R: Presenting outputs

author: Dr N Green date: March 2018 autosize: false width: 2440 height: 1200 css: custom.css

CRAN Task View

CRAN Task View: Reproducible Research

Maintainer: Max Kuhn
Contact: max.kuhn at pfizer.com
Version: 2015-12-03
URL: https://CRAN.R-project.org/view=ReproducibleResearch

The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better understood and verified. Packages in R for this purpose can be split into groups for: literate programming, package reproducibility, code/data formatting tools, format convertors, and object caching.

Literate Programming

The primary way that R facilitates reproducible research is using a document that is a combination of content and data analysis code. The `Sweave` function (in the base R utils package) and the knitr package can be used to blend the subject matter and R code so that a single document defines the content and the analysis. The brew and R.rsp packages contain alternative approaches to embedding R code into various markups.

The resources for literate programming are best organized by the document type/markup language:

LaTeX

Both `Sweave` and knitr can process LaTeX files. lazyWeave can create LaTeX documents from scratch.

Object Conversion Functions:

- *summary tables/statistics* : Hmisc, NMOF, papeR, quantreg, rapport, reporttools, sparktex, tables, xtable, ztable
- *tables/cross-tabulations* : compareGroups, Hmisc, lazyWeave, knitLatex, knitr, reporttools, ztable
- *graphics* : animation, Hmisc, `grDevices:::pictex`, sparktex, tikzDevice
- *statistical models/methods* : apsrtable, memisc, quantreg, r2lh, rms, stargazer, suRtex, TeachingSampling, texreg, xtable, ztable
- *bibtex* : bibtex and RefManageR
- *others* : latex2exp converts LaTeX equations to `plotmath` expressions.

Miscellaneous Tools

- Hmisc contains a function to correctly escape special characters. resumer creates resumes. Standardized exams can be created using the exams package.

HTML

The knitr package can process HTML files directly. `Sweave` can also work with HTML by way of the R2HTML package. Kmisc and lazyWeave can create HTML format documents from scratch.

Object Conversion Functions:

- *summary tables/statistics* : stargazer
- *tables/cross-tabulations* : compareGroups, DT, formattable, htmlTable, HTMLUtils, hwriter, Kmisc, knitr, lazyWeave, SortableHTMLTables, texreg, ztable
- *statistical models/methods* : r2lh, rapport, stargazer, xtable
- *others* : knitcitations, RefManageR

Scatter plot: add points of different colours and symbols

`pch = ...`, `col = ...`

0: □	10: ⊕	20: •	A: A
1: ○	11: ✡	21: ●	a: a
2: △	12: ⊞	22: ■	B: B
3: +	13: ⊗	23: ◆	b: b
4: ×	14: ☑	24: ▲	S: S
5: ◇	15: ■	25: ▼	`: `
6: ▽	16: ●	@: @	.: .
7: ⊠	17: ▲	+: +	,: ,
8: ✳	18: ◆	%: %	?: ?
9: ⊕	19: ●	#: #	*: *

Scatter plot: add points of different colours and symbols

```
points(x, y = NULL, type = "p", ...)
```

```
x <- rnorm(10, sd=5, mean=20)
y <- 2.5*x - 1.0 + rnorm(10, sd=9, mean=0)
plot(x,y,xlab="Independent", ylab="Dependent", main="Random Stuff")

x1 <- runif(8,15,25)
y1 <- 2.5*x1 - 1.0 + runif(8,-6,6)
??

x2 <- runif(8,15,25)
y2 <- 2.5*x2 - 1.0 + runif(8,-6,6)
??
```

Scatter plot: add points of different colours and symbols

```
points(x, y = NULL, type = "p", ...)
```

```
x <- rnorm(10,sd=5,mean=20)
y <- 2.5*x - 1.0 + rnorm(10,sd=9,mean=0)
plot(x,y,xlab="Independent", ylab="Dependent", main="Random Stuff")
x1 <- runif(8,15,25)
y1 <- 2.5*x1 - 1.0 + runif(8,-6,6)
points(x1,y1,col=2)

x2 <- runif(8,15,25)
y2 <- 2.5*x2 - 1.0 + runif(8,-6,6)
points(x2,y2,col=3,pch=2)
```

Different types of lines

```
par(..., no.readonly = FALSE)
```

```
x <- c(1:5); y <- x
par(pch=22, col="red")
par(mfrow=c(2,4))
opts = ???
for(i in 1:length(opts)){
heading = paste("type=",opts[i])
```

