Non-linear regression for the motorcycle data

Phillip Yates - MAT 350 & MAT 427

10/29/2020

This is a useful reference in its entirety so storing it for posterity.

First we will load the `tidyverse` library for data reshaping and visualization.

```
library(tidyverse)
```

In this example X is the time since the motorcycle crash and Y is the acceleration of the driver's head. We will fit the semiparametric model

$$Y_i \sim \text{Normal}\left[g\left(X_i\right), \sigma^2\right]$$

where the mean function g is assumed to have spline basis representation

$$g(X) = \mu + \sum_{j=1}^{J} B_j(X)\beta_j.$$

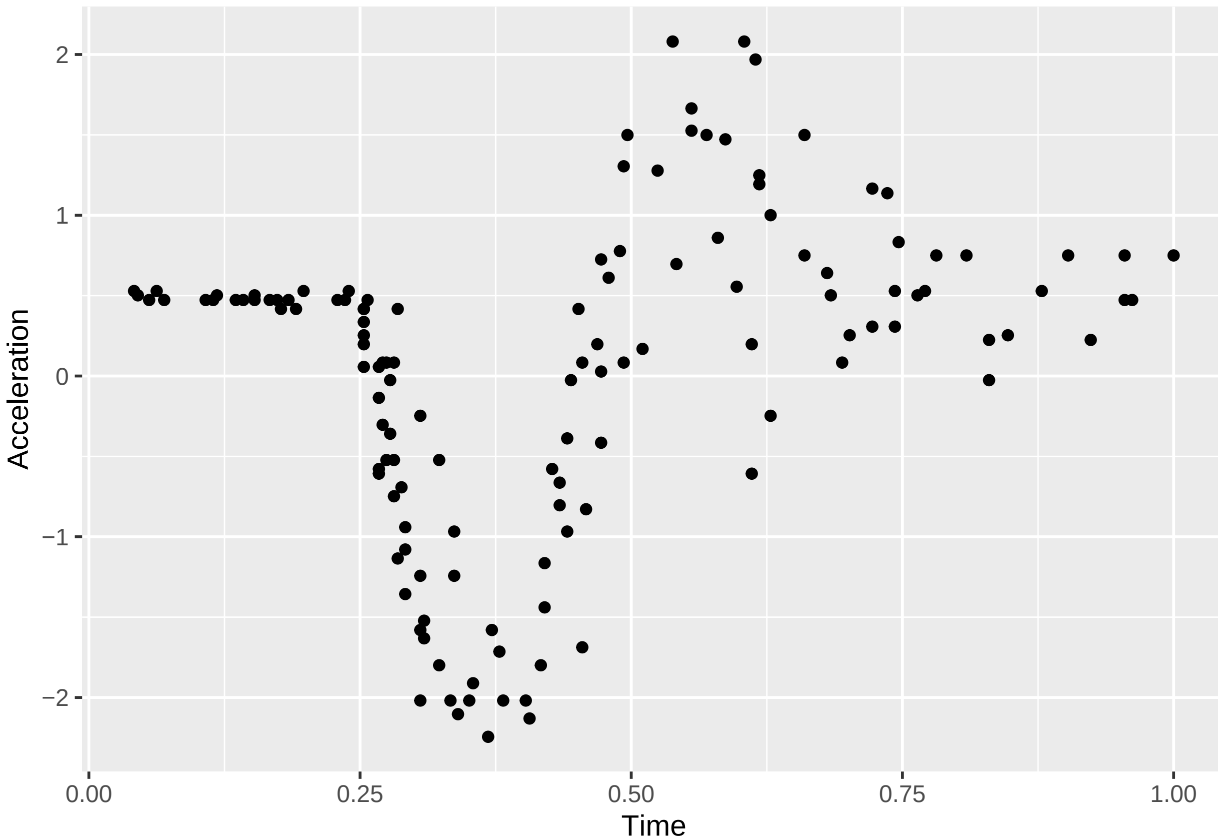
The remaining parameters have uninformative priors: $\mu \sim \text{Normal}(0, 100)$, $\beta_j \sim \text{Normal}\left(0, \tau^2\right)$ and $\sigma^2, \tau^2 \sim \text{InvGamma}(0.1, 0.1)$.

