RStudio

About RStudio

R is an industry grade statistical software package which is available for free under the GNU General Public License for the platforms Linux, Mac OS, and Windlows. RStudio is an integrated environment that employs R for its calculations. Its additional features are very useful, and highly recommended for this course. RStudio can be run from a server through a web browser, or it can be installed on your own computer. Should you prefer the latter approach, it requires that R first be installed; R can be downloaded from `https://cran.r-project.org`. You can then download and install RStudio from `https://rstudio.com/products/rstudio/download/`.

Accessing RStudio via a web browser

One advantage of running RStudio from a server is that, once you have an account, you can access your work using any browser-equipped device. Calvin has its own RStudio server, and accounts have been created for all students enrolled in this course. When you point your browser to the url `https://rstudio.calvin.edu:8787/`, you will arrive at a login screen. Simply use your Calvin credentials, the same as when accessing email.

Packages

- They can provide
  - desired functionality
  - datasets

```
data(package = "datasets")  # lists data sets included in package of same name
```

- recommend: fastR2, mosaic, mosaicData, ggformula

Probability intro

Experiment 1: Twelve True/False responses are required of a student who doesn't know the questions.

Question: Suppose student gets 3 correct. How likely is that outcome?

```
rflip(12)    # lists result of 12 flips of a fair coin, outcomes are H and T
```

```
do(10) * rflip(12)   # returns data frame; each row summarizes result of Exp. 1
```

```
manyRuns <- do(3000) * rflip(12)   # stores result in data frame now named x
head(manyRuns, n=8)     # returns the first 8 rows of output
gf_histogram(~heads, data=manyRuns)   # frequency histogram
gf_dhistogram(~heads, data=manyRuns)   # like above, but rescaled so area=1
nrow(manyRuns)    # predictably 3000
subset(manyRuns, heads==3)     # also  filter(x, heads==3)   works similarly
nrow( subset(manyRuns, heads==3) ) / 3000
```

The result of the last command indicates relative frequency of "3 correct" in 3000 runs of Experiment 1. Some terms:

- random variable : # of heads (in 12 flips)
- sample space : 0, 1, 2, 3, ..., 12
- event 3 (heads)
- frequentist view of probability can get approx. result as ratio $\frac{\#\text{ of successes}}{\#\text{ of runs}}$

Can do the above with zeros and ones

```
c(0, 1)   # a container with two elements, 0 and 1
sample(c(0, 1))   # samples from container without replacement until empty
resample(c(0, 1))   # samples twice from container with replacement
resample(c(0, 1), size=12)   # samples 12 times from container with replacement
sum(resample(c(0, 1), size=12))    # counts how many 1's were drawn

manyRuns <- do(3000) * sum(resample(c(0, 1), size=12))
head(manyRuns)
```

, treats 0 and 1 as equally likely

, since 0's contribute nothing to a sum

Experiment 2: An infinitely long list of True/False questions are unknown to the student. The student nust answer until she gets 3 correct.

Question: Suppose student requires 12 flips. How likely is that outcome?

```
cumsum(resample(c(0, 1), size=2))    # 2 is not enough flips
cumsum(resample(c(0, 1), size=20))   # 20 is usually enough flips
match(3, cumsum(resample(c(0, 1), size=100)))   # finds when 3rd Head occurs
manyRuns2 <- do(1000) * match(3, cumsum(resample(c(0, 1), size=100)))
head(manyRuns2)
nrow( filter(manyRuns2, match=12) )
```

to get 3 1's

(not guaranteed to be enough)

The last result counts how often it took exactly 12 tries to get 3 correct in 1000 runs of Experiment 2. To get "empirical probability", divide this result by 1000.

Interesting(?): Same event, 3 correct in 12 tries, has different (empirical) probability depending on which experiment.