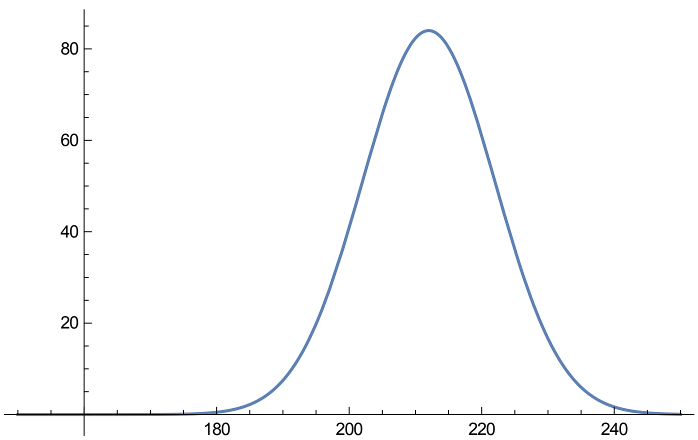
PHYS 310 -- Homework #2, Due Tuesday February 2, 2015

1. Home - written problem

To get estimates see handwritten solution sheet. You can check your estimates by plotting the function and comparing with the given figure.

```
A = 84
m  = 212
s = 10
gau[x_] := A*Exp[-1.0*(x-m )^2/(2*s^2)]
Plot[gau[x], {x, 150, 250}]
```




```
gau[222]
50.9486
```

2. Hughes and Hase, 3.2

see hand – written solution sheets

3. Hughes and Hase, 3.5

Mean weight 502 g, standard deviation 14 g.
What is probability that a bag contains less than 500 g?

```
s = 14
m  = 502
```

You can either integrate directly:

```
Integrate[PDF[NormalDistribution[m , s], x], {x, -∞, 500.0}]
```

```
0.443202
```

Or you can use CDF :

```
CDF[NormalDistribution[m , s], 500] // N
```

```
0.443202
```

Next question: In a sample of 1000 bags, how many will contain at least 530 g? To get that, integrate from 530 up to infinity.

```
Integrate[PDF[NormalDistribution[m , s], x], {x, 530.0, ∞}]
```

```
0.0227501
```

```
1.0 - CDF[NormalDistribution[m , s], 530]
```

```
0.0227501
```

This gave us probability, so for 1000 bags multiply result by 1000.

```
1000 * (1.0 - CDF[NormalDistribution[m , s], 530])
```

```
22.7501
```

4. Hughes and Hase 3.7

Poisson distribution. Radioactive decay with 58 experiments for one second.

```
counts = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
occ = {1, 0, 2, 3, 6, 9, 11, 8, 8, 6, 2, 1, 1}
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
```

```
{1, 0, 2, 3, 6, 9, 11, 8, 8, 6, 2, 1, 1}
```

Total number of experimental runs (as a check) :

```
numruns  = Total[occ]
```

```
58
```

Okay, that checks out.
(i) Now, total number of counts . Simplest is to do a dot product of occ and counts.

```
numcounts  = Dot[occ, counts]
```

```
423
```

(ii) mean count would then be the total number of counts divided by the number of experiments done. (=μ in last Tuesday class)

```
meancount = numcounts / numruns
```

$\frac{423}{58}$

```
meancount // N
```

7.2931

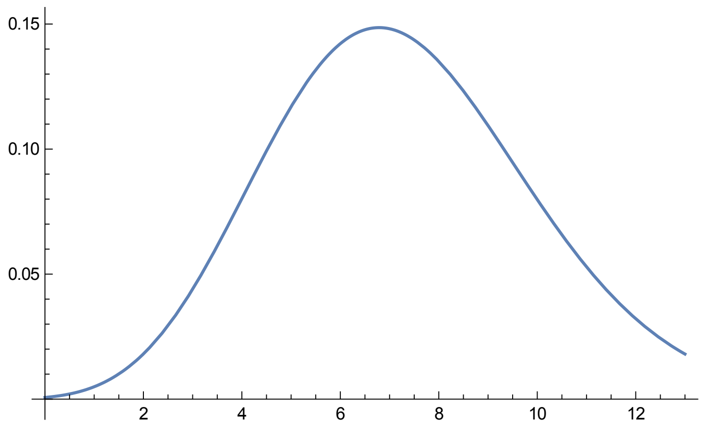
(iii) mean count rate? since each run is for a duration of τ=1 second, the mean count rate is just 7.29 counts/s.

Poisson distribution, defining function:

```
pdist[n_] := Exp[-1.0 * meancount] * meancount ^n / Factorial[n]
```

Let' s plot this to see if it looks reasonable.

```
Plot[pdist[n], {n, 0, 13}]
```

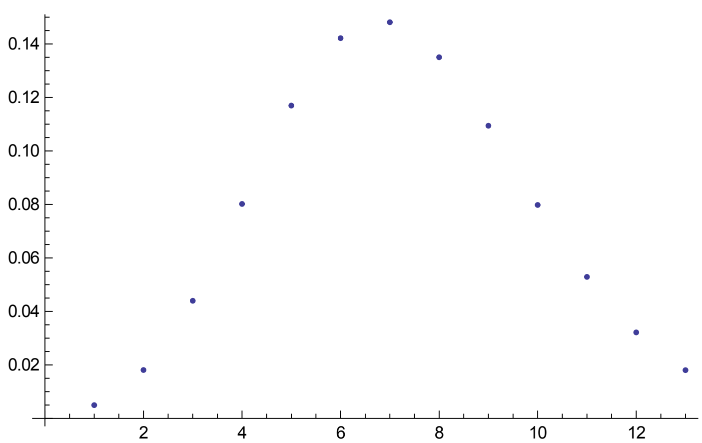



```
meancount * 1.0
```

7.2931

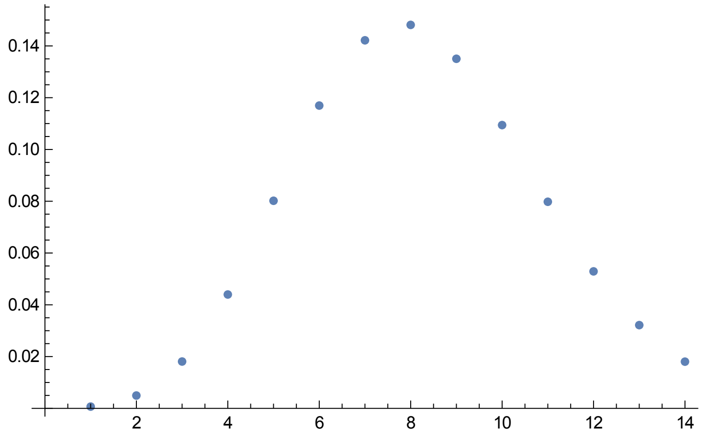
It isn' t a continuous function, so let' s do it as a discrete list using "Table" function

```
ListPlot[Table[{n, pdist[n]}, {n, 1, 13}]]
```



You could have done this using PoissonDistribution.

```
ListPlot[Table[PDF[PoissonDistribution[meancount *1.0], i], {i, 0, 13}]]
```

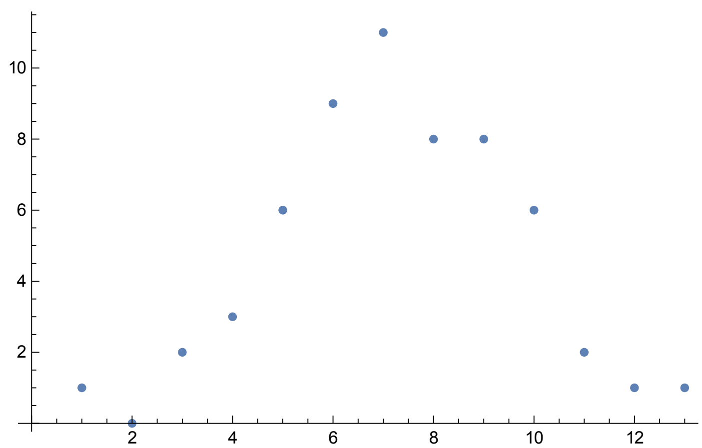


And now plot the actual data for comparison (but note that it isn' t normalized yet)

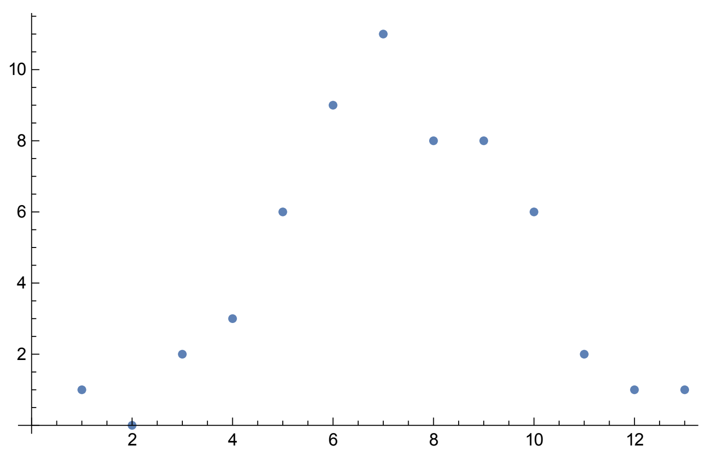
```
Transpose[{counts, occ}]
```

{{1, 1}, {2, 0}, {3, 2}, {4, 3}, {5, 6}, {6, 9}, {7, 11}, {8, 8}, {9, 8}, {10, 6}, {11, 2}, {12, 1}, {13, 1}}

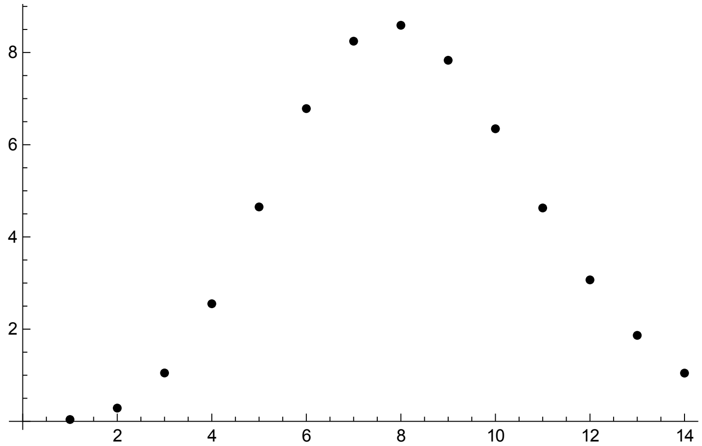
```
ListPlot[Transpose[{counts, occ}]]
```



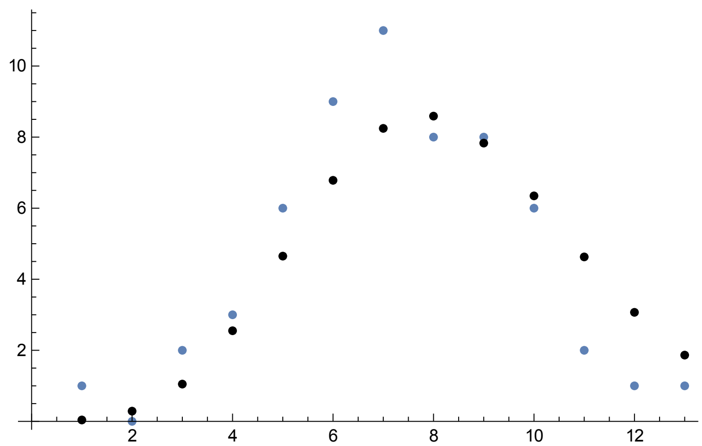

```
plotdatadist = ListPlot[Transpose[{counts, occ}]]
```



```
plotpoisson=
  ListPlot[Table[numruns *PDF[PoissonDistribution[meancount *1.0], i], {i, 0, 13}],
    PlotStyle → Black] (*note numruns  for comparison  below*)
```



```
Show[plotdatadist, plotpoisson]
```



Okay, now that I have had fun plotting these (and making sure that they look reasonable), let' s do the calculations that have been requested by the problem.

(i) Expected probability of occurrences of five counts or fewer :

```
fraclessthanfive= ∑_{n=0}^{5} pdist[n]
```

```
0.264849
```

```
CDF[PoissonDistribution[meancount ], 5] //N (*same  as above*)
```

```
0.264849
```

and therefore the answer for the expected number of occurences of five counts or fewer is here:

```
fraclessthanfive*numruns
```

```
15.3612
```

So, we' d expect around 15 of the runs to have 5 or fewer counts. Compare that to what we had :

```
Total[occ[[1 ;; 5]]]
```

```
12
```

Ballpark agreement.

(ii) Expected number of occurrences of 20 counts or more?

$$\text{fracmorethan20} = \sum_{n=20}^{\infty} \text{pdist}[n]$$

```
0.0000767424
```

```
1 - CDF[PoissonDistribution[meancount], 19] // N
```

```
0.0000767424
```

```
expectedmorethan20 = fracmorethan20 * numruns
```

```
0.00445106
```

In other words, we really wouldn' t expect any of the runs to give us more than 20 counts. Makes sense : if the mean is 7 and the standard deviation is Sqrt[7], then a count of 20 is

```
(20 - 7) / Sqrt[7] // N
```

```
4.91354
```

almost 5 standard deviations away from the mean, which is really unlikely.

5. Hughes and Hase, 3.8

Radioactive decay -- another Poisson distribution. 270 counts recorded during a 1 - minute period. Calculate :

(i) the mean count rate. If we use a minute for the time base, this seems trivial: the mean rate is just 270 counts/minutes.

(ii) Error in the mean rate: if we assume a Poisson distributions, then the error should just be Sqrt[N]

```
errorofmean = Sqrt[270] // N
```

```
16.4317
```

(iii) Fractional error

```
fracerror = errorofmean / 270 // N
```

```
0.0608581
```

If repeat experiment, but for 15 minutes, (iv) expected count

```
countfor15 = 270 * 15
```

```
4050
```

(v) Probability of obtaining exactly 4050 counts? Basically, the question is: What is the probability that we'll actually get precisely the mean? Of course, this is the peak of the distribution, but it is still going to

be a really small number.
Plug into Poisson

```
m  = 4050
Exp[-1*m ]*m ^m / (Factorial[m ]) // N
```

```
4050
```

```
0.00626864416476
```

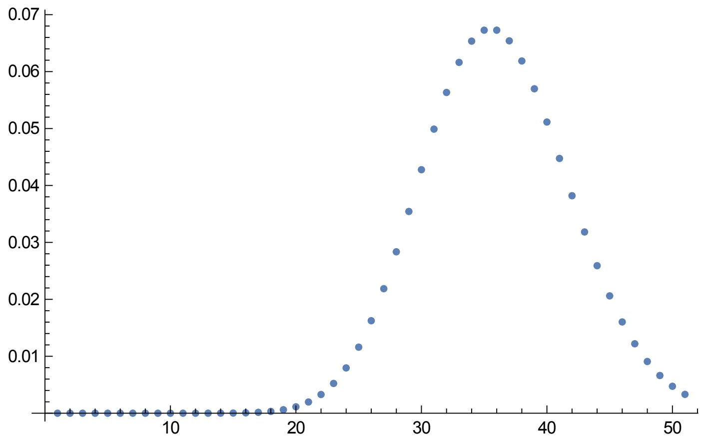
```
PDF[PoissonDistribution[countfor15], countfor15] // N (*same  as above *)
```

```
0.00626864416476
```

6. Hughes and Hase, 3.9

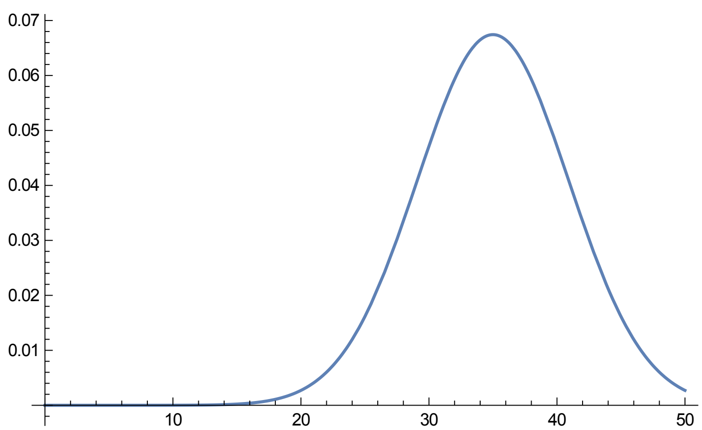
Poisson distribution with mean of 35.

```
graphpoisson = ListPlot[Table[PDF[PoissonDistribution[35], n], {n, 0, 50}]]
```

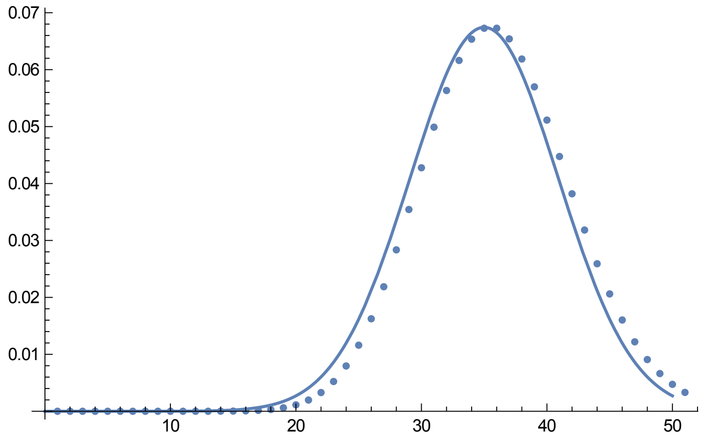


Now, plot this Poisson distribution on the same graph as a Gaussian with a mean of 35 and a standard deviation of Sqrt[35].

```
graphgauss = Plot[PDF[NormalDistribution[35, Sqrt[35]], x], {x, 0, 50}]
```



```
Show[graphpoisson, graphgauss]
```



The distributions look pretty similar, except for a slight skewing of the data. The Gaussians are symmetrical around the mean whereas the Poisson distributions are not.

For fun, play around with this by changing the mean and standard deviation. The differences are more and more pronounced the smaller the mean.

7. Simulation of PHYS 211 M & M Lab

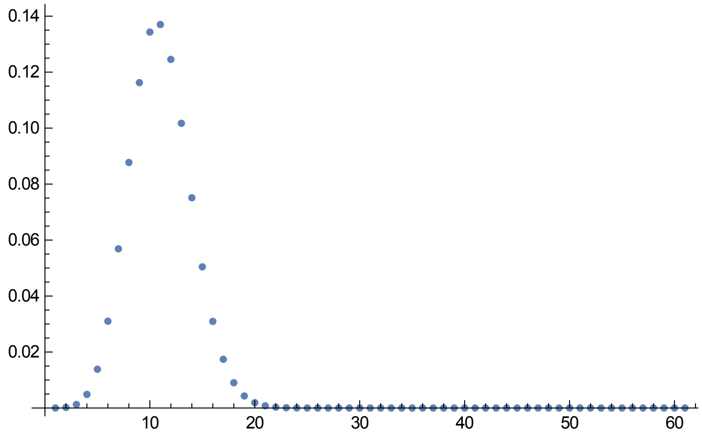
Assumptions :
All bags are statistically equivalent
All bags have 60 M&M's
There are 6 colors of M&M’s (brown, black, red, yellow, green, blue)
All sections have the same size (24 students)

```
nInBag = 60;
nBags = 24;
nSection = 200;
pbrown = 1.0 / 6.0;
meanParent  = nInBag * pbrown;
sdParent = Sqrt[nInBag * pbrown * (1 - pbrown)]
```

```
2.88675
```

Calculation of Parent Standard Deviation

```
ListPlot[
  Table[PDF[BinomialDistribution[nInBag, pbrown], i], {i, 0, nInBag}], PlotRange→All]
```




```
StandardDeviation[BinomialDistribution[nInBag, 1/6]]
```

$\frac{5}{\sqrt{3}}$

```
% // N
```

2.88675

Or

```
Sum [PDF[BinomialDistribution[nInBag, 1/6], i] (i-meanParent )^2, {i, 0, nInBag}]
```

8.33333

```
Sqrt[%]
```

2.88675

"Experiment"

"Make" a bag of M&Ms by generating a set of random numbers between 1 and 6 (inclusive). Let brown correspond to 1.

```
RandomInteger [{1, 6}, nInBag]
```

{2, 3, 4, 4, 4, 1, 1, 5, 4, 6, 5, 4, 4, 3, 4, 5, 6, 2, 1, 2, 1, 5, 4, 1, 3, 1, 5, 4, 2, 2,
  6, 2, 6, 4, 6, 2, 2, 1, 6, 2, 6, 6, 6, 6, 3, 2, 4, 3, 6, 4, 2, 2, 4, 6, 1, 2, 4, 4, 3, 5}

Count the brown M&Ms in random bag. Note delayed evaluation.

```
brownInBag := Count[RandomInteger [{1, 6}, nInBag], 1]
```

Find the number of brown M&Ms in all the bags in a section. Note delayed evaluation.

```
section := Table[brownInBag, {nBags}]
```

```
(* OR can do this without the intermediate definitionof "brownInBag *)
(* section = Table[Count[RandomInteger [{1,6},nInBag],1],{nBags}]; *)
```

```
Table[Count[RandomInteger [{1, 6}, nInBag], 1], {nBags}]
```

{8, 7, 11, 11, 6, 10, 6, 11, 14, 4, 7, 11, 15, 8, 11, 10, 9, 7, 13, 12, 9, 8, 11, 13}

```
Mean[section] // N
StandardDeviation[section] // N
```

10.4167

2.97544

Make a class of many sections. Record means from each section.

```
class = Table[Mean[section], {nSection}] // N;
```

```
(* OR can do this without the intermediate definitionof "section" *)
```

```
(* class =
  Table[Mean[Table[Count[RandomInteger [{1,6},nInBag],1],{nBags}]],{nSection}]//
    N  *)
```

```
m  = Mean[class]
s = StandardDeviation[class]
```

9.9575

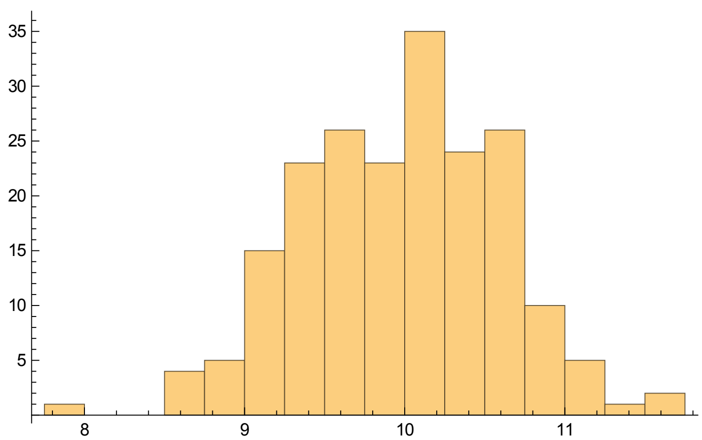
0.615498

Compare with predictions of Central Limit Theorem:

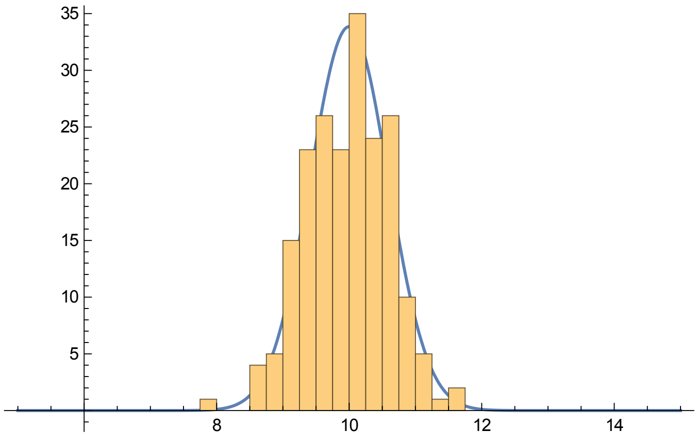
```
sdParent/Sqrt[nBags]
```

0.589256

```
histwidth= 0.25; histgraph = Histogram [class, {histwidth}]
```



```
gaussgraph = Plot[nSection*histwidth*
        PDF[NormalDistribution[meanParent , sdParent/Sqrt[nBags]], x], {x, 5, 15}];
Show[gaussgraph, histgraph]
```



The histogram compares well with a gaussian with the same mean and standard deviation. That' s what the Central Limit Theorem does for you -- even though we're dealing with binomial statistics, when we look at a large distribution of the averages, we find guassian behavior.