Homework 2

Gianni Spiga

2023-01-23

Contents

Problem Set 1

Question 6

a.)

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-76.1070019	6.0198323	-12.6427114	0.0000000
years	0.0405879	0.0030747	13.2005736	0.0000000
sunspotnumber	0.0005740	0.0005996	0.9573381	0.3383966

We can see that the predictors have an effect on the response since we see non-zero values for the estimates.

b.)

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	36	209.42068	NA
years	1	186.0507659	35	23.36991	0.0000000
sunspotnumber	1	0.9106512	34	22.45926	0.3399417

We can see from the χ^2 test that the number of sun spots is not significant at any reasonable α, however the year the melanoma have been diagnosed is highly significant.

c.)

We test the following hypothesis for $i = 1, 2, 3$:

$$H_0 : \beta_i = 0 H_a : \beta_i > 0$$

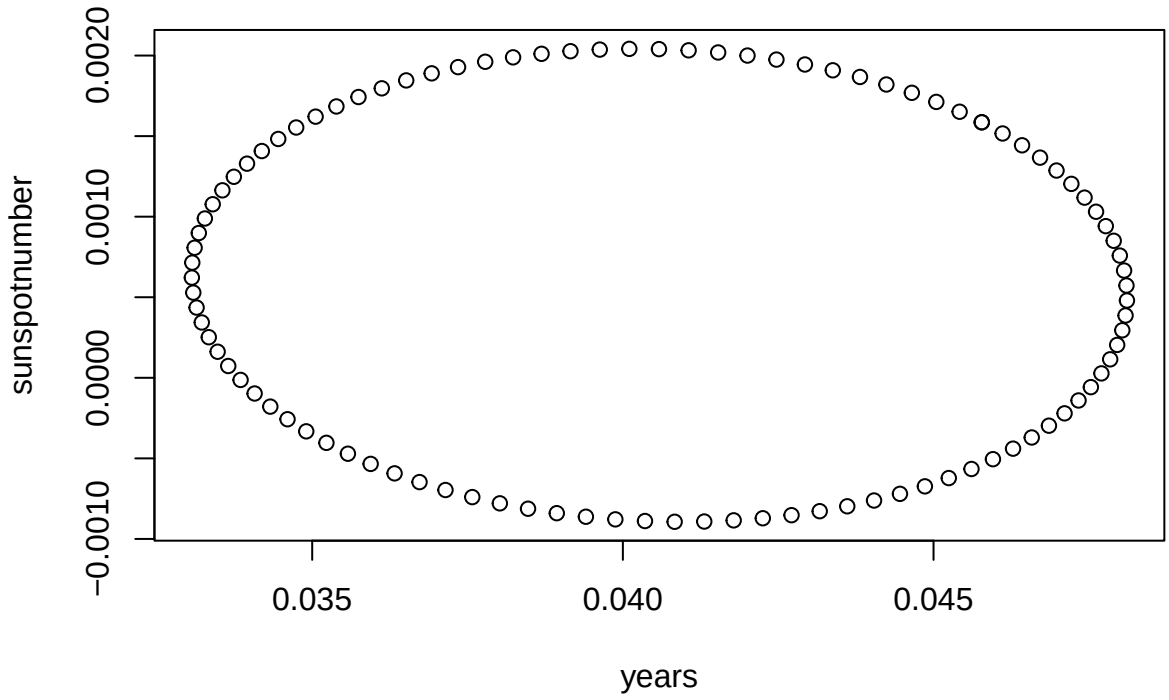
```
## The p-value for the intercept is 1

## The p-value for Years is 4.328902e-40

## The p-value for the Number of Sunspots is 0.1692836
```