Load and plot the motorcycle data

```
library(MASS)

Y <- mcycle$accel
X <- mcycle$times

Y <- (Y-mean(Y))/sd(Y)
X <- X/max(X)

n <- length(Y)
n
```

```
## [1] 133
```

```
ggplot(NULL,aes(x=X,y=Y))+geom_point()+labs(x="Time",y="Acceleration")
```

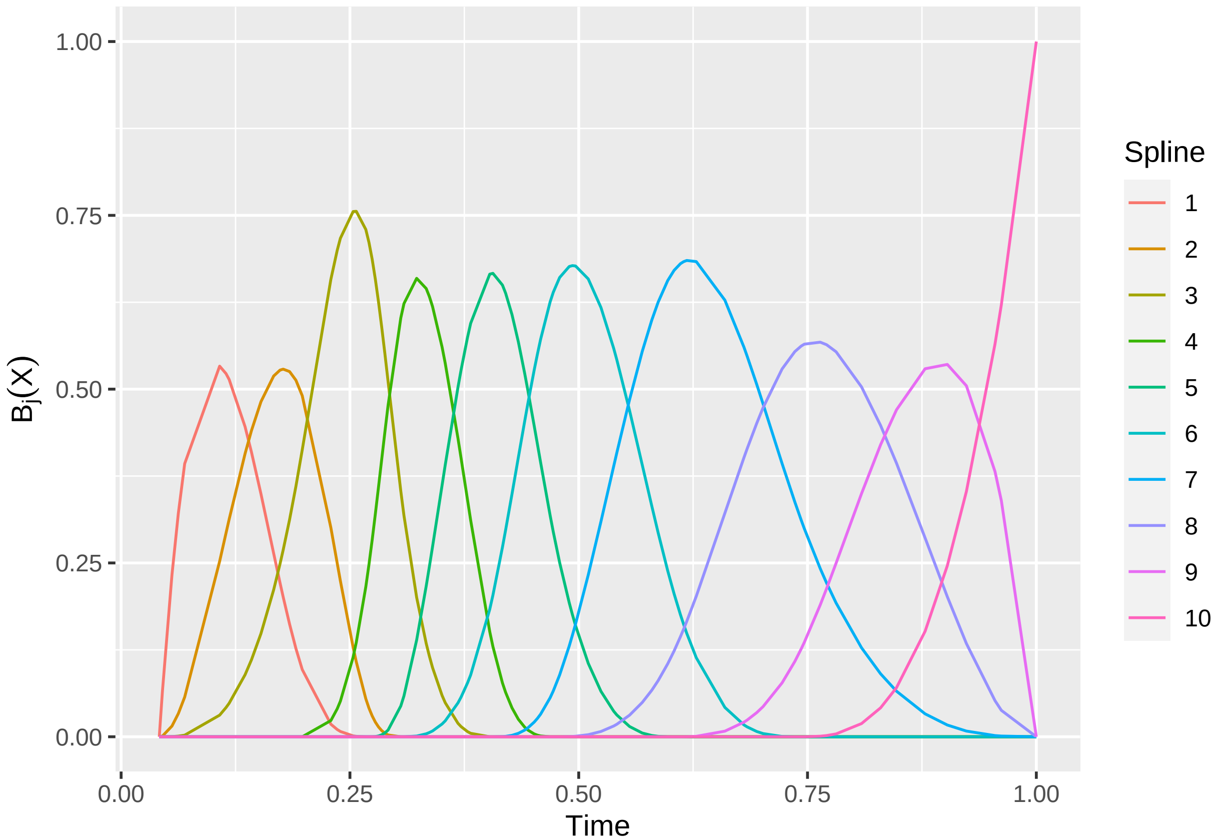


Set up a spline basis expansion

```
library(splines)

# Number of basis functions
J <- 10

# Specify the basis functions -- default is a cubic spline
B <- bs(X,J)
dim(B)
```

```
## [1] 133  10
```

```
new.B <- cbind(X,B)
B.data <- as.data.frame(new.B) %>%
  pivot_longer(-X,names_to="Spline",values_to="basis") %>%
  mutate(Spline=factor(Spline,levels=c("1","2","3","4","5","6","7","8","9","10")))
ggplot(B.data,aes(x=X,y=basis,color=Spline))+geom_line()+
  labs(x="Time",y=expression(B[j](X)))
```



