Entrega 2 - AMECP - Bioestadística

Francisco Javier López Carbonell

16/12/2023

Contents

EJERCICIO 2

En primer luegar, configuramos nuestro directorio de trabajo.

```
getwd()
```

```
## [1] "C:/Users/Javi2/OneDrive/Escritorio/Entrega2-Bioestadistica"
```

```
setwd("C:/Users/Javi2/OneDrive/Escritorio/Entrega2-Bioestadistica")
```

En un estudio sobre el efecto de la radicación en las alteraciones cromosómicas de linfocitos humanos, se registraron los valores experimentales de las variables (cells=número de células en cientos, ca=número de alteraciones cromosómicas, doseamt=cantidad de dosis de exposición in vitro a radiacción gamma, doserate=tasa de dosis de radiacción por hora) cuyos datos observados se encuentran en la base de datos dicentric de la librería faraway. Ver más detalles de los datos en RStudio mediante: ?dicentric. Con el objetivo de analizar la cantidad de alteraciones cromosómicas (ca) a través del resto de características registradas en el experimento:

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.3.2
```

```
?dicentric
```

```
## starting httpd help server ... done
```

Observamos que que el estudio consta de 27 observaciones sobre las siguientes 4 variables donde se han estudiado efectos de las dosis de radiación en las anomalías cromosómicas.

Procedemos a elaborar una pequeña tabla que facilite las variables que vamos a incluir en nuestro modelo.

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.3.2
```

```
datos_tabla <- data.frame(
  Nombre_variable = c("cells", "ca", "doseamt", "doserate"),
  Descripcion_variable = c("número de células en cientos", "número de alteraciones cromosómicas", "canti
```

```
)
kable(datos_tabla, format = "markdown")
```

Nombre_variable	Descripcion_variable
cells	número de células en cientos
ca	número de alteraciones cromosómicas
doseamt	cantidad de dosis de exposición in vitro a radiacción gamma
doserate	tasa de dosis de radiacción por hora

(a) Establecer un modelo para pronosticar el número de alteraciones cromosómicas a partir de las variables disponibles en la base de datos, determinar el modelo con las variables más relevantes.

A continuación, tras observar que tenemos un conjunto de variables que podrían llegar a explicar el comportamiento de una variable respuesta, vamor a realizar un **modelo de regresión lineal múltiple** que representa la relación entre la variable respuesta Y (en este caso, ca) y un conjunto de variables predictoras (X1, . . . ,Xk), que en este caso serán cells, doseamt, doserate. El objetivo será pronosticar los valores de la respuesta a través de este modelo.

```
library(faraway)
attach(dicentric) #extraemos datos de la base dicentric
#Regresión lineal múltiple -> usamos función lm con las 3 varaibles usadas para predecir la respuesta
model.regre1 <- lm(ca ~ cells + doseamt + doserate)
summary(model.regre1)
```

```
##
## Call:
## lm(formula = ca ~ cells + doseamt + doserate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.114 -26.858  -1.741  17.834 214.279
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -74.15392   42.24544  -1.755 0.092518 .
## cells         0.06871    0.02196   3.129 0.004709 **
## doseamt      41.33160    9.13907   4.523 0.000153 ***
## doserate     20.28402    8.29071   2.447 0.022482 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.05 on 23 degrees of freedom
## Multiple R-squared:  0.5213, Adjusted R-squared:  0.4588
## F-statistic: 8.348 on 3 and 23 DF,  p-value: 0.0006183
```

El modelo con todas las variables introducidas como predictores tiene un valor de R cuadrado ajustado (0.4588), es capaz de explicar el 45,88% de la variabilidad observada en el número de alteraciones cromosómicas. El p-value del modelo es significativo (0.0006183) por lo que se puede aceptar que el modelo no es por azar. Todas las variables predictoras son significativas, lo que es un indicativo de que todas ellas contribuyen al modelo y eliminar alguna puede significar una pérdida relevante de variabilidad explicada.

A la hora de seleccionar los predictores que deben formar parte del modelo se pueden seguir varios métodos: método jerárquico, método de entrada forzada o método de paso a paso (stepwise). Nosotros usaremos este

último, donde podemos optar por varias estrategias(forward, backward o doble(mixto).

En concreto, usaremos stepwise mixto, caracterizado por la combinación de ir incluyendo predictores 1 a 1 si mejoran el modelo anterior (forward) y en cada nueva incorporación se realiza un test de extracción de predictores no útiles (backward).

Nos podemos basar en varios criterios matemáticos para determinar si el modelo mejora o empeora con cada incorporación o extracción, el elegido será AIC ya que suele ser mas restrictivo. En R la función step() permite encontrar el mejor modelo basado en AIC.

```
step(object = model.regre1, direction = "both", trace = 1)
```

```
## Start:  AIC=219.12
## ca ~ cells + doseamt + doserate
##
##            Df Sum of Sq    RSS    AIC
## <none>                   67181 219.12
## - doserate  1     17484  84665 223.37
## - cells     1     28602  95783 226.70
## - doseamt   1     59742 126923 234.30
```

```
##
## Call:
## lm(formula = ca ~ cells + doseamt + doserate)
##
## Coefficients:
## (Intercept)        cells      doseamt     doserate
##   -74.15392      0.06871     41.33160     20.28402
```

La función step() podría devolvernos más de un modelo de selección de predictores, aunque en este caso solo nos devuelve uno basado en la inclusión de todas las variables descriptoras disponibles, por lo que en principio no eliminamos ningún predictor para nuestro modelo.

Por tanto, el modelo seleccionado será el que incluye todas las variables predictoras como factores de riesgo para la variable respuesta (número de alteraciones cromosómicas).

Además de el p-value tanto del modelo en conjunto como de las variables predictoras individualmente ya comentado anteriormente, resulta interesante analizar la estimación del error residual (sigma) y la tasa de error a la hora de evaluar un modelo de regresión lineal.

El **RSE** proporciona una medida de error en la predicción por parte del modelo, es decir, cuanto más bajo sea su valor, mejores predicciones de la variable respuesta nos dará el modelo. En este sentido si dividimos este valor entre el valor medio de la variable respuesta obtenemos la tasa de error por cada valor predicho por el modelo.

```
#Funcion sigma() para extraer el error residual estándar de lm
rse=sigma(model.regre1)
rse
```

```
## [1] 54.0454
```

```
rse/mean(dicentric$ca)
```

```
## [1] 0.4487164
```

Si la tasa de error es pequeña, es una indicación positiva de que el modelo está proporcionando un buen ajuste en relación con la escala media de la variable respuesta. En este caso no lo es, con valor de RSE de 54.0454 correspondiendo a una tasa de error alta del 44.87%. Esta tasa de error puede guardar relación con el tamaño muestral por lo que aumentando el número de observaciones podríamos aumentar la eficacia de predicción.

Por otro lado, también resulta interesante mostrar el intervalo de confianza para cada uno de los coeficientes parciales de regresión, donde si el intervalo de confianza es estrecho, sugiere que la estimación del coeficiente es precisa. Si es amplio, la estimación es menos precisa. Además, si el intervalo de confianza no incluye el valor cero, se podría argumentar que hay evidencia de que el coeficiente no es cero (es decir, que la variable tiene un efecto significativo).

```
confint(model.regre1)
```

```
##                     2.5 %     97.5 %
## (Intercept) -161.54526491 13.2374206
## cells          0.02328832  0.1141341
## doseamt       22.42600307 60.2372032
## doserate       3.13338606 37.4346622
```

En este caso, tenemos algunos intervalos confianza amplios como el de doseamt y doserate, por lo que la estimación de esos coeficientes no es muy precisa. Por otro lado, ningan variable incluye el 0, por lo que esto apoya la evidencia de efecto significativo de las variables que nos muestra el p-value.

