A `mosaic` Sampler

Randall Pruim, Calvin College

Can you use R with beginners?

Yes. Anyone who can learn to use a TI calculator can learn to use R.

- But just like with the TI calculator, proficiency is not immediate
- Instructors need to have a plan for teaching and using R.

Can you use R with beginners?

Yes. Anyone who can learn to use a TI calculator can learn to use R.

- But just like with the TI calculator, proficiency is not immediate
- Instructors need to have a plan for teaching and using R.

Students who have trouble with R usually have other problems.

- R reveals more problems than it causes.
- Diagnostic (and teaching) questions:
  - What do you want the computer to do?
  - What does it need to know to do that?

Less Volume, More Creativity

Less Volume, More Creativity

"A lot of times you end up putting in a lot more volume, because you are teaching fundamentals and you are teaching concepts that you need to put in, but you may not necessarily use because they are building blocks for other concepts and variations that will come off of that . . . In the offsea- son you have a chance to take a step back and tailor it more specifically towards your team and towards your players."

– Mike McCarthy, Head Coach, Green Bay Packers

Less Volume, More Creativity

"A lot of times you end up putting in a lot more volume, because you are teaching fundamentals and you are teaching concepts that you need to put in, but you may not necessarily use because they are building blocks for other concepts and variations that will come off of that . . . In the offsea- son you have a chance to take a step back and tailor it more specifically towards your team and towards your players."

– Mike McCarthy, Head Coach, Green Bay Packers

"Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away."

– Antoine de Saint-Exupery

Less Volume, More Creativity

The `mosaic` choices:

- Use a formula interface whenever possible
- Use lattice graphics
- Use RStudio (free from `rstudio.com`)
- Provide web access to RStudio (local server at Calvin)

The `mosaic` package is available on CRAN and includes

- functions that make R easier for beginners
- tools for resampling
- a vignette that lists a possible minimal set of R functions

Help

```
apropos()
?
??
example()
```

Arithmetic

Basic arithmetic is very similar to a calculator.

```
# basic ops: + - * / ^ ( )
log()
exp()
sqrt()
```

```
log10()
abs()
choose()
factorial()
uniroot()   # root finder
```

Randomization/Simulation

```
rflip()      # mosaic
do()         # mosaic
sample()     # mosaic augmented
resample()   # with replacement
```

```
shuffle()    # mosaic
rbinom()
rnorm()      # etc, if needed
```

Formula Theme

The following syntax (often with some parts omitted) is used for graphical summaries, numerical summaries, and inference procedures.

```
fname( y ~ x | z, data=...,
                  groups=... )
```

For plots

- `y`: is y-axis variable
- `x`: is x-axis variable
- `z`: conditioning variable (separate panels)
- `groups`: conditioning variable (overlaid graphs)

For other things `y ~ x | z` can usually be read `y` is modeled by (or depends on) `x` separately for each `z` .
See the sampler for examples.

Distributions

```
pbinom(); pnorm();
pchisq(); pt()
qbinom(); qnorm();
qchisq(); qt()
plotDist();     # mosaic
```

Numerical Summaries

These functions have a formula interface to match plotting.

```
mean()       # mosaic augmented
median()     # mosaic augmented
sd()         # mosaic augmented
var()        # mosaic augmented
quantile()   # mosaic augmented
favstats()   # mosaic
tally()      # mosaic
```

Graphics (mostly lattice)

`lattice` is not the only option, but I find it works well because (a) it allows for easy multi-variable plots with good default settings, and (b) `lattice` uses the formula interface.

```
bwplot()
xyplot()
histogram()     # mosaic
qqmath()
densityplot()
plotFun()       # mosaic
```

```
ladd()          # mosaic
dotPlot()       # mosaic
bargraph()      # mosaic
mosaic()        # in vcd package
xhistogram()    # mosaic
xqqmath()       # mosaic
```

Inference

```
binom.test()  # mosaic augmented
prop.test()   # mosaic augmented
chisq.test()
t.test()
lm()          # linear models
anova( lm() )
summary( lm() )
makeFun( lm() )   # mosaic
resid( lm() )
plot( lm() )
TukeyHSD( lm () ) # mosaic aug
plot( TukeyHSD( lm() ) )
```

```
confint()      # mosaic augmented
pval()         # mosaic
fisher.test()
xchisq.test()  # mosaic
power.t.test()
power.prop.test()
wilcox.test()
```

Data

```
read.csv()
summary()
names()
head()
subset()
factor()
c()
cbind()
rbind()
```

```
merge()
```

Taking Advantage of Formulas

Anatomy:

```
> fname(y ~ x | z, data = ..., groups = ...)
```

- Unneeded parts are omitted.