Fit the model

```
library(rjags)

Moto_model <- "model{
  # Likelihood
    for(i in 1:n){
      Y[i] ~ dnorm(mean[i],taue)
      mean[i] <- mu+inprod(B[i,],beta[])
    }

  # Prior
    mu ~ dnorm(0,0.01)
    taue ~ dgamma(0.1,0.1)
    for(j in 1:J){
      beta[j] ~ dnorm(0,taub)
    }
    taub ~ dgamma(0.1,0.1)
}"

dat <- list(Y=Y,n=n,B=B,J=J)
init <- list(mu=mean(Y),beta=rep(0,J),taue=1/var(Y))
model <- jags.model(textConnection(Moto_model),inits=init,data=dat,quiet=TRUE)
update(model,10000,progress.bar="none")
```

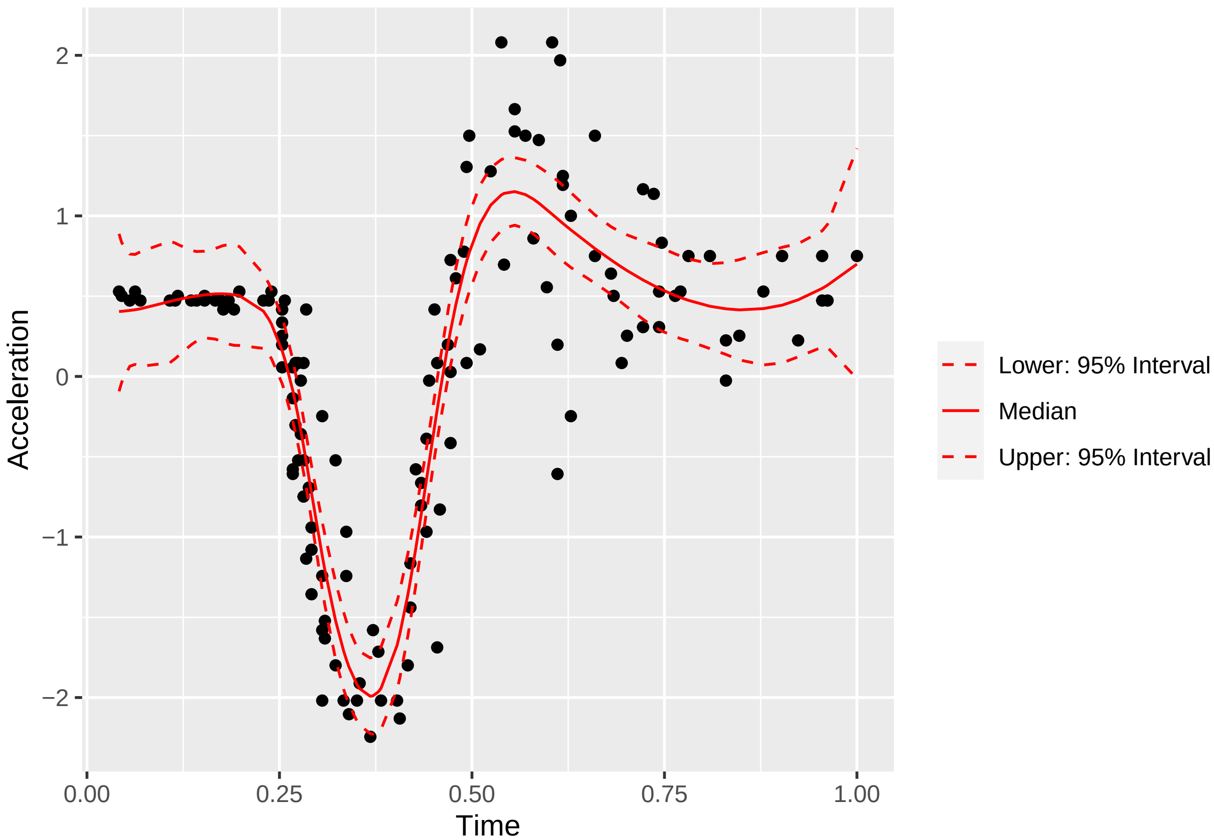
```r
samp <- coda.samples(model,variable.names=c("mean"),n.iter=20000,progress.bar="none")
```

