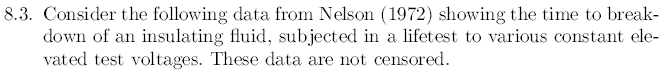
Chapter 8 Fitting parametric regression models

library(survival)

## Warning: package 'survival' was built under R version 4.0.4

knitr::include\_graphics("8\_3.PNG")



### Required data

The first thing we will do is clean the data load the data

### DataSet  
  
voltages <- read.csv("voltages.csv", header=T, sep =";")  
head(voltages)

## V30KV V32KV V34KV V36KV  
## 1 7.74 0.27 0.19 0.35  
## 2 17.05 0.40 0.78 0.59  
## 3 20.46 0.69 0.96 0.96  
## 4 21.02 0.79 1.31 0.99  
## 5 22.66 2.75 2.78 1.69  
## 6 43.40 3.91 3.16 1.97

### Data adjustment to eliminate NA values

Now we will adjust the data in independent sensors to facilitate their manipulation.

V30KV <- voltages$V30KV[1:11]  
V32KV <- voltages$V32KV[1:15]  
V34KV <- voltages$V34KV  
V36KV <- voltages$V36KV[1:15]

## Diagnostic tests

### Diagnostic 1.

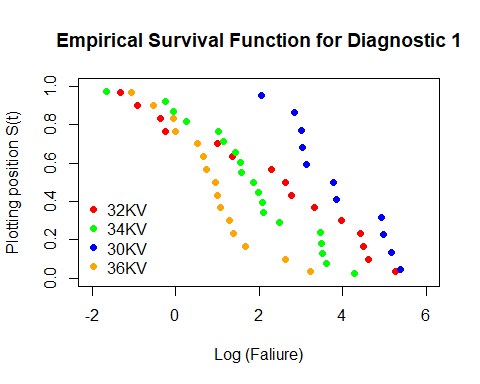
Calculating plotting position for the 4 covariates

### Plotting positions Failure 30KV  
  
i <- seq(1:length(V30KV))  
p30 <- 1 - ((i-0.5)/length(i))  
  
### Plotting positions at\_test 32KV  
  
i <- seq(1:length(V32KV))  
p32 <- 1 - ((i-0.5)/length(i))  
  
### Plotting positions Voltage 34KV  
  
i <- seq(1:length(V34KV))  
p34 <- 1 - ((i-0.5)/length(i))  
  
### Plotting positions levels 36KV  
  
i <- seq(1:length(V36KV))  
p36 <- 1 - ((i-0.5)/length(i))

### Diagnostic 1 graph

For the accelerated models we must calculate the empirical survival function, will show location shifted versions of the baseline survival function when an accelerated lifetimes model is appropriate.

### Plotting the data  
  
plot(log(V32KV),p32,col="red", pch = 19 , ylab= "Plotting position S(t)", xlab="Log (Faliure)",xlim=c(-2,6),ylim=c(0,1), main= "Empirical Survival Function for Diagnostic 1")  
points(log(V34KV),p34,col="green", pch = 19)  
points(log(V30KV),p30,col="blue", pch = 19)  
points(log(V36KV),p36,col="orange", pch = 19)  
legend("bottomleft", legend=c("32KV", "34KV", "30KV", "36KV"), col = c("red","green","blue","orange"), pch = 19, bty = "n")



Although the model presents curvatures, for some of its data they are not very pronounced, so they are not very far from the baseline, so it could be said that it is not an accelerated model.

### Diagnostic 2

By means of the standard deviation, a stabilization is given to the constant variance as proposed by the loglinear model.

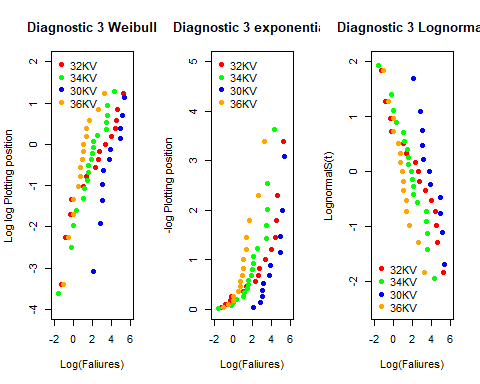
SD32 <- sd(log(V32KV))  
SD34 <- sd(log(V34KV))  
SD30 <- sd(log(V30KV))  
SD36 <- sd(log(V36KV))  
  
cbind(SD32, SD34, SD30, SD36)

## SD32 SD34 SD30 SD36  
## [1,] 2.188146 1.525211 1.124394 1.109901

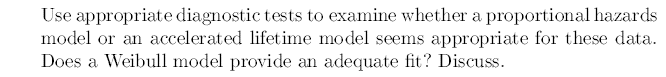
### Diagnostic 3 graph

Verifying the plots of the proportional hazards, it can be seen that these data fit a straight line under the Weibull model, as proposed in the exercise, Although it could also fit a log-normal distribution.

### Plotting the data  
  
par(mfrow=c(1,3))  
  
plot(log(V32KV),log(-1\*log(p32)),col="red", pch = 19 , ylab= "Log log Plotting position", xlab="Log(Faliures)", xlim=c(-2,6), ylim= c(-4,2), main = "Diagnostic 3 Weibull")  
points(log(V34KV),log(-1\*log(p34)),col="green", pch = 19)  
points(log(V30KV),log(-1\*log(p30)),col="blue", pch = 19)  
points(log(V36KV),log(-1\*log(p36)),col="orange", pch = 19)  
legend("topleft", legend=c("32KV", "34KV", "30KV", "36KV"), col = c("red","green","blue","orange"), pch = 19, bty = "n")  
  
plot(log(V32KV),-1\*log(p32),col="red", pch = 19 , ylab= "-log Plotting position", xlab="Log(Faliures)", xlim=c(-2,6), ylim= c(0,5), main = "Diagnostic 3 exponential")  
points(log(V34KV),-1\*log(p34),col="green", pch = 19)  
points(log(V30KV),-1\*log(p30),col="blue", pch = 19)  
points(log(V36KV),-1\*log(p36),col="orange", pch = 19)  
legend("topleft", legend=c("32KV", "34KV", "30KV", "36KV"), col = c("red","green","blue","orange"), pch = 19, bty = "n")  
  
plot(log(V32KV),qnorm(p32),col="red", pch = 19 , ylab= "LognormalS(t)", xlab="Log(Faliures)", xlim=c(-2,6), ylim= c(-2.5,2), main = "Diagnostic 3 Lognormal")  
points(log(V34KV),qnorm(p34),col="green", pch = 19)  
points(log(V30KV),qnorm(p30),col="blue", pch = 19)  
points(log(V36KV),qnorm(p36),col="orange", pch = 19)  
legend("bottomleft", legend=c("32KV", "34KV", "30KV", "36KV"), col = c("red","green","blue","orange"), pch = 19, bty = "n")



knitr::include\_graphics("8\_31.PNG")

 Next, we will compare the survival function with the Kaplan-Meier estimator model proposed by the task.

### Applying KM estimator

### Kaplan Meier models for the different voltages  
  
km.model32 <- survfit(Surv(V32KV,rep(1,15))~1)  
km.model34 <- survfit(Surv(V34KV,rep(1,19))~1)  
km.model30 <- survfit(Surv(V30KV,rep(1,11))~1)  
km.model36 <- survfit(Surv(V36KV,rep(1,15))~1)

Survival and times are taken for each model such that:

### Survival estimators and times for all the different voltages  
  
survEst32 <- km.model32$surv  
survTime32 <- km.model32$time  
  
survEst34 <- km.model34$surv  
survTime34 <- km.model34$time  
  
survEst30 <- km.model30$surv  
survTime30 <- km.model30$time  
  
survEst36 <- km.model36$surv  
survTime36 <- km.model36$time

### Model fitting

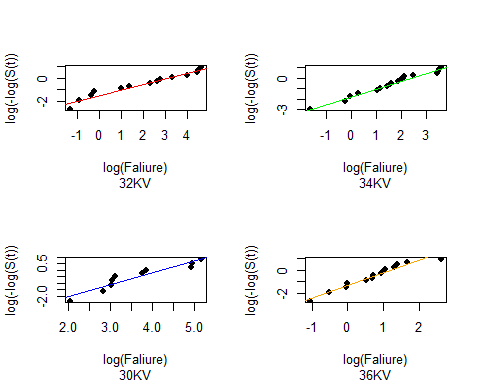
With the above values we fit each model for each covariate:

### Fitting the models  
  
logLogSurvEst32 <- log(-log(survEst32[-15]))  
logSurvTime32 <- log(survTime32[-15])  
result.lm32 <- lm(logLogSurvEst32 ~ logSurvTime32)  
  
logLogSurvEst34 <- log(-log(survEst34[-19]))  
logSurvTime34 <- log(survTime34[-19])  
result.lm34 <- lm(logLogSurvEst34 ~ logSurvTime34)  
  
logLogSurvEst30 <- log(-log(survEst30[-11]))  
logSurvTime30 <- log(survTime30[-11])  
result.lm30 <- lm(logLogSurvEst30 ~ logSurvTime30)  
  
logLogSurvEst36 <- log(-log(survEst36[-15]))  
logSurvTime36 <- log(survTime36[-15])  
result.lm36 <- lm(logLogSurvEst36 ~ logSurvTime36)

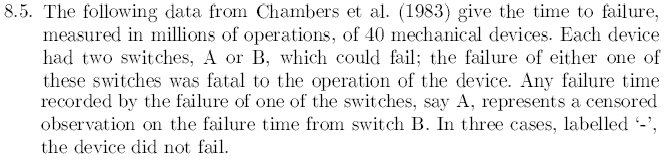
### Weibull model fitting graph

Finally fitting the model for all the covariates we can see how the line through them fits almost all the points and with a minimum distance between them and the regression line, so that we can fit the Weibull model as proposed in the exercise to these data.

par(mfrow=c(2,2))  
  
plot(logLogSurvEst32 ~ logSurvTime32, pch=16, xlab="log(Faliure)", sub="32KV",ylab="log(-log(S(t))")  
abline(result.lm32, col ="red")  
  
plot(logLogSurvEst34 ~ logSurvTime34, pch=16, xlab="log(Faliure)", sub="34KV",ylab="log(-log(S(t))")  
abline(result.lm34, col ="green")  
  
plot(logLogSurvEst30 ~ logSurvTime30, pch=16, xlab="log(Faliure)", sub="30KV",ylab="log(-log(S(t))")  
abline(result.lm30, col ="blue")  
  
plot(logLogSurvEst36 ~ logSurvTime36, pch=16, xlab="log(Faliure)", sub="36KV",ylab="log(-log(S(t))")  
abline(result.lm36, col ="orange")



knitr::include\_graphics("8\_5.PNG")



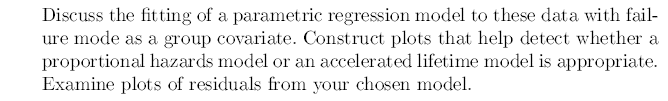
### Required data

The first thing we will do is clean the data load the data

swt <- read.csv("switches.csv", header=T, sep =";")  
head(swt)

## time A B Group  
## 1 1.151 0 1 B  
## 2 1.170 0 1 B  
## 3 1.248 0 1 B  
## 4 1.331 0 1 B  
## 5 1.381 0 1 B  
## 6 1.499 1 0 A

knitr::include\_graphics("8\_51.PNG")



The data were adjusted to independent vectors in order to work better with them, an auxiliary variable has been generated to separate both groups of switches.