Taking Advantage of Formulas

Anatomy:

```
> fname(y ~ x | z, data = ..., groups = ...)
```

- Unneeded parts are omitted.

Physiology: `y` depends on `x`, conditioned on `z`

- Plots:
  - `y` along vertical axis
  - `x` along horizontal axis
  - `z` divides plot into subplots (panels)
  - `groups` used to overlay subsets with different colors/shapes
- Numerical summaries augmented to handle formulas
- Linear Models can generally be fit by replacing a plotting function with `lm()`.

Taking Advantage of Formulas: tally()

```
> tally(substance ~ sex,
+     data=HELPrct)


         sex
substance female  male
  alcohol  0.336 0.408
  cocaine  0.383 0.321
  heroin   0.280 0.272
  Total    1.000 1.000
```

```
> tally(~ substance & sex,
+     data=HELPrct)


         sex
substance female male Total
  alcohol     36  141   177
  cocaine     41  111   152
  heroin      30   94   124
  Total      107  346   453
```

Taking Advantage of Formulas: Numerical Summaries

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

female   male
  36.3   35.5
```

```
> sd(age ~ sex | homeless, data = HELPrct)

female.homeless   male.homeless   female.housed
           6.66            8.61            8.13
    male.housed        homeless          housed
           6.71            8.26            7.17
```

Also works for `var()`, `median()`, `max()`, `min()`, `IQR()`, `sum()`, `prop()`, `count()`

Taking Advantage of Formulas: Testing

```
> binom.test(~sex, data = HELPrct)



Exact binomial test

data:  HELPrct$sex
number of successes = 107, number of trials = 453,
p-value < 2.2e-16
alternative hypothesis: true probability of success is not
95 percent confidence interval:
 0.198 0.278
sample estimates:
probability of success
                 0.236
```

Just the Facts Ma'am

R's output can sometimes be overly verbose for beginners.

```
> interval(t.test(age ~ sex, data = HELPrct))


mean in group female   mean in group male
               36.25                35.47
               lower                upper
               -0.88                 2.45
               level
                0.95
```

```
> pval(t.test(age ~ sex, data = HELPrct))


p.value
  0.354
```

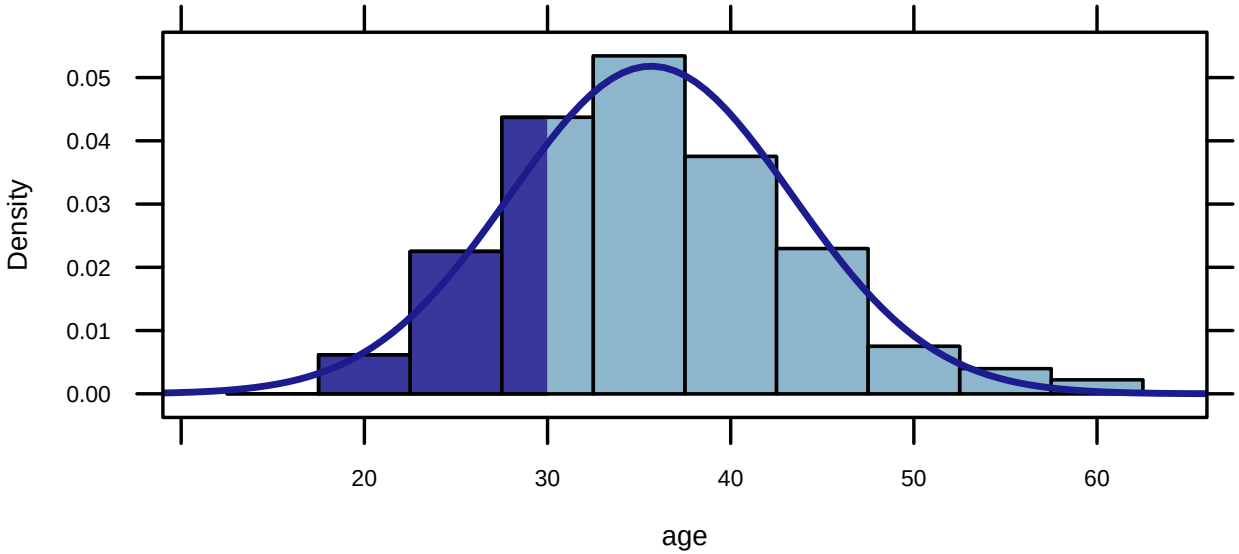
Xtra Output

Other times, it is too terse.

```
> xchisq.test(xtabs(~sex + substance, data = HELPrct))



Pearson's Chi-squared test

data:  xtabs(~sex + substance, data = HELPrct)
X-squared = 2.03, df = 2, p-value = 0.3631

  36.00    41.00    30.00
( 41.81) ( 35.90) ( 29.29)
[0.8068] [0.7236] [0.0173]
<-0.898> < 0.851> < 0.131>

 141.00   111.00    94.00
(135.19) (116.10) ( 94.71)
[0.2495] [0.2238] [0.0053]
< 0.500> <-0.473> <-0.073>

key:
observed
(expected)
[contribution to X-squared]
<residual>
```

Xtra Graphics

```
> xhistogram( ~age , data=HELPrct, fit='normal',
+     width=5, groups = age > 30)
```

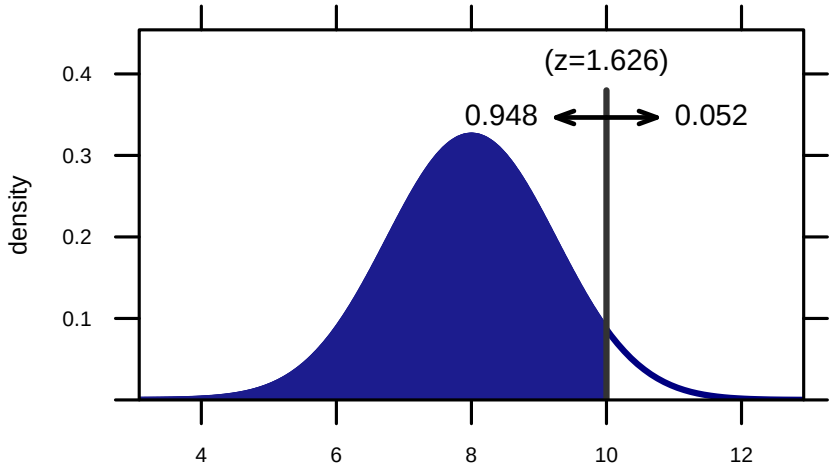


Other features:

- Easy horizontal and vertical reference lines.

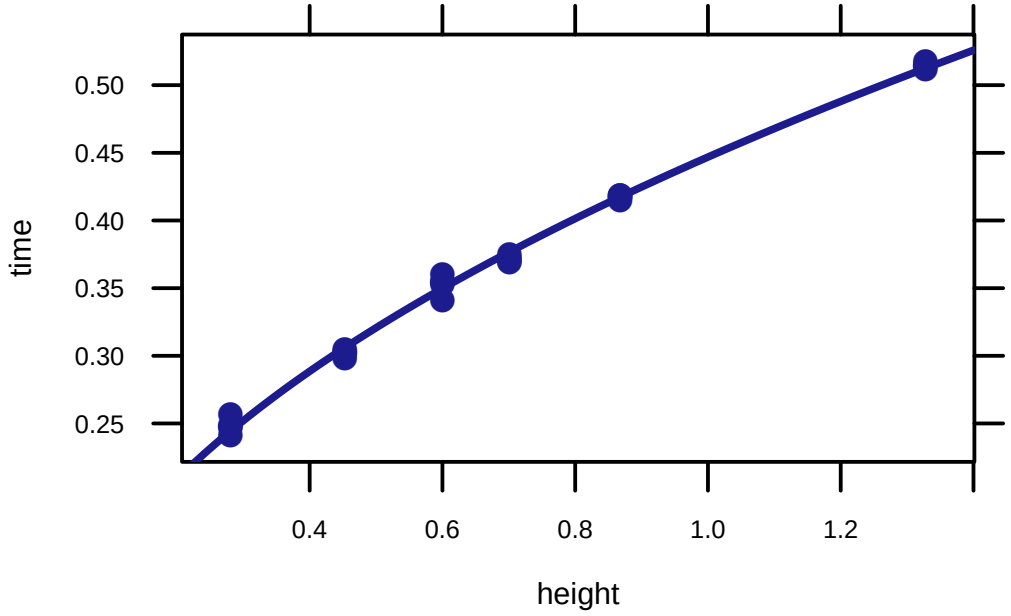
More Xtras

```
> xpnorm(10, mean = 8, sd = 1.23)



If X ~ N(8,1.23), then

P(X <= 10) = P(Z <= 1.626) = 0.948
P(X >  10) = P(Z >  1.626) = 0.052
[1] 0.948
```



More Xtras

```
> require(fastR)
> model <- lm(time ~ sqrt(height), data = balldrop)
> fit <- makeFun(model)
> xyplot(time ~ height, data = balldrop)
> plotFun(fit(height) ~ height, add = TRUE)
```



Tossing Coins

```
> rflip(10)



Flipping 10 coins [ Prob(Heads) = 0.5 ] ...

H H H H T T H H H T

Result: 7 heads.
```

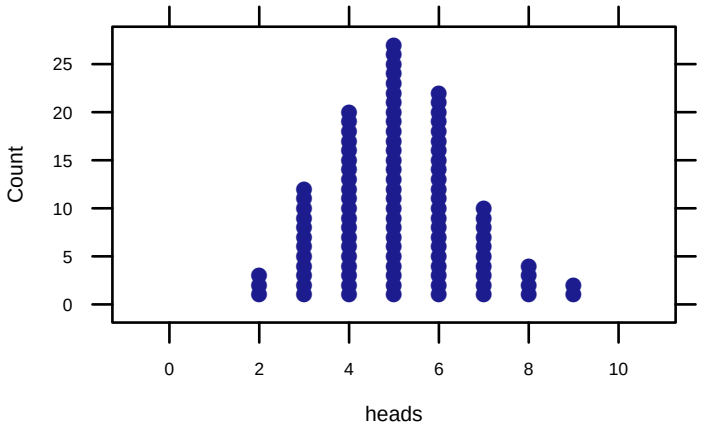
This can be used for simulations early in the course, before students know what a binomial distribution is. We just need a way to replicate the coin tossing easily.

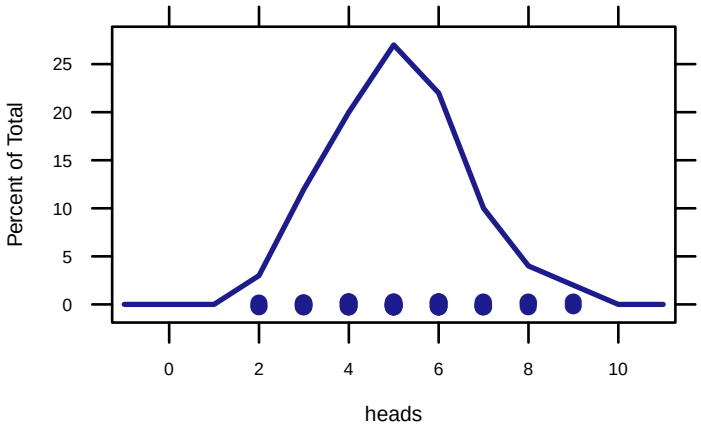
Tossing Coins

```
> rflip(10)



Flipping 10 coins [ Prob(Heads) = 0.5 ] ...

T H T H T H T H H H

Result: 6 heads.
> coinTosses <- do(100) * rflip(10)
> head(coinTosses, 2)


   n heads tails
1 10     3     7
2 10     4     6
```

Additional Plots

```
> mybreaks <- 0.5 + (-1:10)
> dotPlot(~heads, data = coinTosses, breaks = mybreaks)
> freqpolygon(~heads, data = coinTosses, breaks = mybreaks)
```





do()ing the mosiac shuffle()

```
> do(1) * lm(age ~ sex, HELPrct)


  Intercept sexmale sigma r.squared
1      36.3  -0.784  7.71   0.00187
```

```
> null.dist <- do(1000) * lm(age ~ shuffle(sex), HELPrct)
> head(null.dist, 1)


  Intercept sexmale sigma r.squared
1        36  -0.405  7.72  0.000498


> with(null.dist, tally(abs(sexmale) >= 0.784), format = "prop")



 TRUE FALSE Total
  342   658  1000
```

Data is a Distribution, Too

We have implemented functions paralleling `rnorm()`, `pnorm()`, and `qnorm()` for data distributions.

```
> rdata(10, age, data = HELPrct)

 [1] 29 29 49 58 34 41 40 35 37 30

> pdata(30, age, data = HELPrct)

[1] 0.256

> qdata(0.5, age, data = HELPrct)

50%
 35
```

Calculus

Differentiation:

```
> f <- D(A * sin(x + B) ~ x, A = 1, B = 0)
```

```
> f(pi)
```

```
[1] -1
```

```
> f(pi, A = 3, B = pi)
```

```
[1] 3
```

```
> randx <- runif(4, -pi, pi)
```

```
> f(randx) - cos(randx)
```

Anti-differentiation:

```
> F <- antiD(dnorm(x) ~ x)
> # F(0) == 0 by default
> F(randx) - pnorm(randx)
```

```
[1] -0.5 -0.5 -0.5 -0.5
```

```
> # Using G(-Inf) == 0 gives pdf
> G <- antiD(dnorm(x) ~ x, from = -Inf)
```

```
> G(randx) - pnorm(randx)
```

```
[1] -0.5 -0.5 -0.5 -0.5
```

```
> G(2)
```

The mosaic team

R Pruim

Calvin C

D Kaplan

Macalaster C

N Horton

Smith C

You

http://www.mosaic-web.org

The mosaic team

R Pruim

Calvin C

D Kaplan

Macalaster C

N Horton

Smith C

You

http://www.mosaic-web.org