Taller de problemas GRUPO inferencia 2023 MAT3 GIN

Blanca Atiénzar Martínez, Hai Zi Bibiloni Trobat y Khaoula Ikkene

27/12/2023

Contenidos

1 Taller Problemas evaluable 22-23: Estadística Inferencial **1**

1.1 Problema 1: Regresión lineal simple. 7 puntos. 1

1.2 Problema 2: Distribución de los grados de un grafo de contactos. 3 puntos 7

1.3 Problema 3: Longitud reviews mallorca AirBnb 2022. 4 puntos 12

1 Taller Problemas evaluable 22-23: Estadística Inferencial

Valor 14 puntos. Todos los apartados valen 1 punto.

Se trata de resolver los siguientes problemas y cuestiones en un fichero Rmd y su salida en un informe en html, word o pdf.

1.1 Problema 1: Regresión lineal simple. 7 puntos.

Consideremos los siguientes datos

```
x=c(-2,-1,2,0,1,2)
y=c(-7, -5,  5, -3,  3.0,  4)
```

1. Calcular manualmente haciendo una tabla los coeficiente de la regresión lineal de y sobre x.
2. Calcular los valores $\hat{y}_i = b_0 + b_1 \cdot x_1$ para los valores de la muestra y el error cometido.
3. Calcular la estimación de la varianza del error.
4. Resolver manualmente el contraste $\begin{cases} H_0: & \beta_1 = 0 \\ H_1: & \beta_1 \neq 0 \end{cases}$, calculando el p-valor.
5. Calcular SST, SSR y SSE.
6. Calcular el coeficiente de regresión lineal r_{xy} y el coeficiente de determinación R^2. Interpretad el resultado en términos de la cantidad de varianza explicada por el modelo
7. Comprobar que los resultados son los mismos que los obtenidos con la función `summary(lm(y~x))`.

Apartado 1

```r
# Calcular medias
x_media = mean(x)
y_media = mean(y)

# Calcular productos x_i * y_i y x_i^2
xy = x * y
x_cuadrado = x^2

# Crear la tabla
tabla_regresion = data.frame(x = x, y = y, xy = xy, x_cuadrado = x_cuadrado)

# diferencia_x = x_i - x'
tabla_regresion$diferencia_x = x - x_media

# diferencia_y = y_i - y'
tabla_regresion$diferencia_y = y - y_media

# diferencia_xy = (x_i - x')*(y_i - y')
tabla_regresion$diferencia_xy = (x - x_media) * (y - y_media)

# Mostrar la tabla
tabla_regresion
```

```
##    x  y xy x_cuadrado diferencia_x diferencia_y diferencia_xy
## 1 -2 -7 14          4   -2.3333333         -6.5    15.1666667
## 2 -1 -5  5          1   -1.3333333         -4.5     6.0000000
## 3  2  5 10          4    1.6666667          5.5     9.1666667
## 4  0 -3  0          0   -0.3333333         -2.5     0.8333333
## 5  1  3  3          1    0.6666667          3.5     2.3333333
## 6  2  4  8          4    1.6666667          4.5     7.5000000
```

** Apartado 2 ** Para calcular los parametros

$$b_0$$

y

$$b_1$$

utilizaremos las seguientes formulas:

$$b_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2},$$
$$b_0 = \frac{\sum_{i=1}^{n} y_i - b_1 \sum_{i=1}^{n} x_i}{n}.$$

$$b_1 = \frac{\tilde{s}_{xy}}{\tilde{s}_x^2}, \quad b_0 = \bar{y} - b_1\bar{x}.$$

```r
desv_x = sd(x)
desv_x
```

```
## [1] 1.632993
```

```r
desv_y=sd(y)
desv_y
```

```
## [1] 5.128353
```

```r
desv_xy=cov(x,y)
desv_xy
```

```
## [1] 8.2
```

```r
b_1 = desv_xy/desv_x^2
b_1
```

```
## [1] 3.075
```

```r
b_0 = y_media-b_1*x_media
b_0
```

```
## [1] -1.525
```

El error cometido se calcula usando la seguiente formula:

$$E_{x_i} = y_i - b_0 - b_1 \cdot x_i$$

```r
y_calculada = b_0 + b_1*x
y_calculada
```

```
## [1] -7.675 -4.600  4.625 -1.525  1.550  4.625
```

```r
errores = y-y_calculada
errores
```

```
## [1]  0.675 -0.400  0.375 -1.475  1.450 -0.625
```

Apartado 3 Calcularemos la estimación de la varianza del error usando la seguiente formula:

$$S^2 = \frac{SS_E}{n-2}$$

```r
SSe=sum(errores^2)
n = length(x)
var_estimada=SSe/(n-2)
var_estimada
```

```
## [1] 1.35625
```

Apartado 4 Vamos a resolver manualmente el seguiente constraste

$$\begin{cases} H_0: & \beta_1 = 0 \\ H_1: & \beta_1 \neq 0 \end{cases}$$

El estadístico de constraste es el seguiente:

$$T = \frac{b_1}{\frac{S}{\tilde{s}_x\sqrt{n-1}}}$$

```
estadistico_T=b_1/((sqrt(var_estimada)/(desv_x*sqrt(n-1))))
estadistico_T
```

```
## [1] 9.6415
```

Ahora calcularemos el p-valor de acuerdo con la seguiente formula:

$$p = 2 \cdot P\left(t_{n-2} > |t_0|\right)$$

```
p_valor=2*pt(abs(estadistico_T),df=n-2,lower.tail = FALSE)
p_valor
```

```
## [1] 0.0006472191
```

Como que el p-valor es menor que 0.05 podemos decir que tenemos suficientes evidencias para rechazar la hipótesis nula, es decir, rechazamos la hipótesis que

$$\beta_1 = 0$$

.

