log-log-calibration

Unknown Author

March 22, 2014

1 Use of log-log plot in calibraiton and regression

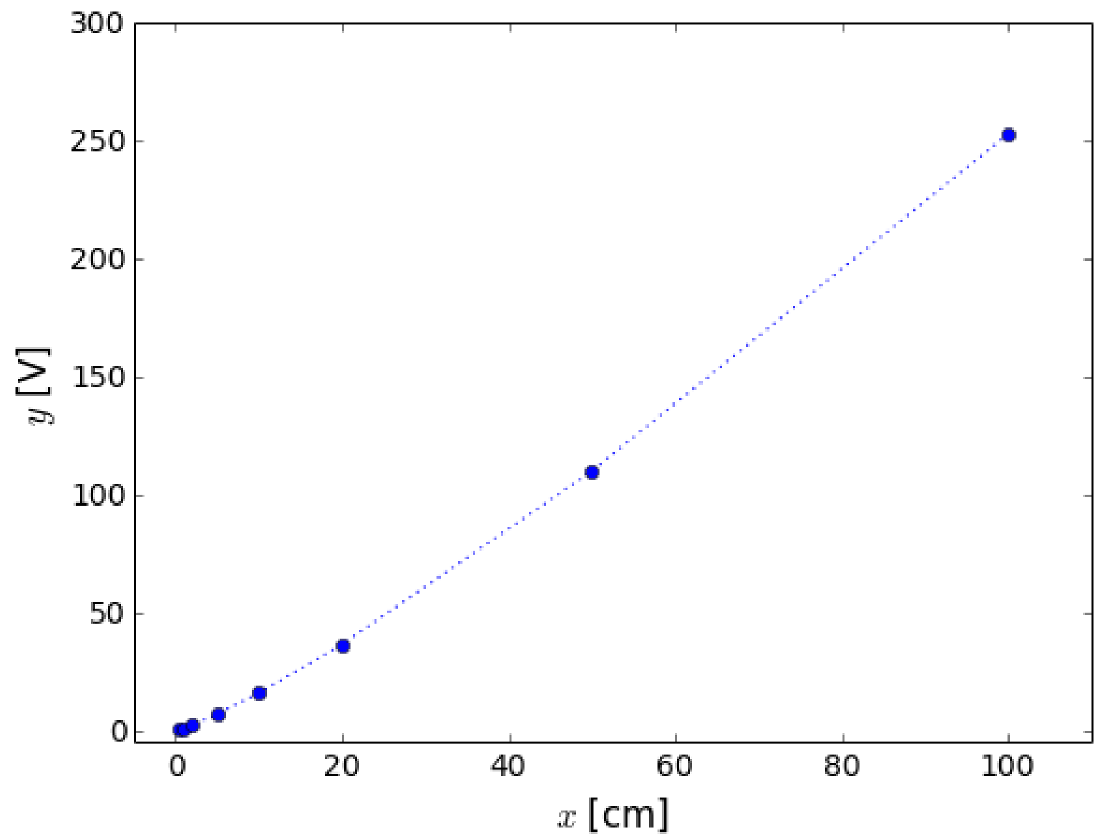
In [10]:
```
# read the data
import numpy as np
x = np.array([0.5, 1.0, 2.0, 5.0, 10.0, 20.0, 50.0, 100.0]) # cm
y = np.array([0.4, 1.0, 2.3, 6.9, 15.8, 36.4, 110.1, 253.2]) # Volt
```

In [11]:
```
mpl.rc_context(rc={'font.size': 14, 'axes.labelsize':'large', 'figure.figsize': (8,6)
```

Out [11]:
```
<matplotlib.rc_context at 0x1062d4850>
```

In [12]:
```
fig = figure()
plot(x,y,'o:')
xlim(-5,110)
ylim(-5,300)
xlabel('$x$ [cm]')
ylabel('$y$ [V]')
```

