# HW4

Thibaud Bruyelle

10/30/2020

## Question 7

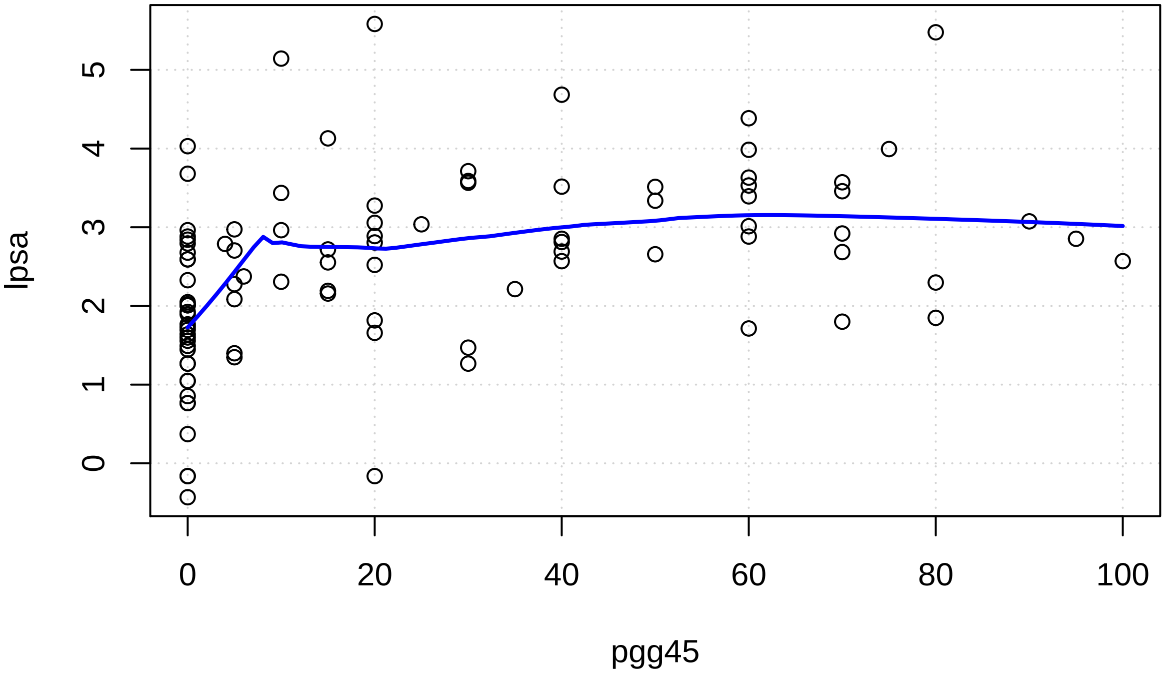
```r
# weight function
W <- function(r){
  res <- ifelse(abs(r)<1, (1-abs(r)^3)^3, 0)
  return(res)
}
```

```r
# the following function computes the local linear regression fit in x0
llr <- function(x, y, x0, span, hat = FALSE){

  n = length(x)
  # compute w_s
  k = floor(span * n)
  w_s = sort(abs(x-x0))[k] # never compute xk
  # FIXME : what if w_s = 0?

  # compute the model by computing each component of vec h(x0)
  m0 = sum(W(abs(x-x0)/w_s))
  n0 = sum(W(abs(x-x0)/w_s) * x)
  p0 = sum(W(abs(x-x0)/w_s) * x^2)
  delta0 = m0 * p0 - n0^2
  h = W(abs(x-x0)/w_s) * (1/delta0) * (p0 - x0*n0 + x*(m0 * x0 - n0))


  if (hat){
    res = list("h" = h, "y0_fit" = t(h) %*% y)
  }
  else{
    res = list("y0_fit" = t(h) %*% y)
  }
  return(res)
}
```

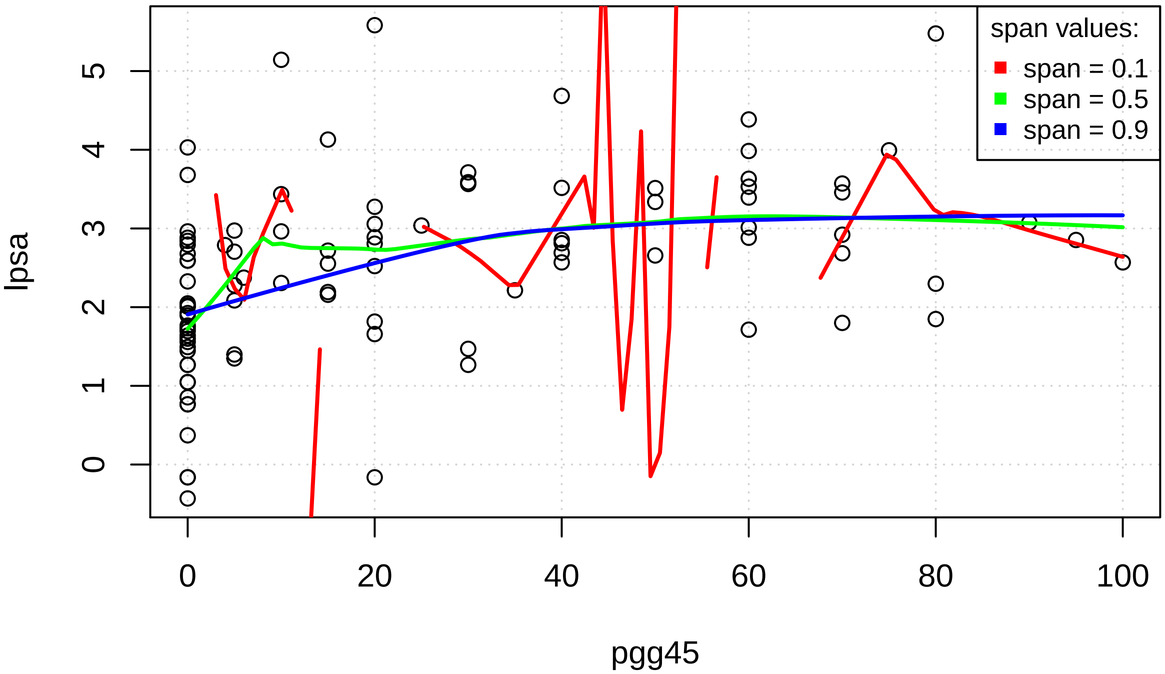
## Question 8

```r
# import data
p <- '/Users/thibaudbruyelle/Documents/Stanford/fall2020/stats305A/datasets/lprostate.dat'
data <- read.table(p, header = T)[,-1]
x = data$pgg45
y = data$lpsa
```

```
# run llr
span = 0.5
x0_seq = seq(min(x), max(x), length.out = 100)
y_fit = c()
for (i in 1:length(x0_seq)){
  mod_i = llr(x, y, x0_seq[i], span)
  y_fit = c(y_fit, as.numeric(mod_i[[1]]))
}

plot(x,y, xlab = 'pgg45', ylab = 'lpsa', panel.first=grid())
lines(x0_seq, y_fit, col = 'blue', lwd = 2)
```



```
generate_y_fit <- function(x, y, x0_seq, span){
  y_fit = c()
  for (i in 1:length(x0_seq)){
    mod_i = llr(x, y, x0_seq[i], span)
    y_fit = c(y_fit, as.numeric(mod_i[[1]]))
  }

  return(y_fit)
}

y_fit_span_1 = generate_y_fit(x, y, x0_seq, 0.1)
y_fit_span_2 = generate_y_fit(x, y, x0_seq, 0.5)
y_fit_span_3 = generate_y_fit(x, y, x0_seq, 0.9)

plot(x,y, xlab = 'pgg45', ylab = 'lpsa', panel.first=grid())
lines(x0_seq, y_fit_span_1, col = 'red', lwd = 2, type ='l')
lines(x0_seq, y_fit_span_2, col = 'green', lwd = 2)
lines(x0_seq, y_fit_span_3, col = 'blue', lwd = 2)
legend('topright', legend = c("span = 0.1", "span = 0.5", "span = 0.9"),
       col = c("red", "green", "blue"), title = 'span values: ', cex = 0.8, pch = 15)
```



```
# lines(x0_seq, y_fit_span_1, col = 'blue', lwd = 2)
```

After having tested multiple values of `span` within the range $[0, 1]$, `span = 0.5` seems to be a good value. We will provide more details about the span selection in question 10.

### Question 9