Lesson 3: Programming *Mathematica*, 1 of 3

Some Terms

- Notebook
- Cell
- Style
- Input
- Output
- Front End
- Pallete
- Demonstration
- Player
- Kernel

Language Basics

```
?InputForm
```

InputForm[*expr*] prints as a version of *expr* suitable for input to *Mathematica*. ≫

```
?StandardForm
```

StandardForm[*expr*] prints as the standard *Mathematica* two-dimensional representation of *expr*. ≫

```
?TraditionalForm
```

TraditionalForm[*expr*] prints as an approximation to the traditional mathematical notation for *expr*. ≫

Case Sensitivity :

```
In[1]:= a == A // Simplify

Out[1]= a == A

In[2]:= a == a // Simplify

Out[2]= True
```

Order of operations :

In[3]:=
```
(2 + 4) * 3
```
Out[3]=
```
18
```
In[4]:=
```
2 + 4 × 3
```
Out[4]=
```
14
```
Functions :

In[5]:=
```
f[x]
```
Out[5]=
```
f[x]
```
In[6]:=
```
f (x)
```
Out[6]=
```
f x
```

▪ Creating One Dimensional Lists

In[7]:=
```
{2, 3, -5}
```
Out[7]=
```
{2, 3, -5}
```
In[8]:=
```
{x, 5, "hi"}
```
Out[8]=
```
{x, 5, hi}
```
In[9]:=
```
Table[i^2, {i, 5}]
```
Out[9]=
```
{1, 4, 9, 16, 25}
```
In[10]:=
```
Table[a, {4}]
```
Out[10]=
```
{a, a, a, a}
```
In[11]:=
```
Table[i^2, {i, 0, 1, 0.3}]
```
Out[11]=
```
{0., 0.09, 0.36, 0.81}
```
In[12]:=
```
Range[5]
```
Out[12]=
```
{1, 2, 3, 4, 5}
```
In[13]:=
```
Table[i, {5}]
```
Out[13]=
```
{i, i, i, i, i}
```
In[14]:=
```
? Table
```

Table[*expr*, {i_{max}}] generates a list of i_{max} copies of *expr*.
Table[*expr*, {i, i_{max}}] generates a list of the values of *expr* when i runs from 1 to i_{max}.
Table[*expr*, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Table[*expr*, {i, i_{min}, i_{max}, *di*}] uses steps *di*.
Table[*expr*, {i, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Table[*expr*, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, …] gives a nested list. The list associated with i is outermost. ≫

In[15]:=
```
ConstantArray[23, 4]
```
Out[15]=
```
{23, 23, 23, 23}
```

■ Manipulating Lists

```
list1 = {a, b, c, d, e}
```

```
{a, b, c, d, e}
```

```
Length[list1]
```

```
5
```

```
list1[[2]]
```

```
b
```

```
Take[list1, 3]
```

```
{a, b, c}
```

```
? Take
```

Take[*list*, *n*] gives the first *n* elements of *list*.
Take[*list*, −*n*] gives the last *n* elements of *list*.
Take[*list*, {*m*, *n*}] gives elements *m* through *n* of *list*.
Take[*list*, seq_1, seq_2, …] gives a nested list in which elements specified by seq_i are taken at level *i* in *list*. ≫

```
Take[list1, {2, 4}]
```

```
{b, c, d}
```

```
list1[[2 ;; 4]]
```

```
{b, c, d}
```

```
? Part
```

expr[[*i*]] or Part[*expr*, *i*] gives the i^{th} part of *expr*.
expr[[−*i*]] counts from the end.
expr[[*i*, *j*, …]] or Part[*expr*, *i*, *j*, …] is equivalent to *expr*[[*i*]][[*j*]] ….
expr[[{i_1, i_2, …}]] gives a list of the parts i_1, i_2, … of *expr*.
expr[[*m* ;; *n*]] gives parts *m* through *n*.
expr[[*m* ;; *n* ;; *s*]] gives parts *m* through *n* in steps of *s*. ≫

```
RotateRight[list1]
```

```
{e, a, b, c, d}
```

```
NestList[RotateRight, list1, Length[list1]]
```

```
{{a, b, c, d, e}, {e, a, b, c, d}, {d, e, a, b, c},
 {c, d, e, a, b}, {b, c, d, e, a}, {a, b, c, d, e}}
```

```
NestList[RotateRight, list1, Length[list1]] // MatrixForm
```

$$\begin{pmatrix} a & b & c & d & e \\ e & a & b & c & d \\ d & e & a & b & c \\ c & d & e & a & b \\ b & c & d & e & a \\ a & b & c & d & e \end{pmatrix}$$

```
?RotateRight
```

