In[4]:=
```
(* Adam Beck *)
(* Problem 1*)


(* hosp-heart.nb data *)

(* {M,V} M = one year mortality rate,
percentage of patiuents that died within one year of the
 transplant operation,
V = average annual number of transplants at that center during the same 4 years *)
heart = {{17.9, 27}, {23.1, 4}, {40, 3}, {6.5, 35}, {14.9, 17}, {12.5, 4}, {15.7, 45},
   {9.8, 28}, {24, 6}, {5.0, 10}, {15.4, 13}, {4.8, 7}, {0, 1}, {19.1, 47}, {4.5, 6},
   {15, 56}, {12.5, 4}, {33.9, 8}, {10.7, 9}, {13, 14}, {28.3, 12}, {57.1, 2}, {6.3, 4},
   {10, 3}, {8.3, 12}, {17.5, 10}, {20, 3}, {29.3, 10}, {21.4, 7}, {27.3, 8}, {13.6, 6},
   {21.8, 30}, {36.4, 3}, {18.2, 11}, {33.3, 2}, {20, 4}, {38.5, 7}, {20.8, 18}, {12.2, 19},
   {22.2, 18}, {29, 8}, {0, 9}, {5.7, 9}, {50, 2}, {21.7, 15}, {66.7, 4}, {29.4, 17},
   {12.1, 27}, {10.7, 14}, {6.3, 4}, {16.2, 9}, {21.1, 5}, {17.4, 33}, {23.9, 17},
   {42.9, 2}, {40, 2}, {6.7, 15}, {44.4, 3}, {18.7, 34}, {14.7, 24}, {7.4, 7}, {12.6, 24},
   {9.7, 26}, {44.4, 2}, {16.7, 6}, {15.8, 14}, {83.3, 2}, {10.9, 22}, {13.3, 5},
   {11.1, 5}, {75, 2}, {19, 20}, {14, 13}, {60, 1}, {21.2, 8}, {9.7, 8}, {50, 2}, {25, 14},
   {18.6, 15}, {0.0, 1}, {35.3, 9}, {23.5, 85}, {15.6, 11}, {37.5, 2}, {14.3, 28},
   {14.3, 4}, {16.7, 6}, {20.0, 15}, {13.0, 17}, {9.6, 26}, {66.7, 3}, {30.8, 3},
   {14.0, 13}, {27.5, 10}, {37.5, 8}, {18.9, 13}, {0.0, 4}, {12.2, 44}, {57.1, 4},
   {21.4, 35}, {23.4, 16}, {10.9, 12}, {15.6, 8}, {16.7, 2}, {13.9, 9}, {18.2, 11},
   {11.5, 26}, {18.4, 13}, {16.7, 3}, {20.4, 14}, {40.0, 5}, {20.7, 56}, {19.6, 13},
   {13.5, 9}, {29.9, 36}, {8.4, 21}, {28.4, 24}, {7.7, 23}, {19.3, 29}, {0.0, 1},
   {22.2, 20}, {30.0, 5}, {7.0, 11}, {23.8, 7}, {18.8, 29}, {14.5, 16}, {17.0, 16},
   {20.0, 15}, {6.7, 15}, {11.4, 20}, {100.0, 1}, {31.4, 9}, {17.6, 26}, {19.6, 14}};
```

In[5]:=
```
(* Split this M and V data into separate
 lists via Transpose[] in order to parse through *)
heartTranspose = Transpose[heart];
MData = heartTranspose[[1]];
VData = heartTranspose[[2]];
```

In[8]:=
```
(* Define mean, median, quantile, and variance functions *)
```

In[9]:=
```
mean[x_] := Sum[x[[i]], {i, 1, Length[x]}] / Length[x];
(* Sum elements, divide by length *)
```

In[10]:=
```
median[x_] := (s = Sort[x]; s[[IntegerPart[.5 * Length[s]]]]);
(* Sort list, take element at index 1/2*length *)
```

