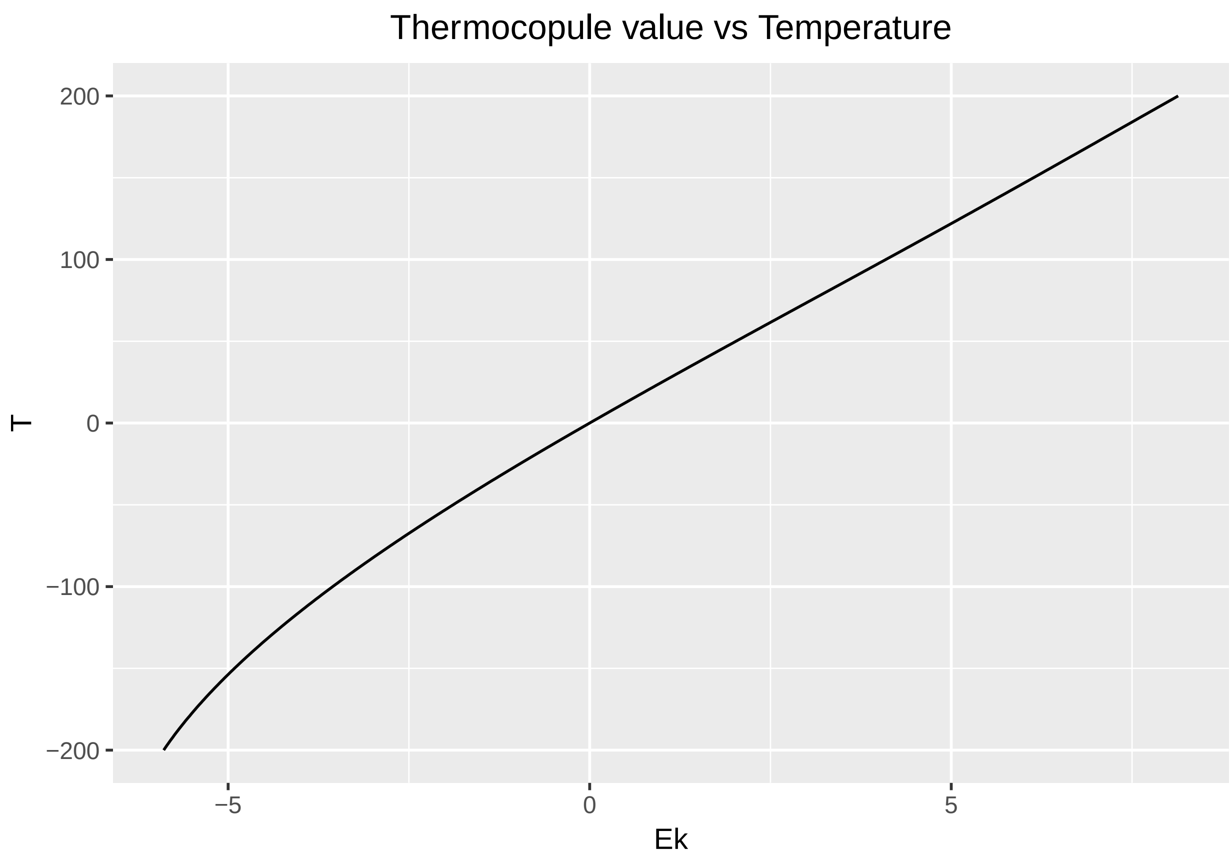
Hall Effect Data Analysis

Thermocopule temperature fit

```
therm.K<-read.csv(here('data',data[1]),sep=",")
```

```
TEkPlot<-ggplot(data=therm.K,aes(x=Ek,y=T))+
         geom_line()+
         ggtitle('Thermocopule value vs Temperature')+
         theme(plot.title = element_text(hjust = 0.5))
TEkPlot
```



