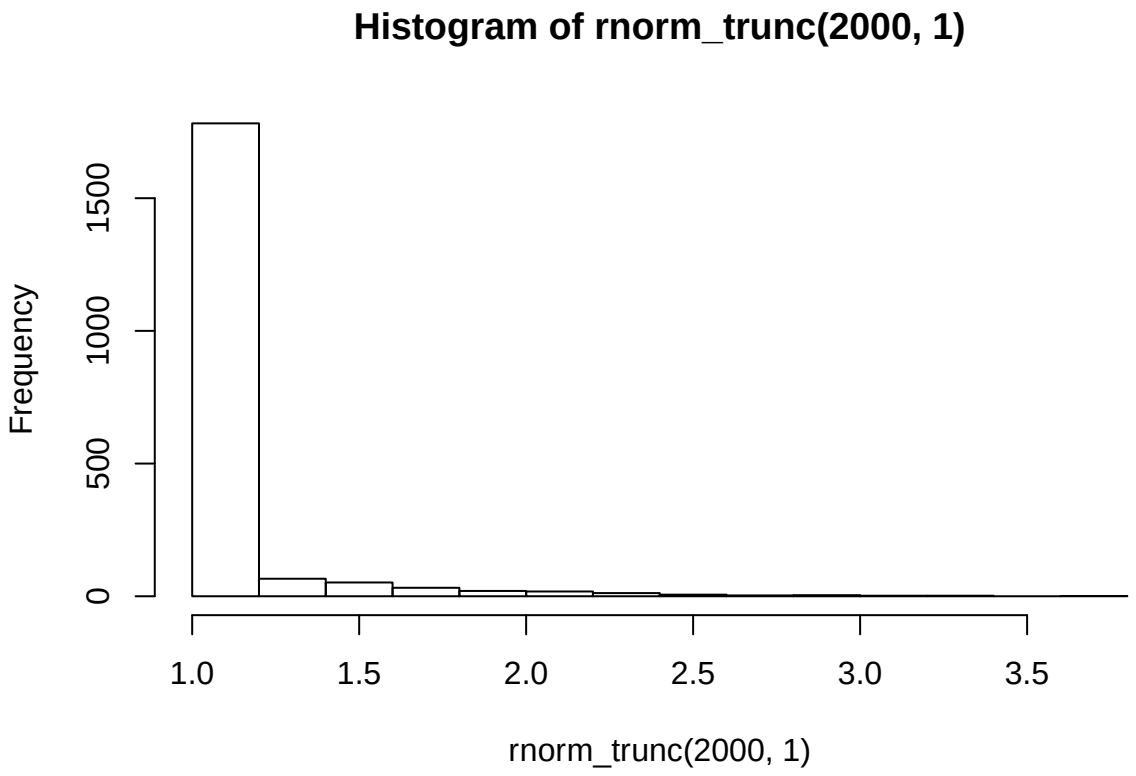

```
hist(all_norm[all_norm > 1])
```



b. Write a function which takes two arguments n and min, and returns n independent random variables from a standard normal distribution truncated below by min. Let min default to 0.

