Random Matrix Theory Hints
(developed with Piet Brouwer)

We're generating M members of the GOE ensemble of NxN matrices, and calculating the difference between the eigenvalues in the middle of the range. Let's demonstrate how using a small sample.

In[1]:=
```
M = 200;
Size = 4;
```

To generate random symmetric matrices, we generate a matrix with entries drawn from a normal distribution, and then symmetrize it.

In[3]:=
```
Mat = RandomVariate[NormalDistribution[], {Size, Size}]
MatSym = Mat + Transpose[Mat]
MatrixForm[MatSym]
```

Out[3]=
```
{{0.712229, 0.924701, -0.828935, -0.437384},
 {-1.09302, 1.64093, -0.461153, -1.38089},
 {1.59872, -0.104854, -0.457538, -0.0570919},
 {0.286581, 0.301822, 0.126942, -0.397078}}
```

Out[4]=
```
{{1.42446, -0.168316, 0.769786, -0.150804},
 {-0.168316, 3.28186, -0.566006, -1.07907},
 {0.769786, -0.566006, -0.915075, 0.06985},
 {-0.150804, -1.07907, 0.06985, -0.794156}}
```

Out[5]//MatrixForm=

$$\begin{pmatrix} 1.42446 & -0.168316 & 0.769786 & -0.150804 \\ -0.168316 & 3.28186 & -0.566006 & -1.07907 \\ 0.769786 & -0.566006 & -0.915075 & 0.06985 \\ -0.150804 & -1.07907 & 0.06985 & -0.794156 \end{pmatrix}$$

Generate M random SizexSize arrays of Gaussian (normal) random numbers;symmetrize them; find their sorted eigenvalues,keep a list of the difference between the two middle ones. You can do this with a For loop (slow and inelegant)

In[6]:=
```
diffs = Table[0, {i, 1, M}];
M11 = Table[0, {i, 1, M}];
For[i = 1, i ≤ M, i++, {Mat = RandomVariate[NormalDistribution[], {Size, Size}];
  MatSym = Mat + Transpose[Mat];
  eigs = Sort[Eigenvalues[MatSym]];
  diffs[[i]] = eigs[[Size / 2 + 1]] - eigs[[Size / 2]]};
 M11[[i]] = MatSym[[1]][[1]];]
```

You can also do this with functional programming: it' s considered more elegant

in Mathematica world

In[9]:=
```
Symmetrized[m_] := m + Transpose[m];
EigSorted[m_] := Sort[Eigenvalues[m]]
EigDiffs[eigs_] := eigs[[Size / 2 + 1]] - eigs[[Size / 2]]
diffs = Map[EigDiffs[EigSorted[Symmetrized[#]]] &,
   RandomVariate[NormalDistribution[], {M, Size, Size}]];
M11 = Map[Symmetrized[#][[1, 1]] &, Table[Random[NormalDistribution[]],
    {m, 1, M}, {i, 1, Size}, {j, 1, Size}]];
```

Or, perhaps, with tables :

In[14]:=
```
Mats = RandomVariate[NormalDistribution[], {M, Size, Size}];
MatsSym = Table[Mats[[i]] + Transpose[Mats[[i]]], {i, 1, M}];
M11 = Table[MatsSym[[i]][[1, 1]], {i, 1, M}];
eigs = Table[Sort[Eigenvalues[MatsSym[[i]]]], {i, 1, M}];
diffs = Table[eigs[[i]][[Size / 2 + 1]] - eigs[[i]][[Size / 2]], {i, 1, M}];
```

Divide out by mean value of the splittings

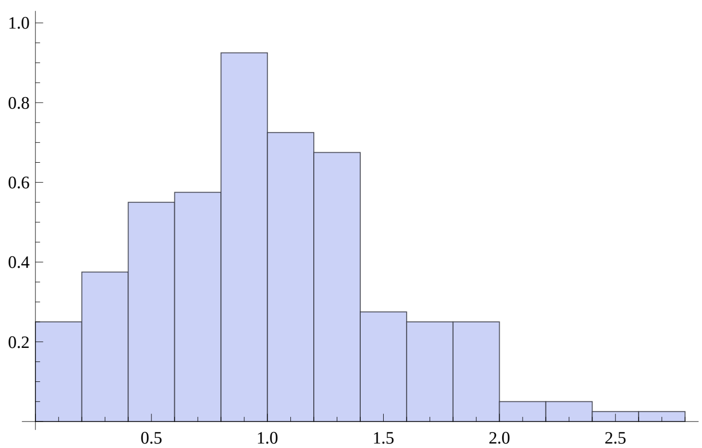
In[19]:=
```
diffAve = Mean[diffs]
```

Out[19]=
```
2.26609
```

Histogram generates a histogram of diffs. ‘Automatic ’tells it to choose the number of bins. ‘Probability ”normalizes the integral to one.