Apartado 5

Se nos pide calcular los seguientes parametros

-SST/Variablidad total

$$SS_T = \sum_{i=1}^{n} (y_i - \bar{y})^2 = (n-1) \cdot \tilde{s}_y^2$$

```
SST=(n-1)*desv_y^2
SST
```

```
## [1] 131.5
```

Como que ya hemos calculado el SSE en los apartados anteriores y tenemos el SST podemos usar la seguiente propiedad que se cumple en nuestro caso, ya que hemos obtenido nuestro parametros b_1 y b_0 usando el método de los mínimos cuadrados.

$$SS_T = SS_R + SS_E$$

```
SSR=SST-SSe
SSR
```

```
## [1] 126.075
```

Apartado 6

Se nos pide calcular el coeficiente de regresión lineal. Para ello usaremos la seguiente formula:

$$r_{xy} = \frac{\tilde{s}_{xy}}{\tilde{s}_x \cdot \tilde{s}_y}.$$

```
r_xy=desv_xy/(desv_x*desv_y)
r_xy
```

```
## [1] 0.9791554
```

Y para el coeficiente de determinación usaremos la seguiente formula

$$R^2 = r_{xy}^2$$

```
R_cuadrado=r_xy^2
R_cuadrado
```

```
## [1] 0.9587452
```

Apartado 7

```
# Resultados obtenidos manualmente
resultados_manuales = list(
  Coeficientes_manuales = c(b_0, b_1),
  Valores_ajustados = y_calculada,
  Errores = errores,
  Varianza_del_error = var_estimada,
  Contraste_hipotesis = c(estadistico_T, p_valor),
  Sumas_de_cuadrados = c(SST, SSR, SSe),
  Coeficiente_correlacion_R_xy = r_xy,
  Coeficiente_determinacion_R2 = R_cuadrado
)

# Resultados de summary(lm(y ~ x))
sol_lm = summary(lm(y~x))
resultados_summary_lm=list(
  Coeficientes_summary = c(sol_lm$coefficients[1, 1], sol_lm$coefficients[2, 1]),
  Valores_ajustados_summary = predict(lm(y ~ x)),
  Errores_summary = sol_lm$residual,
  Varianza_del_error_summary=sigma(lm(y ~ x))^2,
  Contraste_hipotesis_summary = c(sol_lm$coefficients[2,3], sol_lm$coefficients[2,4]),
  Sumas_de_cuadrados_summary = c(sum(sol_lm$residuals^2) + sum((lm(y ~ x)$fitted.values - mean(y))^2), s
  Coeficiente_correlacion_R_xy_summary = cor(lm(y ~ x)$model$x, lm(y ~ x)$model$y),
  Coeficiente_determinacion_R2_summary = sol_lm$r.squared
)
# Comparación de resultados
list(
  Resultados_manuales = resultados_manuales,
  Resultados_summary_lm = resultados_summary_lm
)
```

```
## $Resultados_manuales
## $Resultados_manuales$Coeficientes_manuales
## [1] -1.525  3.075
##
## $Resultados_manuales$Valores_ajustados
## [1] -7.675 -4.600  4.625 -1.525  1.550  4.625
##
## $Resultados_manuales$Errores
## [1]  0.675 -0.400  0.375 -1.475  1.450 -0.625
##
## $Resultados_manuales$Varianza_del_error
## [1] 1.35625
##
## $Resultados_manuales$Contraste_hipotesis
## [1] 9.6415001605 0.0006472191
##
## $Resultados_manuales$Sumas_de_cuadrados
## [1] 131.500 126.075   5.425
##
## $Resultados_manuales$Coeficiente_correlacion_R_xy
## [1] 0.9791554
##
## $Resultados_manuales$Coeficiente_determinacion_R2
## [1] 0.9587452
##
##
## $Resultados_summary_lm
## $Resultados_summary_lm$Coeficientes_summary
## [1] -1.525  3.075
##
## $Resultados_summary_lm$Valores_ajustados_summary
##      1      2      3      4      5      6
## -7.675 -4.600  4.625 -1.525  1.550  4.625
##
## $Resultados_summary_lm$Errores_summary
##      1      2      3      4      5      6
##  0.675 -0.400  0.375 -1.475  1.450 -0.625
##
## $Resultados_summary_lm$Varianza_del_error_summary
## [1] 1.35625
##
## $Resultados_summary_lm$Contraste_hipotesis_summary
## [1] 9.6415001605 0.0006472191
##
## $Resultados_summary_lm$Sumas_de_cuadrados_summary
## [1] 131.500 126.075   5.425
##
## $Resultados_summary_lm$Coeficiente_correlacion_R_xy_summary
## [1] 0.9791554
##
## $Resultados_summary_lm$Coeficiente_determinacion_R2_summary
## [1] 0.9587452
```

1.2 Problema 2: Distribución de los grados de un grafo de contactos. 3 puntos

The marvel chronology project es una web que ha recopilado las apariciones de los personajes Marvel en cada uno de los cómics que se van publicando.

En el artículo Marvel Universe looks almost like a real social network se estudió la red de contactos de los personajes del Universo Marvel de la serie de cómics books. Dos personajes tienen relación si han participado en al menos un mismo cómic; a semejanza del Oracle of Bacon donde se relacionan los actores de las películas de Hollywood que han participado en al menos una película juntos.