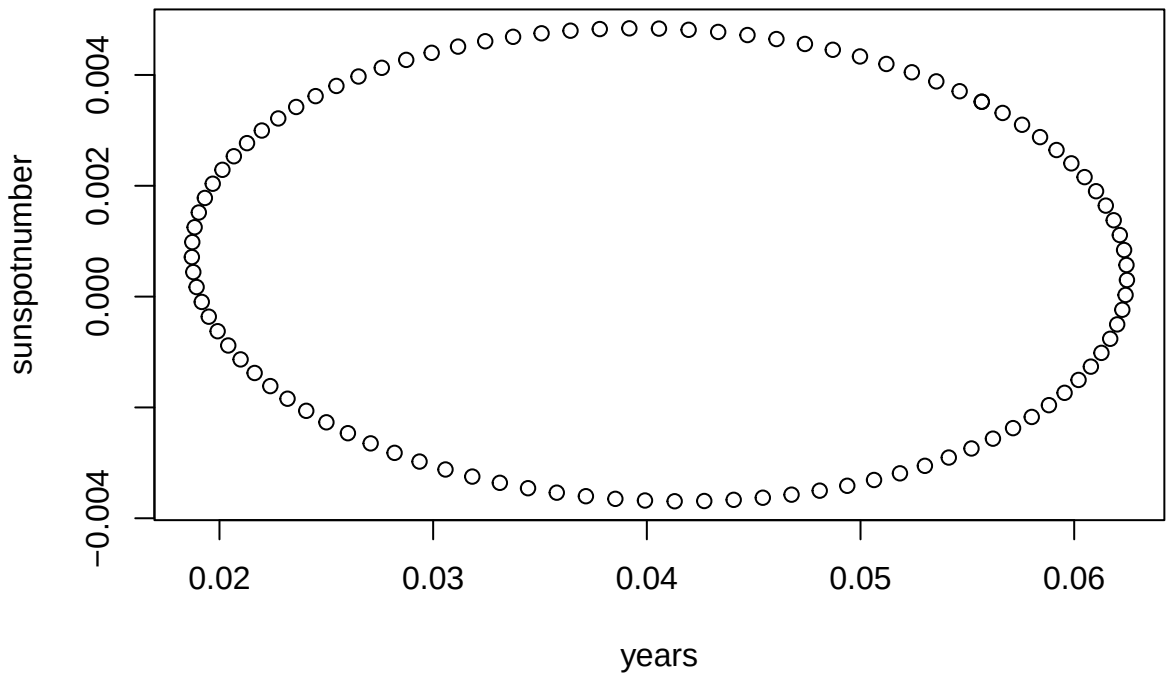
We can see for both the intercept and the years, that we reject our null hypothesis, concluding that we have significant evidence that the number of melanoma increases over the calendar years. However, we do not have evidence to conclude that the number of sunspots increases with the number of melanomas.

Question 7



$$H_0 : \beta_1 = \beta_2 = 0 H_a : \beta_1 \neq \beta_2 \neq 0$$

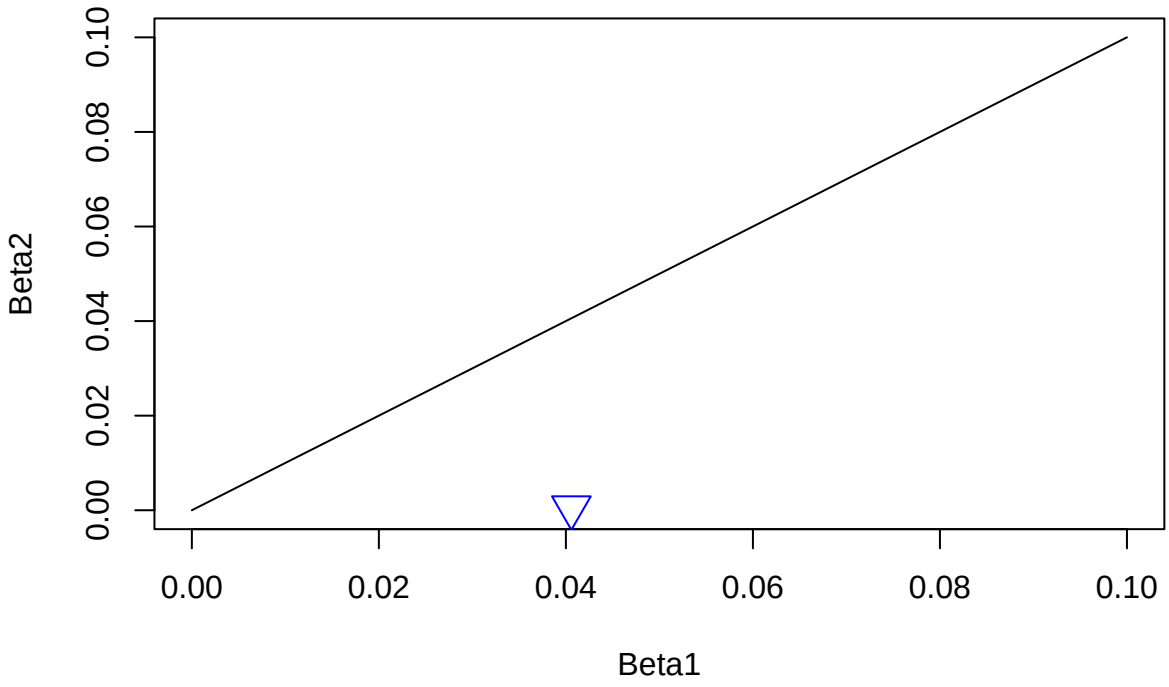
We could determine a p-value to test the null hypothesis is the minimum value of alpha for the null hypothesis to be true i.e. the null hypothesis is within the confidence ellipsoid. However, the following ellipsoid below is a $1 - \alpha = 0.99999999999$ confidence level. The point $(0, 0)$ is encodded to have a blue triangle on it.



