

Intro stats with mosaic

lattice version

Loading packages

```
library(mosaic)
```

Essential R syntax

Names in R are *case sensitive*

Function and arguments

```
rflip(10)
```

Optional arguments

```
rflip(10, prob = 0.8)
```

Assignment

```
x <- rflip(10, prob = 0.8)
```

Getting help on any function

```
help(mean)
```

Arithmetic operations

+	-	*	/	basic operations
^				exponentiation
()				grouping
sqrt(x)				square root
abs(x)				absolute value
log10(x)				logarithm, base 10
log(x)				natural logarithm, base e
exp(x)				exponential function e^x
factorial(k)				$k! = k(k-1) \dots 1$

Logical operators

==	is equal to (note double equal sign)
!=	is not equal to
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to
&	A & B is TRUE if both A and B are TRUE
	A B is TRUE if one or both of A and B are TRUE
%in%	includes; for example
	"C" %in% c("A", "B") is FALSE

Formula interface

Use for graphics, statistics, inference, and modeling operations.

```
goal(y ~ x, data = mydata)
```

Read as "Calculate **goal** for **y** using **mydata** "broken down by" **x**, or "modeled by" **x**.

```
mean(age ~ sex, data = HELPrct)
```

For graphics:

```
goal(y ~ x | z, data = mydata, groups = w)
```

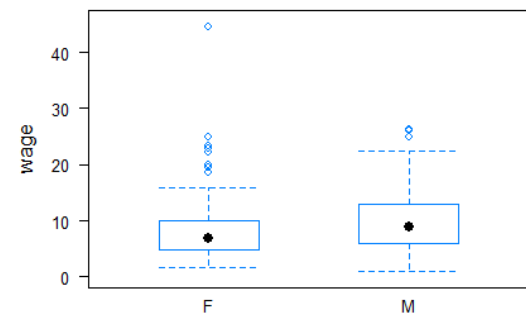
y : y-axis variable (*optional*)

x : x-axis variable (*required*)

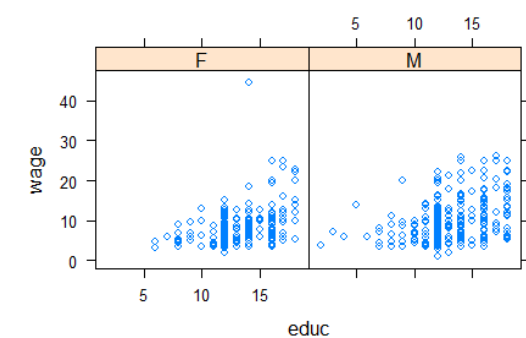
z : panel-by variable (*optional*)

w : color-by variable (*optional*)

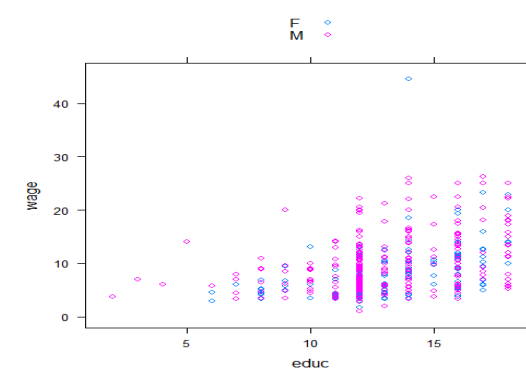
```
bwplot(wage ~ sex, data = CPS85)
```



```
xyplot(wage ~ educ | sex, data = CPS85)
```



```
xyplot(wage ~ educ, data = CPS85, groups = sex, auto.key = TRUE)
```



Examining data

Print short summary of all variables
`inspect(HELPrct)`

Number of rows and columns

```
dim(HELPrct)
```

```
nrow(HELPrct)
```

```
ncol(HELPrct)
```

Print first rows or last rows

```
head(KidsFeet)
```

```
tail(KidsFeet, 10)
```

Names of variables

```
names(HELPrct)
```

One categorical variable

Counts by category

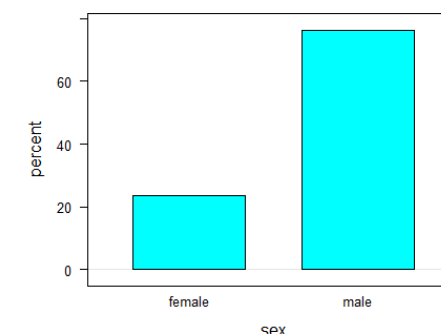
```
tally(~ sex, data = HELPrct)
```

Percentages by category

```
tally(~ sex, data = HELPrct, format = "percent")
```

