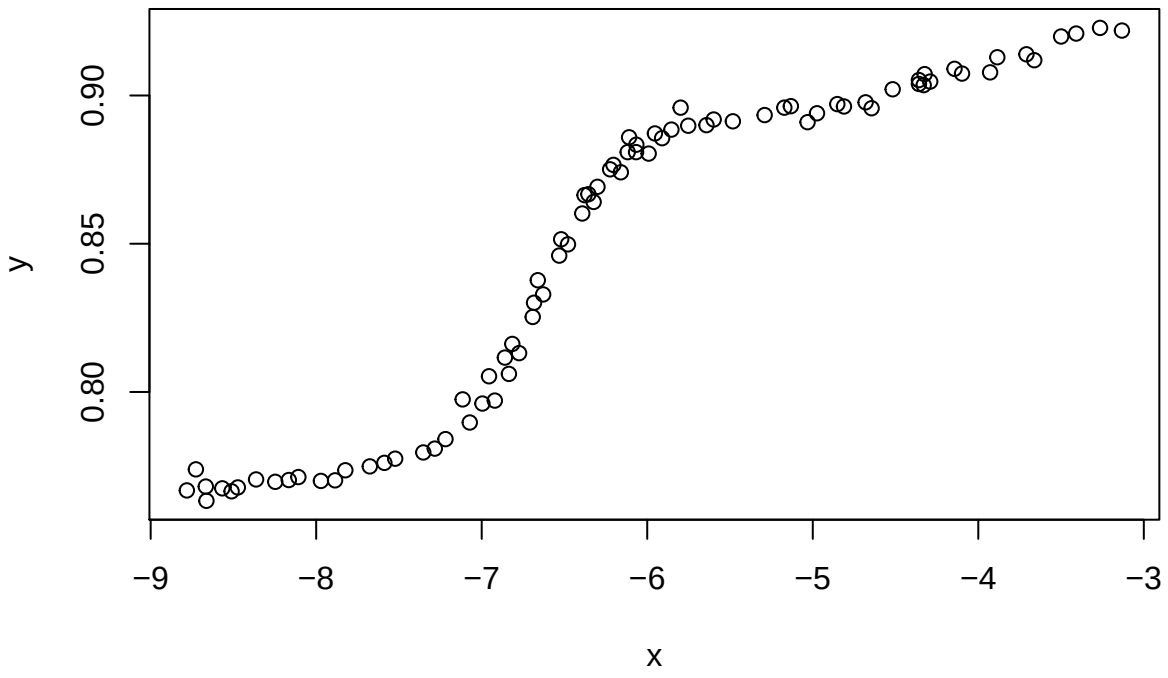
Ajuste de un polinomio de grado 10 para los datos Filip con R

En el presente texto mostraremos dos tipos distintos de regresión con polinomios utilizando la funcion lm() para los datos Filip del Nist que se muestran en la fig.1

Fig.1 Conjunto de datos Filip para regresión lineal



Regresión lineal con un polinomio

El primer tipo de regresión consiste en llamar a lm() indicando la forma polinomial de los regresores en la fórmula que recibe de parámetro. En este caso indicamos a la función que genere la regresión con 9 columnas extra que no están contenidas en los datos originales y que se componen de potencias de la columna 'x'.

```
## 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')
title(main = 'Fig.2 Curva del primer modelo de regresión')
```