GA <- subset(swt, Group == "A")  
GB <- subset(swt, Group == "B")

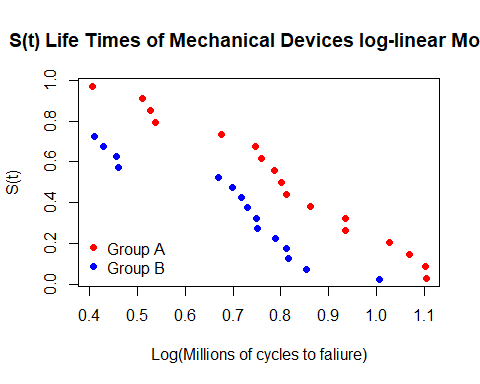
### Plotting position Calculations

j <- seq(1,length(GA$time))  
pA <- 1 - ((j-0.5)/length(GA$time))  
  
j <- seq(1,length(GB$time))  
pB <- 1 - ((j-0.5)/length(GB$time))  
  
inv\_phiA <- qnorm(pA)  
inv\_phiB <- qnorm(pB)

### Diagnostic 1 graph

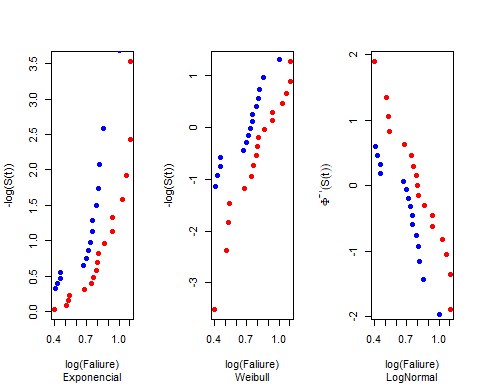
For the accelerated models we must calculate the empirical survival function, will show location shifted versions of the baseline survival function when an accelerated lifetimes model is appropriate.

plot(log(GA$time),pA, main="S(t) Life Times of Mechanical Devices log-linear Model", ylab="S(t)", xlab="Log(Millions of cycles to faliure)", col ="red",pch = 19)  
points(log(GB$time), pB, col ="blue", pch = 19)  
legend("bottomleft", legend=c("Group A", "Group B"), col = c("red","blue"), pch = 19, bty = "n")

 ### Diagnostic 3 graph

Data are projected to find a better model to fit the data

par(mfrow=c(1,3))  
  
plot(log(GA$time), -log(pA), pch=16, xlab="log(Faliure)", sub="Exponencial",ylab="-log(S(t))", col ="red")  
points(log(GB$time), -log(pB),pch=16, col="blue")  
  
plot(log(GA$time), log(-log(pA)), pch=16, xlab="log(Faliure)", sub="Weibull",ylab="-log(S(t))", col ="red")  
points(log(GB$time), log(-log(pB)),pch=16, col="blue")  
  
plot(log(GA$time), inv\_phiA, pch=16, xlab="log(Faliure)", sub="LogNormal",ylab= expression(Phi^-1 \* (S(t))), col ="red")  
points(log(GB$time), inv\_phiB,pch=16, col="blue")



It can be seen that it is the Weibull model, so we proceed to calculate its regressor data

### Regression calculation

reg\_weibullA<-survreg(Surv(GA$time, GA$A)~1, dist='weibull')  
  
reg\_weibullB<-survreg(Surv(GB$time, GB$B)~1, dist='weibull')

### Extraction of residuals

We extract the residuals to visualize the differences between the generated model and the actual model value of S(t).

resid.devianceA <- residuals(reg\_weibullA, type="deviance")  
resid.devianceB <- residuals(reg\_weibullB, type="deviance")

### Visualization of residuals

The residuals are extracted and the quartiles are calculated. As requested in the assignment

par(mfrow=c(1,2))  
boxplot(resid.devianceA, col ="red")  
title("Deviance residuals\nversus Group A", cex.main = 0.7, xlab ="Group A")  
boxplot(resid.devianceB, col ="blue")  
title("Deviance residuals\nversus Group B", cex.main = 0.7, xlab ="Group B")