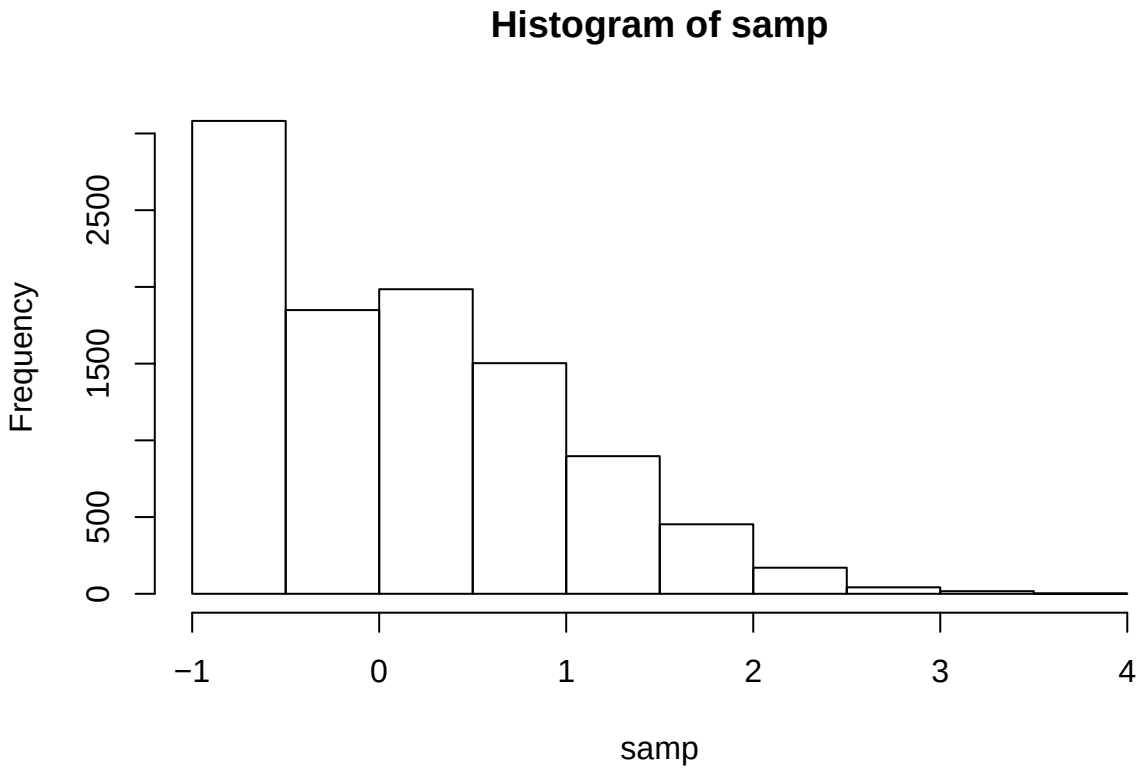
```
rnorm_trunc = function(n, min = 0){
  X = rnorm(n)
  X = unlist(lapply(X, max, min))
  return(X)
}

hist(rnorm_trunc(2000, 1))
```



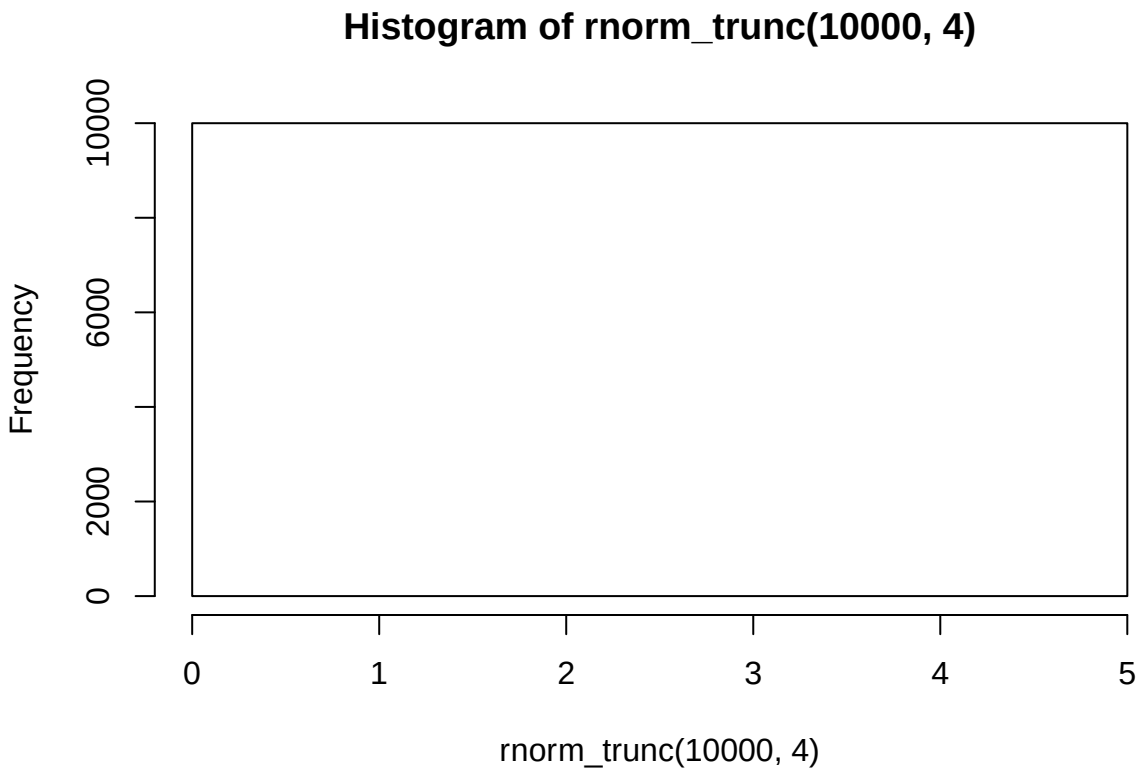
c. Generate 10k truncated normals with min set at -1 and plot as histogram

```
samp = rnorm_trunc(10000, -1)
hist(samp, breaks  = 10)
```



