Máster Universitario en Big Data y Ciencia de Datos

Actividad 1 - Estadística Avanzada

Alumno: Sellés Pérez, Alejandro

Edición Octubre 2024 a 24/02/2024

Índice

1. Introducción

A lo largo de esta actividad, nos pondremos en la piel de una multinacional especializada en la compra y reforma de pisos para su posterior alquiler en plataformas como Booking o Airbnb. Nos enfrentaremos a distintas situaciones clave para optimizar nuestra rentabilidad.

En primer lugar, evaluaremos la adquisición de nuevos pisos, estimando la rentabilidad diaria que podríamos obtener de ellos. Además, analizaremos propiedades que ya están en la plataforma para determinar si su precio es adecuado o si debería ajustarse. Para ello, desarrollaremos varios modelos de regresión que nos permitan predecir el precio óptimo de alquiler en función de diversas variables.

Por otro lado, partiremos de un dataset con datos de cancelaciones de reservas previas de nuestra compañía. Y, nuestro objetivo será construir un modelo de regresión logística capaz de predecir la probabilidad de cancelación de una nueva reserva. De esta manera, podremos anticiparnos a posibles cancelaciones y tomar decisiones estratégicas para minimizar su impacto. Este análisis nos permitirá gestionar mejor la disponibilidad de las propiedades, pudiendo jugar con el overbooking y maximizar la ocupación en épocas de alta demanda.

2. Regresión Lineal y Polinómica

2.1. Descripción del Dataset

A lo largo de esta sección nos centraremos en buscar modelos de regresión en los que trataremos de predecir el precio de una habitación. Para ello, tomaremos ***Hotels.xlsx***, que consta 120 filas y 9 columnas. Sus variables son:

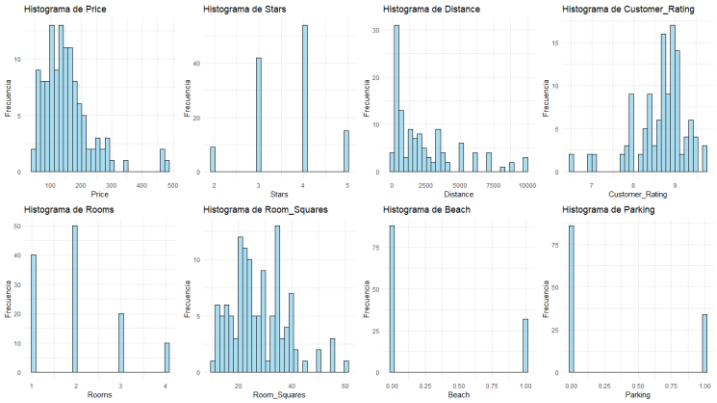
1. ID (int): Identificador único de cada habitación o piso.
2. Price (int): Precio actual por habitación
3. Stars (int): Número de estrellas en la app
4. Distance (int): Distancia al centro de la ciudad (m)
5. Customer_Rating (float): Puntuación basada en opiniones de clientes.
6. Rooms (int): Cantidad de habitaciones en el piso.
7. Room_Squares (int): Superficie total de la habitación o piso en m².
8. Beach (boolean): Indicador (Sí/No) sobre si la playa está cerca.
9. Parking (boolean): Indicador (Sí/No) sobre disponibilidad de estacionamiento.

Al realizar un análisis descriptivo del conjunto de datos mediante un *summary*, obtenemos los siguientes resultados:

ID	Price	Stars	Distance	Customer_Rating	Rooms	Room_Squares	Beach	Parking
Min. : 1.00	Min. : 39.0	Min. :2.000	Min. : 150	Min. :6.500	Min. :1.00	Min. :10.00	Min. :0.0000	Min. :0.0000
1st Qu.: 30.75	1st Qu.: 99.5	1st Qu.:3.000	1st Qu.: 450	1st Qu.:8.400	1st Qu.:1.00	1st Qu.:20.00	1st Qu.:0.0000	1st Qu.:0.0000
Median : 60.50	Median :140.0	Median :4.000	Median : 1550	Median :8.700	Median :2.00	Median :26.00	Median :0.0000	Median :0.0000
Mean : 60.50	Mean :153.3	Mean :3.625	Mean : 2354	Mean :8.643	Mean :2.00	Mean :27.52	Mean :0.2667	Mean :0.2833
3rd Qu.: 90.25	3rd Qu.:180.0	3rd Qu.:4.000	3rd Qu.: 3400	3rd Qu.:9.000	3rd Qu.:2.25	3rd Qu.:35.00	3rd Qu.:1.0000	3rd Qu.:1.0000
Max. :120.00	Max. :474.0	Max. :5.000	Max. :10000	Max. :9.700	Max. :4.00	Max. :60.00	Max. :1.0000	Max. :1.0000

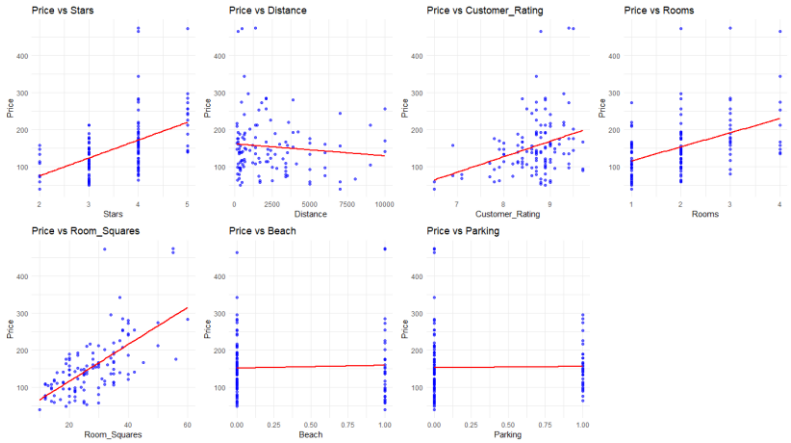
La variable objetivo de nuestro análisis será *"Price"*. Al examinar sus valores, observamos que los precios de las habitaciones varían entre 39$ y 470$ por noche, con una

media de 153,3$. Para comprender mejor la distribución de las variables en el conjunto de datos, procederemos a analizarlas a través de sus histogramas.