However, the point of interest is not within graphing visibility of our ellipsoid. Since we are already at such a small level of α, we can conclude that the p-value is very close to 0, leading us to reject this null hypothesis at any reasonable significance level.

Question 8

$$H_0 : \beta_1 \leq \beta_2 H_a : \beta_1 > \beta_2$$



Since we can see that our observed $\hat{\beta_1} > \hat{\beta_2}$ is below the line $\beta_1 = \beta_2$, we have reason to continue testing our hypothesis. We can then use a chi square test.

	years	sunspotnumber
years	9.5e-06	-1e-07
sunspotnumber	-1.0e-07	4e-07

2.3e-06

We solve for a p-value of 0.0000023, leading us to reject the null and conclude we have statistically sufficient evidence that $\beta_1 > \beta_2$.

Question 9

a.)

One example of a consistent estimator for $\hat{m}(x)$ is the Kernel density estimator. In the paper, "Consistency of the kernel density estimator - a survey", Weid and Weißbach prove that the Kernel Density Estimator is is almost sure convergent, which is stronger than convergence in probability, the requirement for consistency of an estimator.

b.)

We can use the quotient type parameter for smoothing. The bias and variance depend since by definition of consistency, we expect as n approached infinity, the variance will approach zero. Because of this, we must keep in mind the trade off there is with the variance and bias when choosing.

c.)

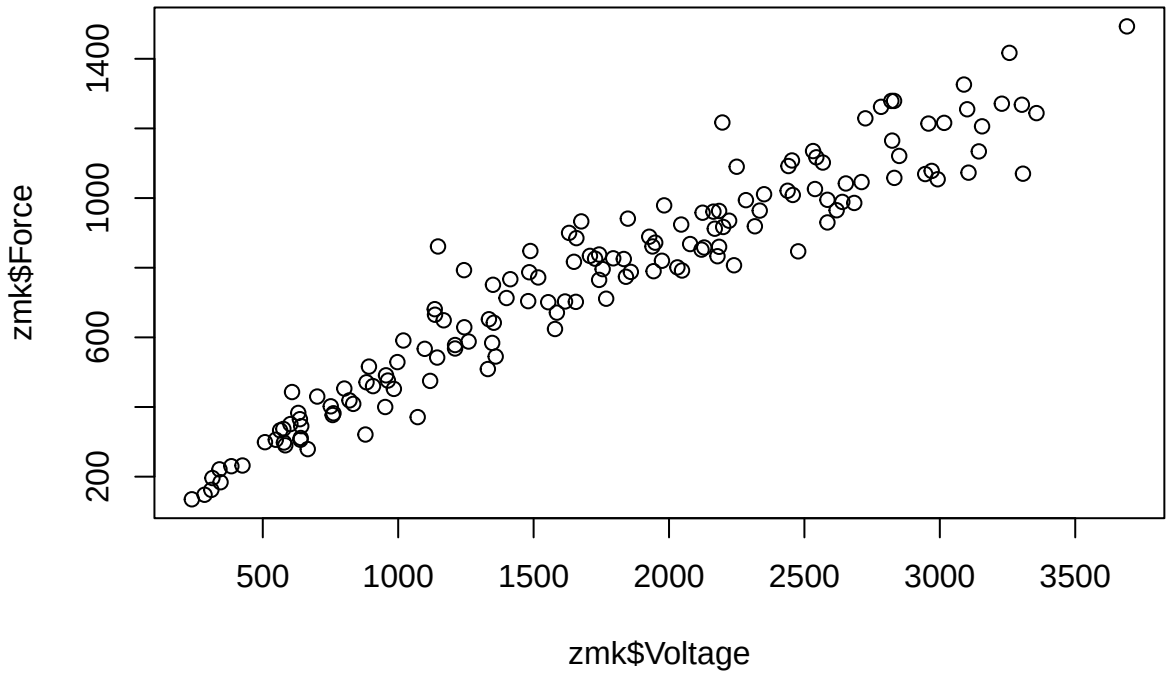
d.)

