R. Pruim

Enough R for Intro Stats

May, 2014

Help

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

Basic Calculations

Basic calculation is very similar to a calculator.

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

```
log10()
abs()
choose()
```

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.

```
goal( 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 differently for each z'.

See the sampler for examples.

Distributions

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

Numerical Summaries

These functions have a formula interface to match plotting.

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

```
quantile()     # mosaic augmented
prop()         # mosaic
perc()         # mosaic
rank()
IQR()          # mosaic augmented
min(); max()   # mosaic augmented
```

Graphics (mostly lattice)

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

```
ladd()       # mosaic
dotPlot()    # mosaic
bargraph()   # mosaic
xqqmath()    # mosaic
```

Interactive Graphics (RStudio)

```
mPlot(data=HELPrct, 'scatter')
mPlot(data=HELPrct, 'boxplot')
mPlot(data=HELPrct, 'histogram')
```

Inference

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

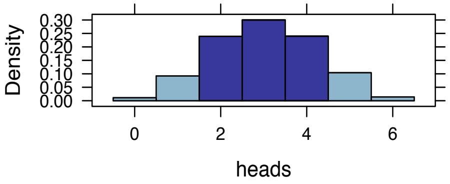
```
confint()        # mosaic augmented
pval()           # mosaic
fisher.test()
xchisq.test()    # mosaic
model <- glm()  # logistic regression
```

Data

```
read.file()      # mosaic
nrow(); ncol()
summary()
str()
names()
head()
subset()
factor()
c()
cbind(); rbind()
transform()
```

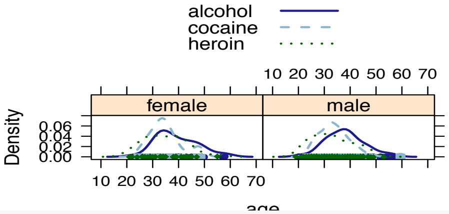
```
merge()
relevel()
ntiles()         # mosaic
cut()
```

```
rflip(6)


Flipping 6 coins [ Prob(Heads) = 0.5 ] ...

T H T H H T

Number of Heads: 3 [Proportion Heads: 0.5]

do(2) * rflip(6)

  n heads tails   prop
1 6     1     5 0.1667
2 6     2     4 0.3333

coins <- do(1000) * rflip(6)
tally(~heads, data = coins)


  0   1   2   3   4   5   6
 11  92 239 300 240 104  14

tally(~heads, data = coins, format = "perc")


   0    1    2    3    4    5    6
 1.1  9.2 23.9 30.0 24.0 10.4  1.4

tally(~(heads >= 5 | heads <= 1), data = coins)


 TRUE FALSE
  221   779

histogram(~heads, data = coins, width = 1,
    groups = (heads >= 5 | heads <= 1))
```

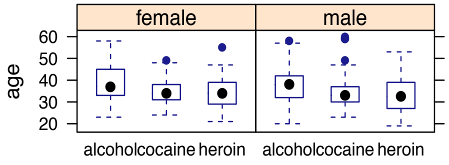


```
tally(~sex + substance, data = HELPrct)

        substance
sex      alcohol cocaine heroin
  female      36      41     30
  male       141     111     94

mean(age ~ sex, data = HELPrct)

female   male
 36.25  35.47

diffmean(age ~ sex, data = HELPrct)

diffmean
 -0.7841

favstats(age ~ sex, data = HELPrct)

  .group min Q1 median   Q3 max  mean
1 female  21 31     35 40.5  58 36.25
2   male  19 30     35 40.0  60 35.47
     sd   n missing
1 7.585 107       0
2 7.750 346       0
```

```
densityplot(~age | sex, groups = substance,
    data = HELPrct, auto.key = TRUE)
```

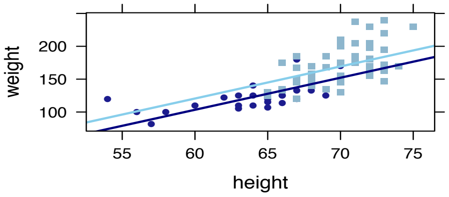


```
bwplot(age ~ substance | sex, data = HELPrct)
```

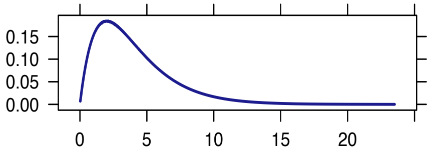


```
pval(binom.test(~sex, data = HELPrct))

  p.value
1.932e-30

confint(t.test(~age, data = HELPrct))

mean of x      lower      upper      level
    35.65      34.94      36.37       0.95
```

```
model <- lm(weight ~ height + gender,
            data=Heightweight)
wt <- makeFun(model)
wt( height=72, gender="male")

    1
179.1

xyplot(weight ~ height, groups=gender,
       data=Heightweight)
plotFun(wt(h,gender="male") ~ h, add=TRUE,
           col="skyblue")
plotFun(wt(h,gender="female") ~ h, add=TRUE,
           col="navy")
```



```
plotDist("chisq", df = 4)
```


0.15
0.10
0.05
0.00
0
5
10
15
20