Statistical Programming Languages – Day 3

Uwe Ziegenhagen

Institut für Statistik and Ökonometrie
Humboldt-Universität zu Berlin
http://www.uweziegenhagen.de

Agenda for Today

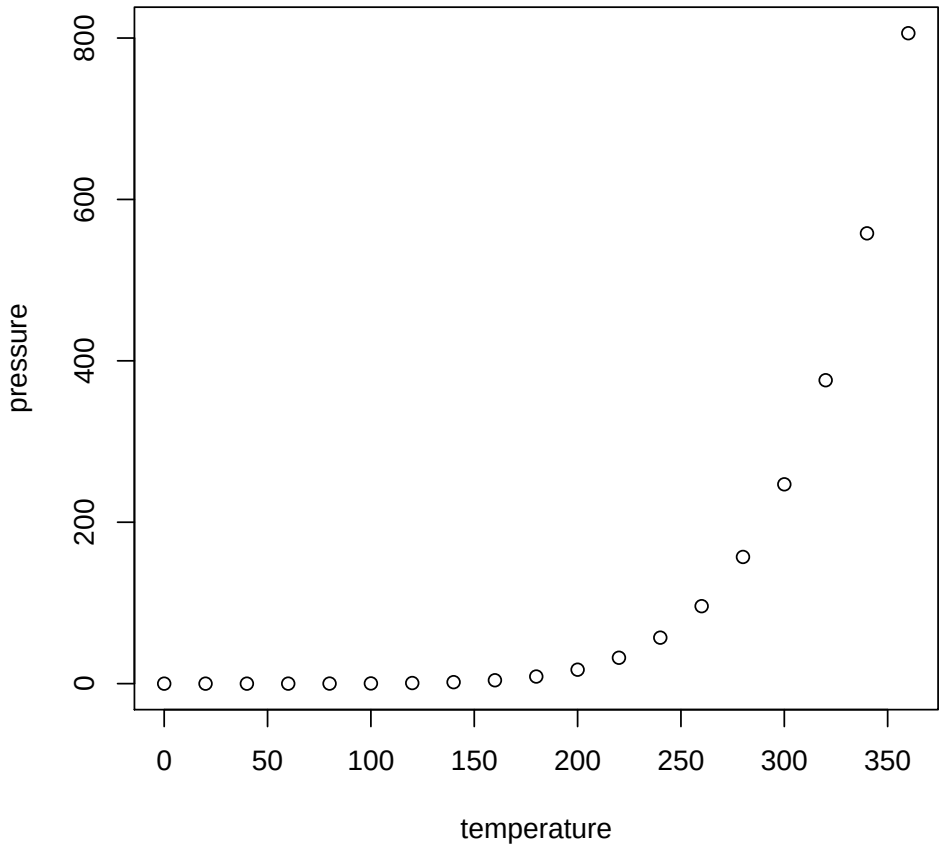
Plotting in ***R***

- Plotting
- high-level plots
- low-level plots
- base-graphics, grid-graphics
- etc. . .

Basic Plot Example

```
library("datasets")
data("pressure") # data(pressure) works, too
plot(pressure)
# alternative
with(pressure, plot(temperature,pressure))
```

Plot example



R Graphics Devices

Output to PNG, jpg() or bmp() exist, too.

```
1 png(filename = "Rplot%03d.png", width = 480,
2 height = 480, units = "px", pointsize = 12,
3 bg = "white", res = NA, restoreConsole = TRUE)
```

Output to PDF:

```
1 pdf(file = "plot3d%03d.pdf",width = 6, height = 6,
2 onefile = FALSE, family = "Helvetica",
3 title = "R Graphics Output", fonts = NULL,
4 version = "1.4",paper = "special")
```

R Graphics Devices

Other graphics devices:

- x11()/X11(), windows(), quartz() for Screen
- postscript(), pictex(), xfig(), win.metafile()
- devGTK(), devJava(), devSVG()

`dev.off()` closes devices sequentially.

Labels and Axes

R uses the variable names for axes labels and computes range for axes. Manual override by

- axes labels: `xlab`, `ylab`
- size of labels: `cex.lab`
- axes range: `xlim`, `ylim`

Try these parameters for `women` data!

Symbols, colors, sizes for points

- point symbol: pch
- col: color
- cex: size factor

Try these parameters for `women` data! Which colors and symbols are available?

The `Hmisc` package

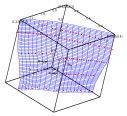
- install `Hmisc` package
- library("Hmisc")
- show.col()
- show.pch()

Plot example

	0		10		20		30		40		50		60		70		80		90
	1		11		21		31		41		51		61		71		81		91
	2		12		22		32		42		52		62		72		82		92
	3		13		23		33		43		53		63		73		83		93
	4		14		24		34		44		54		64		74		84		94
	5		15		25		35		45		55		65		75		85		95
	6		16		26		36		46		56		66		76		86		96
	7		17		27		37		47		57		67		77		87		97
	8		18		28		38		48		58		68		78		88		98
	9		19		29		39		49		59		69		79		89		99

Plot example

□	0	▽	25	2	50	K	75	d	100	}	125	–	150	¯	175	È	200	á	225	ú	250
○	1		26	3	51	L	76	e	101	˜	126	—	151	°	176	É	201	â	226	û	251
△	2		27	4	52	M	77	f	102	•	127	˜	152	±	177	Ê	202	ã	227	ü	252
┼	3		28	5	53	N	78	g	103	€	128	™	153	²	178	Ë	203	ä	228	ý	253
×	4		29	6	54	O	79	h	104	•	129	š	154	³	179	Ì	204	å	229		
◇	5		30	7	55	P	80	i	105	‚	130	›	155	´	180	Í	205	æ	230		
▽	6		31	8	56	Q	81	j	106	ƒ	131	œ	156	µ	181	Î	206	ç	231		
⊠	7		32	9	57	R	82	k	107	„	132	•	157	¶	182	Ï	207	è	232		
✳	8	!	33	:	58	S	83	l	108	…	133	ž	158	·	183	Ð	208	é	233		
⊕	9	"	34	;	59	T	84	m	109	†	134	Ÿ	159	¸	184	Ñ	209	ê	234		
⊕	10	#	35	<	60	U	85	n	110	‡	135		160	¹	185	Ò	210	ë	235		
✡	11	$	36	=	61	V	86	o	111	ˆ	136	¡	161	º	186	Ó	211	ì	236		
⊞	12	%	37	>	62	W	87	p	112	‰	137	¢	162	»	187	Ô	212	í	237		
⊠	13	&	38	?	63	X	88	q	113	Š	138	£	163	¼	188	Õ	213	î	238		
⊠	14	'	39	@	64	Y	89	r	114	‹	139	¤	164	½	189	Ö	214	ï	239		
■	15	(	40	A	65	Z	90	s	115	Œ	140	¥	165	¾	190	×	215	ð	240		
●	16	)	41	B	66	[	91	t	116	•	141	¦	166	¿	191	Ø	216	ñ	241		
▲	17	*	42	C	67	\	92	u	117	Ž	142	§	167	À	192	Ù	217	ò	242		
◆	18	+	43	D	68	]	93	v	118	•	143	¨	168	Á	193	Ú	218	ó	243		
●	19	,	44	E	69	^	94	w	119	•	144	©	169	Â	194	Û	219	ô	244		
•	20	–	45	F	70	_	95	x	120	‘	145	ª	170	Ã	195	Ü	220	õ	245		
○	21	.	46	G	71	`	96	y	121	’	146	«	171	Ä	196	Ý	221	ö	246		
□	22	/	47	H	72	a	97	z	122	“	147	¬	172	Å	197	Þ	222	÷	247		
◇	23	0	48	I	73	b	98	{	123	”	148	-	173	Æ	198	ß	223	ø	248		
△	24	1	49	J	74	c	99	|	124	•	149	®	174	Ç	199	à	224	ù	249		

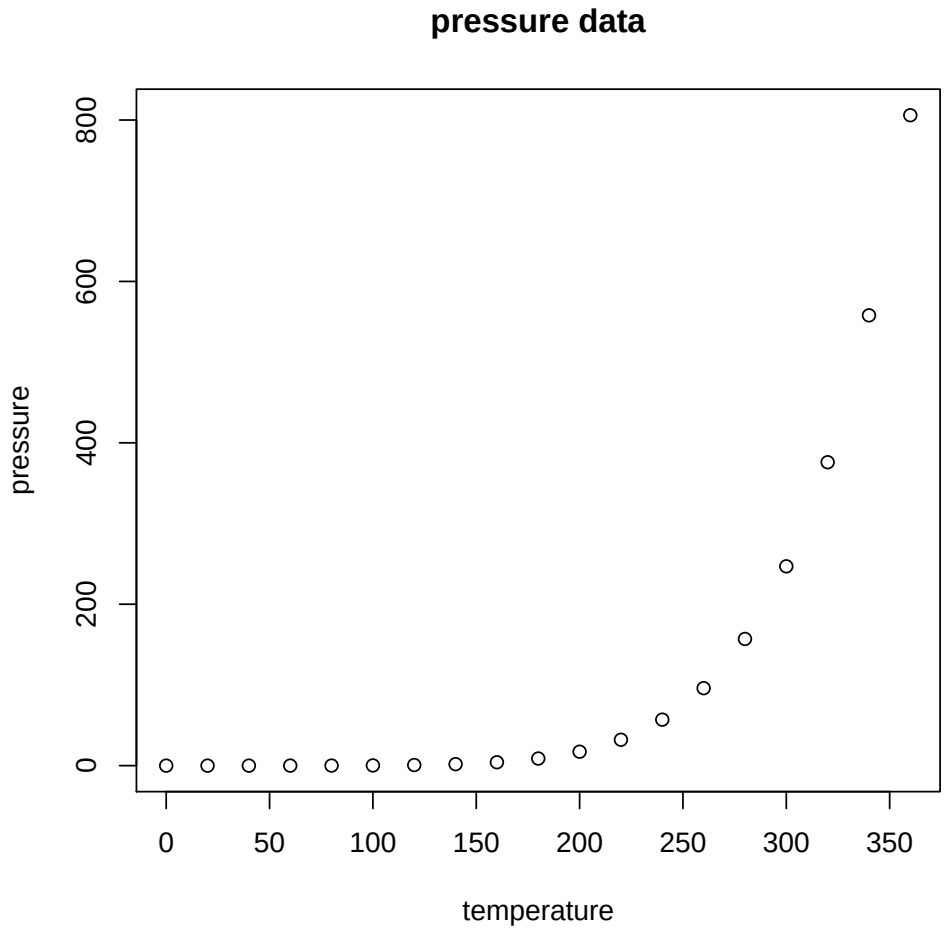


Titles and Lines

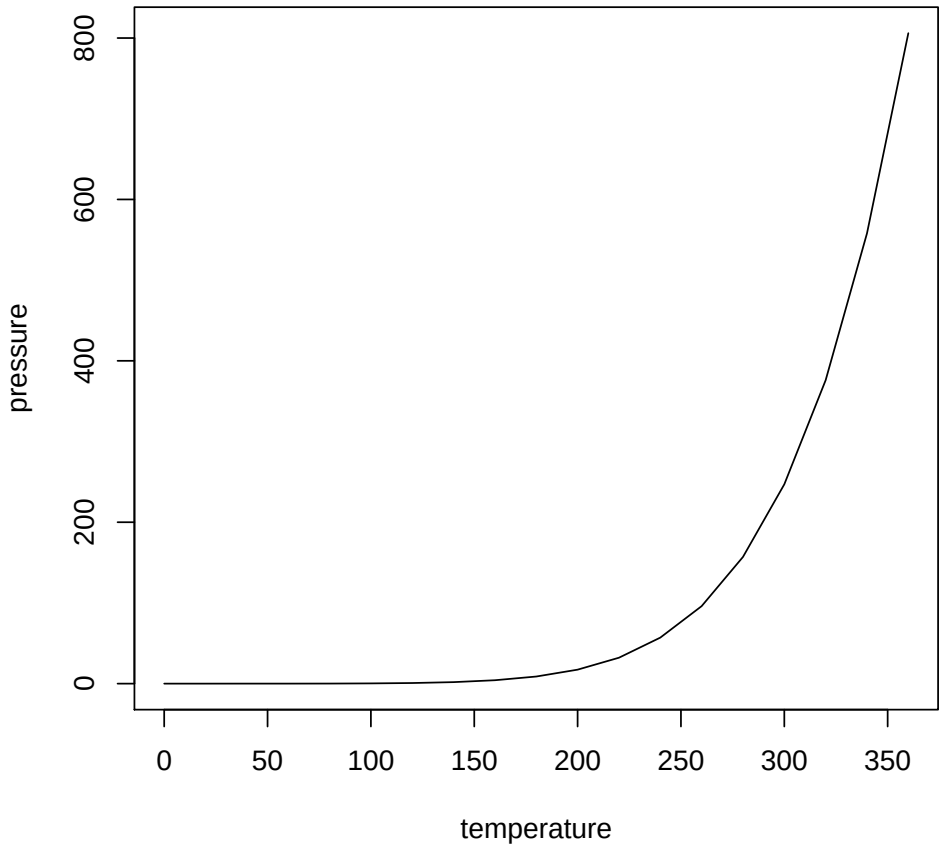
- title: sets titls
- type: "p" for points, "l" for lines, "b" for both

