PRÁCTICA 2 – Limpieza y análisis de datos

Tipología y ciclo de vida de los datos

Semestre 2018/2019-1

Luis Manuel Martín Guerra

Índice

1. Descripción del dataset

Este dataset se generó en la Práctica 1 de la asignatura “Tipología y ciclo de vida de los datos”. Es interesante para todas aquellas personas que quieran obtener información sobre el precio de los “*Gundam Plastic Models*” o “*Gunplas*” que existen actualmente en el mercado y cuales son los factores que determinan el precio de las mismas.

El dataset y el código para obtenerlo se encuentran en la siguiente dirección:

<dirección>

Si consultamos los datos en origen, podemos comprobar que la media del número de kits de la escala 1/144 es superior a la del resto de escalas que se fabrican, siendo por tanto la más popular entre los aficionados.

En esta práctica partiremos del dataset original y trataremos los datos, unificándolos, limpiándolos y transformándolos para emplear en la estimación de un modelo que nos permita, a partir de los datos, poder predecir el precio de futuros “gunplas” que se pongan a la venta, o bien valorar cual el precio de nuestra colección de cara a contratar un seguro que cubra los daños o desperfectos a la misma.

2. Integración y Selección de los datos

El fichero de datos original contiene 2862 registros y 11 variables.

Las variables son: Nombre, Fabricante, Escala, Serie, Original, Fecha de Lanzamiento, Precio Venta, Precio Especial, Código JAN, Embalaje y Peso.

Debido a que el fichero .csv que se generó en la práctica anterior tiene valores de campos desplazados, debido a las variaciones de las páginas web de las que se hizo Web Scrapping, el tratamiento de los datos y parte de la limpieza de los mismos se llevará a cabo usando Excel y el editor de textos SublimeText, tal y como se nos recomienda en el libro de “Clean Data”. El resto de transformaciones y modificaciones se llevará a cabo en R.

Cuando abrimos el fichero .csv en Excel observamos lo siguiente:

2666	,sO,null,null,null					
2667	Luggun and Sealanth	Bandai	1/144	EX Model	Mobile Suit Gundam	Late Jan., 2003
2668	Duel Gundam	Bandai	1/144	Quick Gundam Model	Mobile Suit Gundam	Late Nov., 2002
2669	Strike Gundam	Bandai	1/144	Quick Gundam Model	Mobile Suit Gundam	Late Nov., 2002
2670	HY2M-MG W02	Bandai	1/100	HY2M	Late Oct., 2002	1,620yenabout14
2671						
2672	,sO,null,null,null					
2673	HY2M-MG09	Bandai	1/100	HY2M	Mid Dec., 2002	1,620yenabout14
2674						
2675	,sO,null,null,null					
2676	HY2M-MG08	Bandai	1/100	HY2M	Late Nov., 2002	1,620yenabout14
2677						
2678	,sO,null,null,null					
2679	HY2M-MG07	Bandai	1/100	HY2M	Late Oct., 2002	1,620yenabout14
2680						
2681	,sO,null,null,null					
2682	HY2M-MG06	Bandai	1/100	HY2M	Late Sep., 2002	1,620yenabout14
2683						
2684	,sO,null,null,null					
2685	HY2M-MG05	Bandai	1/100	HY2M	1,620yenabout14.21USD	
2686						
2687	1	350yenabout11.84USD				

Vemos que hay entradas que tienen todos los campos correctamente informados y otras entradas en las que los valores se han desplazado al campo inmediatamente siguiente, provocando un desplazamiento de los datos. Así nos encontramos con entradas que en el campo “Fecha de lanzamiento” está el valor del importe del producto, o que el valor del código JAN, las dimensiones del paquete y su peso, se encuentran en una entrada aparte, que no sigue ningún tipo de estructura.

Por lo tanto tenemos que acometer dos tareas diferenciadas:

a) Tabular correctamente los datos que se hayan movido erróneamente dentro de cada entrada.

b) Mover los datos referentes a las dimensiones, código y peso a la entrada correspondiente, si esta ha sido dividida.

Lo primero que haremos será recuperar los datos referentes a cada entrada, que por error se han añadido como otra entrada

23	RX-78-2 Gundam	Bandai	Master Grade , 1/100	MG Gundam	Mobile Suit Gundam	Jul., 1995	23.69USD	19.74USD	4,90243E+12	31.5 x 20.3 x 8.5 cm	479g
24	RX-78NT1 Gundam NT-1	Bandai	Master Grade , 1/100	MG Other Gundam	Mobile Suit Gundam	3,240yenabout28.43USD	23.69USD				
25											
26	,sO,0070949, 31 x 20 x 10 cm , 612g										

Para ello copiaremos la información correspondiente en un nuevo campo llamado AUX, que emplearemos temporalmente como depósito de esta información.