Fig.2 Curva del primer modelo de regresión

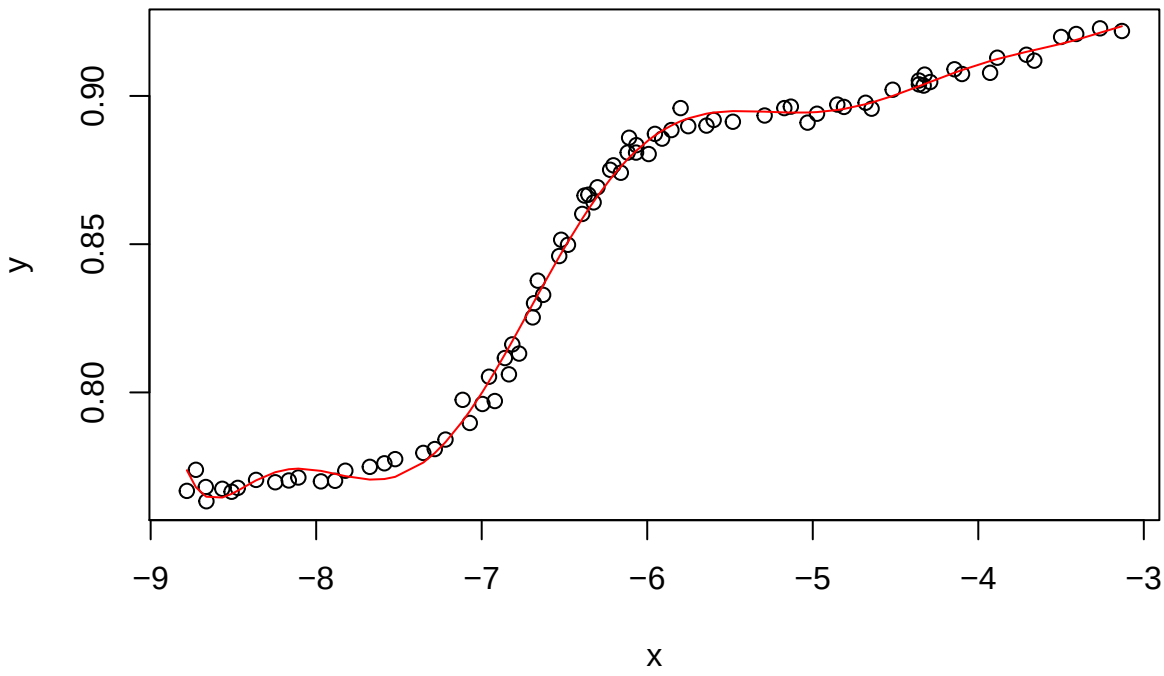


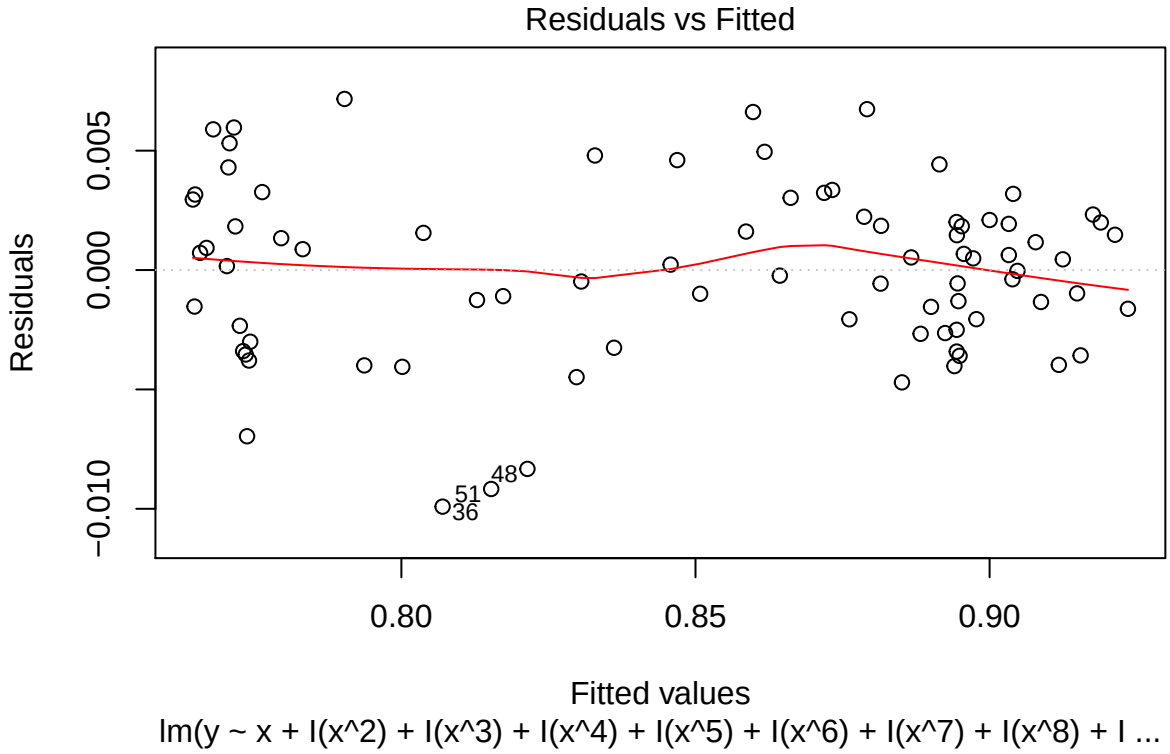
Table 1: Tabla de coeficientes para el primer modelo

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-174.2804413	87.5611625	-1.990385	0.0503455
x	-326.8822027	148.0496187	-2.207923	0.0304356
I(x^2)	-266.0565352	109.5120849	-2.429472	0.0176167
I(x^3)	-123.9216122	46.5247149	-2.663565	0.0095339
I(x^4)	-36.3816704	12.5145201	-2.907157	0.0048449
I(x^5)	-6.9791883	2.2111661	-3.156338	0.0023332
I(x^6)	-0.8746602	0.2567449	-3.406728	0.0010791
I(x^7)	-0.0690601	0.0189006	-3.653865	0.0004872
I(x^8)	-0.0031183	0.0008009	-3.893624	0.0002186
I(x^9)	-0.0000614	0.0000149	-4.122537	0.0000991

Como podemos ver en la Tabla.1 de coeficientes, estos muestran una clara diferencia a los coeficientes certificados del NIST. Todos los coeficientes mostrados tienen un valor-p por debajo de 0.05, pero falta el coeficiente asociado al termino x^10. El último coeficiente falta debido a que tiene un valor 'NA', lo cual puede suceder por distintas razones, pero una posibilidad es que la matrix X'X sea 'numericamente singular' (posibilidad que se discute en este foro de r). También notamos que el valor de la R-ajustada del modelo es de 0.9952 , lo que corresponde al buen ajuste que se puede observar en la fig.2.

Por último mostramos la fig.3 que contiene la gráfica de residuales, en donde se muestra un poco de conglomeración de puntos cerca del valor 0.9 en el eje horizontal.

Fig.3 Gráfica de residuales para el primer modelo

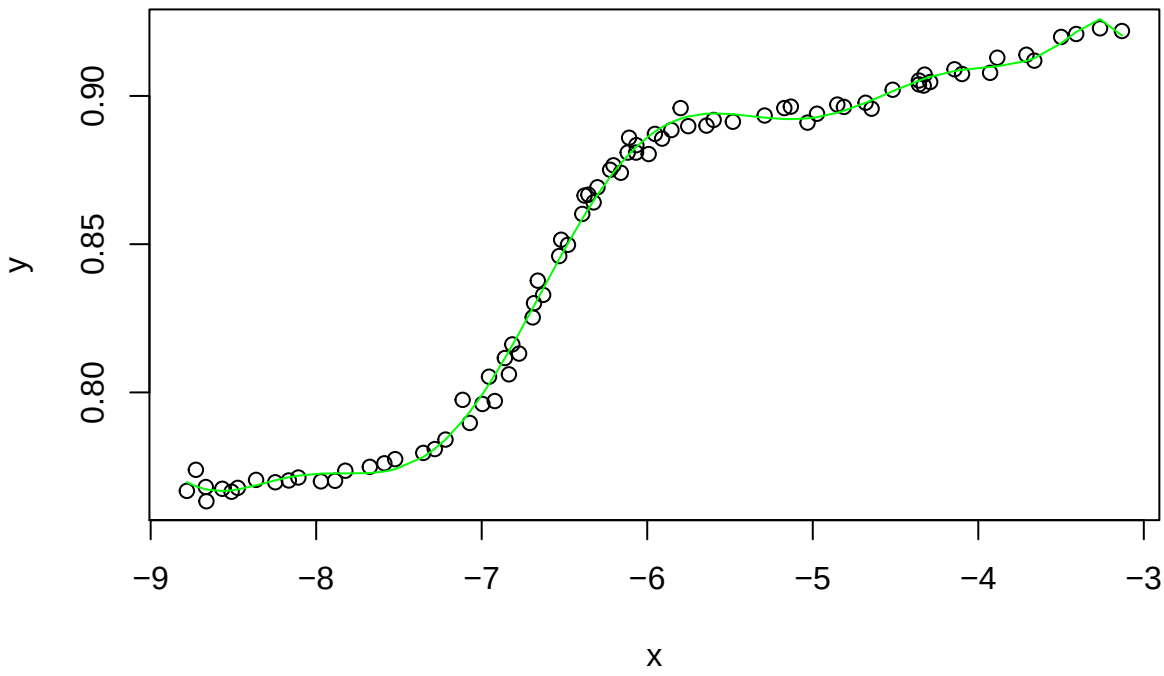


Regresión lineal con polinomios ortogonales

Debido al problema con el último coeficiente del modelo con la técnica anterior, buscamos otro método para realizar el ajuste. En este ejemplo se realiza la regresión con la asistencia de la funcion poly(), la cual utiliza polinomios ortogonales para aliviar el problema de colinearidad de los regresores.

```
## 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')
title('Fig. 4 Gráfica del segundo modelo de regresión')
```

Fig. 4 Gráfica del segundo modelo de regresión



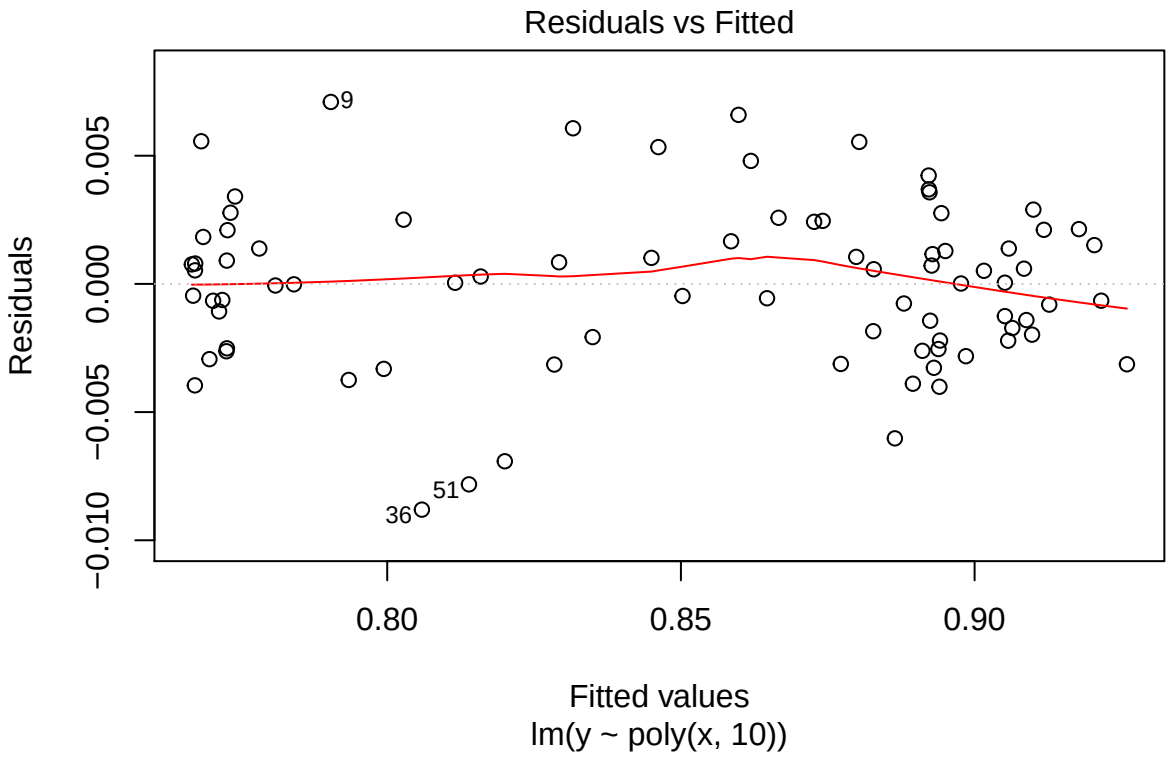
Como podemos observar en la Tabla.2 , la mayoria de los coeficientes tienen un valor-p por debajo de 0.01 a excepcion del coeficiente asociado a x^7, el cual tiene un valor-p ligeramente superior a 0.05. Tambien notamos que con este metodo fue posible la estimación del último coeficiente, a pesar de lo cual, los coeficientes siguen siendo considerablemente distintos a los coeficientes certificados del NIST para este conjunto de datos. Además, el modelo cuenta con una R-ajustada de 0.9963, ligeramente superior al primer modelo.

Table 2: Tabla de coeficientes para el segundo modelo

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.8495756	0.0003697	2297.852507	0.0000000
poly(x, 10)1	0.4613904	0.0033480	137.810307	0.0000000
poly(x, 10)2	-0.0867992	0.0033480	-25.925601	0.0000000
poly(x, 10)3	-0.0826891	0.0033480	-24.697991	0.0000000
poly(x, 10)4	0.0967433	0.0033480	28.895771	0.0000000
poly(x, 10)5	0.0174523	0.0033480	5.212744	0.0000018
poly(x, 10)6	-0.0616874	0.0033480	-18.425090	0.0000000
poly(x, 10)7	0.0066664	0.0033480	1.991166	0.0503114
poly(x, 10)8	0.0340241	0.0033480	10.162472	0.0000000
poly(x, 10)9	-0.0155338	0.0033480	-4.639702	0.0000155
poly(x, 10)10	-0.0150465	0.0033480	-4.494175	0.0000265

Por último mostramos la gráfica de residuales del segundo modelo en la fig.5 en donde notamos , de manera similar al primer modelo, que existe un poco de conglomeración cerca del valor 0.9 en el eje horizontal.

Fig.5 Gráfica de residuales para el segundo modelo



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 su veracidad pero al graficar las predicciones a partir del modelo ajustado vemos que efectivamente corresponde a un polinomio que ajusta los datos de forma correcta. Ante estos resultados, se podría seguir experimentado con el conjunto de datos Filip utilizando un paquete como Rmpfr que permite un nivel arbitrario de precisión en las operaciones aritmeticas con la definición de nuevas clases para los tipos numéricos.