Bar graph of percentages

```
bargraph(~ sex, data = HELPrct, type = "percent")
```



Tests and confidence intervals

Exact test

```
result1 <-
```

```
  binom.test(~ (homeless == "homeless"), data = HELPrct)
```

Approximate test (large samples)

```
result2 <-
```

```
  prop.test(~ (homeless == "homeless"), data = HELPrct, p = 0.4, alternative = "less")
```

Extract confidence intervals and p -values

```
confint(result1)
```

```
pval(result2)
```

One quantitative variable

Make output more readable

```
options(digits = 3)
```

Compute summary statistics

```
mean(~ cesd, data = HELPrct)
```

Other summary statistics work similarly

```
median() iqr() max() min()
```

```
fivenum() sd() var() sum()
```

Table of summary statistics

```
favstats(~ cesd, data = HELPrct)
```

Summary statistics by group

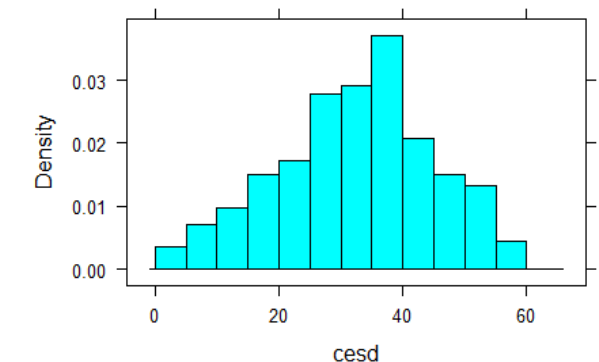
```
favstats(cesd ~ sex, data = HELPrct)
```

Quantiles

```
quantile(~ cesd, data = HELPrct, prob = c(0.25, 0.5, 0.8))
```

Histogram

```
histogram(~ cesd, data = HELPrct, width = 5, center = 2.5)
```



Normal probability plot

```
qqmath(~ cesd, data = HELPrct, dist = "qnorm")
```

Density plot

```
densityplot(~ cesd, data = HELPrct)
```

Dot plot

```
dotPlot(~ cesd, data = HELPrct)
```

One-sample t -test

```
result <- t.test(~ cesd, data = HELPrct, mu = 34)
```

Extract confidence intervals and p -values

```
confint(result)
```

```
pval(result)
```

Paired t -test

```
t_test(extra ~ group, data = sleep, paired = TRUE)
```

Data wrangling

```
Drop, rename, or reorder variables
df <- select(HELPrct,
  c(id, age, sex))

Create new variables from existing ones
KidsFeet <- mutate(KidsFeet,
  width_in = 0.394 * width)

Retain specific rows from data
girls_feet <- filter(KidsFeet,
  sex == "G")

Sort data rows by value in column
df <- arrange(KidsFeet, length)

Compute summary statistics by group
group_by(KidsFeet, sex) %>%
  summarize(mean_width =
    mean(width))

For more, see Tidyverse cheatsheet
```

Importing data

```
Import data from file or URL
MustangPrice <-
  read.file("C:/MustangPrice.csv")
# NOTE: R uses forward slashes!
Dome <-
  read.file("http://www.mosaic-
  web.org/go/datasets/Dome.csv")
```

Randomization and simulation

```
Fix random number sequence
set.seed(42)

Tossing coins
rflip(10) # default prob is 0.5

Do something repeatedly
do(5) * rflip(10, prob = 0.75)

Draw a simple random sample
sample(LETTERS, 10)
deal(Cards, 5) # poker hand

Resample with replacement
Small <- sample(KidsFeet, 10)
resample(Small)

Random permutation (shuffling)
shuffle(Cards)

Random values from distributions
rbinom(5, size = 10, prob = 0.7)
rnorm(5, mean = 10, sd = 2)
```