Por último, mencionar que, en regresión lineal múltiple es posible encontrarnos con un aspecto muy a tener en cuenta y es la multicolinealidad de variables. La **multicolinealidad** ocurre cuando las variables independientes en un modelo de regresión están correlacionadas. Esta correlación es un problema porque las variables independientes deberían ser independientes. Si el grado de correlación entre las variables es lo suficientemente alto, puede causar problemas al ajustar el modelo e interpretar los resultados.

Para evaluar la multicolinealidad analizamos lo que se denomina inflación de varianza (VIF), es decir, donde se evalúa con que factor cada variable predictora influye en la multicolinealidad. En concreto, cuanto mayor sea el VIF, mayor será la multicolinealidad. Un VIF superior a 5 o 10 a menudo se considera una señal de multicolinealidad significativa. Lo ideal es maneternos en torno a la unidad como valor de VIF.

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```
vif(model.regre1)
```

```
##    cells  doseamt doserate
## 2.112665 2.101714 1.010951
```

Obtenemos un valor de VIF aceptable en este caso, por lo que no tenemos problema de multicolinealidad de variables.

Aun así y a modo de contraste, vamos a evaluar dos posibles modelos eliminando la variable con el valor de VIF mas alto (cells).

```
model.regre2 <- lm(ca ~ doseamt + doserate)
summary(model.regre2)
```

```
##
## Call:
## lm(formula = ca ~ doseamt + doserate)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -56.369 -36.097 -12.723   4.038 238.920
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   32.991     28.925   1.141  0.26529
## doseamt       20.626      7.369   2.799  0.00995 **
## doserate      17.584      9.638   1.824  0.08058 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.17 on 24 degrees of freedom
## Multiple R-squared:  0.3175, Adjusted R-squared:  0.2606
## F-statistic: 5.582 on 2 and 24 DF,  p-value: 0.01022
```

Vemos que, el modelo empeora significativamente observando el coeficiente de determinación, perdiendo mas de la mitad de variabilidad explicada (adjusted R-squared) respecto al modelo inicial. Además, aumenta el RSE y la variable doserate deja de tener un efecto dignificativo según este modelo, lo descartamos de inmediato.

Dicho esto, nuestro modelo de regresión lineal seleccionado listo para validar es:

ca = -74.15392 + 0.06871cells + 41.33160doseamt + 20.28402doserate

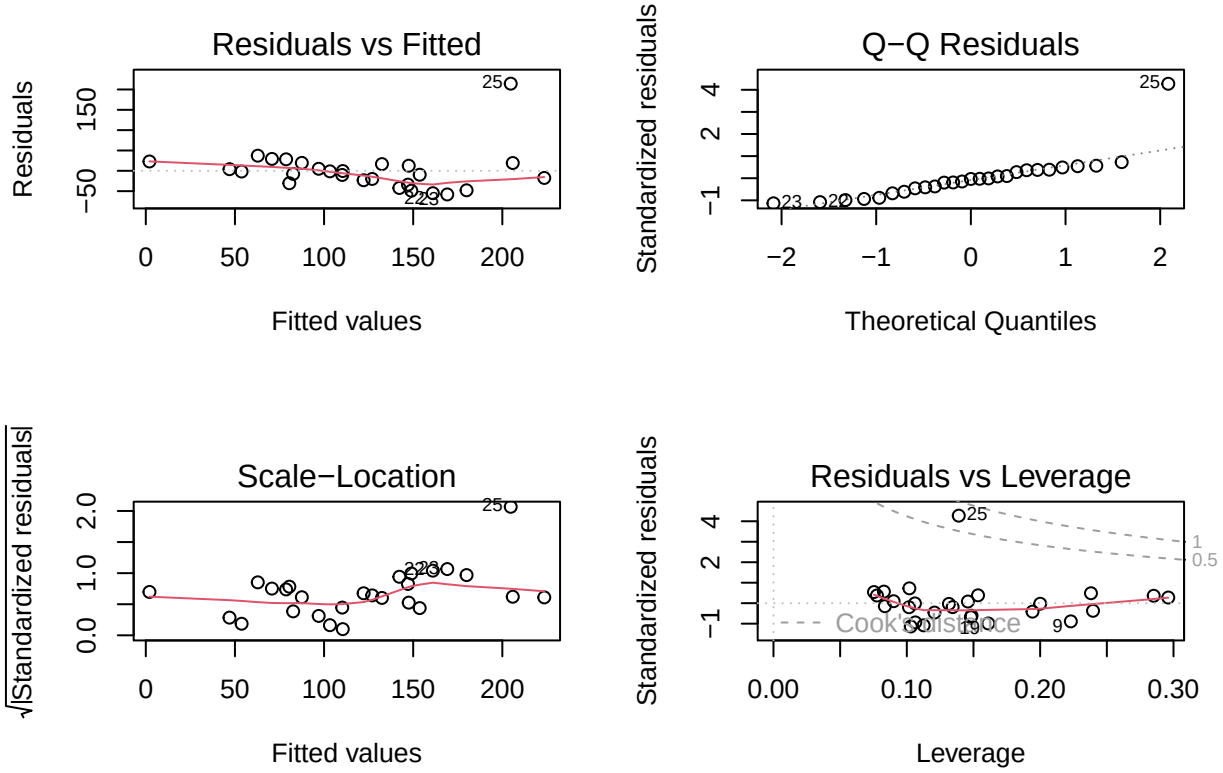
(b) Estudiar la validez del modelo estimado.

Además, el desarrollo del análisis del modelo de regresión (la estimación del modelo, los contrastes de significación y la predicción de la respuesta) requiere de unas ciertas condiciones que deben mantenerse, garantizando la aleatoriedad de los individuos observados y su representatividad en la población bajo estudio. Esto es precisamente lo que vamos a evaluar mediante test estadísticos para validar el modelo.

Las condiciones iniciales que han de cumplirse son: linealidad, homogeneidad de varianzas, incorrelación y normalidad.

Además, todo ello debe ir acompañado de un análasis gráfico de los residuos (diferencias entre los valores observados y los valores predichos por el modelo). En concreto, se procede a la identificación de observaciones atípicas o influyentes que pueden afectar significativamente los resultados del análisis. Estos análisis son importantes para garantizar la robustez y la confiabilidad de los resultados obtenidos a partir del modelo.

