HW1

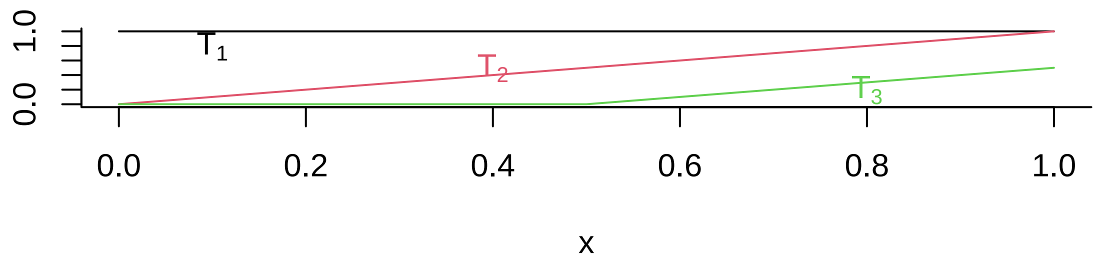
ASG

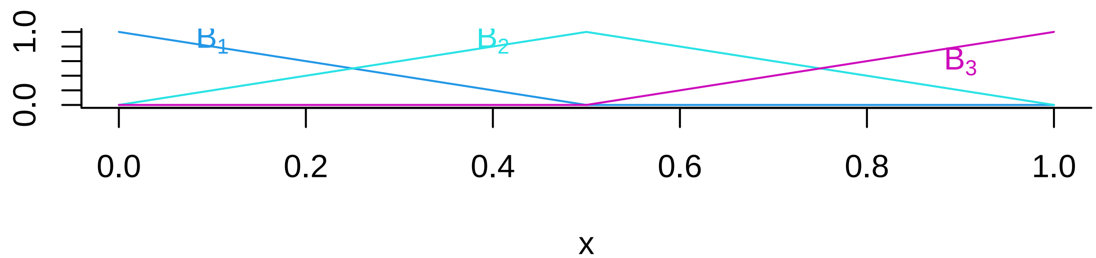
2023-03-15

2.1

a

```
ng <- 101 ; xg <- seq(0,1,length = ng)
T1g <- rep(1,ng) ; T2g <- xg ;
T3g <- (xg - 0.5)*(xg - 0.5>0)
B1g <- (1 - 2*xg)*(1 - 2*xg>0)
B2g <- 1 - abs(2*xg - 1) ; B3g <- 2*T3g
par(mfrow = c(2,1))
plot(0,type = "n",xlim = c(0,1),ylim = c(0,1),xlab = "x",ylab = "",bty = "l")
lines(xg,T1g,col = 1) ; lines(xg,T2g,col = 2)
lines(xg,T3g,col = 3)
text(0.1,0.8,expression(T[1]),col = 1)
text(0.4,0.5,expression(T[2]),col = 2)
text(0.8,0.2,expression(T[3]),col = 3)
plot(0,type = "n",xlim = c(0,1),ylim = c(0,1),xlab = "x",ylab = "",bty = "l")
lines(xg,B1g,col = 4) ; lines(xg,B2g,col = 5)
lines(xg,B3g,col = 6)
text(0.1,0.9,expression(B[1]),col = 4)
text(0.4,0.9,expression(B[2]),col = 5)
text(0.9,0.6,expression(B[3]),col = 6)
```





b

```
B1gh<-T1g-2*T2g+2*T3g
B2gh<-2*T2g-4*T3g
B3gh<-2*T3g

all.equal(B1gh,B1g)
```

```
## [1] TRUE
```

```
all.equal(B2gh,B2g)
```

```
## [1] TRUE
```

```
all.equal(B3gh,B3g)
```

```
## [1] TRUE
```

```
B4g<-B1gh+B2gh+B3gh
all.equal(B4g,T1g)
```

```
## [1] TRUE
```

c

```r
L<-rbind(c(1,-2,2),c(0,2,-4),c(0,0,2))
print(L)
```

```
##      [,1] [,2] [,3]
## [1,]    1   -2    2
## [2,]    0    2   -4
## [3,]    0    0    2
```

d

```r
iL<-solve(L)
dL<-det(L)
print(iL)
```

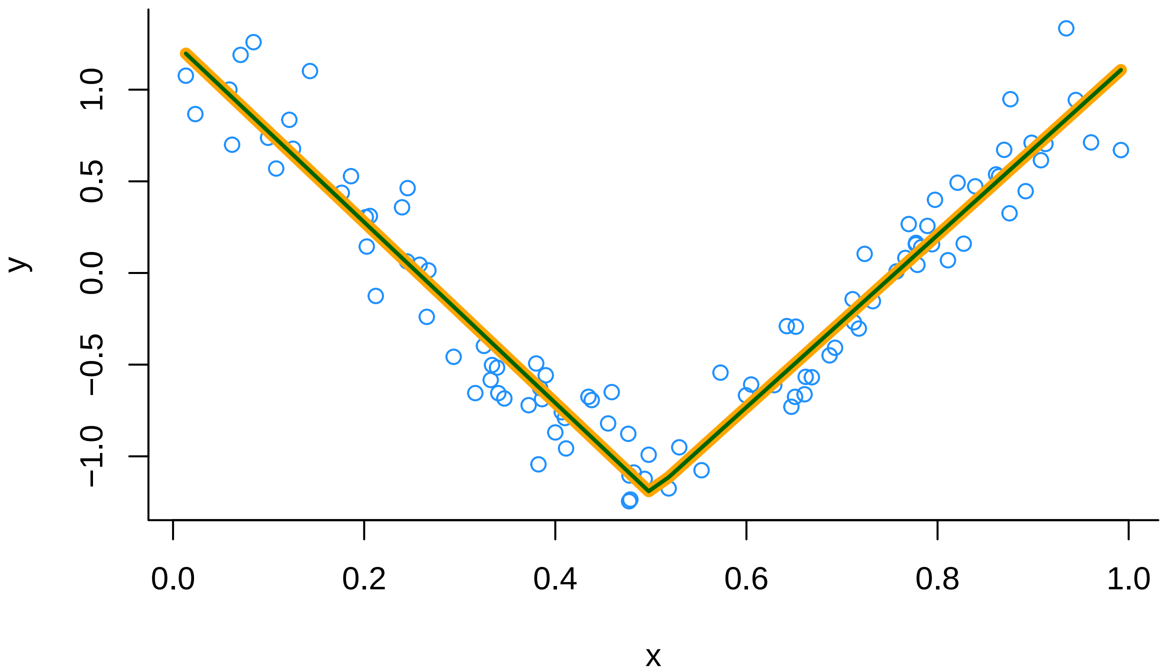
```
##      [,1] [,2] [,3]
## [1,]    1  1.0  1.0
## [2,]    0  0.5  1.0
## [3,]    0  0.0  0.5
```

```r
print(dL)
```

```
## [1] 4
```

e

```r
par(mfrow=c(1,1))
set.seed(1) ; n <- 100 ; x <- sort(runif(100))
y <- cos(2*pi*x) + 0.2*rnorm(n)
plot(x,y,col = "dodgerblue",bty = "l")
XT <- cbind(rep(1,n),x,(2*x - 1)*(2*x - 1>0))
XB <- cbind((1 - 2*x)*(1 - 2*x>0),1 - abs(2*x - 1),
             + (2*x - 1)*(2*x - 1>0))
fitT <- lm(y~-1+XT) ; fitB <- lm(y~-1+XB)
lines(x,fitted(fitT),col = "orange",lwd = 6)
lines(x,fitted(fitB),col = "darkgreen",lwd = 2)
```



f

The alternate spline bases will give the same mathematical fit is when the column functions of XB are an alternative bases of column functions of XT.

g

```
vB<-vcov(fitB)
vT<-vcov(fitT)
summary(vB)
```

```
##       XB1                  XB2                  XB3
##  Min.   :-0.0010280   Min.   :-0.0010280   Min.   :-0.0009297
##  1st Qu.:-0.0002084   1st Qu.:-0.0009789   1st Qu.:-0.0001593
##  Median : 0.0006111   Median :-0.0009297   Median : 0.0006111
##  Mean   : 0.0010974   Mean   :-0.0001313   Mean   : 0.0009114
##  3rd Qu.: 0.0021602   3rd Qu.: 0.0003171   3rd Qu.: 0.0018320
##  Max.   : 0.0037092   Max.   : 0.0015639   Max.   : 0.0030528
```

```
summary(vT)
```

```
##        XT                  XTx                  XT
##  Min.   :-0.0094744   Min.   :-0.022924   Min.   :-0.022924
##  1st Qu.:-0.0028826   1st Qu.:-0.016199   1st Qu.:-0.008274
```

```
##   Median : 0.0037092   Median :-0.009474   Median : 0.006376
##   Mean   : 0.0002037   Mean   :-0.001027   Mean   : 0.001841
##   3rd Qu.: 0.0050427   3rd Qu.: 0.009921   3rd Qu.: 0.014223
##   Max.   : 0.0063763   Max.   : 0.029316   Max.   : 0.022071
```

The linear B-spline basis has lower covariance. This is desirable because having highly correlated spline basis would not able to explain the variability in dataset and would be redundant basis'. Hence having spline basis with lower covariance will be able to model and capture the variability more efficiently by individually contributing more to model complexity. ### h

```r
B4g<-B1gh+B2gh+B3gh
all.equal(B4g,T1g)
```

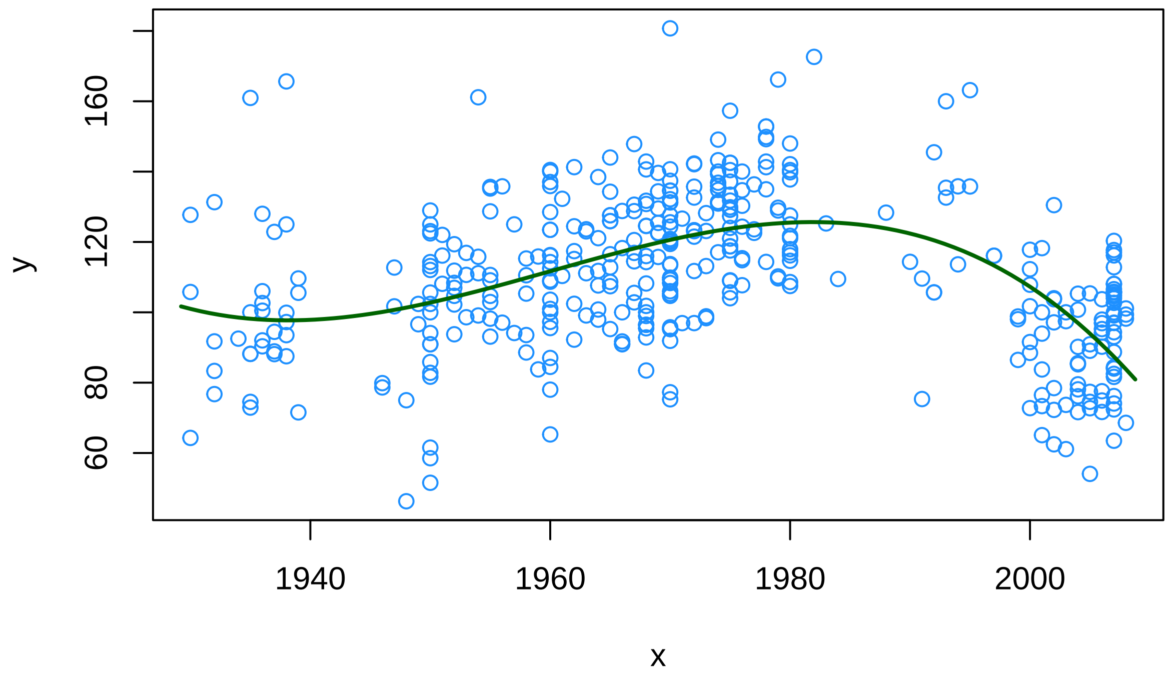
```
## [1] TRUE
```

the plot in 2.1a captures this effect efficiently with each Basis function capturing a certain interval of X. b1 from [0,0.5], b3 from [0.5,1]

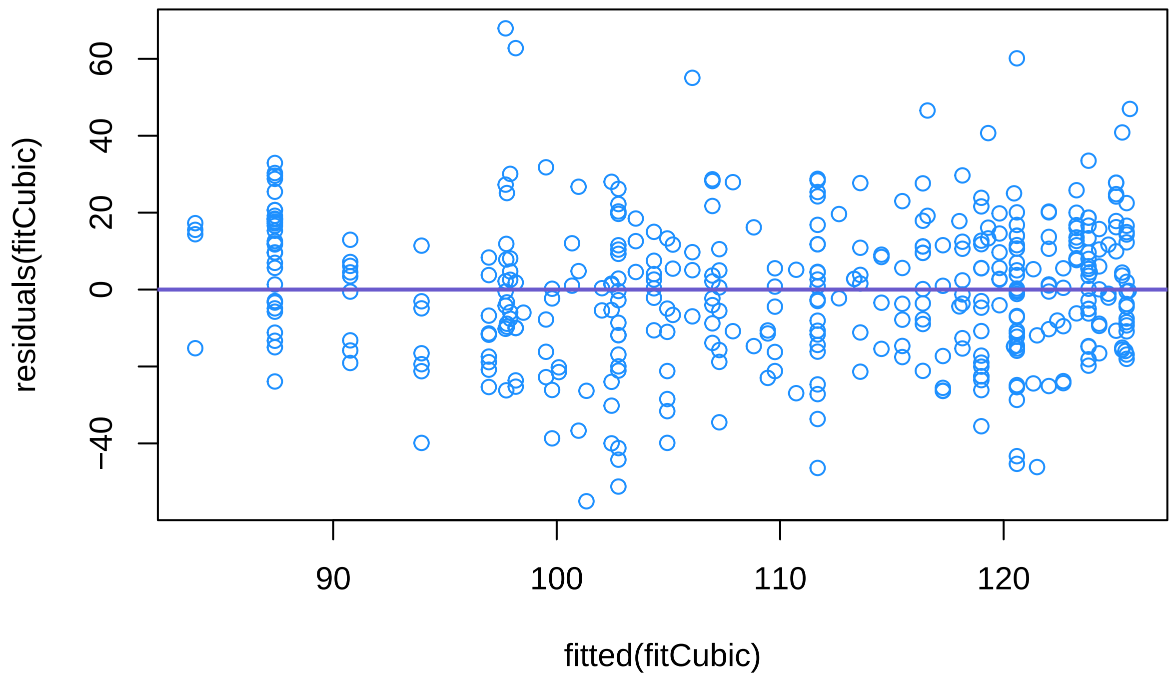
2.2

a

```r
library(HRW); data(WarsawApts)
x <- WarsawApts$construction.date
y <- WarsawApts$areaPerMzloty
fitCubic <- lm(y ~ poly(x,3,raw = TRUE))
ng <- 101 ; xg <- seq(1.01*min(x) - 0.01*max(x),1.01*max(x) - 0.01*min(x),length = ng)
fHatCubicg <- as.vector(cbind(rep(1,ng),xg,xg^2,xg^3)%*%fitCubic$coef)
plot(x,y,col = "dodgerblue")
lines(xg,fHatCubicg,col = "darkgreen",lwd = 2)
```




```r
plot(fitted(fitCubic),residuals(fitCubic),col = "dodgerblue")
abline(0,0,col = "slateblue",lwd = 2)
```

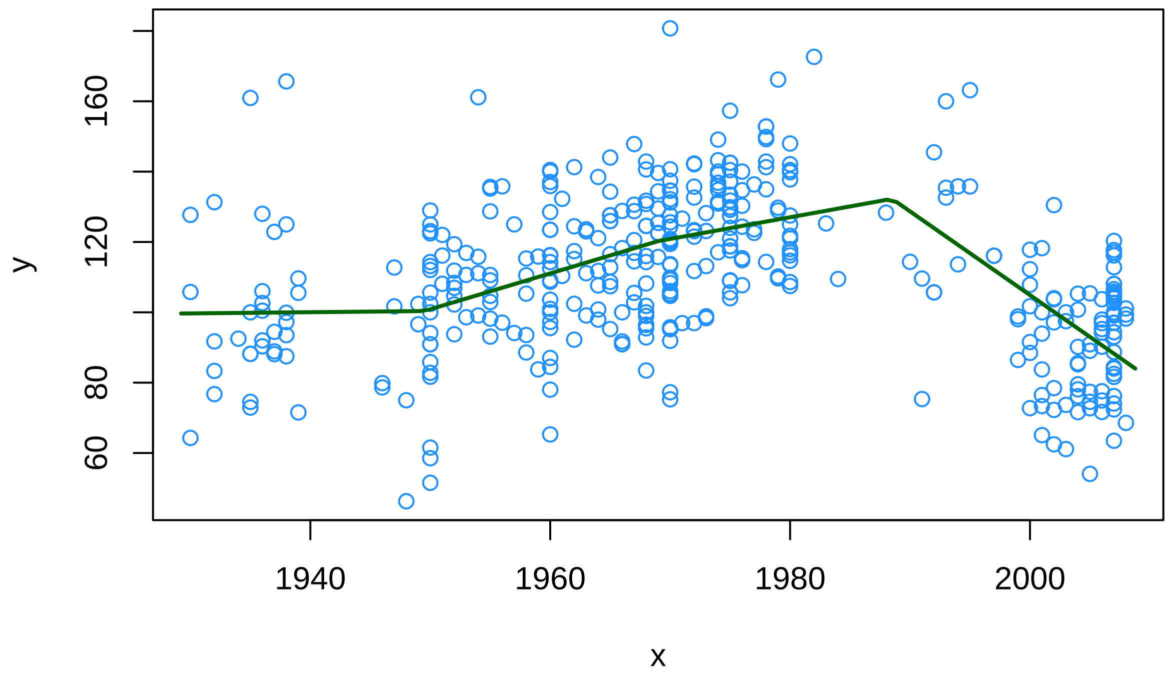


This model is much smoother and comparitively less complicated i.e it is neither highly overfitting nor underfitting. It captures the general trend but using better smoothing parameter we can maybe use fine tune hyperparameterization to overall reduce the loss. But as far as i am concerned this model would be decent to model the relationship but could be much more imporoved. The residual plot shows normal kind of distribution as well but with a more complex model maybe the residuals can be bought down closer to zero. ### b

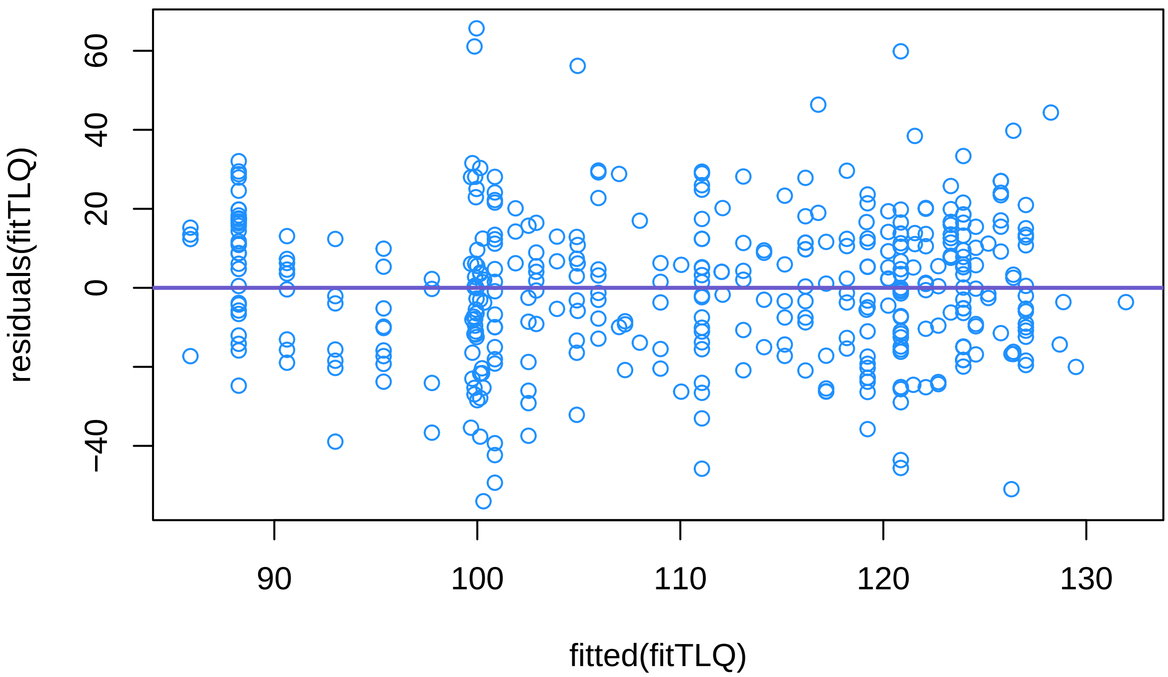
```
trLin <- function(x,kappa) return((x-kappa)*(x>kappa))
```

c

```
knots <- seq(min(x),max(x),length = 5)[-c(1,5)]
X <- cbind(1,x)
for (k in 1:3) X <- cbind(X,trLin(x,knots[k]))
fitTLQ <- lm(y ~ -1 + X)
Xg <- cbind(1,xg)
for (k in 1:3) Xg <- cbind(Xg,trLin(xg,knots[k]))
fHatTLQg <- as.vector(Xg%*%fitTLQ$coef)
plot(x,y,col = "dodgerblue")
lines(xg,fHatTLQg,col = "darkgreen",lwd = 2)
```