Try these parameters for `women` data! Hint: If data is not ordered use `orderBy` from `doBy` package

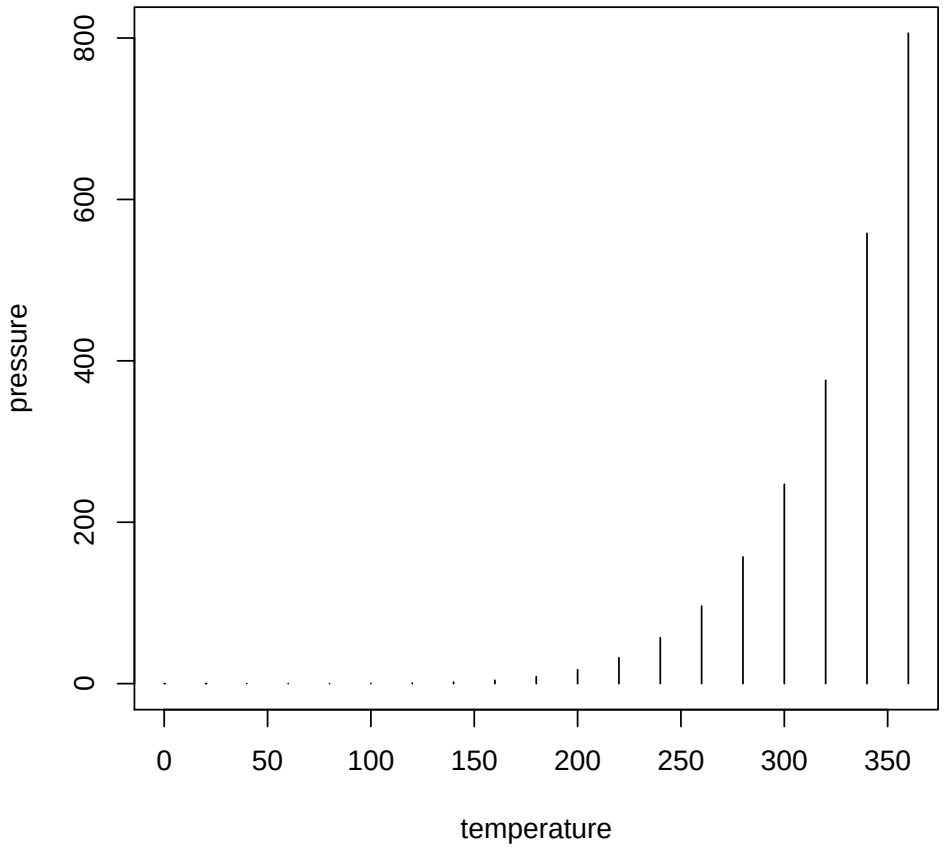
Plot example



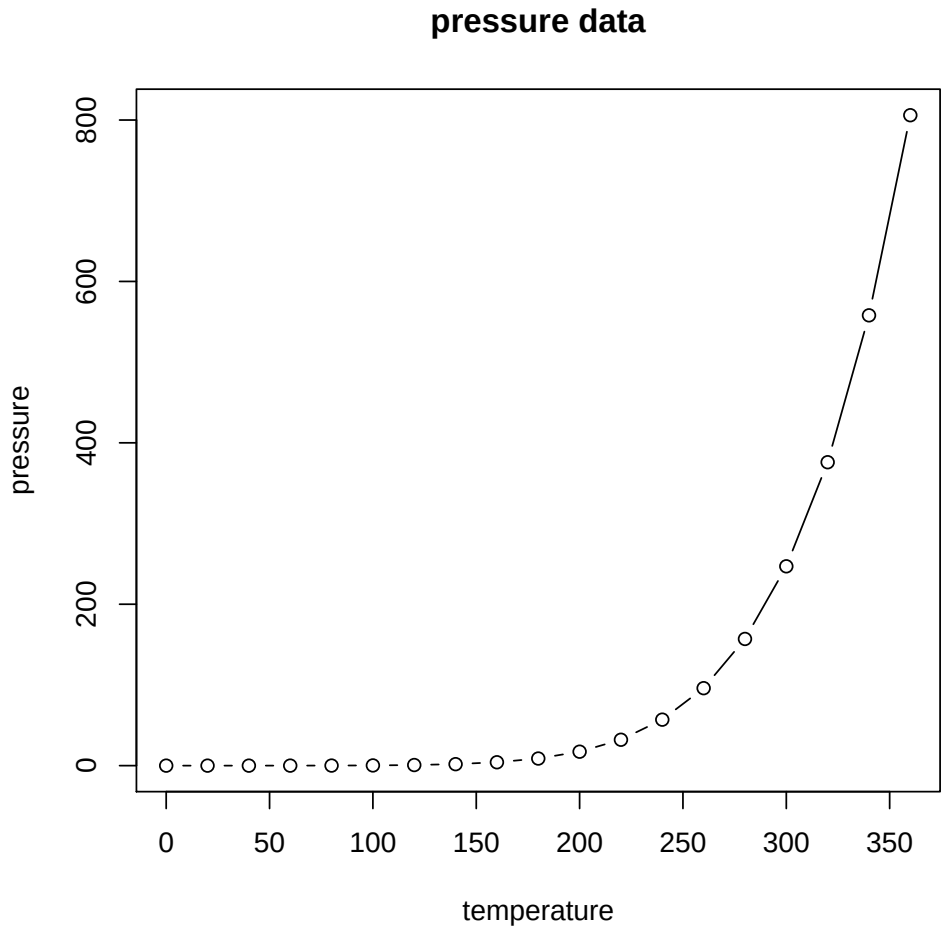
Plot example



Plot example



Plot example



Fitted Lines

- linear: fit lin. model and use `abline()`
- smooth curve: `lowess` or `locfit` (locfit library)
- horizontal/vertical lines: abline(h=100) or abline(v=100)

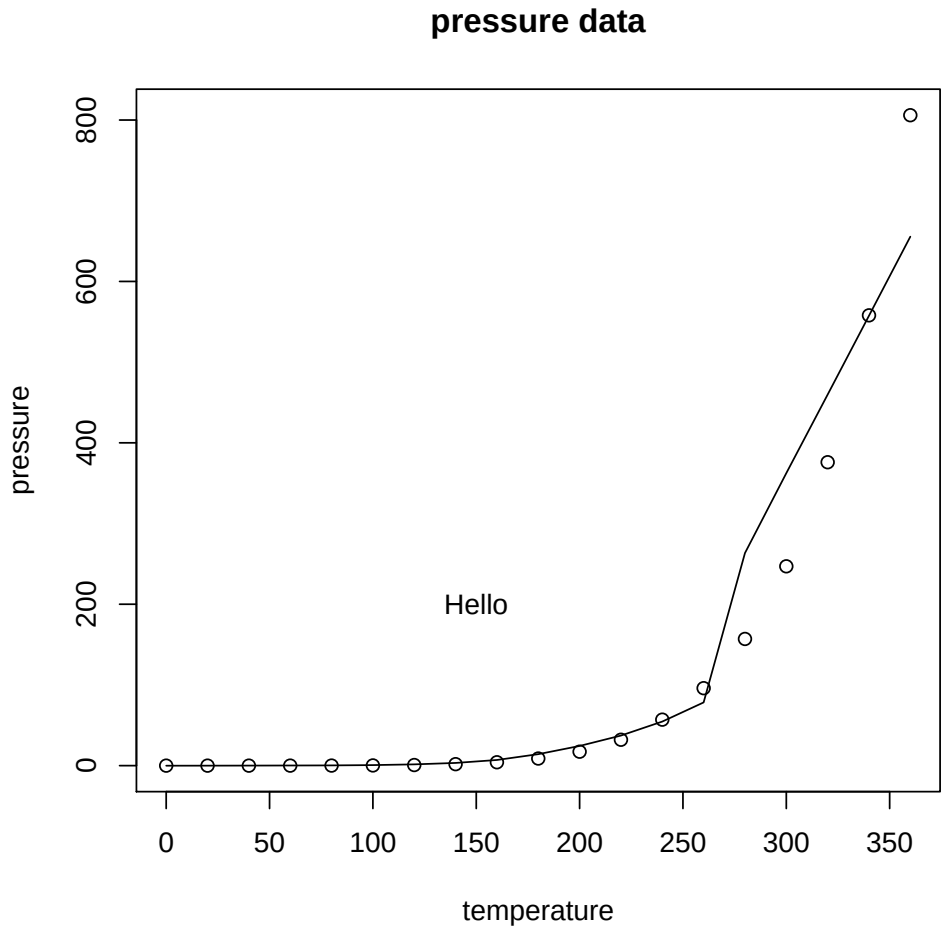
Try these parameters for `women` data!

Adding text

```
library("datasets")
data("pressure") # data(pressure) works, too
plot(pressure)
text(150,200,label="Hello")
```

What does the optional parameter `p`?