In order to get a correct measurement it is necessary to compensate for the non-linearity (see figure 6) of the thermocouple using the following polynomial:

$$t_{\text{calc}} = d_0 + d_1 E + d_2 E^2 + \ldots + d_n E^n$$

A fitting polynomial of the fifth order is sufficient, given the precision of our equipment. Best fit coefficients obtained from NIST data tables ($-200 < t[^\circ C] < 200$)

Coefficients	Value
d_0	0.3837
d_1	25.22

Coefficients	Value
d_2	0.2795
d_3	0.07205
d_4	0.01409
d_5	0.001056

Valori ricalcolati con un modello lineare in forma

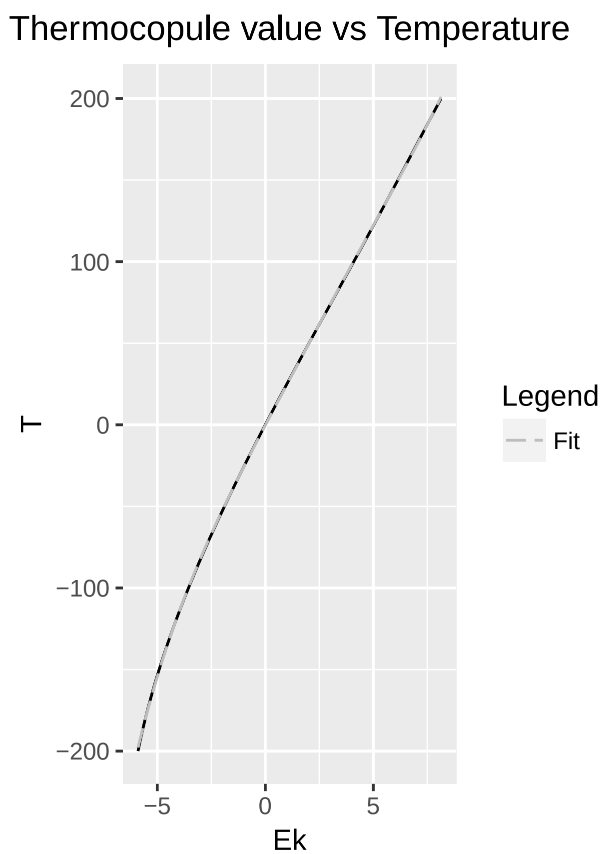
$$T = \beta_0 + \beta_1 E + \ldots + \beta_5 E^5$$

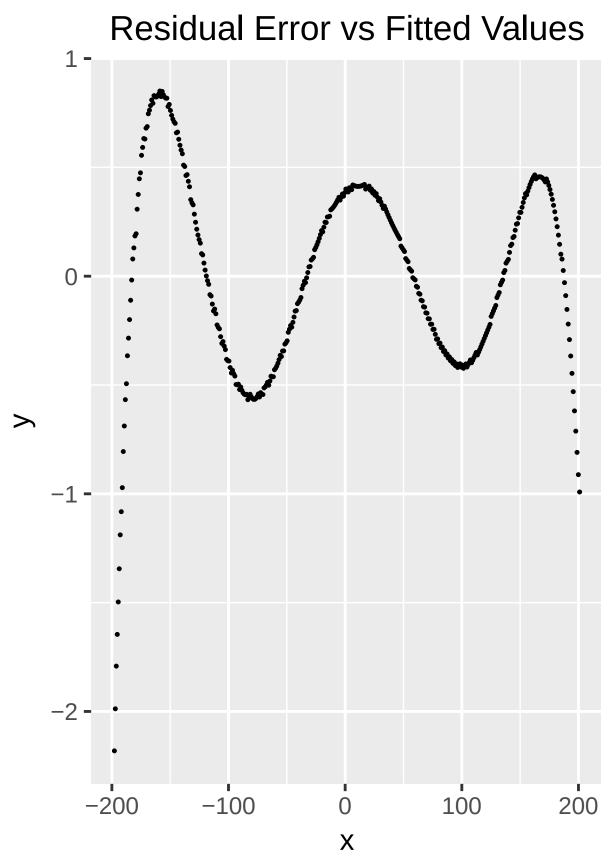
```
model<-lm(T ~ Ek+I(Ek^2)+I(Ek^3)+I(Ek^4)+I(Ek^5))#Coefficients calculated as function of Ek
knitr::kable(summary(model)$coef,digits=6)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.383696	0.047709	-8.042428	0
Ek	25.215124	0.022966	1097.911924	0
I(Ek^2)	-0.279517	0.007309	-38.243458	0
I(Ek^3)	0.072046	0.001705	42.249216	0
I(Ek^4)	-0.014095	0.000210	-67.056166	0
I(Ek^5)	0.001056	0.000034	30.848009	0

Si può notare che, a meno del segno, i coefficienti che vengono ricavati con un modello lineare sono identici, inoltre si può notare graficamente come il grafico dei valori calcolati dal modello sia quasi perfettamente sovrapposto a quelli misurati