In[28]:=
```
Histogram[diffs / diffAve, Automatic, "PDF"]
```

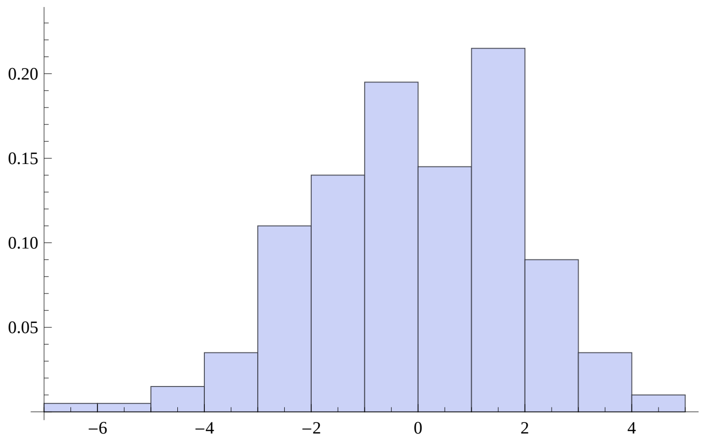
Out[28]=



Let's demonstrate how to compare simulation with theory by using a histogram for the diagonal element M11. First plot the histogram for M11; store as "histGauss" for later plotting with theory

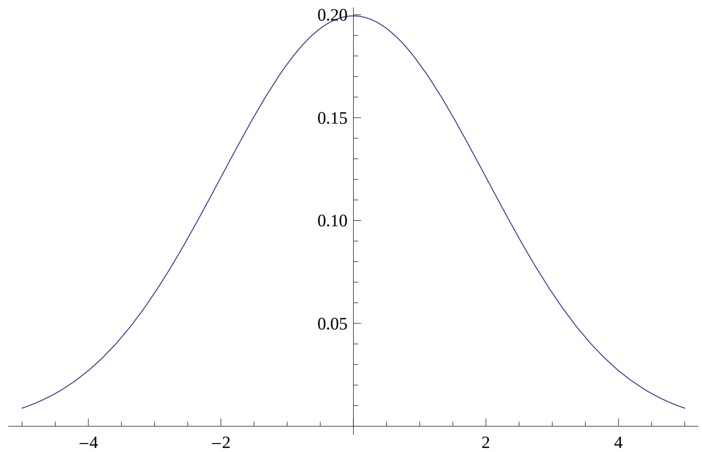
In[29]:=
```
histGauss = Histogram[M11, Automatic, "PDF"]
```

Out[29]=

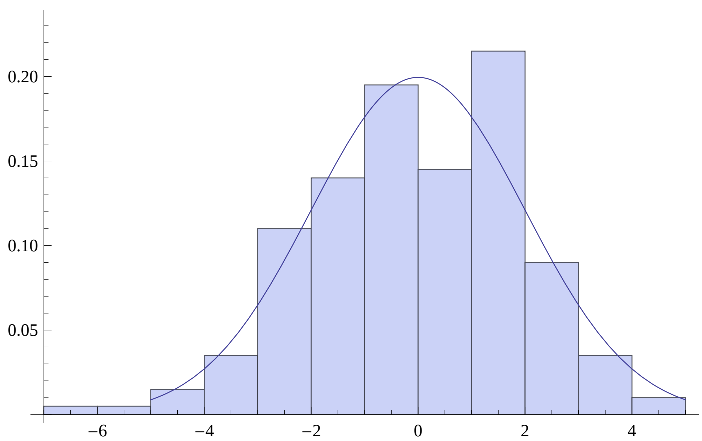


In[30]:=
```
(*Now,plot expected Gaussian fit*)
sigma = 2;
theoryGauss = Plot[(1 / Sqrt[2 Pi sigma^2]) Exp[-x^2 / (2 sigma^2)], {x, -5, 5}]

(*and show the comparison*)

Show[histGauss, theoryGauss]
```

Out[31]=



Out[32]=



We generate random +-1 matrices using RandomInteger (which produces 0/1, which we double and subtract one). Note that *Mathematica* subtracts one "componentwise". For example...

In[25]:=
```
MatPM = RandomInteger[{0, 1}, {Size, Size}] * 2 - 1
```

Out[25]=
```
{{1, 1, 1, 1}, {-1, 1, -1, -1}, {1, 1, -1, 1}, {1, -1, -1, -1}}
```

This is an inelegant way of symmetrizing ...

In[26]:=
```
MatPMSym =
  Table[If[i ≥ j, MatPM[[i]][[j]], MatPM[[j]][[i]]], {i, 1, Size}, {j, 1, Size}];
```

In[27]:=
```
MatrixForm[MatPMSym]
```

Out[27]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 \end{pmatrix}$$