Modèles Linéaires Appliqués

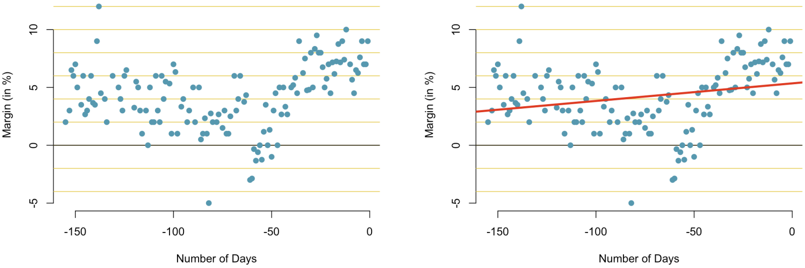
Arthur Charpentier

Automne 2020

OLS #18 (lissage)

Natura non facit saltus

We want a continuous function... but probably not linear...
Data source: http://www.pollster.com/08USPresGEMvO-2.html
pollsters for the popular vote between Obama and McCain (2008 US presidential election), last 150 days.

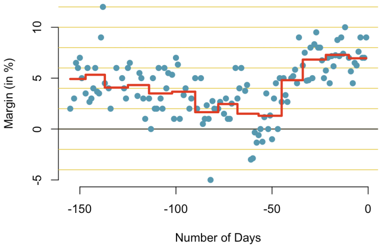


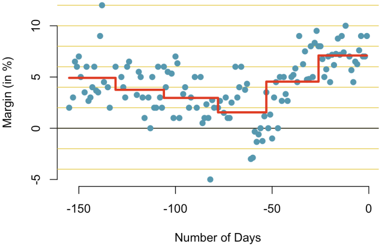

```
> library(dslabs)
> data("polls_2008")
> plot(polls_2008$day, polls_2008$margin*100)
```

Regressogram

From Tukey (1961) Curves as parameters, and touch estimation, the regressogram is defined as

$$\hat{m}_{\mathbf{a}}(x) = \frac{\sum_{i=1}^{n} \mathbf{1}(x_i \in [a_j, a_{j+1}))y_i}{\sum_{i=1}^{n} \mathbf{1}(x_i \in [a_j, a_{j+1}))}$$



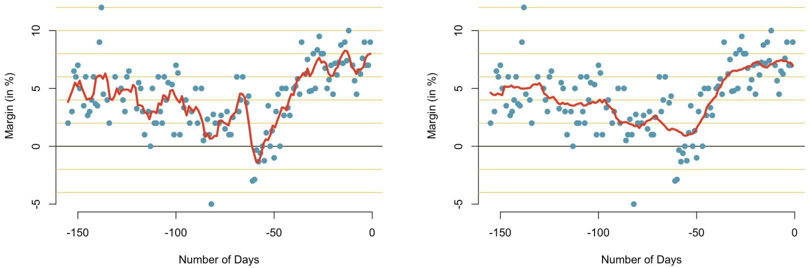



```
> reg=lm(margin~cut(day,seq(-160,0,length=15)),data=
    polls_2008)
```

Moving Regressogram

and the moving regressogram is

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} \mathbf{1}(x_i \in [x \pm h_n]) y_i}{\sum_{i=1}^{n} \mathbf{1}(x_i \in [x \pm h_n])}$$




```
> with(polls_2008, ksmooth(day, margin, kernel = "box
    ", bandwidth = 7)
```

with bandwidth h_n (size of the neighborhood around x)

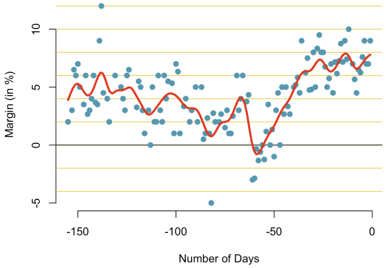
Local Regression

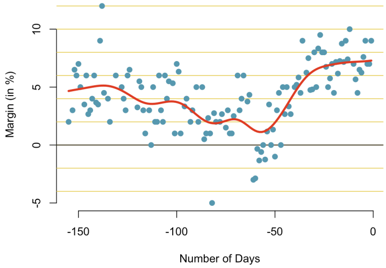
More generally, as moving from the histogram to kernel estimate

$$\tilde{m}(x) = \frac{\sum_{i=1}^{n} y_i \kappa_h(x - x_i)}{\sum_{i=1}^{n} \kappa_h(x - x_i)}$$

Observe that this regression estimator is a weighted average

$$\tilde{m}(x) = \sum_{i=1}^{n} \omega_i(x) y_i \text{ with } \omega_i(x) = \frac{\kappa_h(x - x_i)}{\sum_{i=1}^{n} \kappa_h(x - x_i)}$$





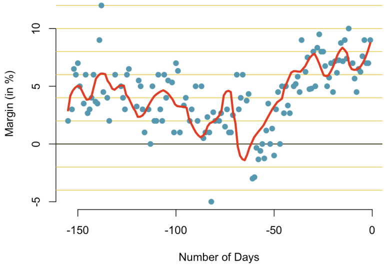
k-Nearest Neighbors

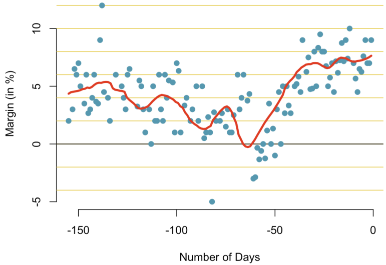
An alternative is to consider

$$\tilde{m}_k(x) = \frac{1}{n}\sum_{i=1}^{n}\omega_{i,k}(x)y_i$$

where $\omega_{i,k}(x) = \dfrac{n}{k}$ if $i \in \mathcal{I}_x^k$ with

$$\mathcal{I}_x^k = \{i : x_i \text{ one of the } k \text{ nearest observations to } x\}$$





Local Regression & k-NN

```
> fit = with(polls_2008, ksmooth(day, margin, kernel =
    "normal", bandwidth = span))
> lines(fit$x, fit$y)
```

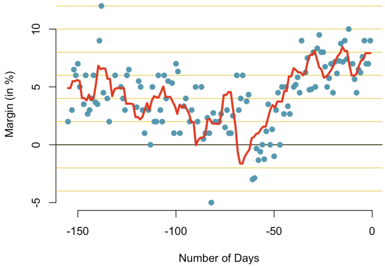
or

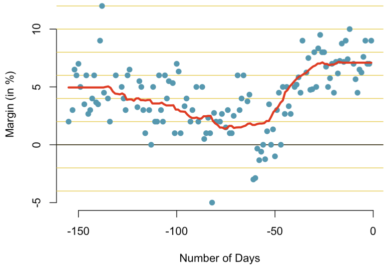
```
> library(FNN)
> p2=knn.reg(train = polls_2008, test = polls_2008, y
   = polls_2008$margin, k = 25)
> lines(polls_2008$day, p2$pred)
```

LOESS (locally weighted polynomial)

Solve

$$\tilde{m}(x) = \text{argmin}\left\{\sum_{i=1}^{n}\omega_i(x)\big(y_i-\alpha-\beta x_i\big)^2\right\},\ \omega_i(x)=\frac{\kappa_h(x-x_i)}{\sum_{i=1}^{n}\kappa_h(x-x_i)}$$

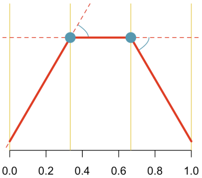





```
> fitL = loess(margin ~ day, degree=1, span = 7, data=
   polls_2008, se=TRUE)
```

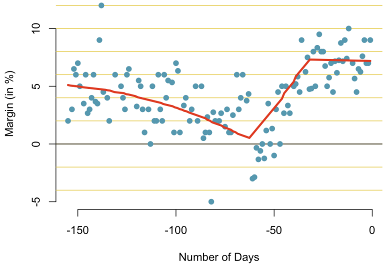
(Linear) Spline Regression

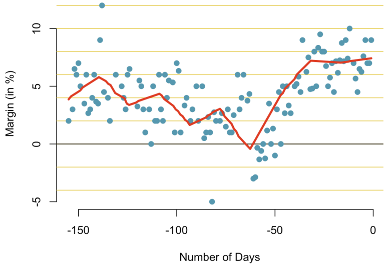
Select some knots $\{s_1, \cdots, s_k\}$, then with $s_0 = 0$



$$\tilde{m}(x) = \alpha + \sum_{j=0}^{k} \beta_j (x - s_k)_+$$

where $(x - s)_+ = (x - s)$ if $x > s$, 0 otherwise



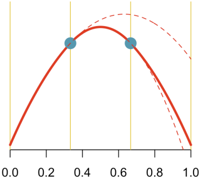



```
> library(splines)
> reg = lm(margin~bs(day, df = 10, degree=1), data=
    polls_2008)
```

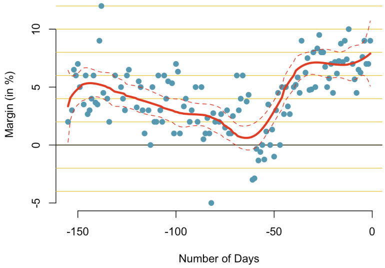
(Quadratic) Spline Regression

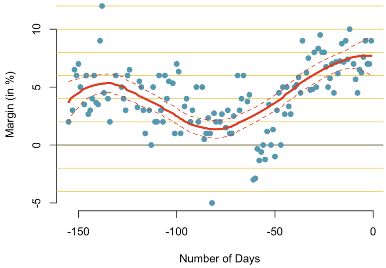
Select some knots $\{s_1, \cdots, s_k\}$, then with $s_0 = 0$

$$\tilde{m}(x) = \alpha + \gamma x + \sum_{j=0}^{k} \beta_j (x - s_k)_+^2$$



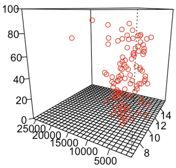
where $(x - s)_+^2 = (x - s)^2$ if $x > s$, 0 otherwise

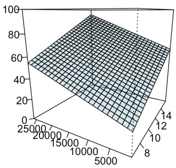


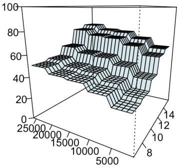


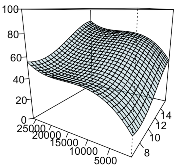

```
> reg = lm(margin~bs(day, df = 10, degree=2), data=
    polls_2008)
```

Can be extended in higher dimension









from the Prestige.txt dataset