chapter 5

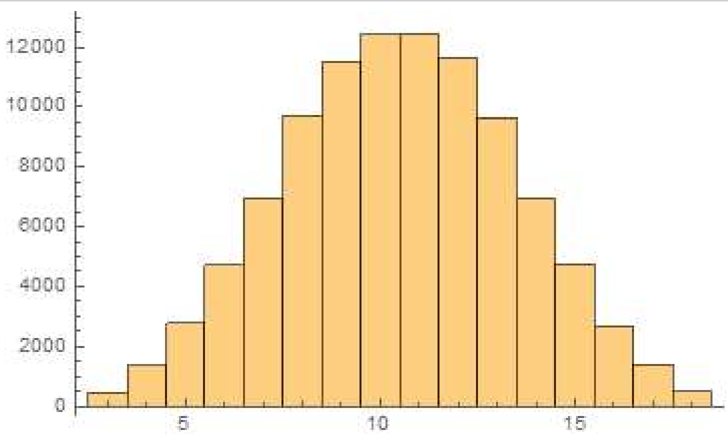
In [1]:
```
dice = Range[1,6]
```
Out[1]: {1, 2, 3, 4, 5, 6}

In [57]:
```
roll := RandomChoice[dice]
roll
```
Out[58]: 3

In [59]:
```
rolls := Table[roll, 3]
rolls
```
Out[60]: {6, 5, 6}

In [69]:
```
sum := Total[rolls]
sum
```
Out[70]: 10

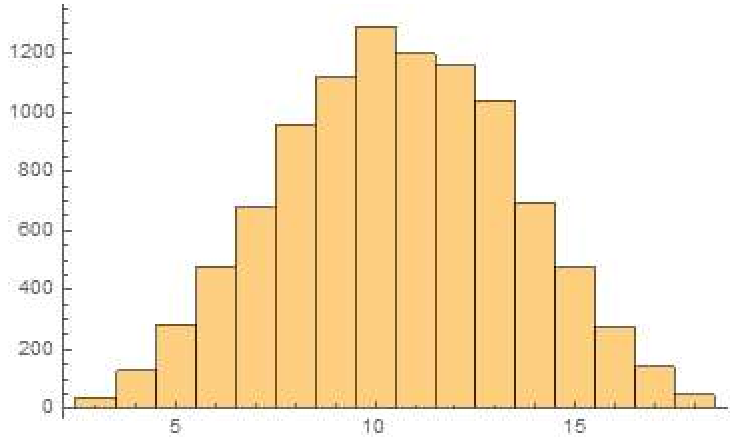
In [86]:
```
dist := Table[sum, 100000]
```

In [87]:
```
Histogram[dist, 36]
```
Out[87]:


In [ ]:
```

```

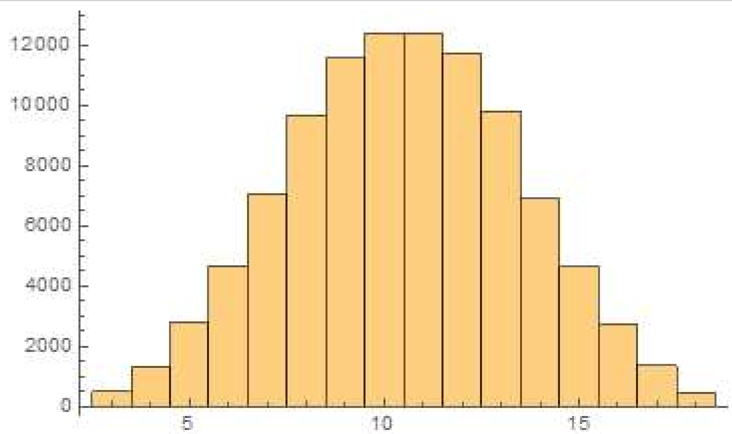
In [105]:
```
1 Histogram[Table[Total[RandomChoice[Range[1, 6], 3]], 10000]]
```

Out[105]:



In [218]:
```
1 Histogram[Table[Total[RandomVariate[DiscreteUniformDistribution[{1, 6}], 3]]
```