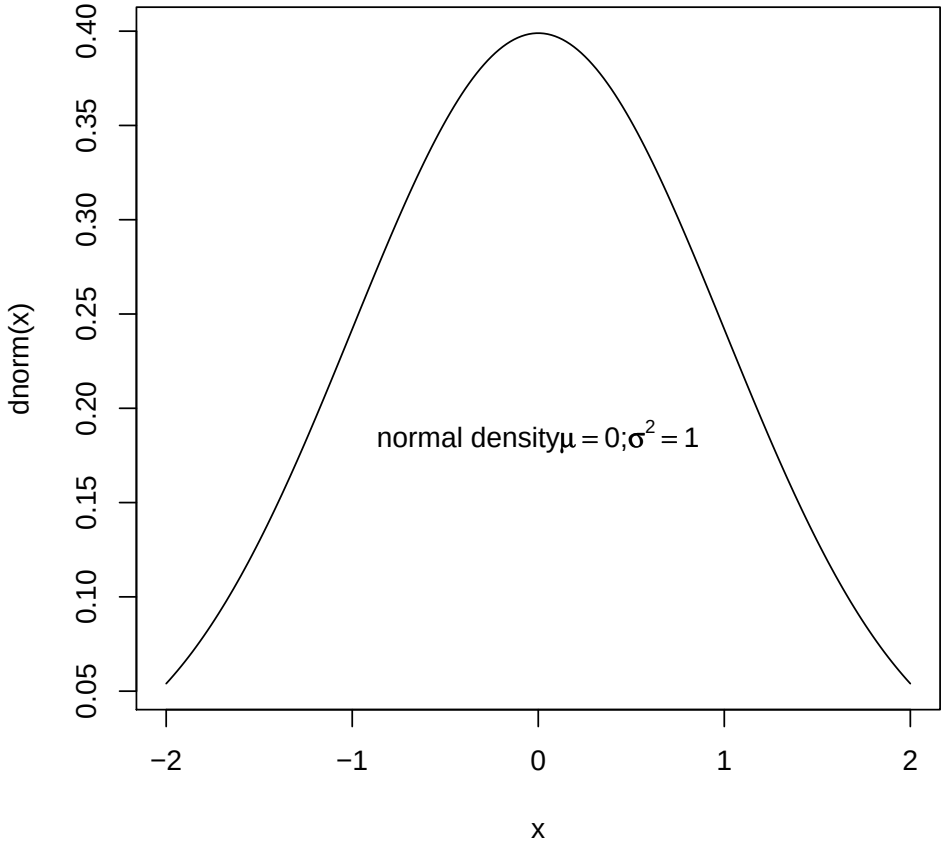
Plot example



Adding math

```
x <- seq(-2, 2, l = 200)
plot(x, dnorm(x), type = "l")
mu <- 0
variance <- 1
text(0, 0.2, pos = 1, label = bquote(paste("normal density",
mu == .(mu), ";", sigma^2 == .(variance))))
```

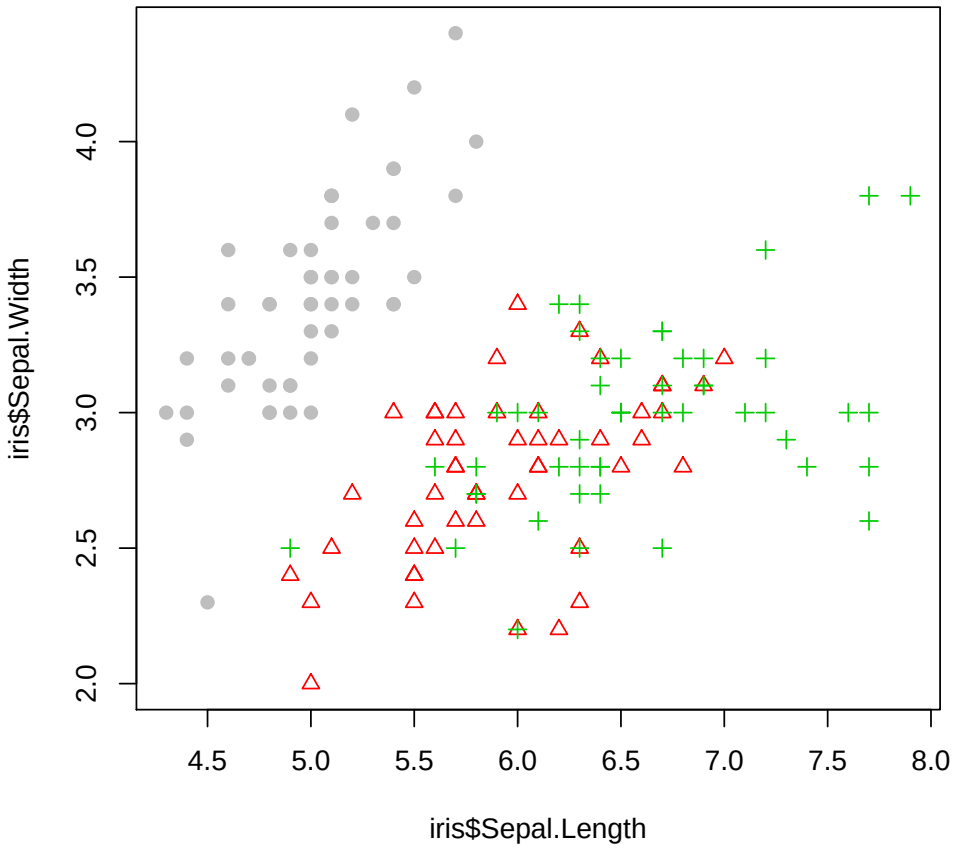
Plot example



Several plots in one display

```
data(iris) # load iris data
pch.vec <- c(16,2,3)[iris$Species]
col.vec <- c(16,2,3)[iris$Species]
plot(iris$Sepal.Length,iris$Sepal.Width,
col = col.vec,pch=pch.vec)
```

Plot example

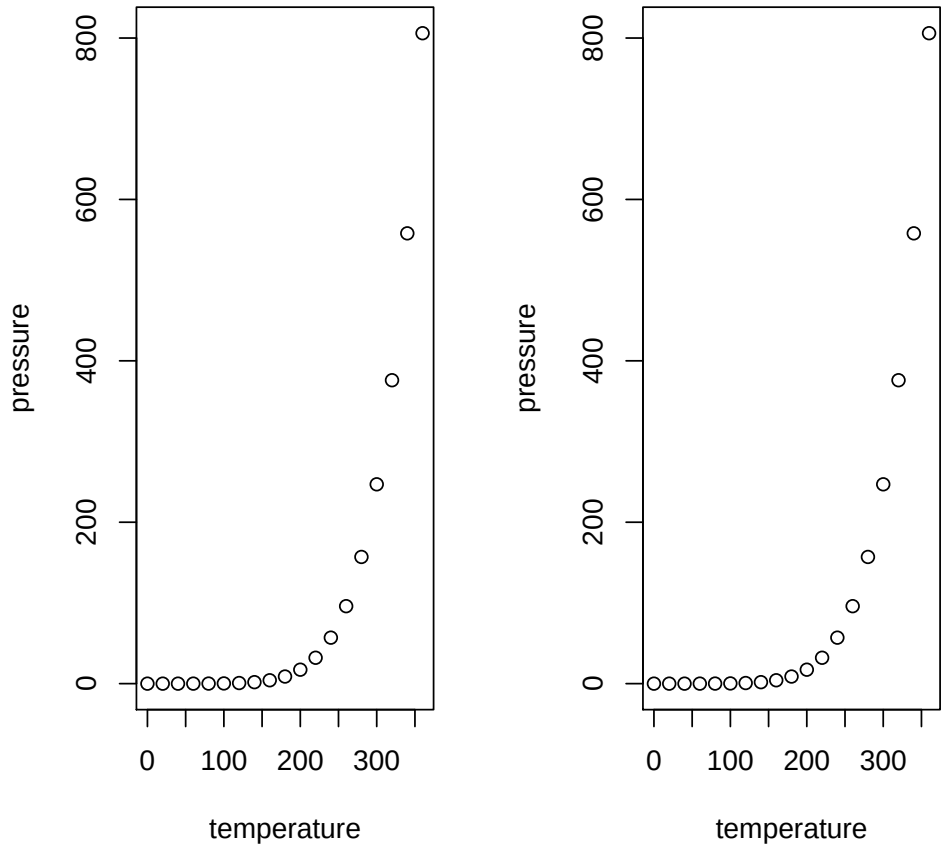


Side by Side Plots

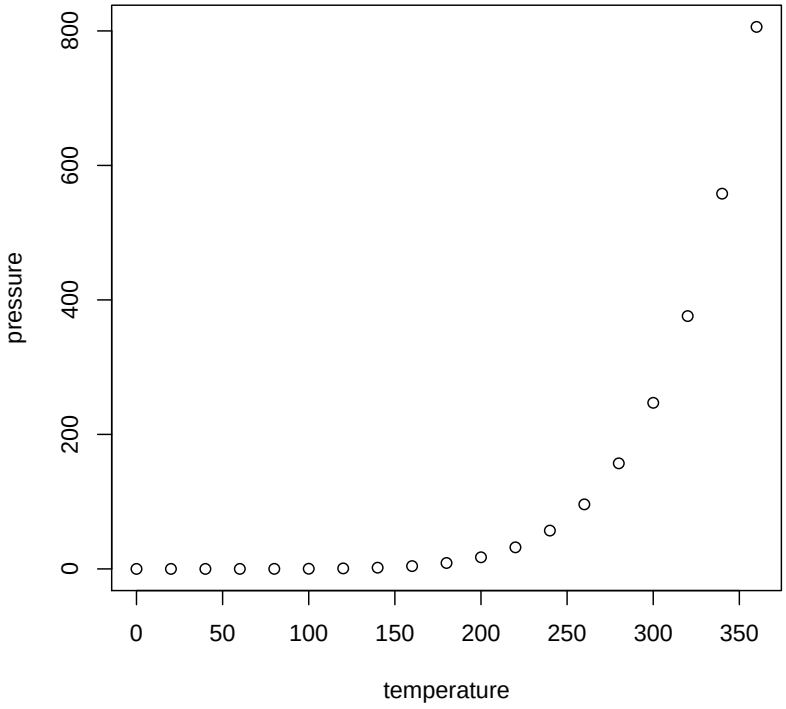
Use `par()` to set global settings.

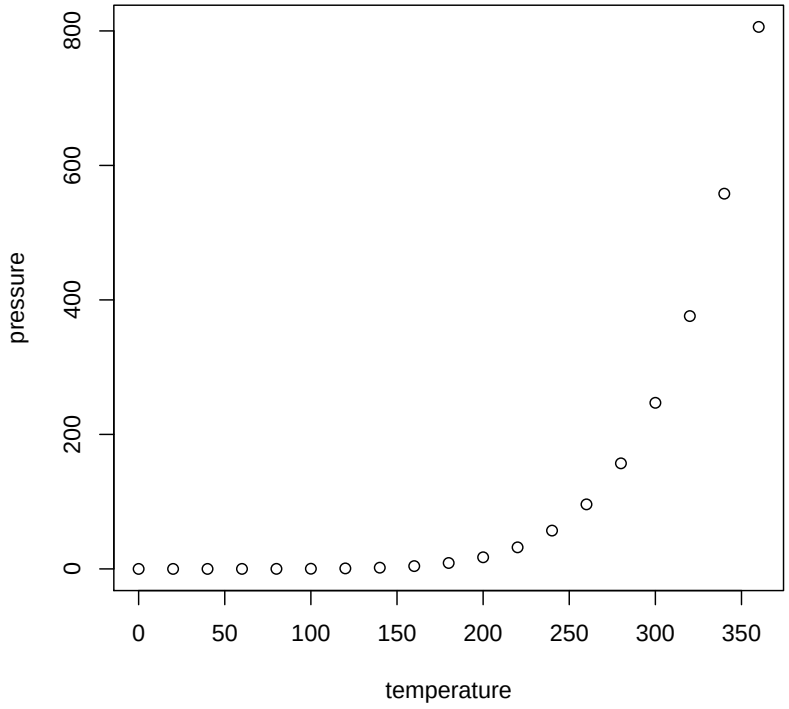
```
par(mfrow=c(1,2)) # two-colum plot
plot(pressure)
plot(pressure)
```

Side by Side Plots



Side by Side Plots





High-level plot options

Used by `par()` and plot() functions.

