Week 6 exercise solutions

Ali Movasati, Isabelle Cretton, Tristan Koning

Oct. 14th, 2024

```
# Set global code chunk options
knitr::opts_chunk$set(warning = FALSE)
```

```
# load required libraries
library(skimr)
library(ggplot2)
library(np)
```

```
## Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-17)
## [vignette("np_faq",package="np") provides answers to frequently asked questions]
## [vignette("np",package="np") an overview]
## [vignette("entropy_np",package="np") an overview of entropy-based methods]
```

```
library(splines)
library(sm)
```

```
## Package 'sm', version 2.2-6.0: type help(sm) for summary information
```

Exercise 1

(a)

```
# Load and explore data
medflies <- read.table(file = "data/medflies.txt", sep = "\t", header = TRUE)
medflies$mort.rate <- as.numeric(medflies$mort.rate)
skim(medflies)
```

Table 1: Data summary

Name	medflies
Number of rows	173
Number of columns	3
Column type frequency:	
numeric	3

Group variables	None

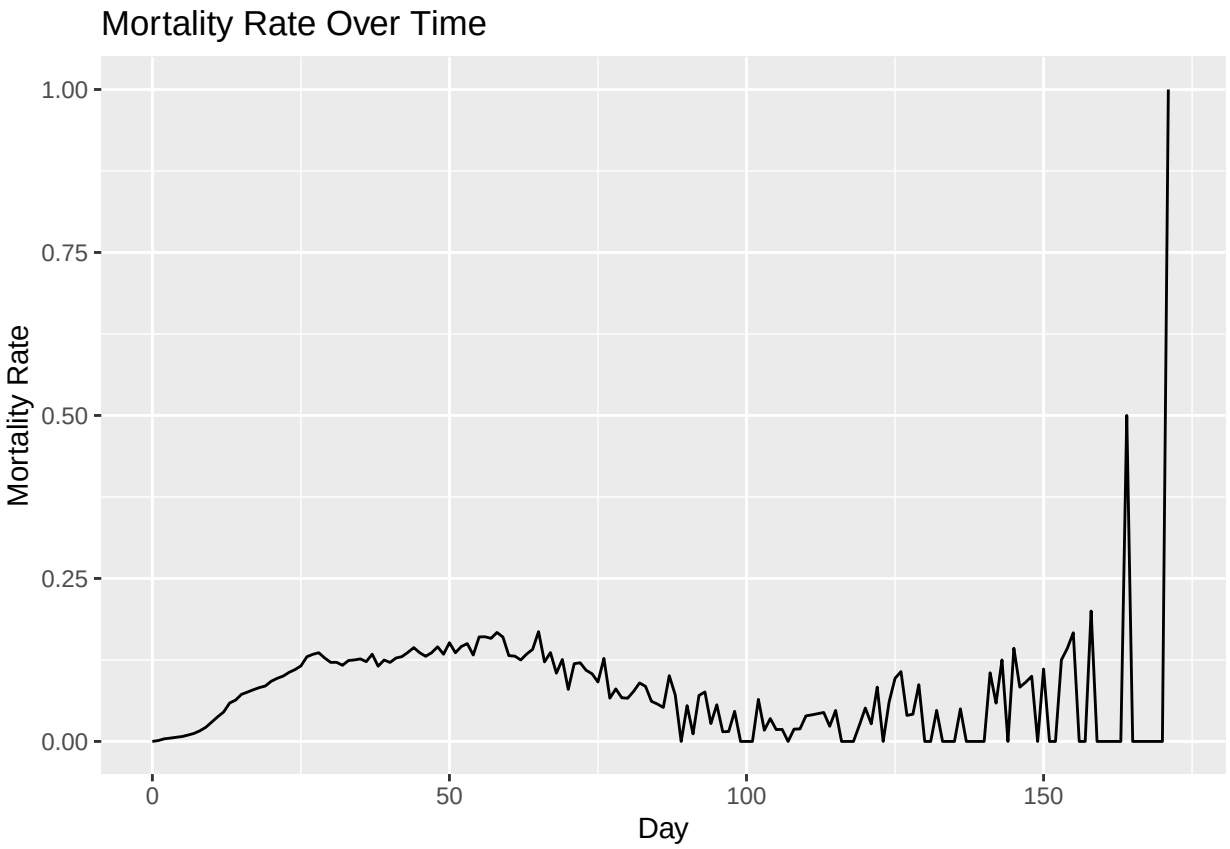
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
day	0	1.00	86.00	50.08	0	43.00	86.00	129.00	172	
living	0	1.00	148501.07	339536.10	0	23.00	115.00	30360.00	1203646	
mort.rate	1	0.99	0.08	0.10	0	0.01	0.07	0.12	1	

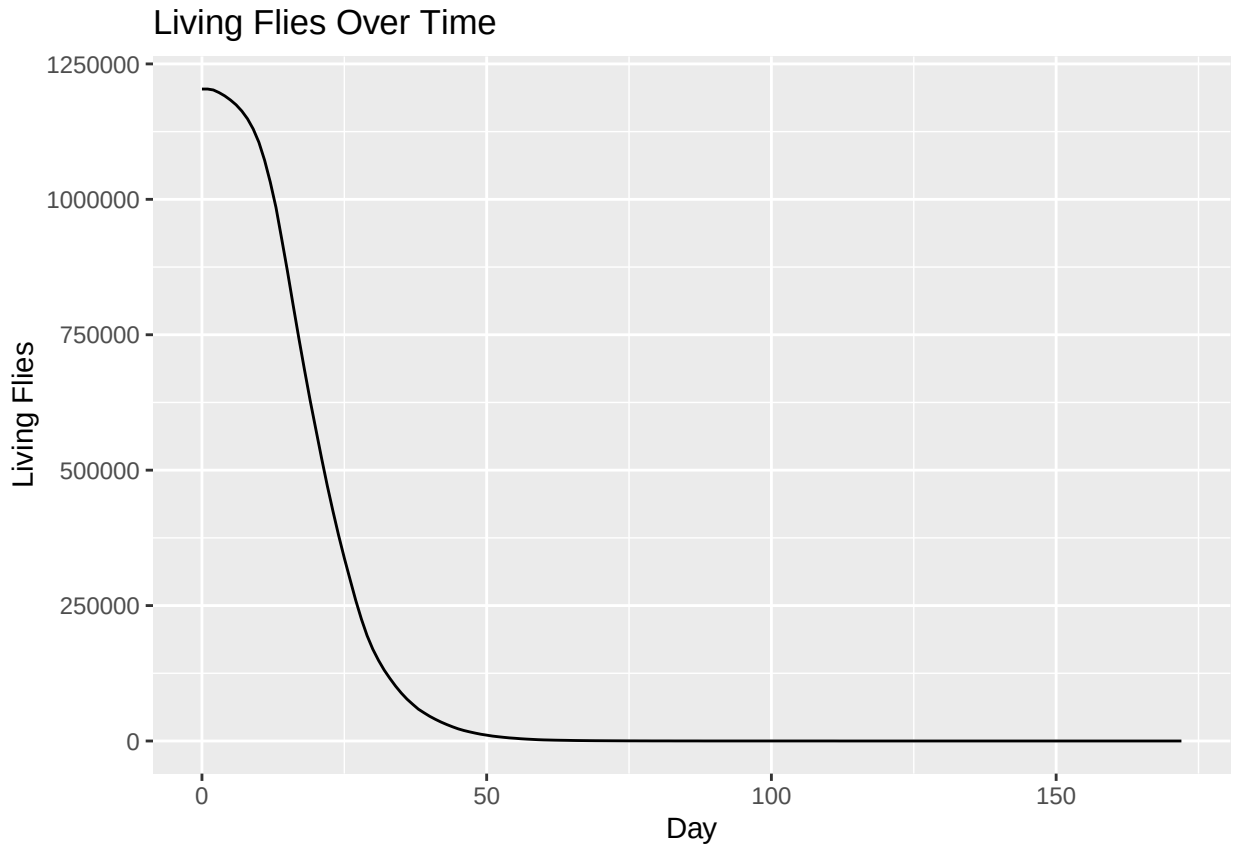
```
str(medflies)
```

```
## 'data.frame':    173 obs. of  3 variables:
##  $ day      : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ living   : int  1203646 1203646 1201913 1197098 1191020 1183419 1174502 1163026 1148693 1129836 .
##  $ mort.rate: num  0 0.0014 0.004 0.0051 0.0064 0.0075 0.0098 0.0123 0.0164 0.0218 ...
```

```
# Graphical EDA
ggplot(medflies, aes(x = day, y = mort.rate)) +
  geom_line() +
  labs(title = "Mortality Rate Over Time", x = "Day", y = "Mortality Rate")
```

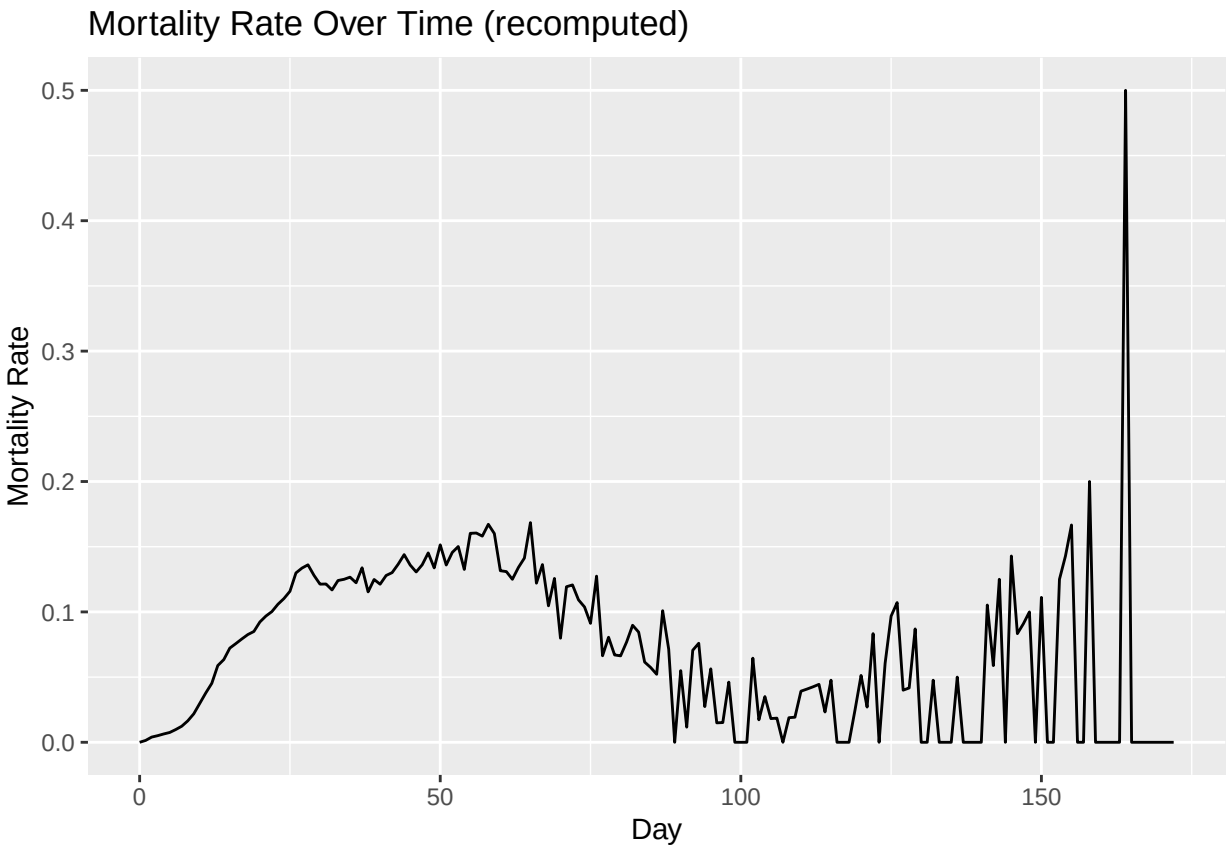


```r
ggplot(medflies, aes(x = day, y = living)) +
  geom_line() +
  labs(title = "Living Flies Over Time", x = "Day", y = "Living Flies")
```



(b)

```r
for (i in 1:(nrow(medflies) - 2)) {
  medflies$mort.rate2[i] <- (medflies$living[i] - medflies$living[i + 1]) / medflies$living[i]
}

ggplot(medflies, aes(x = day, y = mort.rate2)) +
  geom_line() +
  labs(title = "Mortality Rate Over Time (recomputed)", x = "Day", y = "Mortality Rate")
```



mort.rate and mort.rate2 are very similar, but not identical. mort.rate2 is slightly higher than mort.rate. This is likely due to the fact that the original mort.rate was rounding to 4 decimal places, while mort.rate2 is rounding on a higher precision.

(c)

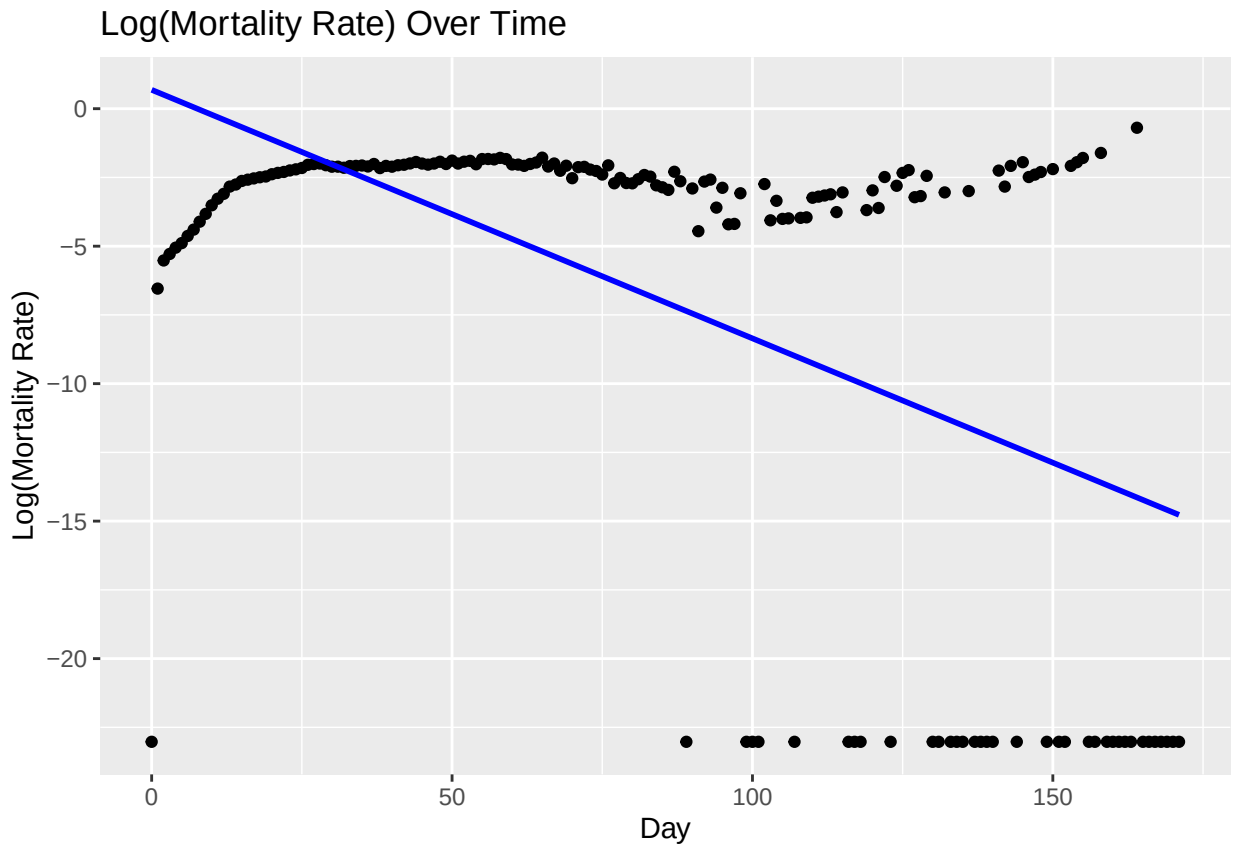
```
# Remove NA values
medflies <- medflies[complete.cases(medflies), ]
medflies$mort.rate2[medflies$mort.rate2 == 0] <- 1e-10

# Transform the mortality rate
medflies$log_mort_rate2 <- log(medflies$mort.rate2)

# Fit a linear regression model
model_A <- lm(log_mort_rate2 ~ day, data = medflies)

summary(model_A)
```

```
##
## Call:
## lm(formula = log_mort_rate2 ~ day, data = medflies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

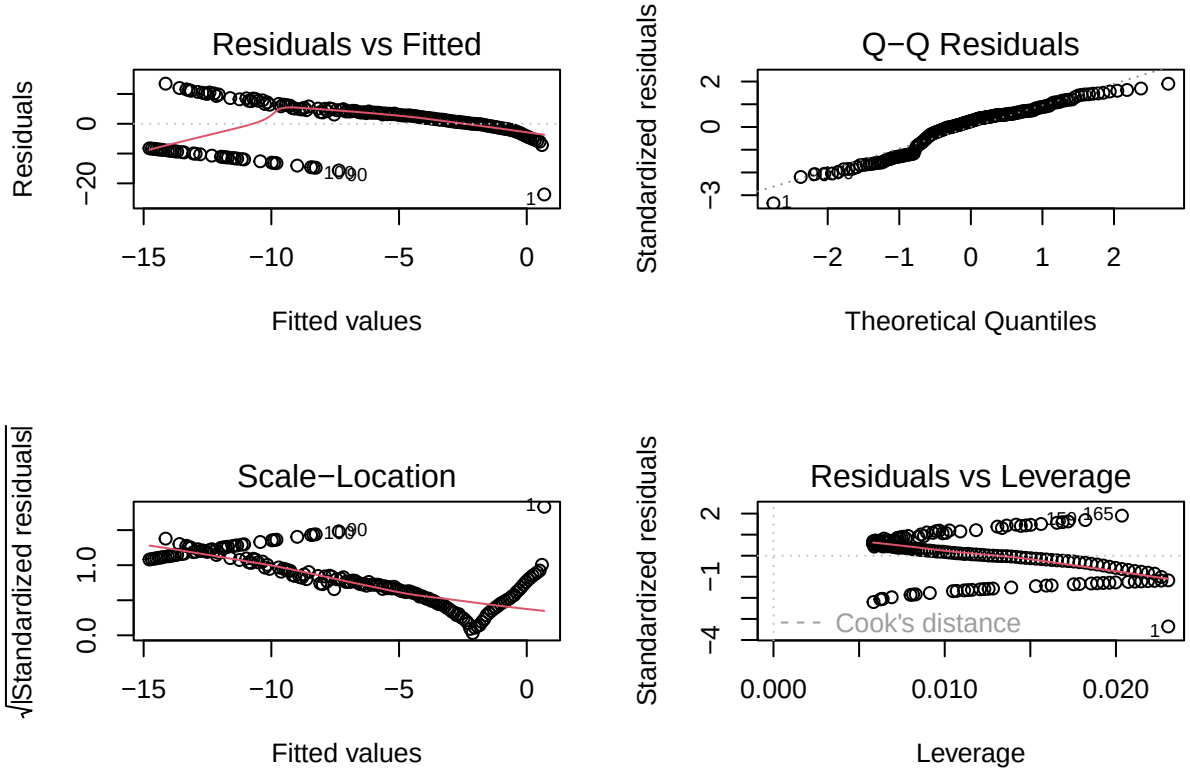
```
## -23.712  -4.530   1.830   4.581  13.446
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.68625    1.08520   0.632    0.528
## day         -0.09040    0.01098  -8.236 4.54e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.147 on 170 degrees of freedom
## Multiple R-squared:  0.2852, Adjusted R-squared:  0.281
## F-statistic: 67.84 on 1 and 170 DF,  p-value: 4.54e-14
```

```r
ggplot(medflies, aes(x = day, y = log_mort_rate2)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Log(Mortality Rate) Over Time", x = "Day", y = "Log(Mortality Rate)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
par(mfrow = c(2, 2))
plot(model_A)
```

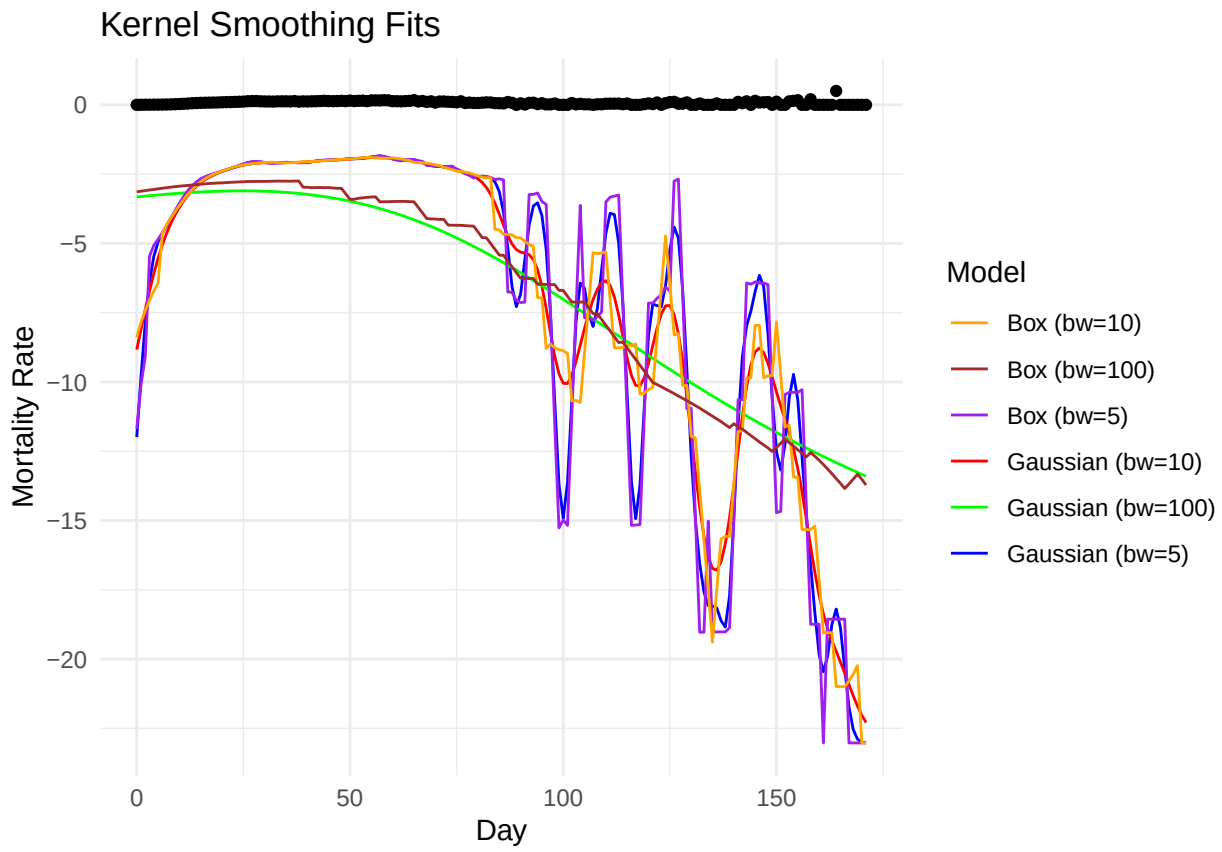


Looking at the plot of the model, we can observe a decrease in the mortality rate over time. However, we can spot that the first few days indeed to have a linear increase in the log(Mortality Rate), therefore there does exist some exponential growth in the beginning. This stops after roughly 10-15 days.

(d)

```r
# Fit models with different kernels and bandwidths
gaussian_A <- ksmooth(medflies$day, medflies$log_mort_rate2, kernel = "normal", bandwidth = 5)
gaussian_B <- ksmooth(medflies$day, medflies$log_mort_rate2, kernel = "normal", bandwidth = 10)
gaussian_C <- ksmooth(medflies$day, medflies$log_mort_rate2, kernel = "normal", bandwidth = 100)

box_A <- ksmooth(medflies$day, medflies$log_mort_rate2, kernel = "box", bandwidth = 5)
box_B <- ksmooth(medflies$day, medflies$log_mort_rate2, kernel = "box", bandwidth = 10)
box_C <- ksmooth(medflies$day, medflies$log_mort_rate2, kernel = "box", bandwidth = 100)

fitted_values <- data.frame(
  day = medflies$day,
  mort.rate2 = medflies$mort.rate2,
  gaussian_A = gaussian_A$y,
  gaussian_B = gaussian_B$y,
  gaussian_C = gaussian_C$y,
  box_A = box_A$y,
  box_B = box_B$y,
  box_C = box_C$y
```

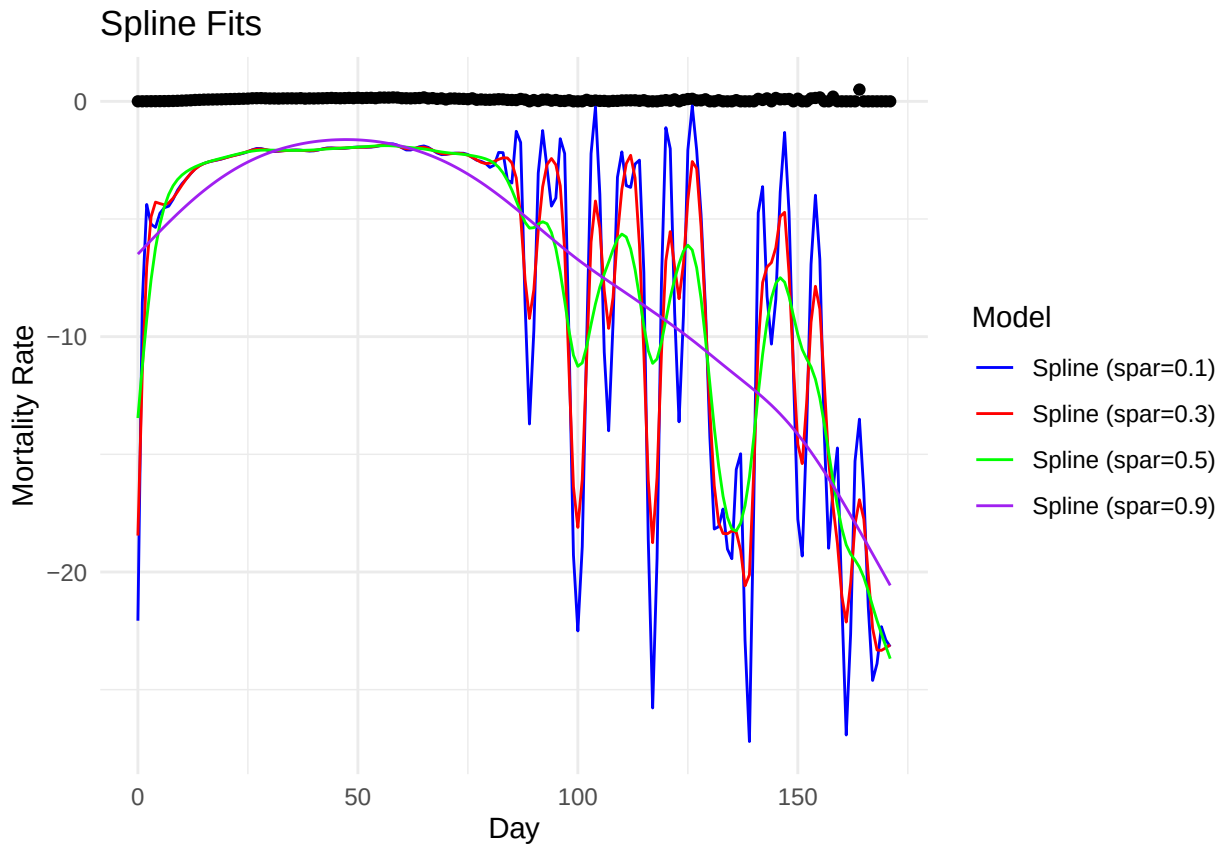
```
)

ggplot(medflies, aes(x = day, y = mort.rate2)) +
  geom_point() +
  geom_line(data = fitted_values, aes(x = day, y = gaussian_A, color = "Gaussian (bw=5)")) +
  geom_line(data = fitted_values, aes(x = day, y = gaussian_B, color = "Gaussian (bw=10)")) +
  geom_line(data = fitted_values, aes(x = day, y = gaussian_C, color = "Gaussian (bw=100)")) +
  geom_line(data = fitted_values, aes(x = day, y = box_A, color = "Box (bw=5)")) +
  geom_line(data = fitted_values, aes(x = day, y = box_B, color = "Box (bw=10)")) +
  geom_line(data = fitted_values, aes(x = day, y = box_C, color = "Box (bw=100)")) +
  scale_color_manual(name = "Model", values = c("Gaussian (bw=5)" = "blue", "Gaussian (bw=10)" = "red",
  labs(title = "Kernel Smoothing Fits", x = "Day", y = "Mortality Rate") +
  theme_minimal()
```



(e) i

```
spline_A <- smooth.spline(medflies$day, medflies$log_mort_rate2, spar = 0.1)
spline_B <- smooth.spline(medflies$day, medflies$log_mort_rate2, spar = 0.3)
spline_C <- smooth.spline(medflies$day, medflies$log_mort_rate2, spar = 0.5)
spline_D <- smooth.spline(medflies$day, medflies$log_mort_rate2, spar = 0.9)

fitted_values <- data.frame(
  day = medflies$day,
```

```
  mort.rate2 = medflies$mort.rate2,
  spline_A = predict(spline_A)$y,
  spline_B = predict(spline_B)$y,
  spline_C = predict(spline_C)$y,
  spline_D = predict(spline_D)$y
)

ggplot(medflies, aes(x = day, y = mort.rate2)) +
  geom_point() +
  geom_line(data = fitted_values, aes(x = day, y = spline_A, color = "Spline (spar=0.1)")) +
  geom_line(data = fitted_values, aes(x = day, y = spline_B, color = "Spline (spar=0.3)")) +
  geom_line(data = fitted_values, aes(x = day, y = spline_C, color = "Spline (spar=0.5)")) +
  geom_line(data = fitted_values, aes(x = day, y = spline_D, color = "Spline (spar=0.9)")) +
  scale_color_manual(name = "Model", values = c("Spline (spar=0.1)" = "blue", "Spline (spar=0.3)" = "red
  labs(title = "Spline Fits", x = "Day", y = "Mortality Rate") +
  theme_minimal()
```

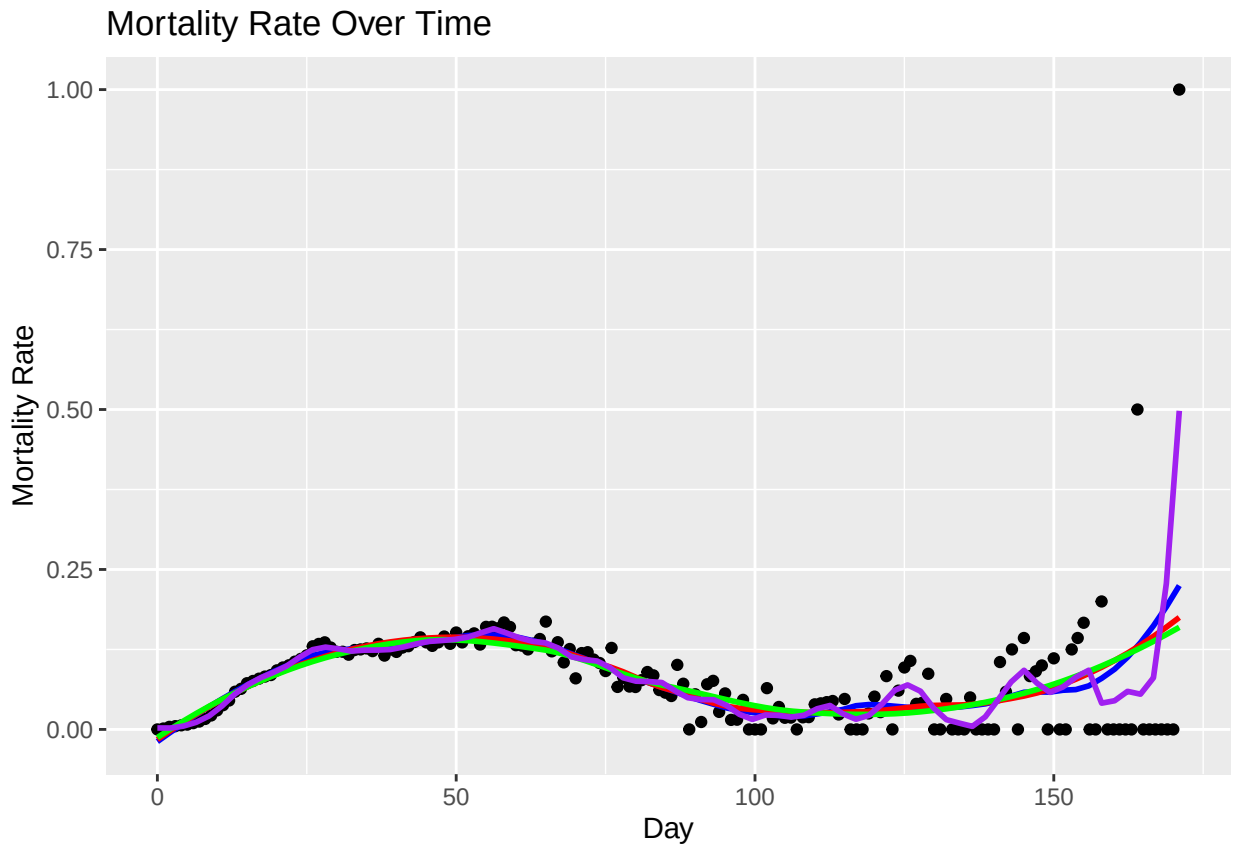


(e) ii

```
# Fit a model with different spans
model_B <- loess(mort.rate ~ day, data = medflies, span = 0.3)
model_C <- loess(mort.rate ~ day, data = medflies, span = 0.5)
model_D <- loess(mort.rate ~ day, data = medflies, span = 0.7)
```

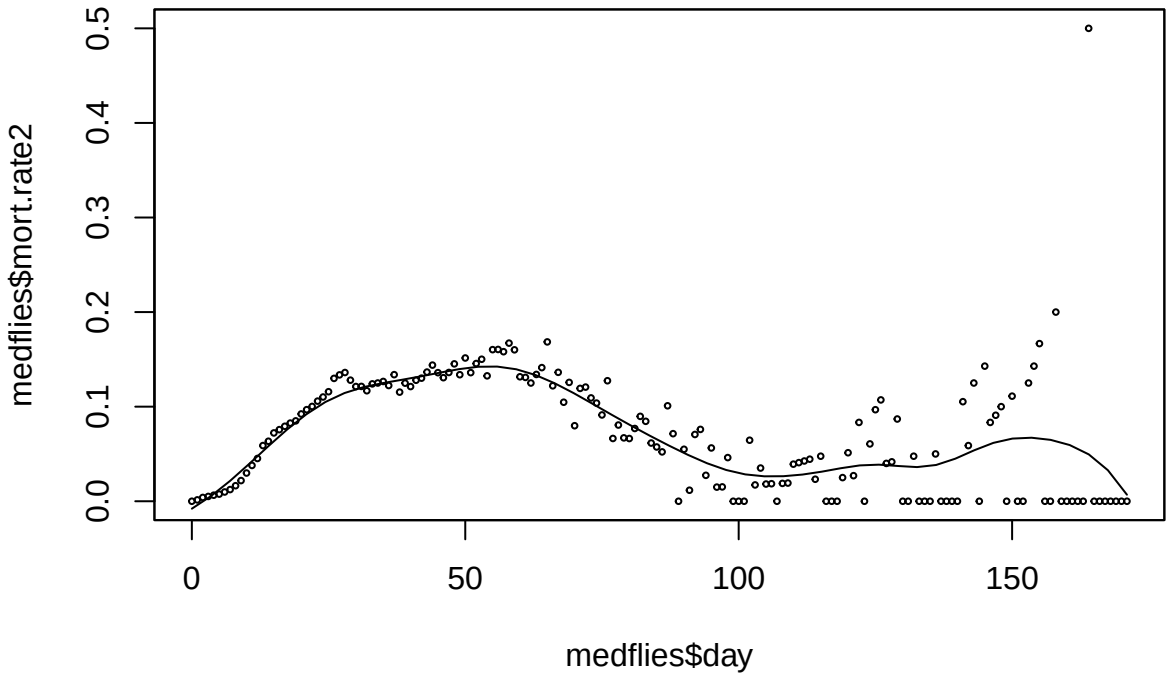
```r
model_E <- loess(mort.rate ~ day, data = medflies, span = 0.1)