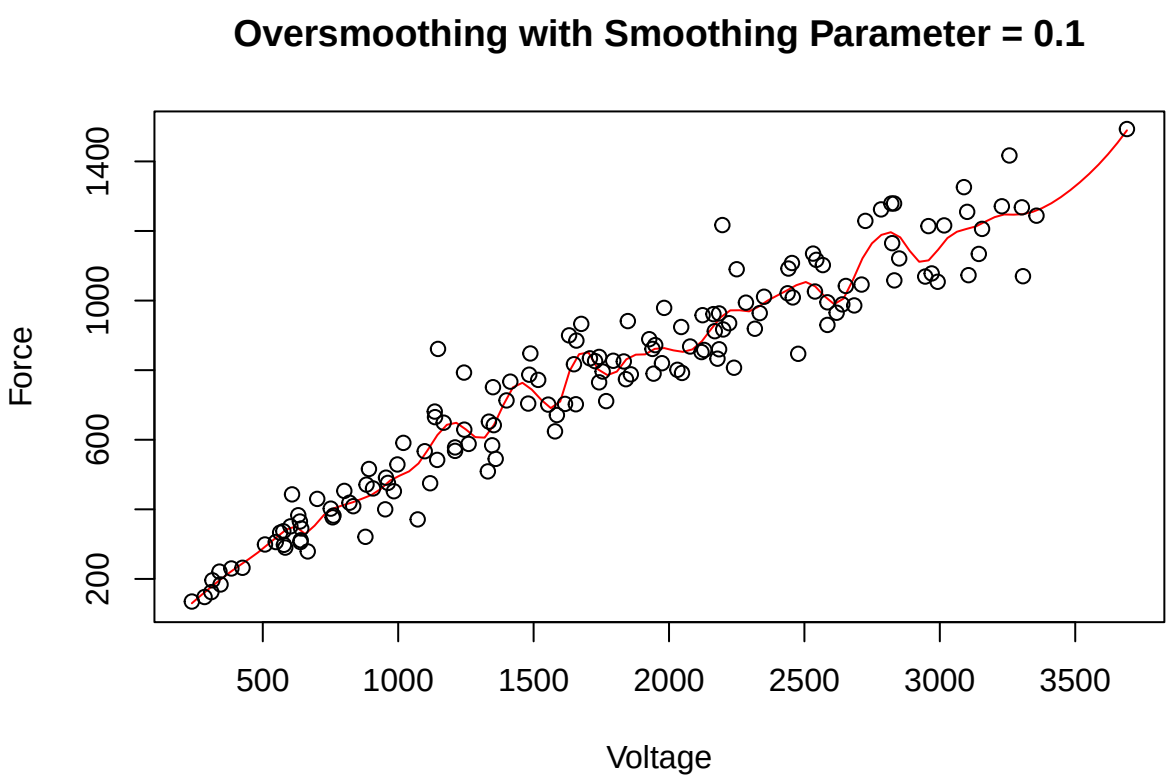
The function $v(x)$ would also be consistent since it is a combination of the $E(Y|X = x)$ and $E(Y^2|X = x)$, which is also a consistent estimator by the continuous mapping theorem.