RotateRight[*expr*, *n*] cycles the elements in *expr* *n* positions to the right.
RotateRight[*expr*] cycles one position to the right.
RotateRight[*expr*, {n_1, n_2, …}] cycles elements at successive levels n_i positions to the right. ≫

```
Sort[RotateRight[list1, 3]]

{a, b, c, d, e}

Join[list1, {f, g}]

{a, b, c, d, e, f, g}

Join[list1, x]

Join[{a, b, c, d, e}, x]

Flatten[NestList[RotateRight, list1, Length[list1]] ]

{a, b, c, d, e, e, a, b, c, d, d, e, a, b, c, c, d, e, a, b, b, c, d, e, a, a, b, c, d, e}

list2 = {a^2, b, b, c, d^2, b, e}

{a^2, b, b, c, d^2, b, e}

Position[list2, b]

{{2}, {3}, {6}}

Extract[list2, %]

{b, b, b}

Position[list2, _^2]

{{1}, {5}}

Extract[list2, %]

{a^2, d^2}

myTest[x_] = (x > 10) && (x < 50)

x > 10 && x < 50

Select[Table[i^2, {i, 10}], myTest]

{16, 25, 36, 49}
```

■ Multidimensional Lists

```
testmat = Table[a^i + b^j, {i, 3}, {j, 3}]

{{a + b, a + b^2, a + b^3}, {a^2 + b, a^2 + b^2, a^2 + b^3}, {a^3 + b, a^3 + b^2, a^3 + b^3}}

MatrixForm[testmat]
```

$$\begin{pmatrix} a+b & a+b^2 & a+b^3 \\ a^2+b & a^2+b^2 & a^2+b^3 \\ a^3+b & a^3+b^2 & a^3+b^3 \end{pmatrix}$$

```
Position[testmat, _^2 + _]
```

{{1, 2}, {2, 1}, {2, 2}, {2, 3}, {3, 2}}

```
Extract[testmat, %]
```

$\{a+b^2,\ a^2+b,\ a^2+b^2,\ a^2+b^3,\ a^3+b^2\}$

```
Map[g, testmat]
```

$\{g[\{a+b,\ a+b^2,\ a+b^3\}],\ g[\{a^2+b,\ a^2+b^2,\ a^2+b^3\}],\ g[\{a^3+b,\ a^3+b^2,\ a^3+b^3\}]\}$

```
? Map
```

Map[*f*, *expr*] or *f* /@ *expr* applies *f* to each element on the first level in *expr*.
Map[*f*, *expr*, *levelspec*] applies *f* to parts of *expr* specified by *levelspec*. ≫

```
Map[g, testmat, 2]
```

$\{g[\{g[a+b],\ g[a+b^2],\ g[a+b^3]\}],$
$g[\{g[a^2+b],\ g[a^2+b^2],\ g[a^2+b^3]\}],\ g[\{g[a^3+b],\ g[a^3+b^2],\ g[a^3+b^3]\}]\}$

```
% // MatrixForm
```

$$\begin{pmatrix} g[\{g[a+b],\ g[a+b^2],\ g[a+b^3]\}] \\ g[\{g[a^2+b],\ g[a^2+b^2],\ g[a^2+b^3]\}] \\ g[\{g[a^3+b],\ g[a^3+b^2],\ g[a^3+b^3]\}] \end{pmatrix}$$

```
Map[g, testmat, {2}]
```

$\{\{g[a+b],\ g[a+b^2],\ g[a+b^3]\},$
$\{g[a^2+b],\ g[a^2+b^2],\ g[a^2+b^3]\},\ \{g[a^3+b],\ g[a^3+b^2],\ g[a^3+b^3]\}\}$

```
% // MatrixForm
```

$$\begin{pmatrix} g[a+b] & g[a+b^2] & g[a+b^3] \\ g[a^2+b] & g[a^2+b^2] & g[a^2+b^3] \\ g[a^3+b] & g[a^3+b^2] & g[a^3+b^3] \end{pmatrix}$$

