

Intro stats with mosaic

ggformula 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 ("A and B") is TRUE if both
  A and B are TRUE
| A | B ("A or B") is TRUE if one or both
  of A and B are TRUE
%in% inclusion; 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,
     color = ~ w)
```

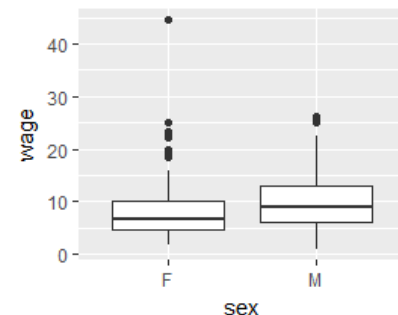
y : y-axis variable (*optional*)

x : x-axis variable (*required*)

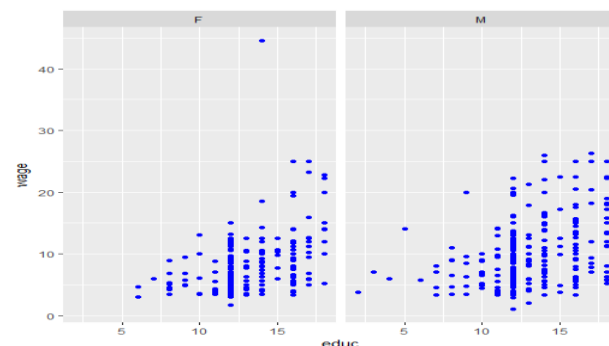
z : panel-by variable (*optional*)

w : color-by formula (*optional*)

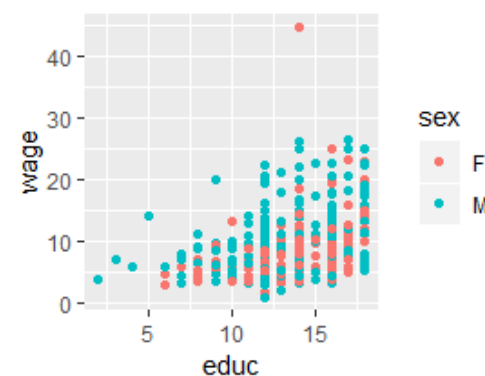
```
gf_boxplot(wage ~ sex,
            data = CPS85)
```



```
gf_point(wage ~ educ | sex,
         data = CPS85, color = "blue")
```



```
gf_point(wage ~ educ,
         data = CPS85, color = ~ sex)
```



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

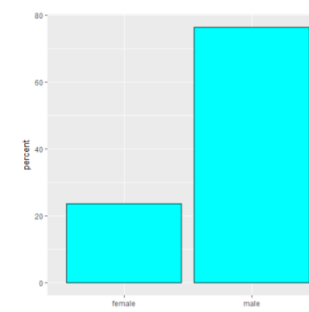
```
tally(~ sex, data = HELPrct)
```

Percentages by category

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

Bar graph of percentages

```
gf_percents(~ sex,
            data = HELPrct, fill = "cyan",
            color = "black")
```



Tests and confidence intervals

Exact test

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

Approximate test (large samples)

```
result2 <- prop.test(~ homeless ==
                    "homeless", data = HELPrct,
                    alternative = "less",
                    p = 0.4)
```

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

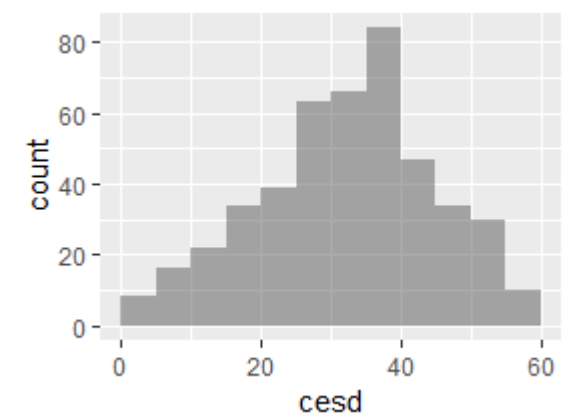
```
favstats(cesd ~ sex,
         data = HELPrct)
```

Quantiles

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

Histogram

```
gf_histogram(~ cesd,
            data = HELPrct, binwidth = 5,
            center = 2.5)
```



Normal probability plot

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

Density plot

```
gf_dens(~ cesd, data = HELPrct,
        color = "blue", size = 1.25)
```

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, gender = sex))

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

Extract 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 in file paths

kidsfeet <-
  read.file("http://www.mosaic-
  web.org/go/datasets/kidsfeet.csv")
```

Randomization and simulation

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

Toss 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(substance ~ sex,
  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)

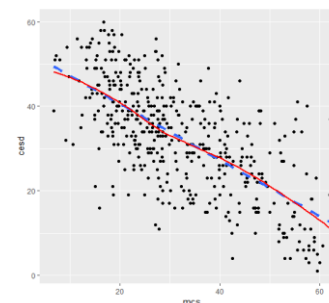
Central portion of distribution
cdist("norm", 0.95)
cdist("t", c(0.90, 0.99), df = 5)

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
gf_point(cesd ~ mcs,
  data = HELPrct) %>%
  gf_lm(size = 1.5, linetype =
    "dashed") %>%
  gf_smooth(color = "red")
```



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

Prediction
lm_fun <- makeFun(cesdmodel)
lm_fun(mcs = 35)
```

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

Diagnostics; plot residuals
gf_dhistogram(~resid(cesdmodel))
gf_qq(~resid(cesdmodel))

Diagnostics; plot residuals vs. fitted
gf_point(resid(cesdmodel) ~
  fitted(cesdmodel))
```

Categorical response, quantitative predictor

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

Odds ratios and confidence intervals
library(broom)
tidy(logit_mod, conf.int = TRUE,
  exponentiate = TRUE)
```

Quantitative response, categorical predictor

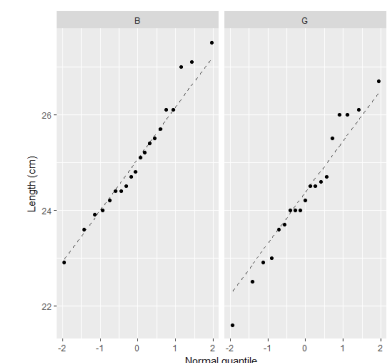
Two-level predictor: two-sample *t* test

Numeric summaries

```
favstats(~length | sex,
  data = KidsFeet)
```

Graphic summaries

```
gf_qq(~ length | sex,
  data = KidsFeet) %>%
  gf_qqline() %>%
  gf_labs(x = "Normal quantile",
    y = "Length (cm)")
```



Two-sample *t*-test and confidence interval

```
result <- t_test(cesd ~ sex,
  data = HELPrct)
result # view results
confint(result)
pval(result)
```

More than two levels (Analysis of variance)

Numeric and graphic summaries

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

Fit and summarize model

```
mod <- lm(age ~ substance,
  data = HELPrct)
anova(mod)
```

Which differences are significant?

```
mpplot(TukeyHSD(mod))
```