```
par(mfrow = c(2, 2))
plot(model.regre1)
```



Intrepretación de representaciones gráficas

A -> **Residuals vs Fitted:** representación de la nube de puntos de los risudos estandarizados frente los valores predichos por el modelo, para evaluar linealidad. En este caso, vemos claramente falta de linealidad en el modelo. Además. el gráfico nos incluye las observaciones mas dispersos y que afectan por tanto más a la desviación de linealidad (22, 23, 25).

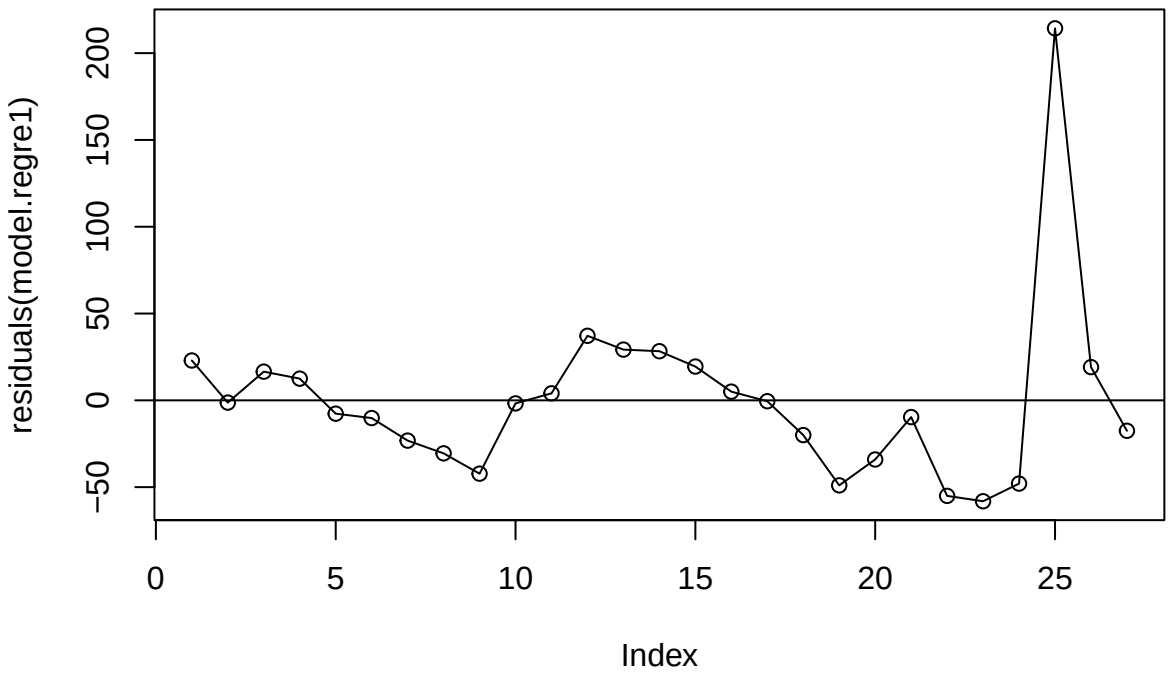
B -> **Normal Q-Q:**este gráfico compara los cuantiles de los residuos con los cuantiles teóricos de una distribución normal. De nuevo, vamos desviación extrema en las observaciones 22, 23 y 25 aunque por lo general la nube de puntos se proxima a una normal.

C -> **Scale-Location:** esta representación ayuda a identificar observaciones influyentes. Puntos lejanos en el eje y pueden indicar observaciones con un alto impacto en el modelo. De nuevo, tenemos observaciones 22, 23 y 25 como las mas influyentes.

D -> **Leverage vs Residuals:**representación de observaciones en base a su valor leverage basado en distancia de cook y residuos estandarizados. Observaciones con alto leverage y alto residuo estandarizado pueden ser influyentes indicadas en el gráfico: 9, 19 y 25.

E -> **Residuals in sample order:**Este gráfico te permite identificar patrones sistemáticos o comportamientos inusuales en los residuos a medida que avanzas a lo largo de las observaciones. Si vemoss patrones claros, como agrupamientos o cambios abruptos, podría indicar que hay aspectos específicos de tus datos que no están siendo capturados adecuadamente por el modelo. Las observaciones mencionadas anteriormente de nuevo se muestran en el gráfico como las más influyentes.

```
plot(residuals(model.regre1),type="o"); abline(h=0)
```



Análisis de linealidad

Debe haber una relación lineal entre las variables dependientes e independientes. Esta condición se puede validar bien mediante diagramas de dispersión entre la variable dependiente y cada uno de los predictores (como se ha hecho en el análisis preliminar) o con diagramas de dispersión entre cada uno de los predictores y los residuos del modelo. Si la relación es lineal, los residuos deben de distribuirse aleatoriamente en torno a 0 con una variabilidad constante a lo largo del eje X.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.3.2
```

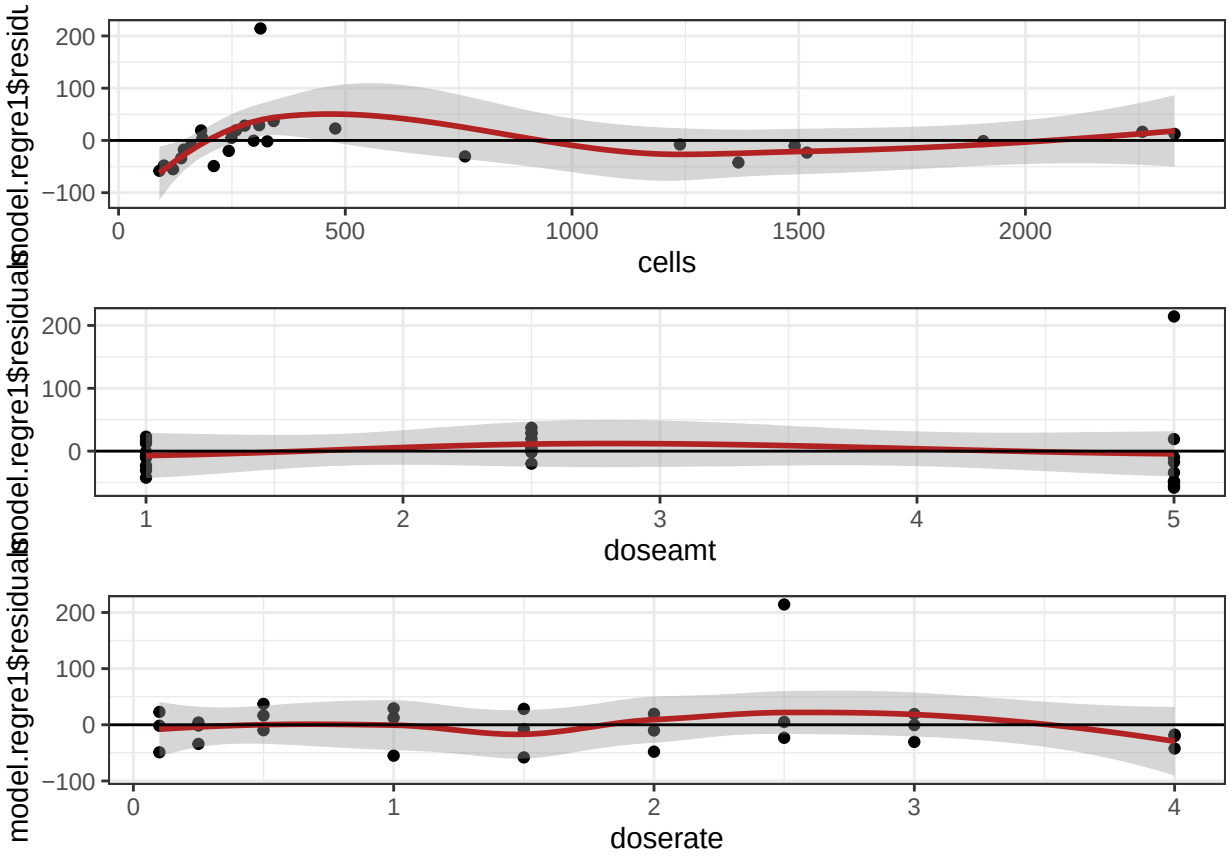
```
plot1 <- ggplot(data= dicentric, aes(cells, model.regre1$residuals)) + geom_point() + geom_smooth(color
plot2 <- ggplot(data = dicentric, aes(doseamt, model.regre1$residuals)) + geom_point() + geom_smooth(co
plot3 <- ggplot(data = dicentric, aes(doserate, model.regre1$residuals)) + geom_point() + geom_smooth(c
grid.arrange(plot1, plot2, plot3)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 0.98
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 4.02
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 7.9482e-17

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 16.16

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 0.98

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 4.02

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 7.9482e-17

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 16.16

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



En este caso, observamos una distribución aleatoria de los residuos en torno a lo largo del eje X en torno a 0,

por lo que *se cumple el criterio de linealidad para todas las variables* evaluadas.

Análisis de homocedascitidad

La homocedasticidad establece que la varianza de los errores o residuos debe ser constante para todos los valores de las variables predictoras o independientes. Esto significa que la dispersión de los residuos alrededor de la línea de regresión debe ser similar en toda la gama de valores de las variables predictoras. La forma de evaluarla en este cado será mediante la función bptest() (test de Breusch-Pagan) para la homocedasticidad. Esta función toma como entrada un modelo de regresión y devuelve el resultado de la prueba de hipótesis para la homocedasticidad de los residuos. En este estadístico, la hipotesis nula establece que hay homogeneidad de varianza para los errores.

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.3.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.3.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(model.regre1)
```

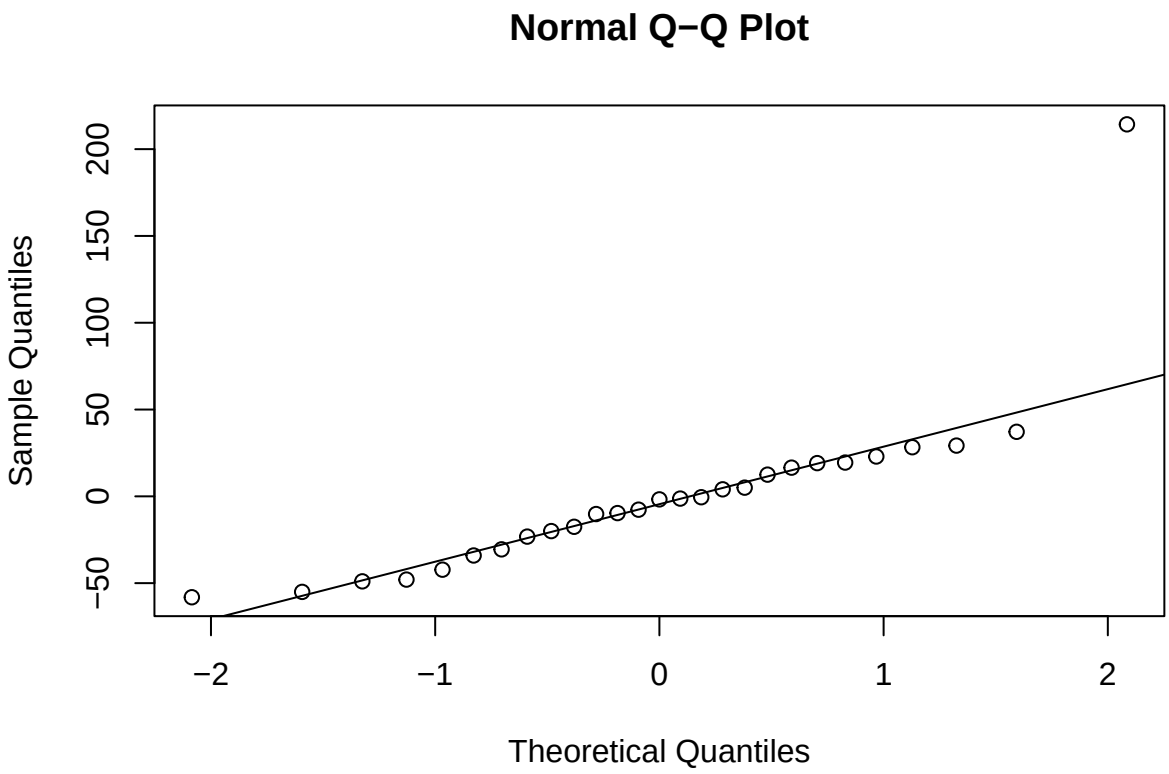
```
##
##  studentized Breusch-Pagan test
##
## data:  model.regre1
## BP = 3.6524, df = 3, p-value = 0.3015
```

Atendiendo al criterio establecido anteriormente, p-value supera 0.05, por lo que no hay evidencias para rechazar la hipótesis nula. En este sentido, asumimos que los residuos tienen varianza constante.

Analisis de normalidad

Existen varias formas de analizar la normalidad de los residuos, como ya hemos comentado se puede analizar gráficamente observando que tan bien se ajusta la nube de puntos a una distribución normal o realizando test estadísticos que evaluen la normalidad. En nuestro caso, tenemos un tamaño de muestra menor de 50 observaciones (27 observaciones) por lo que el test adecuado será Shapiro-Wilk.

```
qqnorm(model.regre1$residuals)
qqline(model.regre1$residuals)
```



El test de Shapiro-Wilks plantea la hipótesis nula que una muestra proviene de una distribución normal y tenemos una hipótesis alternativa que sostiene que la distribución no es normal.

```
shapiro.test(model.regre1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model.regre1$residuals
## W = 0.71838, p-value = 7.059e-06
```

En nuestro caso, rechazamos la hipotesis nula por lo que no hay evidencias de normalidad, aunque las observaciones de los residuos parecen aproximarse en su mayoría a la distribución normal. De hecho, si nos fijamos se observa un dato claramente candidato a ser outlier, excesivamente alejado de la distribución. Por tanto, procedemos a evaluar normalidad sin tenerlo en cuenta por si influyese determinantemente en el valor de p-value del test realizado.

```
which.max(model.regre1$residuals)
```

```
## 25
## 25
```

```
shapiro.test(model.regre1$residuals[-25])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model.regre1$residuals[-25]
## W = 0.95882, p-value = 0.3688
```

Efectivamente, se confirma que los residuos sí se distribuyen de forma normal a excepción de un dato extremo. Es necesario estudiar en detalle la influencia de esta observación para determinar si el modelo es más preciso sin ella.

Analisis de incorrelación

Para analizar esta condición, usamos el estadístico de Durbin-Watson, basado en las autocorrelaciones entre residuos adyacentes, y bajo la hipótesis nula de incorrelación. El estadístico D tiene una distribución simétrica centrada en el punto 2 y acotada en el intervalo (0, 4). Así, el valor del estadístico D próximo a los extremos del intervalo indica una tendencia de autocorrelación (positiva o negativa, según la asimetría) y un valor próximo a 2 no detectaría una falta de incorrelación.

```
library(car)
dwt(model.regre1, alternative = "two.sided")
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.1211934      1.745197    0.26
##  Alternative hypothesis: rho != 0
```

Siguiendo el criterio proporcionado, obtenemos un valor del estadístico D ~ 2, por lo que no hay evidencia de autocorrelación en este caso.

Analisis de valores atípicos

```
library(car)
outlierTest(model.regre1)
```

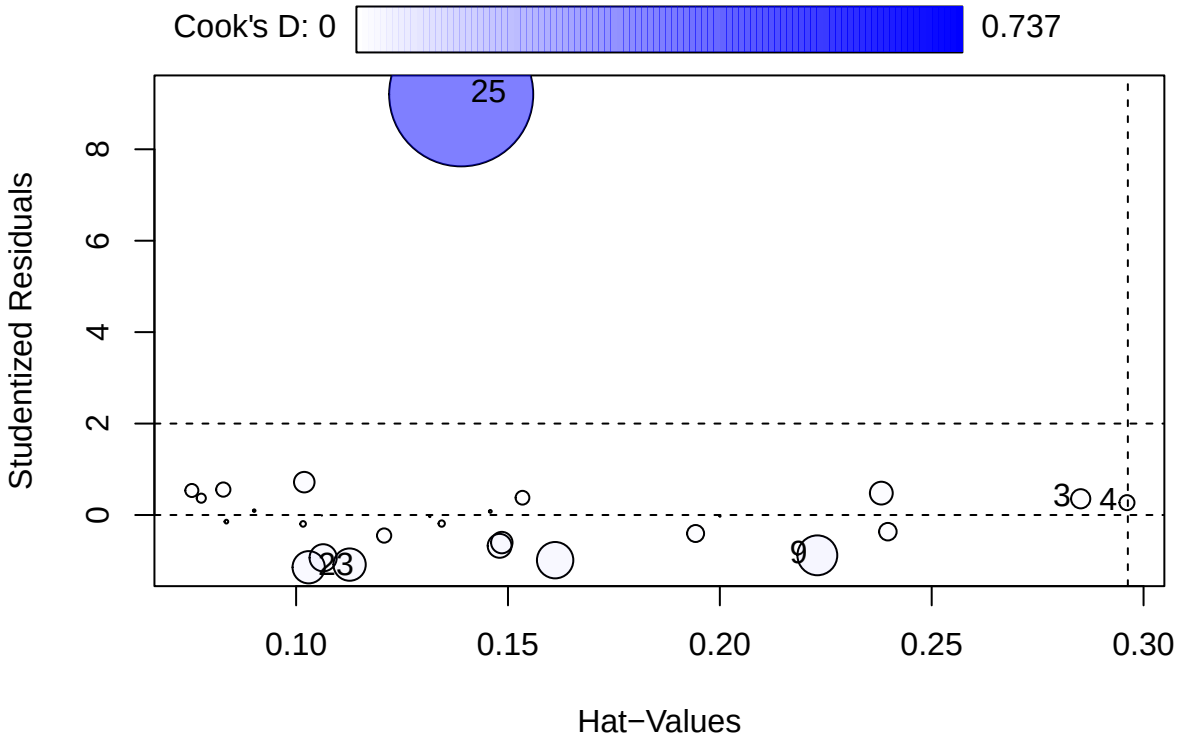
```
##    rstudent unadjusted p-value Bonferroni p
## 25 9.201672         5.3643e-09   1.4484e-07
```

Tal como se apreció en el estudio de normalidad de los residuos, la observación 25 tiene un residuo estandarizado >3 (más de 3 veces la desviación estándar de los residuos), por lo que se considera un dato atípico. El siguiente paso es determinar si es influyente.

```
summary(influence.measures(model.regre1))
```

```
## Potentially influential observations of
##   lm(formula = ca ~ cells + doseamt + doserate) :
##
##    dfb.1_  dfb.clls dfb.dsmt dfb.dsrt dffit   cov.r   cook.d hat
## 3  -0.06    0.17     0.06    -0.06     0.22    1.63_*  0.01   0.29
## 4  -0.06    0.15     0.06    -0.02     0.18    1.67_*  0.01   0.30
## 25 -2.17_*  1.44_*   2.77_*   1.43_*   3.70_*  0.00_*  0.74   0.14
## 27  0.11   -0.04    -0.10    -0.15    -0.20    1.53_*  0.01   0.24
```

```
influencePlot(model.regre1)
```




```
##       StudRes       Hat       CookD
## 3   0.3548201 0.2851625 0.013051743
## 4   0.2702727 0.2960828 0.008003901
## 9  -0.8824267 0.2230116 0.056416929
## 23 -1.1428483 0.1029620 0.036986356
## 25  9.2016715 0.1389529 0.736540501
```

El análisis muestra varias observaciones influyentes, aunque ninguna excede los límites de preocupación para los valores de Leverageshat(>2.5) o Distancia Cook(>1). Estudios más exhaustivos consistirían en rehacer el modelo sin las observaciones y ver el impacto.

Debemos tener en cuenta que, la exclusión de observaciones debe tener una justificación sólida y estar respaldada por un entendimiento adecuado del problema y los datos. En nuestro caso, tras los analisis de condiciones inciales del modelo de regresion lineal múltiple de los residuos nos lleva a a la decisión de analizar la exclusión de ciertas observaciones a priori influyentes en el ajuste de bondad del modelo.

Tras quitar varios outliers y analizar la predicción del modelo en el apartado c del ejercicio, obtenemos no solo que no predice bien el resultado, sino que no incluye el valor real dentro del intervalo de confianza de los valores predichos, por lo que descartamos dichos modelos. No incluyo los analisis por no alargar demasiado el informe.

Por tanto, en una búsqueda por optimizar nuestro modelo inicial, vamos a recurrir a una transformación de los datos. Dichas transformaciones suelen ser comunes cuando no se cumple una o mas condiciones iniciales del modelo de regresión lineal o cuando se desea ver si existen relaciones lineales de alguna variable en el espacio logarítmico.

La elección depende de la distribución de nuestros datos y de la relación que esperas entre las variables. Es importante recordar que la interpretación de los resultados también se verá afectada por estas transformaciones, por lo que debemos ajustar su interpretación en consecuencia.

En nuestro caso, tras realizar varias pruebas y desecharlas como estandarización, inversa de las variables o raices cuadradas nos hemos decantado por explorar el modelo en escala logarítmica.

Análisis de modelo en escala logarítmica

```
log.model <- lm(log(ca) ~ log(cells) + log(doseamt) + log(doserate), data = dicentric)
summary(log.model)
```

```
##
## Call:
## lm(formula = log(ca) ~ log(cells) + log(doseamt) + log(doserate),
##     data = dicentric)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29049 -0.10304  0.03696  0.09870  0.21726
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.13918    0.49430  -4.328 0.000249 ***
## log(cells)     0.91453    0.06932  13.193 3.26e-12 ***
## log(doseamt)   1.61398    0.10424  15.483 1.17e-13 ***
## log(doserate)  0.19666    0.02367   8.309 2.23e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1438 on 23 degrees of freedom
## Multiple R-squared:  0.9337, Adjusted R-squared:  0.9251
## F-statistic:   108 on 3 and 23 DF,  p-value: 1.071e-13
```

El modelo con todas las variables introducidas como predictores **tomando logaritmos** incrementa el valor de R cuadrado ajustado (0.9251), ahora es capaz de explicar el 92.51% de la variabilidad observada en el logaritmo del número de alteraciones cromosómicas, es decir, ahora la interpretación se hace como cambio porcentual en la variable dependiente. Por ejemplo, coeficiente de regresión para el logaritmo de cells de 0.9 indica que si dicha variable varía un 1% , la variable respuesta variará un 0.9%, siendo el resto constantes.

El p-value del modelo es mucho más significativo (1.071e-13) por lo que se puede aceptar que el modelo no es por azar. Todas las variables predictoras son más significativas en valor, indicativo de que todas ellas contribuyen al modelo de forma mas relevante y no cabe plantearse la eliminación de alguna de ellas.

Hacemos de nuevo stepwise, para confirmar si es adecuado no eliminar variables de acuerdo a su AIC (se confirma modelo con todas las variables).

```
step(object = log.model, direction = "both", trace = 1)
```

```
## Start:  AIC=-101.07
## log(ca) ~ log(cells) + log(doseamt) + log(doserate)
##
##                 Df Sum of Sq    RSS      AIC
## <none>                       0.4753 -101.072
## - log(doserate)  1    1.4266 1.9019  -65.631
## - log(cells)     1    3.5966 4.0719  -45.077
## - log(doseamt)   1    4.9540 5.4293  -37.309
##
## Call:
## lm(formula = log(ca) ~ log(cells) + log(doseamt) + log(doserate),
##     data = dicentric)
```

```
##
## Coefficients:
##   (Intercept)     log(cells)   log(doseamt)  log(doserate)
##       -2.1392         0.9145         1.6140         0.1967
```

Analizamos de nuevo el RSE y vemos que es muy bajito en términos de variables logaritmizadas. Por tanto, optenemos mayor capacidad de predicción de la variable respuesta por parte del modelo.

```
#Funcion sigma() para extraer el error residual estándar de lm
rse.log=sigma(log.model)
rse.log
```

```
## [1] 0.143751
```

```
rse.log/mean(dicentric$ca)
```

```
## [1] 0.001193505
```

Por otro lado, mostramos los intervalos de confianza al 95% como hemos realizado en el modelo anterior.

```
confint(log.model)
```

```
##                    2.5 %     97.5 %
## (Intercept)   -3.1617242 -1.1166400
## log(cells)     0.7711261  1.0579258
## log(doseamt)   1.3983448  1.8296153
## log(doserate)  0.1476971  0.2456212
```

En este caso, vemos que intervalos confianza que eran amplios como el de doseamt y doserate ahora se reducen en amplitud considerablemente, haciendo la estimación de los coeficientes más precisa.