Podemos observar que la variable "*Customer Rating"* sigue una distribución aproximadamente normal, con un valor promedio cercano a 8,7. En cuanto a los precios, la mayoría se sitúan en un rango entre 100$ y 200$ por noche. Además, la distribución geográfica de los pisos indica que la mayoría se encuentran relativamente cerca del centro de la ciudad. Por otro lado, una gran parte de las propiedades no disponen de estacionamiento ni están ubicadas en zonas cercanas a la playa.

A continuación, examinaremos gráficos de dispersión para analizar las relaciones entre las diferentes variables y nuestra variable objetivo, *"Price".* Esto nos permitirá identificar posibles patrones o correlaciones que puedan influir en la predicción del precio de las habitaciones.



En términos generales, los resultados parecen coherentes: a mayor distancia del centro, el valor de *Price* tiende a disminuir. Del mismo modo, el precio aumenta conforme se incrementan variables como *Rooms*, *Room Squares* y *Customer Rating*. Por otro lado, al observar los datos, no se aprecia una relación significativa entre la disponibilidad de *Parking* o la proximidad a la *Beach* y el precio de las habitaciones.

Finalmente, al visualizar la matriz de correlaciones, podemos confirmar las observaciones previas. No se aprecia una relación significativa entre la variable objetivo, *Price*, y las variables *Parking* y *Beach*, lo que refuerza la idea de que estos factores no influyen de manera relevante en el precio de las habitaciones. Además, no se ve una correlación excesivamente alta entre ninguna de las variables predictoras.

2.2. Modelos de Regresión

Como mencionamos al inicio, a lo largo de este apartado desarrollaremos distintos modelos de regresión con el objetivo de predecir el precio de una habitación. Comenzaremos con un modelo de Regresión Lineal Simple y, posteriormente, avanzaremos hacia un enfoque multivariante para mejorar la precisión de las predicciones.

2.2.1. Regresión Simple

Regresión Lineal Simple

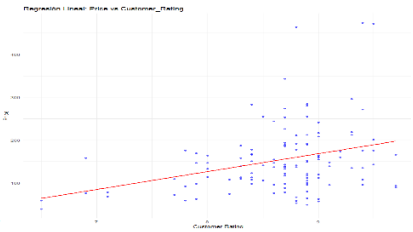
En primer lugar, para llevar a cabo la Regresión Lineal Simple, intentaremos predecir el precio por noche utilizando la variable *Customer Rating* como predictor. Nuestro objetivo es evaluar si el precio actual de los inmuebles en las plataformas es adecuado o si sería conveniente ajustarlo para optimizar la rentabilidad.

Tras llevar a cabo la Regresión Lineal Simple, obtenemos los siguientes resultados:

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -207.70      97.15  -2.138   0.0346 *
Customer_Rating   41.78      11.21   3.726   0.0003 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 75.47 on 118 degrees of freedom
Multiple R-squared:  0.1052,    Adjusted R-squared:  0.09766
F-statistic: 13.88 on 1 and 118 DF,  p-value: 0.0003005
```

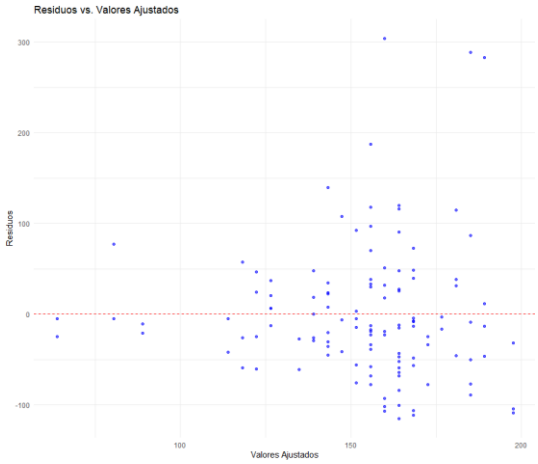
RMSE: 74.83 | MAE: 53.37 | R^2: 0.1052



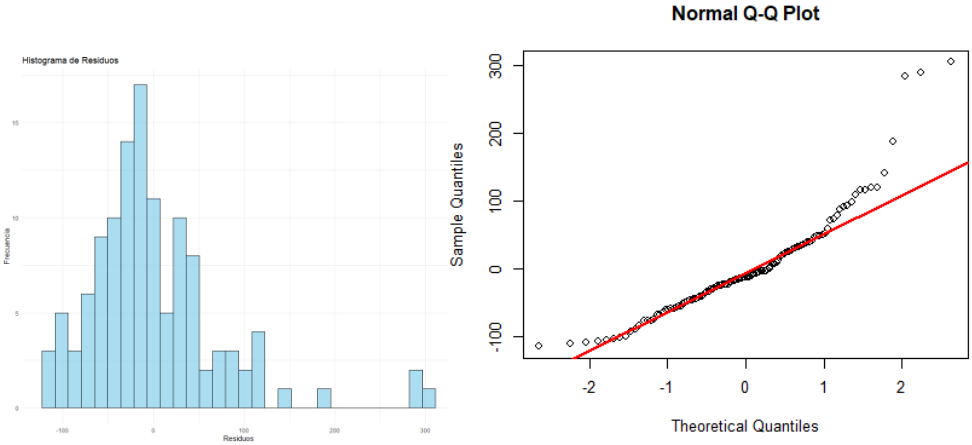
Podemos observar que la variable *Customer Rating* es estadísticamente significativa, lo que indica la existencia de una relación lineal con *Price*. Sin embargo, el valor de R^2 obtenido es relativamente bajo, lo que sugiere que este modelo explica solo una parte de la variabilidad en los precios. Al analizar los coeficientes, vemos que, según nuestro modelo, un aumento de 1 punto en *Customer Rating* incrementaría el precio de una habitación en 41,78$. Esto sugiere que implementar estrategias para incentivar a los clientes a dejar más reseñas positivas podría ser una medida efectiva para mejorar la percepción de los inmuebles y, potencialmente, aumentar sus precios en las plataformas de alquiler.

A continuación, procederemos al análisis de los residuos a través de diferentes gráficas.

En la primera gráfica, observamos que, en general, los residuos se distribuyen de manera equilibrada alrededor de la línea central, lo que sugiere que el modelo no presenta sesgos evidentes. Sin embargo, identificamos cuatro puntos que se alejan significativamente del resto, indicando que el modelo subestima sus valores reales. Éstos corresponden a los apartamentos más caros del conjunto de datos y pueden considerarse valores atípicos, ya que su precio es considerablemente superior al del resto de inmuebles, lo que dificulta que el modelo los prediga con precisión.



Por otro lado, al analizar el histograma y la gráfica Normal-QQ, observamos que los residuos siguen una distribución aproximadamente normal, lo cual es una señal positiva. Esto indica que el modelo cumple, en gran medida, con el supuesto de normalidad de los errores, lo que refuerza la validez de los resultados obtenidos.



Regresión polinómica simple

Ahora que hemos analizado los resultados del primer modelo de Regresión Lineal Simple, intentaremos mejorarlo mediante una Regresión Polinómica, utilizando únicamente la variable *Customer Rating*. Al llevar a cabo este nuevo modelo, obtenemos los siguientes resultados:

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                153.342      6.918  22.166  < 2e-16 ***
poly(Customer_Rating, 2)1  281.152     75.781   3.710 0.000319 ***
poly(Customer_Rating, 2)2  -10.402     75.781  -0.137 0.891059
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 75.78 on 117 degrees of freedom
Multiple R-squared:  0.1054,    Adjusted R-squared:  0.0901
F-statistic: 6.892 on 2 and 117 DF,  p-value: 0.001481
```

Observamos que la variable adicional introducida en la Regresión Polinómica no es estadísticamente significativa y que el valor de R^2 apenas muestra mejoras con respecto al modelo lineal. Dado que los resultados obtenidos son prácticamente equivalentes, optamos por

mantener el modelo de Regresión Lineal Simple, ya que ofrece una interpretación más sencilla y evita una complejidad innecesaria.

Conclusiones

En conclusión, los resultados de estos dos primeros modelos confirman que existe una fuerte relación lineal entre *Customer Rating* y *Price*. A pesar de considerar solo una variable, el modelo de Regresión Lineal Simple logra ofrecer una estimación razonable del precio de una habitación. Además, este análisis sugiere que incentivar a los clientes a dejar más reseñas positivas podría ser una estrategia rentable, ya que un aumento en la valoración de los inmuebles permitiría justificar un incremento en sus precios dentro de las plataformas de alquiler.

2.2.2. Regresión Múltiple

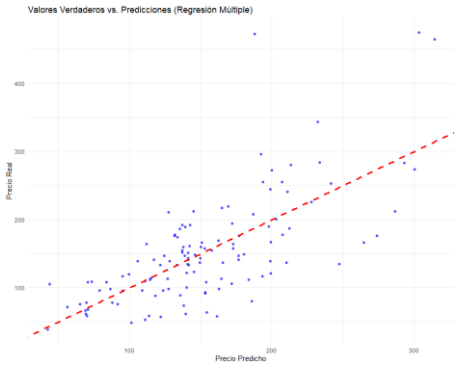
Regresión Lineal Múltiple

En esta ocasión, del mismo modo que en el análisis anterior, intentaremos predecir el precio por noche de un piso. Sin embargo, esta vez el enfoque estará dirigido a evaluar la rentabilidad de futuras propiedades que podrían ser adquiridas. Para ello, nos basaremos únicamente en datos objetivos de los inmuebles, excluyendo variables provenientes de las aplicaciones, como *Stars* y *Customer Rating*.