Si construimos el grafo de asociado a esas relaciones el grado de cada carácter (personaje) será el número de ortos caracteres (personajes) con los que ha colaborado. Cuando más importante es el personaje más colaboraciones tiene.

Los grados de cada caracteres están en el fichero `datasets/degree_Marvel_characters.cvs`. Según algunos estudios la distribución de los grados de los grafos de contactos sigue una ley potencial frecuencia grado $k = \beta_0 \cdot grado^{\beta} 1$ si eliminamos los 20 más pequeños.

```
data=read_csv("datasets/degree_Marvel_characters.csv")
```

Se pide:

1. Cargad los datos. Calcular las frecuencias de los grados, es decir el número de caracteres que tienen 1, 2 ,3 .... colaboradores para cada grado (número de colaboraciones) observado.

2. Ajustar un modelo lineal, potencial y exponencial a la relación entre $y =$ "frecuencia del grado" y $x = grado$ dibujar las gráficas de ajuste de cada modelo con gráficos semi-log y log-log si es necesario.
3. Para el mejor modelo calcular los coeficientes en las unidades originales y escribir la ecuación del modelos.

Apartado 1

```
library(readr)

# Calcular las frecuencias de los grados
frecuencias_grados <- table(data$degree_Marvel_characters)
frecuencias_grados
```

```
##
##   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32
##  45  60 144 202 311 350 427 442 530 514 524 455 429 441 414 411
##  34  36  38  40  42  44  46  48  50  52  54  56  58  60  62  64
## 380 361 344 367 285 247 231 232 230 193 176 179 189 176 174 149
##  66  68  70  72  74  76  78  80  82  84  86  88  90  92  94  96
## 145 118 115 131 145 109  97  88  85 103  91  76  78  66  65  59
##  98 100 102 104 106 108 110 112 114 116 118 120 122 124 126 128
##  90  55  66  80  49  53  51  57  41  55  49  49  43  38  41  38
## 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158 160
##  57  43  65  54  21  40  41  28  43  45  40  34  33  32  33  24
## 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192
##  32  22  28  26  13  38  29  29  22  32  27  21  21  24  21  17
## 194 196 198 200 202 204 206 208 210 212 214 216 218 220 222 224
##  17  20  25  19  16  21  11  39  18  16  18  13  17  26  14  14
## 226 228 230 232 234 236 238 240 242 244 246 248 250 252 254 256
```

```
##   10   10   18   20   12   13   22   22   13   26   18   12   10   14   11   15
##  258  260  262  264  266  268  270  272  274  276  278  280  282  284  286  288
##   11   15    9   10    6   13    6    5    8   12    8    7   18   11   11    9
##  290  292  294  296  298  300  302  304  306  308  310  312  314  316  318  320
##    8   16    7   11   13    6   11    7    6    6    7   14   10    7   10   12
##  322  324  326  328  330  332  334  336  338  340  342  344  346  348  350  352
##    7   10   10    5   11    3   10   19    8    8    7   10   11    8    5    4
##  354  356  358  360  362  364  366  368  370  372  374  376  378  380  382  384
##   12    1    8   11    5    8    6    3    7    3    5    6    5    9    6    8
##  386  388  390  392  394  396  398  400  402  404  406  408  410  412  414  416
##    4    7   10    5    6    4    6    5    4    5    5    4    4    1    6    6
##  418  420  422  424  426  428  430  432  434  436  438  440  442  444  446  448
##    3    2    6    4    2    2    2    3    2    2    4    6    8    5    2    7
##  450  452  454  456  458  460  462  464  466  468  470  472  474  476  478  480
##    7    3    2    1    4    3    5    1    6    3    6    6    3    3    2    2
##  482  484  486  488  490  492  494  496  498  500  502  506  508  510  512  514
##    5    5    3    3    2    2    2    6    4    5    3    7    3    3    4    5
##  516  518  520  522  524  526  528  530  532  534  536  538  540  542  544  546
##    3    3    3    1    3    4    3    3    1    2    2    2    3    1    3    5
##  548  550  552  554  556  558  560  562  564  566  568  570  572  576  580  584
##    6    2    3    3    5    5    4    2    1    5    2    2    2    2    3    2
##  586  588  590  592  594  596  598  600  602  604  606  608  610  614  616  618
##    2    3    2    2    4    6    5    2    1    2    6    2    4    4    1    1
##  620  624  626  628  630  632  634  640  642  644  648  650  654  656  658  660
##    3    2    3    3    1    2    1    5    1    2    3    3    3    1    1    1
##  662  670  672  676  678  680  682  684  686  690  692  696  698  702  704  706
##    3    1    5    3    1    3    2    4    1    3    3    1    3    2    1    2
##  708  710  712  714  716  718  720  722  724  726  728  730  732  734  736  738
##    2    2    1    2    2    1    1    3    2    2    1    3    1    4    1    1
##  740  742  746  748  750  752  754  756  758  760  762  764  766  768  772  774
##    1    1    4    2    1    1    2    1    1    1    3    2    1    3    3    1
##  776  784  788  790  796  798  804  806  810  812  814  816  818  820  824  828
##    2    1    1    1    3    1    4    1    1    1    1    3    2    3    2    1
##  830  832  834  836  842  844  846  848  850  854  856  858  860  862  864  866
##    2    3    1    1    1    1    1    2    1    2    3    1    2    2    2    1
##  868  870  872  874  876  878  882  884  886  888  892  896  902  908  910  914
##    1    1    2    1    1    1    2    2    1    1    1    2    1    1    1    1
##  916  920  922  928  930  934  936  942  948  952  954  956  962  964  966  972
##    2    3    1    1    1    1    2    1    4    1    2    1    3    1    2    1
##  974  976  980  982  984  986  988  990  994  998 1002 1004 1008 1010 1014 1022
##    1    2    1    1    1    1    1    2    2    1    1    1    2    1    1    2
## 1024 1026 1028 1030 1032 1038 1040 1042 1044 1046 1050 1054 1056 1074 1076 1078
##    2    1    1    1    2    2    2    1    1    1    1    1    1    1    1    2
## 1080 1082 1090 1092 1098 1102 1104 1106 1110 1114 1120 1122 1126 1132 1134 1146
##    2    1    1    1    2    2    1    2    1    1    2    2    1    2    1    1
## 1148 1152 1154 1162 1164 1172 1176 1178 1180 1186 1188 1190 1206 1208 1216 1222
##    1    1    3    2    1    1    1    1    1    2    1    1    1    1    1    1
## 1224 1226 1228 1232 1242 1246 1250 1252 1254 1256 1262 1264 1266 1272 1274 1280
##    2    3    1    3    1    1    1    1    1    1    1    1    1    1    1    1
## 1286 1288 1292 1294 1298 1300 1304 1314 1318 1320 1322 1326 1332 1346 1348 1352
##    1    2    1    1    1    1    1    1    2    1    1    1    2    1    1    1
## 1356 1360 1370 1384 1386 1396 1410 1416 1430 1442 1446 1450 1454 1460 1462 1466
##    2    1    1    1    1    1    1    2    2    2    1    1    1    1    1    1
## 1482 1494 1496 1498 1500 1502 1508 1516 1520 1534 1536 1548 1560 1564 1566 1572
```

```
##    1    1    1    1    1    1    2    1    1    3    1    1    1    1    1    2
## 1578 1586 1594 1596 1618 1636 1638 1640 1660 1664 1674 1676 1684 1690 1692 1734
##    1    1    1    1    1    1    1    1    2    1    1    1    1    2    1    1
## 1740 1754 1756 1760 1764 1768 1778 1782 1792 1794 1798 1804 1816 1822 1828 1842
##    1    1    1    1    2    1    1    1    1    1    1    1    1    2    1    1
## 1864 1866 1868 1870 1882 1884 1886 1896 1898 1932 1936 1958 1984 2006 2008 2018
##    1    1    1    1    1    1    1    2    1    1    1    1    1    1    1    1
## 2028 2030 2046 2058 2092 2128 2152 2258 2292 2326 2330 2346 2392 2406 2440 2458
##    1    1    2    1    1    1    1    1    1    1    1    2    1    2    1    1
## 2460 2504 2520 2560 2598 2600 2606 2616 2660 2678 2718 2738 2742 2824 2870 2878
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 2944 2976 2980 3018 3092 3202 3380 3384 3392 3608 3704 3712 3806 3900 4008 4060
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 4066 4108 4174 4282 4382 4384 4402 4432 4506 4532 4678 4798 5286 5378 5616 5678
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 5696 6370 6408 7834 7982 8276 8586
##    1    1    1    1    1    1    1
```