Out [12]:
```
<matplotlib.text.Text at 0x1062cc950>
```



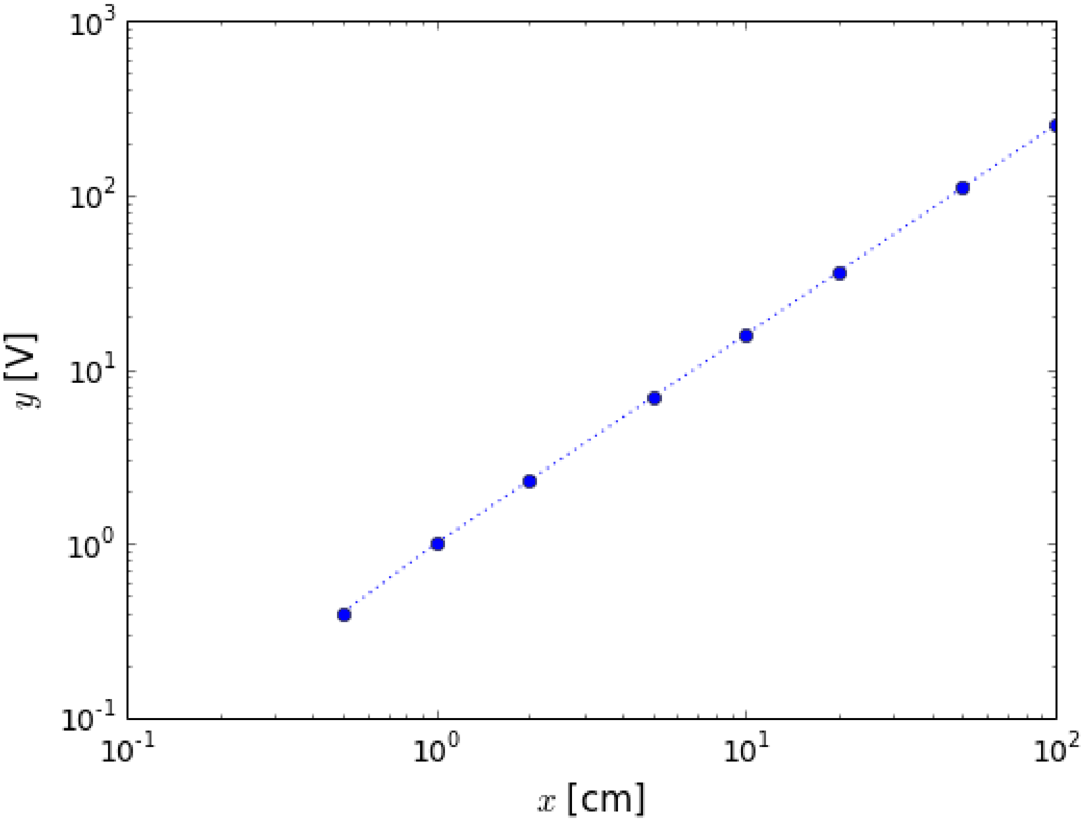
1.1 Conclusion:

1. Obviously the result is non-linear and the best sensitivity we can get is between 20 and 40 cm, but not for small distances
2. If the relationship is a known function and it's a physically relevant one, we could use it to show that distance to voltage should be related as a power law and not really linearly. Let's check the logarithmic scale, since we know that:

$\log y = \log(bx^m) = \log b + m \log x$

In [13]:
```
fig = figure()
loglog(x,y,'o:')
# xlim(-5,110)
# ylim(-5,300)
xlabel('$x$ [cm]')
ylabel('$y$ [V]')
```

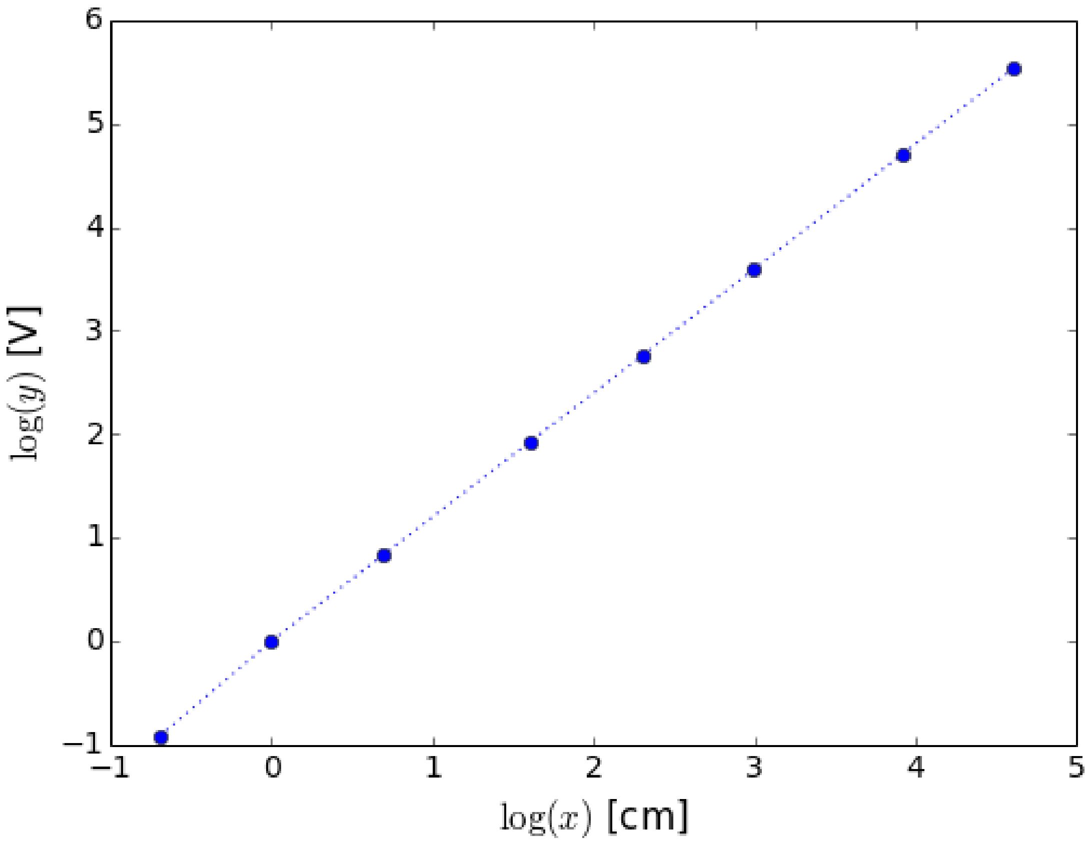
Out [13]:
```
<matplotlib.text.Text at 0x1062e5e10>
```