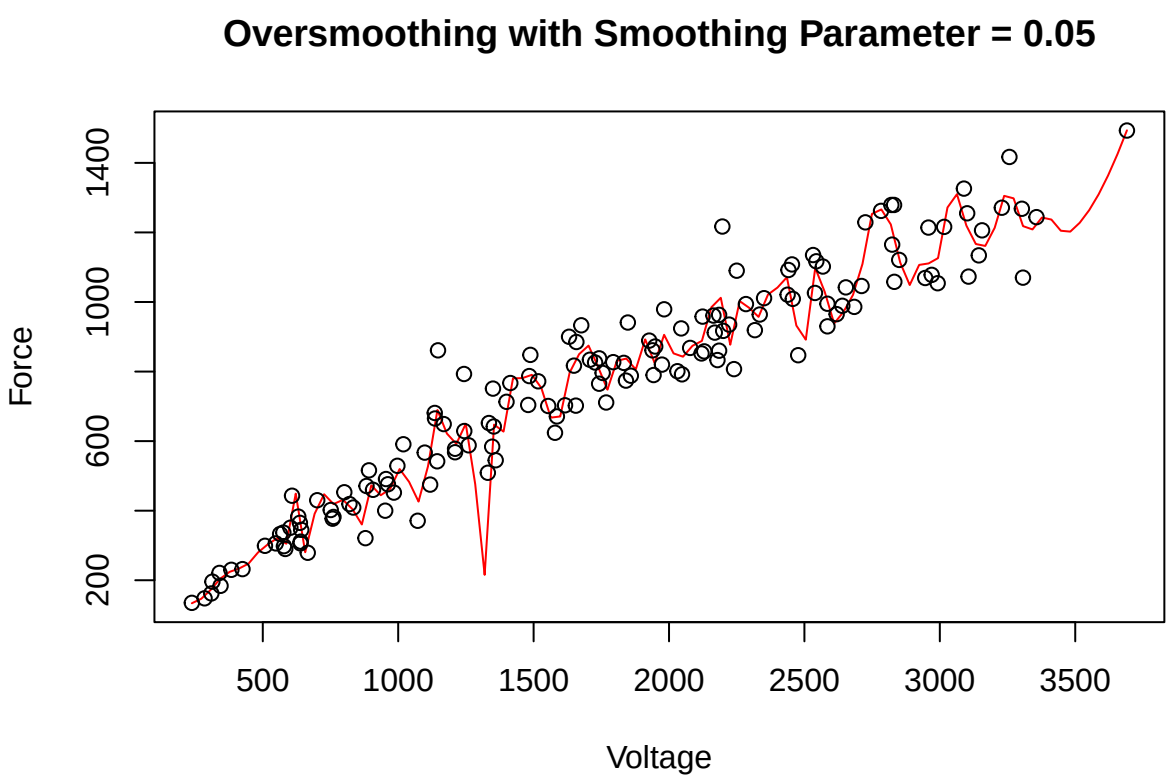
Question 13

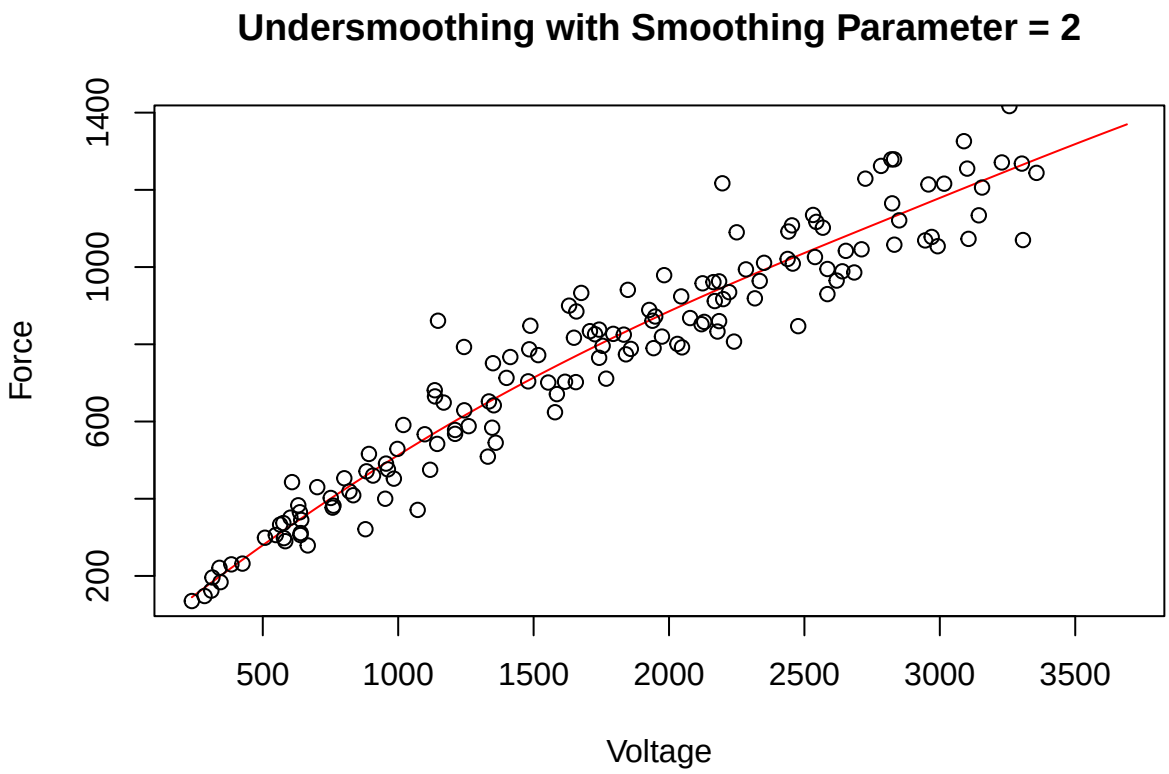
a.)

