Experiment 3: kB

Madeleine Allen Edward Piper

2/6/2019

Recording the resistor values measured during lab.

Experiment 3: Johnson Noise - Boltzmann Constant

```
experiment3data1<-read.csv("/Users/mallen/Documents/128AL/JohnsonNoise128AL/experiment3data1.csv")
```

Calculate Vmeas, V, and Vsystem

```
Vsys<- experiment3data1[1,7] #first row 7th column
VsysError <- experiment3data1[1,9]

Vmeask1<- (experiment3data1[2,7])
Vmeask10<-experiment3data1[3,7]
Vmeask20 <-experiment3data1[4,7]
Vmeask32<-experiment3data1[5,7]
Vmeask48<-experiment3data1[6,7]
Vmeask100<-experiment3data1[7,7]

Vmeas<-c(Vmeask1, Vmeask10, Vmeask20, Vmeask32, Vmeask48, Vmeask100)

#need to redo the error later (2/5)
VmeasError<-sqrt((sum(experiment3data1[2:7,9])^2))

V<- sqrt(-Vsys^2+Vmeas^2)
```

```
## Warning in sqrt(-Vsys^2 + Vmeas^2): NaNs produced
```

```
Verror<- sqrt(VmeasError^2+ VsysError^2)
```

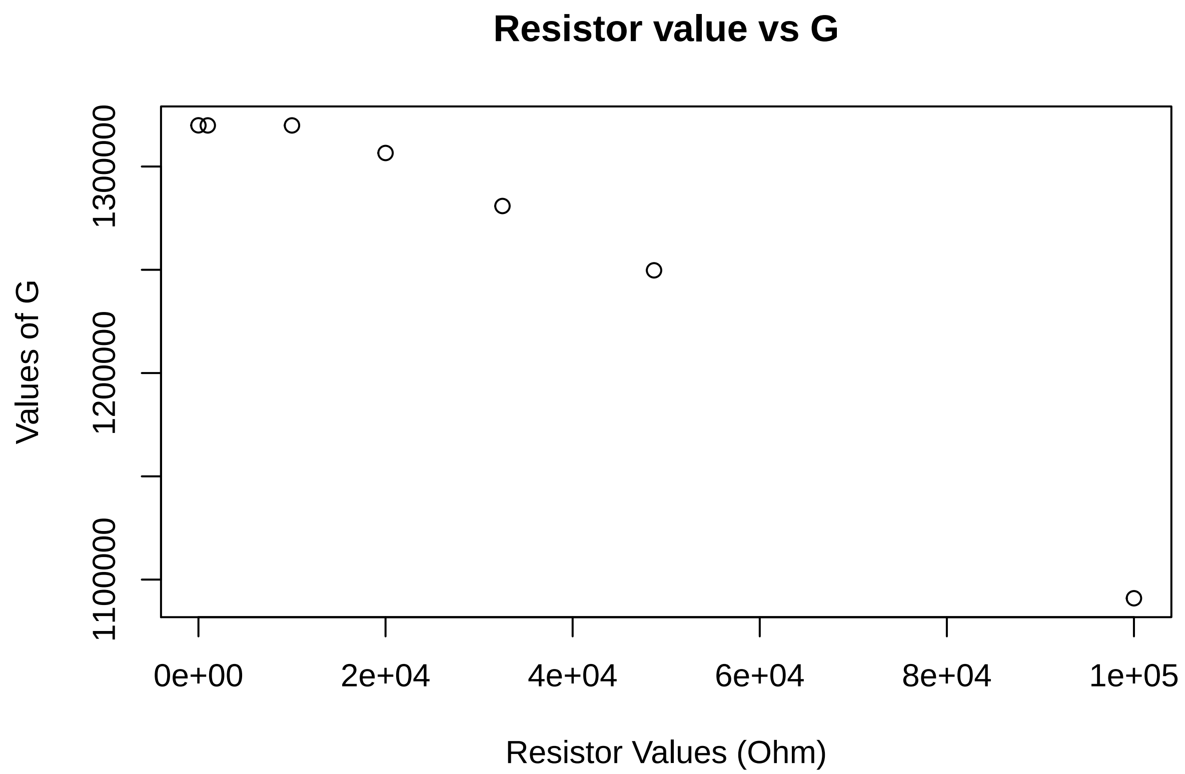
Calculating G

```
gain <- data.frame(Frequency = vout1$x, Gain = (m_vout[2]/m_in))

capacitance <-87.875*(10^-12)
capacitanceError <-.594*(10^-12)
#df is just the x componenent

riemanSum <- function(fa,fb){
  area <-0.5*(125)*(fb-fa)+fa*125
  return(area)
}

resistors<-read.csv("experiment3data1.csv")

C = capacitance
```

```
integrand <- data.frame(
  gain[2]/(1+(2*pi*C*vin1$x*short)^2),
  gain[2]/(1+(2*pi*C*vin1$x*k1)^2),
  gain[2]/(1+(2*pi*C*vin1$x*k10)^2),
  gain[2]/(1+(2*pi*C*vin1$x*k20)^2),
  gain[2]/(1+(2*pi*C*vin1$x*k35)^2),
  gain[2]/(1+(2*pi*C*vin1$x*k48)^2),
  gain[2]/(1+(2*pi*C*vin1$x*k100)^2)
  )
area <- data.frame(
  G1 =0,
  G2 =0,
  G3 =0,
  G4 =0,
  G5 =0,
  G6 =0,
  G7 =0

  )
for(i in 1:length(integrand))
{
    for(l in 1:398)
    {
      if(is.na(integrand[l+1,i]))
      {
        break
      }
      else
      {
        area[i] <- area[i]+ riemanSum(integrand[l,i],integrand[l+1,i])
      }
    }
}
```