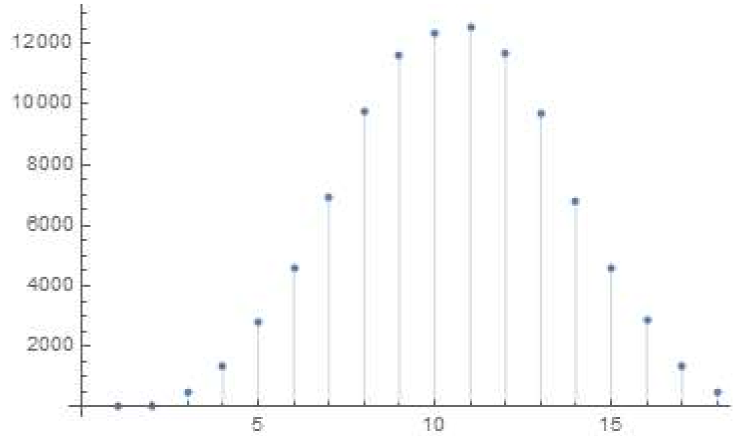
Out[218]:



In [219]:
```
1 results2 := Sort @ Table[Total[RandomVariate[DiscreteUniformDistribution[{1,
```

In [220]:
```
count2[i_] := Count[results2, i]
counts2 := Table[count2[n], {n, 1, 18}]
ListPlot[counts2, Filling->Axis]
```

Out[222]:



In [ ]:

In [ ]:

In [ ]:

Distribution of the maximum of 6 rolls of 3 die

In [181]:
```
results := Table[Max@ Table[Total[RandomVariate[DiscreteUniformDistribution[
```

In [182]:
```
count[i_] := Count[results, i]
```

In [183]:
```
count[17]
```

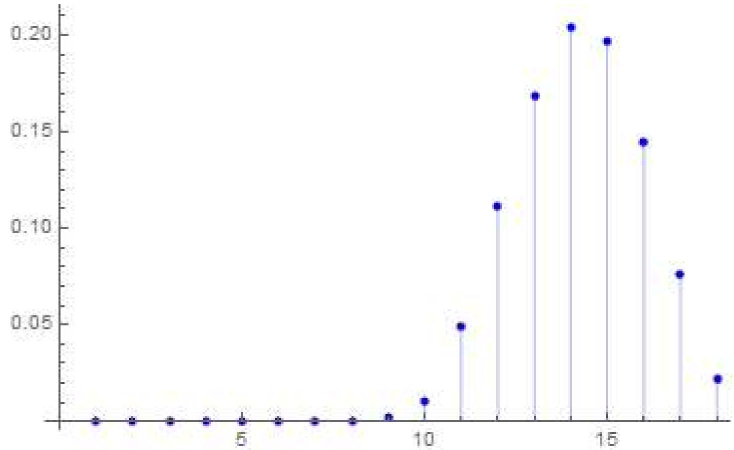
Out[183]: 65

In [187]:
```
counts := Table[count[n], {n, 1, 18}]
```

In [188]:
```
probabilities := N@ counts/1000
```

In [194]:
```
ListPlot[probabilities, Filling-> Axis, PlotStyle -> {Blue}]
```

Out[194]:



In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

Let’s do one more example from Dungeons & Dragons. Suppose I have a box of dicewith the following inventory:5 4-sided dice, 4 6-sided dice, 3 8-sided dice, 2 12-sided dice, 1 20-sided die. I choose a die from the box and roll it. What is the distribution of the outcome?

In [263]:
```
TypesOfDie := {4, 6, 8, 12, 20}
Frequencies := {5, 4, 3, 2, 1}

```

In [265]:
```
TypesOfDie[[3]]
```

Out[265]: 8

In [279]:
```
gen1[p_] := Table[TypesOfDie[[p]], Frequencies[[p]]]
BoxOfDie = Table[gen1[p], {p, 1, 5}] // Flatten
```

Out[280]: {4, 4, 4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 12, 12, 20}