Una vez integrada la información de todas las entradas, tabularemos correctamente los datos de los campos Fecha de lanzamiento y Precio Venta de aquellas entradas que hayan podido quedar afectadas. Esto se produce debido a que cuando se realizó el raspado de datos, nos encontramos con páginas que tenían dichos campos en blanco y por lo tanto al guardarlo se produjo esta desplazamiento.

Ahora ya tenemos el dataset sin huecos ni datos desplazados, únicamente nos falta extraer la información correspondiente al campo Código JAN, Embalaje y Peso de aquellos entradas que tenían la información desplazada.

En este punto emplearemos el editor de texto SublimeText, en el que copiaremos el contenido de la columna AUX, que contiene tanto el código JAN correcto de unas entradas, como el Código JAN, el Embalaje y el peso de las entradas que tuvieron errores en la obtención.
A continuación sustituimos los caracteres *,sO* y sustituiremos los caracteres 'x' por ',' para, de esta manera, ya delimitar por ',' el código JAN, el valor del Alto, Ancho y Fondo de las cajas y por último, el peso en gramos de la caja.

Modificamos la primera fila, que será la cabecera del fichero auxiliar, para que incluya el nombre de los campos que vamos a extraer, así queda la primera fila como: CODIGO JAN, ALTO, ANCHO, FONDO, PESO.

Este fichero lo guardamos con el nombre de aux.csv y a continuación lo importamos desde Excel, como un fichero .csv más, indicando que el separador de campos es el carácter ','.

De esta manera ya tenemos todos los campos con los datos correctamente tabulados y además hemos separado las dimensiones de la caja, que calcularemos en un campo que llamaremos "Embalaje" y que será el resultado de multiplicar el ancho, el alto y el fondo de las cajas.

Con esto obtenemos el dataset, con valores en algunas de las entradas de 0. Ya que bien falta el peso, o el precio o el embalaje. Ordenamos el dataset en Excel y eliminamos los valores de 0 correspondientes, dejando el valor en blanco o "nulo" para que cuando carguemos el dataset en R, este detecte la existencia de entradas con campos sin valor.

Así pues, de las variables que comentamos anteriormente, eliminaremos Fabricante, puesto que es el mismo para todos los registros (Bandai), Original, ya que todos los registros tienen el mismo valor (Mobile Suit Gundam) y, finalmente Precio Especial, ya que el precio que nos interesa es el precio estándar de PVP fijado por el fabricante.

Finalmente, mediante SublimeText formateamos el campo Escala, para quedarnos con el valor de la escala(1/100, 1/60, etc.) y el nivel de detalle en otro (Perfect Grade, Master Grade, etc.).

Por último eliminamos el campo Código JAN, ya que no nos aparta información de ningún valor de cara al análisis que vamos a realizar.

A continuación vamos a eliminar aquellas escalas de las que únicamente hay 2 o 3 referencias, ya que no representarán un gran impacto en los resultados y sí que pueden distorsionarlos con valores extremos. Ordenamos el dataset en Excel por escala y eliminamos todas las entradas que no se correspondan con las principales escalas compradas y coleccionadas: 1/48, 1/60, 1/100, 1/144, y 1/1700.

Para facilitar la lectura y el trabajo con las variables, vamos a crear un nuevo campo llamado Factor_Escala, que será el equivalente numérico de las correspondientes escalas en las que se producen los kits:

- 1/48 →1
- 1/60 →2
- 1/100 →3
- 1/144 →4
- 1/1700→5

El siguiente paso es quedarnos únicamente con los registros que corresponden a maquetas completas y no a accesorios o complementos para las mismos.

De esta manera ya tenemos los datos formateados y listos para proceder a la selección de variables que emplearemos y a la limpieza final de los mismos. Volvemos a filtrar el Excel por Serie y seleccionamos toda la columna de datos, la exportamos a SublimeText y eliminamos los caracteres sobrantes, para quedarnos únicamente con el nombre de la serie de maquetas a la que se corresponde el producto.

Finalmente guardamos el Excel como gunpla_clean.csv y ya tenemos los datos pre-procesados y listos para continuar con la limpieza.

	Nombre	Factor_Escala	Escala	Detalle	Serie	Year	Embalaje	Peso	Precio
2	Mega Size Model Char's Zaku	1	1/48	Mega Size	Mega Size Model	2010	27797,76	1789	73,91
3	Mega Size Model Gundam AGE-1 Norm	1	1/48	Mega Size	Mega Size Model AGE	2011	22455,64	1576	80,54
4	Mega Size Model Gundam AGE-2 Norm	1	1/48	Mega Size	Mega Size Model AGE	2012	26998,4	1979	83,38
5	Mega Size Model RX-78-2 Gundam	1	1/48	Mega Size	Mega Size Model	2010		1498	73,91
6	Mega Size Model Zaku II	1	1/48	Mega Size	Mega Size Model	2011	27724,8	1872	73,91
7	00 Raiser	2	1/60	Perfect Grade	Perfect Grade (PG)	2009	48510	4380	236,88

3. Limpieza de los datos

Una vez integrados los datos y pre-procesados tenemos un dataset con 1366 entradas y 8 variables:

```
gunpla <- read.table("/Users/manu/Documents/UOC - Ciencia de Datos/2 - Tipología y
ciclo de vida de los datos/PRACTICA 2/R/gunpla_clean_agrupado.csv",
header=TRUE, sep=";", na.strings="NA", dec=",", strip.white=TRUE)
NOTA: El conjunto de datos gunpla tiene 1360 filas y 9 columnas.
```

Ahora vamos a ver que los tipos de datos que se ha asignado a cada variable durante la lectura del fichero, es el adecuado:

```
var_class <-sapply(gunpla, class)
kable(data.frame(variable=names(var_class), tipo=as.vector(var_class)))
|variable      |tipo    |
|:-------------|:-------|
|Nombre        |factor  |
|Factor_Escala |integer |
|Escala        |factor  |
|Detalle       |factor  |
|Serie         |factor  |
|Year          |integer |
|Embalaje      |numeric |
|Peso          |integer |
|Precio        |numeric |
```

- Nombre: Es el nombre del modelo que se monta con el kit.
- Factor_Escala: el número asociado a la escala (1,2,3,4, o 5)
- Escala: La escala en que está fabricada el modelo: (1/60, 1/100, etc.)
- Detalle: Nivel de detalle del modelo: (Master Grade, Real Grade, etc.)
- Serie: Nombre de la línea de producto a la que pertenece.
- Year: Año de puesta a la venta.
- Embalaje: Volumen en cm^3 de la caja del producto.
- Peso: Peso en gramos del producto, incluyendo el embalaje.
- Precio: PVP en dólares del producto.

Vemos que los atributos relacionados con el embalaje y el precio ya se encuentran en formato numérico, pero el peso no, por lo que vamos a convertirlo también en numérico:

```
gunpla[8]<-lapply(gunpla[8], as.numeric)
var_class <-sapply(gunpla, class)
kable(data.frame(variable=names(var_class), tipo=as.vector(var_class)))


|variable      |tipo    |
|:-------------|:-------|
|Nombre        |factor  |
|Factor_Escala |integer |
|Escala        |factor  |
|Detalle       |factor  |
|Serie         |factor  |
|Year          |integer |
|Embalaje      |numeric |
|Peso          |numeric |
|Precio        |numeric |
```

El siguiente paso es ver cuantos valores nulos, ceros o valores extremos nos encontramos en el dataset.

3.1 Valores Nulos

Veamos que campos de nuestro dataset contienen valores nulos:

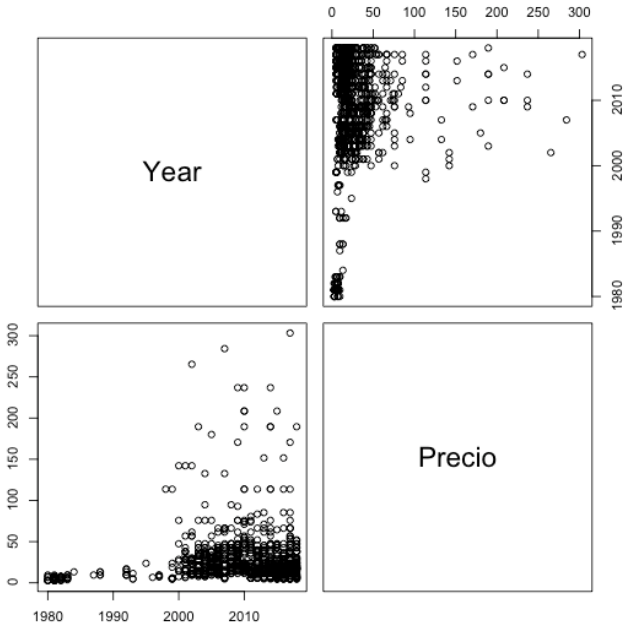
```
sapply(gunpla, function(x)(sum(is.na(x)))) # NA counts
       Nombre Factor_Escala        Escala       Detalle         Serie
            0             0             0             0             0
       Year      Embalaje          Peso        Precio
        290           105             0             0
```

Tenemos que tanto Year, como Embalaje contienen valores nulos

Antes de decidir que hacer con los nulos de Year y Embalaje, vamos a ver que relación pueden tener entre sí

a) Relación entre el año de fabricación y el precio del Kit:

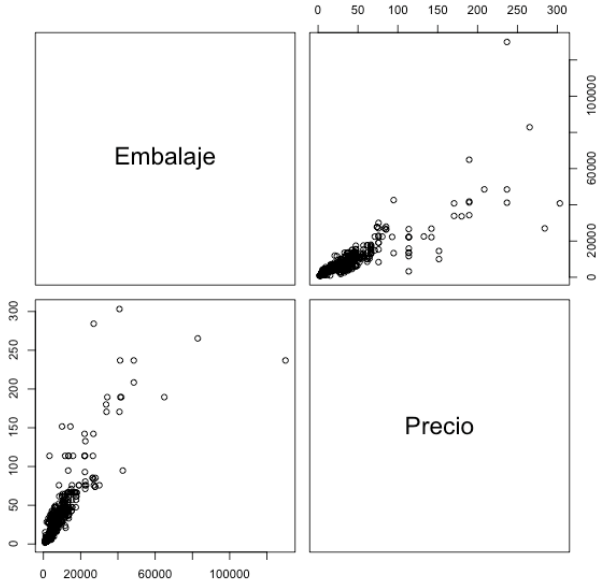
```
gunpla_test <-gunpla[,c("Year","Precio")]
pairs(gunpla_test)
```



Lo que vemos en esta gráfica es que fue a partir de finales de los noventa y principios del 2000 cuando se empezaron a producir kits de precios superiores, pero nada más.

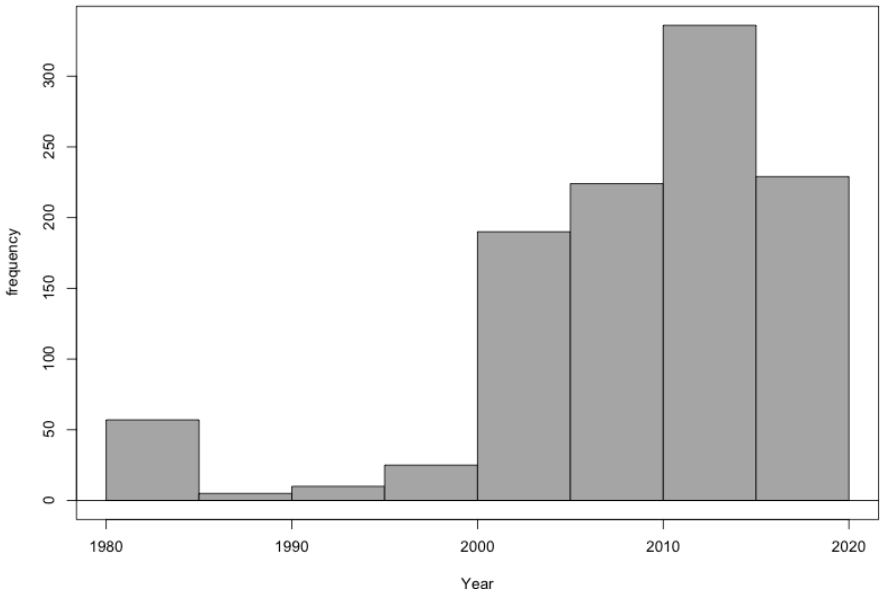
b) Relación entre el volumen del embalaje y el precio:

```
gunpla_test <-gunpla[,c("Embalaje","Precio")]
pairs(gunpla_test)
```



En este caso vemos que la relación entre el volumen del embalaje y el precio del kit es lineal, lo cual tiene sentido, ya que a mayor volumen, mayor cantidad de matrices contendrá.

A la vista de estos datos, concluyo que el año en el que fue puesto a la venta el kit no tiene incidencia a la hora de asignar el precio, únicamente, nos dice que a partir del año 2000 se empezaron a producir nuevos kits:



Por lo tanto vamos a eliminar los campos con valores nulos de Year, y vamos a trabajar con el campo Embalaje.

```
gunpla <- within(gunpla, {Year <- NULL  })
```

Y con respecto al campo Embalaje, como hemos visto anteriormente, tiene una relación lineal con el precio del producto, por lo tanto, perderíamos información si eliminásemos esas entradas del dataset.

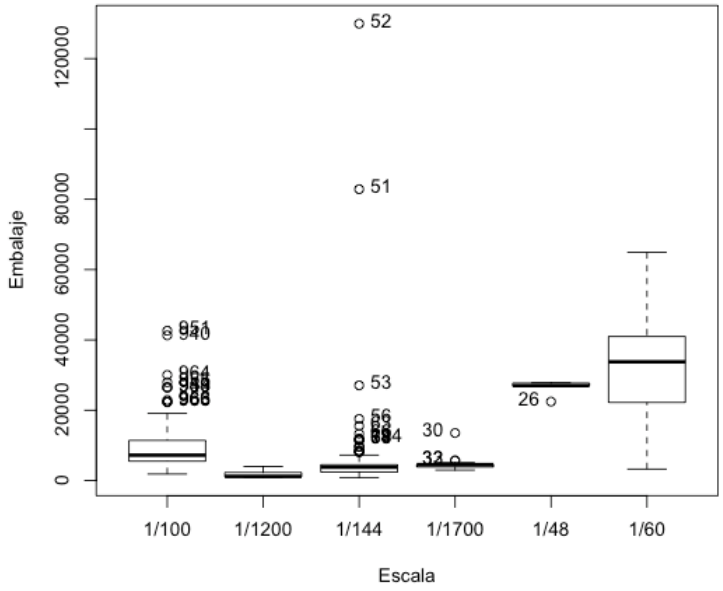
Por lo tanto, emplearemos el método de la imputación de valores basados en la similitud o diferencia entre los registros: la llamada imputación basada en k-vecinos más próximos. Antes de ello, vamos a ordenar los registros del dataset por peso, para que así los datos sobre el volumen que se generen sean los más próximos posibles a unos datos reales, para finalmente generar los datos aproximados:

```
# Ordenamos el dataset por el peso de los kits en orden ASCENDENTE
gunpla <- with(gunpla, gunpla[order(Peso, decreasing=FALSE), ])

# Generamos los datos en función de los valores de los vecinos más próximos.
gunpla$Embalaje <- kNN(gunpla)$Embalaje

# Valores desconocidos por campo
sapply(gunpla, function(x)(sum(is.na(x))))
Nombre Factor_Escala        Escala       Detalle         Serie
     0             0             0             0             0
Embalaje          Peso        Precio
       0             0             0
```

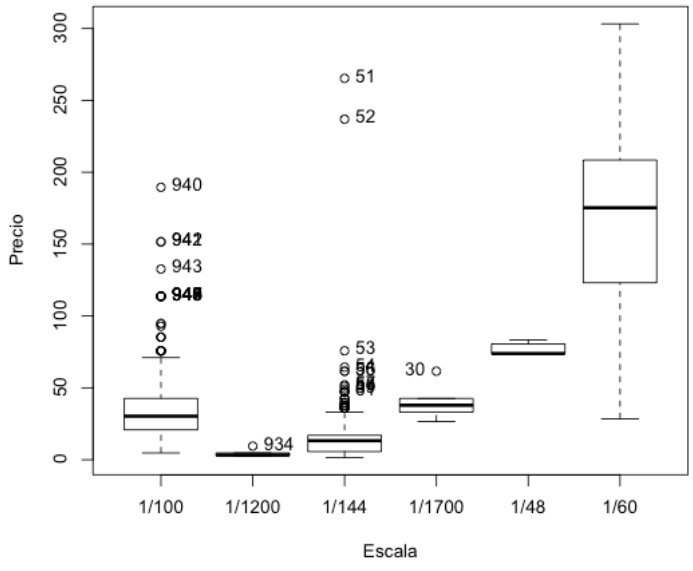
3.2 Valores extremos

Para identificar los valores extremos, vamos a emplear utilizar un diagrama de caja por cada variable, que nos mostrará los valores atípicos, en el caos de que existan, de las variables que los contienen, agrupándolos por la escala del producto:

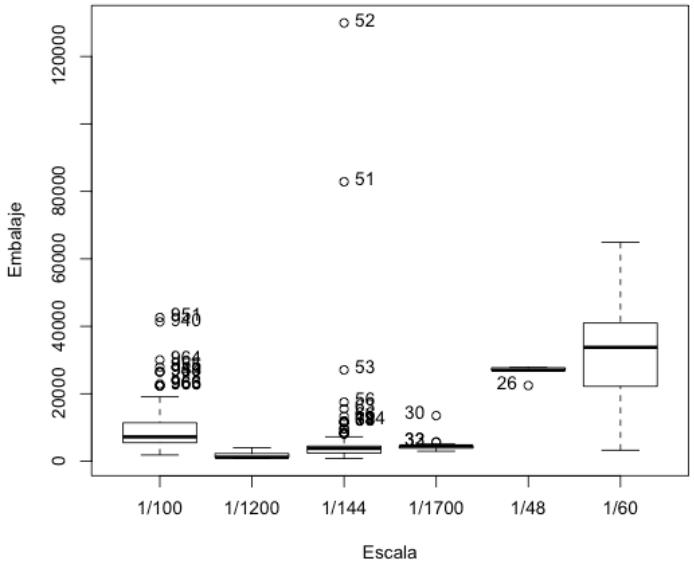
```
Boxplot(Embalaje~Escala, data=gunpla, id.method="y")
 [1] "951" "940" "964" "955" "949" "953" "954" "956" "966" "963" "52"  "51"
[13] "53"  "56"  "63"  "66"  "194" "77"  "68"  "54"  "32"  "33"  "30"  "26"
```




```
Boxplot(Precio~Escala, data=gunpla, id.method="y")
 [1] "940" "941" "942" "943" "944" "945" "946" "947" "949" "948" "934" "51"
[13] "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "61"  "30"
```



```
Boxplot(Peso~Escala, data=gunpla, id.method="y")
[1] "940" "951" "955" "964" "963" "949" "959" "960" "953" "954" "52"  "51"
[13] "53"  "63"  "56"  "73"  "77"  "66"  "68"  "60"  "49"  "50"  "32"  "33"
[25] "30"
```



Si revisamos los datos para diferentes productos, vemos que los resultados encajan perfectamente, ya que hay kits que aunque corresponden a una escala concreta, como la 1/100, debido al modelo que reproducen, su peso, embalaje y por lo tanto precio son mayores de lo esperado, ya que de existir, el tamaño real de este modelo sería muy superior al de la media de los modelos fabricados a esa escala.

Por lo tanto, la decisión que tomamos a la hora de manejar estos valores extremos es dejarlos tal cual están recogidos en el dataset.

4. Análisis de los datos

4.1 Selección de los grupos de datos

Después de haber analizado las relaciones entre los diferentes campos numéricos y no numéricos del dataset, hemos decidido que los grupos de datos que analizaremos son: Precio, Embalaje, Escala y Detalle.

4.2 Comprobación de la normalidad y la homegeneidad de la varianza

Para comprobar si las variables están normalizadas vamos a aplicar el test de Shapiro Wilk en cada una de las variables numéricas de nuestro dataset:

```
shapiro.test(gunpla$Peso)

       Shapiro-Wilk normality test

data:  gunpla$Peso
W = 0.50132, p-value < 2.2e-16

##
shapiro.test(gunpla$Precio)

       Shapiro-Wilk normality test

data:  gunpla$Precio
W = 0.56566, p-value < 2.2e-16

##
shapiro.test(gunpla$Embalaje)

       Shapiro-Wilk normality test

data:  gunpla$Embalaje
W = 0.50154, p-value < 2.2e-16
```

Los test realizados nos indican que ninguna variable está normalizada, ya que para todas ellas el `p-value` obtenido es inferior al coeficiente de 0.05, con lo que podemos rechazar la hipótesis nula y por lo tanto entender que no es normal.

4.3 Aplicación de pruebas estadísticas

Una vez llegados a este punto vamos a tratar de averiguar cuales son las variables que influyen más en el precio final del producto.

Para ello va

Para ello vamos a utilizar una matriz de correlación que utiliza el coeficiente de correlación de Spearman, debido a que hemos verificado en el apartado anterior que tenemos datos que no siguen una distribución normal:

```
cor(gunpla[,c("Embalaje","Factor_Escala","Peso","Precio")],
method="spearman", use="complete")
               Embalaje Factor_Escala       Peso     Precio
Embalaje       1.0000000    -0.6269885  0.9769858  0.9209985
Factor_Escala -0.6269885     1.0000000 -0.6416655 -0.5267124
Peso           0.9769858    -0.6416655  1.0000000  0.9276507
Precio         0.9209985    -0.5267124  0.9276507  1.0000000
```

Con los resultados vemos, que los factores que más influyen en el precio es el Peso y el Embalaje del producto, los cuales influyen el uno en el otro.

A continuación vamos a crear un modelo de regresión lineal que emplee las variables que están más relacionadas con el precio, partiendo de la tabla que hemos obtenido anteriormente, es decir, el Embalaje y el Peso.

```
modelo_gunpla <- lm(Precio~Embalaje+Escala+Peso, data=gunpla_agrupado)
summary(modelo_gunpla)

Call:
lm(formula = Precio ~ Embalaje + Factor_Escala + Peso, data = gunpla)

Residuals:
     Min       1Q   Median       3Q      Max
-207.399   -4.241   -1.254    1.111  170.180

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   25.7357659  3.1428902   8.189 6.04e-16 ***
Embalaje       0.0010874  0.0003071   3.541 0.000412 ***
Factor_Escala -6.2815425  0.7933548  -7.918 5.00e-15 ***
Peso           0.0311085  0.0040235   7.732 2.06e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.03 on 1356 degrees of freedom
Multiple R-squared:  0.7722, Adjusted R-squared:  0.7717
F-statistic:  1532 on 3 and 1356 DF,  p-value: < 2.2e-16
```

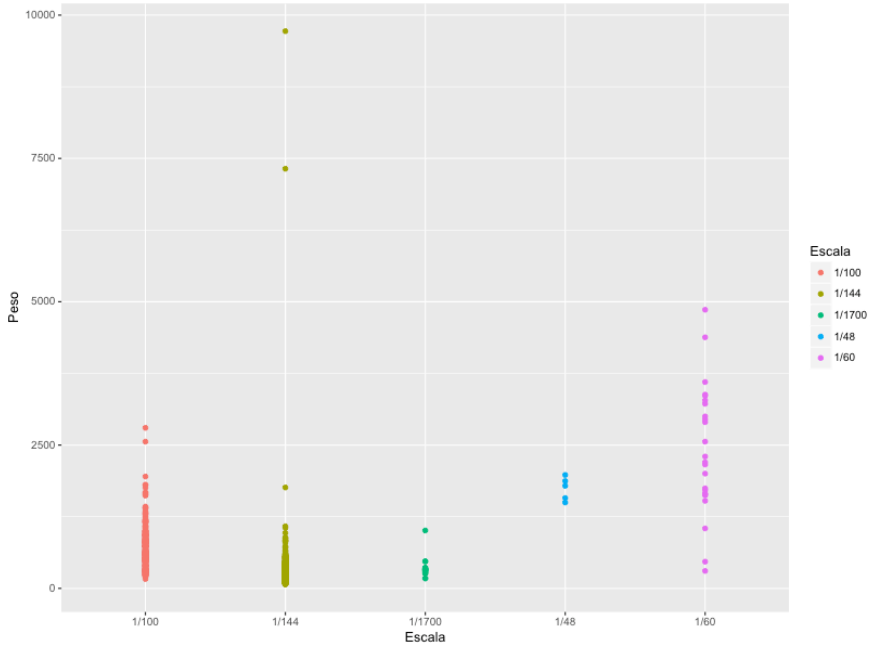
Finalmente probamos el modelo para ver si es capaz de predecir precios de kits:

```
test_gundam <-data.frame(Nombre="Test", Factor_Escala=4,Detalle="High Grade",
Serie="Gundam SEED", Embalaje=1600, Peso =140, Precio=5.3)

predict(modelo_gunpla,test_gundam)
       1
6.704583
```