Plot the fixed curve, $g(X)$

```r
sum <- summary(samp)
names(sum)
```

```
## [1] "statistics" "quantiles"  "start"      "end"        "thin"
## [6] "nchain"
```

```r
q <- sum$quantiles
q.small <- cbind(X,q[,c(1,3,5)])
q.data <- as.data.frame(q.small) %>%
  pivot_longer(-X,names_to="Method",values_to="estimate") %>%
  mutate(Method=ifelse(Method=="50%","Median",
                ifelse(Method=="2.5%","Lower: 95% Interval","Upper: 95% Interval")))

ggplot(NULL,aes(x=X,y=Y))+geom_point()+
  geom_line(data=q.data,aes(x=X,y=estimate,linetype=Method),color="red")+
  scale_linetype_manual(values=c(2,1,2),
                        labels=c("Lower: 95% Interval","Median","Upper: 95% Interval"))+labs(x="Time",y=
```



Summary: The mean trend seems to fit the data well; however, the variance of the observations around the mean varies with X.

Heteroskedastic model

The variance is small for X near zero and increases with X. To account for this, we allow the log of the variance to vary with X following a second spline basis expansion:

$$Y_i \sim \text{Normal}\left[g\left(X_i\right), \sigma^2\left(X_i\right)\right]$$

where $g(X) = \mu + \sum_{i=1}^{J} B_j(X)\beta_j$ is modeled as above and

$$\log\left[\sigma^2(X)\right] = \mu_2 + \sum_{i=1}^{J} B_j(X)\alpha_j.$$

The parameters have uninformative priors $\mu_k \sim \text{Normal}(0, 100)$, $\beta_j \sim \text{Normal}\left(0, \sigma_b^2\right)$, $\alpha_j \sim \text{Normal}\left(0, \sigma_a^2\right)$, and $\sigma_a^2, \sigma_b^2 \sim \text{InvGamma}(0.1, 0.1)$.