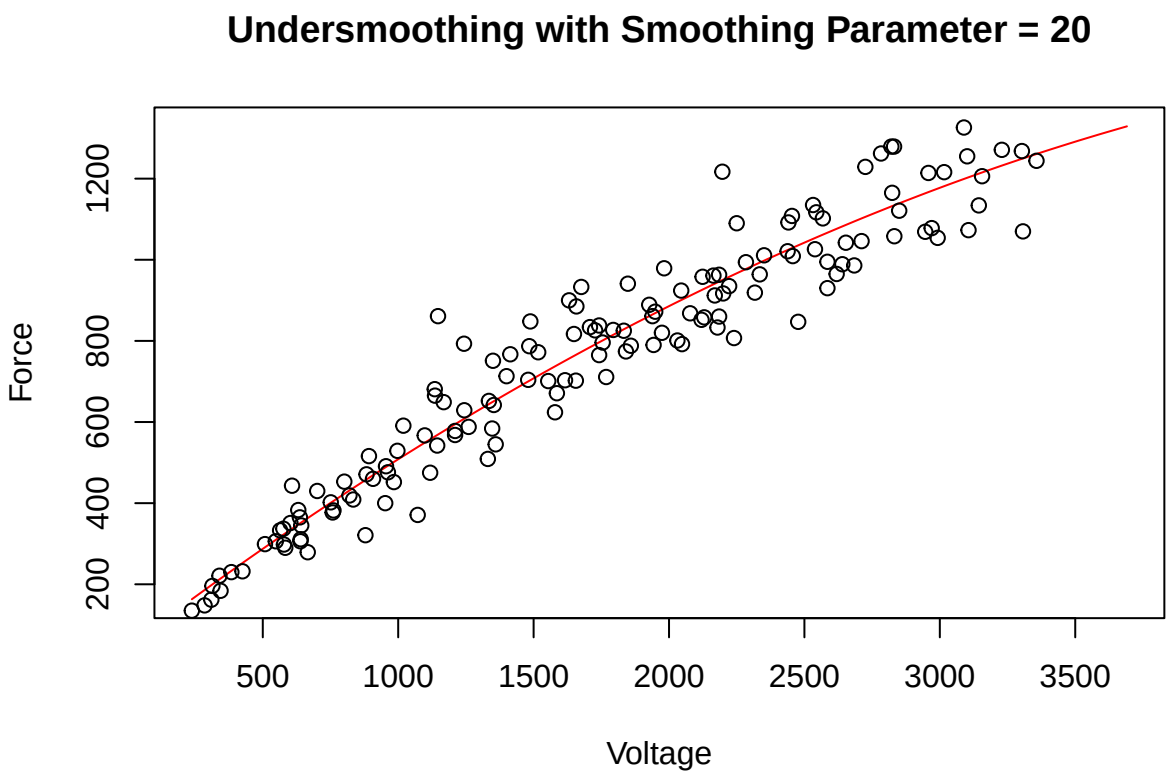
Below we can see 5 graphs, the first one is our data with no fitting, the second and third are with over smoothing and the final two are with under smoothing for the data.



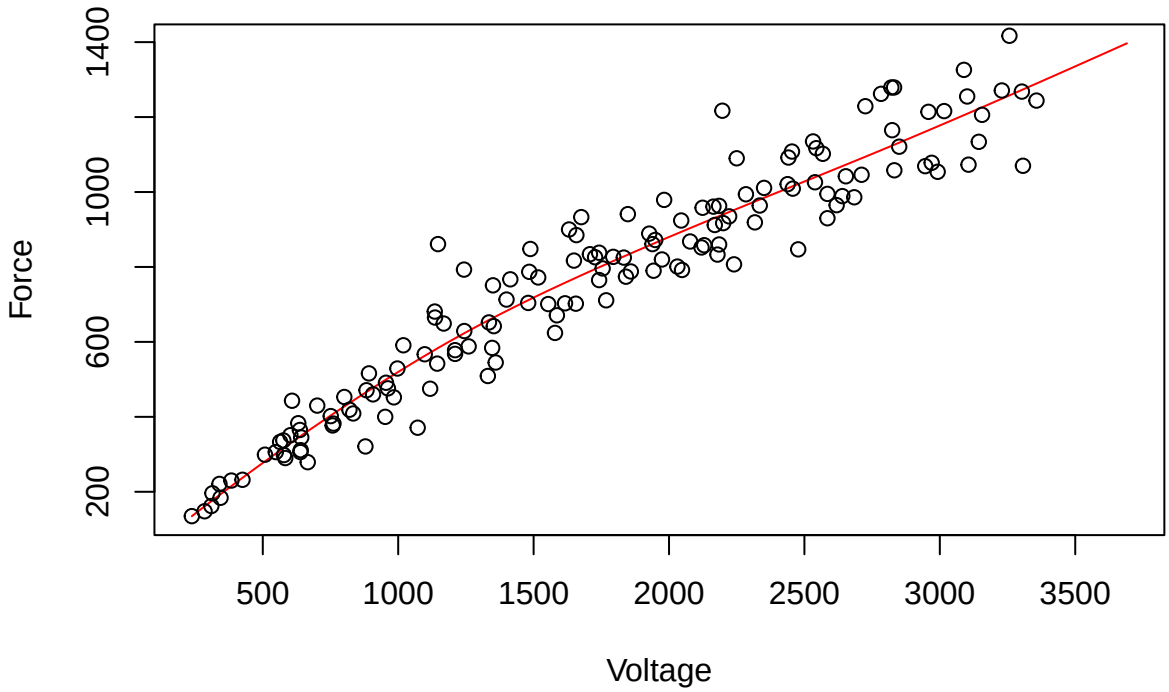

Oversmoothing with Smoothing Parameter = 0.1
Force
200
600
1000
1400
500
1000
1500
2000
2500
3000
3500
Voltage