Comenzamos construyendo un modelo que incluya todas las variables disponibles. Los resultados obtenidos son los siguientes:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.271871  17.544468   0.072   0.9423
Distance     -0.004648   0.002189  -2.123   0.0359 *
Rooms        17.355750   6.710689   2.586   0.0110 *
Room_Squares  4.450929   0.573388   7.763 3.91e-12 ***
Beach        11.799530  12.242675   0.964   0.3372
Parking       9.320016  12.281060   0.759   0.4495
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 57.73 on 114 degrees of freedom
Multiple R-squared:  0.4942,    Adjusted R-squared:  0.472
F-statistic: 22.28 on 5 and 114 DF,  p-value: 1.597e-15
```



RMSE: 56.26 | MAE: 39.73 | R^2: 0.4942

Como habíamos anticipado al inicio de esta sección, observamos que las variables *Beach* y *Parking* no son estadísticamente significativas en el modelo. Y, el modelo obtenido presenta un R^2 de 0,4942, lo que supone una mejora considerable respecto al modelo de Regresión Univariante.

En lugar de proceder directamente con el análisis de los residuos, dado que *Beach* y *Parking* no aportan valor significativo a la predicción, intentaremos optimizar el modelo utilizando el método *step forward* para seleccionar las variables más relevantes. Los resultados obtenidos son los siguientes:

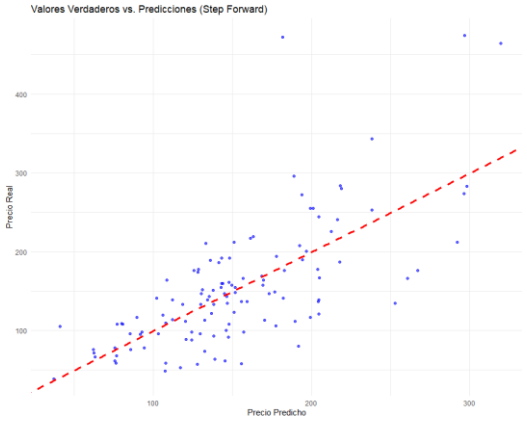
```
Step:  AIC=976.69
Price ~ Room_Squares + Rooms + Distance

           Df Sum of Sq    RSS    AIC
<none>                  384580 976.69
+ Beach     1    2779.8 381801 977.82
+ Parking   1    1603.5 382977 978.19
```

Como resultado, el modelo óptimo incluye únicamente las variables *Room Squares*, *Rooms* y *Distance*. Estas tres variables han demostrado ser las más relevantes para predecir el precio por noche de un piso. A partir de este modelo, obtenemos los siguientes resultados:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.771193  16.602320   0.468  0.64061
Room_Squares  4.421670   0.554472   7.975  1.2e-12 ***
Rooms        17.389703   6.349309   2.739  0.00714 **
Distance     -0.004637   0.002170  -2.137  0.03472 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 57.58 on 116 degrees of freedom
Multiple R-squared:  0.4879,    Adjusted R-squared:  0.4747
F-statistic: 36.85 on 3 and 116 DF,  p-value: < 2.2e-16
```



RMSE: 56.61 | MAE: 39.82 | R^2: 0.4879

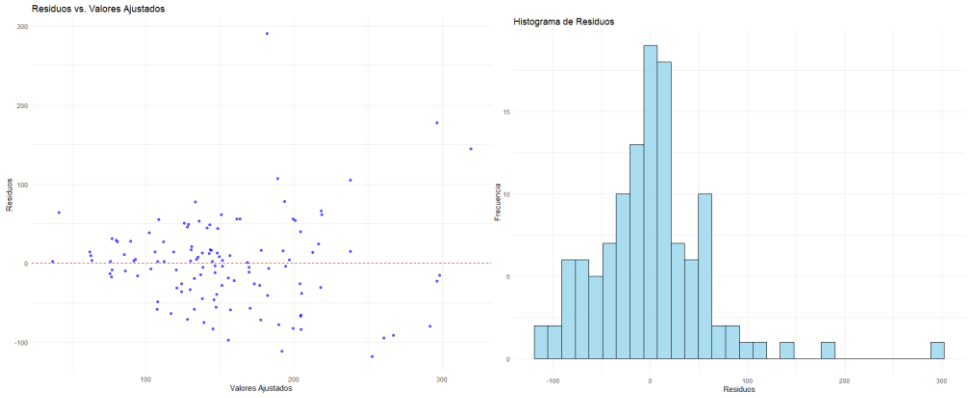
Al analizar los resultados obtenidos, observamos que son muy similares a los del modelo anterior sin *step*, pero con la ventaja de una menor complejidad, ya que se han eliminado dos variables no significativas.

En cuanto a la interpretación de los coeficientes, las tres variables seleccionadas resultan estadísticamente significativas. En particular, un incremento de un metro cuadrado en la superficie del piso (*Room Squares*) aumenta su precio en 4,43$. Del mismo modo, añadir una habitación adicional (*Rooms*) eleva el precio en 17,39$. Por otro lado, la distancia al centro (*Distance*) tiene un efecto negativo sobre el precio: por cada metro adicional de distancia, el precio disminuye en 0,0046$, lo que equivale a una reducción de 4,6$ por cada kilómetro que el inmueble se aleje del centro de la ciudad.

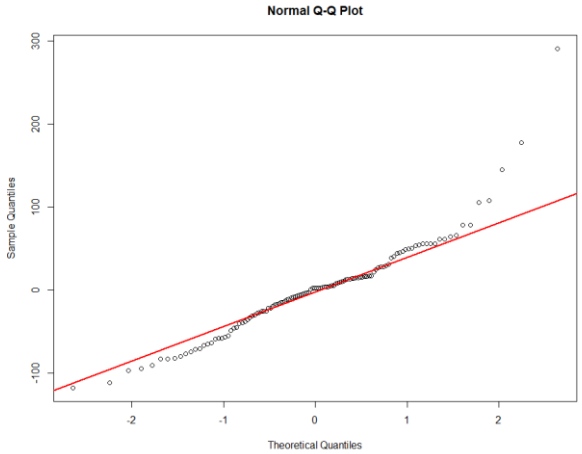
Procederemos ahora con el análisis de los residuos.