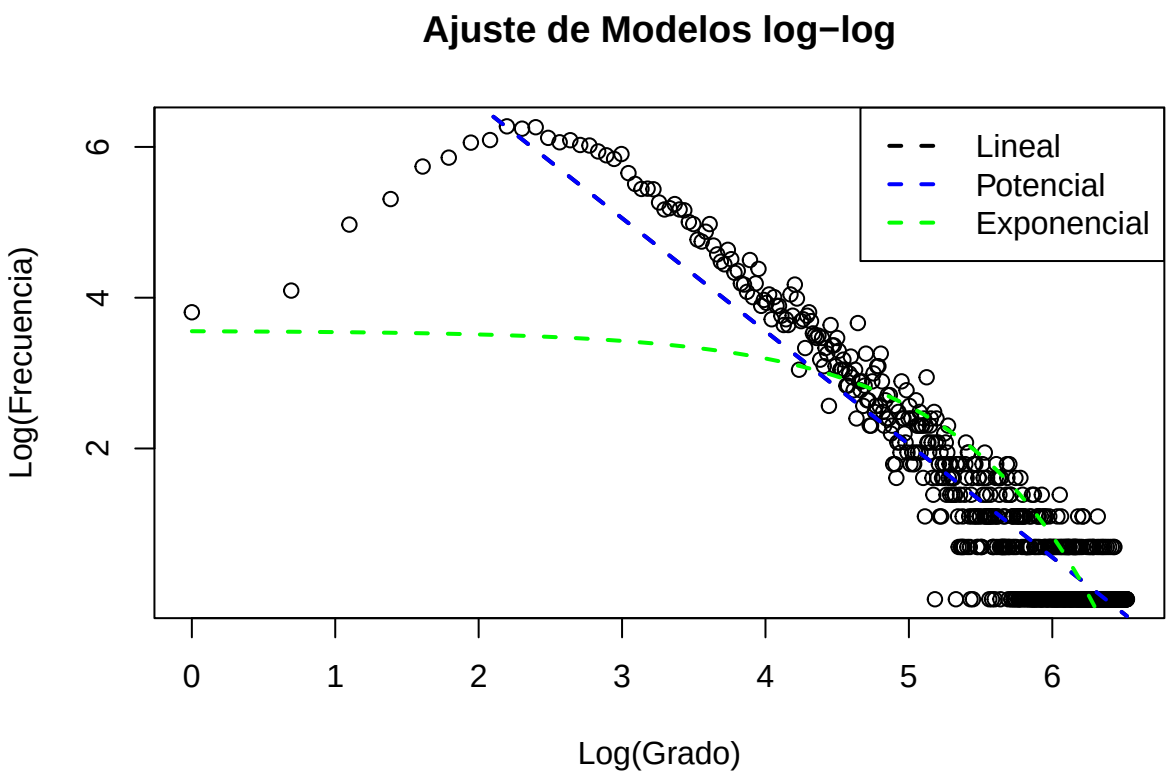
Apartado 2

```r
# Cargar las librerías
library(ggplot2)
library(dplyr)
# Ajustar modelos
modelo_lineal <- lm(log(frecuencias_grados) ~ log(seq_along(frecuencias_grados)))
modelo_potencial <- lm(log(frecuencias_grados) ~ log(seq_along(frecuencias_grados))^2)
modelo_exponencial <- lm(log(frecuencias_grados) ~ seq_along(frecuencias_grados))

# Gráfico log-log
plot(log(seq_along(frecuencias_grados)), log(frecuencias_grados),
     main="Ajuste de Modelos log-log",
     xlab="Log(Grado)",
     ylab="Log(Frecuencia)")

lines(log(seq_along(frecuencias_grados)), predict(modelo_lineal), col="black", lty=2, lwd=2)
lines(log(seq_along(frecuencias_grados)), predict(modelo_potencial), col="blue", lty=2, lwd=2)
lines(log(seq_along(frecuencias_grados)), predict(modelo_exponencial), col="green", lty=2, lwd=2)

legend("topright", legend=c("Lineal", "Potencial", "Exponencial"),
       col=c("black", "blue", "green"), lty=2, lwd=2)
```



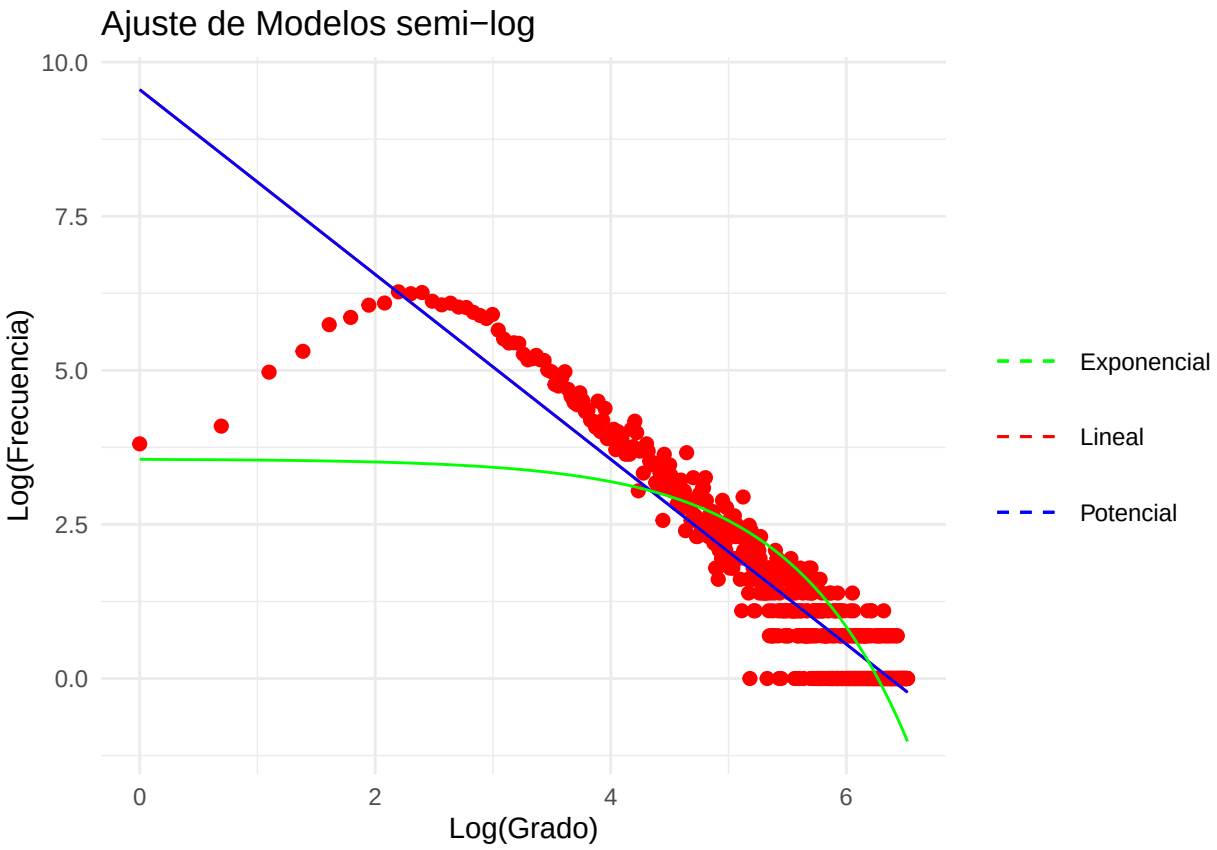

```r
# Dataframe para las predicciones
dframe <- data.frame(log_grado = log(seq_along(frecuencias_grados)))

# Añadir predicciones de los modelos al dataframe
dframe$pre_lineal <- predict(modelo_lineal)
dframe$pre_potencial <- predict(modelo_potencial)
dframe$pre_exponencial <- predict(modelo_exponencial)

# Gráfico semi-log
ggplot() +
  geom_point(aes(x = log(seq_along(frecuencias_grados)), y = log(frecuencias_grados)),
             size = 2, color = "red") +
  geom_line(data = dframe, aes(x = log_grado, y = pre_lineal, color = "Lineal"),
            show.legend = TRUE) +
  geom_line(data = dframe, aes(x = log_grado, y = pre_potencial, color = "Potencial"),
            show.legend = TRUE) +
  geom_line(data = dframe, aes(x = log_grado, y = pre_exponencial, color = "Exponencial"),
            show.legend = TRUE) +
  labs(title = "Ajuste de Modelos semi-log",
       x = "Log(Grado)",
       y = "Log(Frecuencia)") +
  theme_minimal() +
  scale_color_manual(name = "",
                     values = c("Lineal" = "red", "Potencial" = "blue", "Exponencial" = "green"),
                     labels = c("Exponencial", "Lineal", "Potencial")) +
  guides(color = guide_legend(override.aes = list(linetype = c("dashed", "dashed", "dashed")))) +
```

```
  theme(legend.key.size = unit(1, "cm"))
```

```
## Don't know how to automatically pick scale for object of type <table>.
## Defaulting to continuous.
```