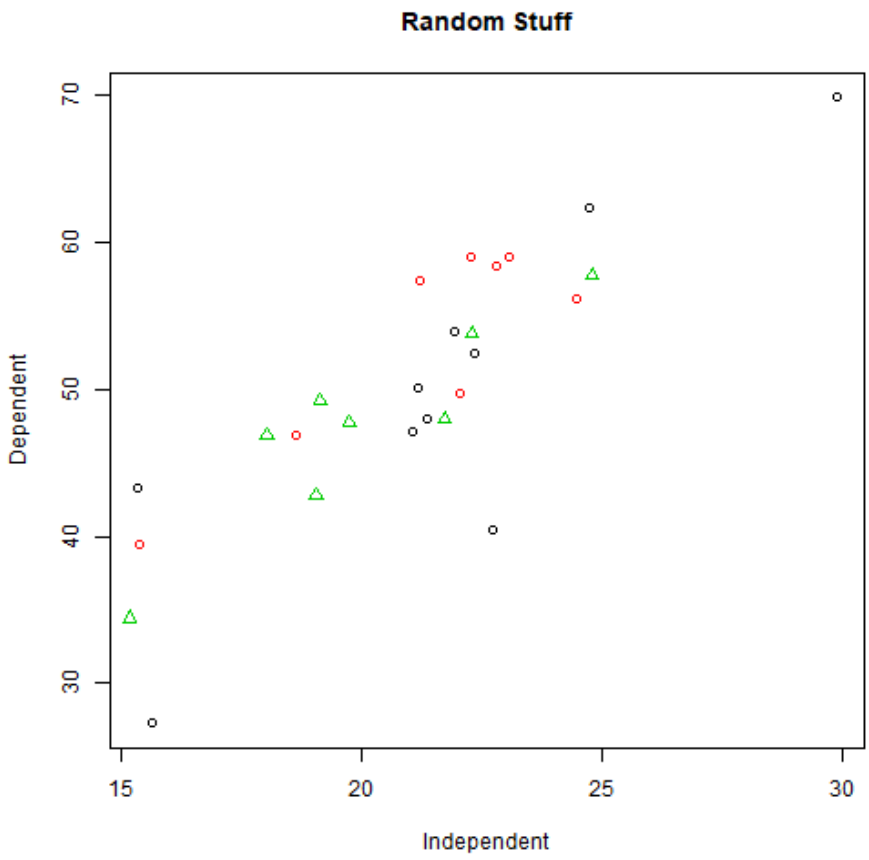

Figure 1: plot of chunk unnamed-chunk-2

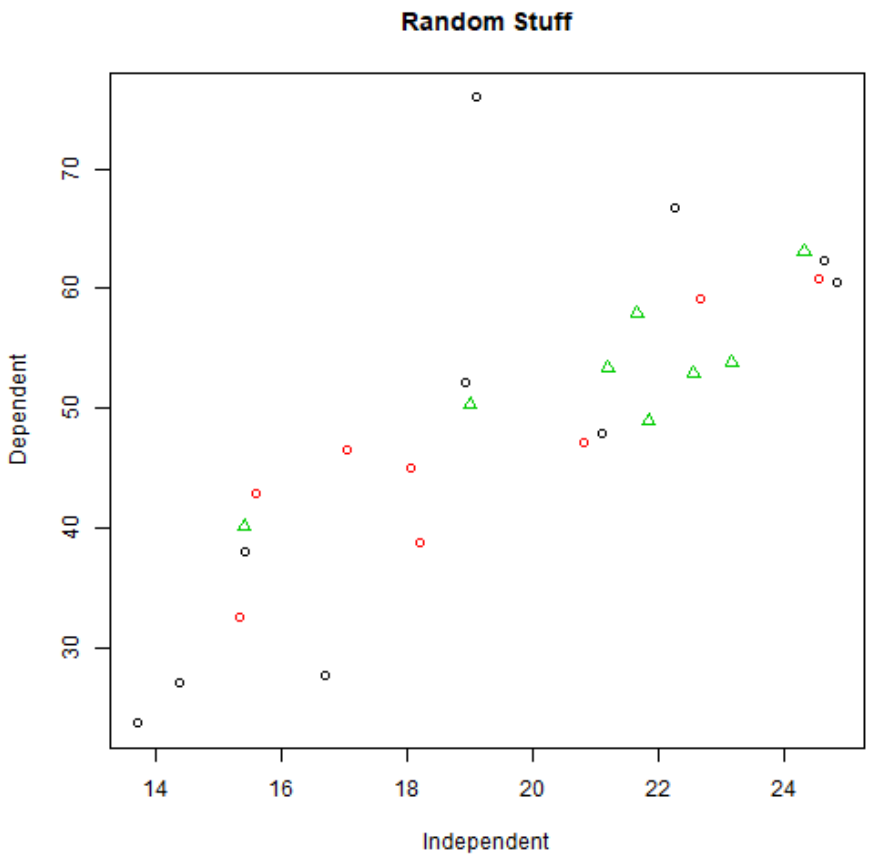

Figure 2: plot of chunk unnamed-chunk-3

```
plot(x, y, type="n", main=heading)
lines(x, y, type=opts[i])
}
```

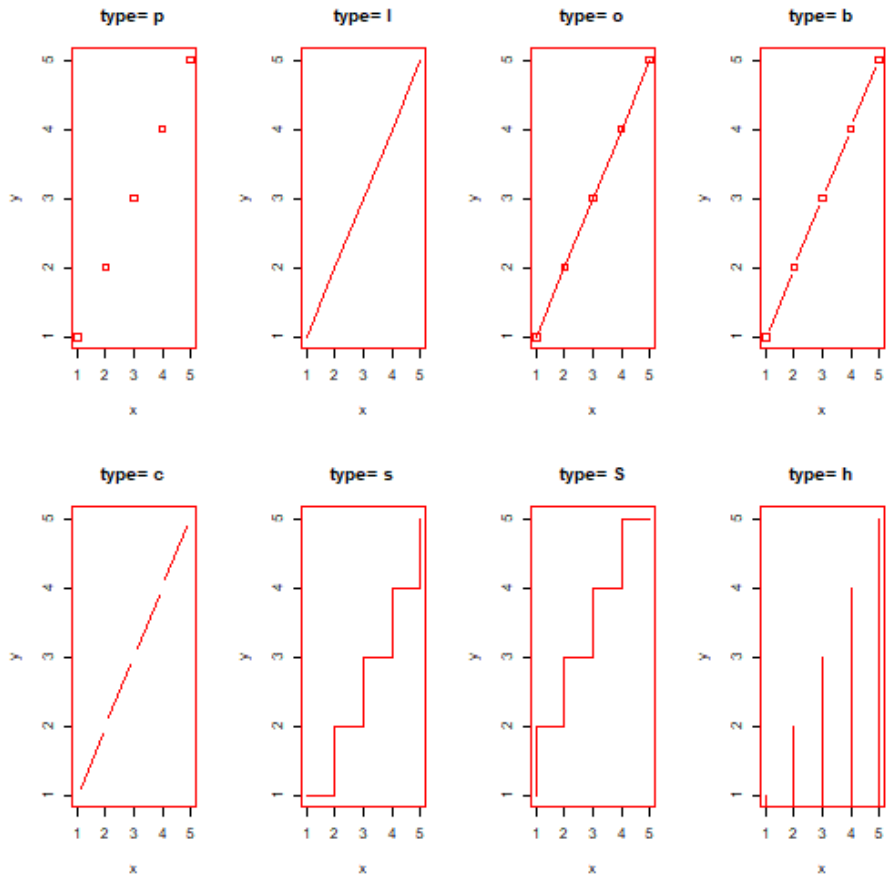

Figure 3: plot of chunk unnamed-chunk-5

Different types of lines

```
x <- c(1:5); y <- x
par(pch=22, col="red")
par(mfrow=c(2,4))
opts = c("p","l","o","b","c","s","S","h")
for(i in 1:length(opts)){
heading = paste("type=",opts[i])
```

```
plot(x, y, type="n", main=heading)
lines(x, y, type=opts[i])
}
```

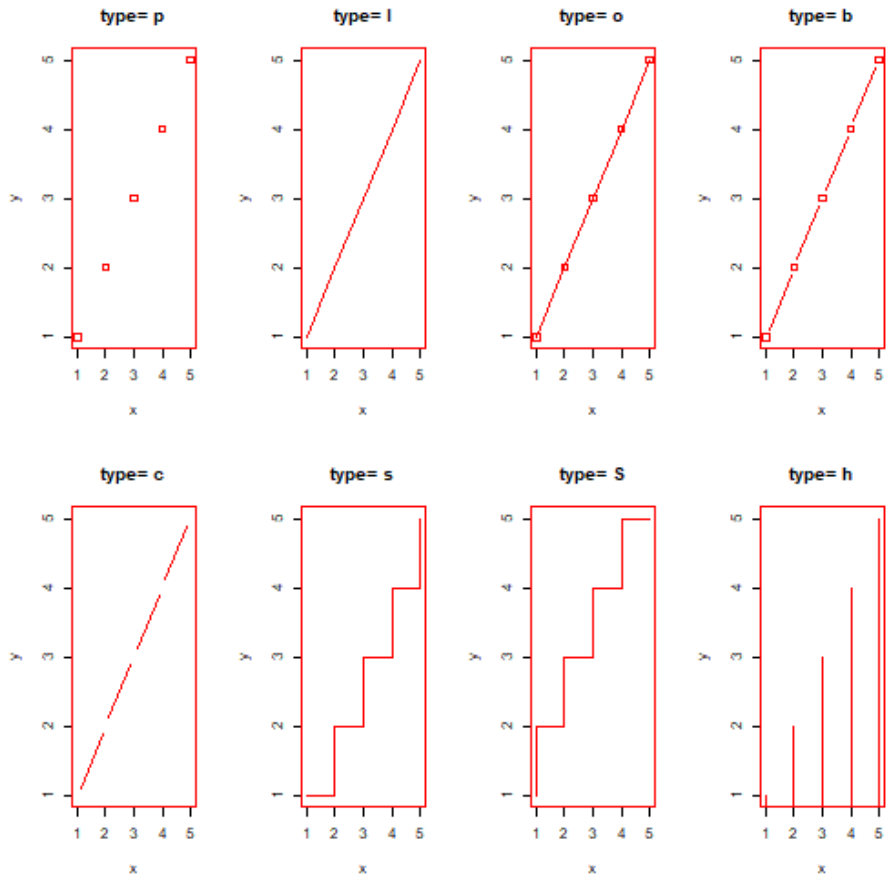

Figure 4: plot of chunk unnamed-chunk-6

Scatter plot: legend

'legend(x, y = NULL, legend, fill = NULL, col = par("col"), border = "black", lty, lwd, pch, angle = 45, density = NULL, bty = "o", bg = par("bg"), box.lwd = par("lwd"), box.lty = par("lty"), box.col = par("fg"), pt.bg = NA, cex = 1, pt.cex = cex, pt.lwd = lwd, xjust = 0, yjust = 1, x.intersp = 1, y.intersp = 1, adj = c(0, 0.5), text.width = NULL, text.col = par("col"), text.font = NULL, merge = do.lines && has.pch, trace = FALSE, plot = TRUE, ncol = 1, horiz =

FALSE, title = NULL, inset = 0, xpd, title.col = text.col, title.adj = 0.5, seg.len = 2)'

```
plot(x,y,xlab="Independent",ylab="Dependent",main="Random Stuff")
points(x1,y1,col=2,pch=3)
points(x2,y2,col=4,pch=5)
???
```

