Basic R

Create a script.

Create a variable:

```
> x = 2
> x
[1] 2
```

Simple math.

Concatenate:

```
> myfirstvector = c(1,4,2)
> myfirstvector
[1] 1 4 2
```

Repeat:

```
> intercept1 = rep(1,7)
> intercept1
[1] 1 1 1 1 1 1 1
```

Create a matrix:

```
> amatrix = matrix(c(1,2,3,4,5,6),2,3)
> amatrix
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Summary statistics:

```
mean(myfirstvector)     var()     length()
```

Loading data:

```
labdata =
read.csv("http://home.cc.umanitoba.ca/~godwinrt/7010/
testdata.csv")
```

Look at the data:

```
summary(labdata)      head(labdata)
```

Extract a variable from the dataframe:

```
labdata$y
```

Load all variables in the dataframe into memory:

```
attach(labdata)
```

Plot the data:

```
plot(x2,y)
pairs(~ y + x2 + x3)
```

Estimate the model $y = \beta_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$, by OLS:

```
myOLS = lm(y ~ x2 + x3)
summary(myOLS)
```

Extract statistics from the regression:

```
yhat = myOLS$fitted.values
coef = myOLS$coefficients
resid = myOLS$residuals
rsquared = summary(myOLS)$r.squared
```

Plot a fitted line:

```
plot(x3,y, main = "Title!", col = "red")
abline(coef[1],coef[3])
```

Miscellaneous R

Getting help (Google first!):

```
?plot
```

Installing a package

Monte Carlo

Basic idea: (1) Create the population model; (2) Pretend we don't know the population model and calculate statistics; (3) Compare the calculated statistics to the population model. Monte Carlo relies on (pseudo) random number generation.

Random numbers (try this command at least twice):

```
rnorm(10)
```

Setting a random seed (try twice):

```
set.seed(7010)
rnorm(10)
```

1. Create the population model

Use the model: $y = 5 + 3x_2 + 2x_3 + \varepsilon$

```
beta1 = 5
beta2 = 3
beta3 = 2
```

Set the sample size:

```
n = 500
```

Generate the *X* data:

```
x2 = runif(n)
x3 = rnorm(n)
```

Generate the random error term:

```
eps = rnorm(n)
```

"Draw" a sample of *y* data:

```
y = beta1 + beta2*x2 + beta3*x3 + eps
```

2. Ignore the pop. model, estimate by OLS

```
monte = lm(y ~ x2 + x3)
b1 = monte$coef[1]
b2 = monte$coef[2]
b3 = monte$coef[3]
```

3. Compare!

Monte Carlo Loop

Now, step (3) wasn't very interesting, because we only performed a single Monte Carlo loop. If we want to emulate the sampling distribution of the OLS estimators, we need to perform the above steps many, many times.

To do this we can use a "for loop". We will repeat the above steps 10,000 times:

```
nrep = 10000
```

We will need "empty" vectors to record all 10000 of the OLS estimates:

```
b1 = b2 = b3 = rep(0, nrep)
```

Perform the experiment:

```
for(j in 1:nrep){
y = beta1 + beta2*x2 + beta3*x3 + rnorm(n)
monte = lm(y ~ x2 + x3)
b1[j] = monte$coef[1]
b2[j] = monte$coef[2]
b3[j] = monte$coef[3]
}
```

Compare the sampling distribution of the OLS estimators to the pop. model:

```
mean(b1)

hist(b3)
```

If we didn't know OLS was unbiased, we'd have a pretty good idea now!