Basic Linear Algebra Subprograms (BLAS)

How to use other people's hard work to make your statistical code run faster

Amy F. Szczepański
Art of Problem Solving

Linear Regression: Least squares calculation

$$\{y_i, x_{i1}, x_{i2}, \ldots, x_{ip}\}_{i=1}^{p}$$

$$y_i = \beta_0 \cdot 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

← **Matrix algebra!**

5000 by 5000 matrix. Time the calculation $\mathbf{X}^T\mathbf{X}$.

This should be typed all on one line.

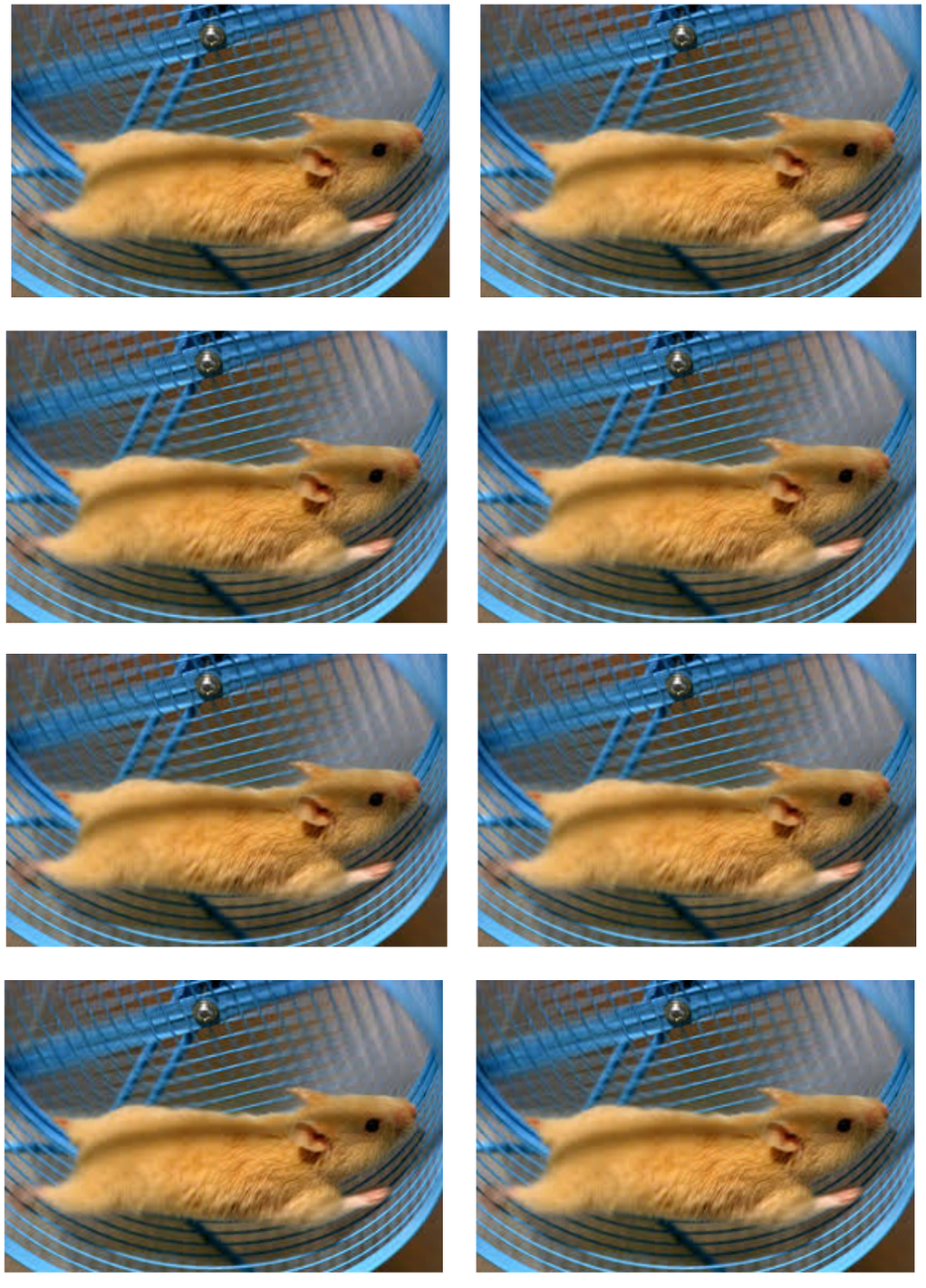
```
system.time({ x <-
replicate(5e3, rnorm(5e3));
tcrossprod(x) })
```

Result should look something like:

```
   user  system elapsed
 13.187   0.262   6.304
```

Computer Architecture and BLAS

vs.



Using a faster BLAS on a Mac

- R's default BLAS is in a subfolder of `/Library/Frameworks/R.framework`

- **Rename** R's BLAS to set it aside:

```
cd /Library/Frameworks/R.framework/Versions/Current/Resources/lib

mv libRblas.dylib libRblas.dylib.bak
```

- **Create a symbolic link** to tell R to use system BLAS:
(This is all one line.)

```
ln -s /System/Library/Frameworks/Accelerate.framework/Versions/
Current/Frameworks/vecLib.framework/Versions/Current/libBLAS.dylib
libRblas.dylib
```

Using a faster BLAS on another OS

- **Linux:**
  Use your preferred package manager to install or update `libblas`, `libatlas`, and/or `libopenblas`.
  Use `update-alternatives` or another tool to manage symbolic links to the BLAS library.

- **Windows:**
  Option 1: Use Microsoft R Open
  Option 2: See recommendations at:
  https://cran.r-project.org/bin/windows/base/rw-FAQ.html#Can-I-use-a-fast-BLAS_003f

Telling the system how many cores to use

- **Mac:** With Xcode installed, open the **Instruments** app. (Use Spotlight to find it.)
  Preferences → CPUS

- **Linux:**
  If you launch R from the command line, set the environment variable `OMP_NUM_THREADS`.
  If you launch R from a GUI, use an R command such as `Sys.setenv(OMP_NUM_THREADS=4)`

- **Windows:** Google “set number of cores Windows”