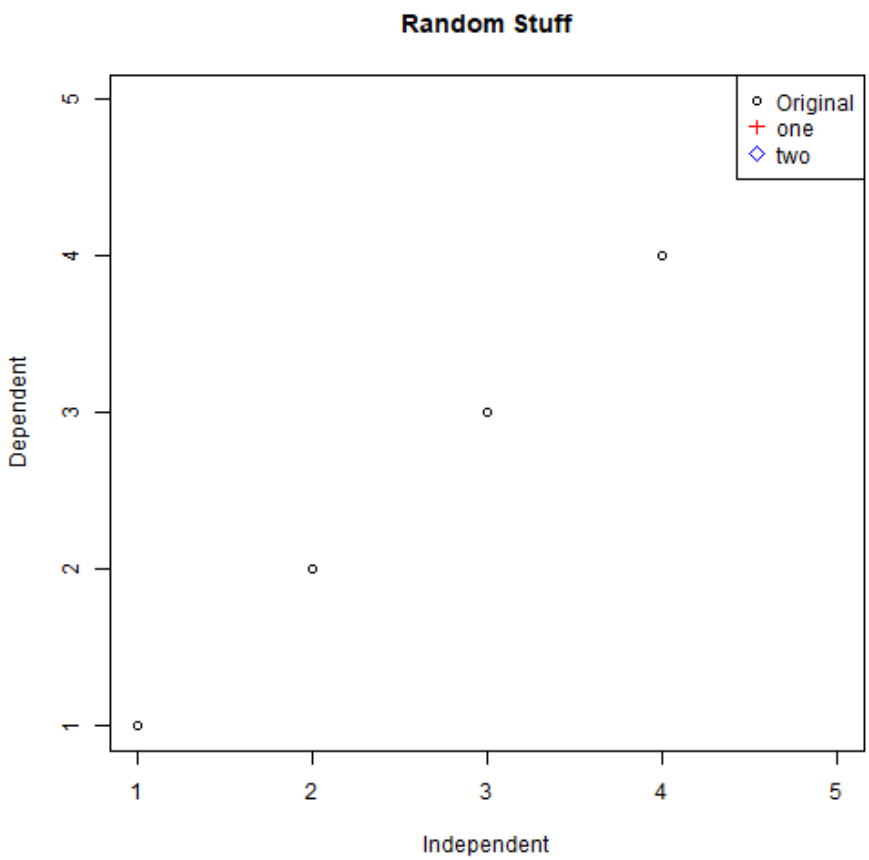

Figure 5: plot of chunk unnamed-chunk-8

Scatter plot: legend

```
plot(x,y,xlab="Independent",ylab="Dependent",main="Random Stuff")
points(x1,y1,col=2,pch=3)
```

```
points(x2,y2,col=4,pch=5)
legend("topright", c("Original","one","two"),col=c(1,2,4),pch=c(1,3,5))
legend(20, 50, c("Original","one","two"),col=c(1,2,4),pch=c(1,3,5))
```

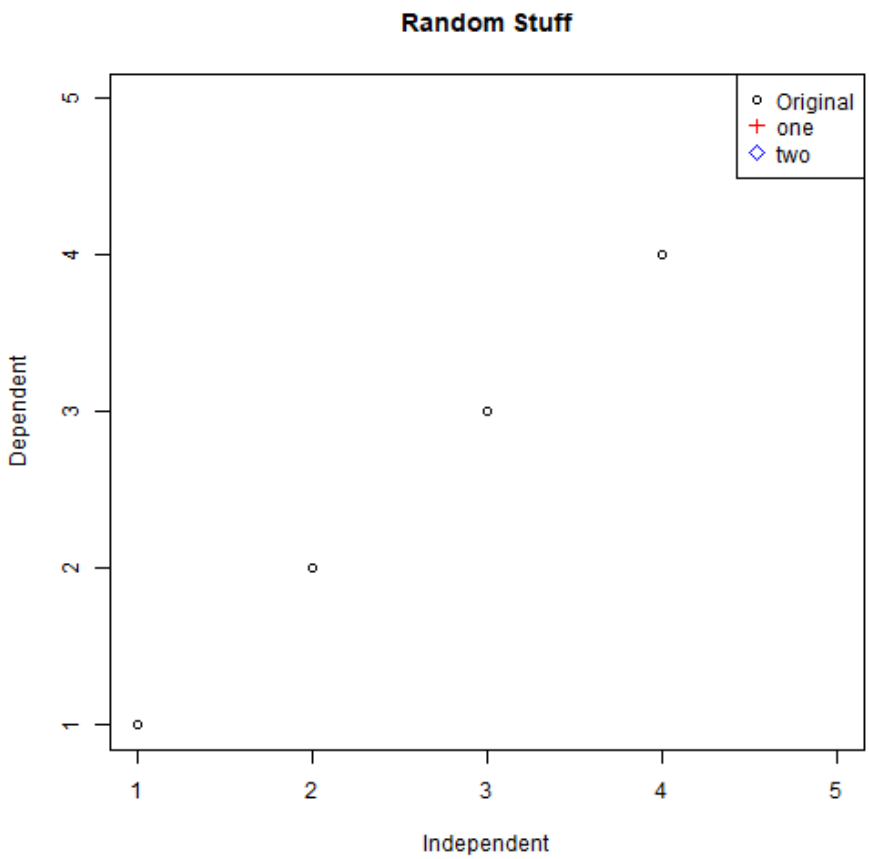

Figure 6: plot of chunk unnamed-chunk-9

Simple fitted lines

```
abline(a = NULL, b = NULL, h = NULL, v = NULL, reg = NULL,        coef
= NULL, untf = FALSE, ...)
```

```
lm(formula, data, subset, weights, na.action,    method = "qr",
model = TRUE, x = FALSE, y = FALSE, qr = TRUE,    singular.ok =
TRUE, contrasts = NULL, offset, ...)
```

```
lowess(x, y = NULL, f = 2/3, iter = 3, delta = 0.01 * diff(range(x)))
```

```
x <- rnorm(10,sd=5,mean=20)
y <- 2.5*x - 1.0 + rnorm(10,sd=9,mean=0)
x2 <- runif(8,15,25)
y2 <- 2.5*x2 - 1.0 + runif(8,-6,6)

plot(x,y,xlab="Independent",ylab="Dependent",main="Random Stuff")
points(x1,y1,col=2,pch=3)
points(x2,y2,col=4,pch=5)
legend(25,80,c("Original","one","two"),col=c(1,2,4),pch=c(1,3,5))

???
```