```
FitVsReal<-TEkPlot+
           geom_line(data=data.frame(x=Ek,y=model$fitted.values),aes(x=x,y=y,colour='Fit'),linetype=5)+
           scale_colour_manual(name='Legend',values=c("Real"="black","Fit"="grey"))
ResidualErr<-ggplot(data=data.frame(x=fitted(model),y=model$residuals),aes(x,y))+
             geom_point(size=0.2)+
             ggtitle('Residual Error vs Fitted Values')+
             theme(plot.title = element_text(hjust = 0.5))

grid.arrange(FitVsReal,ResidualErr,ncol=2)
```





Resistenza e Tensione di Hall al variare della temperatura

```
hall.data<-read.csv(here('data',data[2]),sep=",")
```

Per questo avremo bisogno intanto di calcolare la temperatura in quanto abbiamo la tensione sulla termocoppia, quindi usiamo il modello precedente e andiamo a calcolare le temperature, quindi a convertire i risultati in gradi Kelvin. Per fare questo bisogna inoltre riscalare la tensione sulla termocoppia come

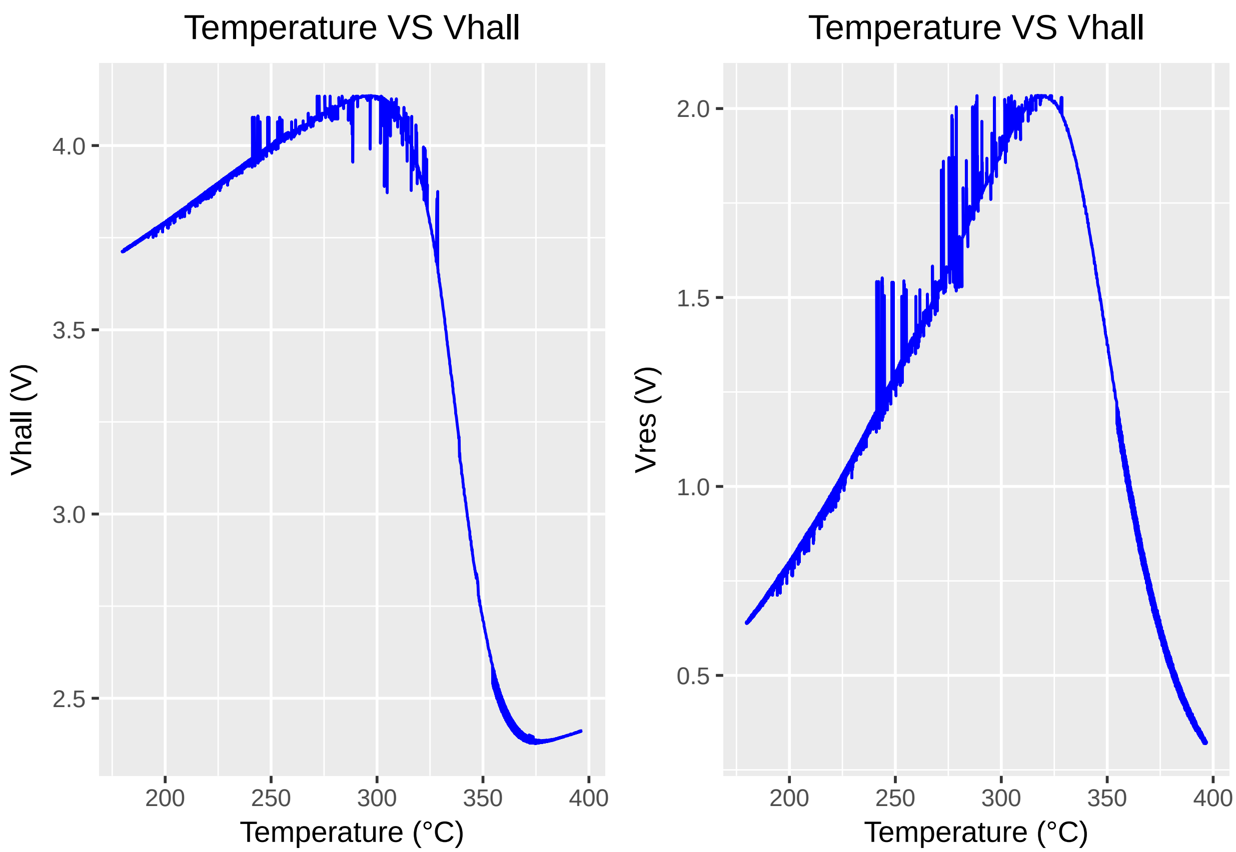
$$E = \frac{1}{2}\frac{V_{\text{out}} - 2.5 - 1.25 \cdot 10^{-3}}{122.4}$$

