

HW 2 - ASTR503

Daniel George - dgeorge5@illinois.edu

Q1)

Calculations

Generating data

```
In[1011]:= Needs["ErrorBarPlots`"];  
nQM = N@QuantityMagnitude;  
  
dx = .5; (*Bin width*)  
nP = 10^6; (*Number of samples*)  
A = dx nP; (*Area under histogram*)  
  
probF = NormalDistribution[10, 2];  
(*Gaussian distribution*)  
  
dataH = RandomVariate[probF, nP];  
(*Random sample from Gaussian*)
```

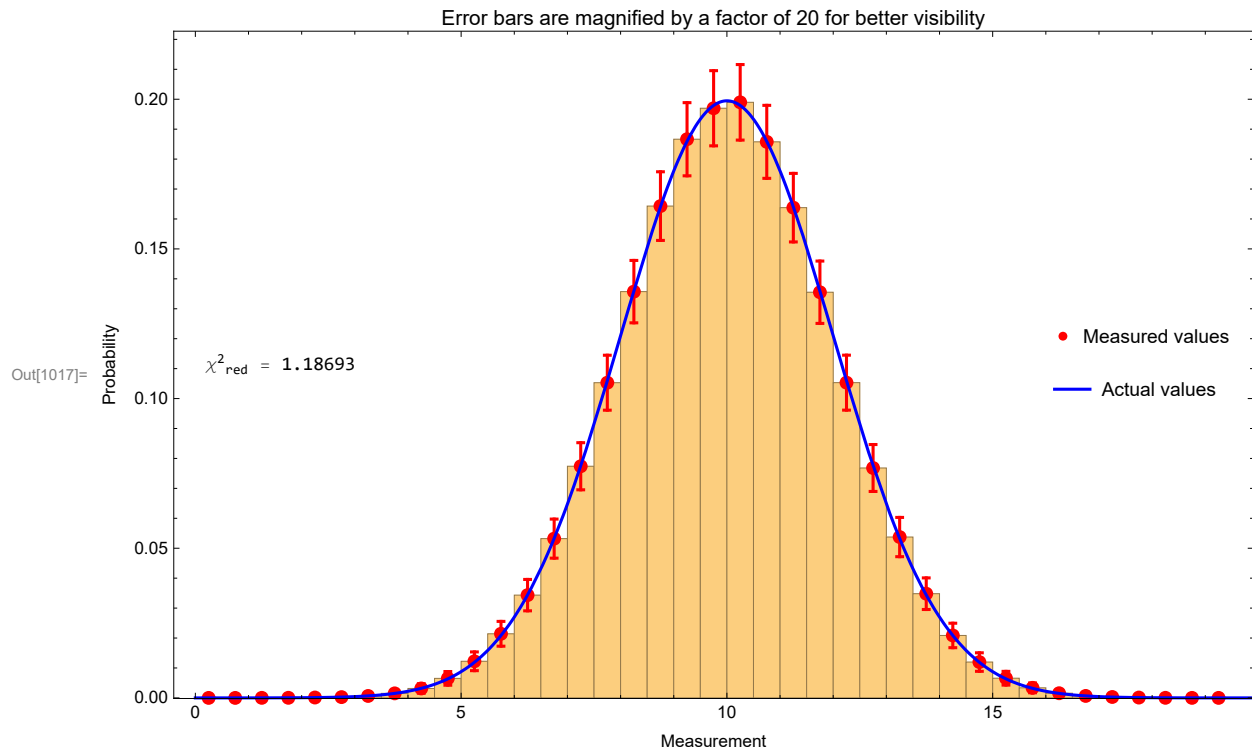
Bin counts and errors

```
In[1015]:= binData = Transpose[  
  {Transpose[{MovingAverage[#[[1]], 2], #[[2]]/A}], ErrorBar /@ (Sqrt[#[[2]]]/A)} &@  
  HistogramList[dataH, {dx}]];
```

Reduced χ^2 for 10 bins near the peak

```
In[1016]:= chi2R =  
  Total[(PDF[probF]@binData[[#, 1, 1]] - binData[[#, 1, 2]])^2 / binData[[#, 2, 1]]^2] /  
  10 & [Span@@ (Ceiling[Length@binData / 2 + #] & /@ {-4, 5})]  
Out[1016]:= 1.186933321493
```