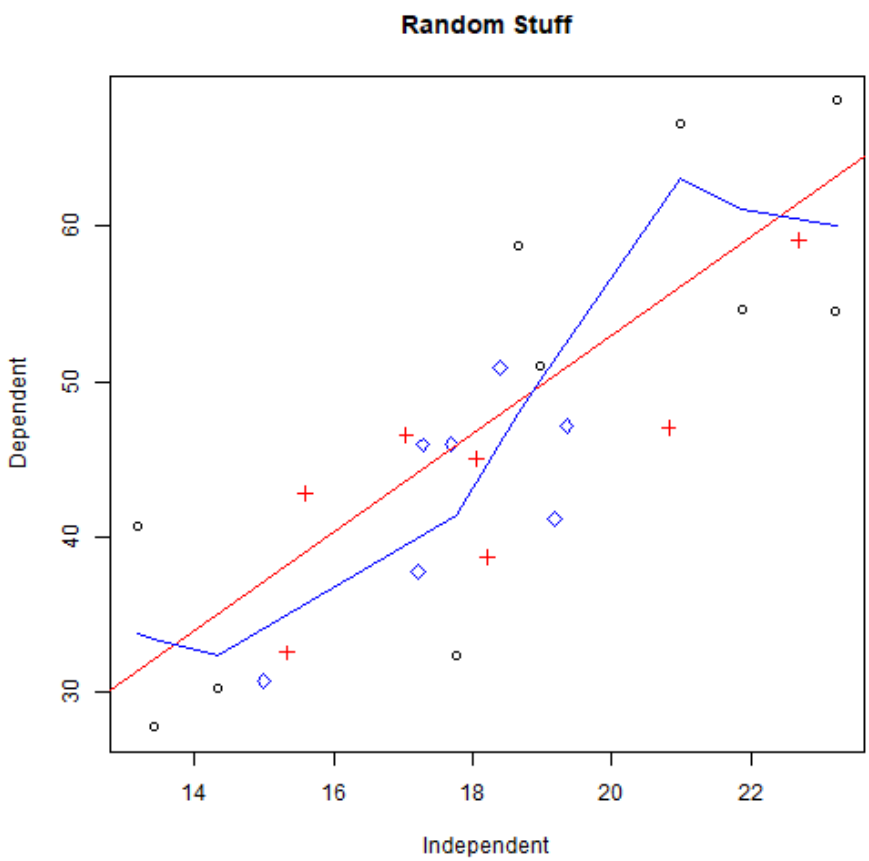

Figure 7: plot of chunk unnamed-chunk-11

Simple fitted lines

```
x <- rnorm(10,sd=5,mean=20)
y <- 2.5*x - 1.0 + rnorm(10,sd=9,mean=0)
x2 <- runif(8,15,25)
y2 <- 2.5*x2 - 1.0 + runif(8,-6,6)

plot(x,y,xlab="Independent",ylab="Dependent",main="Random Stuff")
points(x1,y1,col=2,pch=3)
points(x2,y2,col=4,pch=5)
legend(25,80,c("Original","one","two"),col=c(1,2,4),pch=c(1,3,5))

abline(lm(y~x), col="red")
lines(lowess(x,y), col="blue")
```

With error lines

```
arrows(x0, y0, x1 = x0, y1 = y0, length = 0.25, angle = 30,
code = 2, col = par("fg"), lty = par("lty"),        lwd = par("lwd"),
...)
```

```
plot(x,y,xlab="Independent",ylab="Dependent",main="Random Stuff")
xHigh <- x
yHigh <- y + abs(rnorm(10,sd=3.5))
xLow <- x
yLow <- y - abs(rnorm(10,sd=3.1))
???
```

With error lines

```
plot(x,y,xlab="Independent",ylab="Dependent",main="Random Stuff")
xHigh <- x
yHigh <- y + abs(rnorm(10,sd=3.5))
xLow <- x
yLow <- y - abs(rnorm(10,sd=3.1))
arrows(xHigh,yHigh,xLow,yLow,col=2,angle=90,length=0.1,code=3)
```