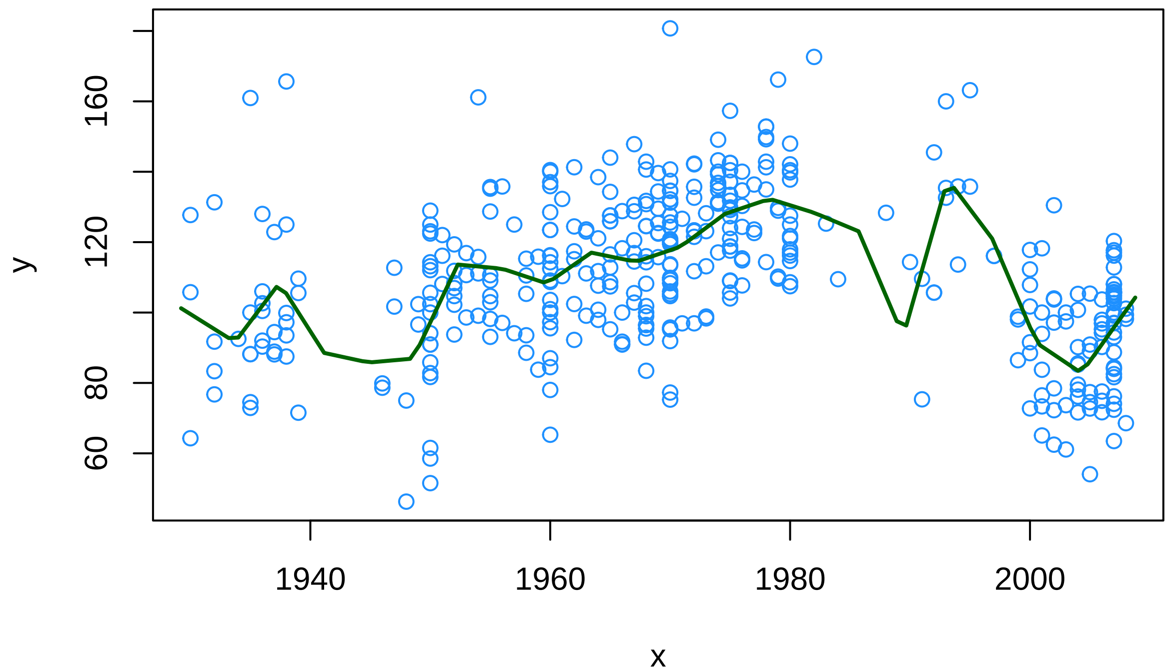



```
plot(fitted(fitTLQ),residuals(fitTLQ),col = "dodgerblue")
abline(0,0,col = "slateblue",lwd = 2)
```