■ Example of lists: Sampling from the uniform distribution

```
ourData = Table[Random[], {10 000}];
```

```
Length[ourData]
```

10 000

```
Short[ourData]
```

{0.304756, 0.44377, ≪9996≫, 0.613116, 0.0223821}

```
? Mean
```

Mean[*list*] gives the statistical mean of the elements in *list*.
Mean[*dist*] gives the mean of the symbolic distribution *dist*. ≫

```
Mean[ourData]
```

0.495353

```
Variance[ourData]
```

```
0.0826139
```

Does it seem like Random[] generated Uniform Random Variables?

```
?UniformDistribution
```

UniformDistribution[{*min*, *max*}] represents a
continuous uniform statistical distribution giving values between *min* and *max*. ≫

```
?*Distribution
```

▼ System`

BernoulliDistribution	LaplaceDistribution
BetaBinomialDistribution	LogisticDistribution
BetaDistribution	LogNormalDistribution
BetaNegativeBinomialDistribution	LogSeriesDistribution
BinomialDistribution	MaxwellDistribution
CauchyDistribution	NegativeBinomialDistribution
ChiDistribution	NoncentralChiSquareDistribution
ChiSquareDistribution	NoncentralFRatioDistribution
DiscreteUniformDistribution	NoncentralStudentTDistribution
ExponentialDistribution	NormalDistribution
ExtremeValueDistribution	ParetoDistribution
FRatioDistribution	PoissonDistribution
GammaDistribution	RayleighDistribution
GeometricDistribution	StudentTDistribution
GumbelDistribution	TriangularDistribution
HalfNormalDistribution	UniformDistribution
HypergeometricDistribution	WeibullDistribution
InverseGaussianDistribution	ZipfDistribution

```
Mean[UniformDistribution[{0, 1}]]
```

$$\frac{1}{2}$$

```
Variance[UniformDistribution[{0, 1}]]
```

$$\frac{1}{12}$$

```
% // N
```

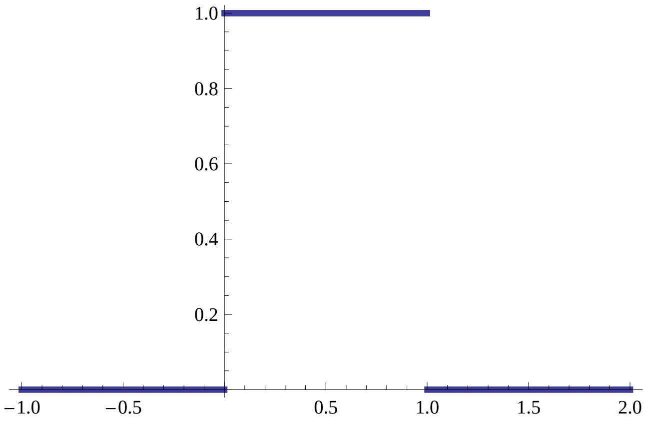
```
0.0833333
```

```
f[x_] := PDF[UniformDistribution[{0, 1}], x]
```

```
f[x]
```

$$\begin{cases} 1 & 0 \le x \le 1 \end{cases}$$

```
Plot[f[x], {x, -1, 2}, PlotStyle → Thickness[0.01]]
```




```
Integrate[f[x], {x, -Infinity, Infinity}]
```

1

```
Integrate[ x f[x], {x, -Infinity, Infinity}]
```

$\frac{1}{2}$

```
Integrate[ x^2 f[x], {x, -Infinity, Infinity}]
```

$\frac{1}{3}$

```
%31 - %30^2
```

$\frac{1}{12}$

```
? Random*
```

▼ System`

Random	RandomComplex	RandomPrime	RandomSample
RandomChoice	RandomInteger	RandomReal	RandomSeed

```
? RandomSample
```

RandomSample[{e_1, e_2, …}, n] gives a pseudorandom sample of n of the e_i.
RandomSample[{w_1, w_2, …} -> {e_1, e_2, …}, n]
gives a pseudorandom sample of n of the e_i chosen using weights w_i.
RandomSample[{e_1, e_2, …}] gives a pseudorandom permutation of the e_i. ≫

```
sample = RandomSample[ourData, 30]
```

```
{0.302693, 0.37064, 0.583611, 0.70708, 0.20229, 0.890997, 0.363054,
 0.948171, 0.989845, 0.657318, 0.188458, 0.692985, 0.68076, 0.135439, 0.705515,
 0.351483, 0.685066, 0.770434, 0.0241253, 0.842135, 0.837745, 0.538629,
 0.99207, 0.164469, 0.470614, 0.761288, 0.997831, 0.666971, 0.990504, 0.400457}
```

```
Mean[sample]
```