- adj: text justification
- ann: draw plot labels and titles
- bg: background color
- bty: type of box by box()
- cex: multiplier text size
- cex.axis: size axis tick labels

High-level plot options

- cex.lab: size axis labels
- cex.main: size plot title
- cex.sub: size plot subtitle
- col: color of lines and data symbols
- col.lab: color of labels
- col.main: color main title

High-level plot options

- col.sub: color subtitle
- fg: foreground color
- font: font face (bold, italic, etc)
- font.axis: font face axis tick label
- font.lab: font face axis label
- font.main: font face main title

High-level plot options

- font.sub: font face subtitle
- gamma: gamma correction
- lab: number of ticks on axis
- las: rotation of text in margins
- lty: line type
- lwd: line width

High-level plot options

- mgp: placement axis ticks and tick labels
- pch: data symbol
- srt: rotation of text in plot region
- tck: rel. lenght of axis ticks (plotsize)
- tcl: rel. length of axis ticks (textsize)
- tmag: rel. size plot title (other labels)

High-level plot options

- type: type of plot
- xaxp: number of ticks on x-axis
- xaxs: calculation of scale range on x-axis
- xaxt: x-axis style
- xpd: clipping region
- yaxp: number of ticks on y-axis
- yaxs: calculation of scale range on y-axis
- yaxt: y-axis style

Low-level plot options

Only used by `par()`.

- ask: prompt user befor next page
- family: font family
- fig: location of figure region
- fin: size of figure region in inch
- lend: line end style
- lheight: line spacing multiplier

Low-level plot options

- ljoin: line join style
- lmitre: line mitre limit
- mai: size of figure margins (inch)
- mar: size of figure margin (lines of text)
- mex: line spacing in margins
- mfcol: number of figures on a page

Low-level plot options

- mfg: which figure is used next
- mfrow: number of figures on a page
- new: has a new plot started?
- oma: size of outer margins
- omd: location of inner region
- omi: size of outer margins (inch)

Low-level plot options

- pin: size of plot region (inch)
- plt: location of plot region
- ps: size of text (points)
- pty: aspect ratio of plot region
- usr: range of scales on axes
- xlog: logarithmic scale on x-axis?
- ylog: logarithmic scale on y-axis?

Boxplot

Visualization of Tukey's five numbers

```
boxplot(iris$Sepal.Length) # boxplot for iris
  variable
```

Boxplot separated by factors

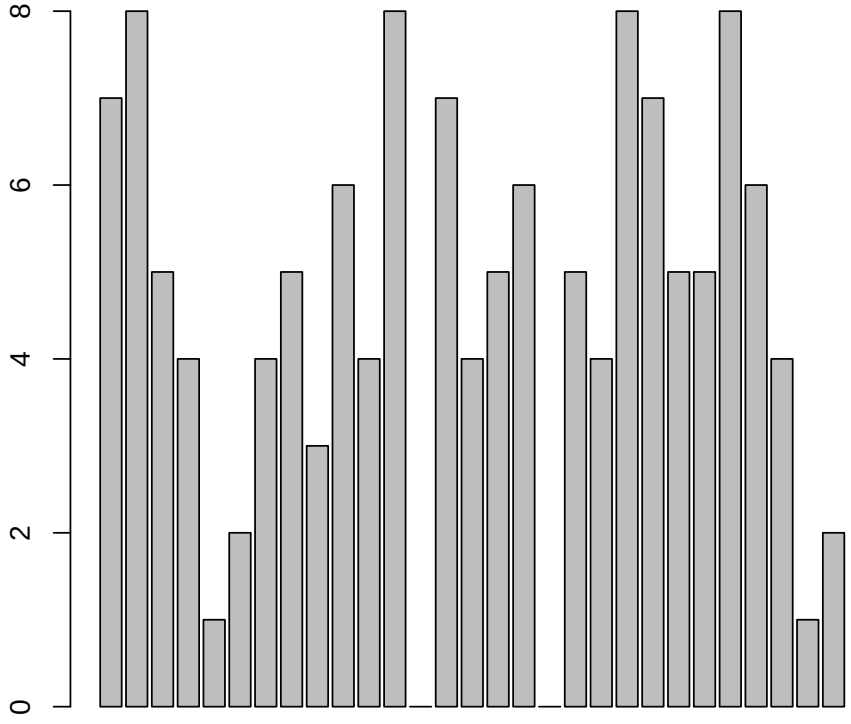
```
boxplot(Sepal.Length~Species, data = iris,
  horizontal=TRUE,col = c("red","blue","green"))
legend(y = 1.5, x = 6.5, legend = c("setosa", "
  versicolor","virginica"),pch=c(2,2,2),col = c("red
  ","blue","green"))
```

Barcharts

frequency distribution for discrete variables

```
data <- scan() # enter some data
barplot(data)
```

Histograms



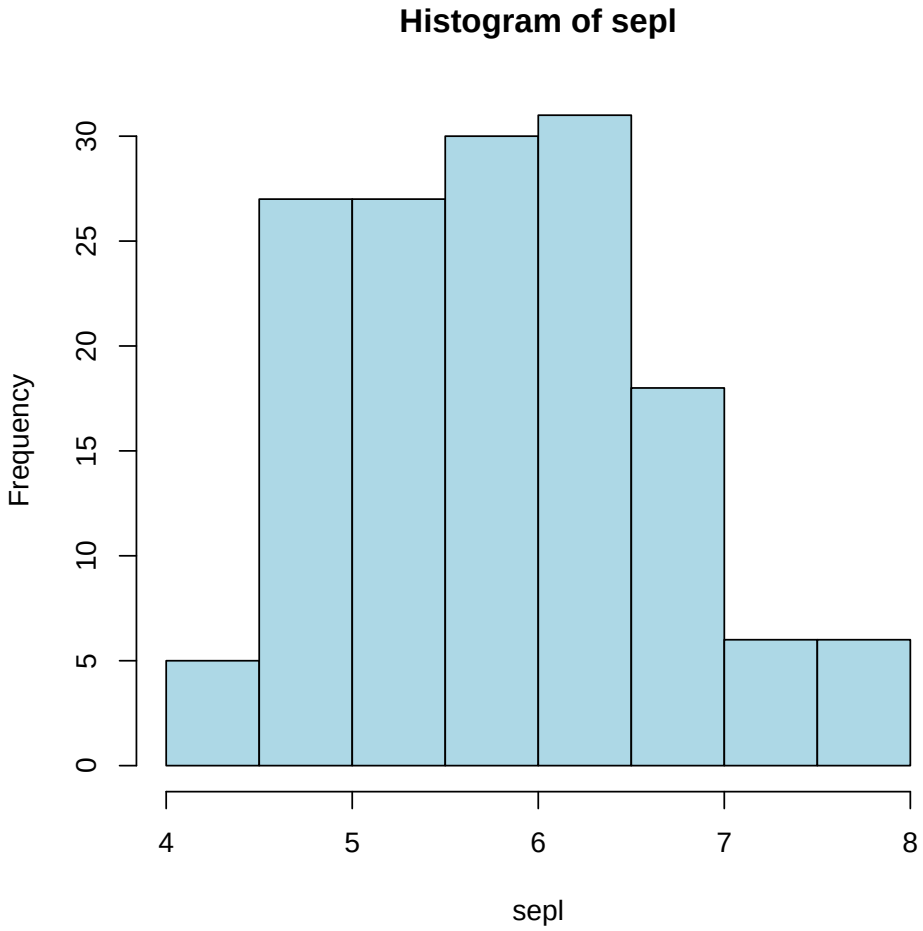
Histograms

frequency distribution for continous variables

```
hist(iris$Sepal.Length) # Histogram with frequencies
hist(iris$Sepal.Length,freq=FALSE) # with density
```

How is number of bin determined?

Histograms

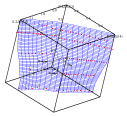


Histogram with dnorm()

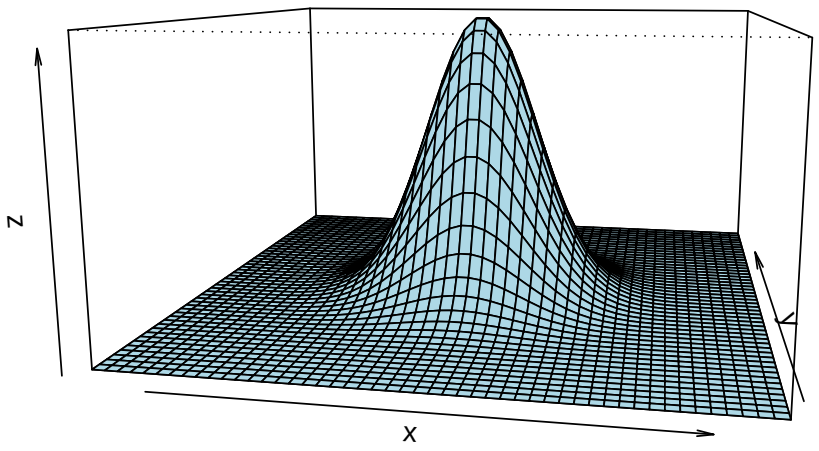
```
hist(iris$Sepal.Length,freq=FALSE) # with density
seq(min(iris$Sepal.Length),max(iris$Sepal.Length),
  length=100)->grid
d1<-dnorm(grid,mean(iris$Sepal.Length),sd(iris$Sepal
  .Length))
lines(grid,d1,col="red")
```

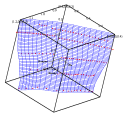
Multivariate Plots

```
## Bivariate normal distrib. density
library(mvtnorm)
x <- y <- seq(-5, 5, length = 50)
f <- function(x,y) { dmvnorm(cbind(x,y)) }
z <- outer(x, y, f)
persp(x, y, z, theta = 10, phi = 20, expand = 0.5,
  col = "lightblue", shade = 0.75)
persp(x, y, z, theta = 10, phi = 20, expand = 0.5,
  col = "lightblue")
```