In[11]:=
```
quantile[x_, alpha_] := (s = Sort[x]; s[[IntegerPart[alpha * Length[s]]]])
 (* Sort list, take element at index .alpha*length *)
```

In[12]:=
```
variance[x_] := (m = mean[x]; Sum[(x[[i]] - m)^2, {i, 1, Length[x]}] / Length[x]);
(* difference of every element from mean, squared, times 1/length *)
```

In[13]:=
```
(* Find the mean, median, q1 and q3, and variance *)
hospMeanM = mean[MData]
```

Out[13]= 21.9045

In[14]:=
```
hospMeanV = N[mean[VData]]
```
Out[14]= 13.8657

In[15]:=
```
hospMedianM = median[MData]
```
Out[15]= 18.2

In[16]:=
```
hospMedianV = median[VData]
```
Out[16]= 10

In[17]:=
```
hospQ1M = quantile[MData, .25]
```
Out[17]= 12.2

In[18]:=
```
hospQ1V = quantile[VData, .25]
```
Out[18]= 4

In[72]:=
```
hospQ3M = quantile[MData, .75]
```
Out[72]= 25

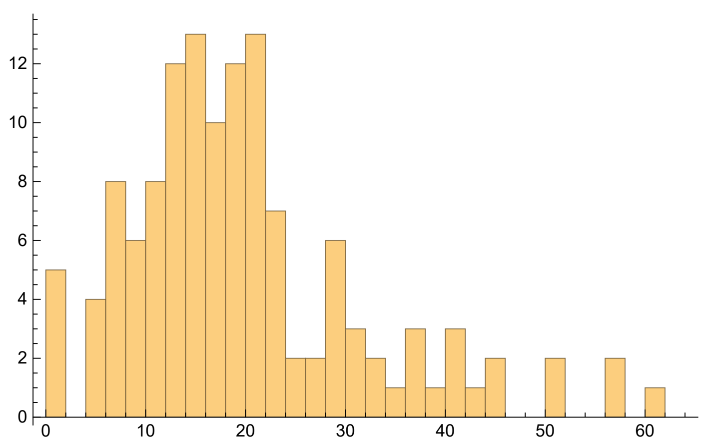
In[73]:=
```
hospQ3V = quantile[VData, .75]
```
Out[73]= 17

In[21]:=
```
hospVarianceM = variance[MData]
```
Out[21]= 268.634

In[22]:=
```
hospVarianceV = N[variance[VData]]
```
Out[22]= 166.46

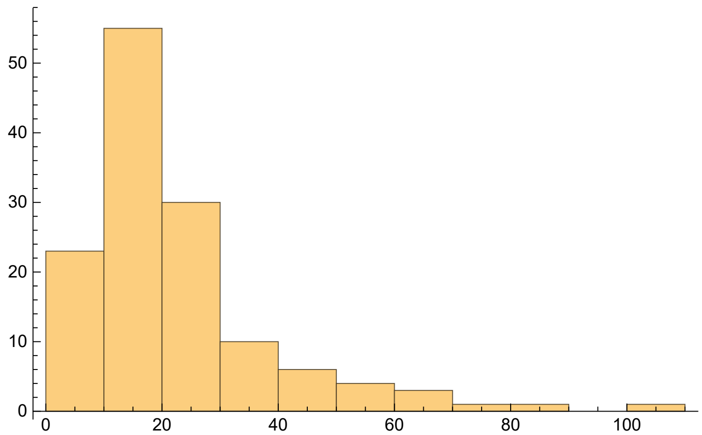
```
(* Histograms using two difference bin sizes *)
(* I will use a bin size Length/2 for a very large bin count,
and Length/10 for a smaller bin count *)

(* MData large bin count *)
Histogram[MData, IntegerPart[Length[MData] / 2]]
```

Out[23]=



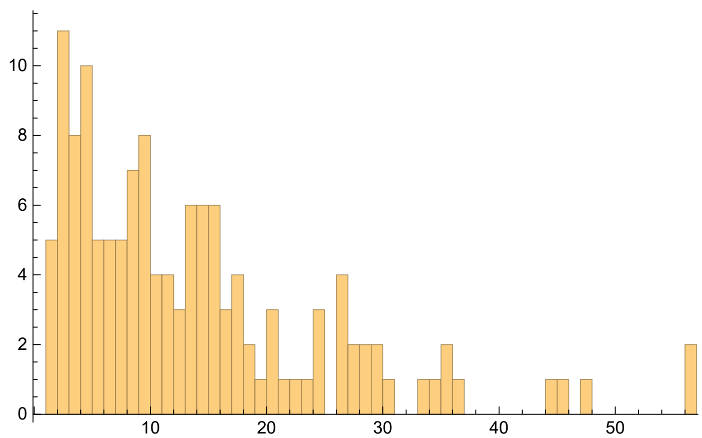
```
(* MData small bin count *)
Histogram[MData, IntegerPart[Length[MData] / 10]]
```

Out[24]=



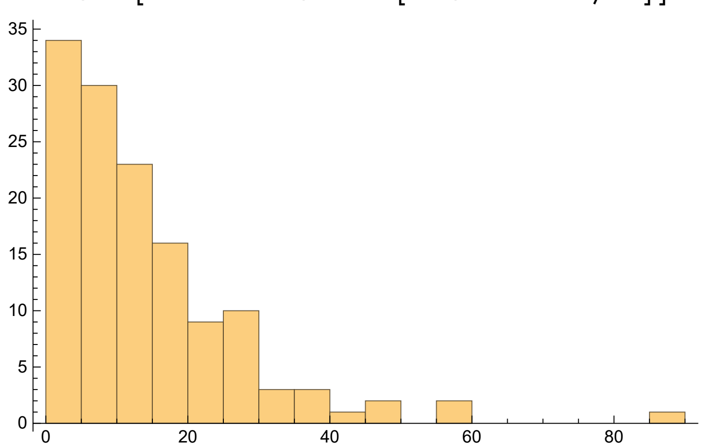

```
(* VData, large bin count *)
Histogram[VData, IntegerPart[Length[VData] / 2]]
```

Out[25]=



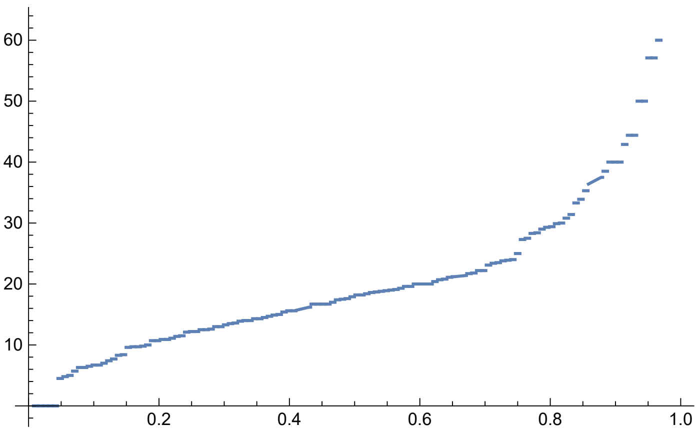

```
(* VData, small bin count*)
Histogram[VData, IntegerPart[Length[VData] / 10]]
```

Out[26]=



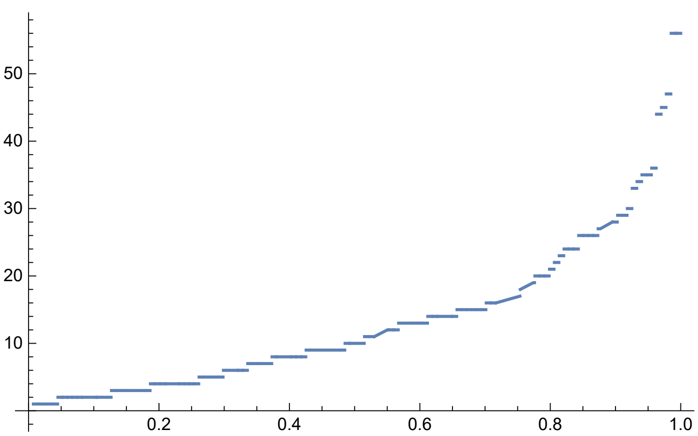
```
(* Produce plots of quantile functions, moment functions, and CDFs *)

(* Define functions for moments and CDF *)

(* Sum elements raised to the kth power, divide by length *)
moment[x_, k_] := N[Sum[x[[i]]^k, {i, 1, Length[x]}] / Length[x]];

In[28]:= cdf[x_, xi_] := N[Sum[If[x[[i]] <= xi, 1, 0], {i, 1, Length[x]}]];
(* Count that an element is less than or equal to a given element *)

In[29]:= (* Plot the quantile functions *)
(* MData *)
Plot[quantile[MData, i], {i, 0, 1}]
```

Out[29]=




```
In[30]:= (* VData *)
Plot[quantile[VData, i], {i, 0, 1}]
```

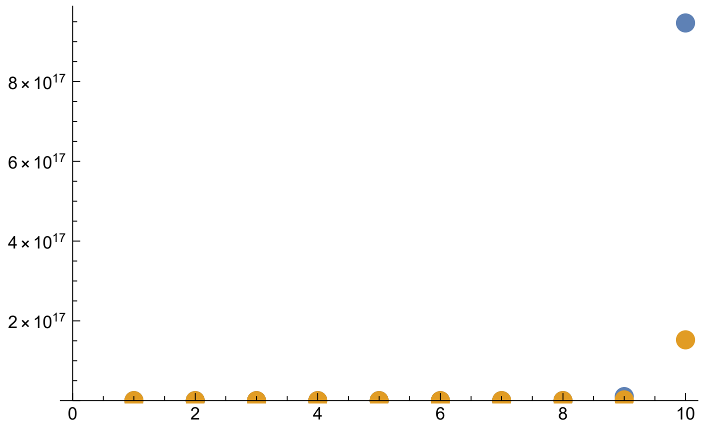
Out[30]=




```
In[31]:= (* Plot moment functions *)
(* Get the first 10 moments for MData and VData in a Table *)
momMData  = Table[moment[MData, i], {i, 1, 10}];

In[32]:= momVData = Table[moment[VData, i], {i, 1, 10}];
```

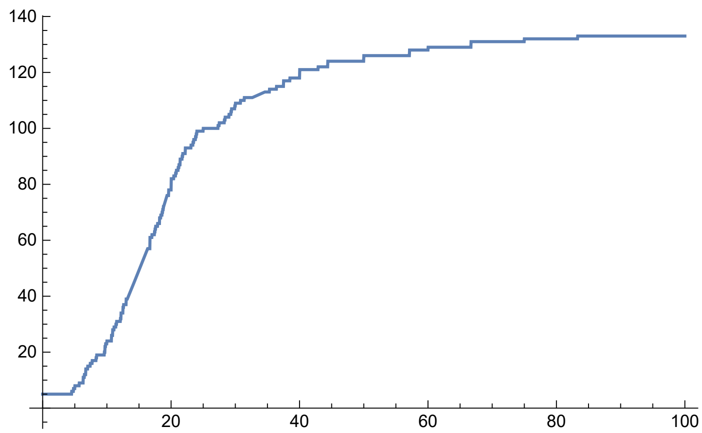
In[33]:=
```
(* Plot the two moments *)
ListPlot[{momMData, momVData}, PlotStyle → PointSize[.03], PlotRange → {0, 9.9 * 10^17}]
(* Blue is MData, Orange is VData*)
```

Out[33]=



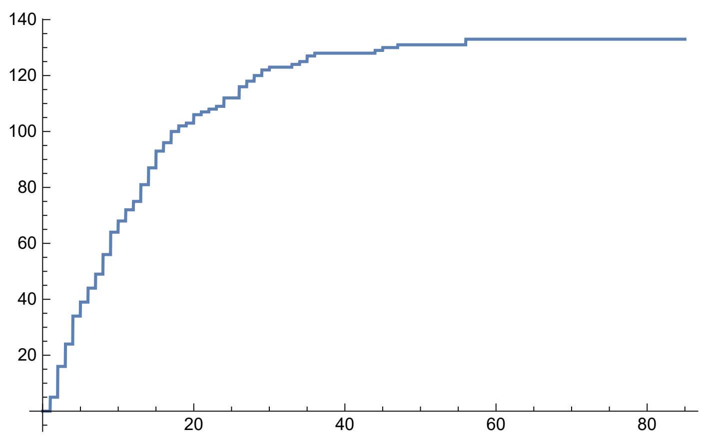
In[34]:=
```
(* Plot CDF functions *)
(* MData *)
Plot[cdf[MData, i], {i, 0, Max[MData]}]
```

Out[34]=



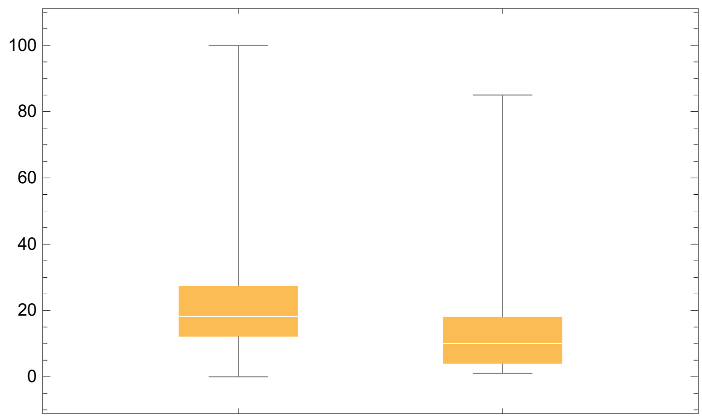
In[35]:=
```
(* VData *)
Plot[cdf[VData, i], {i, 0, Max[VData]}]
```

Out[35]=



```
(* Although the instructions do not say to compare
 any box and whisker and QQ plots for this hospital data
 against other sets of data, I will product them anyways. QQ
 will be MData(x axis) against VDaya (y axis) *)

(* Box an whisker plots *)
(* A box and whisker plot takes a min, q1, q2 (median), q3, and max *)

(* Box and whisker, MData and VData, outliers not shown *)
BoxWhiskerChart[{MData, VData}]
```

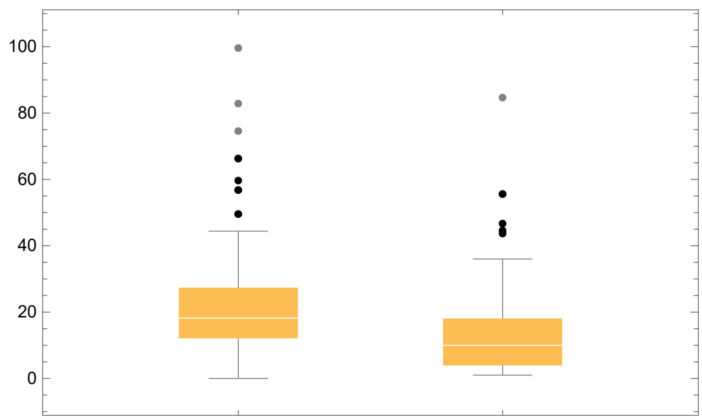
Out[46]=



In[47]:=

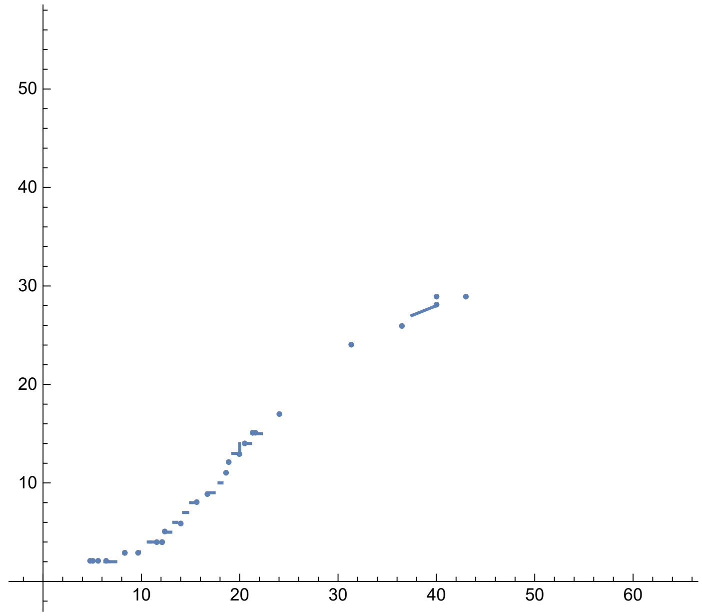
```
(* Box and whisker, MData and VData, outliers shown *)
BoxWhiskerChart[{MData, VData}, "Outliers"]
```

Out[47]=



In[54]:=
```
(* Parametric Plots (QQ Plots), MData on x axis, VData on y axis *)
ParametricPlot[{quantile[MData, i], quantile[VData, i]}, {i, 0, 1}]
```

Out[54]=



In[63]:=
```
(* If the two sets come from a population with the same distribution,
the points should fall approximately
 along a 45 degree reference line. As we can see,
the 2 batches do not appear to have come from populations with
 a common distribution, as they do not fit along a 45 degree line *)

(* resistor.nb data *)
(* The data represents a listing of the resistances
 (in ohms) of 200 resistors which are all rated at 10 kiloohms. *)
resistors = {9.97910927, 9.833997401, 10.48797923, 9.778286587, 10.4127049, 9.729651074,
   10.34005333, 9.894176108, 10.07983211, 9.933230947, 9.977783398, 10.13141411,
   10.1266421, 9.37852757, 10.26785423, 9.907086669, 9.744503691, 9.971603949,
   9.693939764, 9.620137112, 12.28072506, 10.0580338, 10.33764317, 9.757096213,
   9.593230848, 9.713741738, 9.432574293, 9.62099431, 9.802732952, 9.971484578,
   10.22548428, 10.3352728, 9.989841592, 10.29860424, 9.52298034, 10.08499861,
   9.394148142, 9.944944954, 10.21438162, 10.36193691, 10.02987499, 9.603449021,
   9.742946181, 9.875414084, 10.05078967, 10.12314509, 10.15281111, 5.870566193,
   9.484863417, 9.973958404, 9.94911044, 9.374762262, 9.788310356, 10.06500849,
   9.77439594, 10.03864565, 10.32397119, 9.916142963, 9.967350072, 10.09860352,
   9.987682395, 10.15563395, 9.537918791, 9.945042157, 10.02686399, 9.74540807,
   10.26915708, 9.696347652, 10.13930795, 9.51572712, 9.367227099, 9.831637831,
   10.1807235, 9.88921993, 9.923452458, 9.944225885, 9.779727284, 10.26538836, 10.2298635,
   10.2461264, 9.694717951, 9.771545526, 9.679096242, 10.15118993, 10.25894345,
   9.613968464, 10.14607857, 10.3809408, 10.00425765, 10.30422606, 9.938641588,
   10.14989447, 9.62901378, 6.613345698, 10.48706974, 10.10426569, 10.15476425,
   9.839152246, 9.74229305, 9.712882265, 10.09355753, 9.655283966, 10.01073951,
   10.23032052, 9.896222755, 9.646005983, 10.22741355, 9.916736976, 9.853518852,
   9.797304974, 9.542975581, 9.582644329, 10.06420074, 10.1110437, 9.09833499,
   9.694181349, 10.0837185, 9.990310834, 9.680224016, 9.544769559, 10.12220661,
   10.35625939, 9.68922915, 9.816272486, 9.838797828, 9.787675983, 10.01512384,
   9.672549018, 9.166747182, 9.839861368, 10.0490497, 9.9589975, 9.707653239, 9.642065029,
   10.14670044, 9.704657023, 9.851454583, 9.92931813, 10.05903936, 9.749898131,
   10.12904658, 9.776733909, 9.956306817, 10.10913774, 9.25291271, 9.823820724,
   9.581313056, 9.84027462, 9.738894951, 9.923279654, 9.815685862, 9.754906605,
   10.19531748, 9.718578829, 9.830784043, 9.860661512, 9.665515781, 9.956836598,
   10.06308718, 9.401201273, 10.10992616, 9.738494773, 9.991823154, 9.877411846,
   10.23755441, 10.04556889, 9.978626954, 10.06519891, 9.774786454, 10.26202664,
   10.10298671, 9.558598995, 9.352852535, 9.611078544, 9.807194024, 9.684415081,
   10.17326848, 9.683191811, 10.03918111, 9.891267714, 9.707087079, 9.68933829,
   10.10867702, 9.770431111, 9.697278747, 10.15024178, 10.17638293, 9.676198933,
   9.765484028, 9.952918381, 10.15444308, 10.03372073, 9.607316647, 9.856609145,
   9.805244863, 9.728007162, 9.951510938, 10.03217857, 10.19504918, 10.23059564};
```

In[70]:=
```
(* Find the mean, median, q1 and q3, and variance *)
resistorMean = mean[resistors]
```

Out[70]= 9.87989

In[71]:=
```
resistorMedian = median[resistors]
```

Out[71]= 9.91674

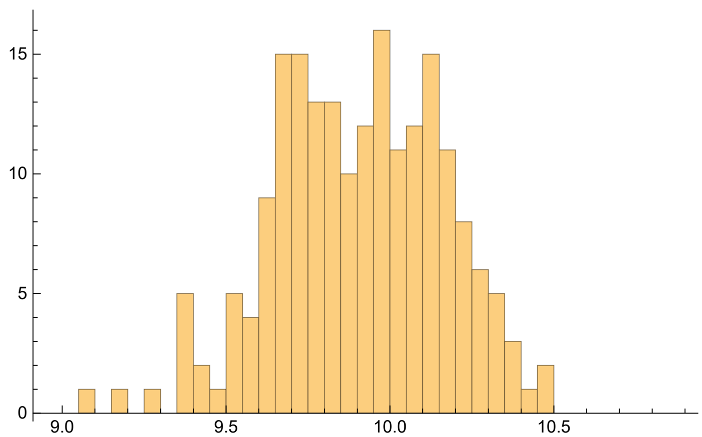
In[74]:=
```
resistorQ1 = quantile[resistors, .25]
```
Out[74]=
```
9.71288
```

In[75]:=
```
resistorQ3 = quantile[resistors, .75]
```
Out[75]=
```
10.1043
```

In[79]:=
```
resistorVariance = variance[resistors]
```
Out[79]=
```
0.229448
```

In[80]:=
```
(* Histograms using two difference bin sizes *)
(* I will use a bin size Length/2 for a very large bin count,
and Length/10 for a smaller bin count*)

(* Large bin count*)
Histogram[resistors, IntegerPart[Length[resistors] / 2]]
```

Out[80]=



In[81]:=
```
(* Small bin count *)
Histogram[resistors, IntegerPart[Length[resistors] / 10]]
```

Out[81]=