Experiment 2: Temperatures for Gain 150

Madeleine Allen, Edward Piper

2/7/2019

Experiment 2: Johnson Noise 1

Recording the resistor values measured during lab.

```
#everything will be in ohms
short<- .03
shortError<-.001

k20<-20090
k20error<-1

k35 <- 35230 #secretly 35.2 but that would be an ugly variable name
k35error<-1

k100 <- 100700
k100error <- 1

k10 <- 999.05
k10error<- .01

k1 <- 998.17
k1error <- .01

k48 <- 48650 #secretly 48.7k but again that would be an ugly variable name
k48error<- 1

resistors<-c(k1,k10,k20,k35, k48,k100)
resistorserror<-c(k1error, k10error, k20error, k35error, k48error, k100error)*1000
```

Import Band Voltage measurements from experiment 2

```
experiment2data<-read.csv("/Users/mallen/Documents/128AL/JohnsonNoise128AL/experiment2data1.csv")
```

Calculate Vmeas, V, and Vsystem

```
Vsys<- experiment2data[1,7] #first row 7th column
VsysError <- experiment2data[1,9]

Vmeask1<- (experiment2data[2,7])
Vmeask10<-experiment2data[3,7]
Vmeask20 <-experiment2data[4,7]
Vmeask32<-experiment2data[5,7]
Vmeask48<-experiment2data[6,7]
Vmeask100<-experiment2data[7,7]

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

#need to redo the error later (2/5)
```

```
VmeasError<-sqrt((sum(experiment2data[2:7,9])^2))

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

Calculating G

```
capacitance <-87.875*(10^-12)
capacitanceError <-.594*(10^-12)
#df is just the x componenent
#fixed it
riemanSum <- function(f){
  area<-(125/2)*(f[1]+2*sum(f[2:398])+f[399])
  return(area)
}


#resistors<-read.csv("experiment2data1.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))
{
        area[i] <- riemanSum(unlist(integrand[i]))
}
area2error=sqrt((capacitance/capacitanceError)^2+(resistors/resistorserror)^2)
```

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)",
```

Plotting R as a function of V^2, kB, and G

```
kb<- 1.38064852 *10^-23 #m2 kg s-2 K-1

area2<-area[2:7] #take away the short's data
y_value<- (V^2)/(4*kb*area2) #area is the vector that contains all G's

#prepare data for graphing
resistors2 <-resistors[1:7]
y<- unlist(y_value, use.names=FALSE)
```

```
#I'll try finding temperatures

Temperature<- ((V^2)/(4*kb*area2*resistors2))/100
Temperature2<-unlist(Temperature, use.names = FALSE)
Temperature2[2]<-Temperature2[2]/10
print(Temperature2) #these are the correct values
```

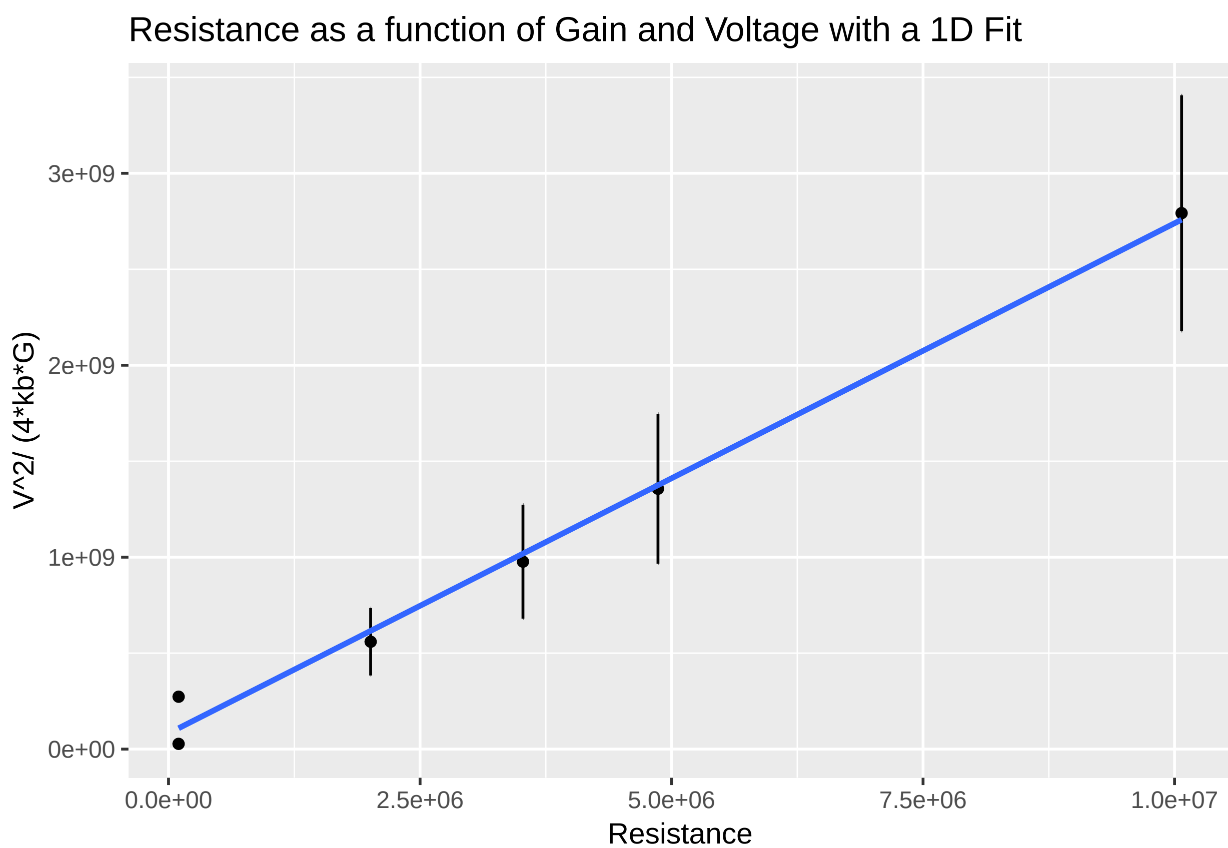
```
## [1] 273.7386 273.1847 278.4322 277.2176 278.8351 277.2966
```

```
#Error Propagation
```

Create a fit line

```
#
#resistor as the x axis and the other term as the y axis.
#
resistors3<-resistors2[1:6]*100
resistors3error <- resistors3*sqrt((V^2/Verror^2)+((area2/area2error)^2))/100
fit  <- lm(y~0+resistors3)
```

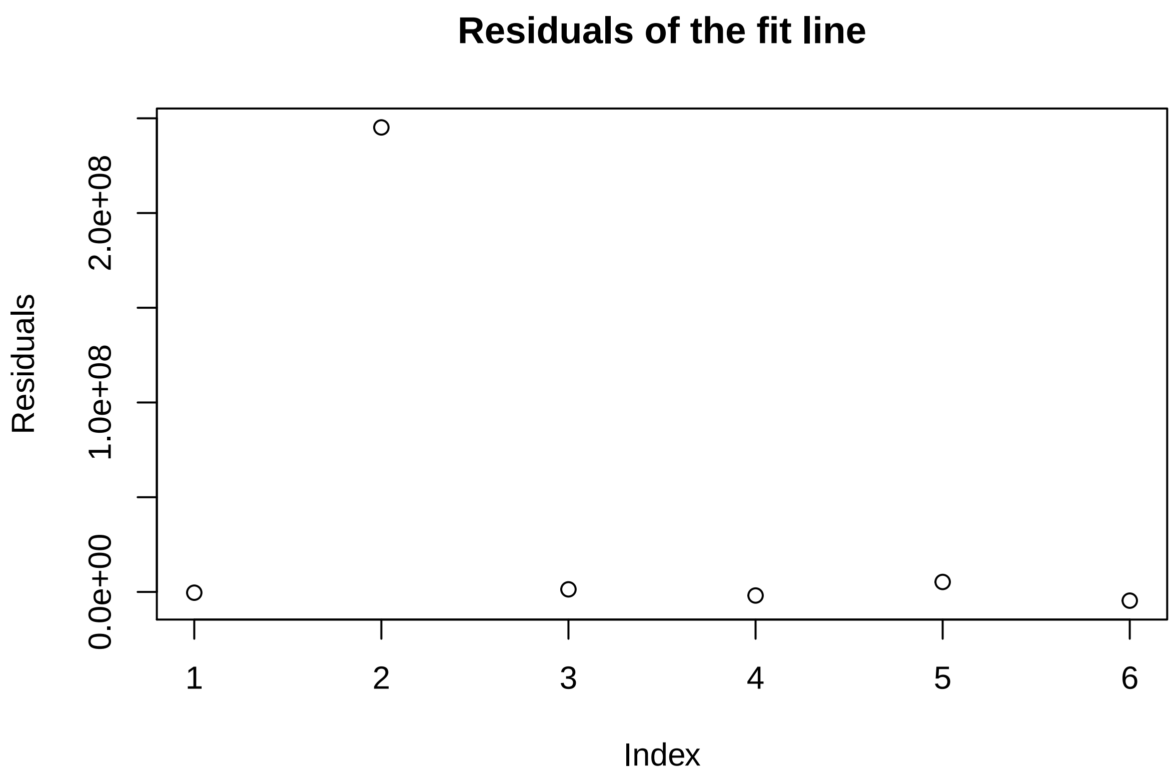
```
rm(gain)
rm(gainerror)

gain=data.frame()
gainerror=data.frame()
library(ggplot2)
qplot(unlist(resistors3),unlist(y))+geom_errorbar(aes(x=unlist(resistors3), ymin=unlist(y-resistors3erro
  geom_smooth(method="lm", se=FALSE, fullrange=TRUE, level=0.95)+labs(title = "Resistance as a function
```




```r
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ 0 + resistors3)
##
## Residuals:
##         1         2         3         4         5         6
##   -400619 245176371   1366167  -1883144   5268498  -4587479
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## resistors3   277.75       9.22   30.12 7.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 109700000 on 5 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9934
## F-statistic: 907.4 on 1 and 5 DF,  p-value: 7.562e-07
```

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




```r
#lines(resistors3, predict(fit, data.frame(resistors3)), col="red")
#legend("topleft", legend=c("Fit Line"), col=c("red"), lty=1:2, cex=0.8)

shapiro.test(y)#the data is normally distributed
```

```
##
##  Shapiro-Wilk normality test
##
## data:  y
## W = 0.89627, p-value = 0.3524
```

Then, to find the value of absolute 0 in Celsius, we use the recorded room temperature (20.5 C) and the average Temperature (277.75K). Solving for absolute 0, you get -257.3. We expected to find -273.15.