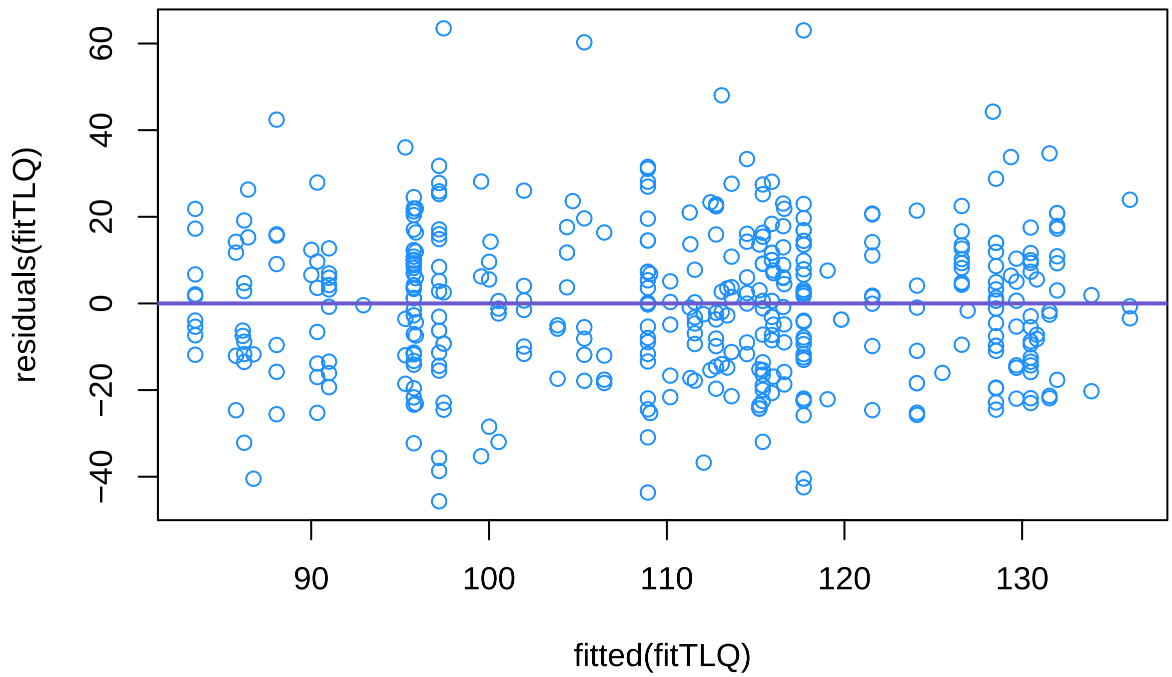


This model also able to generalize pretty well albiet it could be more robust by tuning in more knots for more basis function. The residual plot seems very similar to the previous model and shows that they have similar behaviour. ### d

```
knots <- seq(min(x),max(x),length = 22)[-c(1,22)]
X <- cbind(1,x)
for (k in 1:20) X <- cbind(X,trLin(x,knots[k]))
fitTLQ <- lm(y ~ -1 + X)
Xg <- cbind(1,xg)
for (k in 1:20) Xg <- cbind(Xg,trLin(xg,knots[k]))
fHatTLQg <- as.vector(Xg%*%fitTLQ$coef)
plot(x,y,col = "dodgerblue")
lines(xg,fHatTLQg,col = "darkgreen",lwd = 2)
```




```
plot(fitted(fitTLQ),residuals(fitTLQ),col = "dodgerblue")
abline(0,0,col = "slateblue",lwd = 2)
```