Apartado 3 La elección de un modelo adecuado se basa en la evaluación de varias métricas y consideraciones contextuales. En el contexto de ajustar un modelo a la distribución de grados de un grafo de contactos, se han propuesto tres modelos: lineal, potencial y exponencial. La elección del modelo potencial se basa en su adecuación teórica a la distribución de grados en redes complejas, su interpretación en el contexto del problema y su rendimiento en términos de ajuste a los datos.

```
mejor_modelo = modelo_potencial

# Obtener coeficientes
coeficient_0 <- exp(coef(mejor_modelo)[1])
coeficient_1 <- coef(mejor_modelo)[2]

# Ecuación del modelo en las unidades originales
ecuacion_modelo <- paste("Frecuencia = ", coeficient_0, " * Grado ^ ", coeficient_1, sep="")

# Imprimir la ecuación
cat("Ecuación del modelo potencial (en unidades originales):", ecuacion_modelo, "\n")
```

```
## Ecuación del modelo potencial (en unidades originales): Frecuencia = 14097.6302581351 * Grado ^ -1.4
```

1.3 Problema 3: Longitud reviews mallorca AirBnb 2022. 4 puntos

El siguiente código cuenta cuantas palabras hay en un la variable `commnets` del fichero `reviews.csv` de los comentario a cada apartamento de Mallorca extraído de la web Inside AirBnb que recoge datos de los alquileres vacacionales por zonas del mundo de la web de alquiler de apartamentos vacacionales AirBnb. Se puede leer con el siguiente código y contar el número de palabras con la `stringr::str_count`.

```
read_csv("datasets/reviews.csv")->reviews
```

```
## Rows: 342750 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): reviewer_name, comments
## dbl  (3): listing_id, id, reviewer_id
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
names(reviews)
```

```
## [1] "listing_id"    "id"            "date"          "reviewer_id"
## [5] "reviewer_name" "comments"
```

```
library(stringr)
#str_count(str, pattern = "")
str_count(str=reviews$comments[1],pattern ="\\s+")
```

```
## [1] 78
```

Es habitual que la frecuencia de la longitud de los comentarios, es decir cuantos comentarios tienen 5, 6, 7 palabras y sus frecuencias siguen una ley que puede ser: lineal, exponencial o potencial. Como hemos hecho en el tema de regresión lineal calcular se trata de calcular y dibujar los tres modelos y decidir cuál es el más ajustado.

Se pide:

1. Calcular las longitudes de todos los comentarios (utilizar funciones como `mutate`, `arrange`, `filter`....) y las frecuencias de cada longitud y filtrar (con la función `filter`) solo los comentarios con **MÁS de 20 palabras y MENOS de 800** y guardarlos en una tibble con dos columnas N_{words}= número de palabras y $Frec$=frecuencia absoluta de las palabras.

2. Calcular los tres modelos lineal $Freq = \beta_0 + \beta_1 \cdot N_{words}$, potencial $Freq = \beta_0 \cdot (N_{words})^{\beta_1}$ y exponencial $Freq = \beta_0 \cdot \beta_1^{N_{words}}$.

