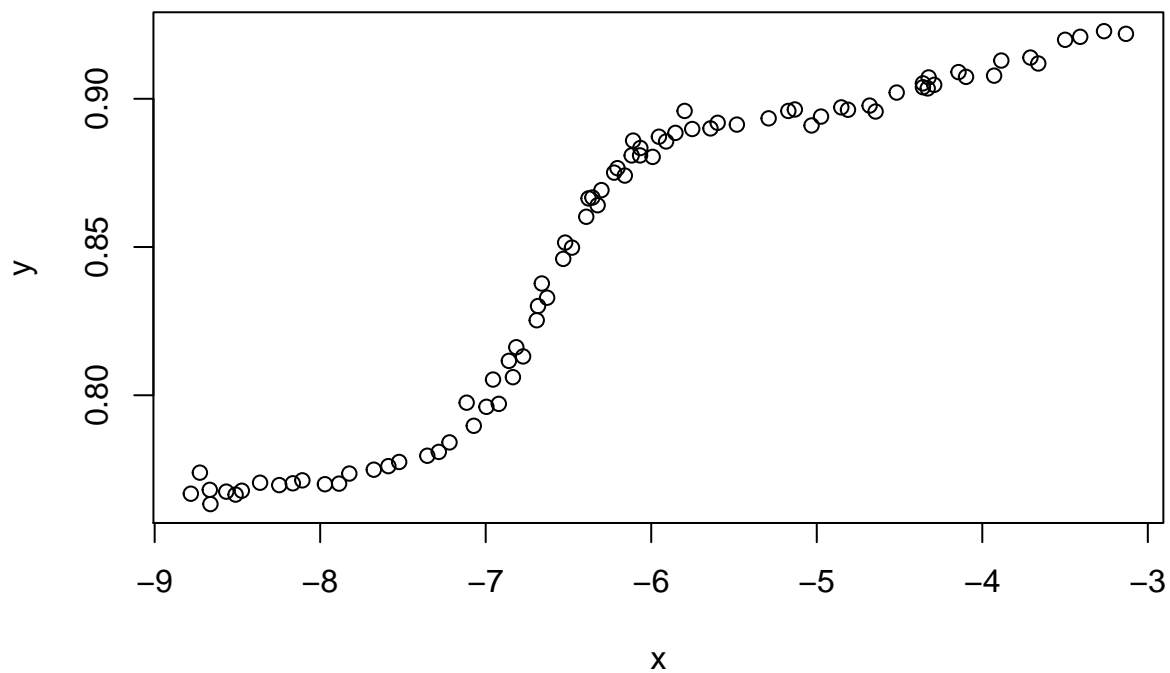


Nist

Realizaremos el ajuste de un polinomio de grado 10 a los datos Nist

```
data <- read.csv("./datos.txt", sep="")
attach(data)
## Función auxiliar para graficar el polinomio resultante
plotPolinomio <- function(x,y, color){
  p <- data.frame(x, y)
  p <- p[order(x), ]
  lines(p, col=color)
}
plot(x, y)
```

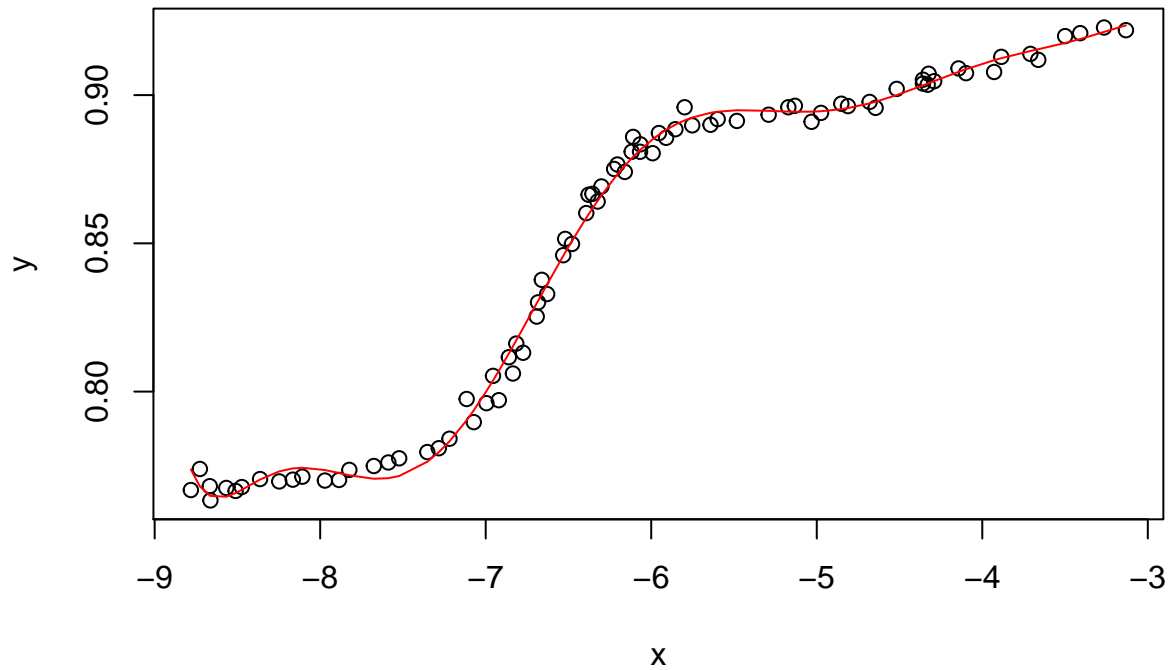


Regresión lineal con un polinomio

```
## Regresión con polinomios
m1 <- lm(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10))
y_hat <- predict(object = m1, newdata = data)

## Warning in predict.lm(object = m1, newdata = data): prediction from a rank-
## deficient fit may be misleading

plot(x,y)
plotPolinomio(x, y_hat, 'red')
```



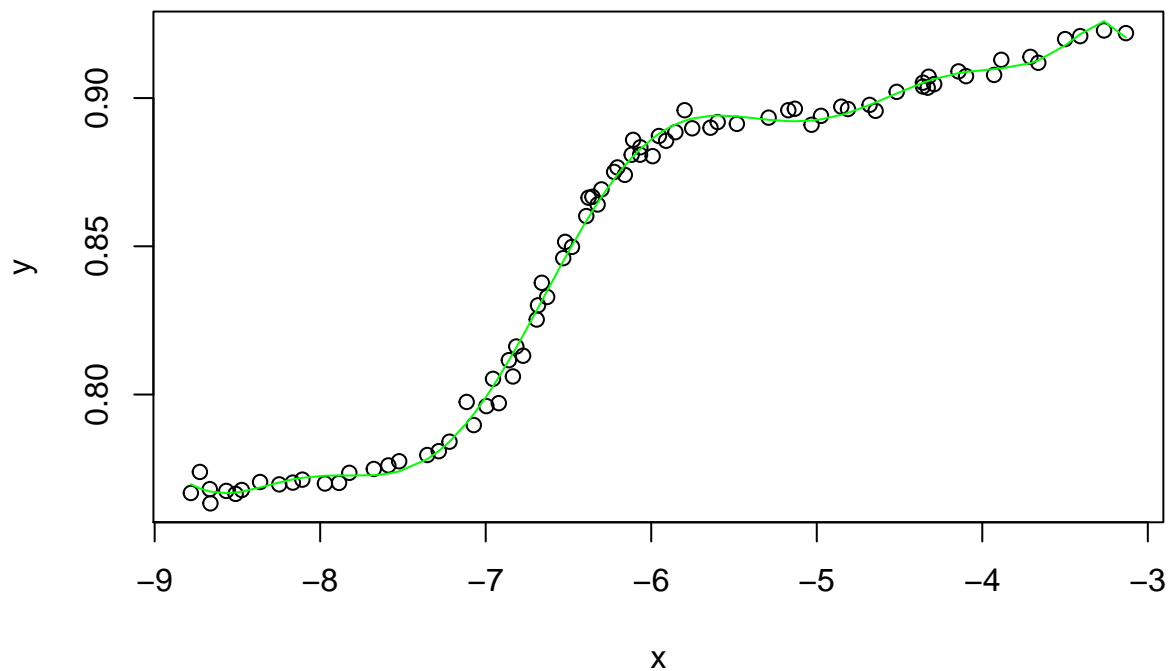
```
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
##      I(x^7) + I(x^8) + I(x^9) + I(x^10))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0099087 -0.0024610  0.0003385  0.0020743  0.0071654
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.743e+02  8.756e+01  -1.990  0.050345 .
## x            -3.269e+02  1.480e+02  -2.208  0.030436 *
## I(x^2)       -2.661e+02  1.095e+02  -2.429  0.017617 *
## I(x^3)       -1.239e+02  4.652e+01  -2.664  0.009534 **
## I(x^4)       -3.638e+01  1.251e+01  -2.907  0.004845 **
## I(x^5)       -6.979e+00  2.211e+00  -3.156  0.002333 **
## I(x^6)       -8.747e-01  2.567e-01  -3.407  0.001079 **
## I(x^7)       -6.906e-02  1.890e-02  -3.654  0.000487 ***
## I(x^8)       -3.118e-03  8.009e-04  -3.894  0.000219 ***
## I(x^9)       -6.139e-05  1.489e-05  -4.123  9.91e-05 ***
## I(x^10)              NA          NA      NA      NA
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003768 on 72 degrees of freedom
## Multiple R-squared:  0.9958, Adjusted R-squared:  0.9953
## F-statistic: 1895 on 9 and 72 DF,  p-value: < 2.2e-16
```

Regresión lineal con polinomios ortogonales

```
## Regresión con polinomios ortogonales
m2 <- lm(y ~ poly(x,10))
y_hat2 <- predict(object = m2, newdata = data)
plot(x,y)
plotPolinomio(x, y_hat2, 'green')
```



```
summary(m2)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 10))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.008804 -0.002176  0.000045  0.002029  0.007096
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8495756  0.0003697 2297.853 < 2e-16 ***
```

```

## poly(x, 10)1    0.4613904  0.0033480  137.810 < 2e-16 ***
## poly(x, 10)2   -0.0867992  0.0033480  -25.926 < 2e-16 ***
## poly(x, 10)3   -0.0826891  0.0033480  -24.698 < 2e-16 ***
## poly(x, 10)4    0.0967433  0.0033480   28.896 < 2e-16 ***
## poly(x, 10)5    0.0174523  0.0033480    5.213 1.75e-06 ***
## poly(x, 10)6   -0.0616874  0.0033480  -18.425 < 2e-16 ***
## poly(x, 10)7    0.0066664  0.0033480    1.991  0.0503 .
## poly(x, 10)8    0.0340241  0.0033480   10.162 1.74e-15 ***
## poly(x, 10)9   -0.0155338  0.0033480   -4.640 1.55e-05 ***
## poly(x, 10)10  -0.0150465  0.0033480   -4.494 2.65e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003348 on 71 degrees of freedom
## Multiple R-squared:  0.9967, Adjusted R-squared:  0.9963
## F-statistic: 2162 on 10 and 71 DF,  p-value: < 2.2e-16

```

Conclusiones

Como notamos de las tablas de resultados, los coeficientes son muy distintos de los coeficientes certificados del Nist. En el primer caso inclusive el último coeficiente queda con valor indeterminado. Al primer momento de obtener estos resultados dudamos de la veracidad y pensamos que había un error, pero al graficar las predicciones a partir del modelo ajustado vemos que efectivamente corresponde a un polinomio que ajusta los datos de forma correcta.