d. what happens if min is large?

```
hist(rnorm_trunc(10000,4))
```



Could improve by adding params for the mean and SD of the normal from which you'd like to sample.

2. Data

```
library(MASS)
data(hills)
summary(hills)
```

```
##       dist            climb          time       
##  Min.   : 2.000   Min.   : 300   Min.   : 15.95  
##  1st Qu.: 4.500   1st Qu.: 725   1st Qu.: 28.00  
##  Median : 6.000   Median :1000   Median : 39.75  
##  Mean   : 7.529   Mean   :1815   Mean   : 57.88  
##  3rd Qu.: 8.000   3rd Qu.:2200   3rd Qu.: 68.62  
##  Max.   :28.000   Max.   :7500   Max.   :204.62
```

a. what sort of object is hills?

```
class(hills)
```

```
## [1] "data.frame"
```

b. how many columns?

```
ncol(hills)
```

```
## [1] 3
```

c. Change "Two Breweries" to "Three Breweries"

```
hills[which(rownames(hills) == 'Two Breweries'),]
```

```
##                dist climb   time
## Two Breweries   18  5200 170.25
```

```
row.names(hills)[which(rownames(hills) == 'Two Breweries')] <-'Three Breweries'
hills[which(rownames(hills) == 'Two Breweries'),]
```

```
## [1] dist  climb time
## <0 rows> (or 0-length row.names)
```

```
hills[which(rownames(hills) == 'Three Breweries'),]
```

```
##                  dist climb   time
## Three Breweries   18  5200 170.25
```

d. Find the mean time for races iwth a climb greater than 1000ft

```
with(hills[hills$climb > 1000,], mean(dist))
```

```
## [1] 10.41176
```

e. What sort of object is Orthodont? How is it different from `hills`?

```
library(nlme)
data(Orthodont)
head(Orthodont)
```

```
## Grouped Data: distance ~ age | Subject
##   distance age Subject  Sex
## 1     26.0   8     M01 Male
## 2     25.0  10     M01 Male
## 3     29.0  12     M01 Male
## 4     31.0  14     M01 Male
## 5     21.5   8     M02 Male
## 6     22.5  10     M02 Male
```

```
#class(Orthodont)
```