En el primer gráfico, del mismo modo que en el modelo univariante, observamos que, salvo tres apartamentos cuyo precio no se estima con precisión, el resto de los residuos se distribuyen de manera equilibrada alrededor de la línea central. Esto sugiere además que no existen relaciones no lineales significativas en los datos.

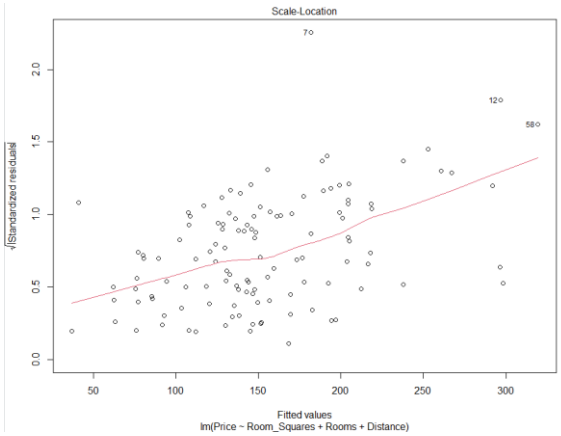
Además, en la segunda imagen, se aprecia que los residuos siguen una distribución aproximadamente normal en torno a 0, lo que es una señal positiva. Este comportamiento indica que el modelo está realizando predicciones adecuadas y no presenta un desbalance significativo.



Nuevamente, en el gráfico Normal-QQ, observamos que los residuos, salvo algunas excepciones, siguen la tendencia de la línea roja. Esto sugiere que los errores se distribuyen de manera aproximadamente normal, lo cual es una señal favorable para la validez del modelo.



A continuación, analizaremos el supuesto de homocedasticidad. En el gráfico correspondiente, observamos que la varianza de los residuos no se mantiene constante a lo largo de los valores de la variable dependiente, lo que puede representar un problema en la fiabilidad del modelo.



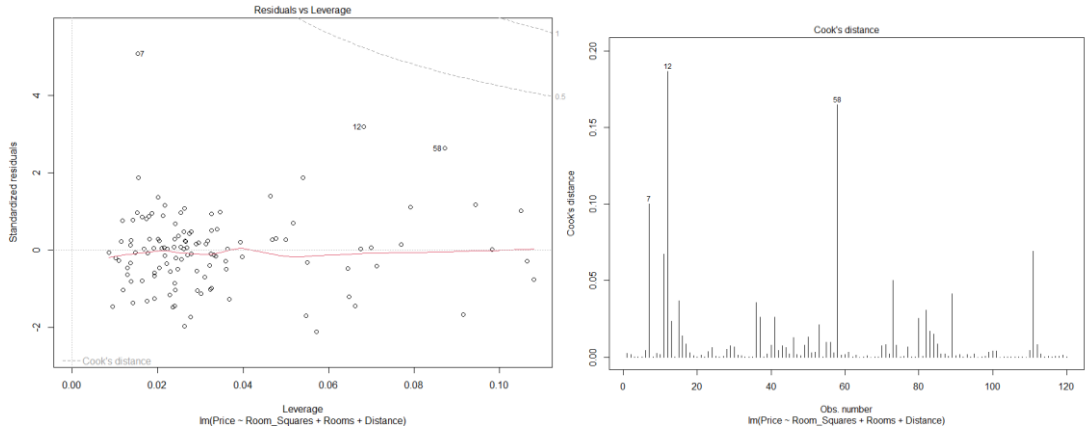
Para verificar esta hipótesis, realizamos la *Prueba de Breusch-Pagan (BP Test)*, cuya hipótesis nula (H_0) establece que los residuos presentan varianza constante, mientras que la hipótesis alternativa indica que la varianza de los residuos varía en función de las variables independientes. Los resultados obtenidos arrojan un p-valor de 0.008752, lo que nos lleva a rechazar la hipótesis nula (H_0) y confirmar la presencia de heterocedasticidad en el modelo, lo que indica que su validez podría estar comprometida.

```
        studentized Breusch-Pagan test

data:  modelo_step
BP = 11.633, df = 3, p-value = 0.008752
```

Por otro lado, al analizar los gráficos que muestran la influencia de las observaciones en el modelo, identificamos puntos con un impacto significativo en los resultados. En particular, el gráfico de las *Distancias de Cook* revela la presencia de dos observaciones que tienen una influencia excesivamente alta en el modelo, además de otras cuatro que, aunque en menor

medida, también ejercen un impacto considerable. Estos puntos podrían estar afectando la estabilidad del modelo y sería recomendable analizarlos en mayor profundidad para determinar si deben ser tratados como valores atípicos o si reflejan características relevantes de los datos.



Para analizar este fenómeno en mayor detalle, calculamos las *Distancias de Cook* y seleccionamos las cinco observaciones con mayor influencia en el modelo. Los resultados confirman que los tres datos más influyentes corresponden a los valores atípicos previamente identificados, caracterizados por precios significativamente altos. La presencia de estos puntos extremos podría estar desbalanceando el modelo, afectando su capacidad de generalización y reduciendo la precisión de las predicciones. Por lo tanto, sería recomendable considerar estrategias para mitigar su impacto, como transformaciones de variables, el uso de modelos más robustos o incluso la exclusión de estos datos en ciertos análisis.