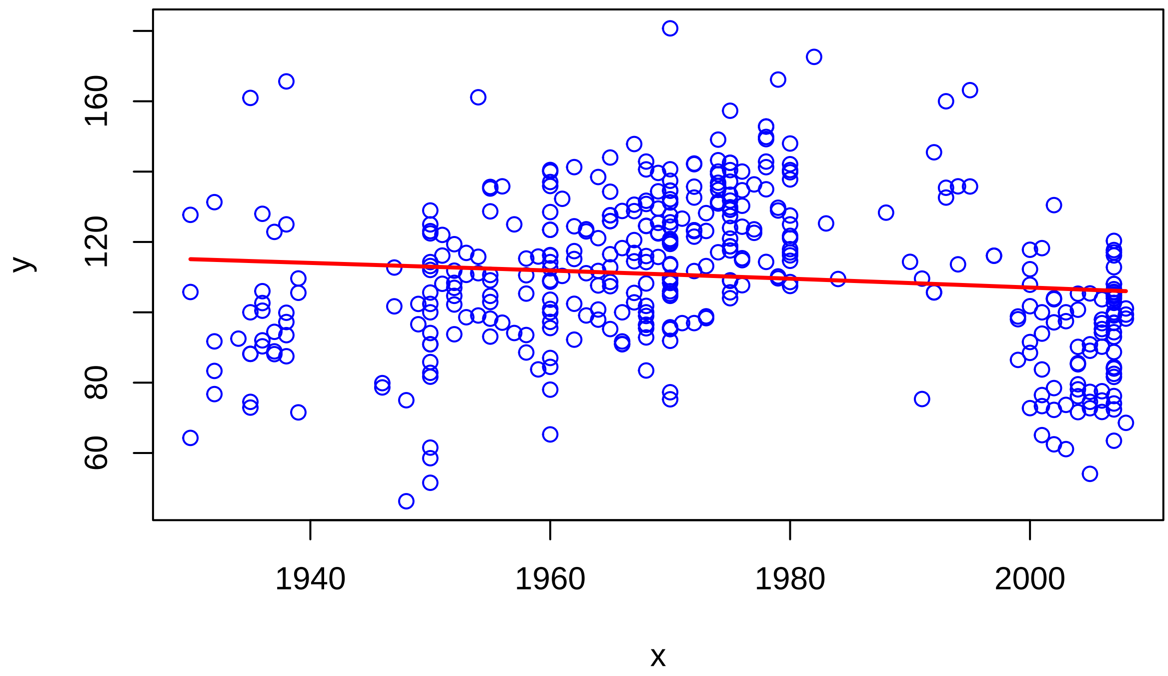


e

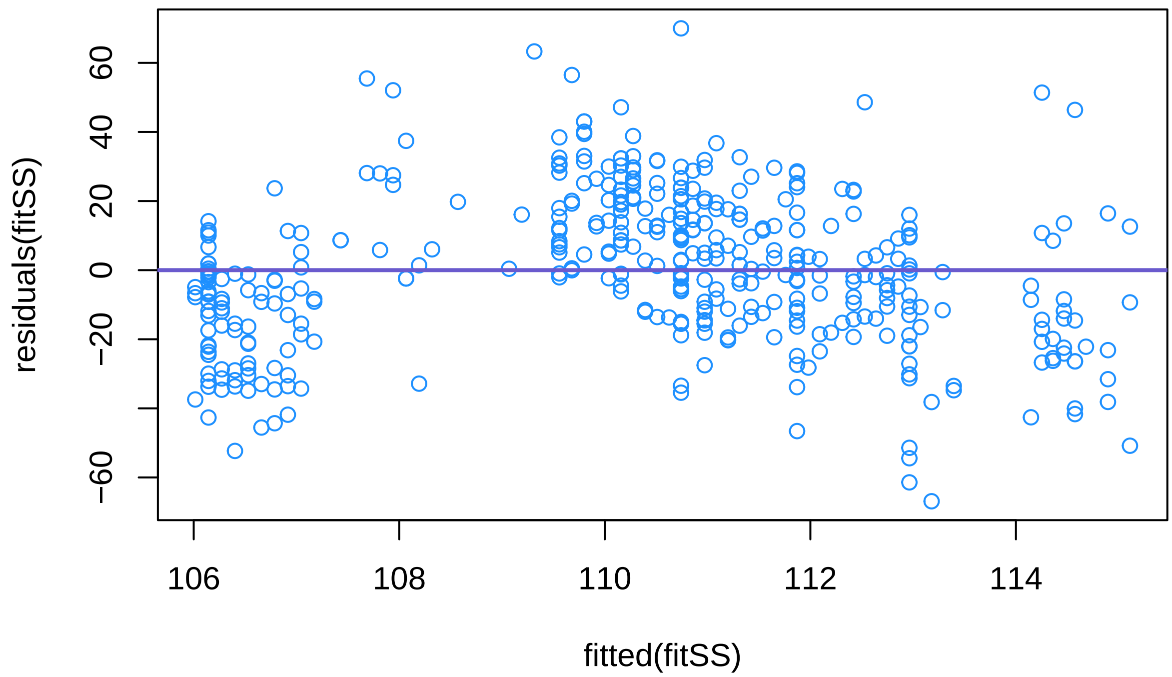
```
fitSS <- smooth.spline(x,y,lambda=100)
fitSS
```

```
## Call:
## smooth.spline(x = x, y = y, lambda = 100)
##
## Smoothing Parameter  spar= NA  lambda= 100
## Equivalent Degrees of Freedom (Df): 2.010618
## Penalized Criterion (RSS): 92877.62
## GCV: 484.8038
```

```
plot(x,y,col="blue")
lines(fitSS,lwd=2,col="red")
```




```
plot(fitted(fitSS),residuals(fitSS),col = "dodgerblue")
abline(0,0,col = "slateblue",lwd = 2)
```



60
40
20
0
−20
−60
residuals(fitSS)
106
108
110
112
114
fitted(fitSS)