Local polynomial regression

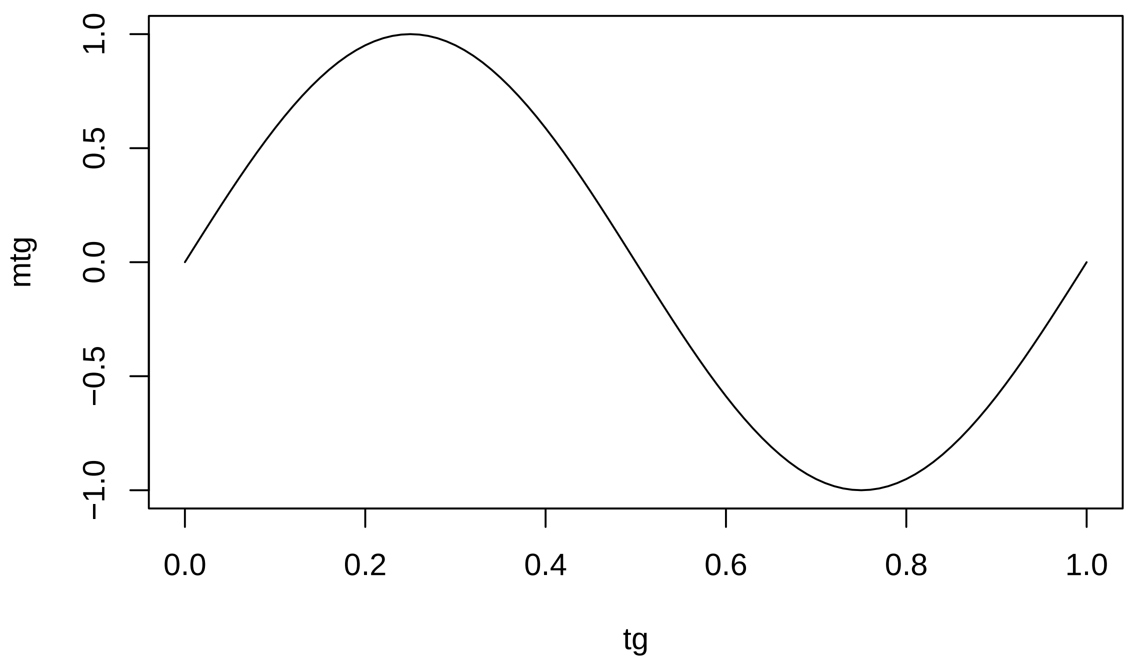
Write your own local linear regression function

Pedro Delicado

2023-10-23

Ploting the function $m(t) = \sin(2\pi t)$, $t \in [0, 1]$.

```
tg <- seq(0,1,by=.01) # Regular grid of values t
nt <- length(tg)       # number of points t in the regular grid
mtg <- sin(2*pi*tg)    # m(tg), true regression function
plot(tg, mtg, type="l")
```

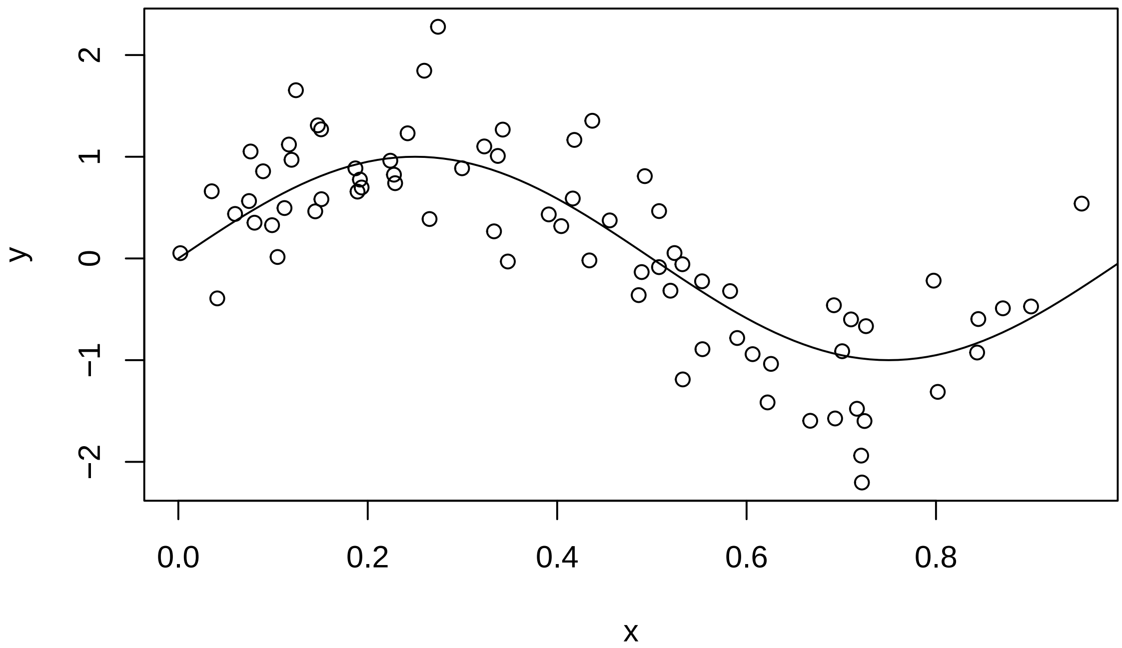


Generating data according to the regression model

$$y = m(x) + e$$

```
n <- 75        # number of observed data
sigma.e <- .5 # standard deviation of the random noise
x <- runif(n)
mx <- sin(2*pi*x)
y <- mx + rnorm(n,0,sigma.e)
```

```r
plot(x,y)
lines(tg,mtg)
```



Write your own local linear regression code for estimating $m(t)$ at $t = 0.4$.