It's grouped data.

```
methods(print)
```

```
##   [1] print.abbrev*
##   [2] print.acf*
##   [3] print.AES*
##   [4] print.anova*
##   [5] print.Anova*
##   [6] print.anova.lme*
##   [7] print.anova.loglm*
##   [8] print.aov*
##   [9] print.aovlist*
##  [10] print.ar*
##  [11] print.Arima*
##  [12] print.arima0*
##  [13] print.AsIs
##  [14] print.aspell*
##  [15] print.aspell_inspect_context*
##  [16] print.bibentry*
##  [17] print.Bibtex*
```

```
##  [18] print.browseVignettes*
##  [19] print.by
##  [20] print.bytes*
##  [21] print.changedFiles*
##  [22] print.check_code_usage_in_package*
##  [23] print.check_compiled_code*
##  [24] print.check_demo_index*
##  [25] print.check_depdef*
##  [26] print.check_details*
##  [27] print.check_details_changes*
##  [28] print.check_doi_db*
##  [29] print.check_dotInternal*
##  [30] print.check_make_vars*
##  [31] print.check_nonAPI_calls*
##  [32] print.check_package_code_assign_to_globalenv*
##  [33] print.check_package_code_attach*
##  [34] print.check_package_code_data_into_globalenv*
##  [35] print.check_package_code_startup_functions*
##  [36] print.check_package_code_syntax*
##  [37] print.check_package_code_unload_functions*
##  [38] print.check_package_compact_datasets*
##  [39] print.check_package_CRAN_incoming*
##  [40] print.check_package_datasets*
##  [41] print.check_package_depends*
##  [42] print.check_package_description*
##  [43] print.check_package_description_encoding*
##  [44] print.check_package_license*
##  [45] print.check_packages_in_dir*
##  [46] print.check_packages_used*
##  [47] print.check_po_files*
##  [48] print.check_pragmas*
##  [49] print.check_Rd_contents*
##  [50] print.check_Rd_line_widths*
##  [51] print.check_Rd_metadata*
##  [52] print.check_Rd_xrefs*
##  [53] print.check_RegSym_calls*
##  [54] print.check_so_symbols*
##  [55] print.check_T_and_F*
##  [56] print.check_url_db*
##  [57] print.check_vignette_index*
##  [58] print.checkDocFiles*
##  [59] print.checkDocStyle*
##  [60] print.checkFF*
##  [61] print.checkRd*
##  [62] print.checkReplaceFuns*
##  [63] print.checkS3methods*
##  [64] print.checkTnF*
##  [65] print.checkVignettes*
##  [66] print.citation*
##  [67] print.codoc*
##  [68] print.codocClasses*
##  [69] print.codocData*
##  [70] print.colorConverter*
##  [71] print.compactPDF*
```

```
##  [72] print.compareFits*
##  [73] print.condition
##  [74] print.connection
##  [75] print.corNatural*
##  [76] print.correlation*
##  [77] print.correspondence*
##  [78] print.corStruct*
##  [79] print.corSymm*
##  [80] print.CRAN_package_reverse_dependencies_and_views*
##  [81] print.data.frame
##  [82] print.Date
##  [83] print.default
##  [84] print.dendrogram*
##  [85] print.density*
##  [86] print.difftime
##  [87] print.dist*
##  [88] print.Dlist
##  [89] print.DLLInfo
##  [90] print.DLLInfoList
##  [91] print.DLLRegisteredRoutines
##  [92] print.dummy_coef*
##  [93] print.dummy_coef_list*
##  [94] print.ecdf*
##  [95] print.eigen
##  [96] print.factanal*
##  [97] print.factor
##  [98] print.family*
##  [99] print.fileSnapshot*
## [100] print.findLineNumResult*
## [101] print.fitdistr*
## [102] print.flatGridListing*
## [103] print.formula*
## [104] print.fractions*
## [105] print.fseq*
## [106] print.ftable*
## [107] print.function
## [108] print.gamma.shape*
## [109] print.getAnywhere*
## [110] print.gList*
## [111] print.glm*
## [112] print.glm.dose*
## [113] print.gls*
## [114] print.gpar*
## [115] print.grob*
## [116] print.groupedData*
## [117] print.hclust*
## [118] print.help_files_with_topic*
## [119] print.hexmode
## [120] print.HoltWinters*
## [121] print.hsearch*
## [122] print.hsearch_db*
## [123] print.htest*
## [124] print.html*
## [125] print.html_dependency*
```

```
## [126] print.infl*
## [127] print.integrate*
## [128] print.intervals.gls*
## [129] print.intervals.lme*
## [130] print.intervals.lmList*
## [131] print.isoreg*
## [132] print.kmeans*
## [133] print.knitr_kable*
## [134] print.Latex*
## [135] print.LaTeX*
## [136] print.lda*
## [137] print.libraryIQR
## [138] print.listof
## [139] print.lm*
## [140] print.lme*
## [141] print.lmList*
## [142] print.loadings*
## [143] print.loess*
## [144] print.logLik*
## [145] print.loglm*
## [146] print.lqs*
## [147] print.ls_str*
## [148] print.mca*
## [149] print.medpolish*
## [150] print.MethodsFunction*
## [151] print.modelStruct*
## [152] print.mtable*
## [153] print.NativeRoutineList
## [154] print.news_db*
## [155] print.nls*
## [156] print.noquote
## [157] print.numeric_version
## [158] print.object_size*
## [159] print.octmode
## [160] print.packageDescription*
## [161] print.packageInfo
## [162] print.packageIQR*
## [163] print.packageStatus*
## [164] print.pairwise.htest*
## [165] print.path*
## [166] print.PDF_Array*
## [167] print.PDF_Dictionary*
## [168] print.pdf_doc*
## [169] print.pdf_fonts*
## [170] print.PDF_Indirect_Reference*
## [171] print.pdf_info*
## [172] print.PDF_Keyword*
## [173] print.PDF_Name*
## [174] print.PDF_Stream*
## [175] print.PDF_String*
## [176] print.pdMat*
## [177] print.person*
## [178] print.polr*
## [179] print.POSIXct
```

```
## [180] print.POSIXlt
## [181] print.power.htest*
## [182] print.ppr*
## [183] print.prcomp*
## [184] print.princomp*
## [185] print.proc_time
## [186] print.qda*
## [187] print.ranef.lme*
## [188] print.raster*
## [189] print.Rcpp_stack_trace*
## [190] print.Rd*
## [191] print.recordedplot*
## [192] print.restart
## [193] print.reStruct*
## [194] print.RGBcolorConverter*
## [195] print.ridgelm*
## [196] print.rle
## [197] print.rlm*
## [198] print.rms.curv*
## [199] print.roman*
## [200] print.root_criterion*
## [201] print.sessionInfo*
## [202] print.shingle*
## [203] print.shingleLevel*
## [204] print.shiny.tag*
## [205] print.shiny.tag.list*
## [206] print.simple.list
## [207] print.simulate.lme*
## [208] print.smooth.spline*
## [209] print.socket*
## [210] print.srcfile
## [211] print.srcref
## [212] print.stepfun*
## [213] print.stl*
## [214] print.StructTS*
## [215] print.subdir_tests*
## [216] print.summarize_CRAN_check_status*
## [217] print.summary.aov*
## [218] print.summary.aovlist*
## [219] print.summary.corNatural*
## [220] print.summary.corStruct*
## [221] print.summary.corSymm*
## [222] print.summary.ecdf*
## [223] print.summary.glm*
## [224] print.summary.gls*
## [225] print.summary.lm*
## [226] print.summary.lme*
## [227] print.summary.lmList*
## [228] print.summary.loess*
## [229] print.summary.loglm*
## [230] print.summary.manova*
## [231] print.summary.modelStruct*
## [232] print.summary.negbin*
## [233] print.summary.nls*
```

```
## [234] print.summary.packageStatus*
## [235] print.summary.pdMat*
## [236] print.summary.polr*
## [237] print.summary.ppr*
## [238] print.summary.prcomp*
## [239] print.summary.princomp*
## [240] print.summary.rlm*
## [241] print.summary.table
## [242] print.summary.trellis*
## [243] print.summary.varComb*
## [244] print.summary.varFixed*
## [245] print.summary.varFunc*
## [246] print.summary.warnings
## [247] print.summaryDefault
## [248] print.table
## [249] print.tables_aov*
## [250] print.terms*
## [251] print.trellis*
## [252] print.ts*
## [253] print.tskernel*
## [254] print.TukeyHSD*
## [255] print.tukeyline*
## [256] print.tukeysmooth*
## [257] print.undoc*
## [258] print.unit*
## [259] print.varComb*
## [260] print.VarCorr.lme*
## [261] print.VarCov*
## [262] print.varFunc*
## [263] print.viewport*
## [264] print.vignette*
## [265] print.warnings
## [266] print.xgettext*
## [267] print.xngettext*
## [268] print.xtabs*
## see '?methods' for accessing help and source code
```

```
nlme:::print.groupedData(Orthodont)
```

```
## Grouped Data: distance ~ age | Subject
##     distance age Subject    Sex
## 1       26.0   8     M01   Male
## 2       25.0  10     M01   Male
## 3       29.0  12     M01   Male
## 4       31.0  14     M01   Male
## 5       21.5   8     M02   Male
## 6       22.5  10     M02   Male
## 7       23.0  12     M02   Male
## 8       26.5  14     M02   Male
## 9       23.0   8     M03   Male
## 10      22.5  10     M03   Male
## 11      24.0  12     M03   Male
## 12      27.5  14     M03   Male
## 13      25.5   8     M04   Male
## 14      27.5  10     M04   Male
```

```
## 15      26.5  12     M04   Male
## 16      27.0  14     M04   Male
## 17      20.0   8     M05   Male
## 18      23.5  10     M05   Male
## 19      22.5  12     M05   Male
## 20      26.0  14     M05   Male
## 21      24.5   8     M06   Male
## 22      25.5  10     M06   Male
## 23      27.0  12     M06   Male
## 24      28.5  14     M06   Male
## 25      22.0   8     M07   Male
## 26      22.0  10     M07   Male
## 27      24.5  12     M07   Male
## 28      26.5  14     M07   Male
## 29      24.0   8     M08   Male
## 30      21.5  10     M08   Male
## 31      24.5  12     M08   Male
## 32      25.5  14     M08   Male
## 33      23.0   8     M09   Male
## 34      20.5  10     M09   Male
## 35      31.0  12     M09   Male
## 36      26.0  14     M09   Male
## 37      27.5   8     M10   Male
## 38      28.0  10     M10   Male
## 39      31.0  12     M10   Male
## 40      31.5  14     M10   Male
## 41      23.0   8     M11   Male
## 42      23.0  10     M11   Male
## 43      23.5  12     M11   Male
## 44      25.0  14     M11   Male
## 45      21.5   8     M12   Male
## 46      23.5  10     M12   Male
## 47      24.0  12     M12   Male
## 48      28.0  14     M12   Male
## 49      17.0   8     M13   Male
## 50      24.5  10     M13   Male
## 51      26.0  12     M13   Male
## 52      29.5  14     M13   Male
## 53      22.5   8     M14   Male
## 54      25.5  10     M14   Male
## 55      25.5  12     M14   Male
## 56      26.0  14     M14   Male
## 57      23.0   8     M15   Male
## 58      24.5  10     M15   Male
## 59      26.0  12     M15   Male
## 60      30.0  14     M15   Male
## 61      22.0   8     M16   Male
## 62      21.5  10     M16   Male
## 63      23.5  12     M16   Male
## 64      25.0  14     M16   Male
## 65      21.0   8     F01 Female
## 66      20.0  10     F01 Female
## 67      21.5  12     F01 Female
## 68      23.0  14     F01 Female
```

```
## 69      21.0   8     F02 Female
## 70      21.5  10     F02 Female
## 71      24.0  12     F02 Female
## 72      25.5  14     F02 Female
## 73      20.5   8     F03 Female
## 74      24.0  10     F03 Female
## 75      24.5  12     F03 Female
## 76      26.0  14     F03 Female
## 77      23.5   8     F04 Female
## 78      24.5  10     F04 Female
## 79      25.0  12     F04 Female
## 80      26.5  14     F04 Female
## 81      21.5   8     F05 Female
## 82      23.0  10     F05 Female
## 83      22.5  12     F05 Female
## 84      23.5  14     F05 Female
## 85      20.0   8     F06 Female
## 86      21.0  10     F06 Female
## 87      21.0  12     F06 Female
## 88      22.5  14     F06 Female
## 89      21.5   8     F07 Female
## 90      22.5  10     F07 Female
## 91      23.0  12     F07 Female
## 92      25.0  14     F07 Female
## 93      23.0   8     F08 Female
## 94      23.0  10     F08 Female
## 95      23.5  12     F08 Female
## 96      24.0  14     F08 Female
## 97      20.0   8     F09 Female
## 98      21.0  10     F09 Female
## 99      22.0  12     F09 Female
## 100     21.5  14     F09 Female
## 101     16.5   8     F10 Female
## 102     19.0  10     F10 Female
## 103     19.0  12     F10 Female
## 104     19.5  14     F10 Female
## 105     24.5   8     F11 Female
## 106     25.0  10     F11 Female
## 107     28.0  12     F11 Female
## 108     28.0  14     F11 Female
```

3. Recursion