Experiment 3: kB for Gain 150

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)

VmeasError<-sqrt((sum(experiment3data1[2:7,9])^2))

V<- sqrt(-Vsys^2+Vmeas^2)
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
#also changed here
riemanSum <- function(f){
  area<-(125/2)*(f[1]+2*sum(f[2:398])+f[399])
  return(area)
}


C = capacitance
integrand <- data.frame(
  gain[2]^2/(1+(2*pi*C*vin1$x*short)^2),
  gain[2]^2/(1+(2*pi*C*vin1$x*k1)^2),
  gain[2]^2/(1+(2*pi*C*vin1$x*k10)^2),
  gain[2]^2/(1+(2*pi*C*vin1$x*k20)^2),
```

```
  gain[2]^2/(1+(2*pi*C*vin1$x*k35)^2),
  gain[2]^2/(1+(2*pi*C*vin1$x*k48)^2),
  gain[2]^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))
{
        area[i] <- riemanSum(unlist(integrand[i]))
}
```

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

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

area2<-area[2:7]/100 #take away the short's data
resistors2 <-resistors[1:6]
y_value2<- (V^2)/(4*resistors2*area2)
y2<- unlist(y_value2, use.names=FALSE)

#calculate the temperatures
Temperature<- ((V^2)/(4*kb*area2*resistors2))
Temperature2<-unlist(Temperature, use.names = FALSE)
fit <- lm(y2~Temperature2)
```

```
Temperature2<-Temperature2/100
Temperature2[2]<- Temperature2[2]/10
print(Temperature2)
```

```
## [1] 70.75617 66.41194 66.79345 66.13097 68.47735 64.37755
```

```
Temperature2Error<-Temperature2*sqrt((V^2/Verror^2)+(resistors2/resistorserror)^2)
```

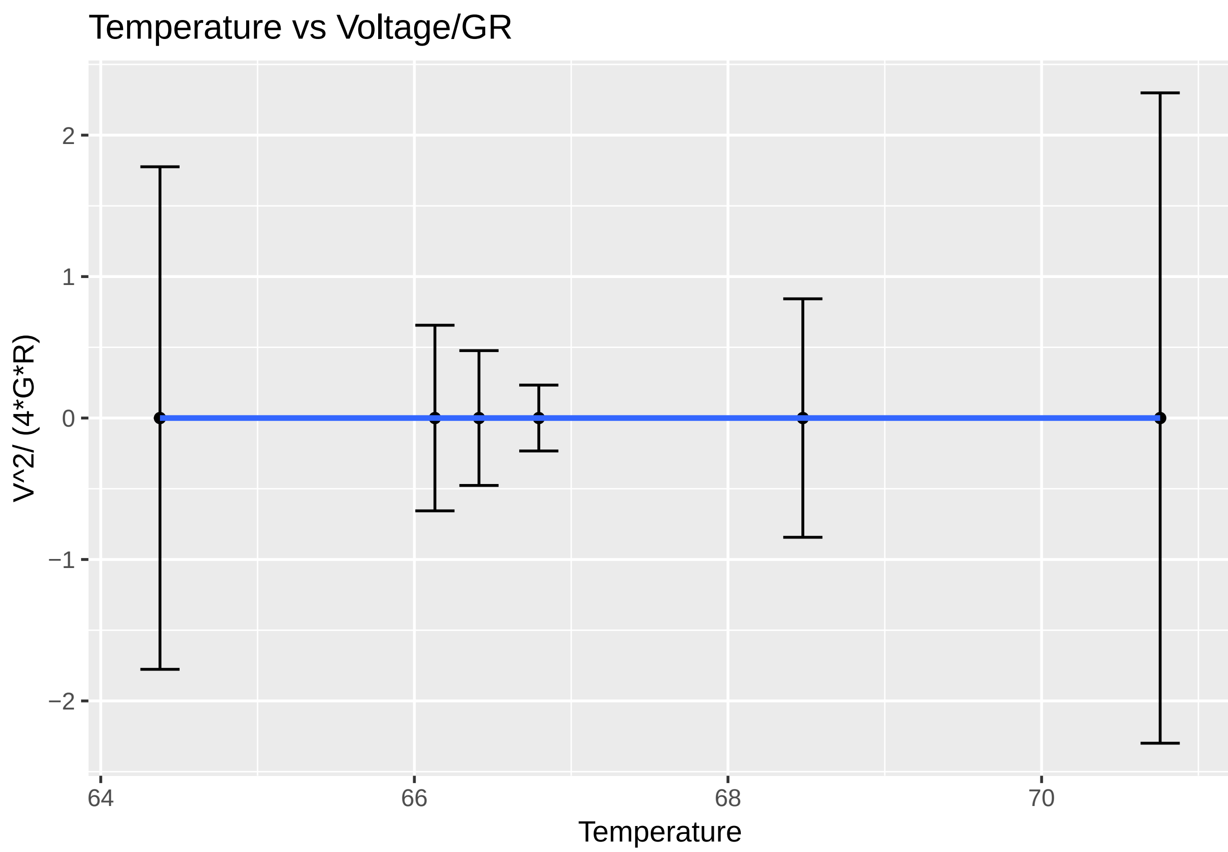
Calculating Errors

```
tempAvg<-mean(Temperature2)
t1 = (((tempAvg-Temperature2[1])^2))/sqrt(6)
t2 = (((tempAvg-Temperature2[2])^2))/sqrt(6)
t3 = (((tempAvg-Temperature2[3])^2))/sqrt(6)
t4 = (((tempAvg-Temperature2[4])^2))/sqrt(6)
t5 = (((tempAvg-Temperature2[5])^2))/sqrt(6)
t6 = (((tempAvg-Temperature2[6])^2))/sqrt(6)

Temperature2Error2<- sqrt(c(t1, t2, t3, t4, t5, t6))
```

```
library(ggplot2)
qplot((Temperature2),(y2))+geom_errorbar(aes(x=(Temperature2), ymin=(y2-Temperature2Error2), ymax=(y2+Te
```

```r
  geom_smooth(method="lm", se=FALSE, fullrange=TRUE, level=0.95)+labs(title = "Temperature vs Voltage/Gl
```