Plots



Q2)

a)

Uniformly distributing points inside sphere

```
In[891]:= nS = 5 × 10^3; (*Number of supernovae*)
dists = 2000 Norm /@ RandomPoint[Ball[], nS];
```

Mean distance

```
In[892]:= Mean@dists Mpc
```

```
Out[892]= 1499.527337911 Mpc
```

b)

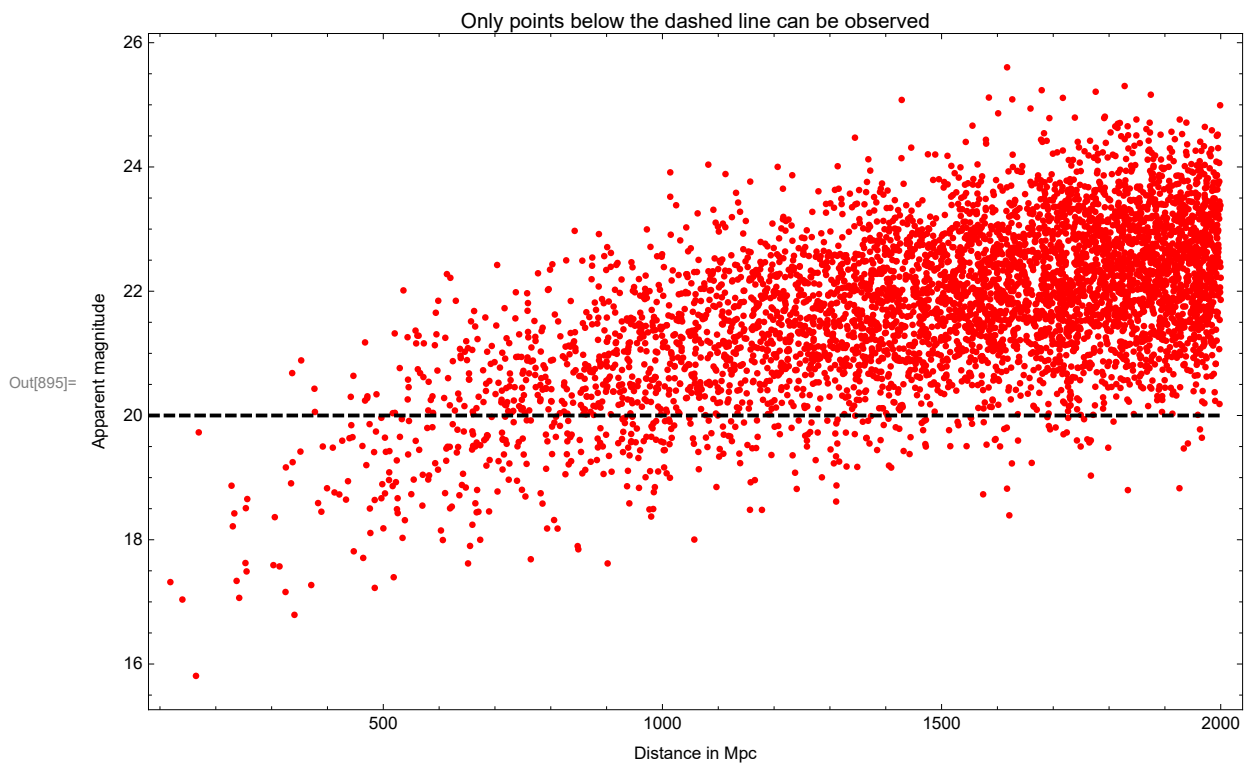
Randomly sampling absolute magnitudes

```
In[893]:= MList = -19 + RandomVariate[NormalDistribution[], nS];
```

Finding apparent magnitudes

```
In[894]:= mList = MList + 5 Log10[dists 10^6 / 10];
```

Plot of apparent magnitude vs distance



Selecting observable supernovae

```
In[896]:= iObs = Flatten@Position[mList, m_ /; m ≤ 20];
(*Positions in the list for observed supernovae*)
```

