A couple dozen functions suffice to carry out your work in Introduction to Statistical Modeling. This sheet provides the names of functions, a review of formula syntax, and some examples of use.

Help

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

Arithmetic

Basic arithmetic is very similar to a calculator.

```
# basic ops: + - * / ^ ( )
log()
exp()
sqrt()
log10()
abs()
```

Randomization/Iteration

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

Graphics

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

Numerical Summaries

These functions have a formula interface to match plotting.

```
mean()   # mosaic augmented
median() # mosaic augmented
sd()     # mosaic augmented
var()    # mosaic augmented
tally()  # mosaic
qdata()  # mosaic
pdata()  # mosaic
IQR()
```

Model Building and Inference

```
mm()      # mosaic
lm()      # linear models
glm()     # for logistic models
resid()
fitted()
confint()
anova()
summary()
makeFun() # mosaic
listFun() # devel
```

Interactive

```
mLM()
mLineFit()
mCI()
mLinAlgebra()
mHypTest()
mPower()
```

Formula Theme

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

```
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 or depends on x separately for each z .

Data and Variables

```
fetchData()   # mosaic
names()
head()
levels()
subset()
with()
transform()
as.factor()
merge()
rank()
```

Model Terms

```
# All cases the same:
response ~ 1
# Main effects & intercept
response ~ X + Y
# Exclude intercept
# (Rarely used.  Be careful!)
response ~ X + Y - 1
# Main effects and interaction:
response ~ X * Y
# Pure interaction (Rarely used.)
response ~ X:Y
#Polynomial terms:
response ~ poly(X,2)
# Random model vectors (pedagogical)
response ~ rand(2) # mosaic
```

Common Example Datasets

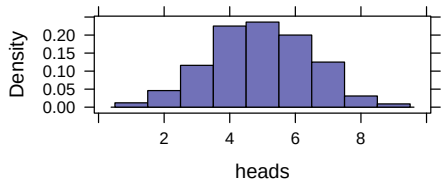
Can be used directly with `data=`:

```
Galton # heights
CPS85 # wages
KidsFeet
Marriage
SAT
```

Read in with `fetchData()`:

```
utils = fetchData("utilities.csv")
alder = fetchData("alder.csv")
grades = fetchData("grades.csv")
courses = fetchData("courses.csv")
# Load software in development:
fetchData("m155development.R")
```

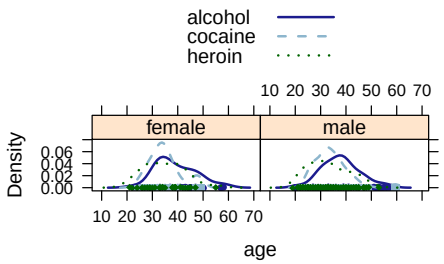
```
rflip(10)


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

T H T H H T H H H T

Result: 6 heads.


do(2) * rflip(10)

   n heads tails
1 10     8     2
2 10     5     5

results <- do(1000) * rflip(10)
tally(~heads, data = results)


    1     2     3     4     5     6
   12    46   116   225   236   200
    7     8     9 Total
  125    31     9  1000

xhistogram(~heads, data = results, width = 1)
```



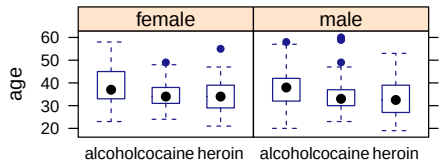

```
tally(~(heads > 8 | heads < 2), data = results)


 TRUE FALSE Total
   21   979  1000
```

```
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

mean(age ~ substance, data = HELPrct)

alcohol cocaine  heroin
  38.20   34.49   33.44

sd(age ~ substance, data = HELPrct)

alcohol cocaine  heroin
  7.652   6.693   7.986
```

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




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




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

mean in group female
              36.252
  mean in group male
              35.468
               lower
              -0.880
               upper
               2.448
               level
               0.950
```

```
anova(lm(age ~ sex + substance, data = HELPrct))

Analysis of Variance Table

Response: age
            Df Sum Sq Mean Sq F value
sex          1     50      50    0.91
substance    2   1997     999   18.06
Residuals  449  24823      55
            Pr(>F)
sex           0.34
substance 2.8e-08
Residuals
```

```
xyplot(Sepal.Length ~ Sepal.Width, data = iris,
    groups = Species)
```