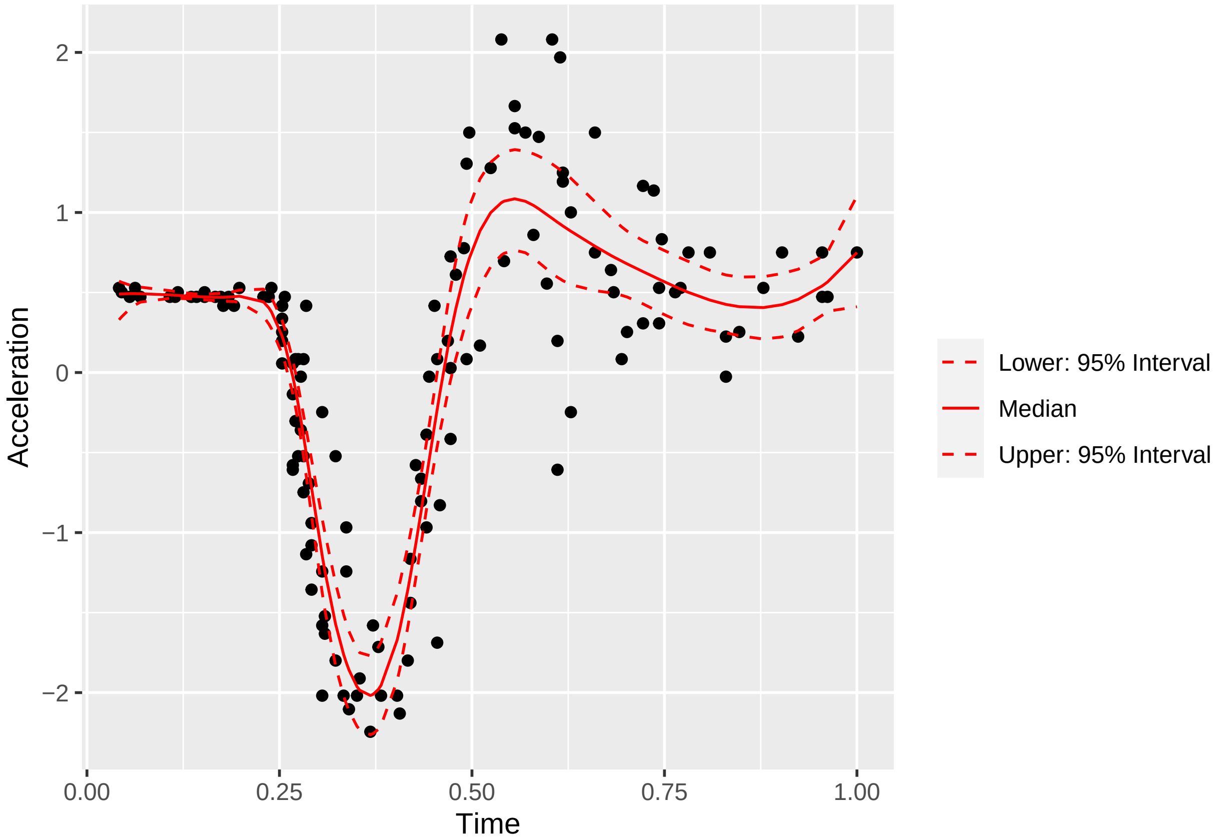
```
moto_model2 <- "model{
  # Likelihood
    for(i in 1:n){
      Y[i] ~ dnorm(mean[i],inv_var[i])
      mean[i] <- mu1+inprod(B[i,],beta[])
      inv_var[i] <- 1/sig2[i]
      log(sig2[i]) <- mu2+inprod(B[i,],alpha[])
    }

  # Prior
    mu1 ~ dnorm(0,0.01)
    mu2 ~ dnorm(0,0.01)
    for(j in 1:J){
      beta[j] ~ dnorm(0,taub)
      alpha[j] ~ dnorm(0,taua)
    }
    taua ~ dgamma(0.1,0.1)
    taub ~ dgamma(0.1,0.1)

  # Prediction intervals
    for(i in 1:n){
      low[i] <- mean[i]-1.96*sqrt(sig2[i])
      high[i] <- mean[i]+1.96*sqrt(sig2[i])
    }
}"
```

Fit the model

```
dat <- list(Y=Y,n=n,B=B,J=J)
init <- list(mu1=mean(Y),beta=rep(0,J),mu2=log(var(Y)),alpha=rep(0,J))
model <- jags.model(textConnection(moto_model2),inits=init,data=dat,quiet=TRUE)

update(model,10000,progress.bar="none")

samp2 <- coda.samples(model,variable.names=c("mean","sig2","low","high"),
                      n.iter=20000,progress.bar="none")
```