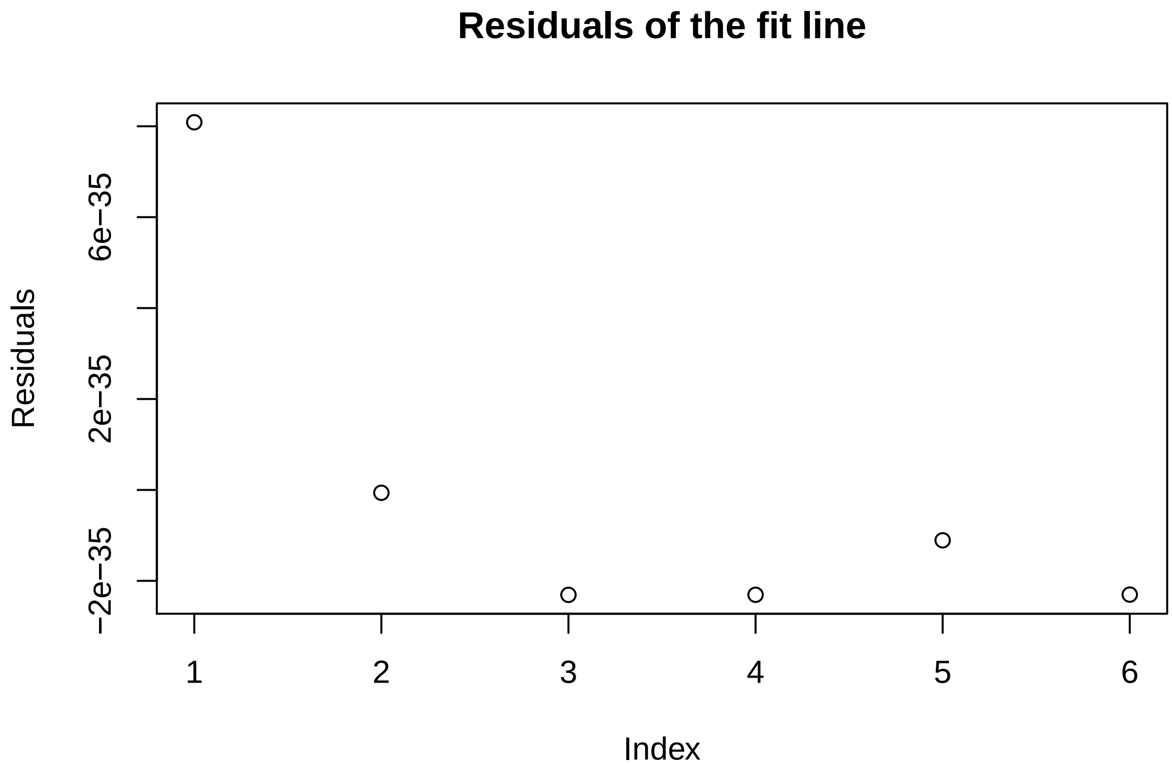



```r
summary(fit)
```

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

##
## Call:
## lm(formula = y2 ~ Temperature2)
##
## Residuals:
##          1          2          3          4          5          6
##  8.087e-35 -6.240e-37 -2.308e-35 -2.306e-35 -1.108e-35 -2.303e-35
##
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept)  7.863e-35  2.319e-35 3.391e+00   0.0275 *
## Temperature2 1.381e-23  8.340e-40 1.655e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.544e-35 on 4 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.741e+32 on 1 and 4 DF,  p-value: < 2.2e-16
```

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



Residuals of the fit line
Residuals
6e−35
2e−35
−2e−35
1
2
3
4
5
6
Index