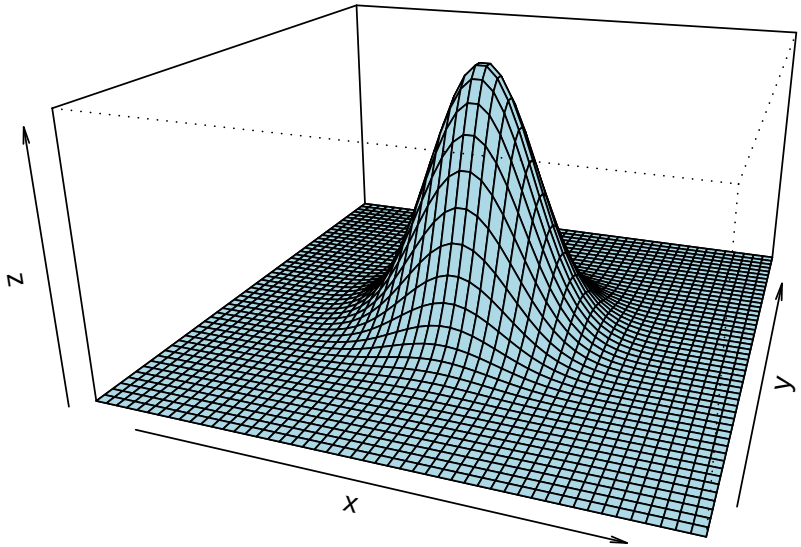



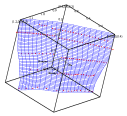
Multivariate Plots



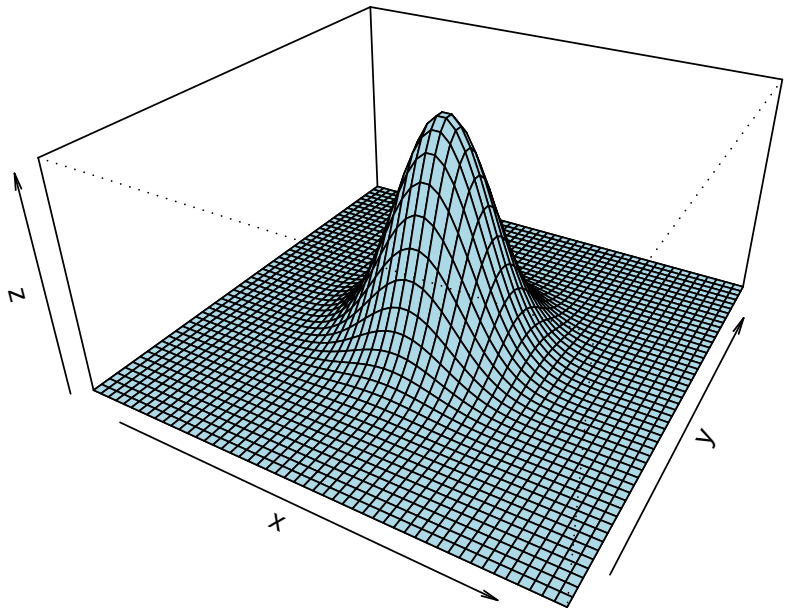


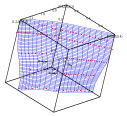
Multivariate Plots



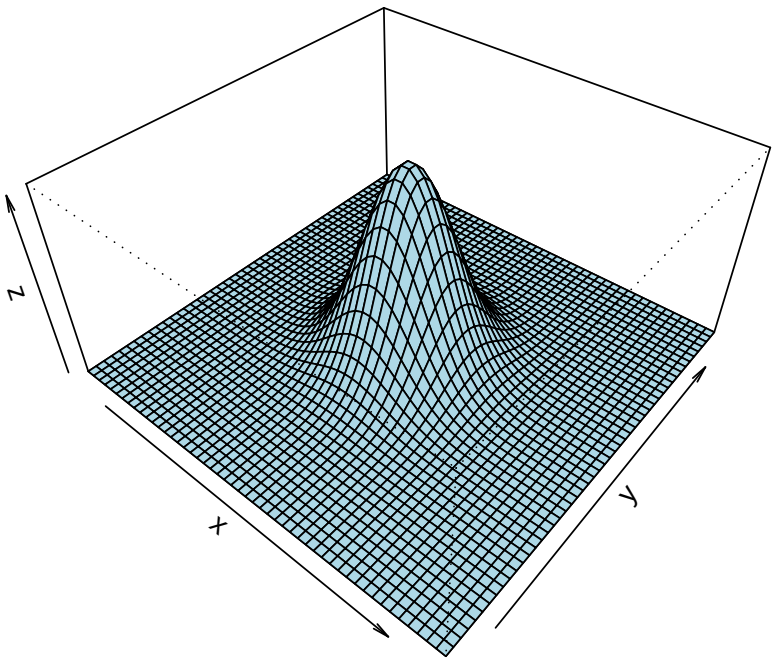


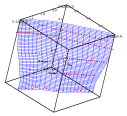
Multivariate Plots



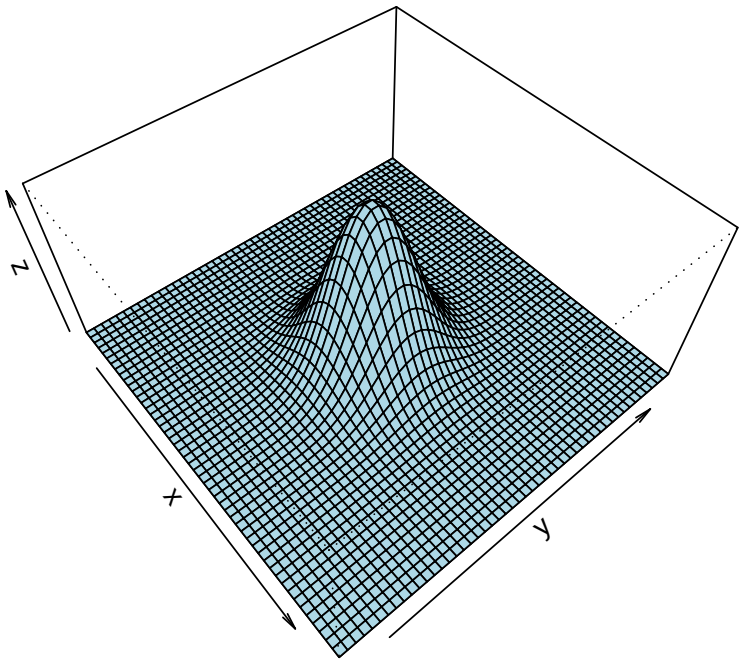


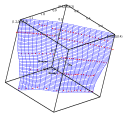
Multivariate Plots



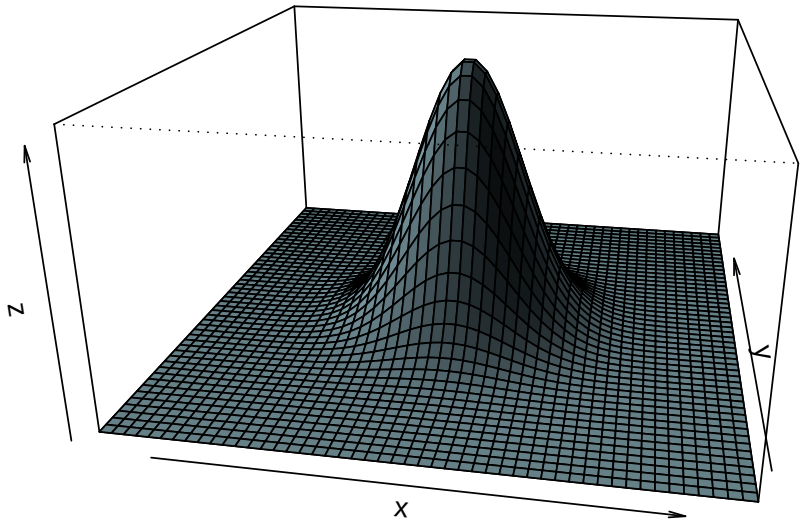


Multivariate Plots





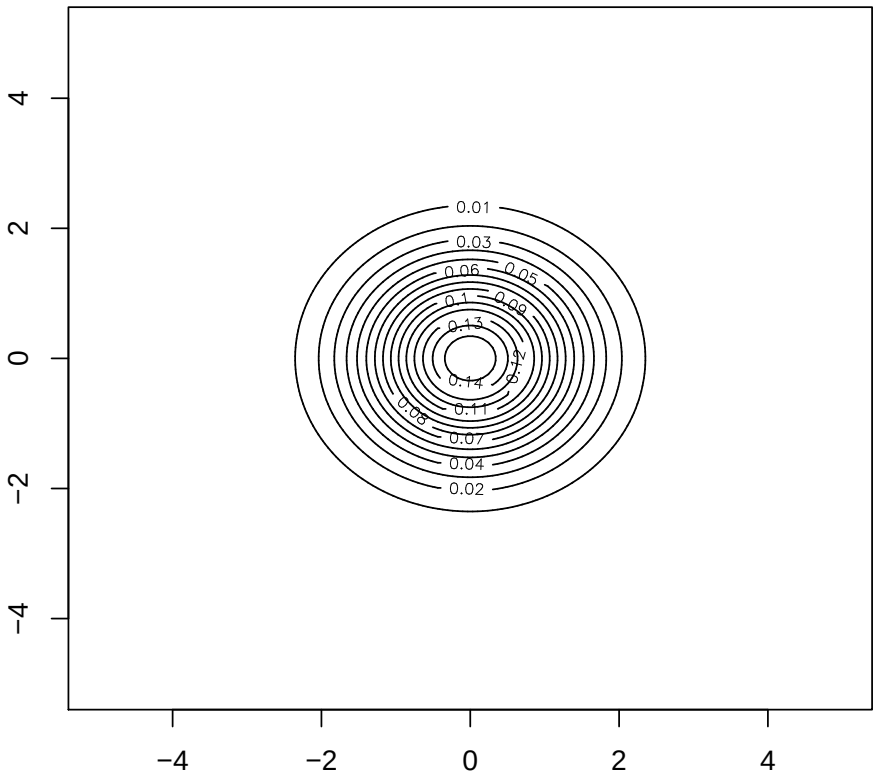
Multivariate Plots



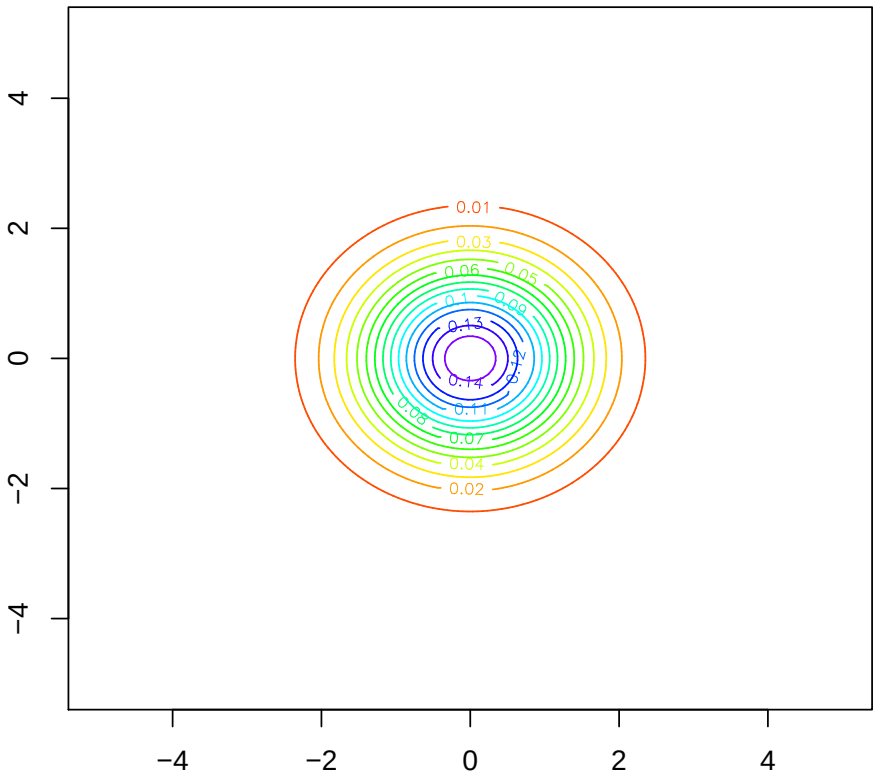
Multivariate Plots

```
## contour curves for bivariate density
x <- y <- seq(-5, 5, length = 150)
z <- outer(x, y, f)
contour(x, y, z, nlevels=20)
contour(x, y, z, nlevels=20, col=rainbow(20))
contour(x, y, z, nlevels=20, col=rainbow(20), labels
  ="")
```

