STAT547 Homework 3

Xingche Guo

10/12/2019

Problem 1

$B(t) = W(t) - tW(1)$, $W(\cdot) \sim \mathcal{GP}(0, \min(s,t))$, also:

$$\text{Cov}(B(s), B(t)) = \text{Cov}(W(s) - sW(1), W(t) - tW(1)) = \min(s,t) - st - st + st = \min(s,t) - st. \tag{1}$$

Therefore, $B(\cdot) \sim \mathcal{GP}(0, \min(s,t) - st)$. We can find the eigenvalue and eigenfunction pairs of $B(t)$ by solving:

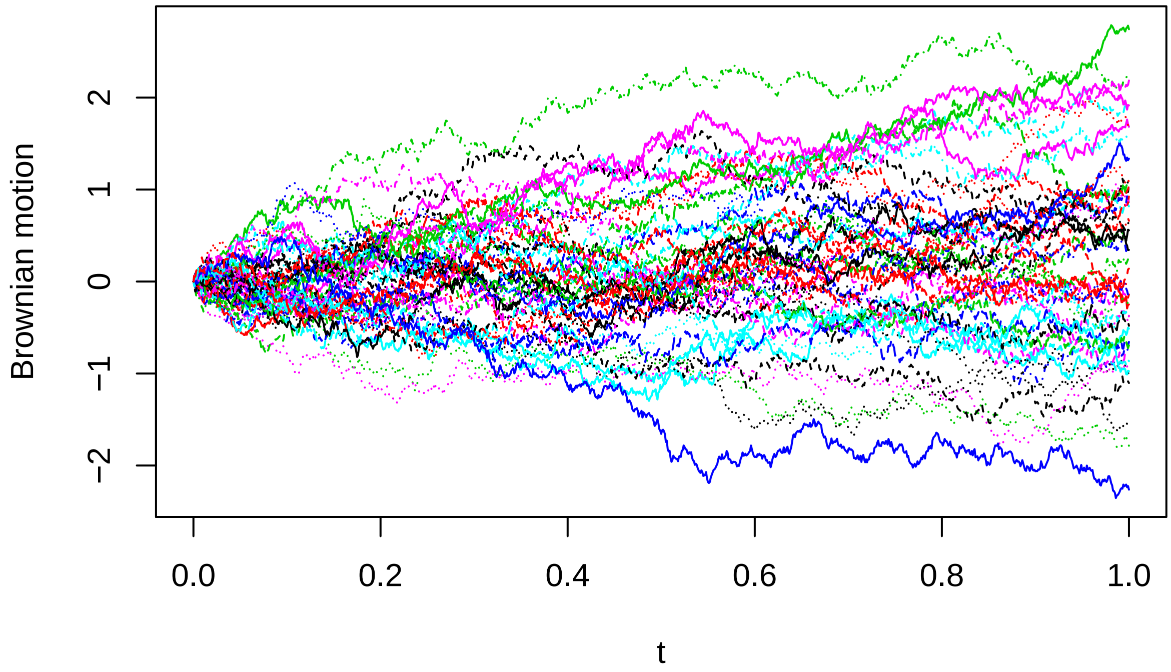
$$\begin{aligned}
&\int_0^1 (\min(s,t) - st)e(s)ds = \lambda e(t) \\
\Longrightarrow\ &(1-t)\int_0^t se(s)ds + t\int_t^1 (1-s)e(s)ds = \lambda e(t) \\
\Longrightarrow\ &-\int_0^t se(s)ds + \int_t^1 (1-s)e(s)ds = \lambda e^{'}(t) \\
\Longrightarrow\ &-e(t) = \lambda e^{''}(t) \\
\Longrightarrow\ &e(t) = A\sin(t/\sqrt{\lambda}) + B\cos(t/\sqrt{\lambda})
\end{aligned}$$

By $B(0) = B(1) = 0$, we have $e(0) = e(1) = 0$. Also using the condition that $\int_0^1 e^2(t)dt = 1$, therefore $A = \sqrt{2}$, $B = 0$, $\lambda = 1/(k^2\pi^2)$, $k = 1, 2, \ldots$. Thus, the K-L expansion of $B(t)$ has the form:

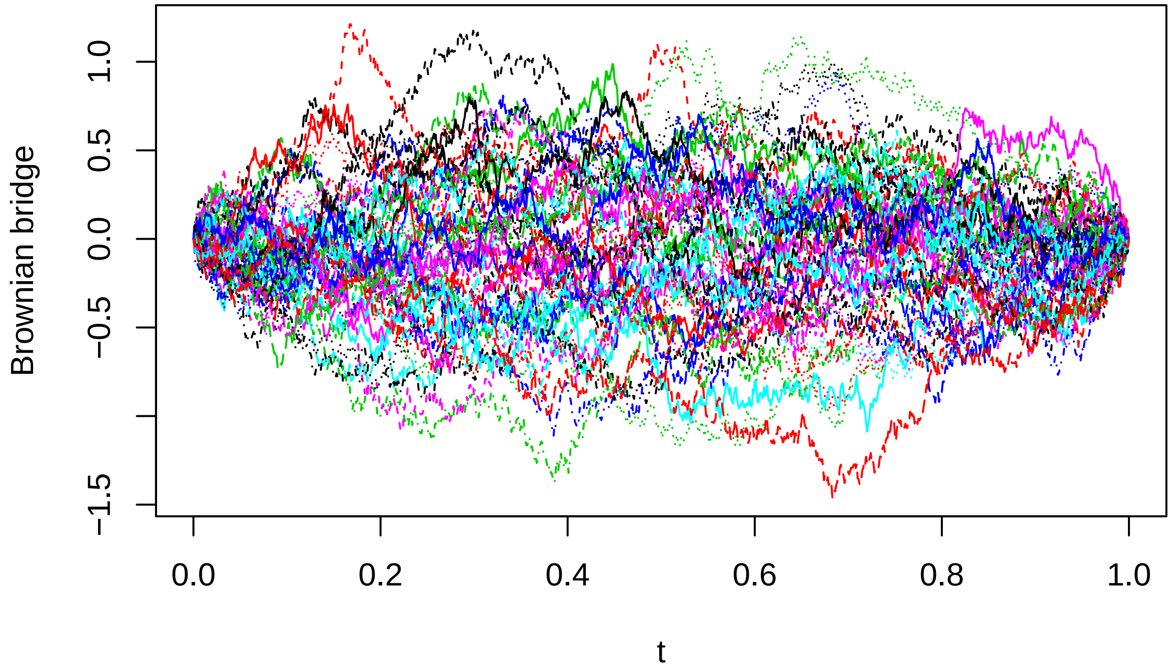
$$B(t) = \sum_{k=1}^{\infty} \sqrt{2}\xi_k \sin(k\pi t), \quad \xi_k \sim \mathcal{N}(0, 1/(k^2\pi^2)).$$

Problem 2

```r
m <- 1000 # number of grid points
n <- 50 # number of subjects
tt <- seq(0, 1, length.out=m+1)


## Brownian motion
Z <- matrix(rnorm(m * n, sd = 1 / sqrt(m)), n, m)
X <- t(apply(Z, 1, cumsum))
W <- cbind(rep(0,50), X)
matplot(tt, t(W), type='l', xlab = "t", ylab = "Brownian motion")
```



```
## Brownian bridge
Z <- matrix(rnorm(m * n, sd = 1 / sqrt(m)), n, m)
X <- t(apply(Z, 1, cumsum))
W <- cbind(rep(0,50), X)
tW1 <- W[,m+1] %*% t(tt)
B <- W - tW1
matplot(tt, t(B), type='l', xlab = "t", ylab = "Brownian bridge")
```



Problem 3

Note that the empirical cdf can be expressed as:

$$\hat{F}(x_0) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x_0),$$

Therefore, one reasonable way to approximate pdf is by:

$$\hat{f}(x_0) = \frac{\hat{F}(x_0 + h) - \hat{F}(x_0 - h)}{2h} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2h} I(-h \le X_i - x_0 \le h) = \frac{1}{n}\sum_{i=1}^{n} K_{U,h}(X_i - x_0),$$

where $K_{U,h}(x)$ denotes the uniform kernel.

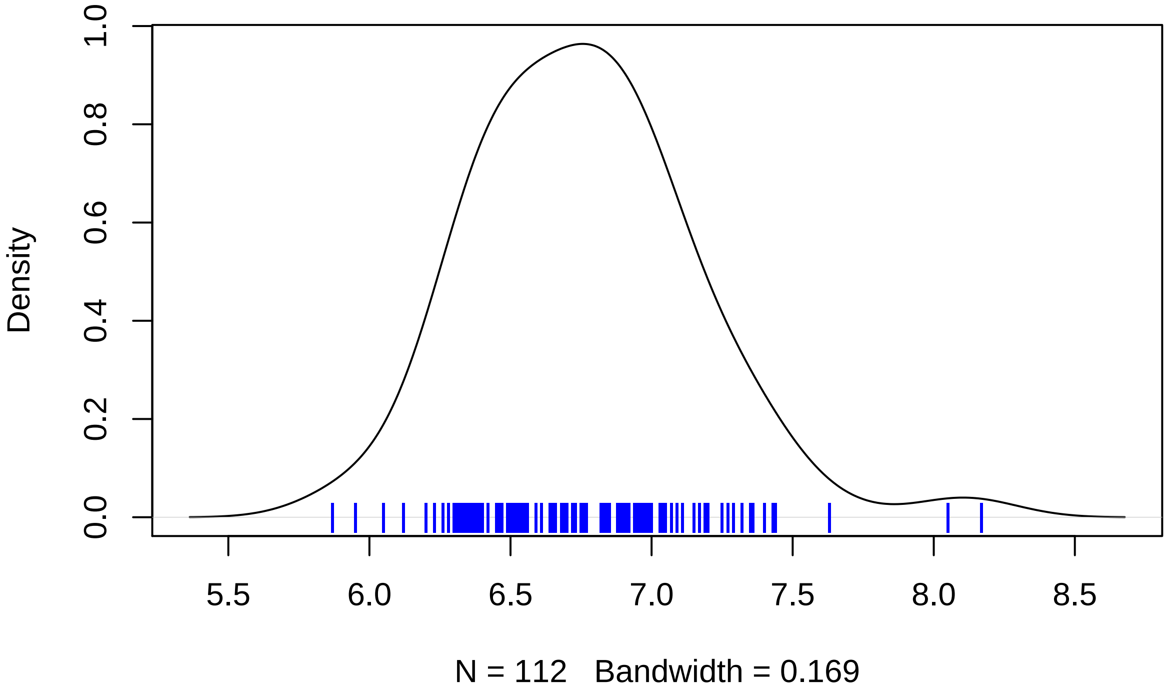
Problem 4

(a), (b)