ID	Price	Stars	Distance	Customer_Rating	Rooms	Room_Squares	Beach	Parking
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
64	474	4	1400	9.4	3	55	1	0
98	464	4	250	8.8	4	55	0	0
79	472	5	500	9.5	2	32	1	0
68	176	3	5000	7.8	2	56	1	0
25	135	4	300	9.4	4	40	0	0

Por último, analizaremos la posible presencia de multicolinealidad entre las variables del modelo. Para ello, calculamos los Valores de Inflación de la Varianza (VIF). Los resultados muestran que las tres variables seleccionadas presentan valores cercanos a 1, lo que indica que no existe una correlación excesiva entre ellas. Dado que estos valores están muy por debajo del umbral de 5, podemos concluir que la multicolinealidad no representa un problema significativo en nuestro modelo.

Room_Squares	Rooms	Distance
1.215709	1.215972	1.004382

Regresión Lineal Múltiple (Ridge-Lasso)

Con el objetivo de mejorar el modelo obtenido, intenté aplicar técnicas de regularización mediante Ridge y Lasso, ajustando distintos parámetros y combinaciones. Sin embargo, tras múltiples pruebas, los mejores resultados obtenidos fueron los siguientes:

```
Ridge -> RMSE: 56.63  |  MAE: 39.58  |  R^2: 0.49

Lasso -> RMSE: 56.61  |  MAE: 39.82  |  R^2: 0.49
```

Observamos que los resultados obtenidos con Ridge y Lasso son muy similares a los del modelo original, lo que indica que la regularización no aporta una mejora significativa. Dado que no se observa un beneficio claro en términos de precisión o estabilidad, no resulta conveniente añadir complejidad innecesaria al modelo mediante estas técnicas.

Conclusiones

En conclusión, hemos desarrollado un modelo de Regresión Lineal Múltiple que mejora la capacidad de predicción del precio de las viviendas y que puede ser una herramienta útil para evaluar la rentabilidad de futuras adquisiciones. No obstante, el modelo no es completamente preciso, ya que la presencia de valores atípicos dificulta su capacidad de generalización. Además, el problema de heterocedasticidad detectado indica que la varianza de los errores no es constante, lo que podría afectar la fiabilidad de las predicciones y sugiere la necesidad de explorar enfoques más robustos.

3. Regresión Logística

3.1. Descripción del dataset

A lo largo de esta sección vamos a centrarnos en elaborar un modelo de regresión logística cuyo objetivo será predecir si una futura reserva va a ser cancelada o no. Para ello, tomaremos ***Hotel_Cancellation.xlsx***, que consta de 119376 registros. En nuestro análisis, trabajaremos con las siguientes variables:

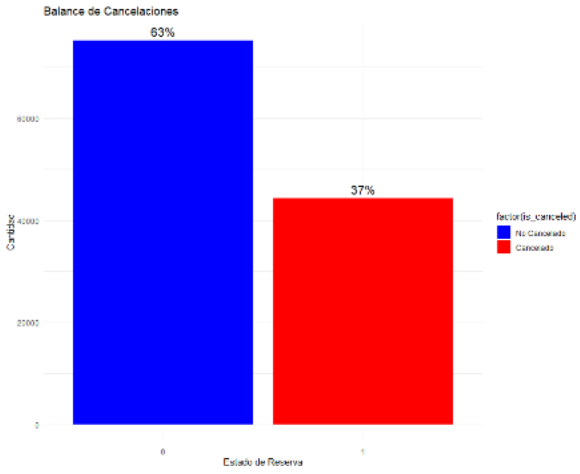
1. is_canceled (boolean): Indica si la reserva fue cancelada (Sí/No).
2. adults (int): Número de adultos incluidos en la reserva.
3. children (int): Número de niños incluidos en la reserva.
4. babies (int): Número de bebés incluidos en la reserva.
5. is_repeated_guest (boolean): Indica si el cliente ya ha reservado anteriormente (Sí/No).
6. previous_cancellations (int): Número de reservas anteriores canceladas por el cliente.
7. previous_bookings_not_canceled (int): Número de reservas anteriores que no fueron canceladas.
8. booking_changes (int): Número de modificaciones realizadas en la reserva.
9. adr (float): Tarifa diaria promedio (Average Daily Rate) en la moneda correspondiente.
10. lead_time (int): Días entre la fecha de reserva y la fecha de llegada.

Al realizar un análisis descriptivo del conjunto de datos mediante un summary, obtenemos los siguientes resultados:

is_canceled	adults	children	babies	is_repeated_guest	previous_cancellations	previous_bookings_not_canceled	booking_changes	adr	lead_time
Min. :0.0000	Min. : 0.000	Min. : 0.0000	Min. : 0.00000	Min. :0.00000	Min. : 0.00000	Min. : 0.0000	Min. : 0.0000	Min. : 0.00	Min. : 0
1st Qu.:0.0000	1st Qu.: 2.000	1st Qu.: 0.0000	1st Qu.: 0.00000	1st Qu.:0.00000	1st Qu.: 0.00000	1st Qu.: 0.0000	1st Qu.: 0.0000	1st Qu.: 69.29	1st Qu.: 18
Median :0.0000	Median : 2.000	Median : 0.0000	Median : 0.00000	Median :0.00000	Median : 0.00000	Median : 0.0000	Median : 0.0000	Median : 94.60	Median : 69
Mean :0.3704	Mean : 1.853	Mean : 0.1039	Mean : 0.00795	Mean :0.03191	Mean : 0.08713	Mean : 0.1371	Mean : 0.2211	Mean :101.80	Mean :104
3rd Qu.:1.0000	3rd Qu.: 2.000	3rd Qu.: 0.0000	3rd Qu.: 0.00000	3rd Qu.:0.00000	3rd Qu.: 0.00000	3rd Qu.: 0.0000	3rd Qu.: 0.0000	3rd Qu.:126.00	3rd Qu.:160
Max. :1.0000	Max. :10.000	Max. :10.0000	Max. :10.00000	Max. :1.00000	Max. :26.00000	Max. :72.0000	Max. :21.0000	Max. :510.00	Max. :737

Observamos que la mayoría de las reservas incluyen únicamente a dos adultos, sin la presencia de niños ni bebés. Además, en términos generales, la mayoría de los clientes son nuevos y no cuentan con reservas previas en el sistema.