This result shows that the result is close to linear in the log-log space. Let's do calibration in $\log(x)$ and $\log(y)$. It can be either $\log$ or $\log_{10}$

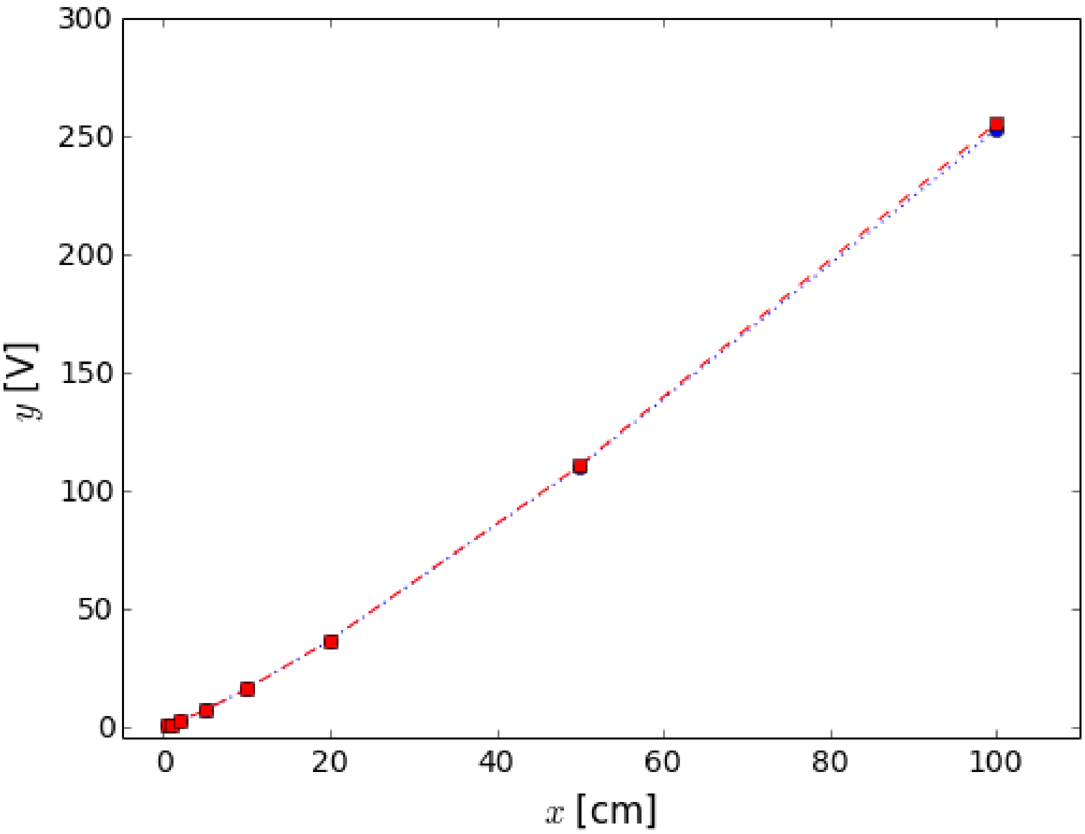
In [14]:
```
fig = figure()
plot(log(x),log(y),'o:')
# xlim(-5,110)
# ylim(-5,300)
xlabel('$\log (x)$ [cm]')
ylabel('$\log (y) $ [V]')
```