So this returns a gain value G for each resistor (called “area”)

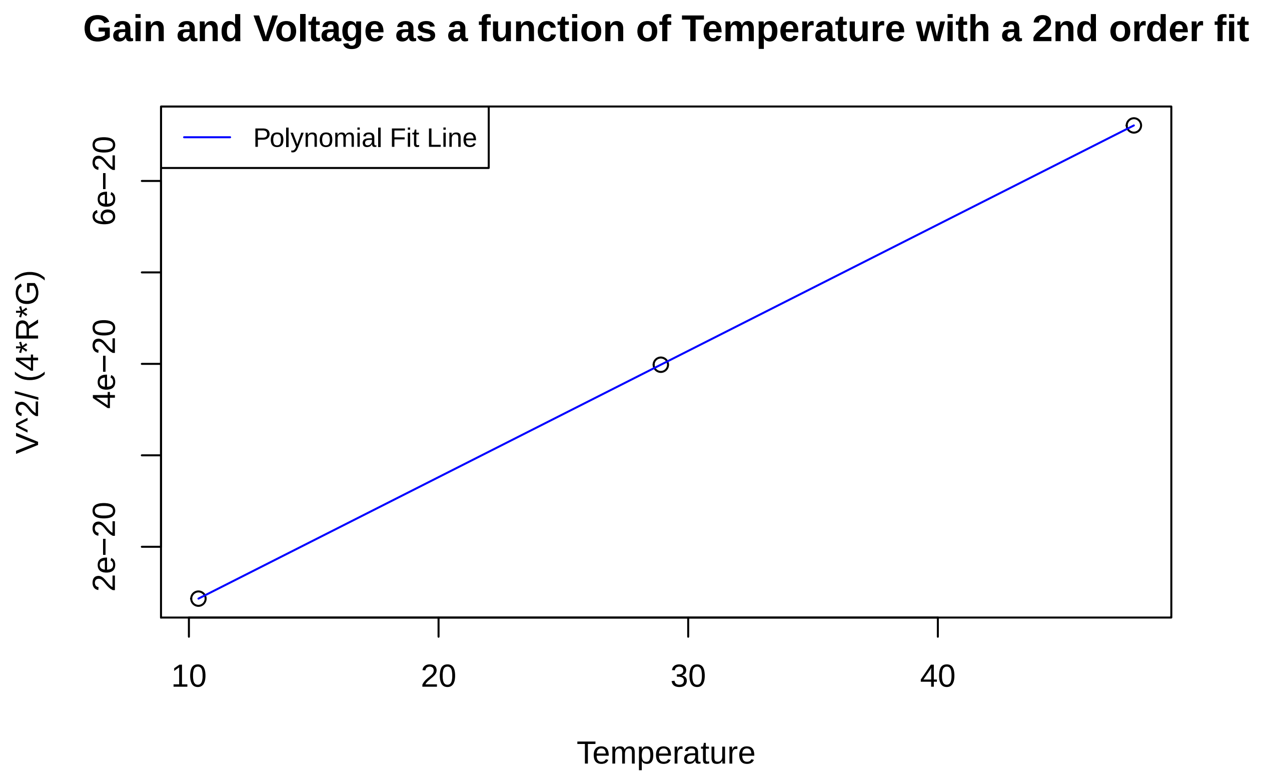
```
resistors<-c(0,1000,10000,20000,32500, 48700,100000)
plot(resistors,area, main= "Resistor value vs G", ylab = "Values of G", xlab = "Resistor Values (Ohm)")
```



```
#prepare the data for graphing
kb<- 1.38064852 *10^-23 #m2 kg s-2 K-1

area2<-area[2:7] #take away the short's data
resistors2 <-resistors[2:7] #take away the short
y_value2<- (V^2)/(4*resistors2*area2)
y2<- unlist(y_value2, use.names=FALSE)


Temperature<- ((V^2)/(4*kb*area2*resistors2))/100
Temperature2<-unlist(Temperature, use.names = FALSE)

summary(Temperature2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   10.38   19.64   28.90   29.05   38.38   47.85       3
```