La variable objetivo de nuestro estudio es *"is_canceled"*, que indica si una reserva ha sido cancelada o no. Al analizar la distribución de los datos, observamos que el conjunto está relativamente equilibrado, con un 63% de reservas no canceladas y un 37% de cancelaciones. Este balance en la distribución nos permite intuir que el modelo no debería presentar un sesgo excesivo hacia una de las dos categorías, favoreciendo así predicciones más precisas y representativas de la realidad.

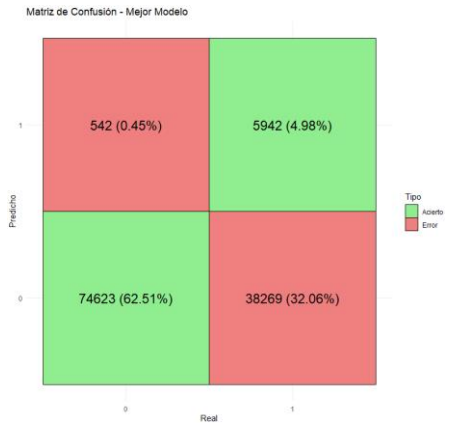


3.2. Modelos de Regresión Logística

En esta sección, nos enfocaremos en desarrollar un modelo de regresión logística para predecir la cancelación de una reserva. Comenzaremos con una regresión logística simple y, posteriormente, exploraremos modelos multivariantes para mejorar la precisión de nuestras predicciones.

Regresión Logística Simple

Para comenzar, desarrollaremos un modelo de regresión logística simple. Con este propósito, se ha implementado un bucle que permite probar la regresión logística con cada una de las variables de forma individual, seleccionando aquella que genere el modelo con la mayor Accuracy (% de predicciones acertadas). Como resultado de este proceso, se identificó que la variable *"Previous_Cancellations"* es la mejor predictora, obteniendo los siguientes resultados:



Métrica	Valor
Accuracy	0.67
Precision	0.92
Recall	0.13
F1 Score	0.23

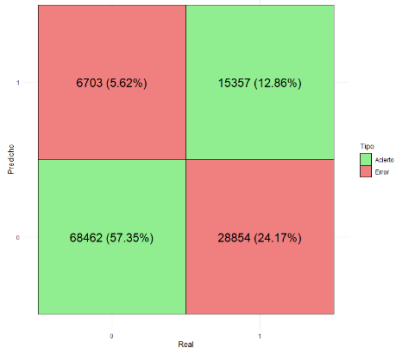
El modelo resultante es bastante conservador, ya que clasifica un número muy reducido de reservas como canceladas. Sin embargo, cuando predice una cancelación, lo hace con un alto nivel de precisión. En particular, al analizar la métrica de precisión, observamos que el modelo solo comete un error en aproximadamente el 8% de los casos en los que identifica una cancelación, lo que indica una alta fiabilidad en estas predicciones. Debido a que la variable del modelo es *"Previous_Cancellations",* todo hace indicar que el modelo simplemente ha predicho una cancelación cuando el valor de esta variable es positivo.

Regresión Logística Múltiple

Con el objetivo de mejorar el modelo anterior y reducir su carácter conservador en la predicción de cancelaciones, procederemos a realizar una regresión logística utilizando todas las variables disponibles. De este modo, buscamos obtener un modelo más equilibrado y preciso en la clasificación de reservas canceladas. Los resultados obtenidos son los siguientes:

```
Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -1.391e+00  2.874e-02 -48.416  < 2e-16 ***
adults                        -6.362e-02  1.457e-02  -4.368 1.25e-05 ***
children                      -2.424e-03  1.696e-02  -0.143    0.886
babies                        -4.777e-01  8.892e-02  -5.372 7.77e-08 ***
is_repeated_guest             -1.135e+00  7.955e-02 -14.263  < 2e-16 ***
previous_cancellations         2.986e+00  5.340e-02  55.923  < 2e-16 ***
previous_bookings_not_canceled -6.701e-01  2.695e-02 -24.868  < 2e-16 ***
booking_changes               -7.200e-01  1.557e-02 -46.246  < 2e-16 ***
adr                            4.470e-03  1.543e-04  28.966  < 2e-16 ***
lead_time                      5.248e-03  6.568e-05  79.900  < 2e-16 ***
```

Métrica	Valor
Accuracy	0.70
Precision	0.70
Recall	0.35
F1 Score	0.46



Podemos observar que todas las variables, excepto la correspondiente al número de niños, resultan estadísticamente significativas en el modelo. Además, al analizar la matriz de confusión, notamos que el modelo presenta un sesgo hacia la predicción de "no cancelación", clasificando un 82% de los casos como tales. Esto contrasta con la distribución real de los datos, donde el porcentaje de reservas no canceladas es del 63%, lo que sugiere un posible desequilibrio en las predicciones.