Average apparent magnitudes

-- For all samples

```
In[897]:= Mean@mList
```

```
Out[897]= 21.787430252207
```

-- For only observed ones

```
In[898]:= Mean[mList[[iObs]]]
```

```
Out[898]= 19.249944461711
```

c)

Assigning velocities

```
In[899]:= H0 = 72.; (*Hubble constant*)
          vels = H0 * dists; (*True velocities*)
```

Estimated distances for all supernovae

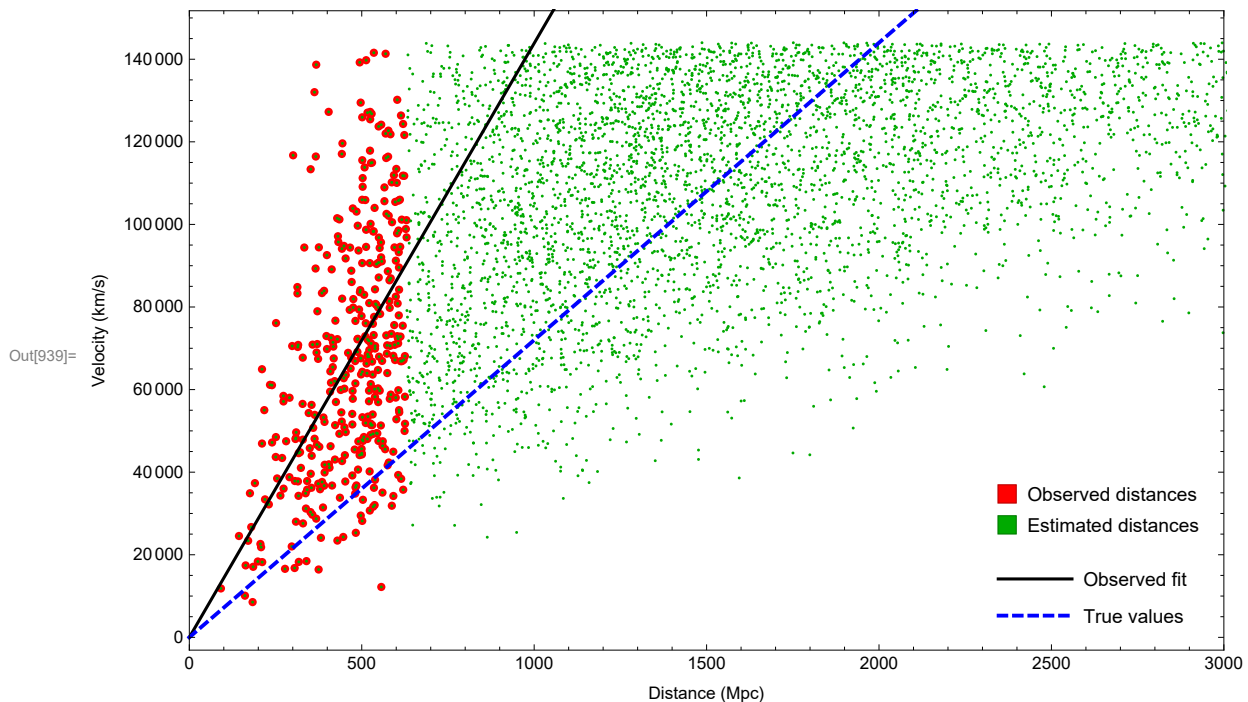
```
In[901]:= eDists = 10^(1 + (mList + 19) / 5) * 10^-6;
```

Fit through origin for velocity vs distance

```
In[902]:= fitH = LinearModelFit[Transpose[{eDists[[iObs]], vels[[iObs]]},
          {r}], r, IncludeConstantBasis -> False]
```

```
Out[902]= FittedModel[143.85744575094 r]
```

Velocity vs distance plots



We can see that the observed distances are clearly biased. We consistently estimate the distances to be less than the true values. This is because of two reasons:

1. Although the absolute magnitude (M) is biased equally in both directions, the estimated distance is proportional to 10^M . This causes the estimated distances to be skewed in one direction.
2. We are unable to observe the supernovae whose magnitudes are shifted to values higher than 20 due to the observing limit. This means we will see more supernovae with absolute magnitudes lower than -19, thus causing us to estimate lower distances.

d)

Estimated value of Hubble constant

```
In[904]:= "H0" → fitH["BestFitParameters"][[1]] "km/s/Mpc"
Out[904]= H0 → 143.85744575094 km/s/Mpc
```

The effect of this bias is that it causes us to overestimate the value of the Hubble constant.

Correcting distance estimates

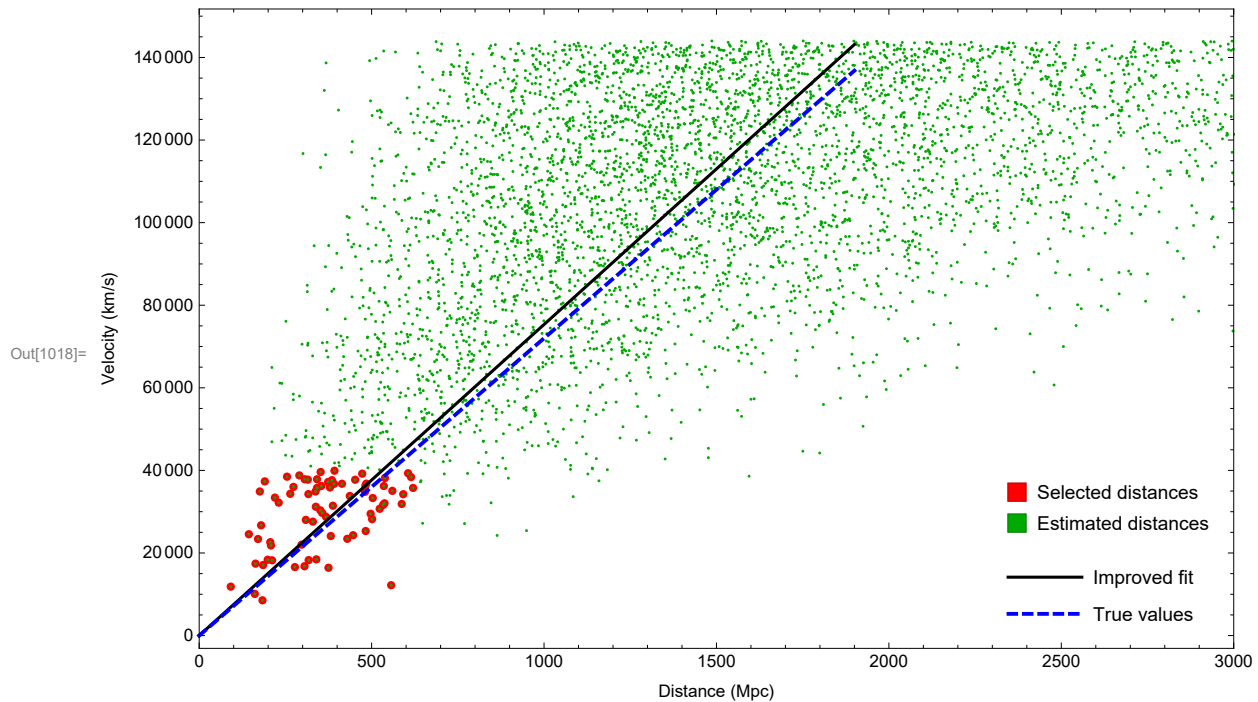
Since we know that for each velocity there is an equal number of distances above and below the true value, we should use the median of bin counts of distances for each bin of velocity in order to estimate distances.

We should use the above formula to estimate the distance given apparent magnitude. This can be further corrected by first truncating observed samples with high velocities to correct for the finite observing distances.