Además, evaluamos de nuevo multicolinealidad, donde vemos que valores de algunas variables suben según el criterio anteriormente mencionado (no superar valor de VIF 5), aunque no es excesivamente alto, es algo que debemos tener en cuenta cuando analicemos la predicción del modelo.

A veces, aunque las variables individuales estén altamente correlacionadas (multicolinealidad), el efecto conjunto de las variables en la predicción podría ser informativo. El modelo puede estar capturando la relación global entre las variables de manera efectiva, por lo que seguimos analizando.

```
vif(log.model)
```

```
##    log(cells)  log(doseamt) log(doserate)
##      6.170066      6.168440      1.001626
```

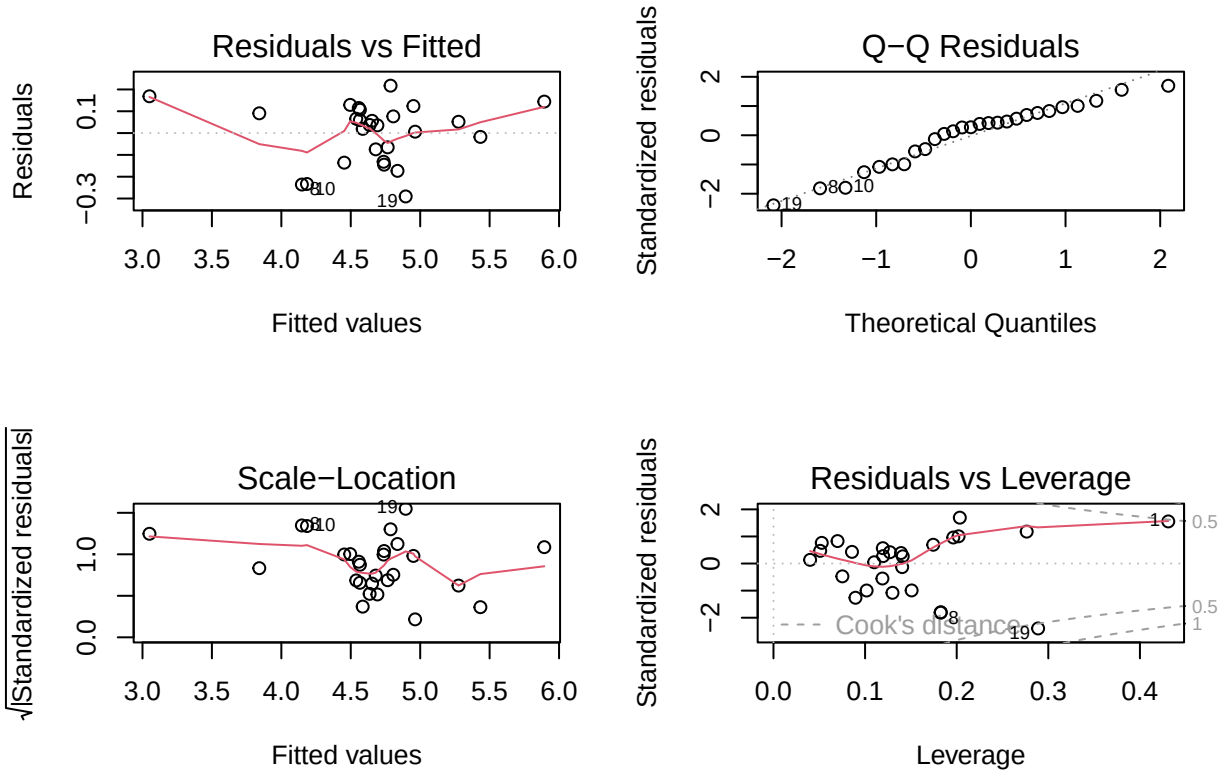
Dicho esto, nuestro modelo de regresión lineal múltiple en escala logarítmica listo para validar es:

ln(ca) = -2.13918 + 0.91453ln(cells) + 1.61398ln(doseamt) + 0.19666ln(doserate)

Análisis de validez de modelo logarítmico

En cuanto al analisis gráfico de la nube de puntos de los residuos, en este caso hay otras observaciones que se desvían indicadas en cada gráfico, que es lo que cabría esperar al cambiar el modelo. Lo importante es ver como afectan esas desviaciones a las predicciones de la variable respuesta y si se mantienen las condiciones iniciales del modelo de regresión lineal múltiple.

```
par(mfrow =c(2, 2))
plot(log.model)
```



Linealidad

```
library(ggplot2)
library(gridExtra)
dicentric.ex = dicentric
plot1.ex <- ggplot(data= dicentric.ex, aes(cells, log.model$residuals)) + geom_point() + geom_smooth(co
plot2.ex <- ggplot(data = dicentric.ex, aes(doseamt, log.model$residuals)) + geom_point() + geom_smooth
plot3.ex <- ggplot(data = dicentric.ex, aes(doserate, log.model$residuals)) + geom_point() + geom_smoot
grid.arrange(plot1.ex, plot2.ex, plot3.ex)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 0.98
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 4.02
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 7.9482e-17
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 16.16
```

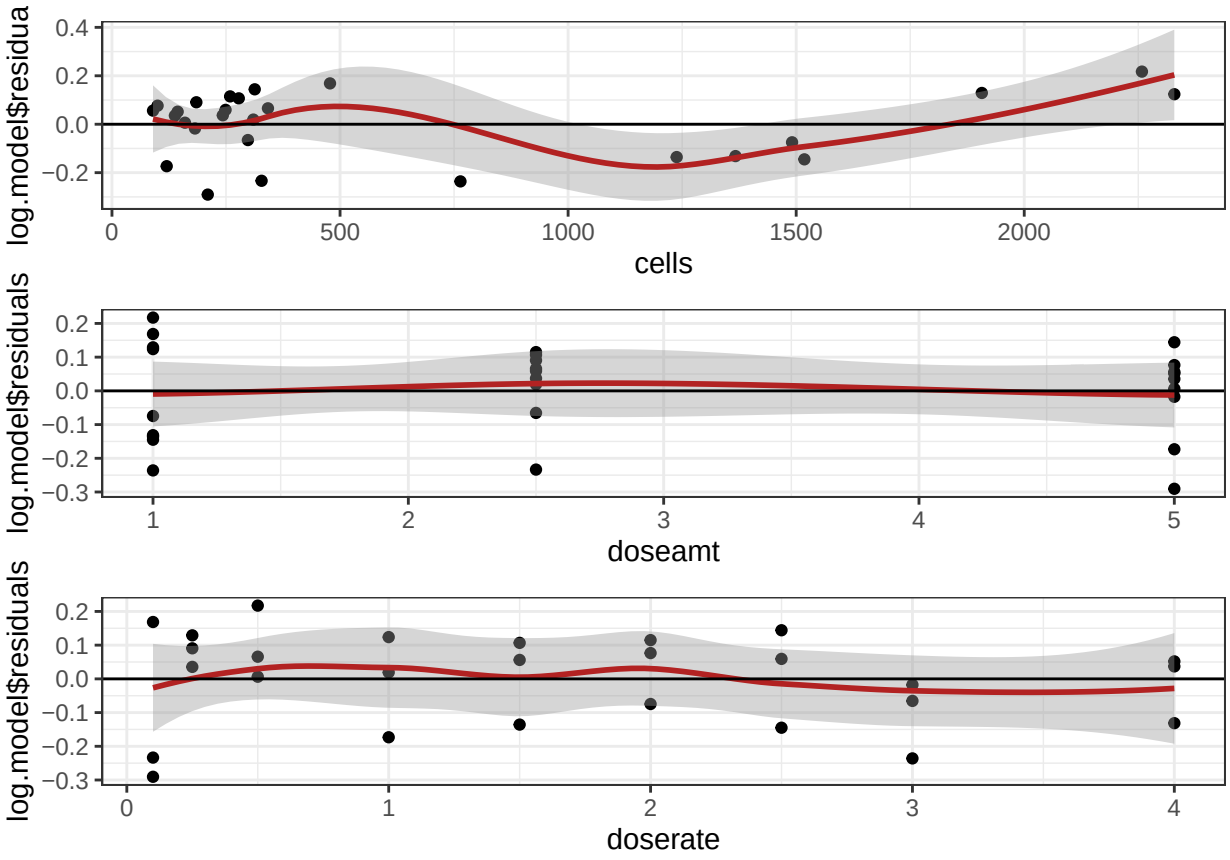
```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 0.98
```

```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 4.02
```

```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 7.9482e-17
```

```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 16.16
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



En este caso, de la misma manera que en el modelo anterior, se cumple el criterio de linealidad para las 3 variables predictoras, distribuyendose de forma uniforme en torno a 0 a lo largo del eje x para todas las variables.

Incorrelacion