Out [14]:
```
<matplotlib.text.Text at 0x106e1df10>
```



In [15]:
```
# linear fit
p = polyfit(log10(x),log10(y),1)
p
```
Out [15]:
```
array([ 1.21031575, -0.01252781])
```

In [16]:
```
# let's check the fit:
fig = figure()
yfit = 10**(p[0]*log10(x) + p[1])
plot(x,y,'o:',x,yfit,'rs--')
xlim(-5,110)
ylim(-5,300)
xlabel('$x$ [cm]')
ylabel('$y$ [V]')
```
Out [16]:
```
<matplotlib.text.Text at 0x106e4bf90>
```



1.2 how do we measure?

1. measure the y [V], take a $\log_{10}(y)$, estimate the $\log_{10}(x)$ using the calibration curve:

$\log_{10}(x) = (\log_{10}(y) + 0.0125)/1.21$

2. Note that we can use this function in the full input scale and full output scale, take:

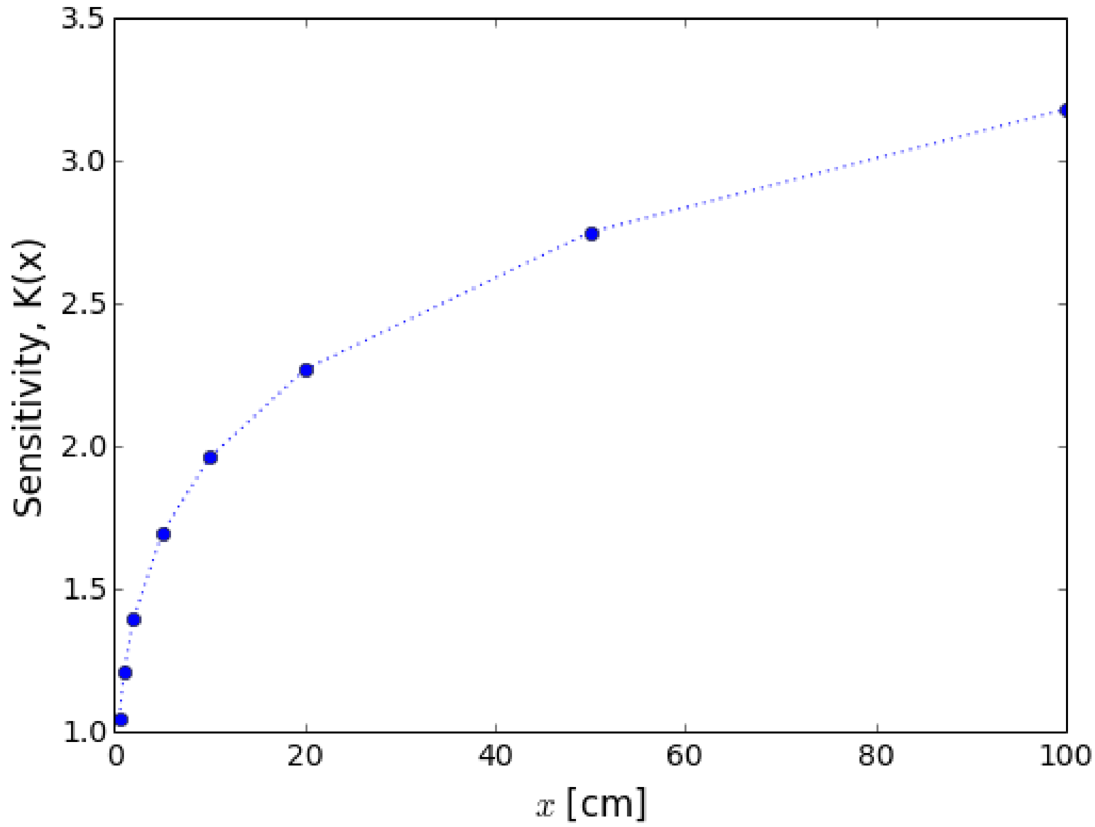
$10^{\log_{10}(x)}$

3. The sensitivity of this function is:

$K(x) = dy/dx = d/dx(x^{1.2}) = 1.2x^{0.2}$

In [22]:
```
fig = figure()
K = 1.21*x**0.21
plot(x,K,'o:')
xlabel('$x$ [cm]')
ylabel('Sensitivity, K(x) ')
```

Out [22]:
```
<matplotlib.text.Text at 0x106eeb750>
```



In [ ]: