Introduction to Machine Learning

Lab 1 Block 2

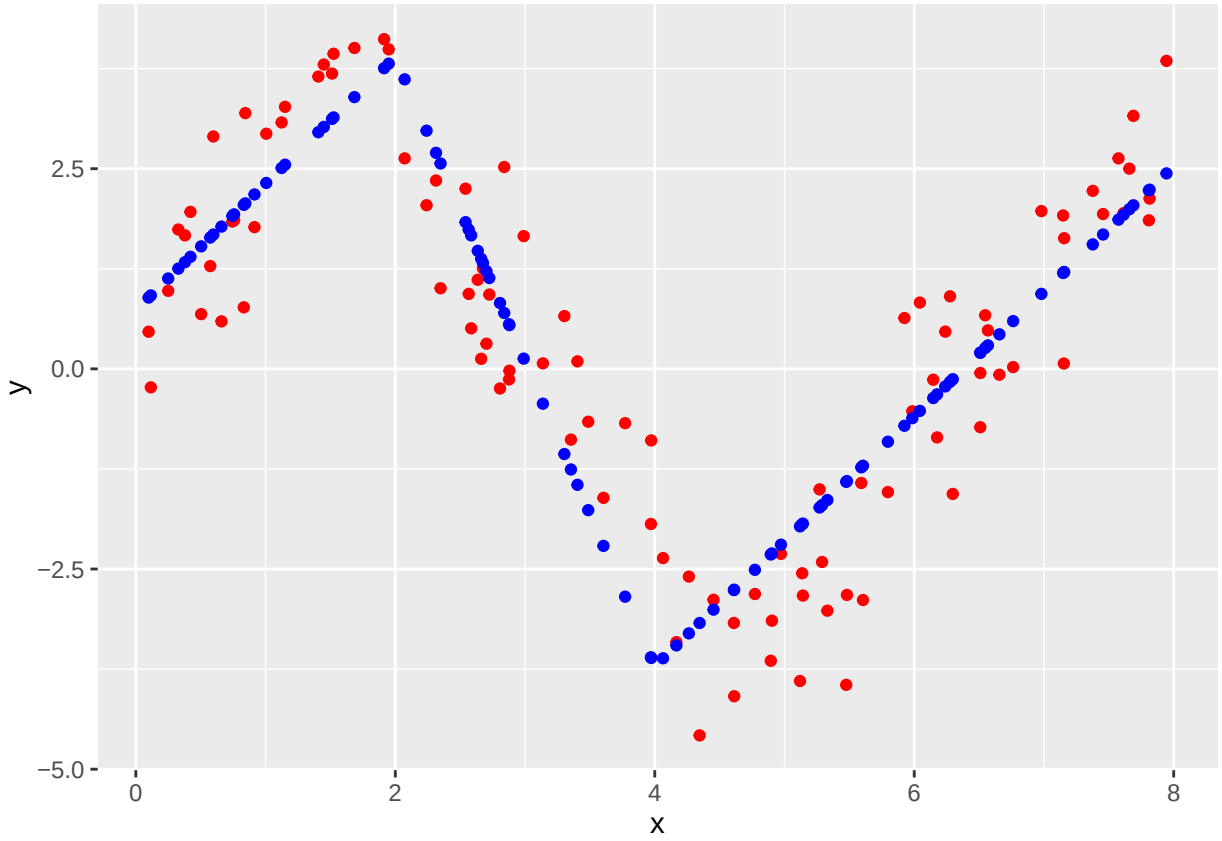
Rasmus Holm

2016-11-16

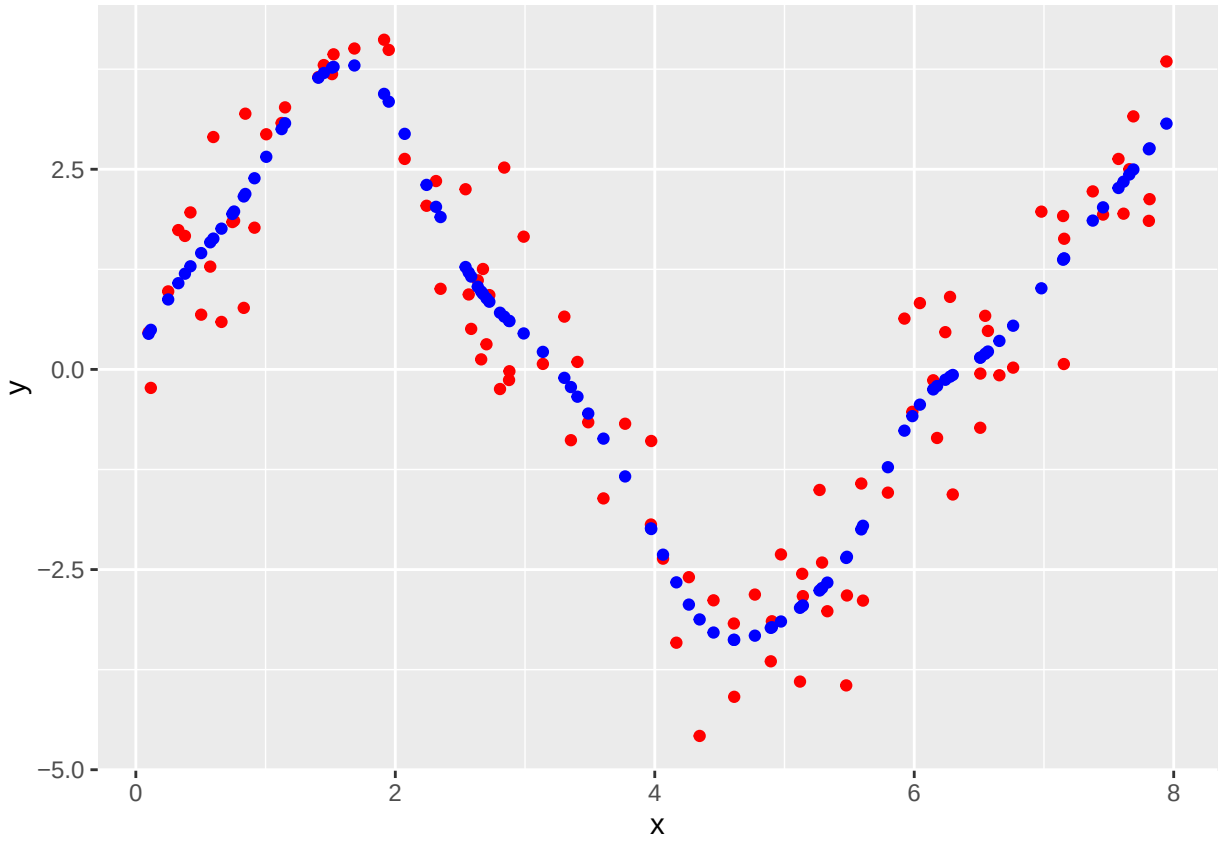
Contents

Assignment 1

2

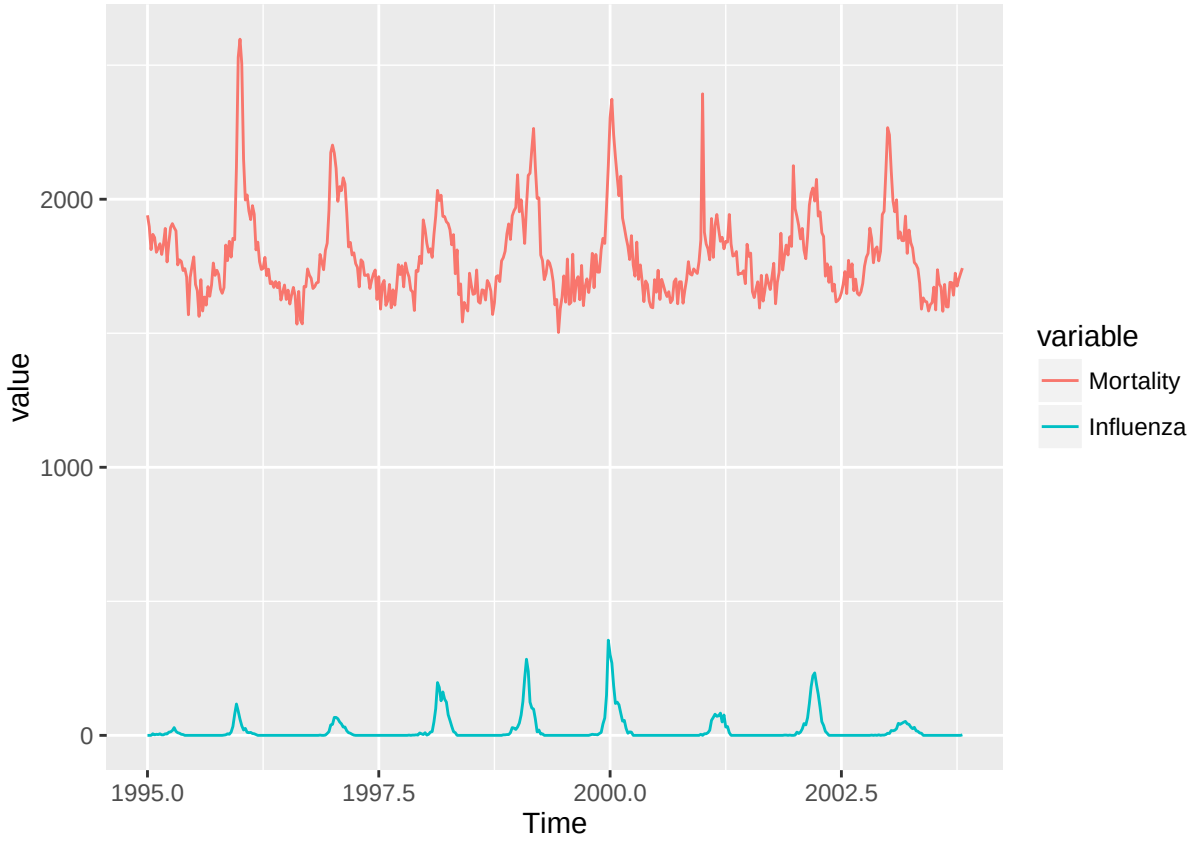


3



Assignment 2

1



2

3

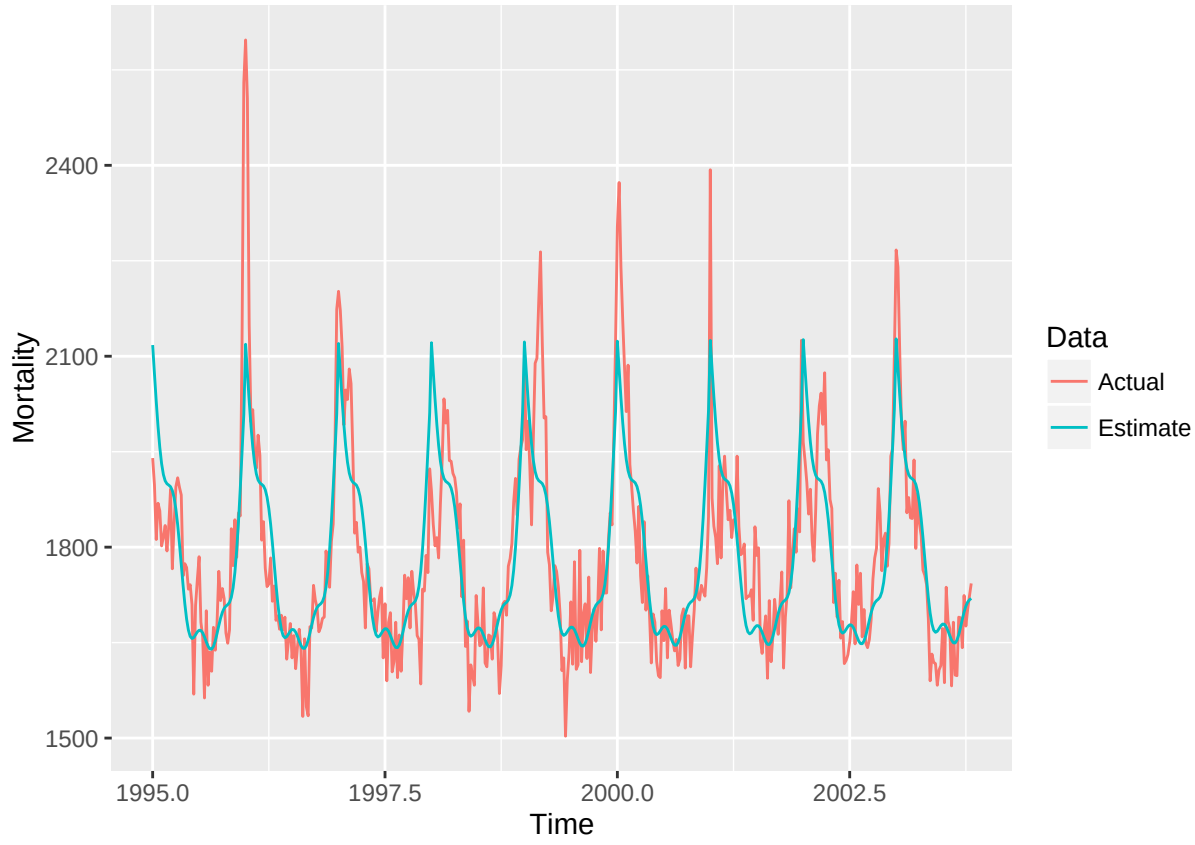


Figure 1: Caption.

```
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> Mortality ~ Year + s(Week)
#>
#> Parametric coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -652.058   3448.379  -0.189     0.85
#> Year           1.219      1.725   0.706     0.48
#>
#> Approximate significance of smooth terms:
#>           edf Ref.df     F p-value
#> s(Week) 8.587  8.951 100.6  <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.661   Deviance explained = 66.8%
```

```
#> GCV = 9014.6  Scale est. = 8806.7    n = 459
```