Two categorical variables

```
Contingency table with margins
tally(~ substance + sex,
  data = HELPrct, margins = TRUE)

Percentages by column
tally(~ sex | substance,
  data = HELPrct,
  format = "percent")

Mosaic plot
my_tbl <- tally(sex ~ substance,
  data = HELPrct)
mosaicplot(my_tbl, color = TRUE)
```



```
Test for proportions (approximate)
prop.test(homeless ~ sex,
  success = "homeless",
  data = HELPrct)
```

Distributions

```
Normal distribution function
pnorm(13, mean = 10, sd = 2)

Normal distribution function with graph
xpnorm(1.645, mean = 0, sd = 1)

Normal distribution quantiles
qnorm(0.95) # mean = 0, sd = 1

Normal distribution quantiles with graph
xqnorm(0.85, mean = 10, sd = 2)

Binomial density function ("size" means n)
dbinom(5, size = 8, prob = 0.65)

Binomial distribution function
pbinom(5, size = 8, prob = 0.65)

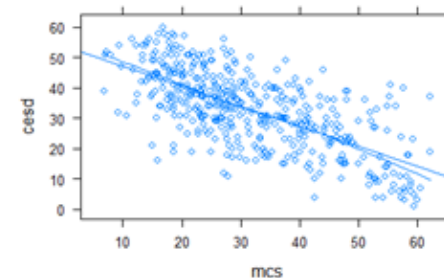
Central portion of distribution
cdist("norm", 0.95)
cdist("t", c(0.90, 0.99), df = 5)

Plotting distributions
plotDist("binom", size = 8,
  prob = 0.65, xlim = c(-1, 9))
plotDist("norm", mean = 10,
  sd = 2)
```

Two quantitative variables

```
Correlation coefficient
cor(cesd ~ mcs, data = HELPrct)

Scatterplot with regression line and smooth
xyplot(cesd ~ mcs,
  data = HELPrct,
  type = c("p", "r", "smooth"))
```



```
Simple linear regression
cesdmodel <- lm(cesd ~ mcs,
  data = HELPrct)
msummary(cesdmodel)

Prediction
lmfunction <- makeFun(cesdmodel)
lmfunction(mcs = 35)

Extract useful quantities
anova(cesdmodel)
coef(cesdmodel)
confint(cesdmodel)
rsquared(cesdmodel)

Diagnostics; plot residuals
histogram(~resid(cesdmodel),
  density = TRUE)
qqmath(~resid(cesdmodel))

Diagnostics; plot residuals vs. fitted
xyplot(resid(cesdmodel) ~
  fitted(cesdmodel),
  type = c("p", "smooth", "r"))
```

Categorical response, quantitative predictor

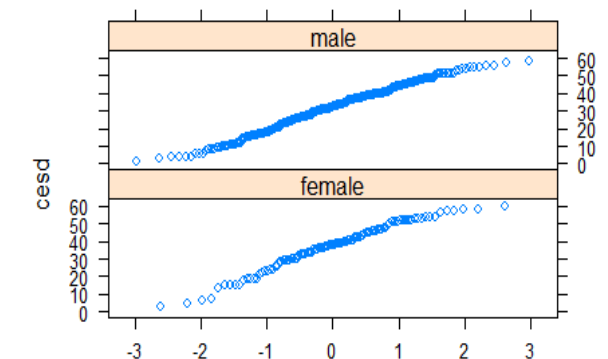
```
Logistic regression
logit_mod <- glm(homeless ~ age,
  data = HELPrct,
  family = binomial)
msummary(logit_mod)

Odds ratios and confidence intervals
exp(coef(logit_mod))
exp(confint(logit_mod))
```

Quantitative response, categorical predictor

```
Two-level predictor: Two-sample t test
Numeric summaries
favstats(~cesd | sex,
  data = HELPrct)

Comparative normal probability plot
qqmath(~cesd | sex, data = HELPrct,
  layout = c(1, 2)) # also bwplot
```



```
Two-sample t-test and confidence interval
result <- t_test(cesd ~ sex,
  data = HELPrct)
confint(result)
pval(result)
```

```
More than two levels: Analysis of variance
Numeric summaries
favstats(cesd ~ substance,
  data = HELPrct)
```

```
Graphic summaries
bwplot(cesd ~ substance,
  data = HELPrct, pch = "|")

Fit and summarize model
modsubstance <- lm(cesd ~ substance,
  data = HELPrct)
anova(modsubstance)

Which differences are significant?
pairwise <- TukeyHSD(modsubstance)
mplot(pairwise)
```