```
E<-function(Vout){
  (1/2)*(Vout-2.5-1.25e-3)/122.4
}

temperatures<-predict(model,newdata=data.frame(Ek=E(Vtermoc)*1000))+ 273.15
```

Fatto questo possiamo disegnare il grafico dell'andamento di tensione di hall e resistenza al variare della temperatura

```
tempVSVhall.plot<-ggplot()+
                  geom_line(data=data.frame(x=temperatures,y=Vhall),aes(x,y),col="blue")+
                  xlab("Temperature (°C)")+ylab("Vhall (V)")+
                  ggtitle("Temperature VS Vhall")+
                  theme(plot.title = element_text(hjust = 0.5))
```

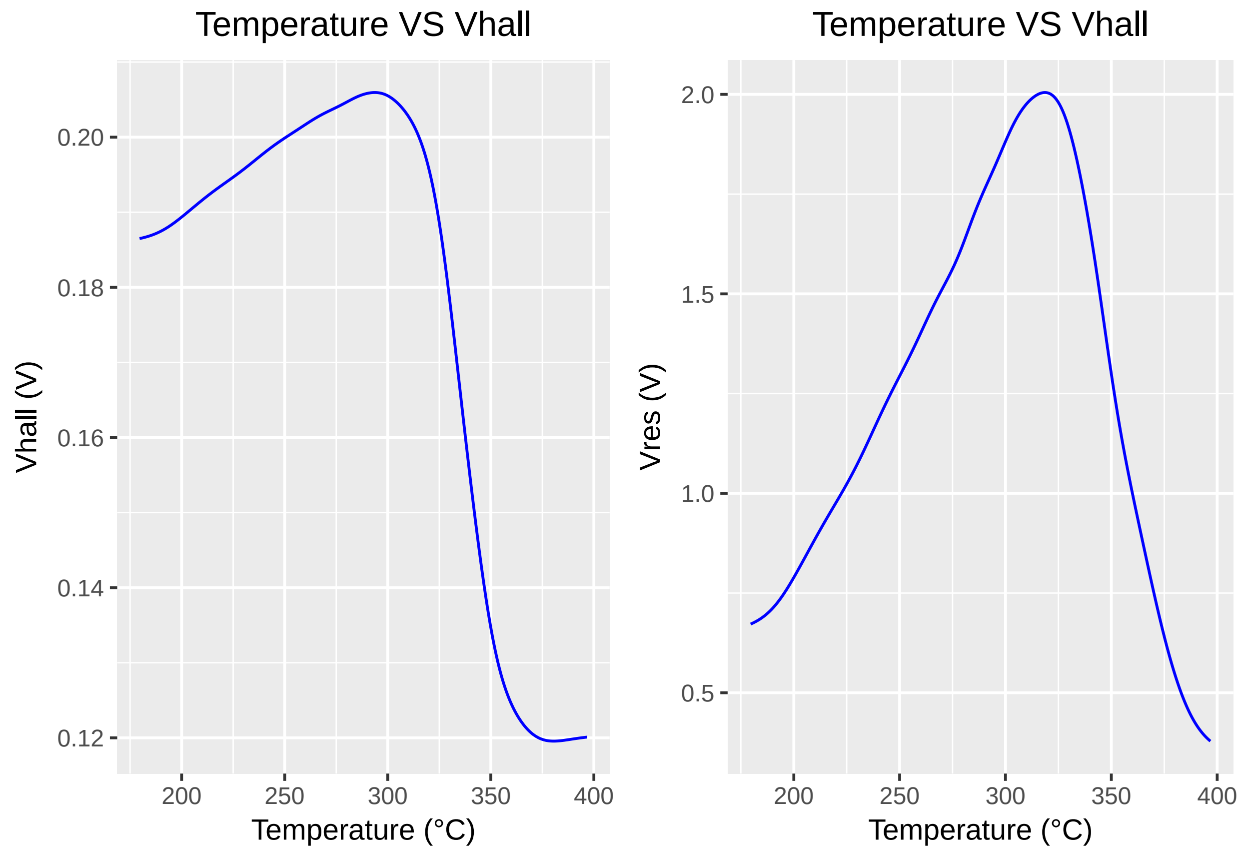
```
tempVSVres.plot<-ggplot()+
                 geom_line(data=data.frame(x=temperatures,y=Vres.fix),aes(x,y),col="blue")+
                 xlab("Temperature (°C)")+
                 ylab("Vres (V)")+
                 ggtitle("Temperature VS Vhall")+
                 theme(plot.title = element_text(hjust = 0.5))

grid.arrange(tempVSVhall.plot,tempVSVres.plot,ncol=2)
```



Applichiamo un lisciamento (a kernel normale) per eliminare i disturbi sperimentali in modo da ottenere dei grafici più chiari

```
tempVSVhall<-ksmooth(temperatures,Vhall/20,"normal",bandwidth =20)
tempVSVres<-ksmooth(temperatures,Vres.fix,"normal",bandwidth =20)