Adding jitter

```
numberWhite <- ??
numberChipped <- ??
```

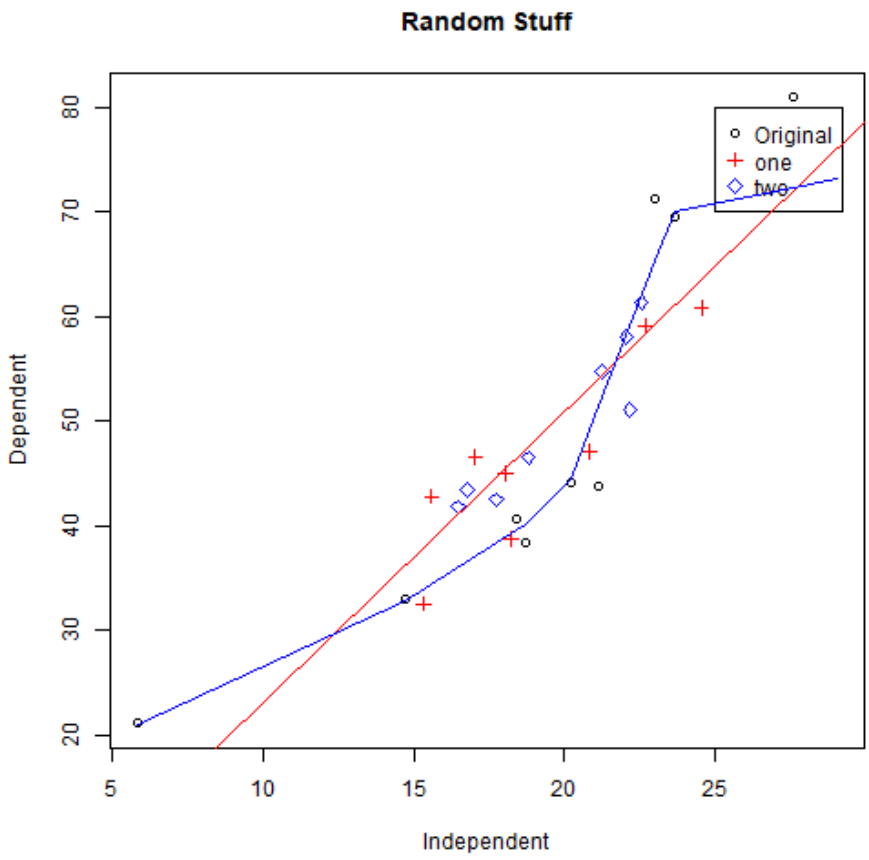

Figure 8: plot of chunk unnamed-chunk-12

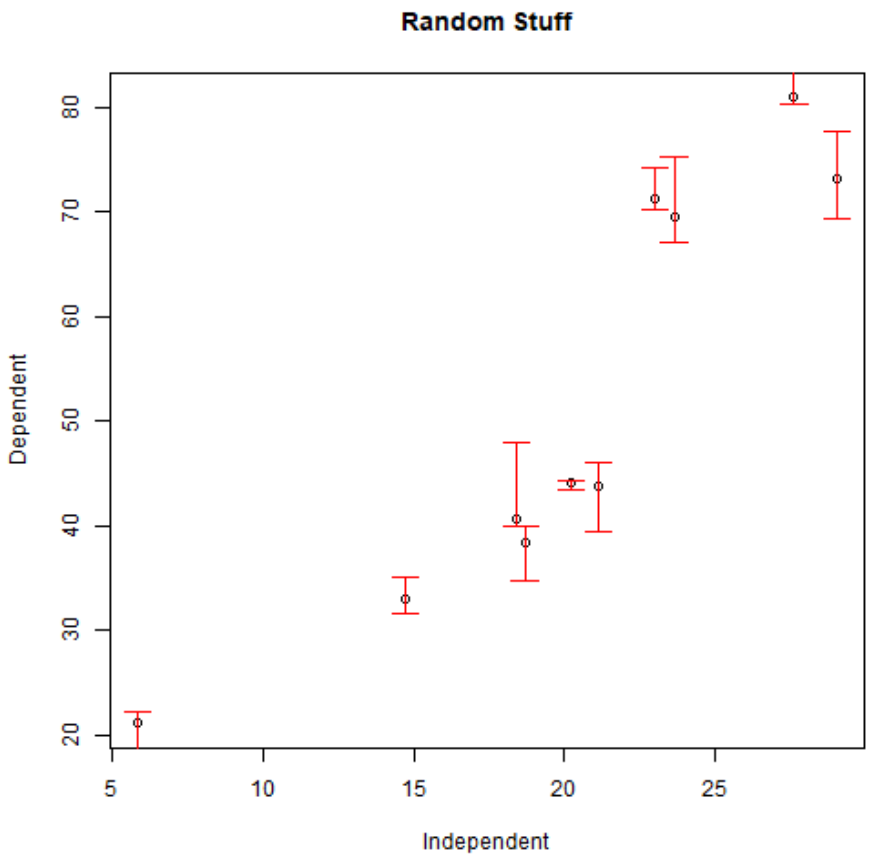

Figure 9: plot of chunk unnamed-chunk-14

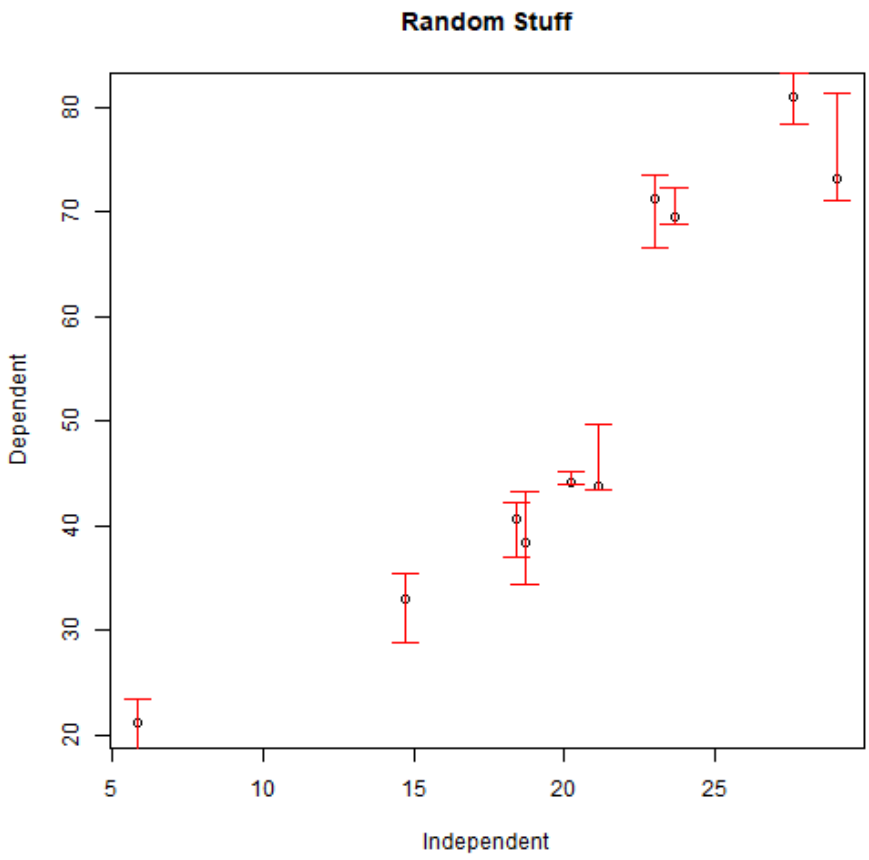

Figure 10: plot of chunk unnamed-chunk-15

```r
par(mfrow=c(1,2))
plot(numberWhite,numberChipped,xlab="Number White Marbles Drawn",
ylab="Number Chipped Marbles Drawn",main="Pulling Marbles")
plot(jitter(numberWhite),jitter(numberChipped),xlab="Number White Marbles Drawn",
ylab="Number Chipped Marbles Drawn",main="Pulling Marbles With Jitter")
```

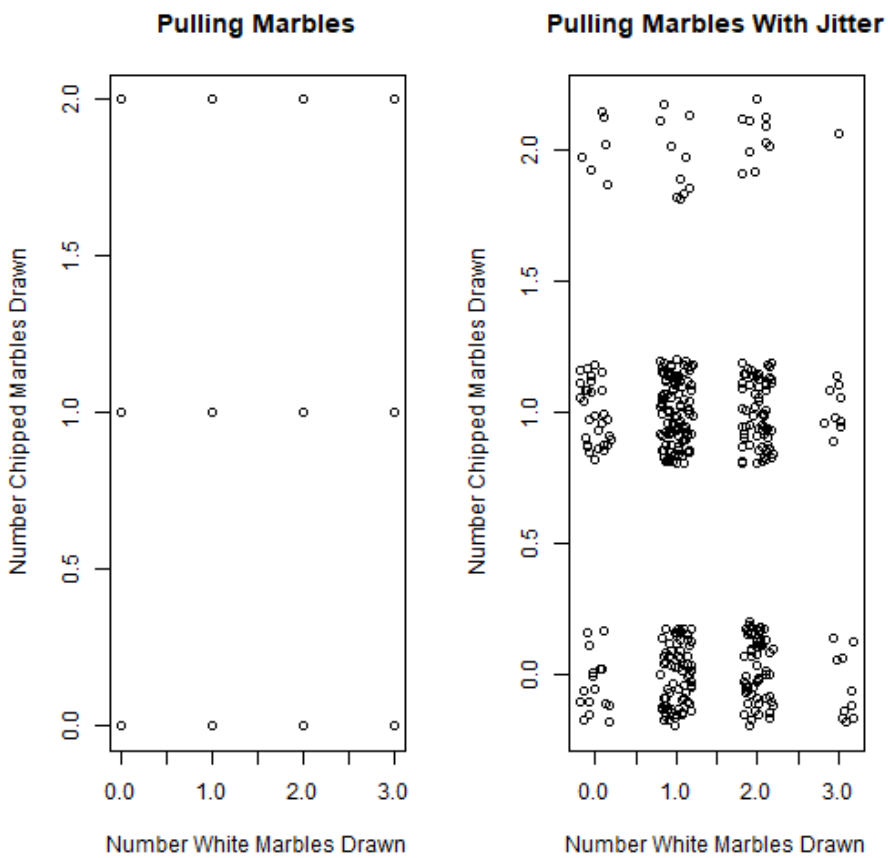

Figure 11: plot of chunk unnamed-chunk-17

Adding jitter

```r
numberWhite <- rhyper(400,4,5,3)
numberChipped <- rhyper(400,2,7,3)
par(mfrow=c(1,2))
```

```r
plot(numberWhite,numberChipped,xlab="Number White Marbles Drawn",
ylab="Number Chipped Marbles Drawn",main="Pulling Marbles")
plot(jitter(numberWhite),jitter(numberChipped),xlab="Number White Marbles Drawn",
ylab="Number Chipped Marbles Drawn",main="Pulling Marbles With Jitter")
```

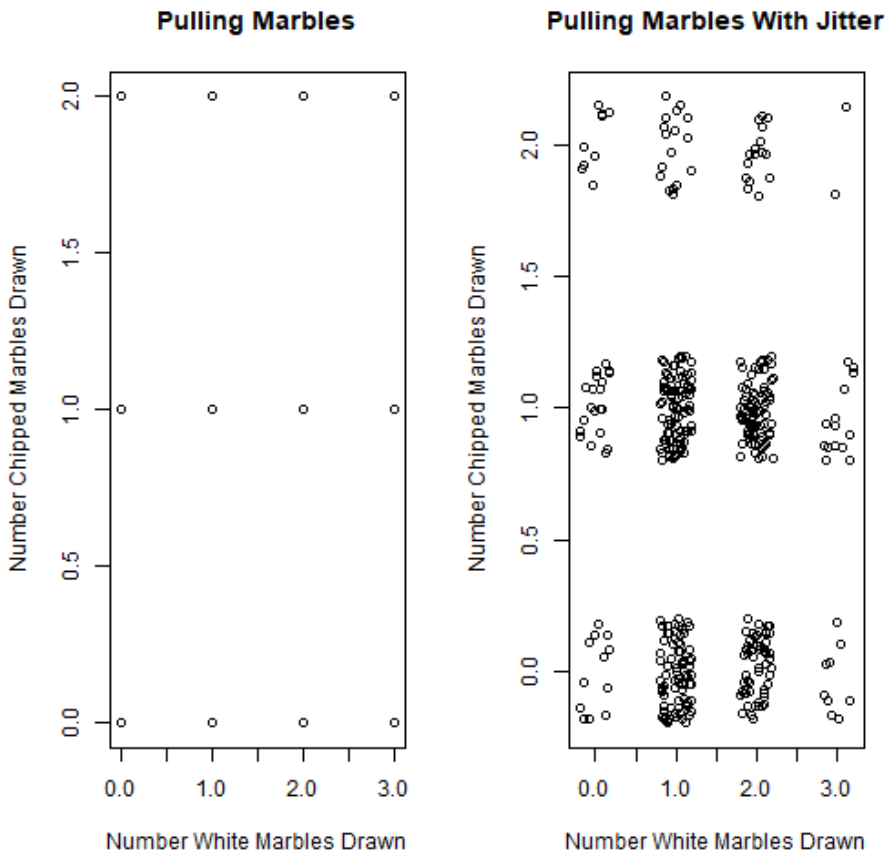

Figure 12: plot of chunk unnamed-chunk-18

Mosaic plots

```
table(...,       exclude = if (useNA == "no") c(NA, NaN),       useNA
= c("no", "ifany", "always"),       dnn = list.names(...), deparse.level
= 1)
```

```r
mosaicplot(??, main="sixth plot")
```