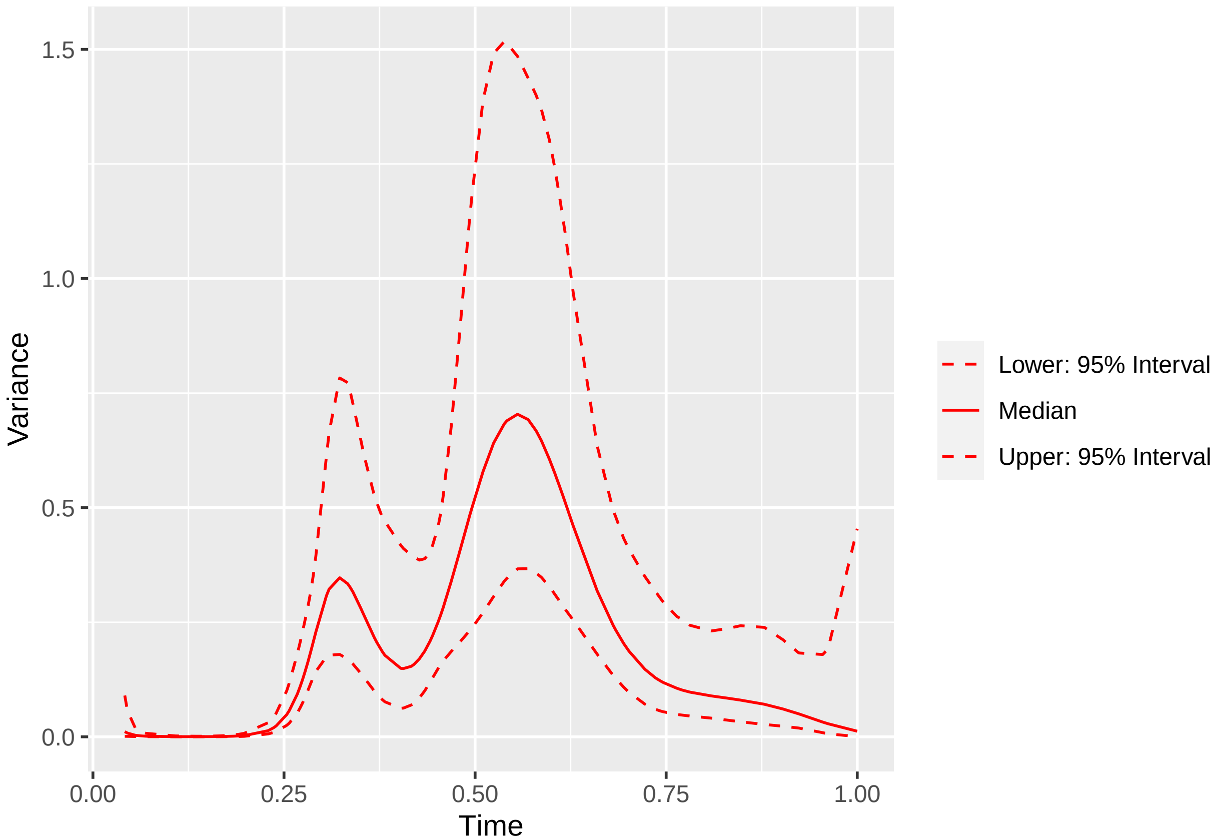
Plot the fixed curve, $g(X)$

```
q2 <- summary(samp2)$quantiles
q2.small.mean <- cbind(X,q2[1:n+2*n,c(1,3,5)])

q2.data.mean <- as.data.frame(q2.small.mean) %>%
  pivot_longer(-X,names_to="Method",values_to="estimate") %>%
  mutate(Method=ifelse(Method=="50%","Median",
              ifelse(Method=="2.5%","Lower: 95% Interval","Upper: 95% Interval")))

ggplot(NULL,aes(x=X,y=Y))+geom_point()+
  geom_line(data=q2.data.mean,aes(x=X,y=estimate,linetype=Method),color="red")+
  scale_linetype_manual(values=c(2,1,2),
                        labels=c("Lower: 95% Interval","Median","Upper: 95% Interval"))+labs(x="Time",y=
```