In [276]:
```
PossibleOutcomes = Range[1, 20]
```

Out[276]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}

In [287]:
```
tab1 = Table[Select[BoxOfDie, #>=x&], {x, 1, 20}]
```

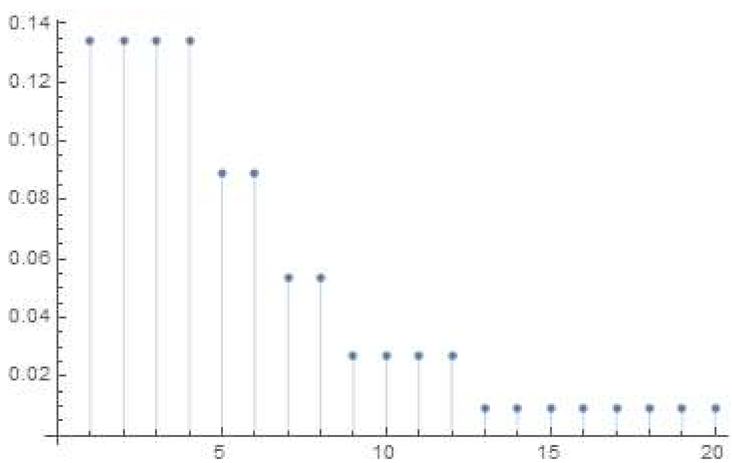
Out[287]:
```
{{4, 4, 4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 12, 12, 20},

>   {4, 4, 4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 12, 12, 20},

>   {4, 4, 4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 12, 12, 20},

>   {4, 4, 4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 12, 12, 20}, {6, 6, 6, 6, 8, 8,
8, 12, 12, 20},

>   {6, 6, 6, 6, 8, 8, 8, 12, 12, 20}, {8, 8, 8, 12, 12, 20}, {8, 8, 8, 1
2, 12, 20},

>   {12, 12, 20}, {12, 12, 20}, {12, 12, 20}, {12, 12, 20}, {20}, {20},
{20}, {20},

>   {20}, {20}, {20}, {20}}
```

In [300]:
```
tab2 = Table[Length[tab1[[x]]], {x, 1, 20}]
```

Out[300]:
```
{15, 15, 15, 15, 10, 10, 6, 6, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1}
```

In [305]:
```
ListPlot[tab2/Total[tab2], Filling -> Axis]
```

Out[305]:



In [ ]:
```

```

In [ ]:
```

```

Chapter 6 Decision Analysis

Price is Right is a television game show played with 2 contestants. The objective of the players is to guess the total price of the items in their bundle. The player who comes closest to guessing the price of his/her bundle without overestimating it wins.

showcases.2011.csv is a file containing the historical prices of the bundles used in the show and the corresponding bids. (Showcase 1 represents the prices of the bundles presented to player 1, Showcase 2 to player 2---the players are presented different bundles, otherwise the player bidding second would always simply bid 0.01 higher.)

In [170]:
```
showcase = Import["showcases.2011.csv"];
```

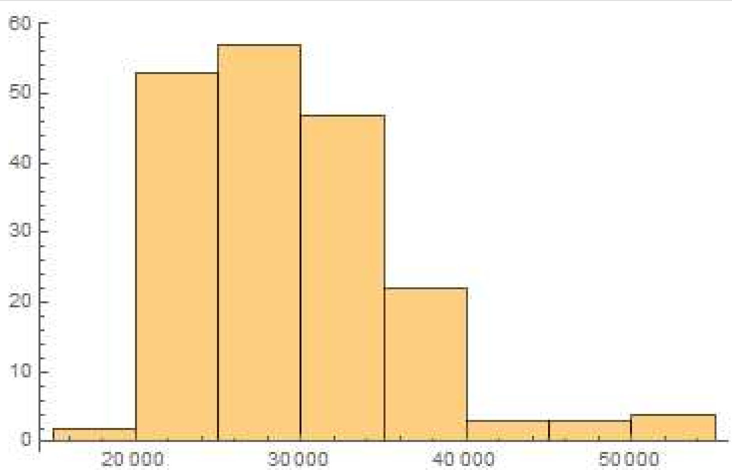
In [80]:
```
showcase[[1]] // Short
showcase[[2]] // Short
showcase[[3]] // Short
showcase[[4]] // Short
showcase[[5]] // Short
showcase[[6]] // Short
showcase[[7]] // Short
showcase[[8]] // Short
showcase[[9]] // Short
showcase[[10]] // Short
showcase[[11]] // Short
```

Out[80]:
```
{, Sep. 19, Sep. 20, Sep. 21, Sep. 22, <<183>>, Jun. 19, Jul. 4, Aug. 17, Sep. 4}
{, 5631K, 5632K, 5633K, 5634K, <<182>>, 6024K, 5685K, 6021K, 6022K, 6023K}
{, , , , , , , , , , , , , <<166>>, , , , , , , , , , , , , }
{Showcase 1, 50969, 21901, 32815, 44432, 24273, <<180>>, 25263, 26993, 29390, 34920, 30323, 46638}
{Showcase 2, 45429, 34061, 53186, 31428, 22320, <<180>>, 32646, 33337, 26314, 31278, 31285, 42319}
{, , , , , , , , , , , , , <<166>>, , , , , , , , , , , , , }
{Bid 1, 42000, 14000, 32000, 27000, 18750, <<180>>, 25500, 41000, 23052, 27800, 27000, 29900}
{Bid 2, 34000, 59900, 45000, 38000, 23000, <<180>>, 26800, 26888, 16000, 30022, 21000, 33000}
{, , , , , , , , , , , , , <<166>>, , , , , , , , , , , , , }
{Difference 1, 8969, 7901, 815, 17432, 5523, 3332, <<179>>, -237, -14007, 6338, 7120, 3323, 16738}
{Difference 2, 11429, -25839, 8186, -6572, -680, <<180>>, 5846, 6449, 10314, 1256, 10285, 9319}
```

In [102]:
```
showcase1 = showcase[[4]];
```

In [113]:
```
Histogram[showcase1]
```

Out[113]:



I collect all the historic prices arrays called showcase1d and showcase2d, use the FindDistribution function to find the closest distribution that the data might have come from, and then plot it.

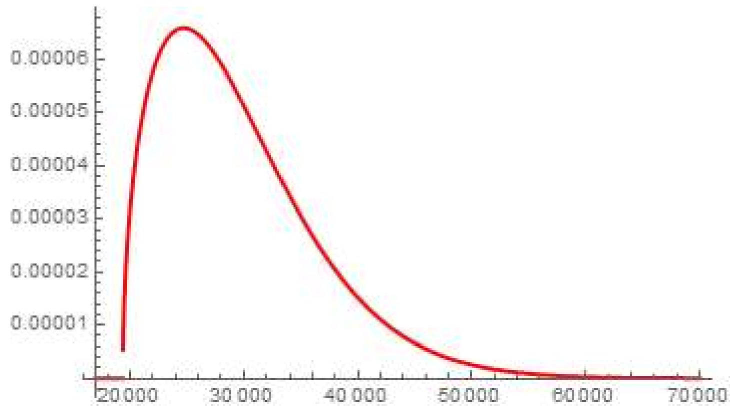
In [174]:
```
showcase1d = showcase1[[2;;]];

FindDistribution[showcase1d]
```

The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.: The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.

Out[175]: WeibullDistribution[1.4908, 11286.8, 19351.1]

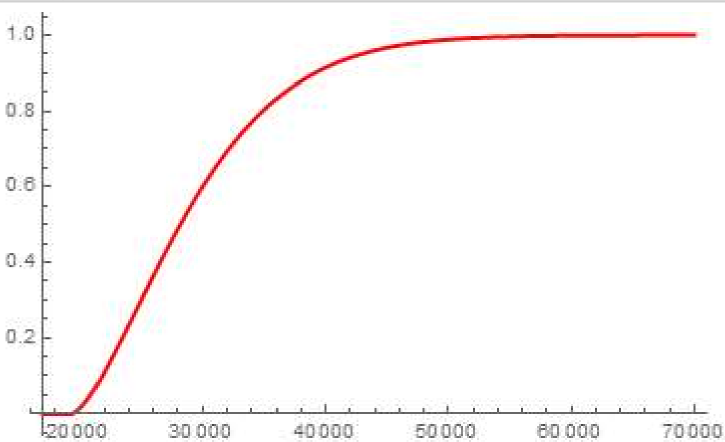
In [186]:
```
pdf1 = Plot[PDF[WeibullDistribution[1.4908, 11286.8, 19351.1], x], {x, 17000
```

Out[186]:



In [188]:
```
cdf1 = Plot[CDF[WeibullDistribution[1.4908, 11286.8, 19351.1], x], {x, 17000
```

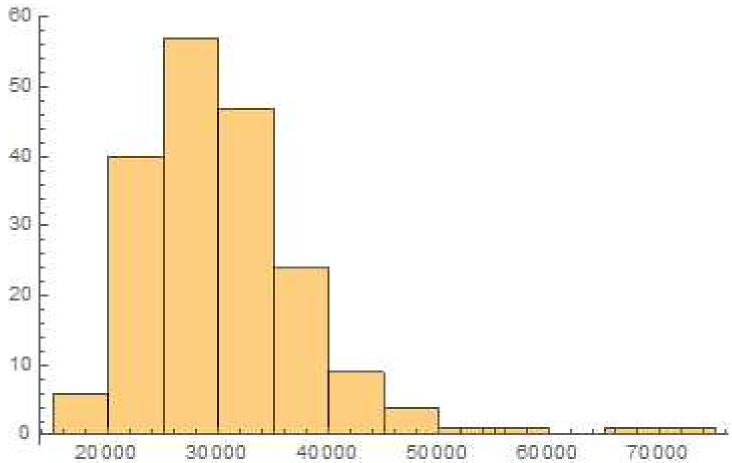
Out[188]:



In [159]:
```
showcase2d = showcase[[5]][[2;;]];
Histogram @ showcase2d
FindDistribution[showcase2d]
```

```
The data will be treated as continuous. Use the option TargetFunctions->Discret
e otherwise.: The data will be treated as continuous. Use the option TargetFunc
tions->Discrete otherwise.
```

Out[160]:

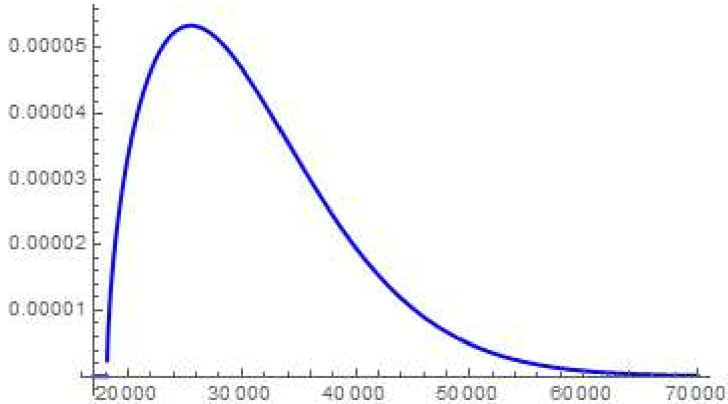


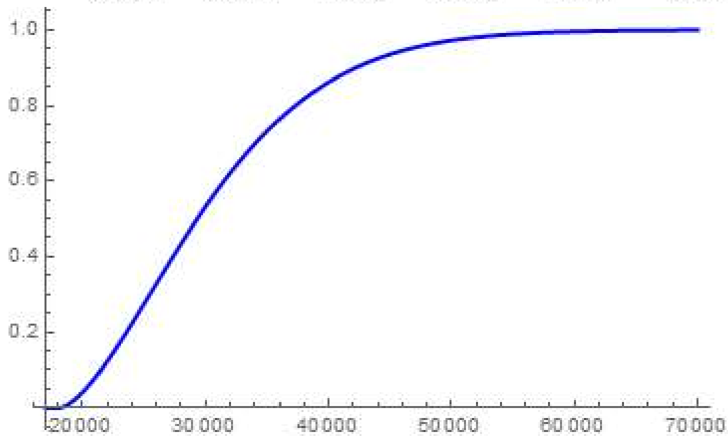
WeibullDistribution[1.56773, 14131.3, 18133.1]

In [189]:

```
pdf2 = Plot[PDF[WeibullDistribution[1.56773, 14131.3, 18133.1], x], {x, 1700
cdf2 = Plot[CDF[WeibullDistribution[1.56773, 14131.3, 18133.1], x], {x, 1700

cdf2d = WeibullDistribution[1.56773, 14131.3, 18133.1]
```

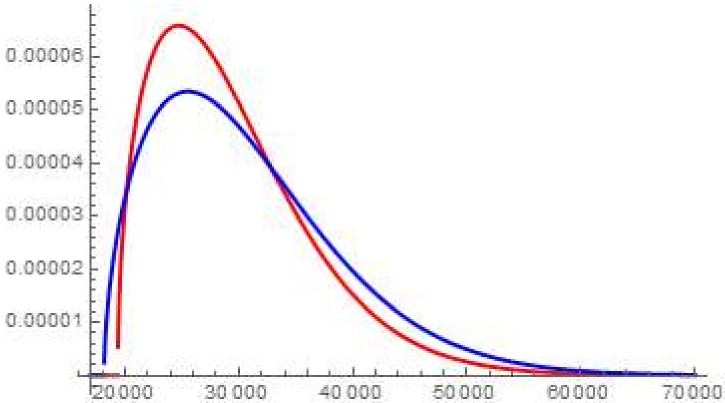
Out[189]:

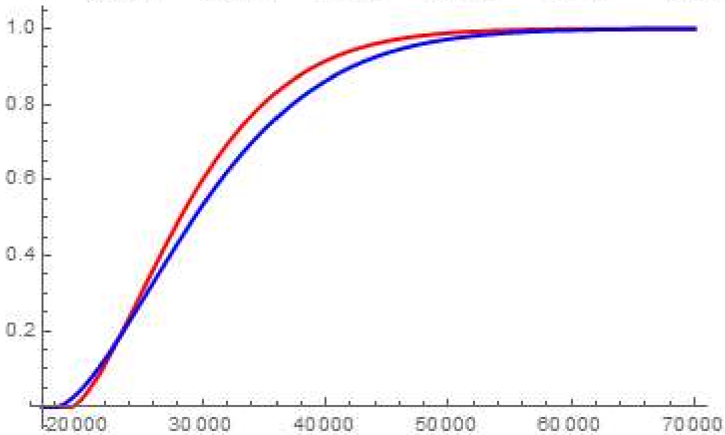




In [192]:
```
Show[pdf1, pdf2]
Show[cdf1, cdf2]
```

Out[192]:

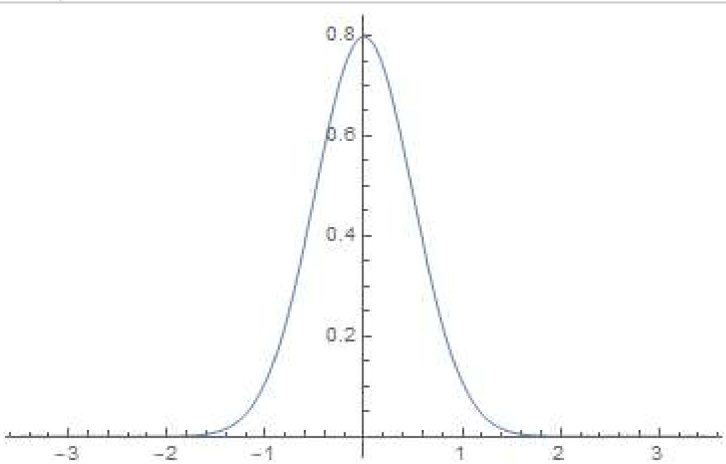




In [ ]:
```

```

In [249]:
```
like = Plot[PDF[NormalDistribution[0, 0.5], x], {x, -3.5, 3.5}]
```

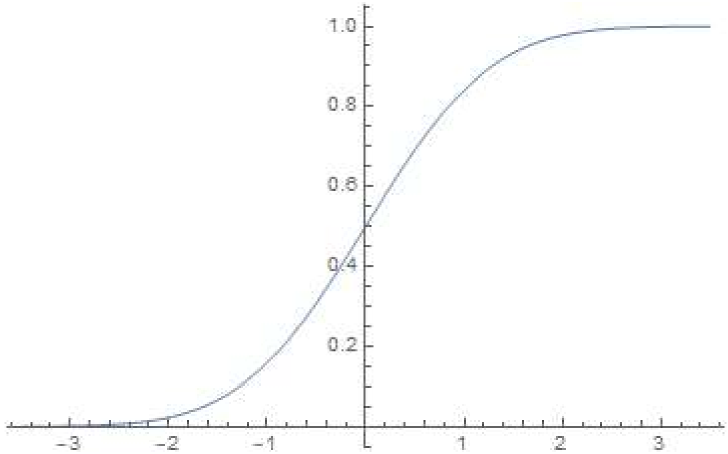
Out[249]:



In [250]:

```
Plot[CDF[NormalDistribution[0, 1], x], {x, -3.5, 3.5}]
```

Out[250]:



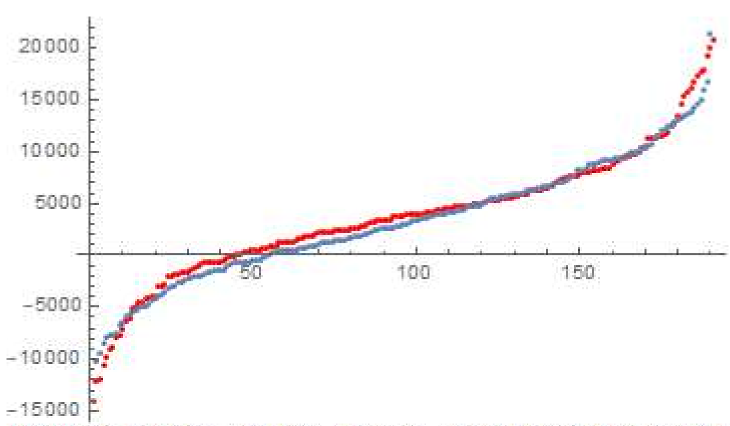
In [243]:

```
diff1 = Sort@ showcase[[10]][[2;;]];
diff2 = Sort@ showcase[[11]][[2;;]];

Show[
ListPlot[diff1, PlotStyle -> {Red}],
ListPlot[diff2]]


FindDistribution[diff1, 3]

FindDistribution[diff2, 3]

```

```
The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.: The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.
```

```
The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.: The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.
```

Out[245]:



```
{MixtureDistribution[{0.434667, 0.565333}, {NormalDistribution[3561.17, 3064.23], NormalDistribution[3870.88, 7949.59]}],
 NormalDistribution[3736.26, 6311.24], LogisticDistribution[3736.26, 3488.62]}
{LogisticDistribution[3439.33, 4203.71], LaplaceDistribution[3439.65, 5239.51], NormalDistribution[3439.65, 7610.08]}
```

In [220]:

```
FindDistribution[diff2]
```

```
The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.: The data will be treated as continuous. Use the option TargetFunctions->Discrete otherwise.
```

Out[220]: LogisticDistribution[3439.33, 4203.71]

prior1 : the result of the FindDistribution function applied to the array of historic prices.

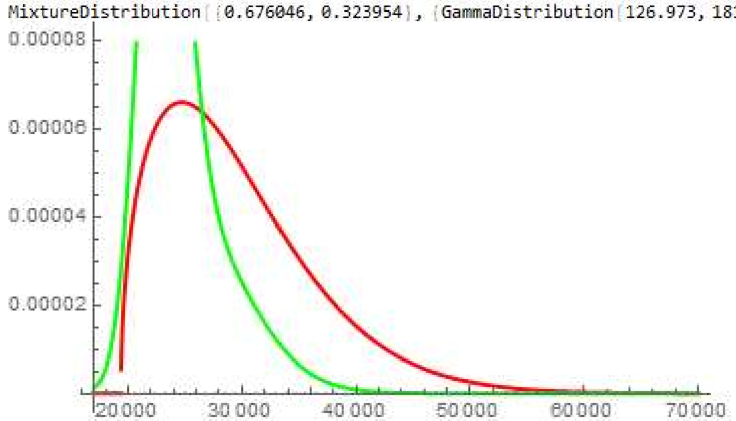
likelihood : how likely is the data (here, the bundle shown) given the player's hypothesis (the amount guessed)? We are forced to make certain assumptions here. I'm assuming the player guesses well enough such that the the likelihood distribution is NormalDistribution[guess, 0.5].

posterior1 : I just average prior1 and likelihood and apply FindDistribution.

In [491]:

```
guess = 20000;

prior1 = WeibullDistribution[1.4908, 11286.8, 19351.1];
prior2 = WeibullDistribution[1.56773, 14131.3, 18133.1];
likelihood = NormalDistribution[guess, 0.5];

posterior1 = FindDistribution[(RandomVariate[prior1, 10000] + RandomVariate[

Show[Plot[PDF[prior1, x], {x, 17000, 70000}, PlotStyle->{Red, Thick}],
     Plot[PDF[posterior1, x], {x, 17000, 70000}, PlotStyle->{Green, Thick}],
     PlotRange -> Full]
```

Out[495]: MixtureDistribution[{0.676046, 0.323954}, {GammaDistribution[126.973, 181.526], LogNormalDistribution[10.2194, 0.147391]}]

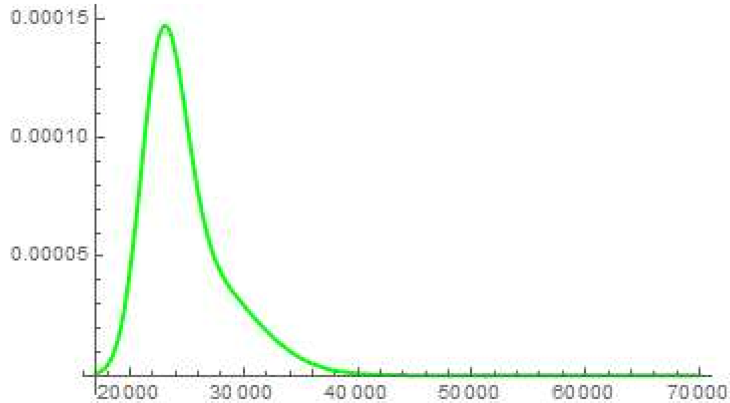


In [ ]:

```

```

In [376]:
```
Plot[PDF[posterior1, x], {x, 17000, 70000}, PlotStyle->{Green, Thick}, PlotR
```

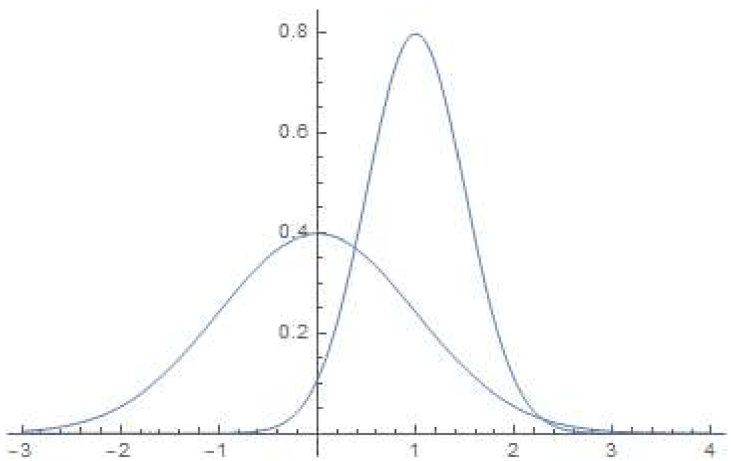
Out[376]:



In [ ]:
```

```

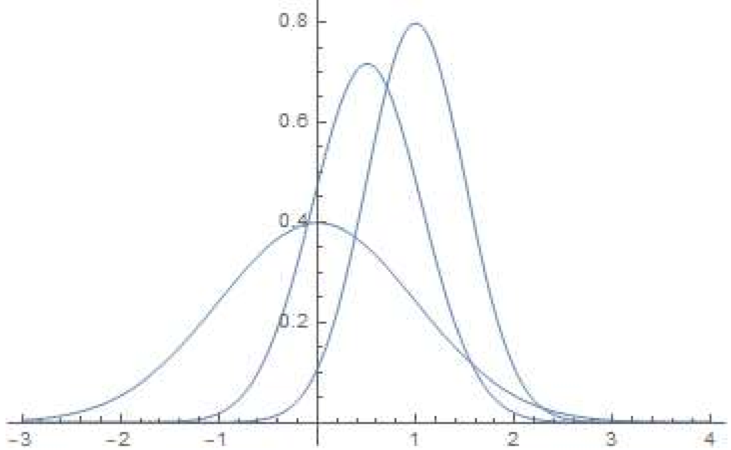
In [337]:
```

Show [Plot[PDF[NormalDistribution[0,1], x], {x, -3, 4}],
      Plot[PDF[NormalDistribution[1, 0.5], x], {x, -3, 4}],
      PlotRange -> All]
```

Out[338]:



In [352]:
```
Post = FindDistribution[(RandomVariate[NormalDistribution[0, 1], 10000] +
    RandomVariate[NormalDistribution[1, 0.5], 10000])/2]

Show [Plot[PDF[NormalDistribution[0,1], x], {x, -3, 4}],
      Plot[PDF[NormalDistribution[1, 0.5], x], {x, -3, 4}],
      Plot[PDF[Post, x], {x, -3, 4}],
      PlotRange -> All]
```

Out[352]: NormalDistribution[0.504871, 0.556821]



In [354]:
```
Post = FindDistribution[(RandomVariate[NormalDistribution[0, 1], 10000] +
    RandomVariate[NormalDistribution[1, 0.5], 10000])/2]


```

Out[354]: NormalDistribution[0.491501, 0.556427]

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [343]:
```
RandomVariate[NormalDistribution[0, 1], 10000] +
    RandomVariate[NormalDistribution[1, 0.5], 10000]]
```

```
ToExpression::sntx: Invalid syntax in or before "RandomVariate[NormalDistributi
on[0, 1], 10000] +    RandomVariate[NormalDistribution[1, 0.5], 10000]] ".

^
```

Out[343]: $Failed

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [330]:
```
mix = CDF[NormalDistribution[]] + CDF[NormalDistribution[]]
```

Out[330]: 2 Function$\left[x, \frac{1}{2} \text{Erfc}\left[-\frac{x}{\sqrt{2}}\right]\right]$

In [331]:
```
Show[Plot[mix, x], {x, -3, 4}]
```

```
Plot::pllim: Range specification x is not of the form {x, xmin, xmax}.

Could not combine the graphics objects in `1`.: Could not combine the graphics
 objects in Show[Plot[mix, x], {x, -3, 4}].
```

Out[331]: Show[Plot[mix, x], {x, -3, 4}]

In [333]:
```
CDF[NormalDistribution[]]
```

Out[333]: Function$\left[x, \frac{1}{2} \text{Erfc}\left[-\frac{x}{\sqrt{2}}\right]\right]$

In [ ]: