Intro stats with mosaic

lattice version

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)
```

Loading packages

```
library(mosaic)
```

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)`	$k! = k(k-1)\ \ldots 1$

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, groups = w,
  data = mydata)
```

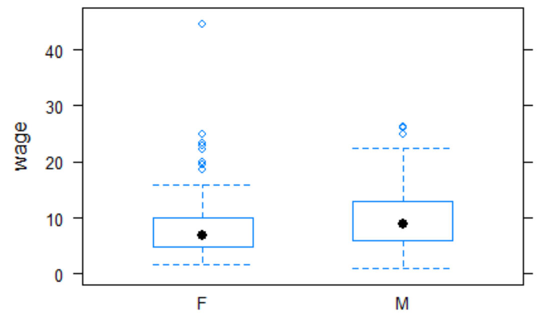
y : *y*-axis variable (*optional*)

x : *x*-axis variable (*required*)

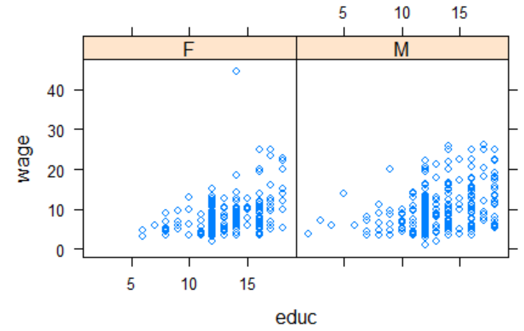
z : panel-by variable (*optional*)

w : color-by variable (*optional*)

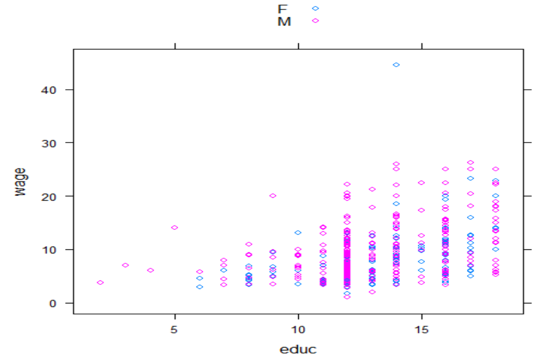
```
bwplot(wage ~ sex, data = CPS85)
```



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



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

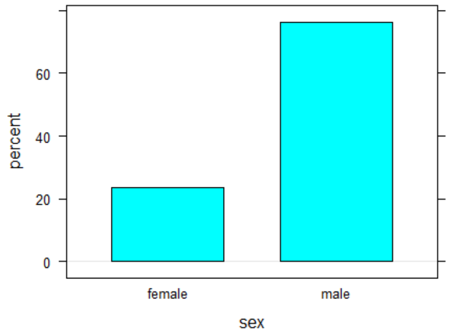
One categorical variable

Counts by category

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

Percentages by category

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



Tests and confidence intervals

Exact test

```
result1 <-
  binom.test(~ (homeless ==
  "homeless"), data = HELPrct)
```

Approximate test (large samples)

```
result2 <-
  prop.test(~ (homeless ==
  "homeless"), data = HELPrct)
```

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

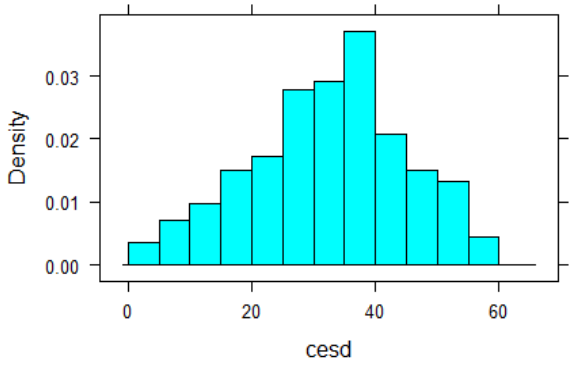
```
favstats(cesd ~ sex,
  data = HELPrct)
```

Quantiles

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

Histogram

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



Normal probability plot

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

Density plot

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

Dot plot

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

One-sample *t*-test

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

Extract confidence intervals and *p*-values

```
confint(result)
pval(result)
```

RStudio® is a trademark of RStudio, Inc. • CC BY Michael Laviolette • statman54@gmail.com

Adapted from *A Student's Guide to R* by NJ Horton, R Pruim & DT Kaplan • Updated: 07/18

Data manipulation

From `dplyr` package
For details, see Tidyverse cheatsheet

Drop, rename, or reorder variables
```
select()
```
Create new variables from existing ones
```
mutate()
```
Retain specific rows from data
```
filter()
```
Sort data rows
```
arrange()
```
Compute summary statistics by group
```
group_by()
summarize()
```