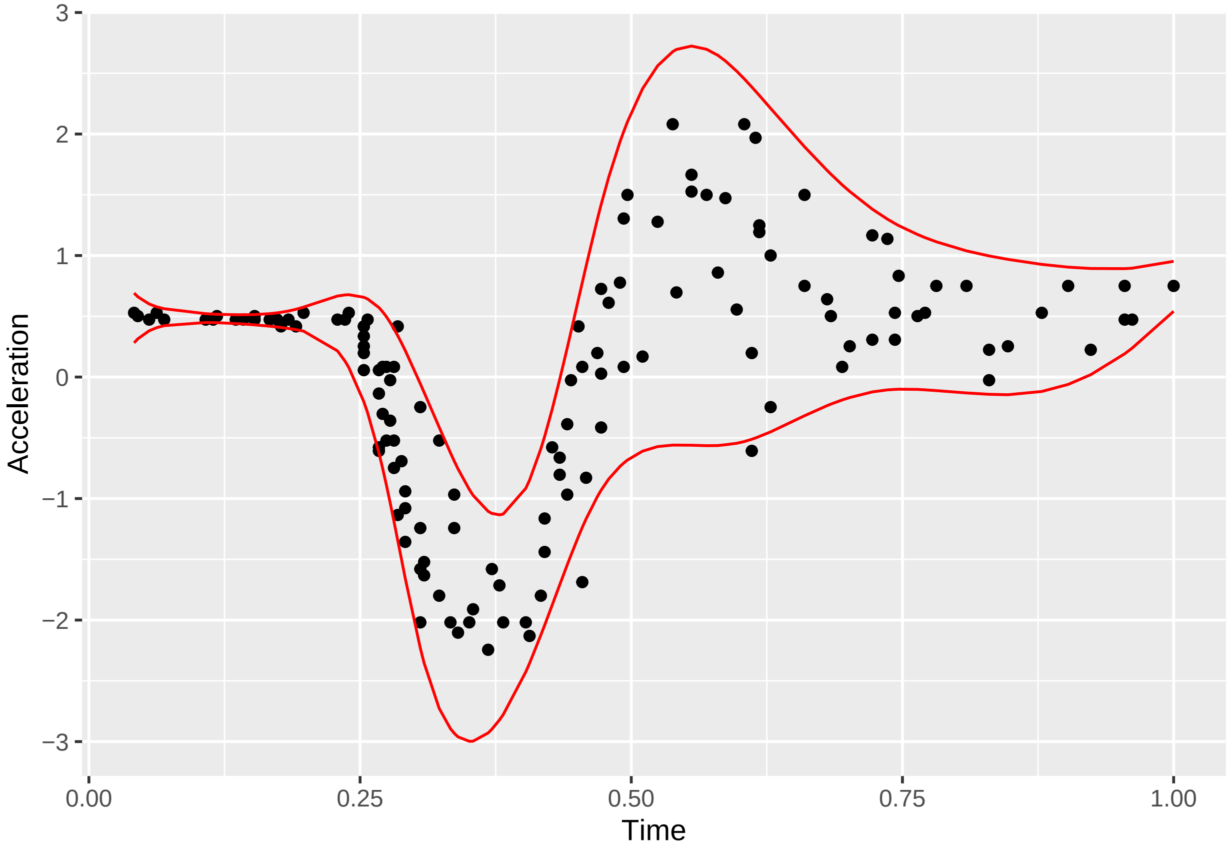
We can also plot the fitted variance function.

```
q2.small.sig <- cbind(X,q2[1:n+3*n,c(1,3,5)])

q2.data.sig <- as.data.frame(q2.small.sig) %>%
  pivot_longer(-X,names_to="Method",values_to="estimate") %>%
  mutate(Method=ifelse(Method=="50%","Median",
              ifelse(Method=="2.5%","Lower: 95% Interval","Upper: 95% Interval")))

ggplot(q2.data.sig,aes(x=X,y=estimate,linetype=Method))+geom_line(color="red")+
  scale_linetype_manual(values=c(2,1,2),
                        labels=c("Lower: 95% Interval","Median","Upper: 95% Interval"))+labs(x="Time",y=
```



Finally we can plot the prediction intervals.

```
q2.small.low <- cbind(X,q2[1:n+1*n,3])
q2.small.high <- cbind(X,q2[1:n+0*n,3])

ggplot(NULL,aes(x=X,y=Y))+geom_point()+
  geom_line(data=as.data.frame(q2.small.low),aes(x=X,y=V2),color="red")+
  geom_line(data=as.data.frame(q2.small.high),aes(x=X,y=V2),color="red")+
  labs(x="Time",y="Acceleration")
```



Acceleration
Time
3
2
1
0
−1
−2
−3
0.00
0.25
0.50
0.75
1.00