```
library(car)
dwt(log.model, alternative = "two.sided")
```

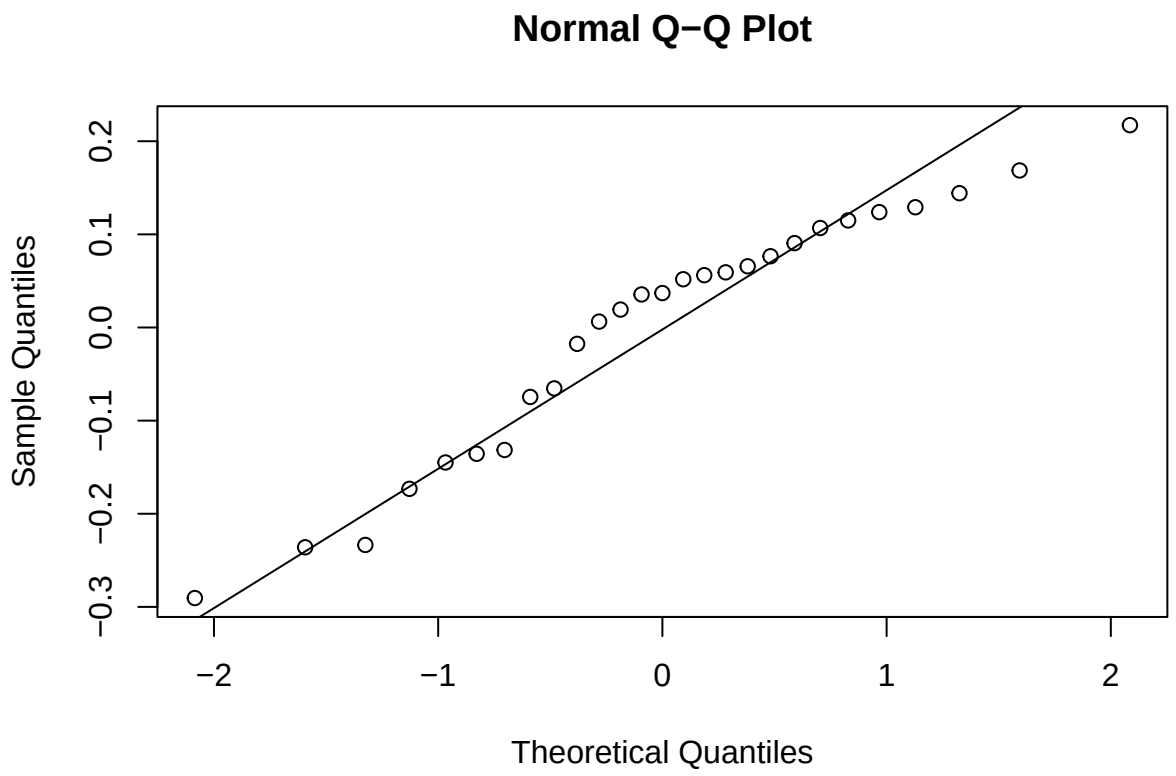
```
##  lag Autocorrelation D-W Statistic p-value
##    1        0.332301      1.269933   0.018
##  Alternative hypothesis: rho != 0
```

Según el criterio establecido, en este caso se nos reduce el estadistico D-W un poquito, aproximandose mas al extremo del intervalo (0,4), lo cual indica cierta autocorrelación positiva respecto al modelo anterior.

Normalidad

Ya analizada anteriormente, en este caso si hay normalidad en la distribucion de los residuos.

```
qqnorm(log.model$residuals)
qqline(log.model$residuals)
```




```
shapiro.test(log.model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log.model$residuals
## W = 0.93977, p-value = 0.1202
```

No rechazamos hipotesis nula de que existe normalidad, por lo que no hay evidencias para descartar que no la haya. La asumimos por tanto en este caso.

Homocedasticidad

De igual manera que el anterior modelo, usamos test de Breusch-Pagan, donde la hipotesis nula recordamos que es ausencia de homocedasticidad, es decir, heterogeneidad de varianza.

```
library(lmtest)
bptest(log.model)
```

```
##
```

```
##  studentized Breusch-Pagan test
##
## data:  log.model
## BP = 7.5971, df = 3, p-value = 0.05512
```

P-value>0.05, por lo que rechazamos H0, dando lugar a asumir que la varianza de los errores es constante.

Analisis de valores atípicos

Obtenemos de nuevo un outlier, pero ninguna excede los límites de preocupación para los valores de Hat (>2.5) o Distancia Cook(>1). La transformación logarítmica aplicada a las variables no afecta directamente la interpretación de estos estadísticos, ya que están diseñados para evaluar propiedades del modelo y sus residuos.

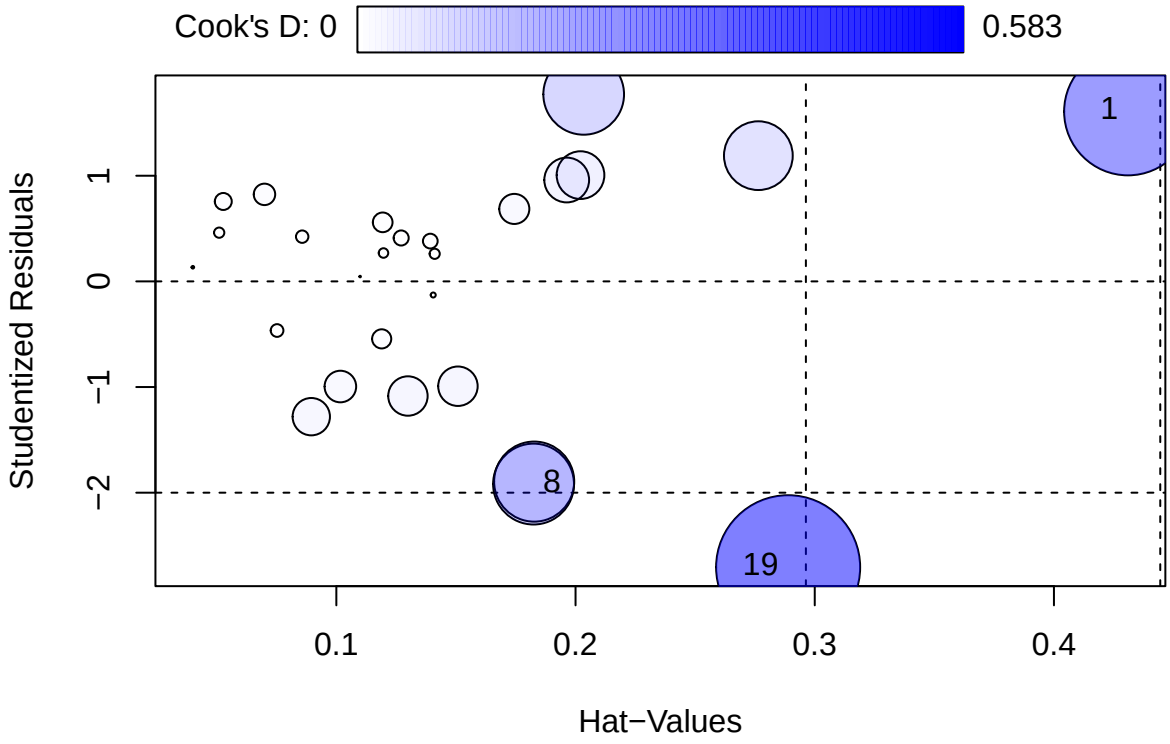
```
library(car)
outlierTest(log.model)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 19 -2.705659           0.012914      0.34866
```