Importing data

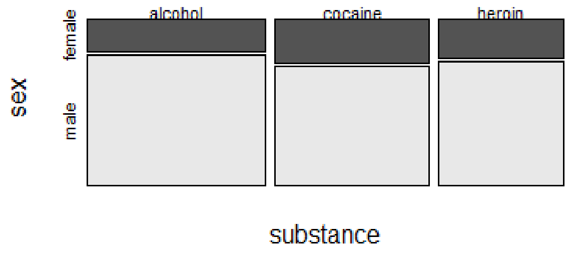
Import data from file or URL
```
MustangPrice <-
  read.file("C:/MustangPrice.csv")
# NOTE: R uses forward slashes!
Dome <-
  read.file("http://www.mosaic-
  web.org/go/datasets/Dome.csv")
```

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,
  margins = TRUE,
  data = HELPrct)
```
Percentages by column
```
tally(~ sex | substance,
  format = "percent",
  data = HELPrct)
```
Mosaic plot
```
mosaicplot(~ substance + sex,
  color = TRUE, data = HELPrct)
```



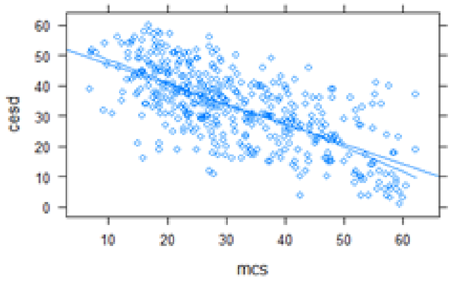
Chi-square test
```
xchisq.test(~ substance + sex,
  data = HELPrct,
  correct = FALSE)
```

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,
  type = c("p", "r", "smooth"),
  data = HELPrct)
```



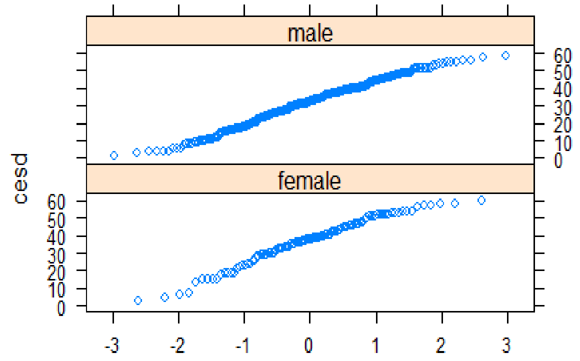
Simple linear regression
```
cesdmodel <- lm(cesd ~ mcs,
  data = HELPrct)
msummary(cesdmodel)
```
Prediction
```
lmfunction <- makeFun(cesdmodel)
lmfunction(mcs = 35)
```
Extract useful quantities
```
anova(cesdmodel)
coef(cesdmodel)
confint(cesdmodel)
rsquared(cesdmodel)
```
Diagnostics; plot residuals
```
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

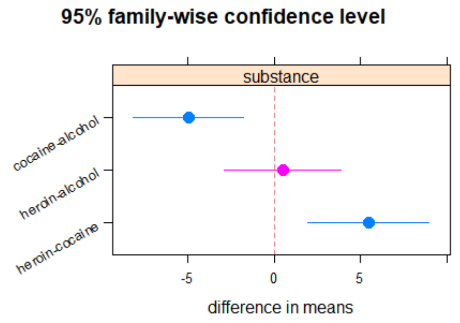
Logistic regression
```
logit_mod <-
  glm(homeless ~ age + female,
  family = binomial, data = HELPrct)
msummary(logitmod)
```
Odds ratios and confidence intervals
```
exp(coef(logit_mod))
exp(confint(logit_mod))
```

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
```



Dotplot for smaller samples
```
xyplot(sex ~ length, alpha = 0.6,
  cex = 1.4, data = KidsFeet)
```
Two-sample *t*-test and confidence interval
```
result <- t.test(cesd ~ sex,
  var.equal = FALSE, data = HELPrct)
confint(result)
```
More than two levels: Analysis of variance
Numeric summaries
```
favstats(cesd ~ substance,
  data = HELPrct)
```
Graphic summaries
```
bwplot(cesd ~ substance, pch = "|",
  data = HELPrct)
```
Fit and summarize model
```
modsubstance <- lm(cesd ~ substance,
  data = HELPrct)
anova(modsubstance)
```
Which differences are significant?
```
pairwise <- TukeyHSD(modsubstance)
mplot(pairwise)
```



RStudio® is a trademark of RStudio, Inc. • CC BY Michael Laviolette • statman54@gmail.com

Adapted from *A Student’s Guide to R* by NJ Horton, R Pruim & DT Kaplan • Updated: 07/18