0.597089

```
samples = Table[RandomSample[ourData, 30], {100}];
```

```
samples[[1]]
```

```
{0.559851, 0.502773, 0.289647, 0.47802, 0.270841, 0.866202, 0.340389,
 0.305006, 0.637494, 0.740661, 0.846846, 0.67632, 0.618343, 0.225248, 0.696601,
 0.0831069, 0.0693631, 0.344197, 0.577508, 0.829848, 0.818782, 0.989969,
 0.833993, 0.240564, 0.807176, 0.678042, 0.739822, 0.243666, 0.682984, 0.382594}
```

```
samples[[100]]
```

```
{0.111759, 0.0789288, 0.997451, 0.624412, 0.628741, 0.25182, 0.34938, 0.915915,
 0.0514405, 0.307025, 0.963383, 0.217924, 0.597368, 0.631794, 0.277547,
 0.254864, 0.390124, 0.803803, 0.404806, 0.0421032, 0.767332, 0.500818,
 0.992485, 0.0816655, 0.174252, 0.611549, 0.54674, 0.524102, 0.899098, 0.275133}
```

```
means = Map[Mean, samples]
```

```
{0.545862, 0.475899, 0.387716, 0.432088, 0.478424, 0.479651, 0.451693, 0.462885, 0.535296,
 0.563002, 0.435162, 0.569283, 0.502188, 0.571075, 0.431634, 0.323477, 0.354051,
 0.442544, 0.454651, 0.535602, 0.460469, 0.54832, 0.514451, 0.416076, 0.520306, 0.497823,
 0.478605, 0.449416, 0.535307, 0.531644, 0.585825, 0.470822, 0.586766, 0.501242,
 0.443217, 0.512095, 0.402677, 0.489764, 0.399802, 0.477029, 0.397437, 0.534005,
 0.547564, 0.580781, 0.499145, 0.463369, 0.424216, 0.582127, 0.427738, 0.492026,
 0.500838, 0.505383, 0.488593, 0.437664, 0.547738, 0.513922, 0.499723, 0.464064,
 0.475797, 0.526908, 0.580574, 0.438944, 0.409058, 0.509437, 0.427171, 0.483015,
 0.512636, 0.445918, 0.522026, 0.411741, 0.472917, 0.505708, 0.465459, 0.534598,
 0.417698, 0.547659, 0.455752, 0.556894, 0.532771, 0.479994, 0.542324, 0.521229,
 0.585725, 0.461993, 0.479259, 0.512961, 0.527426, 0.49568, 0.502062, 0.487711, 0.496964,
 0.402537, 0.4982, 0.463683, 0.546745, 0.47318, 0.523226, 0.455717, 0.554938, 0.475792}
```

```
Variance[means]
```

```
0.00289479
```

```
exactVarOfUniformSample[n_] := Variance[UniformDistribution[{0, 1}]] / n
```

```
exactVarOfUniformSample[30] // N
```

```
0.00277778
```

```
samples = Table[RandomSample[ourData, 30], {1000}];
```

```
means = Map[Mean, samples];
```

```
Length[means]
```

```
1000
```

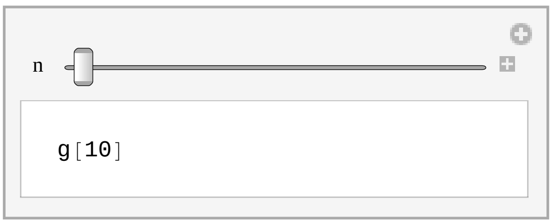
```
Variance[means]
```

```
0.00272096
```

In[18]:=

```
Manipulate[g[n],
 {n, 10, 40, 10}]
```

Out[18]=



In[19]:=
```
ourData = Table[Random[], {10 000}];
Manipulate[
 samples = Table[RandomSample[ourData, n], {1000}];
 means = Map[Mean, samples];
 {1/(12 n) // N, Variance[means]}
 ,
 {n, 10, 40, 10}]
```

Out[20]=