```
summary(influence.measures(log.model))
```

```
## Potentially influential observations of
##   lm(formula = log(ca) ~ log(cells) + log(doseamt) + log(doserate),      data = dicentric) :
##
##    dfb.1_ dfb.lg(c) dfb.lg(dsm) dfb.lg(dsr) dffit   cov.r cook.d hat
## 1   0.98  -0.92     -1.06_*     -0.77        1.40_*  1.35  0.46   0.43
## 19  0.78  -0.76     -0.99        1.25_*     -1.72_*  0.53  0.58   0.29
```

```
influencePlot(log.model)
```




```
##      StudRes       Hat     CookD
## 1   1.607553 0.4309387 0.4577185
## 8  -1.918496 0.1824394 0.1839004
## 19 -2.705659 0.2888728 0.5831773
```

Llegados a este punto, cabe comentar que el **tamaño de muestra** es importante a la hora de analizar la validez del modelo, donde en el libro Hanbook of biological statistics recomiendan que el número de observaciones sea como mínimo entre 10 y 20 veces el número de predictores del modelo trantando de evitar que una variable parezca influyente cuando no lo es. En nuestro caso, 3 variables lo recomendable serían 50-60 observaciones pero disponemos de 27 resultando escaso para una evaluación óptima del modelo.

(c) Discutir la capacidad del modelo obtenido para pronosticar correctamente la cantidad de alteraciones y la significación de los términos predictores en el modelo.

Una vez establecidos los modelos de regresión, debemos comprobar como se produce la predicción de la variable respuesta (alteraciones cromosómicas). Para ello, lo vamos a aplicar a un nuevo conjunto de datos, extraido del propio dataframe para comprobar si predice valores similares a los obtenidos en las observaciones.

En concreto, usaremos la función predict() que toma como argumento el modelo y el nuevo dataframe extraido:

Vamos a registrar 3 situaciones para evaluar la predicción:

-Con una dosis de 1.0 Grays, una tasa de dosis en 0.25 Grays/h y 1907 cientos de células.

-Con una dosis de 2.5 Grays, una tasa de dosis en 1.0 Grays/h y 310 cientos de células.

-Con una dosis de 3 Grays, una tasa de dosis en 1 Grays/h y 182 cientos de células.

```
df.pred <- data.frame(cells=c(1907, 310, 182), doseamt=c(1.0, 2.5, 5), doserate=c(0.25, 1, 3))
kable(df.pred, format = "markdown")
```

cells	doseamt	doserate
1907	1.0	0.25
310	2.5	1.00
182	5.0	3.00

Evaluamos **primer** modelo:

```
predict(object=model.regre1, newdata=df.pred)
```

```
##         1         2         3
## 103.28094  70.75958 205.86160
```

Hemos seleccionado datos de cells variados en valor para ver si influía y aqui mostramos una comparación con los valores observados.

```
comp_ca_model1 <- data.frame(Ca_observados=c(102, 100, 225), Ca_predichos=c(103.28094, 70.75958, 205.86
kable(comp_ca_model1, format = "markdown")
```

Ca_observados	Ca_predichos
102	103.28094
100	70.75958
225	205.86160

Vemos que la precisión es más alta en en el valor alto de cells. Mostramos a continuación el intervalo de confianza al 95% para la predicción de los valores proporcionados.

```
predict(object=model.regre1, newdata=df.pred, interval="confidence", level=0.95)
```

```
##         fit       lwr      upr
## 1 103.28094  53.27754 153.2843
## 2  70.75958  38.58655 102.9326
## 3 205.86160 162.06996 249.6532
```

De esta manera, se puede observar que el intervalo de confianza es muy amplio para las 3 variables predictoras, por lo que el modelo no es preciso y existe una gran probabilidad de registrar valores erroneos predichos de alteraciones cromosómicas.

Evaluamos **segundo** modelo:

```
predict(object=log.model, newdata=df.pred)
```

```
##        1        2        3
## 4.495867 4.585937 5.433670
```

```
comp.ca.model1 <- data.frame(Ca.observados=c(102, 100, 225), Ca.predichos=c(exp(4.495867),  exp(4.58593
kable(comp.ca.model1, format = "markdown")
```

Ca.observados	Ca.predichos
102	89.64586
100	98.09506
225	228.98809

En este caso, observamos valores mucho mas cercanos al valor real. Parece que, el modelo en escala logarítmica a pesar de aumentar la multicolinealidad, esto no siempre conduce a un rendimiento deficiente del modelo, y su impacto puede variar según el contexto y la naturaleza de los datos.

```
int.pred.log <- predict(object=log.model, newdata=df.pred, interval="confidence", level=0.95)
pred.exp <- exp(int.pred.log)
pred.exp
```

```
##         fit       lwr      upr
## 1  89.64587  78.42888 102.4671
## 2  98.09507  92.43092 104.1063
## 3 228.98808 204.83577 255.9882
```

Los intervalos donde se encuentra el valor Y (alteraciones cromosómicas) ahora es mucho mas estrecho, lo que indica que la capacidad predictora del modelo ha mejorado considerablemente y que existe una menor probabilidad de obtener datos erróneos acercandose mucho a al valor de los datos reales.

(d) Comentar las conclusiones del análisis.

En resumen, nos encontramos con dos modelos de regresión lineal múltiple extraidos de las observaciones de 3 variables predictoras, donde en uno de ellos hemos detectado ausencia de evidencia de normalidad causada por un outlier, retirando esa observación y volviendo a elaborar el mismo modelo excluyéndola.

ca = -74.15392 +0.06871cells + 41.33160doseamt + 20.28402doserate

Dichas pruebas de modelo retirando outliers han dado lugar a un modelo muy poco preciso.

Hemos recurrido a la transformación logarítmica, donde parece que el modelo alcanza gran parte de robustez para explicar el comportanmiento de la variable respuesta en términos relativos y predecirla.

ln(ca) = -2.13918 + 0.91453**ln(cells)** + 1.61398**ln(doseamt)** + 0.19666**ln(doserate)**

Para obtener la interpretación en la escala original, aplica la función exponencial a cada término logarítmico:

ca = **exp**(-2.13918)exp(0.91453**ln(cells)**)**exp**(1.61398**ln(doseamt)**)**exp**(0.19666**ln(doserate)**)

Concluimos que el modelo es capaz de explicar el 92.51% de la variabilidad observada en las alteraciones cromosómicas en términos relativos. El test F muestra que es significativo (p-value: ~ 0). Se satisfacen todas las condiciones para este tipo de regresión múltiple, a excepción de la multicolinealidad de cells y doseamt. No obstante, el tamaño muestral es insuficiente. Puede que, con un tamaño de muestra suficiente no hubiéramos tenido que transformar los datos.