3. Repetir el ajuste anterior pero sustituyendo el la variable N_{words} por el rango u orden de N_{words}.

Apartado 1

```r
# Cargar bibliotecas
library(readr)
library(dplyr)
library(stringr)
library(ggplot2)

# Cargar datos
 reviews <- read.csv("datasets/reviews.csv")

# Calcular longitudes de comentarios
reviews %>%
  mutate(comment_length = str_count(comments, "\\S+")) -> reviews_with_lengths

# Filtrar comentarios con más de 20 y menos de 800 palabras
filtered_reviews <- reviews_with_lengths %>%
  filter(comment_length > 20, comment_length < 800) %>%
  select(comment_length)

# Verificar la tibble resultante
head(filtered_reviews)
```

```
##   comment_length
## 1             79
## 2             73
## 3            115
## 4            106
## 5            100
## 6            122
```

```r
# Mostrar las primeras filas del conjunto de datos
head(reviews)
```

```
##   listing_id      id       date reviewer_id reviewer_name
## 1      69998  881474 2012-01-24     1595616   Jean-Pierre
## 2     548218 2565139 2012-10-09     3679427   Nils Gunnar
## 3      69998 4007103 2013-04-02     3868130   Jo And Mike
## 4      69998 4170371 2013-04-15     5730759     Elizabeth
## 5      69998 4408459 2013-05-03     5921885          Jone
## 6      69998 4485779 2013-05-07      810469        Andrea
##
## 1
## 2
## 3 We had a four night stay at this gorgeous apartment and it was absolutely perfect. It's really pre
## 4                                                                                                 Lor's
## 5
## 6         My boyfriend and I, had a lovely stay at Lorenzo's flat in Mallorca. It really did feel li
```

Ahora, vamos a crear la tibble con las frecuencias de cada longitud de palabras.

```r
# Contar frecuencias
word_freq <- filtered_reviews %>%
  group_by(comment_length) %>%
```

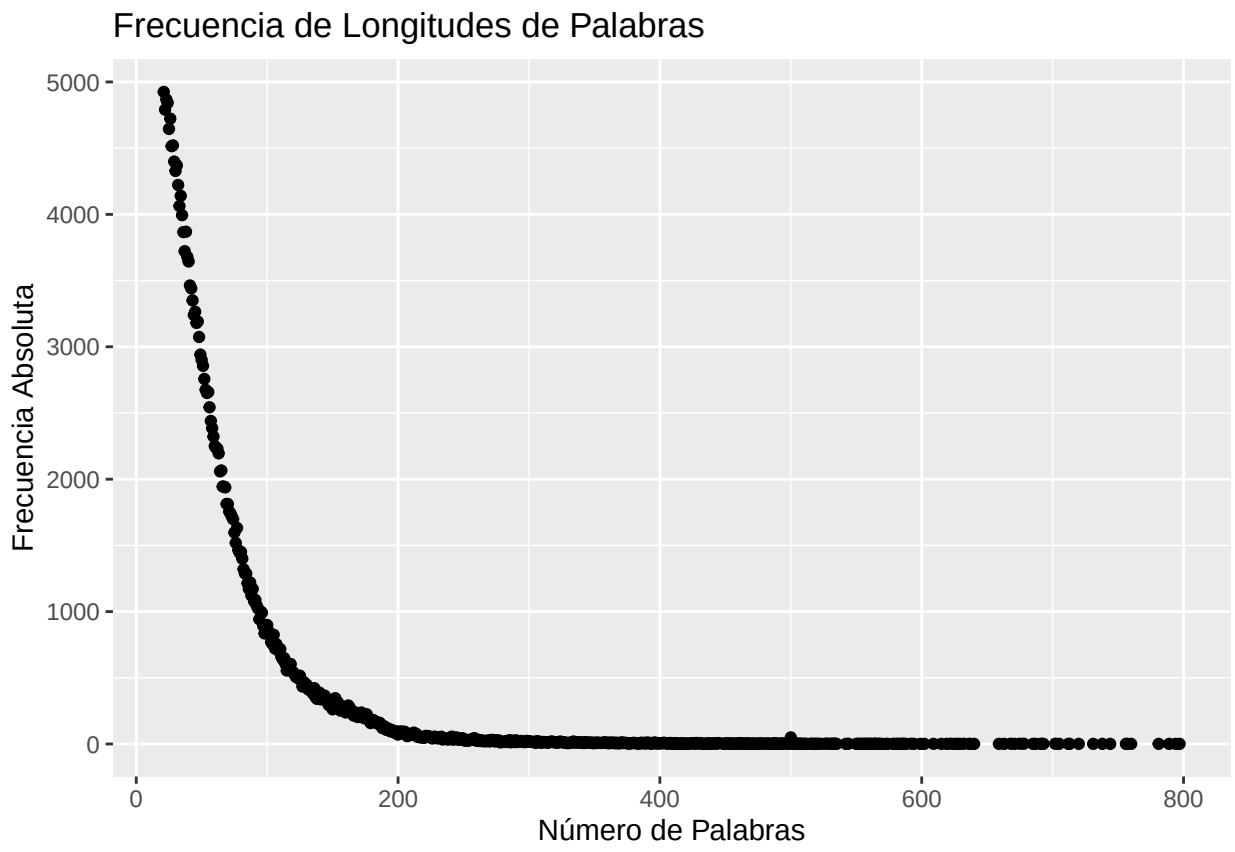
```r
  summarise(frequency = n())

# Verificar la tibble de frecuencias
head(word_freq)
```

```
## # A tibble: 6 x 2
##   comment_length frequency
##            <int>     <int>
## 1             21      4925
## 2             22      4791
## 3             23      4868
## 4             24      4842
## 5             25      4645
## 6             26      4723
```

Ahora, visualizaremos los datos.