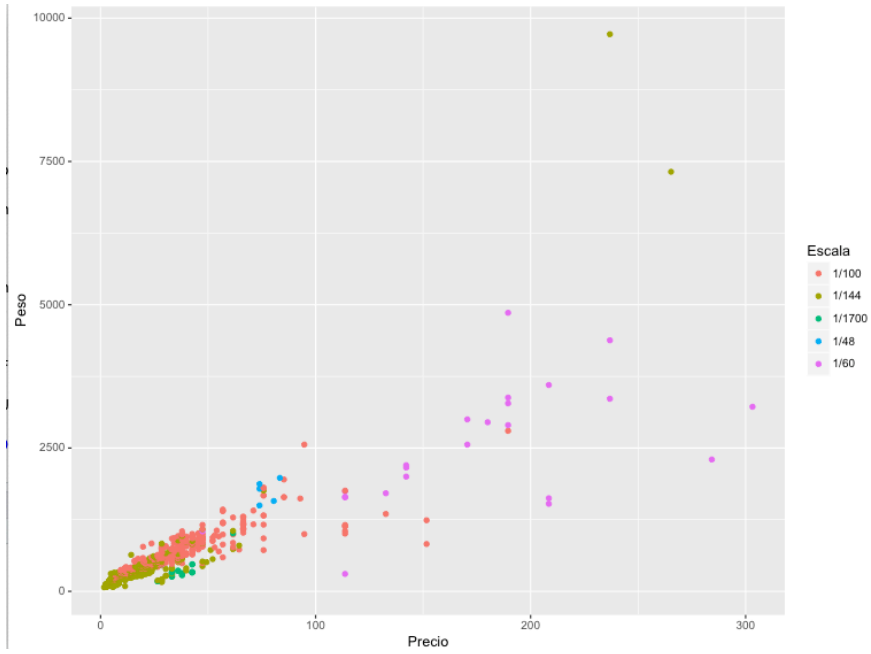
5. Representación de resultados a partir de tablas y gráficas

5.1 Diagramas de dispersión

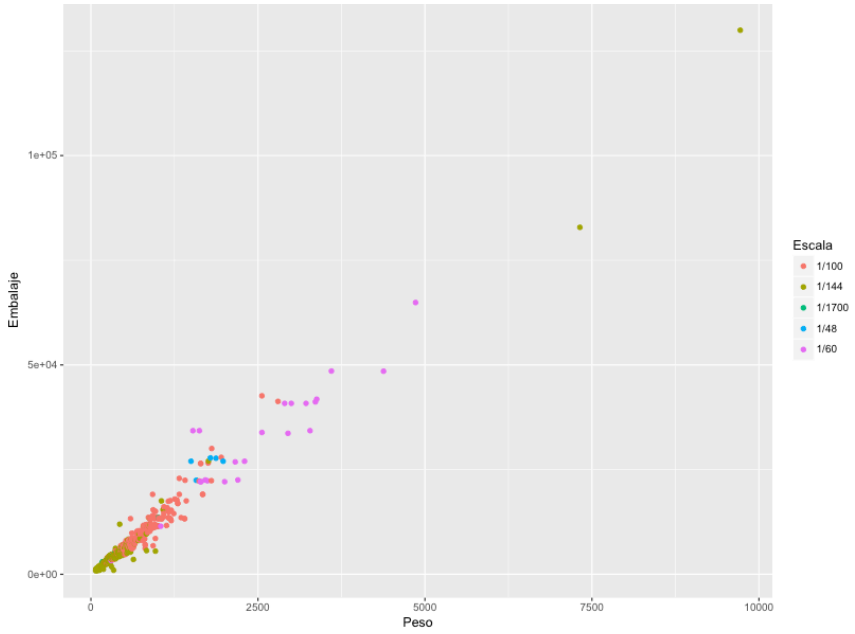
Veamos el diagrama de relación entre el peso y la escala de los kit:



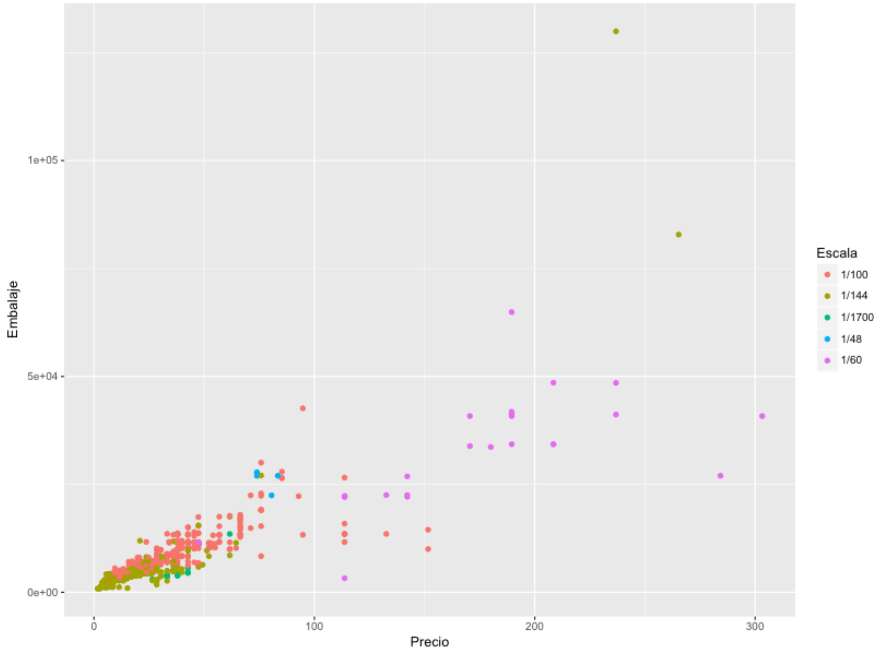
En este diagrama observamos la relación entre el precio de un kit y el peso del mismo:



En este otro diagrama de dispersión vemos la relación entre el peso y el embalaje, agrupado por escalas:

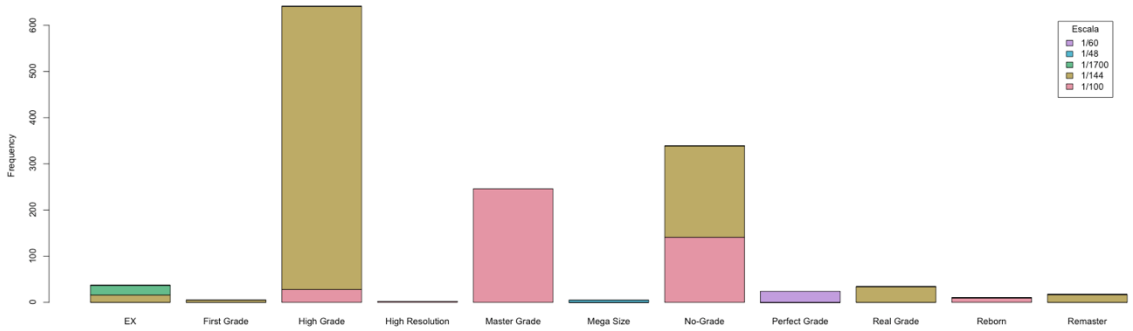


Finalmente, en este otro visualizamos la relación entre el peso y el embalaje:



En todos ellos vemos que existe una relación directa entre el precio y el embalaje, así como entre el precio y el peso.

5.2 Diagramas de Barras



En este diagrama de barras, podemos observar que cuales son las escalas más fabricadas y el nivel de detalle de las mismas. Así, vemos que las más producidas son las maquetas de escala 1/144 y nivel de detalle High Grade, seguidas por las de escala 1/100 y nivel de detalle Master Grade.

6. Resolución del problema

Una vez llegados a este punto tenemos claro que el precio de una maqueta está condicionado por el peso y el tamaño del embalaje de la misma. Además sabemos que según la escala en la que se fabrique una maqueta, o se planee lanzar, cual puede ser su peso aproximado, con lo que podemos concluir que podemos extrapolar el precio de casi cualquier kit de *gunpla* que se ponga a la venta, a partir de las dimensiones de la caja, o el peso de la misma y la escala en la que se va a fabricar.

Finalmente se procede a exportar el dataset modificado para la salida.

7. Código

El código en R empleado en esta práctica está incluido en un fichero con extensión .r y se puede descargar desde la siguiente dirección:

https://github.com/manumarting/UOC-Gunpla-Analysis/blob/master/code/gunpla_analysis.r

Los datos de salida se exportan a un fichero .csv que se puede descargar desde el repositorio en la siguiente dirección**:**

https://github.com/manumarting/UOC-Gunpla-Analysis/blob/master/data/gunpla_final.csv