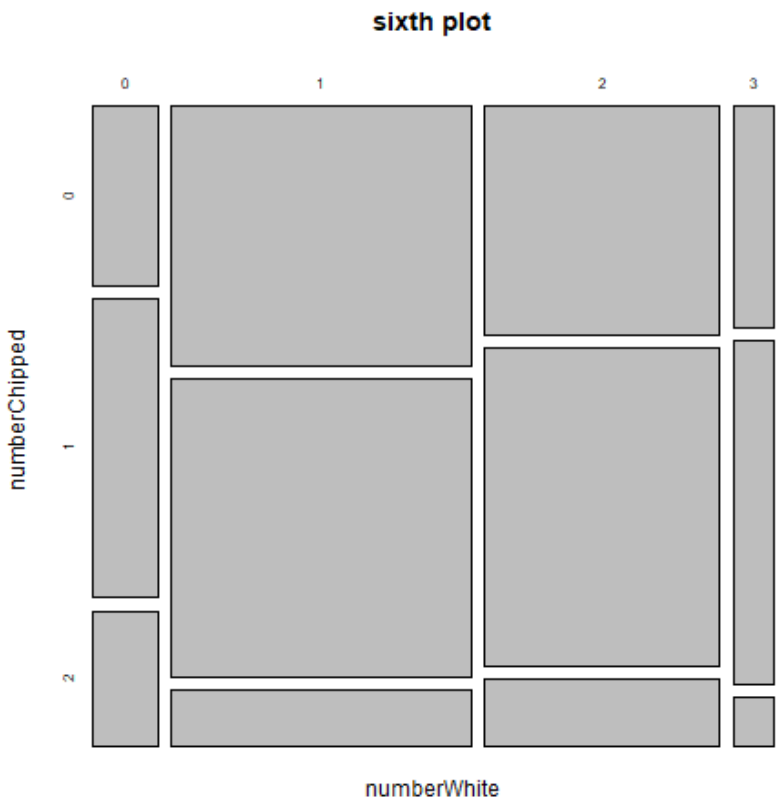

Figure 13: plot of chunk unnamed-chunk-20

Mosaic plots

```
mosaicplot(table(numberWhite,numberChipped),main="sixth plot")
```

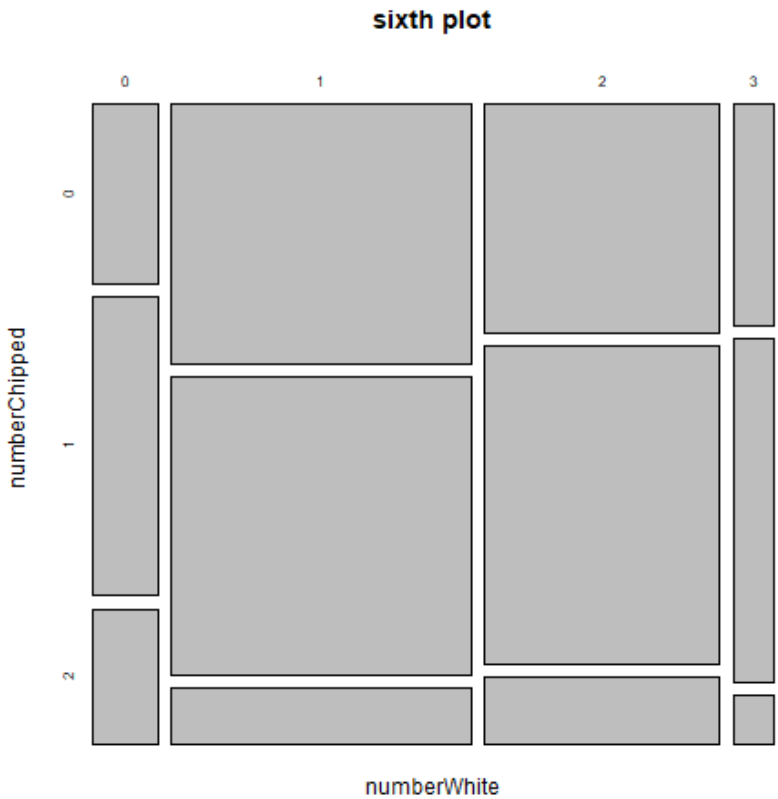

Figure 14: plot of chunk unnamed-chunk-21

Pair-wise scatter plots

```
uData <- rnorm(20)
vData <- rnorm(20,mean=5)
wData <- uData + 2*vData + rnorm(20,sd=0.5)
xData <- -2*uData+rnorm(20,sd=0.1)
yData <- 3*vData+rnorm(20,sd=2.5)
```

```r
pairs(?)
```

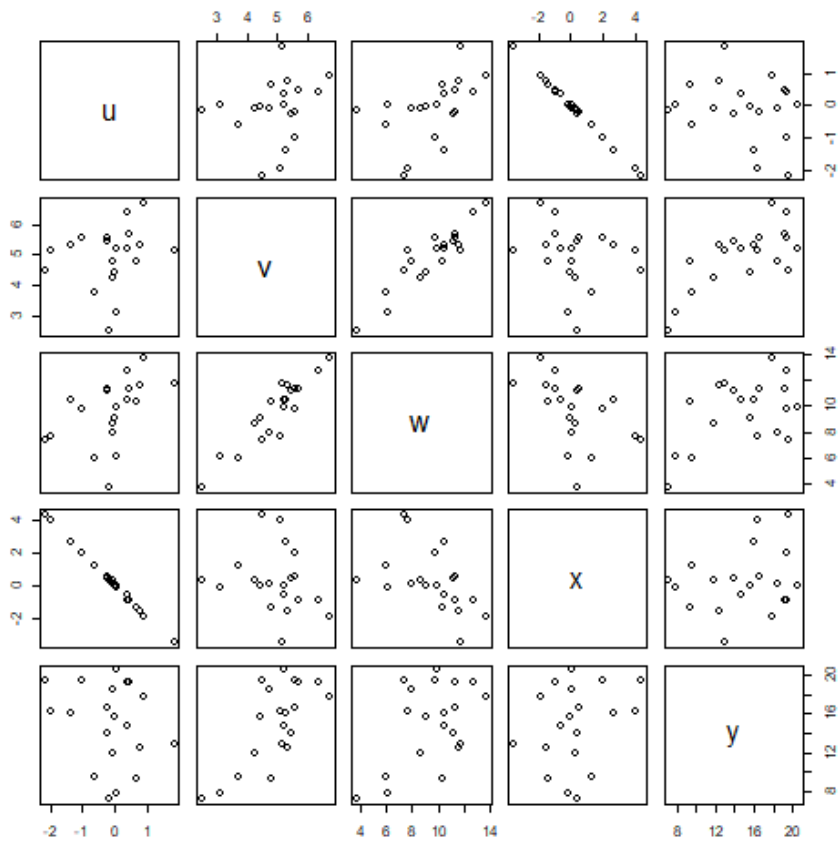

Figure 15: plot of chunk unnamed-chunk-23

Pair-wise scatter plots

```r
uData <- rnorm(20)
vData <- rnorm(20,mean=5)
wData <- uData + 2*vData + rnorm(20,sd=0.5)
xData <- -2*uData+rnorm(20,sd=0.1)
yData <- 3*vData+rnorm(20,sd=2.5)
d <- data.frame(u=uData,v=vData,w=wData,x=xData,y=yData)
pairs(d)
```

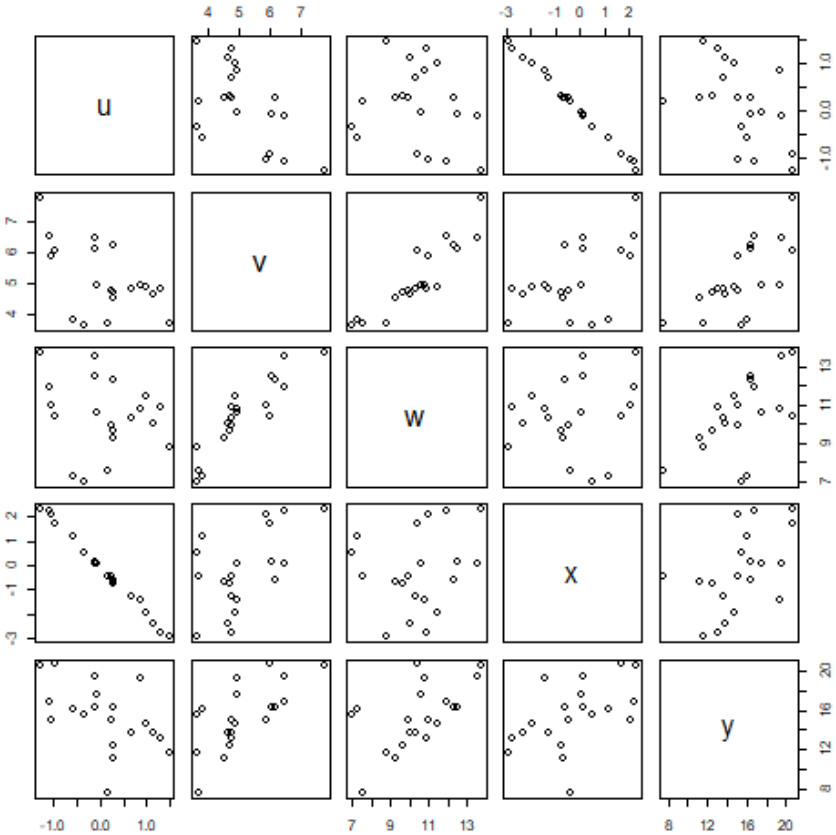

Figure 16: plot of chunk unnamed-chunk-24

Shaded areas

```
polygon(x, y = NULL, density = NULL, angle = 45,         border
= NULL, col = NA, lty = par("lty"),         ..., fillOddEven =
FALSE)
```

```
stdDev <- 0.75; x <- seq(-5,5,by=0.01); y <- dnorm(x,sd=stdDev)
right <- qnorm(0.95,sd=stdDev)
plot(x,y,type="l",xaxt="n",ylab="p",xlab=expression(paste('Assumed Distribution of',bar(x)))
axis(1,at=c(-5,right,0,5), pos = c(0,0),labels=c(expression(' '),expression(bar(x)[cr]),expr
axis(2)
xReject <- seq(right,5,by=0.01);  yReject <- dnorm(xReject,sd=stdDev)
polygon(???, col='red')
```

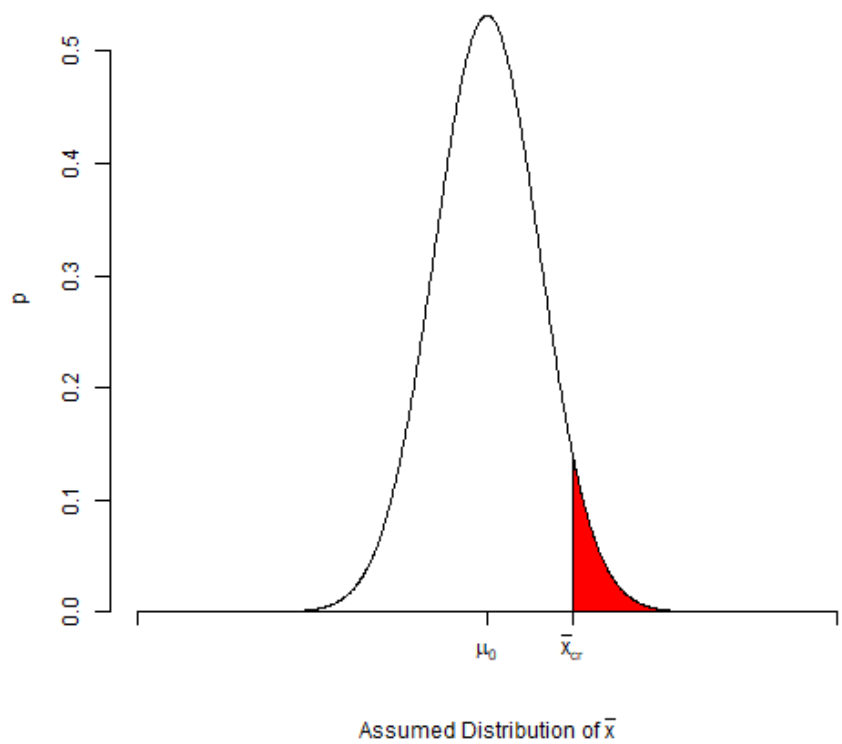

Figure 17: plot of chunk unnamed-chunk-26

Shaded areas

```
stdDev <- 0.75;
x <- seq(-5,5,by=0.01)
y <- dnorm(x,sd=stdDev)
right <- qnorm(0.95,sd=stdDev)
plot(x,y,type="l",xaxt="n",ylab="p",xlab=expression(paste('Assumed Distribution of ',bar(x))
axis(1,at=c(-5,right,0,5),       pos = c(0,0),   labels=c(expression(' '),expression(bar(x)
axis(2)
xReject <- seq(right,5,by=0.01);  yReject <- dnorm(xReject,sd=stdDev)
polygon(c(xReject,xReject[length(xReject)],xReject[1]),  c(yReject,0, 0), col='red')
```

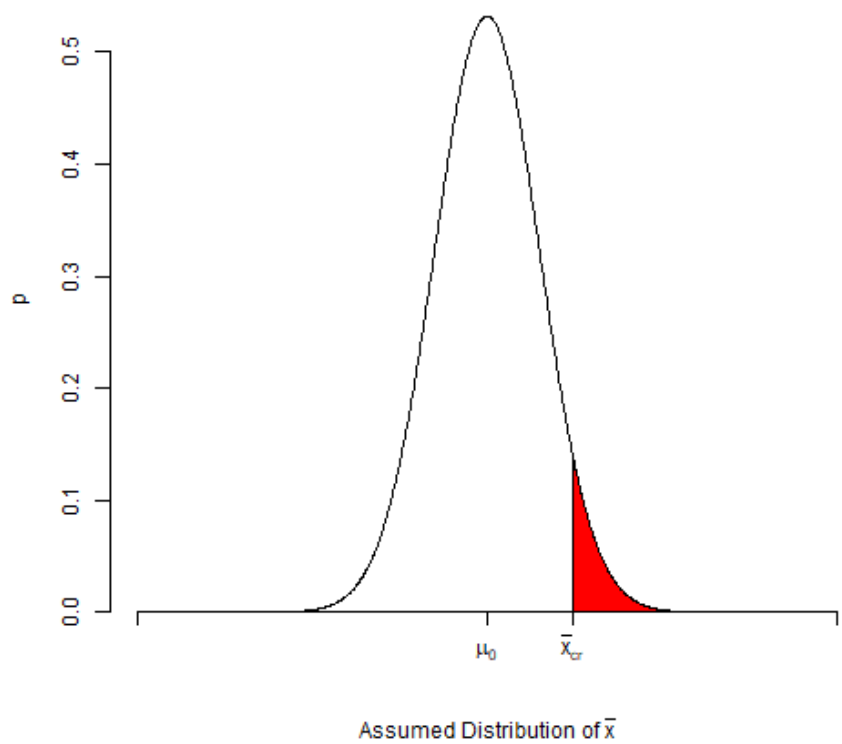

Figure 18: plot of chunk unnamed-chunk-27

Barplot

```
barplot(height, width = 1, space = NULL,         names.arg =
NULL, legend.text = NULL, beside = FALSE,         horiz = FALSE,
density = NULL, angle = 45,         col = NULL, border = par("fg"),
main = NULL, sub = NULL, xlab = NULL, ylab = NULL,         xlim
= NULL, ylim = NULL, xpd = TRUE, log = "",         axes = TRUE,
axisnames = TRUE,         cex.axis = par("cex.axis"), cex.names =
par("cex.axis"),         inside = TRUE, plot = TRUE, axis.lty =
0, offset = 0,         add = FALSE, args.legend = NULL, ...)
```

```
numberWhite <- rhyper(30,4,5,3)
numberWhite <- as.factor(numberWhite)
```

```
???
```

```
numberWhite
 0  1  2  3
 3 12 13  2
```

Barplot

```
numberWhite <- rhyper(30,4,5,3)
numberWhite <- as.factor(numberWhite)

totals <- table(numberWhite)
totals
```

```
numberWhite
 0  1  2
 3 16 11
```

```
barplot(totals,main="Number Draws",ylab="Frequency",xlab="Draws")
```

plotly

plotly website here

ggplot2: structure

```
ggplot(data = <default data set>,        aes(x = <default x axis
variable>,            y = <default y axis variable>,            ...
```

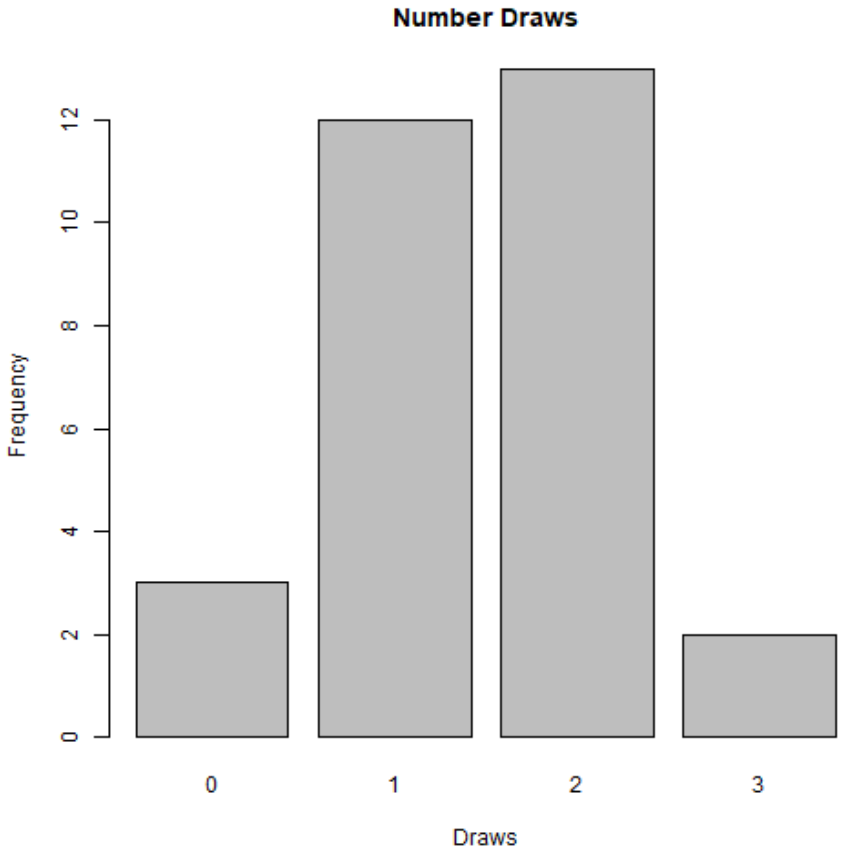

Figure 19: plot of chunk unnamed-chunk-29

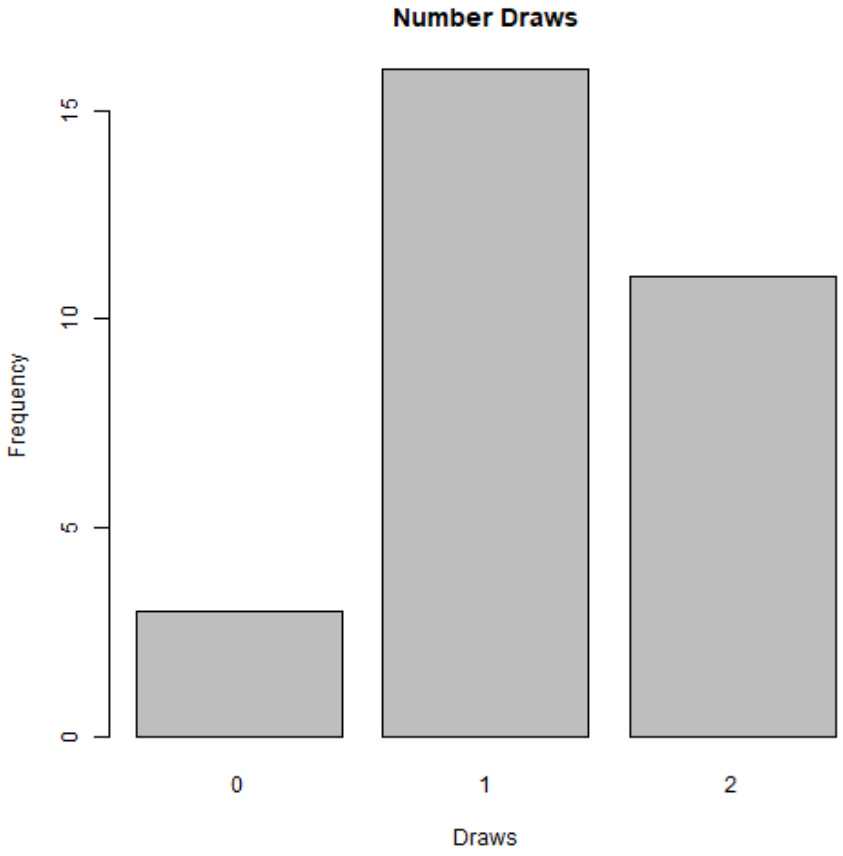

Figure 20: plot of chunk unnamed-chunk-30

```
<other default aesthetic mappings>),        ... <other plot
defaults>) +        geom_<geom type>(aes(size = <size variable
for this geom>,                        ... <other aesthetic mappings>),
data = <data for this point geom>,                    stat =
<statistic string or function>,                   position =
<position string or function>,                   color = <"fixed
color specification">,                   <other arguments, possibly
passed to the _stat_ function) +   scale_<aesthetic>_<type>(name
= <"scale label">,                       breaks = <where to put
tick marks>,                       labels = <labels for tick
marks>,                       ... <other options for the scale>) +
theme(plot.background = element_rect(fill = "gray"),         ...
<other theme elements>)
```

qplot: overlapping densities

```
qplot(x, y = NULL, ..., data, facets = NULL, margins = FALSE,
geom = "auto", xlim = c(NA, NA), ylim = c(NA, NA), log = "",
main = NULL, xlab = deparse(substitute(x)),   ylab = deparse(substitute(y)),
asp = NA, stat = NULL, position = NULL)
```

```r
library(ggplot2)
data("mtcars")
mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5), labels=c("3gears","4gears","5gears"))
mtcars$am <- factor(mtcars$am,levels=c(0,1), labels=c("Automatic","Manual"))
mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8), labels=c("4cyl","6cyl","8cyl"))

qplot(mpg,
      data = ??,
      geom = ??,
      fill = ??,
      alpha = ??,
      main = "Distribution of Gas Milage", xlab="Miles Per Gallon", ylab="Density")
```

```
Error in loadNamespace(j <- i[[1L]], c(lib.loc, .libPaths()), versionCheck = vI[[j]]) :
  there is no package called 'tibble'
In addition: Warning message:
package 'ggplot2' was built under R version 3.2.5
Quitting from lines 521-533 (R-plots-msc.Rpres)
Error: package or namespace load failed for 'ggplot2'
Execution halted
```