Fit with corrected distances

```
In[940]:= fitC = LinearModelFit[Select[Transpose[{eDists[[iObs]], vels[[iObs]]}], #[[2]] < 40000 &],
{r}, r, IncludeConstantBasis → False]
Out[940]= FittedModel[75.333608994341 r]
```

Plot and fit with reduced sample



Corrected value of H0:

```
In[935]:= "Corrected H0" → fitC["BestFitParameters"][[1]] "km/s/Mpc"
```

```
Out[935]= Corrected H0 → 75.333608994341 km/s/Mpc
```

Q3)

Defining constants

```
In[908]:= k = nQM[k, "J/K"]; h = nQM[h, "J/Hz"];
c = nQM[c, "m/s"];
```

a)

Defining specific intensity function

```
In[910]:= i[v_, T_] := 2 h / c^2 v^3 / (Exp[h v / (k T)] - 1)
```

Energy density of CMB over the band

```
In[911]:= uCMB1 =
  UnitConvert[4 Pi / c NIntegrate[i[v, 2.73], {v, 824 × 10^6, 894 × 10^6}] J/m^3, ergs/cm^3 ]

Out[911]= 1.8032721778545 × 10^-20 ergs/cm^3
```

b)

Photon number density formula

$$\frac{4 \pi \int_{\nu_{\min}}^{\nu_{\max}} \frac{I(\nu, T)}{h \nu} d\nu}{c}$$

Number density over all frequencies

For any CMB temperature

```
In[912]:= 4 Pi / c Integrate[i[v, T_CMB] / (h v), {v, 0, Infinity}, Assumptions → T_CMB > 0]
  10^-6 / (cm^3 * K^3)

Out[912]= (20.286825885333 / (cm^3 K^3)) T_CMB^3
```

For given temperature (2.73K)

```
In[913]:= UnitConvert[4 Pi / c Integrate[i[v, 2.73] / (h v), {v, 0, Infinity}] per meter^3, /cm^3 ]

Out[913]= 412.76421906938 /cm^3
```

Number density at T = 2.73K in the given band

```
In[914]:= UnitConvert[
  4 Pi / c NIntegrate[i[v, 2.73] / (h v), {v, 824 × 10^6, 894 × 10^6}] per meter^3, /cm^3 ]

Out[914]= 0.0031664573814847 /cm^3
```

c)

Flux at 1km

Flux is power per unit area. Power is 3 MW.

```
In[915]:= F1 = 3. MW / (4 Pi (1 km)^2);
```

Energy density at 1km

Energy density is flux divided by speed of light.

```
In[916]:= uCP = UnitConvert[F1 / c, "erg/cm^3"]
```

```
Out[916]= 7.9632561883142 × 10-9 ergs/cm3
```

Ratio to CMB energy density

```
In[917]:= uCP / uCMB1
```

```
Out[917]= 4.4160034664256 × 1011
```

d)

Energy density of CMB in new band

```
In[918]:= uCMB2 = UnitConvert[
  4 Pi / c NIntegrate[i[v, 2.73], {v, 161.0 × 10^9, 161.07 × 10^9}] J/m^3, ergs/cm^3 ]
```

```
Out[918]= 1.1319367730945 × 10-16 ergs/cm3
```

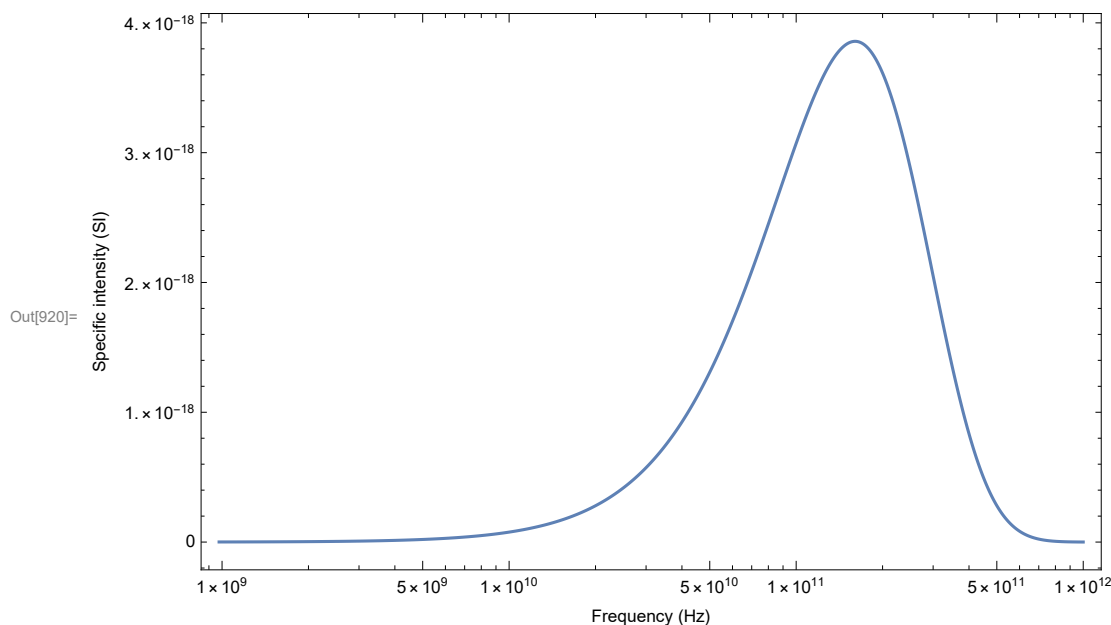
Ratio to new CMB energy density

```
In[919]:= uCP / uCMB2
```

```
Out[919]= 7.0350715495742 × 107
```