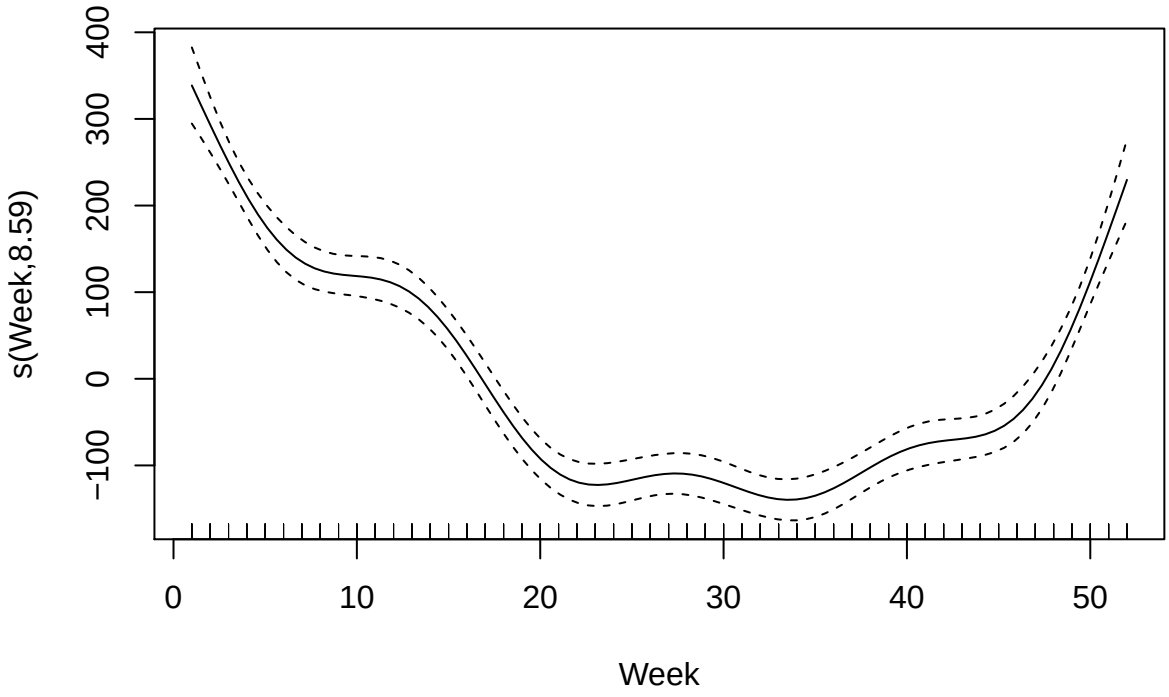
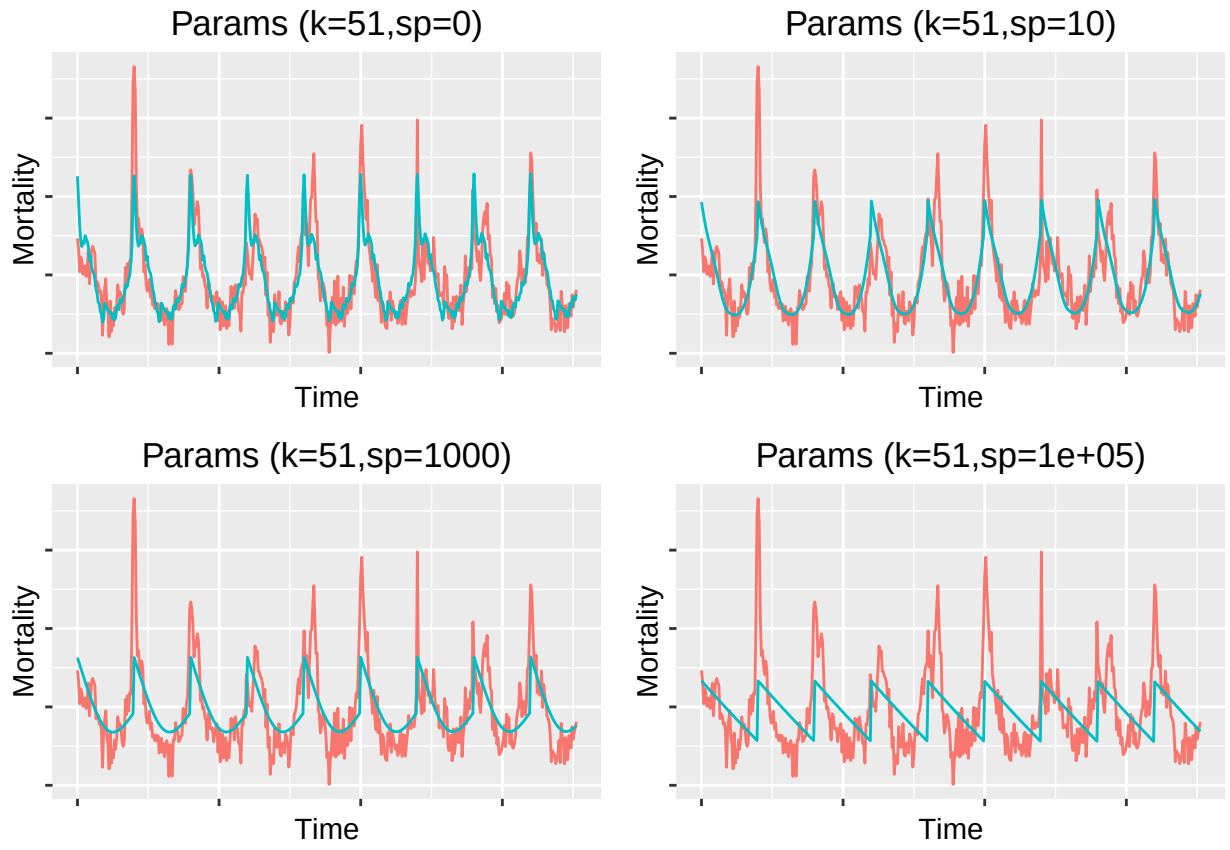
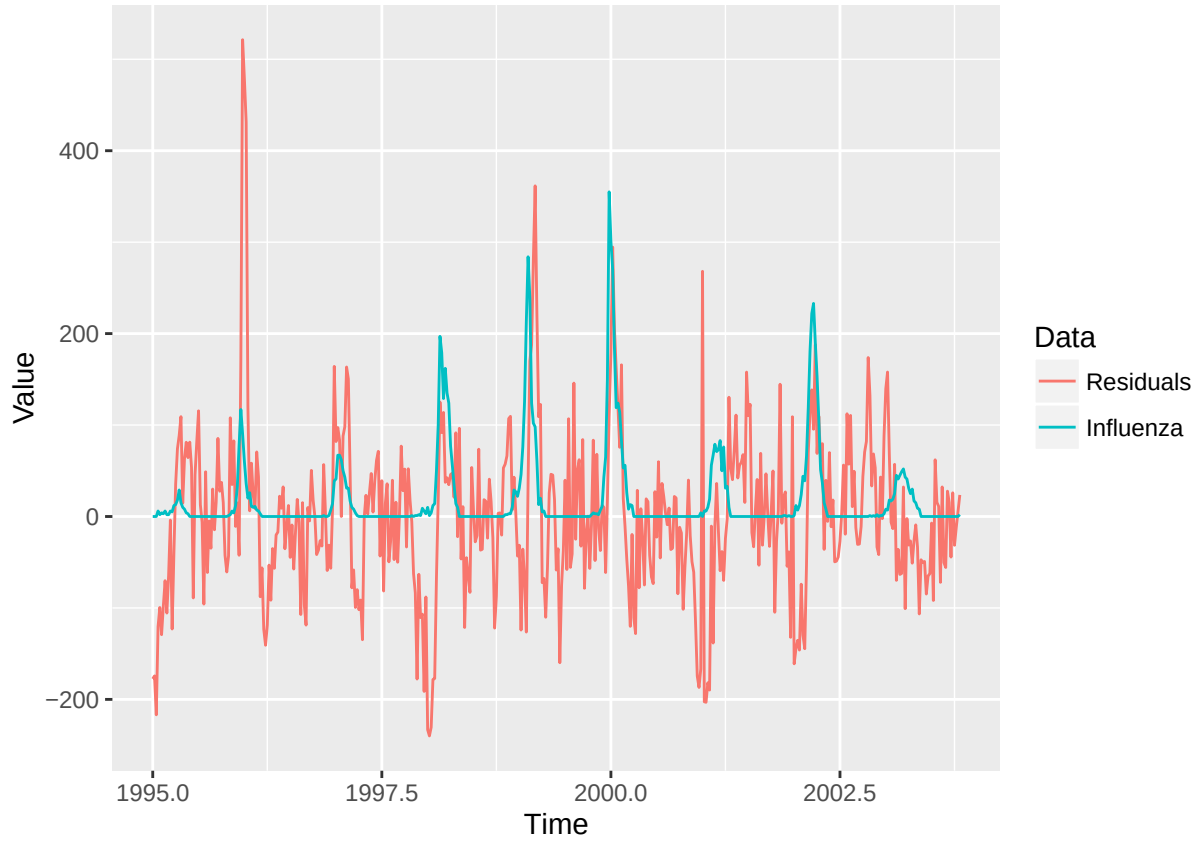