tempVSVhallsmooth.plot<-ggplot()+
                        geom_line(data=as.data.frame(tempVSVhall),aes(x,y),col="blue")+
                        xlab("Temperature (°C)")+
                        ylab("Vhall (V)")+
                        ggtitle("Temperature VS Vhall")+
                        theme(plot.title = element_text(hjust = 0.5))
tempVSVressmooth.plot<-ggplot()+
                       geom_line(data=as.data.frame(tempVSVres),aes(x,y),col="blue")+
                       xlab("Temperature (°C)")+
                       ylab("Vres (V)")+
                       ggtitle("Temperature VS Vhall")+
```

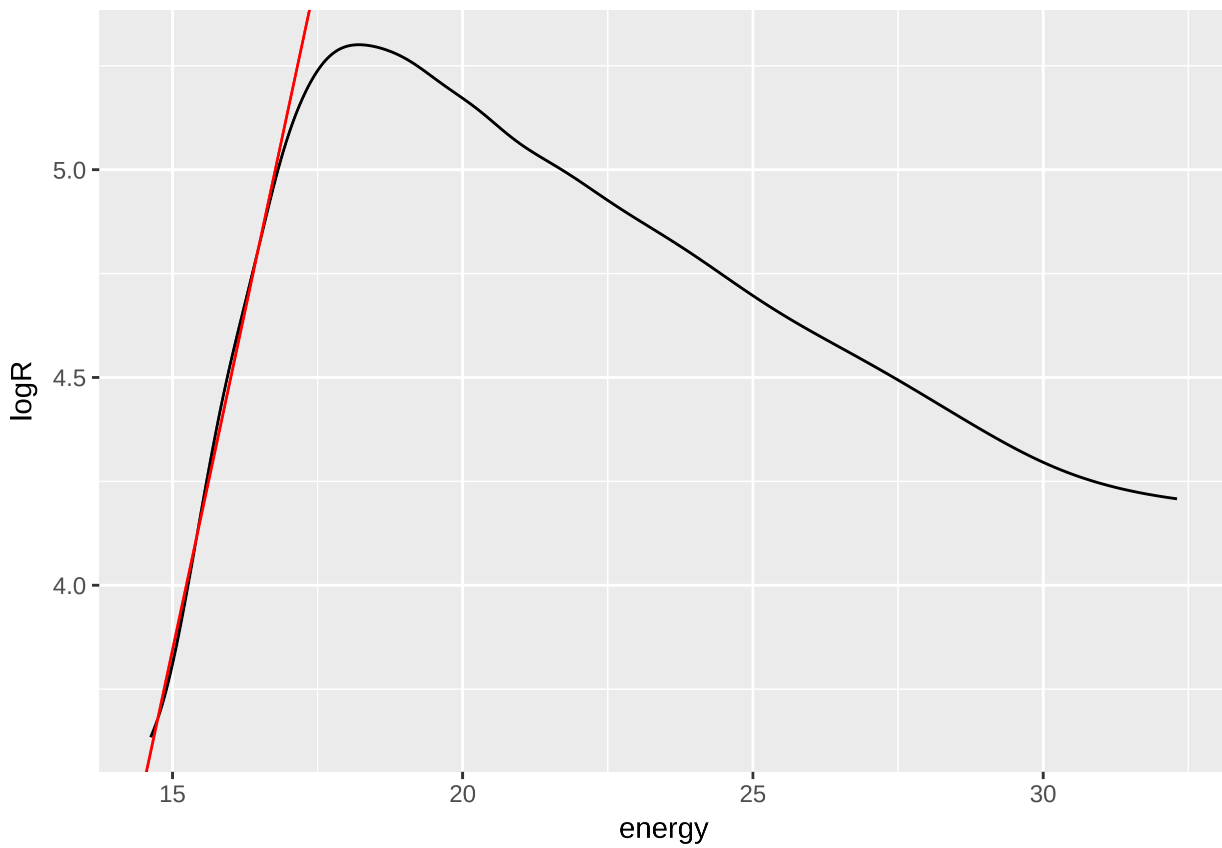
```
                                   theme(plot.title = element_text(hjust = 0.5))
grid.arrange(tempVSVhallsmooth.plot,tempVSVressmooth.plot,ncol=2)
```



Energia vs log R

Linearità nella prima parte della relazione tra eneriga e logR

```
K<-8.617e-5
energy<-1/(2*K*tempVSVres$x)
log.Vres<-log(tempVSVres$y)
logR<-log(tempVSVres$y/0.01)
model.x<-energy[2660:length(log.Vres)]
model.y<-logR[2660:length(log.Vres)]
model.lm<-lm(model.y~model.x)
model.lm.intercept<-model.lm$coefficients[1]
model.lm.slope<-model.lm$coefficients[2]
energyVsVres.plot<-ggplot()+geom_line(aes(energy,logR))+geom_abline(intercept=model.lm.intercept,slope=
energyVsVres.plot
```


5.0
4.5
4.0
logR
15
20
25
30
energy