# Plot the models
ggplot(medflies, aes(x = day, y = mort.rate)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue", span = 0.3) +
  geom_smooth(method = "loess", se = FALSE, color = "red", span = 0.5) +
  geom_smooth(method = "loess", se = FALSE, color = "green", span = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "purple", span = 0.1) +
  labs(title = "Mortality Rate Over Time", x = "Day", y = "Mortality Rate")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

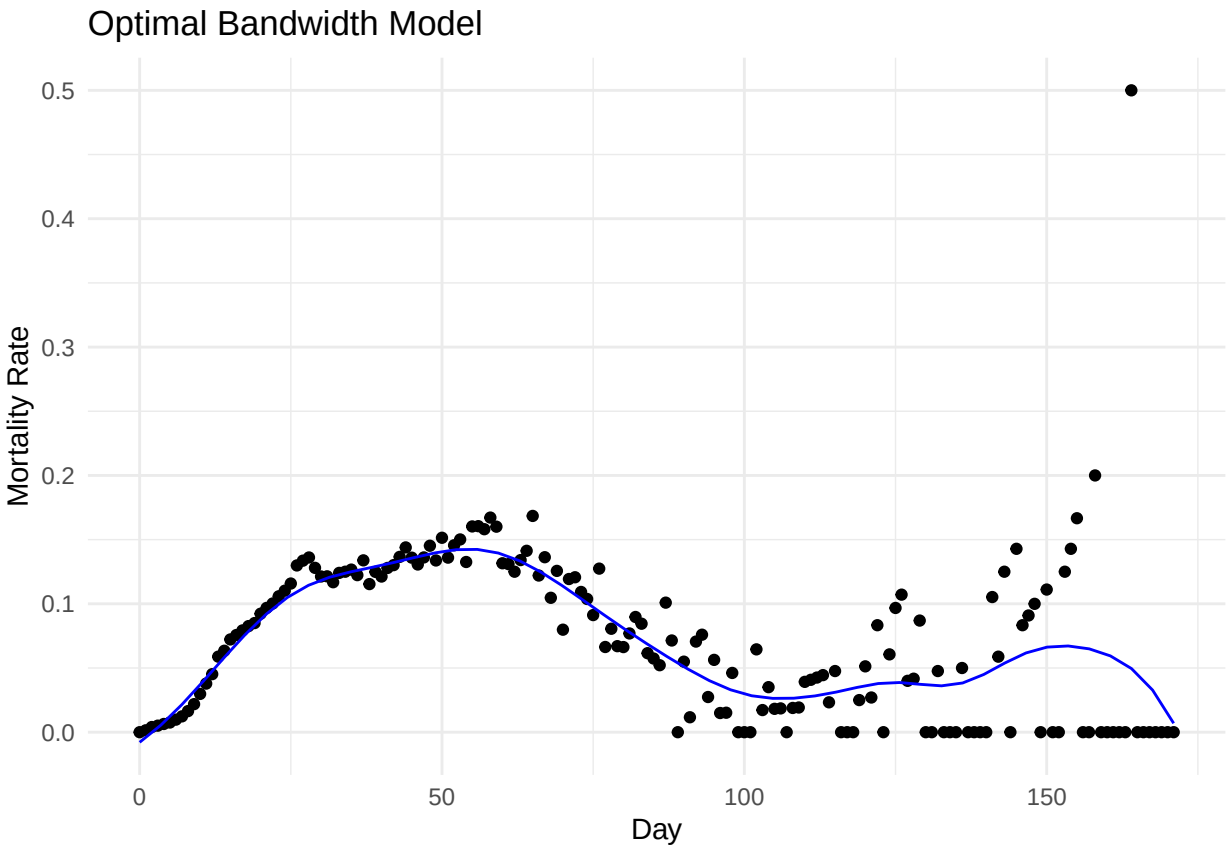


(f)

```r
opt_bw <- hcv(medflies$day, medflies$mort.rate2, kernel = "normal")

model_opt <- sm.regression(x = medflies$day, y = medflies$mort.rate2, h = opt_bw)
```




```
fitted_values_opt <- data.frame(
  day = model_opt$eval.points,
  mort.rate2 = model_opt$estimate
)

# Plot the data and the resulting fit
ggplot(medflies, aes(x = day, y = mort.rate2)) +
  geom_point() +
  geom_line(data = fitted_values_opt, aes(x = day, y = mort.rate2), color = "blue") +
  labs(title = "Optimal Bandwidth Model", x = "Day", y = "Mortality Rate") +
  theme_minimal()
```




```
## (g)
```

```
# TODO
```

(h)

We want to use non-parametric models when we dont know the distribution of our data, have very little data to work with or if the data does not follow a known distribution that allows it to be used in a parametric model.

A question that could be asked is whether the chance of getting cancer given different lifestyle choices? This could be solved more easily with non-parametric rather than parametric model since the interactions of a persons lifestyle will probably not follow a certain distribution.

A question that can be answered with a linear model, could be if smoking affects the cholesterol level.