Figure 2: Caption.

4


Params (k=51,sp=0)
Params (k=51,sp=10)
Params (k=51,sp=1000)
Params (k=51,sp=1e+05)
Mortality
Time

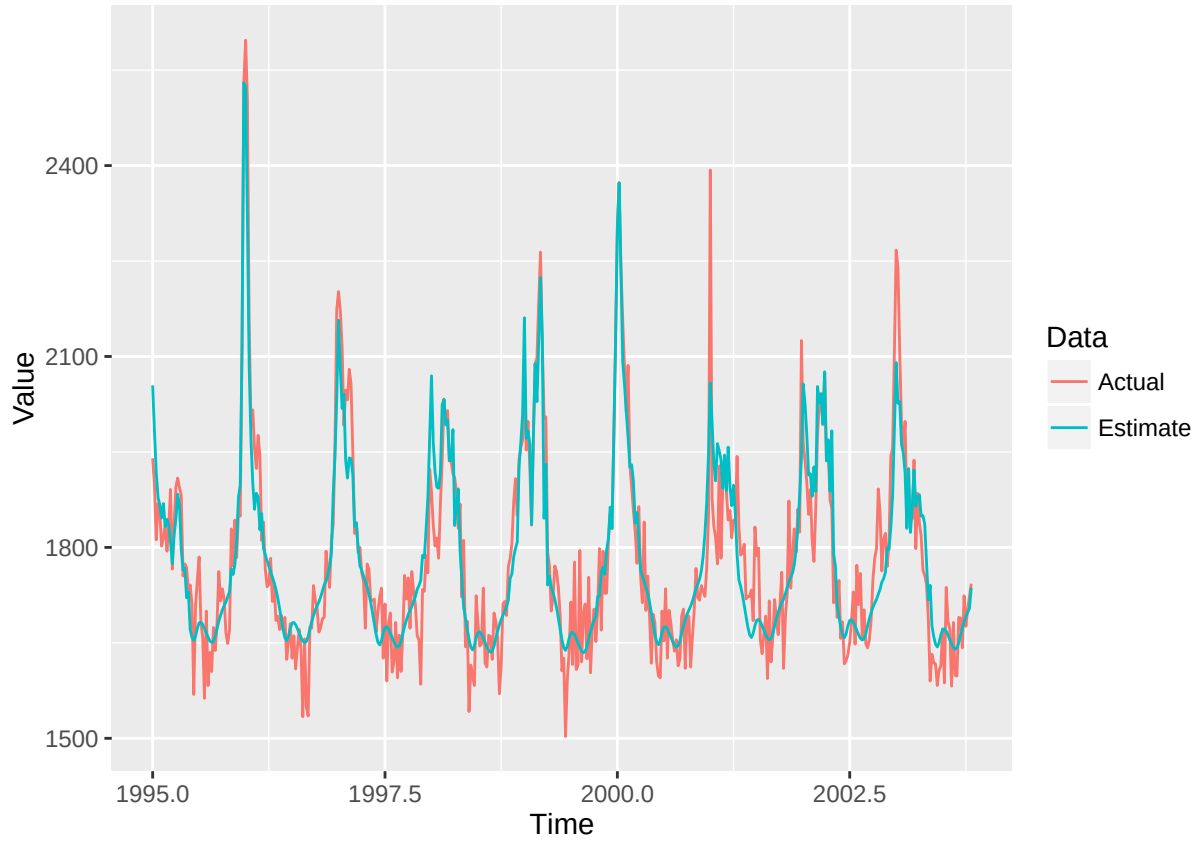
5



6

```
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> Mortality ~ s(Year, k = length(unique(data$Year)) - 1) + s(Week,
#>     k = length(unique(data$Week)) - 1) + s(Influenza, k = length(unique(data$Influenza)))
#>
#> Parametric coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  1783.77       3.28   543.9   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                edf Ref.df      F p-value
#> s(Year)      3.913  4.756  1.179   0.292
#> s(Week)     13.886 17.271 20.271  <2e-16 ***
#> s(Influenza) 59.204 65.527  5.336  <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#>
#> Rank: 132/142
#> R-sq.(adj) =   0.81   Deviance explained = 84.2%
#> GCV = 5947.8  Scale est. = 4937      n = 459
```



Appendix

Code for Assignment 1

```
library(ggplot2)

myspline <- function(X, y, knots) {
    n <- length(X)
    m <- length(knots)
    df <- m + 2

    H <- matrix(0, nrow=n, ncol=df)
    H[, 1] <- 1
    H[, 2] <- X

    for (i in 3:df) {
        H[, i] <- pmax(X - knots[i - 2], 0)
    }

    data <- data.frame(y=y, H)
    ## Removes the intercept term (have it already)
    lmfit <- lm(y ~ 0 + ., data=data)
    coefficients <- as.numeric(coef(lmfit))
    yhat <- H %*% coefficients

    yhat
}

data <- read.csv2("../data/cube.csv", header=TRUE, sep=";")
knots <- c(2, 4)
yhat <- myspline(data$x, data$y, knots)

plot_data <- data.frame(x=data$x, y=data$y, yhat=yhat)

ggplot(plot_data) +
    geom_point(aes(x, y), color="red") +
    geom_point(aes(x, yhat), color="blue")
smooth_fit <- smooth.spline(x=data$x, y=data$y)
yhat <- fitted(smooth_fit)

## plot(smooth_fit, col="blue")
## points(data$x, data$y, col="red")

plot_data <- data.frame(x=data$x, y=data$y, yhat=yhat)

ggplot(plot_data) +
    geom_point(aes(x, y), color="red") +
    geom_point(aes(x, yhat), color="blue")
```

Code for Assignment 2

```
library(ggplot2)
library(readxl)
library(reshape2)
library(mgcv)
library(grid)
library(gridExtra)

data <- read_excel("../data/Influenza.xlsx")
plot_data <- melt(data[, c("Time", "Mortality", "Influenza")], id="Time")
ggplot(plot_data) +
    geom_line(aes(x=Time, y=value, color=variable))
gamfit <- gam(Mortality ~ Year + s(Week), family=gaussian, data=data, method="GCV.Cp")
yhat <- predict(gamfit, data)

plot_data <- data.frame(Time=data$Time, Actual=data$Mortality, Estimate=as.numeric(yhat))
plot_data <- melt(plot_data, id="Time", value.name="Mortality", variable.name="Data")

ggplot(plot_data) +
    geom_line(aes(x=Time, y=Mortality, color=Data))
summary(gamfit)
plot(gamfit)
k <- length(unique(data$Week)) - 1
penalty_values <- c(0, 10, 1000, 100000)

plots <- list()

for (i in 1:length(penalty_values)) {
    fit <- gam(Mortality ~ Year + s(Week, k=k, sp=penalty_values[i]),
               family=gaussian, data=data, method="GCV.Cp")

    title <- paste("Params (k=", k, ",sp=", penalty_values[i], ")", sep="")

    plot_data <- data.frame(Time=data$Time, Actual=data$Mortality, Estimate=fitted(fit))
    plot_data <- melt(plot_data, id="Time", value.name="Mortality", variable.name="Data")

    plots[[i]] <- ggplot(plot_data) +
        geom_line(aes(x=Time, y=Mortality, color=Data), show.legend=FALSE) +
        ggtitle(title) +
        theme(axis.text=element_blank())
}

do.call(grid.arrange, c(plots, list(ncol=2)))
gamfit <- gam(Mortality ~ Year + s(Week), family=gaussian, data=data, method="GCV.Cp")
residuals <- resid(gamfit)
plot_data <- data.frame(Time=data$Time, Residuals=residuals, Influenza=data$Influenza)
plot_data <- melt(plot_data, id="Time", value.name="Value", variable.name="Data")

ggplot(plot_data) +
    geom_line(aes(x=Time, y=Value, color=Data))
gamfit <- gam(Mortality ~ s(Year, k=length(unique(data$Year)) - 1) +
                  s(Week, k=length(unique(data$Week)) - 1) +
```

```
                s(Influenza, k=length(unique(data$Influenza))),
            data=data)
summary(gamfit)

plot_data <- data.frame(Time=data$Time, Actual=data$Mortality, Estimate=fitted(gamfit))
plot_data <- melt(plot_data, id="Time", value.name="Value", variable.name="Data")

ggplot(plot_data) +
    geom_line(aes(x=Time, y=Value, color=Data))
```