Non-linear regression for the motorcycle data

Nonparametric regression models

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

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

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

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

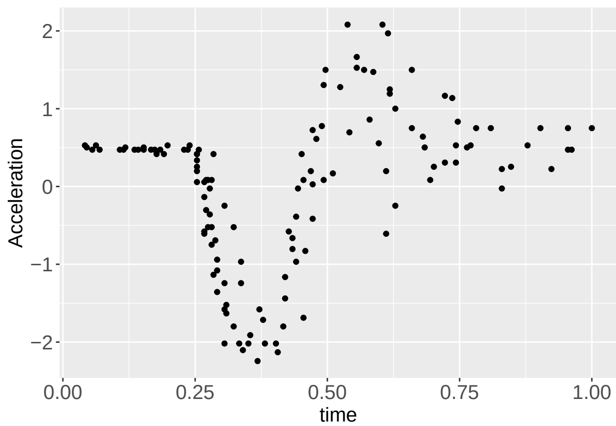
The remaining parameters have uninformative priors: $\mu \sim \text{Normal}(0, 100)$, $\beta_j \sim \text{Normal}(0, \sigma^2\tau^2)$, and $\sigma^2, \tau^2 \sim \text{InvGamma}(0.1, 0.1)$.

Load and plot the motorcylce data

```
 library(MASS)
library(ggplot2)
 Y = mcycle$accel
 X = mcycle$times
 Y = (Y-mean(Y))/sd(Y)
 X = X/max(X)
 n = length(Y)
 n
```

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

```
# Create a data frame for ggplot
data =data.frame(time = X, Acceleration = Y)
ggplot(data, aes(x = time, y = Acceleration)) +
  geom_point() +
  labs(x = "time", y = "Acceleration") +
  theme(
    axis.title = element_text(size = 15),
    axis.text = element_text(size = 15)
  )
```



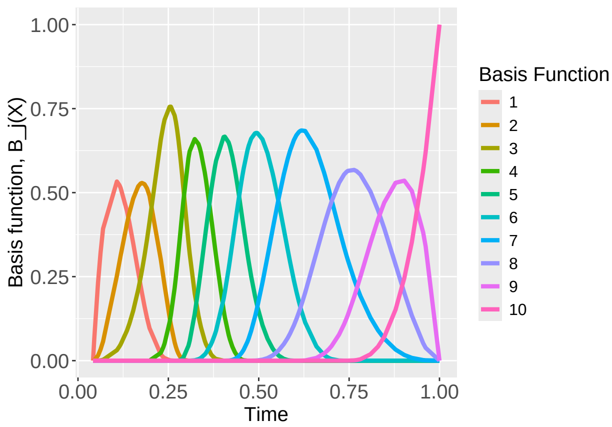
Spline basis expansion

```
library(splines)
J = 10       # Number of basis functions
B = bs(X,J)  # Specify the basis functions
dim(B)
```

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

```
# Convert the basis matrix to a data frame for ggplot
basis_df <- data.frame(Time = rep(X, J),
                       Basis = as.vector(B),
                       Function = factor(rep(1:J, each = length(X))))


ggplot(basis_df, aes(x = Time, y = Basis, color = Function)) +
  geom_line(size = 1.5) +
  labs(x = "Time", y = "Basis function, B_j(X)") +
  theme(
    axis.title = element_text(size = 15),
    axis.text = element_text(size = 15),
    legend.title = element_text(size = 15),
    legend.text = element_text(size = 12)
  ) +
  guides(color = guide_legend(title = "Basis Function"))
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```r
library(rjags)
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.1
```

```
## Loaded modules: basemod,bugs
```

```r
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,taue*taub)
   }
   taub ~ dgamma(0.1,0.1)

  }"
```

Fit the model

```r
library(rjags)
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")
```

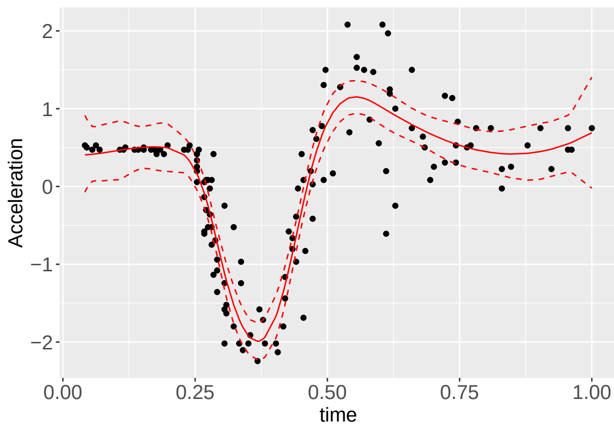
Plot for the fixed curve, g(X)

```r
sum = summary(samp)
names(sum)
```

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

```r
q = sum$quantiles

q = data.frame(
  lower = q[,1],
  median =  q[,3],
  upper =  q[,5]
)

data <- data.frame(time = X, Acceleration = Y, lower = q$lower, median = q$median, upper = q$upper)
ggplot(data, aes(x = time, y = Acceleration)) +
  geom_point() +
  geom_line(aes(y = median), color = "red", linetype = "solid") +
  geom_line(aes(y = lower), color = "red", linetype = "dashed") +
  geom_line(aes(y = upper), color = "red", linetype = "dashed") +
  labs(x = "time", y = "Acceleration") +
  theme(
    axis.title = element_text(size = 15),
    axis.text = element_text(size = 15)
  ) +
  scale_color_manual(values = c("Median" = "red", "95% interval" = "red")) +
  guides(color = guide_legend(override.aes = list(linetype = c("solid", "dashed")))) +
  theme(legend.position = "bottomright", legend.background = element_rect(fill = "white")) +
  labs(color = "Legend") +
  theme(legend.text = element_text(size = 12), legend.title = element_text(size = 15))
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```



Conclusion: The mean trend seems to fit the data well. Nonetheless, the spread 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:

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

where $g(X) = \mu + \sum_{j=1}^{J} B_j(X)\beta_j$ s modeled as above and $log[\sigma^2(X)] = \mu_2 + \sum_{j=1}^{J} B_j(X)\alpha_j$..

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

```
library(rjags)
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

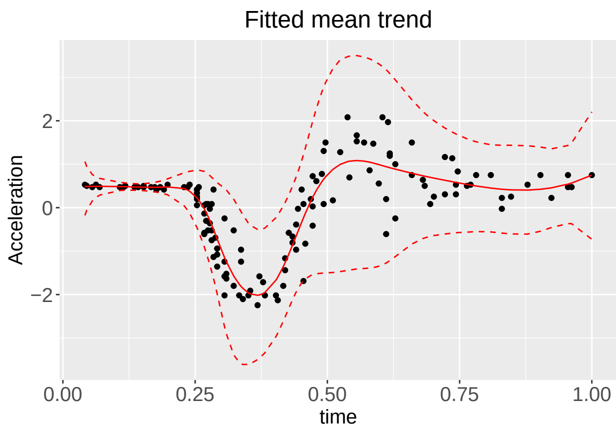
```
library(rjags)
dat    = list(Y=Y,n=n,B=B,J=J)
init   = list(mu1=mean(Y),beta=rep(0,J),
           mu2=log(var(Y)),alpha=rep(0,J))
model = jags.model(textConnection(Moto_model2),
               inits=init,data = dat, quiet=TRUE)

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)

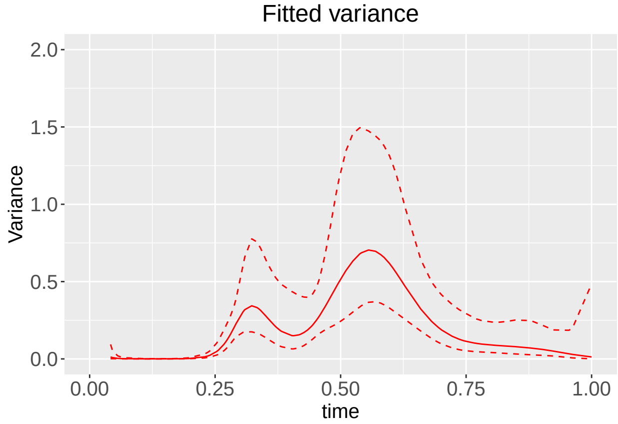
```
q2 = summary(samp2)$quantiles
 high = q2[1:n+0*n,]
 low  = q2[1:n+1*n,]
 mean = q2[1:n+2*n,]
 sig2 = q2[1:n+3*n,]


data = data.frame(
time = X,
Acceleration = Y,
lower = low[,1],
median = mean[,3],
upper = high [,5]
)

ggplot(data, aes(x = time, y = Acceleration)) +
  geom_point() +
  geom_line(aes(y = median), color = "red", linetype = "solid") +
  geom_line(aes(y = lower), color = "red", linetype = "dashed") +
  geom_line(aes(y = upper), color = "red", linetype = "dashed") +
  labs(x = "time", y = "Acceleration", title = "Fitted mean trend") +
  theme(
    axis.title = element_text(size = 15),
    axis.text = element_text(size = 15),
    plot.title = element_text(size = 18, hjust = 0.5)
  ) +
  scale_color_manual(values = c("Median" = "red", "95% interval" = "red")) +
```

```
  guides(color = guide_legend(override.aes = list(linetype = c("solid", "dashed")))) +
  theme(legend.position = "bottomright", legend.background = element_rect(fill = "white")) +
  labs(color = "Legend") +
  theme(legend.text = element_text(size = 12), legend.title = element_text(size = 15))
```

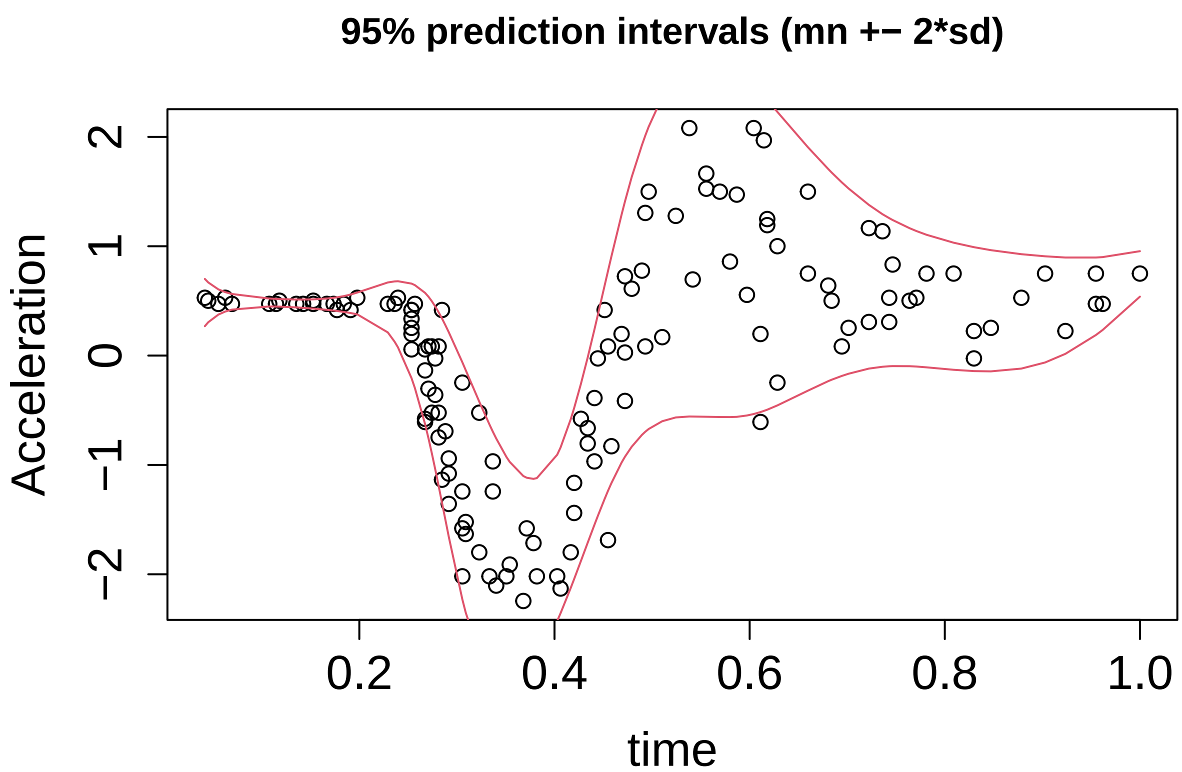
```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s colour values.
```




```
sig2 = data.frame(
  lower = sig2[,1],
  median = sig2[,3],
  upper = sig2[,5]
)

# Create a data frame for plotting
data <- data.frame(
  time = X,
  lower = sig2$lower,
  median = sig2$median,
  upper = sig2$upper
)

ggplot(data, aes(x = time)) +
  geom_line(aes(y = median), color = "red", linetype = "solid") +
  geom_line(aes(y = lower), color = "red", linetype = "dashed") +
  geom_line(aes(y = upper), color = "red", linetype = "dashed") +
  labs(x = "time", y = "Variance", title = "Fitted variance") +
  theme(
    axis.title = element_text(size = 15),
    axis.text = element_text(size = 15),
    plot.title = element_text(size = 18, hjust = 0.5)
  ) +
  scale_y_continuous(limits = c(0, 2)) +
  scale_x_continuous(limits = c(0, 1)) +
  theme(legend.position = "topleft", legend.background = element_rect(fill = "white")) +
  guides(color = guide_legend(title = "Legend", override.aes = list(linetype = c("solid", "dashed")))) +
  theme(legend.text = element_text(size = 12), legend.title = element_text(size = 15))
```




```r
plot(X,Y,xlab="time",ylab="Acceleration",
        main="95% prediction intervals (mn +- 2*sd)",
        cex.lab=1.5,cex.axis=1.5)

lines(X,low[,3],col=2,lty=1)
lines(X,high[,3],col=2,lty=1)
```



Summary

In this analysis, we explored non-linear regression models to fit and understand the motorcycle crash data. The initial approach used a semiparametric model with a spline basis representation for the mean function, capturing the relationship between time and acceleration effectively.

The fitted mean trend was visualized and indicated a good fit to the data. However, it was observed that the variance of the acceleration data around the mean was not constant and varied with time, suggesting heteroskedasticity. This led to the development of a more complex heteroskedastic model, which allowed the variance to change with time, modeled using a second spline basis expansion.

The heteroskedastic model provided a better fit by accounting for the varying variance, which was small near zero and increased with time. The prediction intervals and fitted mean trends were plotted, showcasing the improved model's ability to capture the underlying patterns in the data. Overall, the non-linear regression approaches effectively modeled the complex dynamics of the motorcycle crash data, with the heteroskedastic model offering a more nuanced understanding of the variance behavior over time.