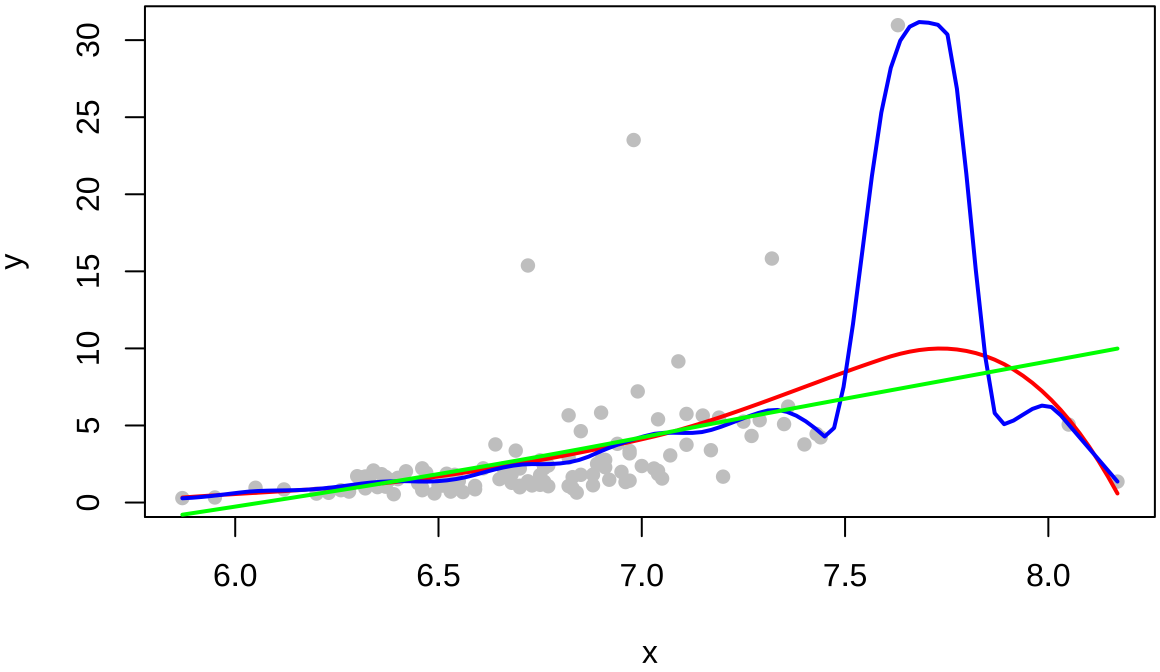
```r
library(gss)
library(locfit)
```

```
## locfit 1.5-9.1    2013-03-22
```

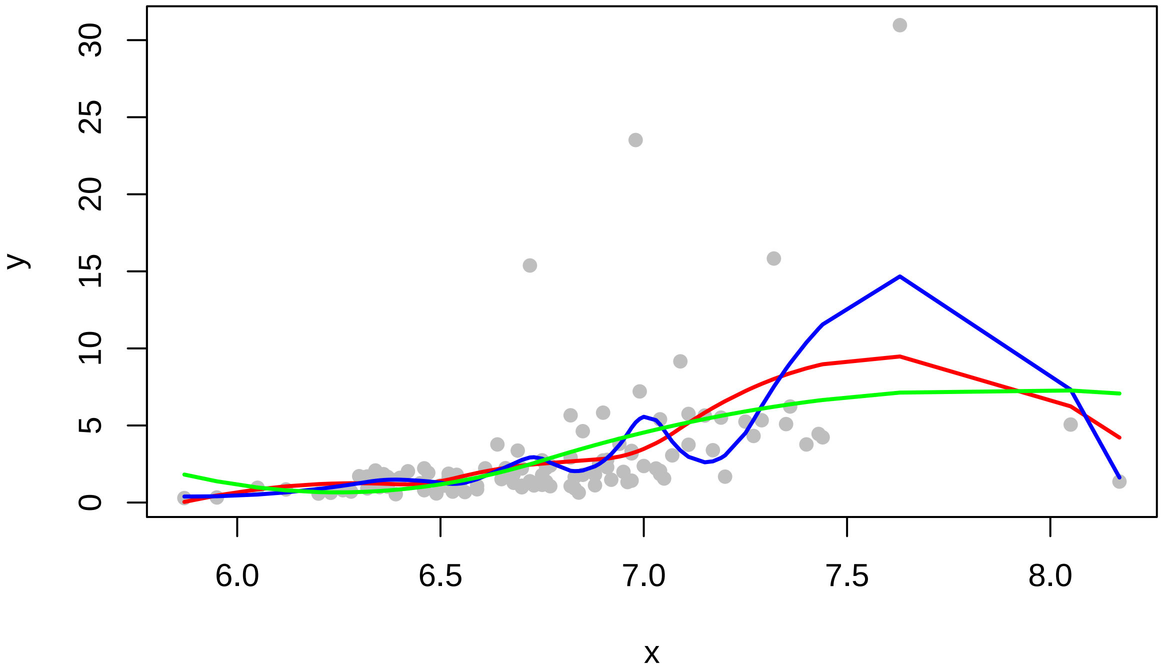
```r
library(splines)
data(LakeAcidity)

y <- LakeAcidity$cal
x <- LakeAcidity$ph

# (a)
plot(density(x, bw = "ucv"), main = "")
points(x = x, y = rep(0, length(x)), pch = "|", col = "blue")
```



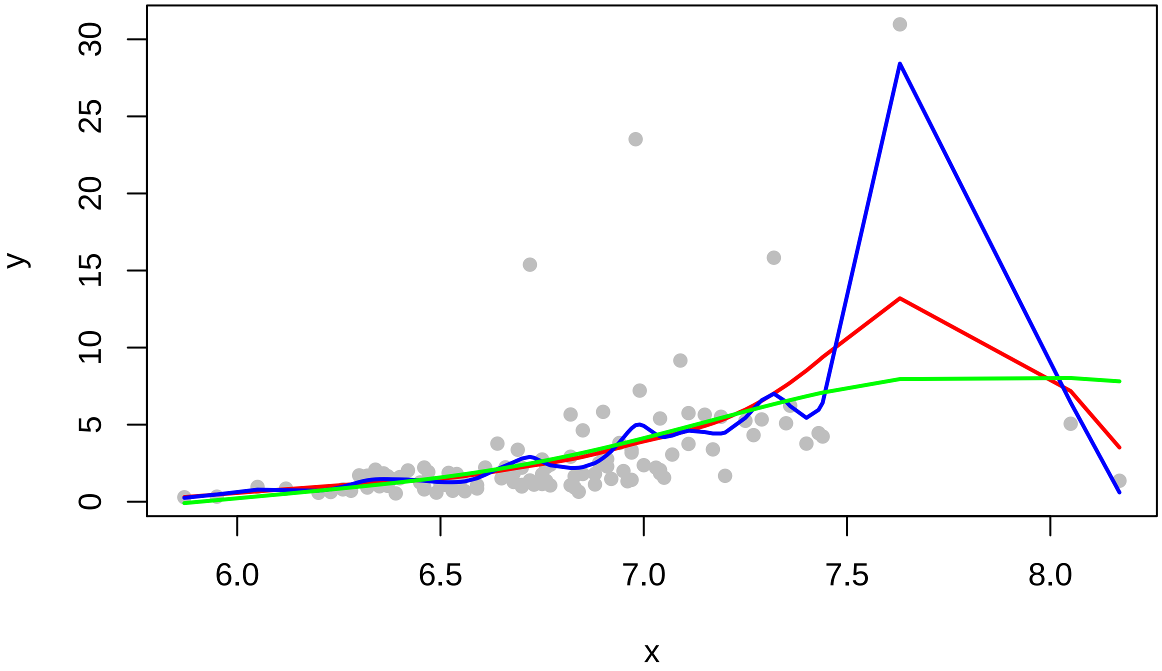
```
# (b)

## local linear regression
res1 <- locfit(y ~ lp(x, deg = 1, h = 0.8))
res2 <- locfit(y ~ lp(x, deg = 1, h = 0.2))
res3 <- locfit(y ~ lp(x, deg = 1, h = 2))

plot(x,y, pch = 16, col = "gray")
lines(res1, col = "red", lwd = 2)
lines(res2, col = "blue", lwd = 2)
lines(res3, col = "green", lwd = 2)
```



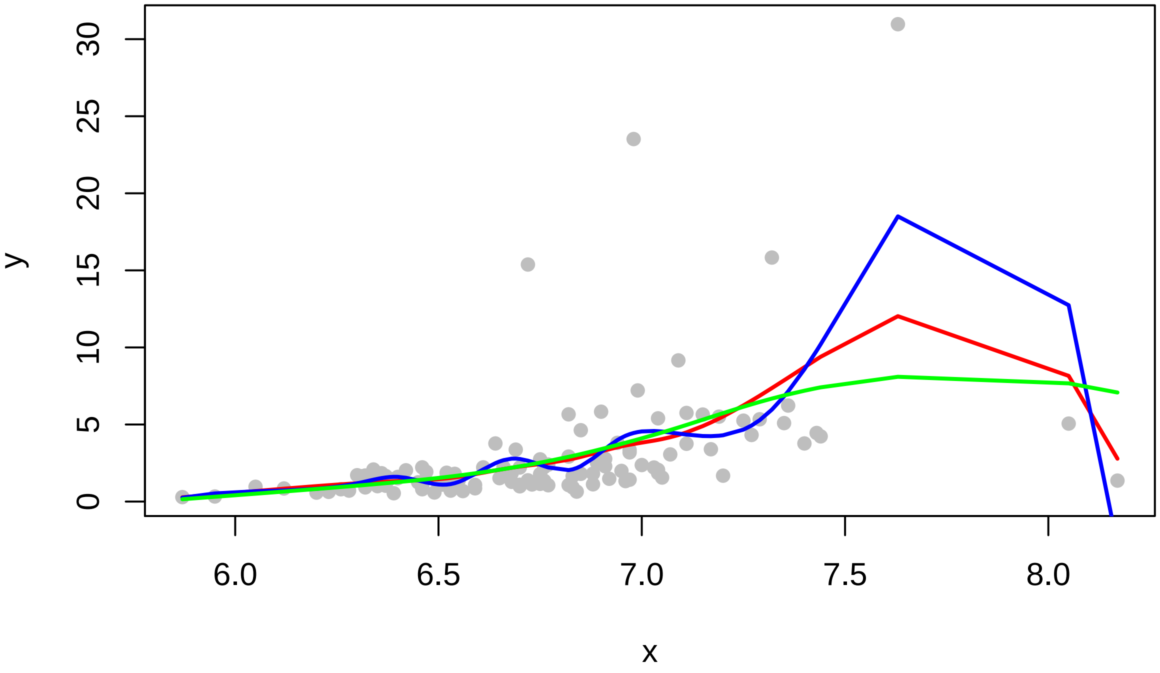
```
## regression spline
J1 <- 5
knots1 <- quantile(x, probs = seq(0, 1, length.out = J1 + 2))
knots1 <- knots1[-c(1, J1+2)]
bs1 <- bs(x, degree = 2, intercept = TRUE, knots = knots1)
res4 <- lm(y~0+bs1)

J2 <- 10
knots2 <- quantile(x, probs = seq(0, 1, length.out = J2 + 2))
knots2 <- knots2[-c(1, J2+2)]
bs2 <- bs(x, degree = 2, intercept = TRUE, knots = knots2)
res5 <- lm(y~0+bs2)

J3 <- 1
knots3 <- quantile(x, probs = seq(0, 1, length.out = J3 + 2))
knots3 <- knots3[-c(1, J3+2)]
bs3 <- bs(x, degree = 2, intercept = TRUE, knots = knots3)
res6 <- lm(y~0+bs3)

plot(x,y, pch = 16, col = "gray")
lines(x[order(x)], res4$fitted.values[order(x)], col = "red", lwd = 2)
lines(x[order(x)], res5$fitted.values[order(x)], col = "blue", lwd = 2)
lines(x[order(x)], res6$fitted.values[order(x)], col = "green", lwd = 2)
```



```r
## smoothing spline
res7 <- smooth.spline(x = x, y = y, lambda = 1e-3)
res8 <- smooth.spline(x = x, y = y, lambda = 1e-5)
res9 <- smooth.spline(x = x, y = y, lambda = 2e-2)

plot(x,y, pch = 16, col = "gray")
lines(res7, col = "red", lwd = 2)
lines(res8, col = "blue", lwd = 2)
lines(res9, col = "green", lwd = 2)
```



```
## penalized spline
res10 <- smooth.spline(x = x, y = y, lambda = 5e-4, nknots = 10)
res11 <- smooth.spline(x = x, y = y, lambda = 5e-6, nknots = 10)
res12 <- smooth.spline(x = x, y = y, lambda = 1e-2, nknots = 10)

plot(x,y, pch = 16, col = "gray")
lines(res10, col = "red", lwd = 2)
lines(res11, col = "blue", lwd = 2)
lines(res12, col = "green", lwd = 2)
```



(c)

The four methods provide similar results (they all can overfit and underfit under different tuning parameters settings). The non-uniform distributed predictor introduces more difficulty for nonparametric estimation.