

```

> #####
> # Regresion Inversa
> #####
>
> # tenemos una nueva observacion de fluorescencia, medida en una
> # solucion que no se utilizo para el ajuste. Su fluorescencia fue
> # de 15. Dar un intervalo de confianza para el valor de concentracion
> # del que proviene.
>
> install.packages("investr")
> library(investr)
> # para usar la instruccion de este paquete hay que tener los datos
> # en un data frame. Creemos uno
> fluoresc<-data.frame(cbind(fluo,conc))
> names(fluoresc)<-c("fluo","conc")
>
> View(fluoresc)
> plotFit(salida, data=fluoresc,interval = "both", level = 0.95, shade = TRUE, col.pred =
"skyblue")

> # en gris que intervalo se grafica? y en celeste?

>
> # intervalo inverso, obtenido usando propagacion de errores
> xnue <- calibrate(salida, y0 = 15, interval = "Wald", level = 0.95)
> xnue
  estimate      lower      upper      se
6.9842738 6.3657418 7.6028059 0.2406195
> #no coincide exactamente con invertir el intervalo de prediccion, pero casi
> abline(h = 15, v = c(xnue$lower, xnue$estimate, xnue$upper), lty = 2)
>
>
> # esta version que no vimos da el intervalo exacto que se obtiene al
> # invertir el IP. El problema con esta opcion es que a, a veces, no da
> # intervalo con extremos finitos. Puede imaginar cuando?
> xnue1 <- calibrate(salida, y0 = 15, interval = "inversion", level = 0.95)
> abline(h = 15, v = c(xnue1$lower, xnue1$estimate, xnue1$upper), lty = 2)
> xnue1
  estimate      lower      upper
6.984274 6.367744 7.606661
>
> # a mano
> xnuevo<-as.numeric((15-salida$coefficients[1])/salida$coefficients[2])
> xnuevo
[1] 6.984274
>
> #cuantil
> cuantil<-qt(0.975,df=5)
> ese<-summary(salida)$sigma
> deltaX <- var(conc)*(6)
> cuantil
[1] 2.570582
> ese
[1] 0.4328477
> deltaX
[1] 112
> desvest <- as.numeric( (ese/salida$coefficients[2])*sqrt(1 + 1/7 + (15-mean(fluo))^2 /
(salida$coefficients[2]^2*deltaX) ))
> desvest
[1] 0.2406195
>
> xnuevo - cuantil * desvest
[1] 6.365742
> xnuevo + cuantil * desvest
[1] 7.602806

```