Comparison to previous ratio

This value is smaller than the previous ratio because there is more energy density from the CMB in the second band than the first. This is because the peak of the specific intensity is at about 4000 GHz for $T = 2.73\text{K}$ as shown in the plot below:



This peak value can also be obtained using the Wien's displacement law:

```
In[921]:= UnitConvert[FormulaData[{"WienDisplacementLaw", "Frequency"}, {T -> 2.73 K}][[2]], "Hz"]
```

```
Out[921]= 1.6049462005015 x 10^11 Hz
```

Q4)

2)

Function to solve for wavelength given energy

```
In[922]:= fλ[En_] := NSolve[En == h c / λ, λ][[1, 1]]
```

a) Ionization energy of Hydrogen

```
In[923]:= fλ[13.6 eV]
```

```
Out[923]= λ -> 9.116485102677 x 10^-8 m
```

b) Dissociation energy of Hydrogen

```
In[924]:= fλ[4.48 eV]
```

```
Out[924]= λ -> 2.7675044061698 x 10^-7 m
```

c) Dissociation energy of CO₂

In[925]:= $f_{\lambda}[11.02 \text{ eV}]$

Out[925]= $\lambda \rightarrow 1.1250834609474 \times 10^{-7} \text{ m}$

4)

Finding f_{λ} / f_{ν}

Since total energy is conserved:

$$f_{\lambda} d\lambda = -f_{\nu} d\nu$$

Solving for f_{λ} / f_{ν} :

In[926]:= $f_{\lambda} / f_{\nu} == \text{With}[\{\nu = c / \lambda\}, -\text{Dt}[\nu] / \text{Dt}[\lambda]] \text{ m/s} // \text{TraditionalForm}$

Out[926]//TraditionalForm=

$$\frac{f_{\lambda}}{f_{\nu}} = \frac{2.99792458 \times 10^8 \text{ m/s}}{\lambda^2}$$

6)

Photon density

In[927]:= $\text{UnitConvert}[1 \text{ Jy} * 1 \text{ Hz} / (h * 1 \text{ MHz}), "1 / (\text{cm}^2 \text{ s})"]$

Out[927]= 0.001509190 per centimeter² per second

8)

Relation between magnitudes and fluxes

In[928]:= $\text{eqM} := m_1 - m_2 == -2.5 \text{ Log10}[f_1 / f_2]$

a) $m_b = 8.33$

In[929]:= $\text{Solve}[\text{eqM} /. \{m_1 \rightarrow 4.71, f_1 \rightarrow 375, m_2 \rightarrow 8.33\}, f_2][[1, 1, 2]] \text{ Jy}$

Out[929]= 13.366917503484 Jy

b) $m_b = -0.32$

```
In[930]:= Solve[eqM /. {m1 → 4.71, f1 → 375, m2 → -.32}, f2] [[1, 1, 2]] Jy
```

```
Out[930]:= 38550.611179743 Jy
```

9)

Total magnitude formula

```
In[931]:= mT := -2.5 Log10[10^(-m1 .4) + 10^(-m2 .4)]
```

Substituting values

```
In[932]:= "Total magnitude" → mT /. {m1 → 8.34, m2 → 8.34}
```

```
Out[932]:= Total magnitude → 7.58742501084
```