Oversmoothing with Smoothing Parameter = 0.05
Force
200
600
1000
1400
500
1000
1500
2000
2500
3000
3500
Voltage


Undersmoothing with Smoothing Parameter = 2
Force
200
600
1000
1400
500
1000
1500
2000
2500
3000
3500
Voltage


Undersmoothing with Smoothing Parameter = 20
Force
200
400
600
800
1200
500
1000
1500
2000
2500
3000
3500
Voltage

b.)



c.)

The strategy for finding a good fit is to visualzie the smoothing fits over the points and find the curve that accurately fits the data without fitting too many indivudal points and still showing visual changes in directions of the data.

Question 19

See handwritten portion

Problem Set 2

Questions 1-3

See handwritten portion

Question 6

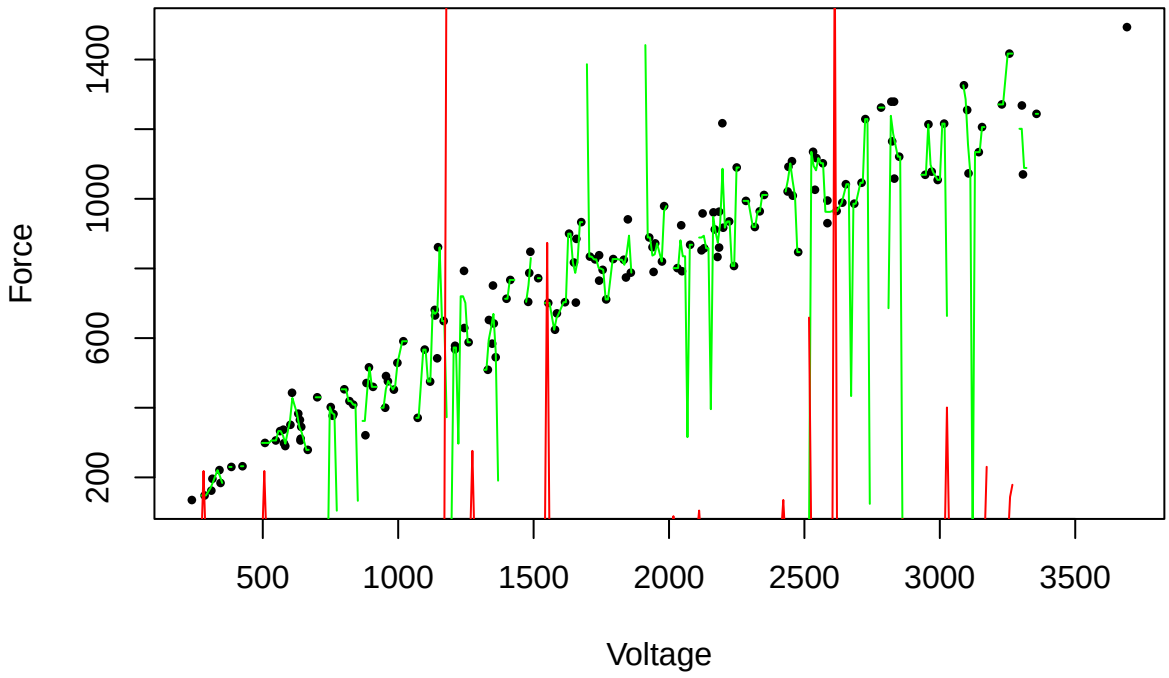
a.)

For the k-th derivative, we can use the estimator $k!\hat{\beta}_k$. This would be an estimator that would be simple to write and relatively easy to compute for reasonable values of k. If one were to look for a large k-th derivative however, they could run into potential computational slow down due to the high growth rate of the factorial function.

b.)

There is multiple methods of choosing the bandwidth. One example is the leave-one-out cross validation, however this can get computationally expensive as n grows large, so a slightly better option would be to do m-fold cross validation. Any value of m less than $n-1$ would be computationally faster. If we did not want to use so much computational power, and were not interested in finding the most optimal value, we could perform subjective choosing as well.

c.)



We can see here that the locpoly() function is overfitting, however, the bandwidth is at the minimum it can be set before creating NaNs, suggesting that polynomial fitting is not the best choice for this data.