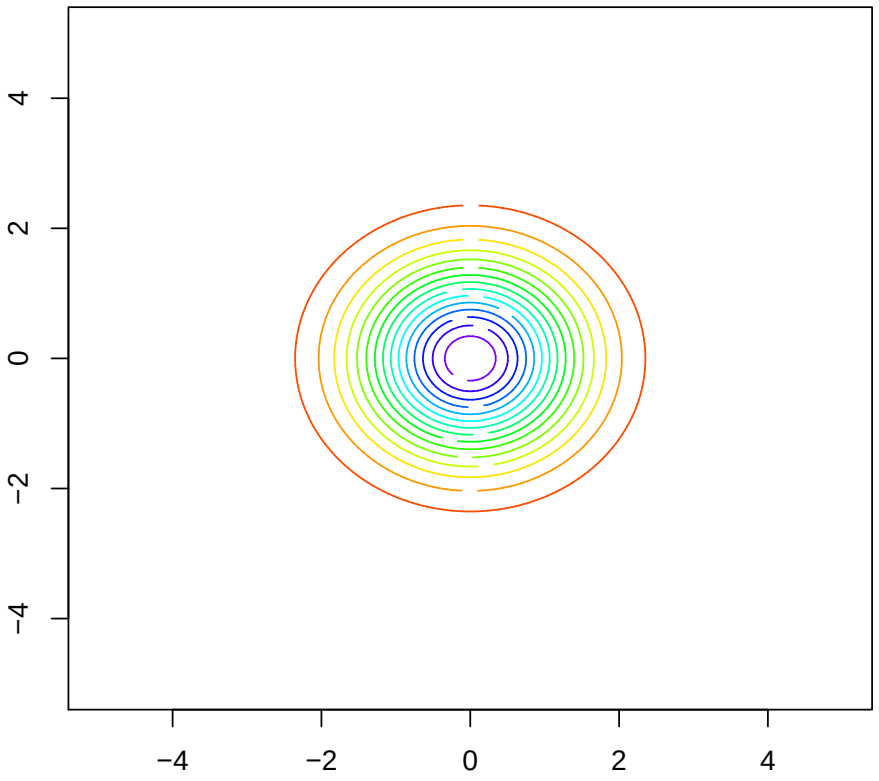
Contour Example



Contour Example

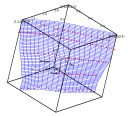


Contour Example

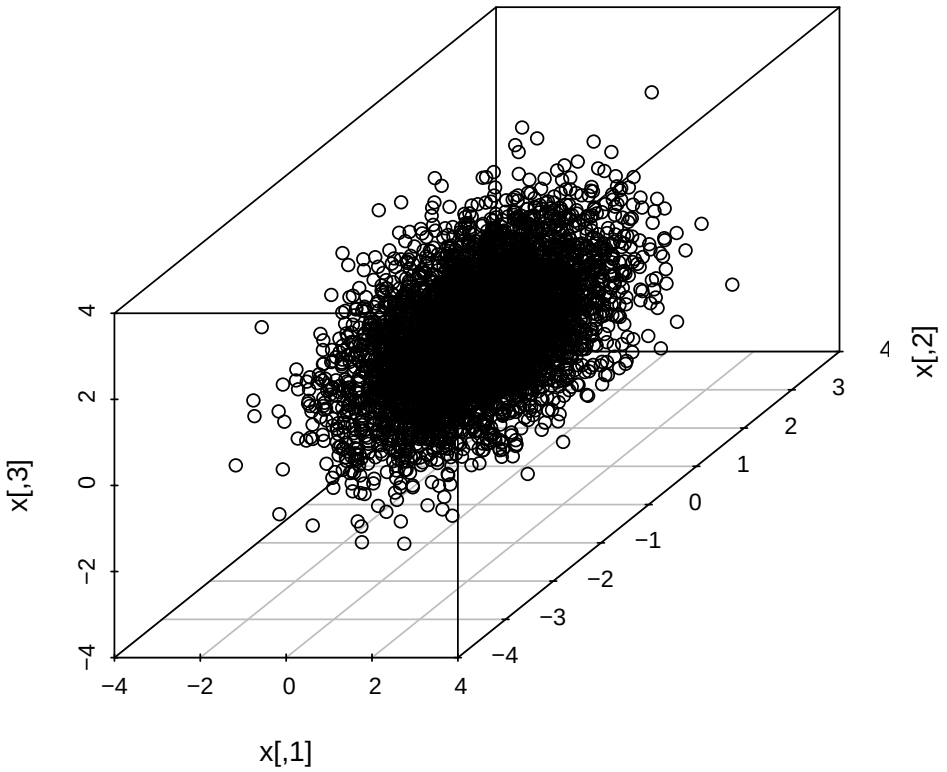


3D Plots

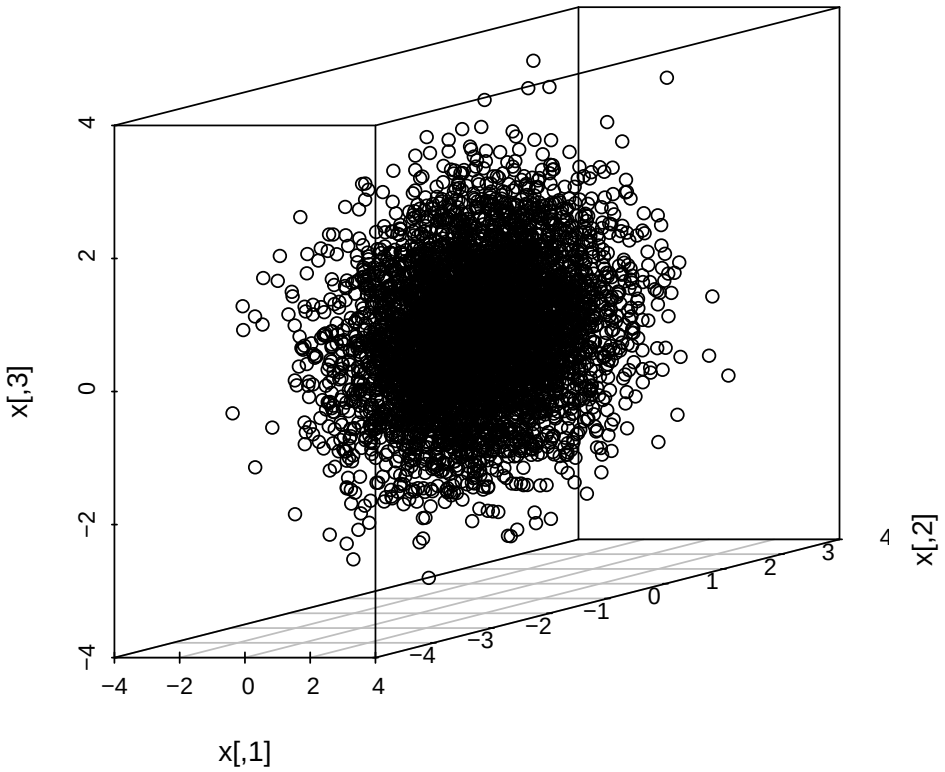
```
## 3D normally distrib. data
library(scatterplot3d) # extra package!
x <- matrix(rnorm(15000),ncol=3)
scatterplot3d(x)
scatterplot3d(x, angle=20)
```

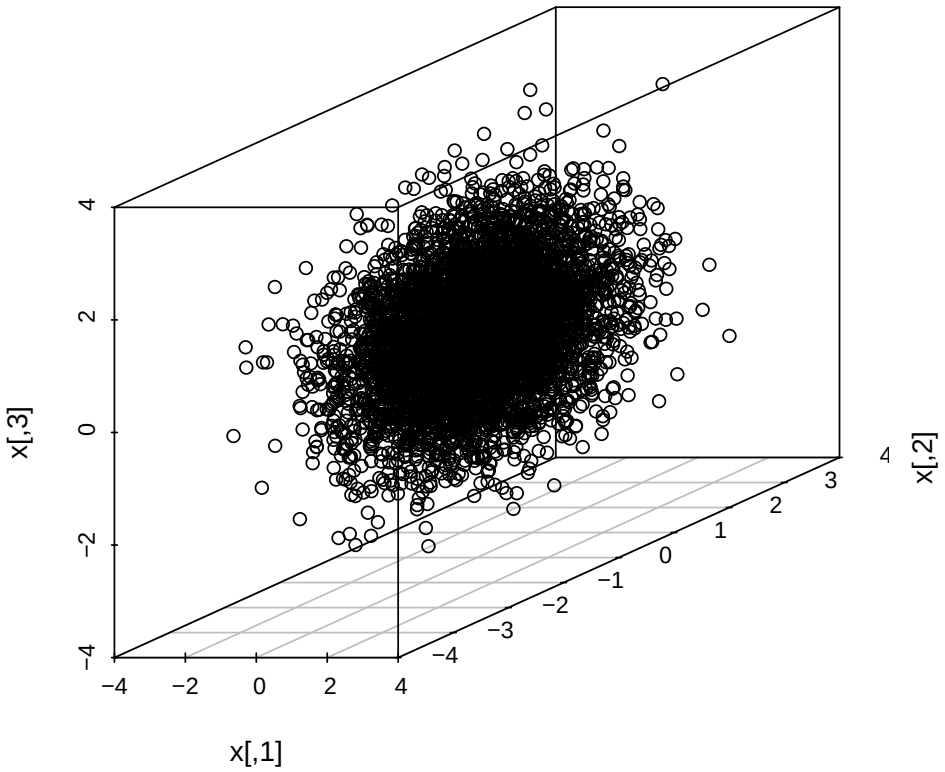
3D Example



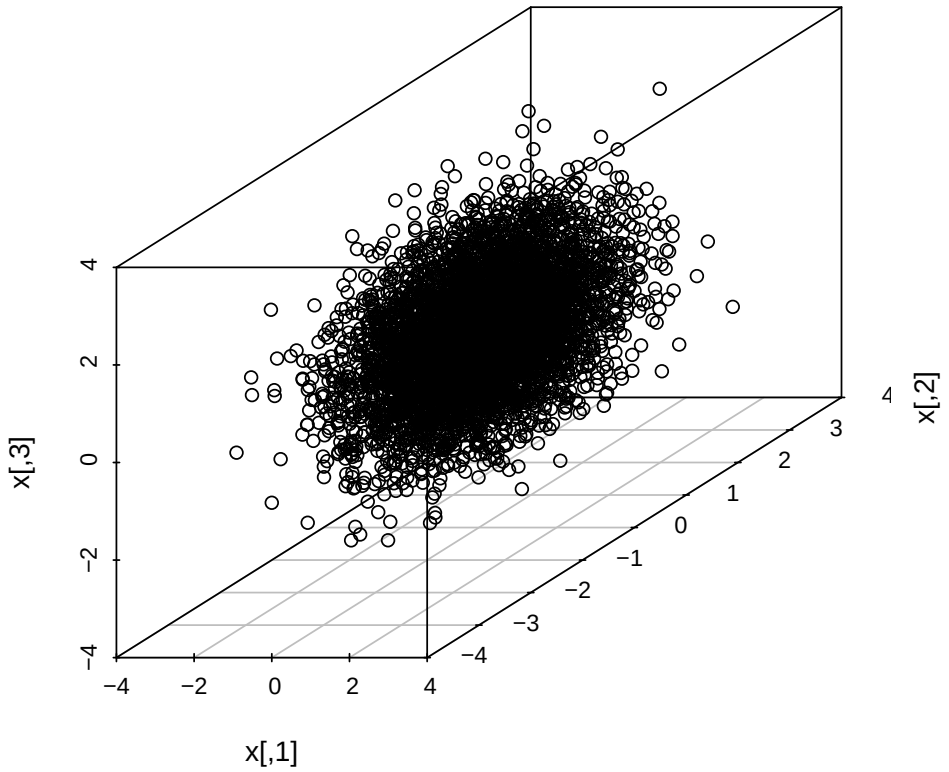
3D Example



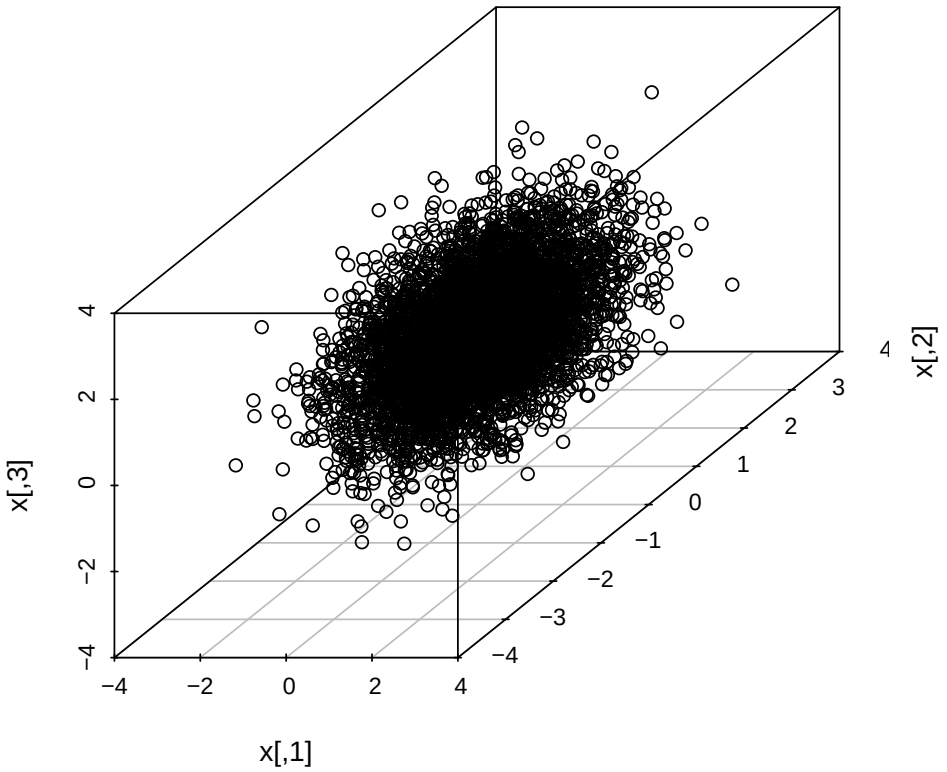
3D Example



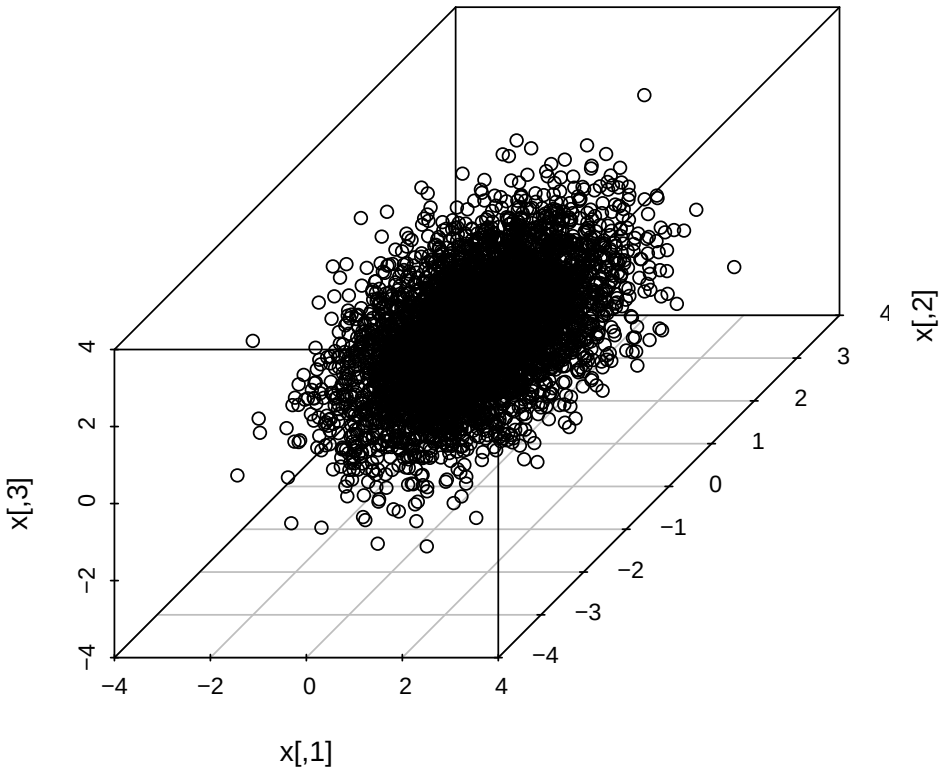
3D Example



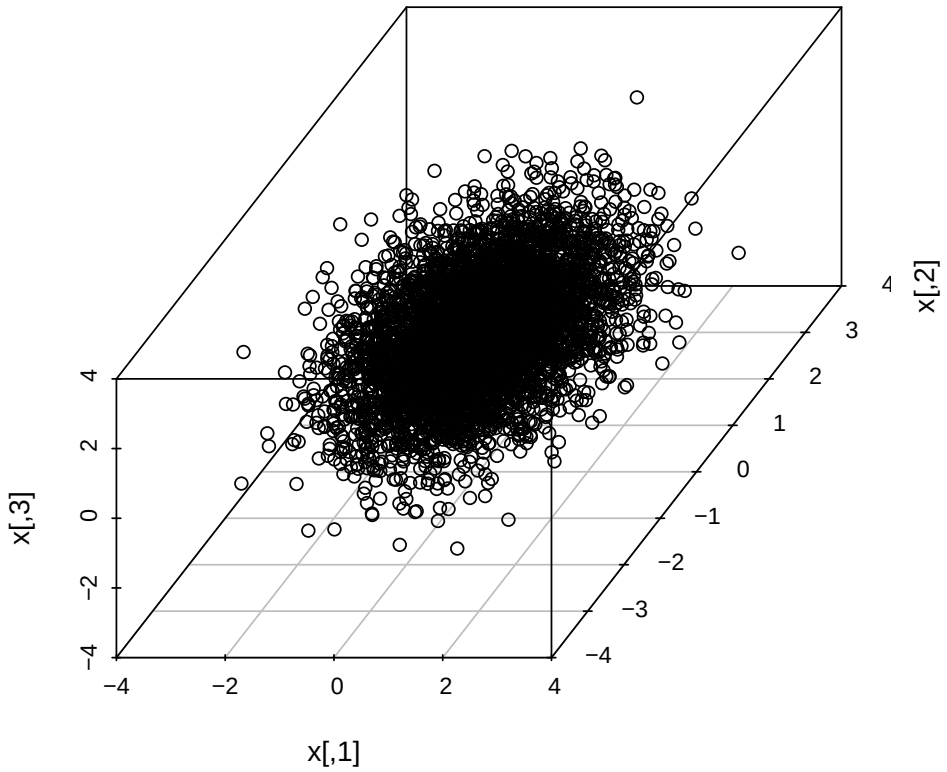
3D Example



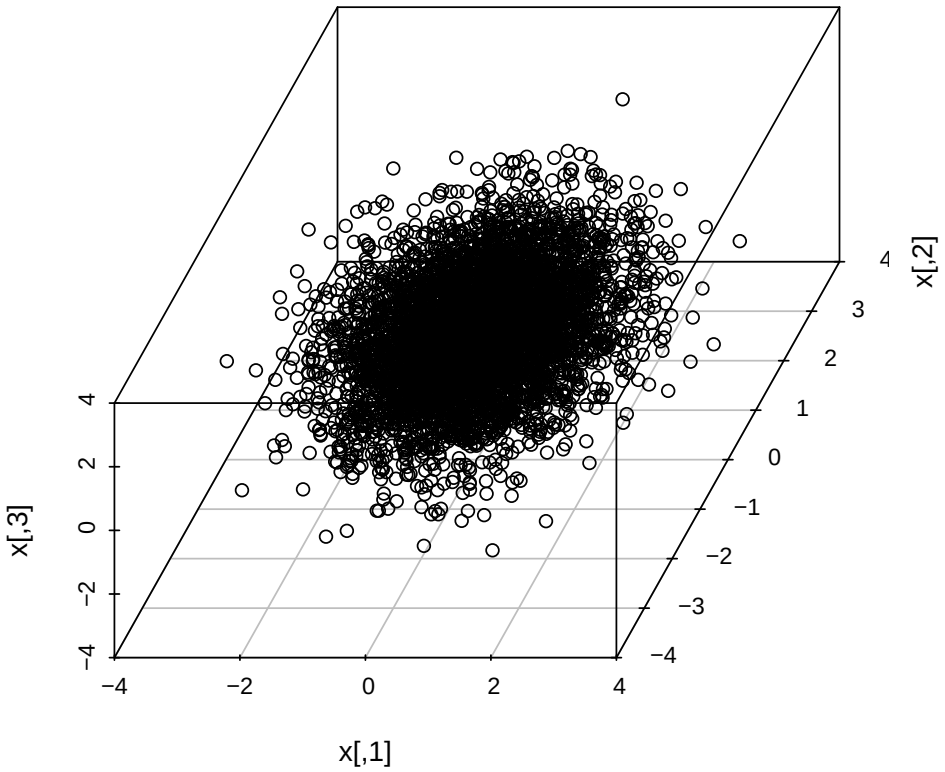
3D Example



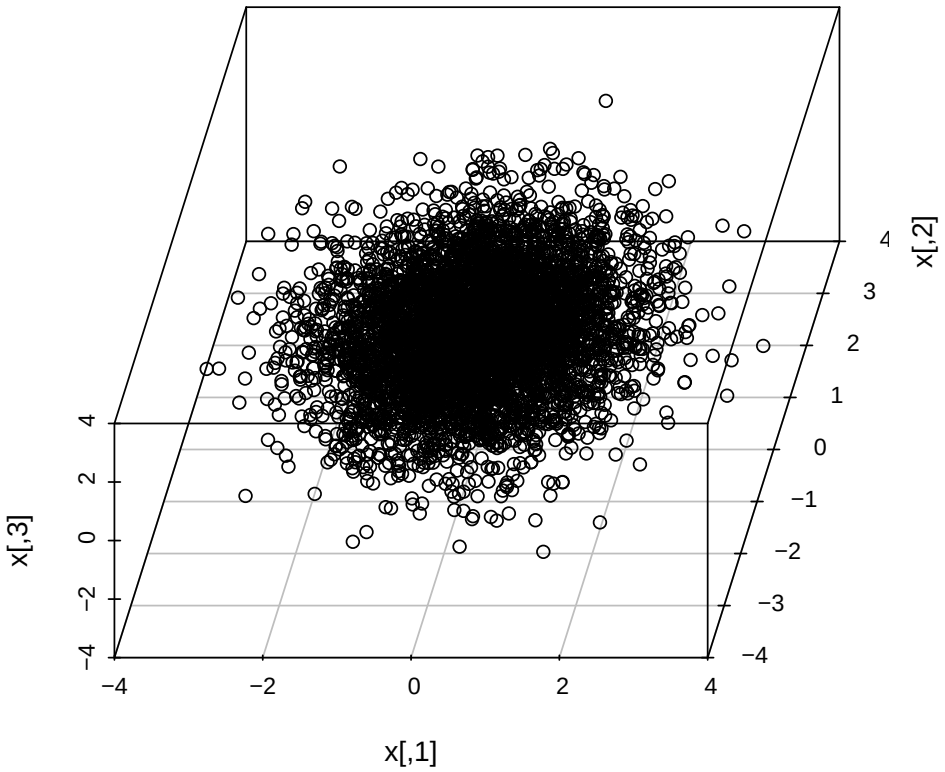
3D Example



3D Example



3D Example



Other Multivariate Plots

- parallel coordinate plots `parcoord()`
- star plots `stars()`
- sunflower plot `sunflowerplot()`
- dot plots `dotchart()`
- Flury faces `faces()/faces2()` from TeachingDemos package

The Lattice Package

- based on the `grid` system
- complete graphics system
- more object-oriented compared with traditional graphics
- includes re-implementations of trad. graphics
- allows more control on how output is arranged

The Lattice Package

Lattice	Description	Traditional
`barchart()`	Barchart	`barplot()`
`bwplot()`	Box-plots	`boxplot()`
`densityplot()`	Condition kde plot	*none*
`dotplot()`	Dotplots	`dotchart()`
`histogram()`	Histograms	`hist()`
`qqmath()`	Quantile-Quantile plots	`qqnorm()`
`stripplot()`	One-dim. scatterplots	`stripchart()`