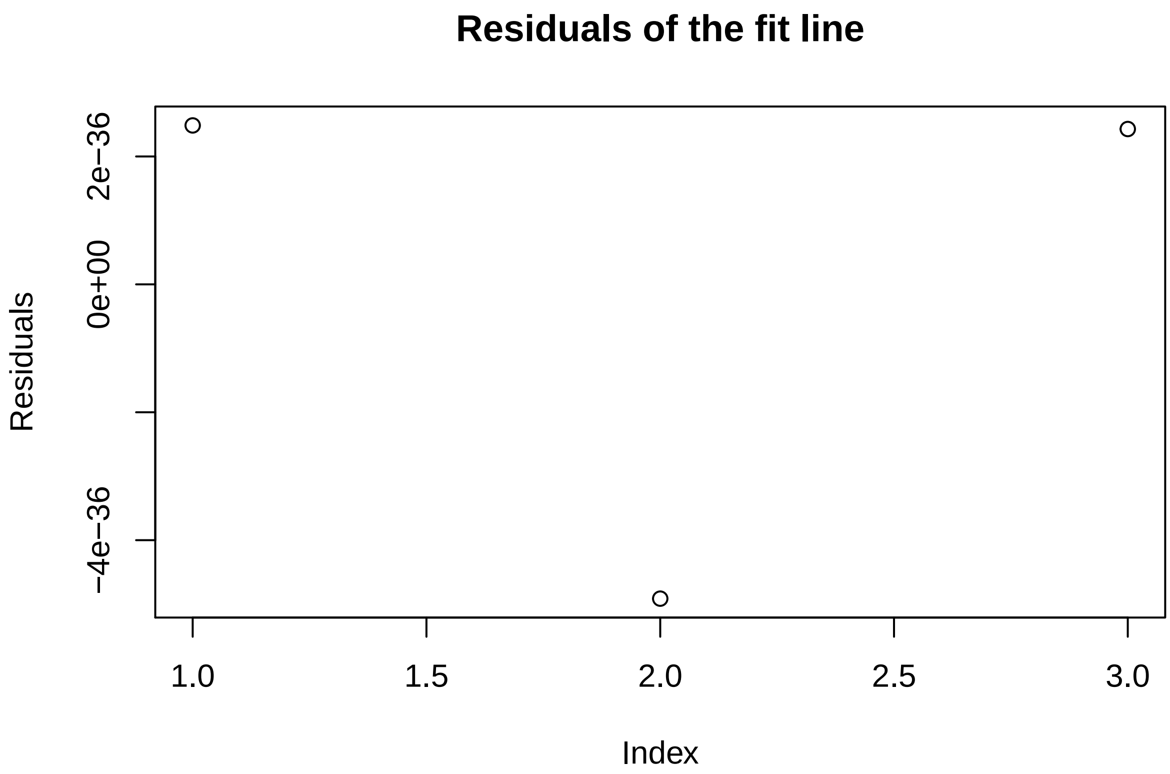
```
#2nd degree fit
fit <- lm(y2~Temperature2)

#plot same as above this time with a 2d fit line
plot(Temperature2,y2, main= "Gain and Voltage as a function of Temperature with a 2nd order fit", ylab=
lines(Temperature2, predict(fit, data.frame(Temperature2)), col="blue")
legend("topleft", legend=c("Polynomial Fit Line"),
       col=c("blue"), lty=1:2, cex=0.8)
```



```
summary(fit)
```

```
## Warning in summary.lm(fit): essentially perfect fit: summary may be
## unreliable

##
## Call:
## lm(formula = y2 ~ Temperature2)
##
## Residuals:
##          4          5          6
##  2.485e-36 -4.914e-36  2.429e-36
##
## Coefficients:
##                Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  -6.950e-36  7.456e-36 -9.320e-01    0.522
## Temperature2  1.381e-21  2.271e-37  6.079e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.019e-36 on 1 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.695e+31 on 1 and 1 DF,  p-value: < 2.2e-16
```

```
plot(fit$residuals, main = "Residuals of the fit line", ylab= "Residuals")
```



Seems small for kb