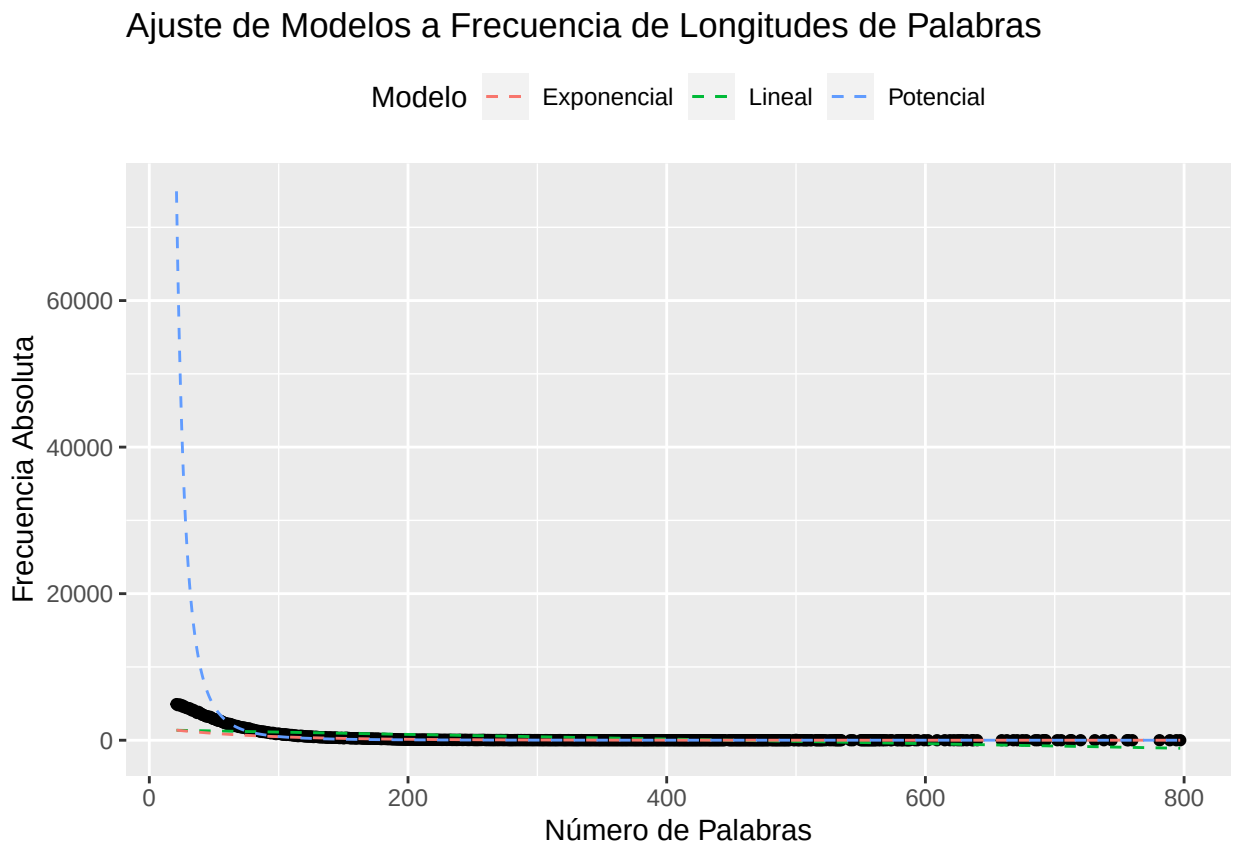
```r
# Visualizar los datos
ggplot(word_freq, aes(x = comment_length, y = frequency)) +
  geom_point() +
  labs(title = "Frecuencia de Longitudes de Palabras",
       x = "Número de Palabras",
       y = "Frecuencia Absoluta")
```



Apartado 2

```r
# Ajustar modelos lineales, exponenciales y potenciales
linear_model <- lm(frequency ~ comment_length, data = word_freq)
exp_model <- lm(log(frequency) ~ comment_length, data = word_freq)
power_model <- lm(log(frequency) ~ log(comment_length), data = word_freq)

# Agregar líneas de ajuste a la gráfica con leyenda
ggplot(word_freq, aes(x = comment_length, y = frequency)) +
  geom_point() +
  geom_line(aes(x = comment_length, y = predict(linear_model), color = "Lineal"), linetype = "dashed") +
  geom_line(aes(x = comment_length, y = exp(predict(exp_model)), color = "Exponencial"), linetype = "das
  geom_line(aes(x = comment_length, y = exp(predict(power_model)), color = "Potencial"), linetype = "das
  labs(title = "Ajuste de Modelos a Frecuencia de Longitudes de Palabras",
       x = "Número de Palabras",
       y = "Frecuencia Absoluta",
       color = "Modelo") +
  theme(legend.position = "top")
```



Apartado 3

```r
# Calcular el rango de N_words
word_freq <- word_freq %>%
  mutate(rank_n_words = rank(comment_length))

# Ajustar modelos lineales, exponenciales y potenciales con el rango
linear_model_rank <- lm(frequency ~ rank_n_words, data = word_freq)
exp_model_rank <- lm(log(frequency) ~ rank_n_words, data = word_freq)
```

```r
power_model_rank <- lm(log(frequency) ~ log(rank_n_words), data = word_freq)

# Agregar líneas de ajuste a la gráfica con leyenda
ggplot(word_freq, aes(x = comment_length, y = frequency)) +
  geom_point() +
  geom_line(aes(x = comment_length, y = predict(linear_model_rank), color = "Lineal"), linetype = "dash
  geom_line(aes(x = comment_length, y = exp(predict(exp_model_rank)), color = "Exponencial"), linetype =
  geom_line(aes(x = comment_length, y = exp(predict(power_model_rank)), color = "Potencial"), linetype =
  labs(title = "Ajuste de Modelos a Frecuencia de Longitudes de Palabras",
       x = "Número de Palabras",
       y = "Frecuencia Absoluta",
       color = "Modelo") +
  theme(legend.position = "top")
```