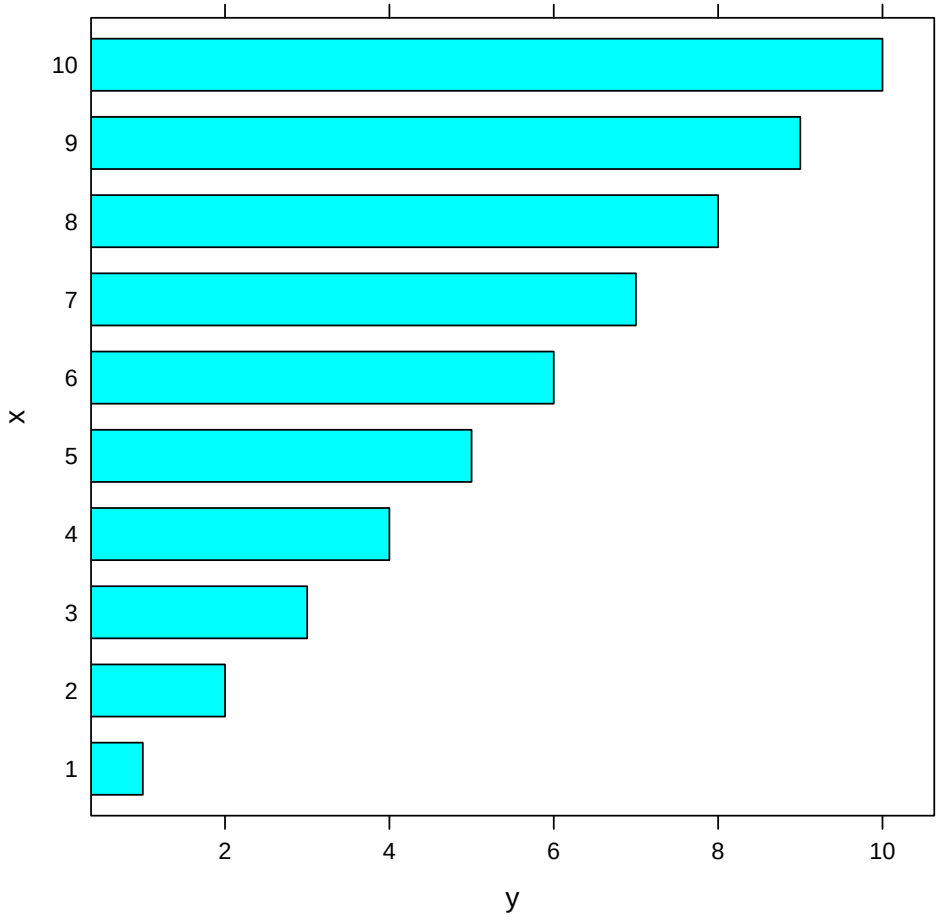
The Lattice Package

Lattice	Description	Traditional
`qq()`	Quantile-Quantile plots	`qqplot()`
`xyplot()`	Scatterplot	`plot()`
`levelplot()`	Level plots	`image()`
`contourplot()`	Contour plots	`contour()`
`cloud()`	3-dim. scatterplot	*none*
`wireframe()`	3-dim. surface	`persp()`
`splom()`	Scatterplot matrices	`pairs()`
`parallel()`	Parallel coordinate plots	*none*

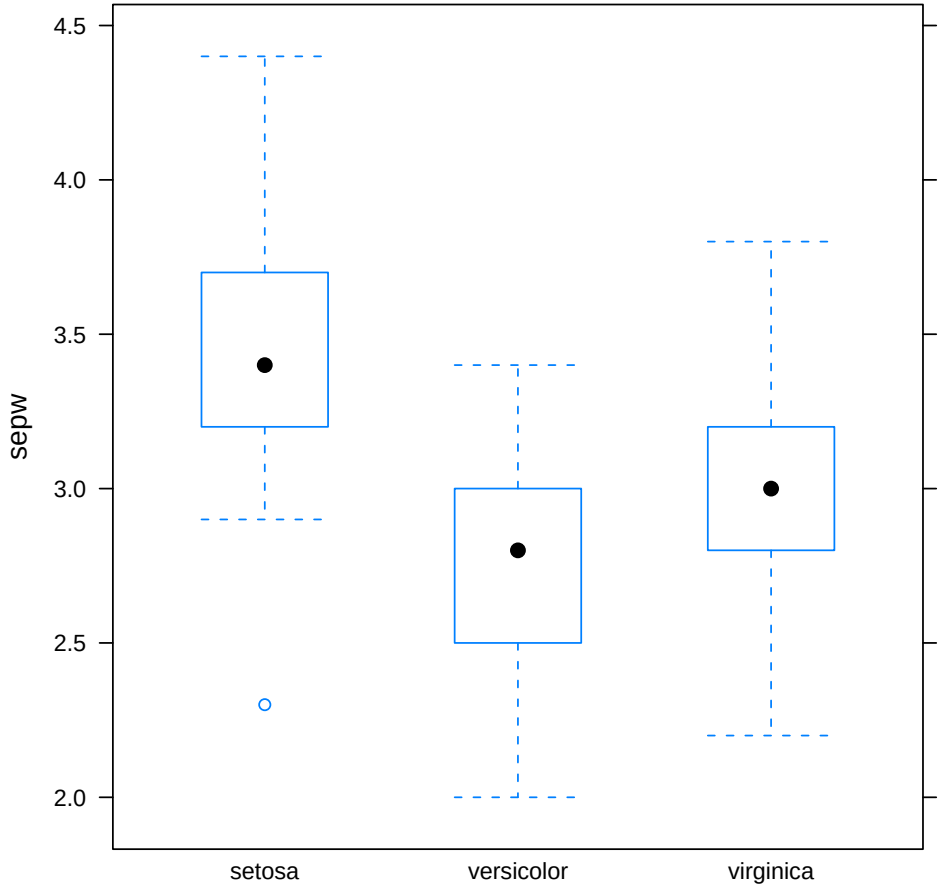
Lattice

```
library("lattice");
x = 1:10; y = 1:10;
p <- xyplot(x~y);
print(p);
update(p,main = "title");
```

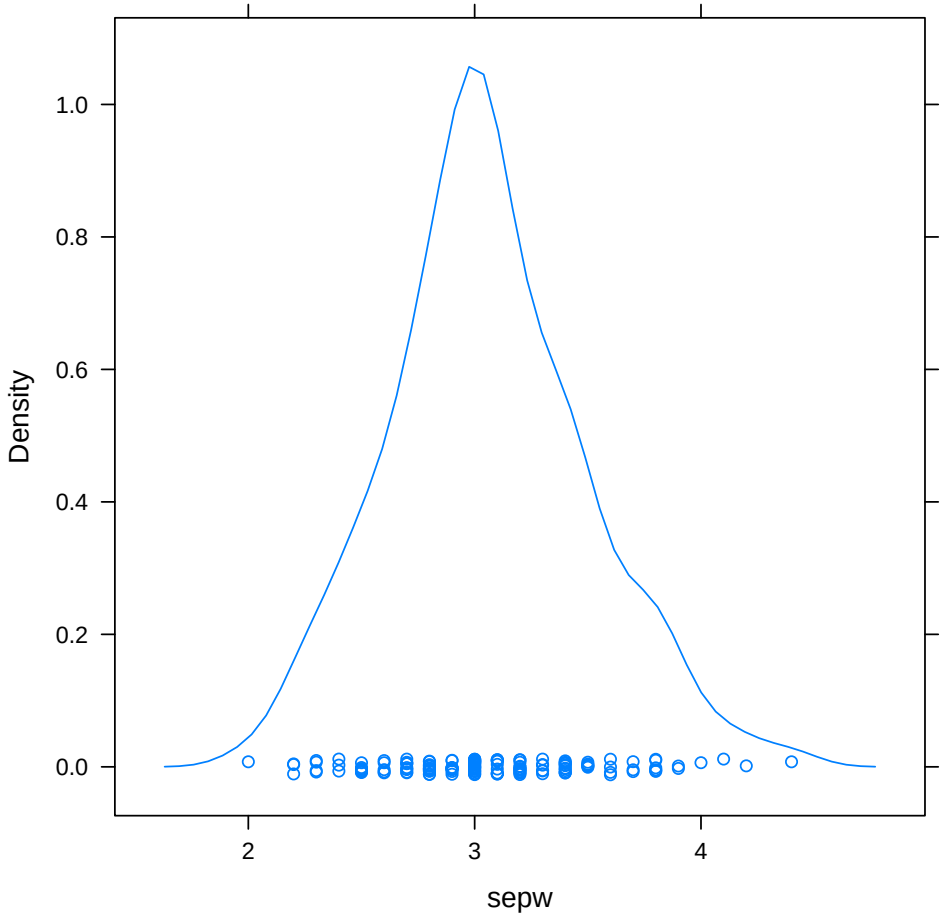
Lattice Example



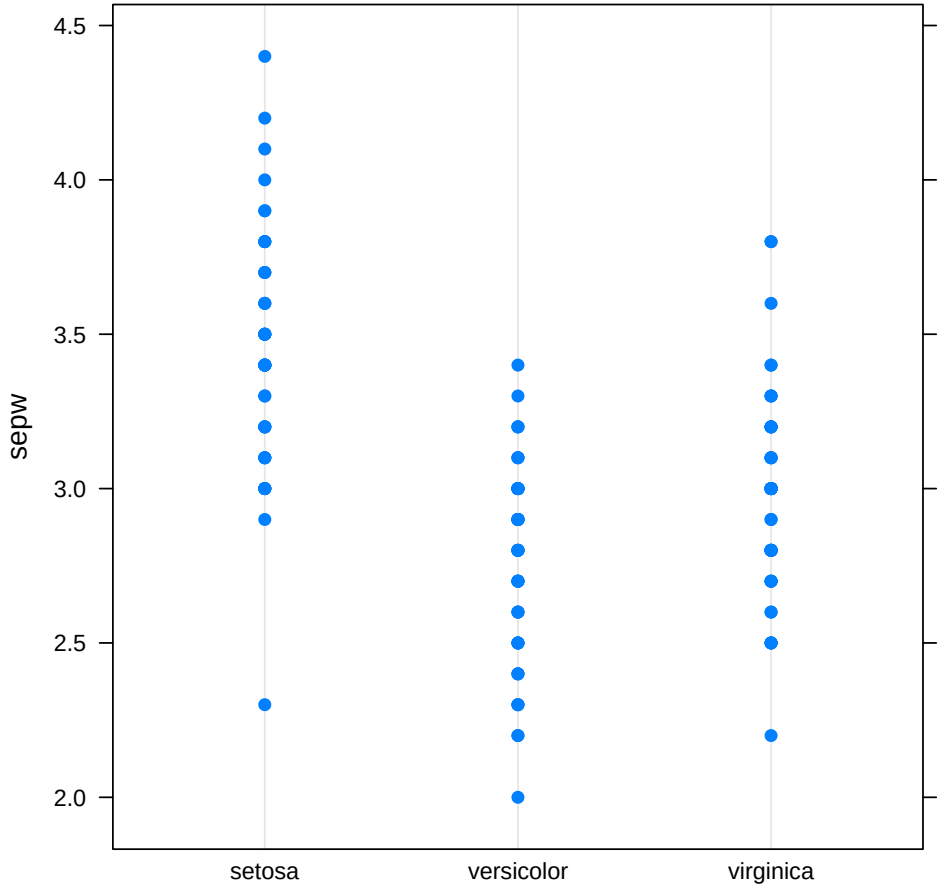
Lattice Example



Lattice Example



Lattice Example



Lattice Example