Appendix

```
library(xtable)
library(knitr)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
mela <- read.table("melanoma.txt", header = TRUE)
#xtable(head(mela))
glm.mela <- glm(totalincidence ~ years + sunspotnumber, data = mela, family = poisson())
sum.mela <- summary(glm.mela)
#print(xtable(summary(glm.mela)), type = "latex", comment= FALSE)
kable(sum.mela$coefficients)
#kable(summary(glm.mela))
kable(anova(glm.mela, test = "Chisq"))
# Intercept
cat("The p-value for the intercept is", pnorm(-12.643, lower.tail = FALSE), "\n")

# Years
cat("The p-value for Years is", pnorm(13.201, lower.tail = FALSE), "\n")

# Sunspot number
cat("The p-value for the Number of Sunspots is",pnorm(0.957, lower.tail = FALSE))

# Find the variance covariance matrix
mela.vmat <- vcov(glm.mela)

# Find the eigen values
mela.evals <- eigen(mela.vmat)$values

# get the eigenvalues corresponding to the variables (no intercept)
mela.evals <- mela.evals[2:3]

library(ellipse)
{plot(ellipse(glm.mela, which = c(2,3), level = 0.95))
points(x = c(0), y = c(0), cex = 2, pch = 6, col = "blue")}
{plot(ellipse(glm.mela, which = c(2,3), level = 0.99999999999))
points(x = c(0), y = c(0), cex = 2, pch = 6, col = "blue")}
### QUESTION 8
x <- seq(0,.1, 0.01); y <- seq(0,.1, 0.01)
{plot(x,y, type = 'l', xlab = "Beta1", ylab = "Beta2")
points(glm.mela$coefficients[2], glm.mela$coefficients[3],cex = 2, pch = 6, col = "blue")
}
info <- vcov(glm.mela)[2:3, 2:3]
kable(info)

betahat <- c(glm.mela$coefficients[2], glm.mela$coefficients[3])


# We know that Sigma is equal to inverse of finitely estimated information matrix, so the inverse of si
chi.stat <- (betahat - c(1,-1)) %*% info %*% (betahat - c(1,-1))

# We print the p-value for the one sided test
kable(pchisq(chi.stat, df = 2, lower.tail = TRUE) / 2)
zmk <- read.table("zmk.txt", header = FALSE)
names(zmk) <- c("Voltage", "Force")
```

```r
plot(zmk$Voltage, zmk$Force)
library(locfit)

fit1over <- locfit(Force ~ lp(Voltage, nn = 0.1), data = zmk)
fit2over <- locfit(Force ~ lp(Voltage, nn = 0.05), data = zmk)
#summary(fit1)


plot(fit1over, col = "red", main = "Oversmoothing with Smoothing Parameter = 0.1")
points(zmk$Voltage, zmk$Force)
plot(fit2over, col = "red", main = "Oversmoothing with Smoothing Parameter = 0.05")
points(zmk$Voltage, zmk$Force)


# Now we plot with underfitting
fit1under <- locfit(Force ~ lp(Voltage, nn = 1), data = zmk)
fit2under <- locfit(Force ~ lp(Voltage, nn = 20), data = zmk)
#summary(fit1)


plot(fit1under, col = "red", main = "Undersmoothing with Smoothing Parameter = 2")
points(zmk$Voltage, zmk$Force)
plot(fit2under, col = "red", main = "Undersmoothing with Smoothing Parameter = 20")
points(zmk$Voltage, zmk$Force)
fit.good <- locfit(Force ~ lp(Voltage, nn = 0.8), data = zmk)
plot(fit.good, col = "red")
points(zmk$Voltage, zmk$Force)
library(KernSmooth)
###  This is not fitting
# fitpolyMean <-
#   locpoly(x = zmk$Voltage, y = zmk$Force, bandwidth = 0.9,gridsize = 1000, drv = 0)
# fitpolyDeriv <-
#   locpoly(x = zmk$Voltage, y = zmk$Force, bandwidth = 75, gridsize = 500, drv = 1)
# #plot(fit1poly, type = "l")
#
# with(zmk, {
#   plot(
#     Voltage,
#     Force,
#     ylim = range(fitpolyDeriv),
#     pch = 16,
#     cex = 0.6
#   )
#   lines(fitpolyMean$x, fitpolyMean$y, col = 'green')
#   lines(fitpolyDeriv$x, fitpolyDeriv$y, col = 'red')
# })



# Trying cleaner code with no gridsize adjustment
mean = locpoly(zmk$Voltage, zmk$Force, bandwidth = 2.18, drv = 0)
deriv = locpoly(zmk$Voltage, zmk$Force, bandwidth = 5.7, drv = 1)
with(zmk, {
```

```
  plot(Voltage,
       Force,
       #ylim = range(deriv),
       pch = 16,
       cex = 0.6)
  lines(mean$x, mean$y, col = 'green')
  lines(deriv$x, deriv$y, col = 'red')
})
```