The pseudo-code should be like that:

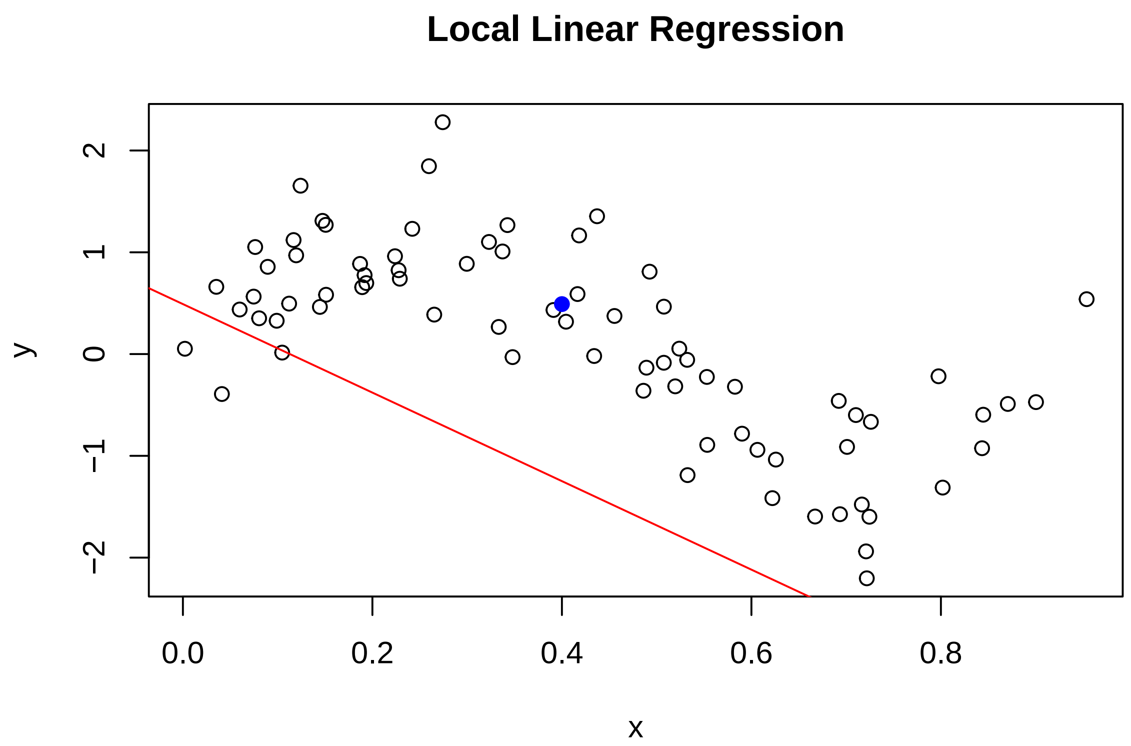
0. `t <- 0.4`
1. Decide the bandwidth value: $h = 0.1$, for instance.
2. Define `x.t <- x - t` (you'll find it usefull in the next step).
3. Compute the weight of each x_i in `x` as $K((x_i - t)/h)$, where K is a density function symmetric around 0 (use, for instance, `dnorm`).
4. Fit the weighted linear model $y \sim (x - t)$ with the weights computed before.
5. Take the intercept of the preceding fitted linear model as estimation $\hat{m}(t)$ of $m(t)$.
6. Add to the last graphic the estimated point $(t, \hat{m}(t))$.

```r
t <- 0.4
h <- 0.1
x.t <- x - t

# Step 3: Compute the weight of each xi in x
K <- dnorm(x.t, mean = 0, sd = h)  # Using the standard normal density function (dnorm)

# Step 4: Fit the weighted linear model
weighted_lm <- lm(y ~ x.t, weights = K)

# Step 5: Estimate m(t) (intercept of the fitted model)
m_hat <- coef(weighted_lm)[1]
```

```r
# Print the estimated value of m(t)
cat("Estimated m(t) at t =", t, "is:", m_hat, "\n")
```

```
## Estimated m(t) at t = 0.4 is: 0.4909914
```

```r
# Step 6: Plot the estimated point (t, m_hat(t))
plot(x, y, main = "Local Linear Regression", xlab = "x", ylab = "y")
abline(weighted_lm, col = "red")  # Add the weighted linear fit line
points(t, m_hat, col = "blue", pch = 19)  # Estimated point (t, m_hat(t))
```



Write a script in R that fits a local polynomial regression of each element of vector tg.

The pseudo-code should be like that:

1. Decide the bandwidth value: $h = 0.1$, for instance.
2. For each t in `tg` do: (it can be useful to define `x.t <- x - tg[j]`)
   i. Compute the weight of each x_i in `x` as $K((x_i - t)/h)$, where K is a density function symmetric around 0 (use, for instance, `dnorm`).
   ii. Fit the weighted linear model $y \sim (x - t)$ with the weights computed before.
   iii. Take the intercept of the preceding fitted linear model as estimation $\hat{m}(t)$ of $m(t)$.
3. Add to the last graphic the estimated function $\hat{m}(t)$, for $t \in$ `{tg}`.