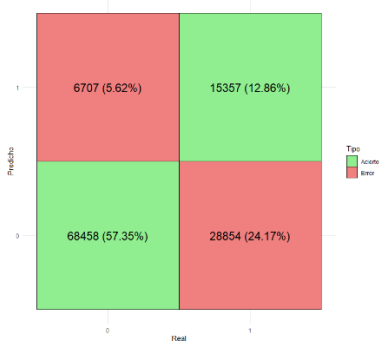
Por otro lado, aunque la evaluación de métricas del modelo no era el foco principal del análisis, al examinarlas encontramos que alcanza un 70% de precisión. En términos prácticos, esto implica que, de cada 10 cancelaciones predichas, 3 son incorrectas. Aunque este nivel de error no es excesivamente alto, sí podría representar un problema si consideramos las implicaciones económicas y operativas de una predicción errónea. Un modelo que sobreestima o subestima las cancelaciones podría generar costos adicionales, ya sea en términos de compensaciones a clientes o en una gestión ineficiente de la disponibilidad de habitaciones, afectando la rentabilidad y la reputación de la empresa.

Regresión Logística con Step

Para finalizar, intentaremos optimizar el modelo aplicando el método *step forward*, con el objetivo de encontrar una combinación de variables que mejore su desempeño y reduzca su complejidad.

```
Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -1.391e+00  2.868e-02 -48.512  < 2e-16 ***
lead_time                      5.248e-03  6.568e-05  79.901  < 2e-16 ***
previous_cancellations         2.987e+00  5.340e-02  55.924  < 2e-16 ***
previous_bookings_not_canceled -6.701e-01  2.695e-02 -24.868  < 2e-16 ***
booking_changes                -7.201e-01  1.556e-02 -46.276  < 2e-16 ***
adr                            4.462e-03  1.440e-04  30.980  < 2e-16 ***
is_repeated_guest              -1.135e+00  7.955e-02 -14.263  < 2e-16 ***
babies                         -4.778e-01  8.892e-02  -5.374 7.71e-08 ***
adults                         -6.346e-02  1.452e-02  -4.371 1.24e-05 ***
```

Métrica	Valor
Accuracy	0.70
Precision	0.70
Recall	0.35
F1 Score	0.46



Los resultados obtenidos son muy similares a los del modelo anterior. Como mencionamos previamente, la variable correspondiente al número de niños no era estadísticamente significativa, y tras aplicar el método *step forward*, el modelo final ha excluido únicamente esta variable. Sin embargo, la eliminación de dicha variable no ha generado una mejora notable en el desempeño del modelo, ya que los resultados obtenidos siguen siendo prácticamente idénticos a los del modelo previo.

Conclusiones

En última instancia, la elección del modelo dependerá del objetivo estratégico de la empresa. Si se busca un enfoque más conservador, podríamos considerar el primer modelo, que básicamente predice una cancelación únicamente cuando el cliente tiene antecedentes de cancelaciones previas. No obstante, esta aproximación resulta de utilidad limitada, ya que no toma en cuenta otros factores que podrían influir en la cancelación de una reserva.

Por otro lado, el modelo final ofrece un mejor equilibrio en las predicciones. Y, aunque no es perfecto, puede ser una herramienta valiosa si la empresa está dispuesta a asumir un enfoque más arriesgado. Es fundamental, sin embargo, evaluar cuidadosamente los costes asociados a posibles errores en la predicción, ya que una mala clasificación de cancelaciones podría derivar en pérdidas económicas o afectar la gestión operativa de las reservas.

4. Conclusiones generales

Este estudio ha permitido analizar el impacto de distintas variables en la rentabilidad y gestión de propiedades de alquiler mediante modelos de regresión. En el caso de la regresión lineal, se ha demostrado que factores como la puntuación de los clientes (*Customer Rating*) influyen significativamente en el precio de las habitaciones. Sin embargo, pese a que la relación entre el precio y la calificación de los clientes es clara, el modelo obtuvo un R^2 bajo, lo que sugiere que hay otros factores que afectan el precio y que no han sido considerados. Además, la regresión polinómica no aportó mejoras significativas, por lo que se optó por la simplicidad del modelo lineal.

En cuanto a la regresión múltiple, se confirmó que variables como la cercanía a la playa y la disponibilidad de parking no eran determinantes en la fijación del precio, mientras que el tamaño

del piso y la cantidad de habitaciones tienen un efecto positivo. Sin embargo, la presencia de valores atípicos y la heterocedasticidad afectaron la precisión del modelo, lo que indica la necesidad de técnicas más avanzadas para mejorar las predicciones.

En el caso de la regresión logística, el objetivo era predecir la cancelación de reservas, lo que permitiría optimizar la gestión de disponibilidad y minimizar pérdidas. Se encontró que la variable *Previous Cancellations* era la mejor predictora de cancelaciones, lo que sugiere que los clientes con antecedentes de cancelaciones previas tienen mayor probabilidad de repetir este comportamiento. No obstante, el modelo inicial resultó ser demasiado conservador, identificando pocas cancelaciones y subestimando la frecuencia real de estos eventos. Al incluir más variables en el modelo, se obtuvo una mejora, aunque seguía existiendo un sesgo hacia la clasificación de reservas como “no canceladas”. A pesar de estos desafíos, el modelo final proporciona una base para anticiparse a cancelaciones y tomar decisiones estratégicas, como la implementación de políticas de overbooking controlado o incentivos para clientes con mayor riesgo de cancelar.

5. Siguientes pasos

Para mejorar estos resultados, en futuros trabajos sería recomendable explorar modelos más avanzados, como árboles de decisión o redes neuronales, que podrían capturar relaciones más complejas entre las variables. Además, el tratamiento de los valores atípicos y la heterocedasticidad podría mejorar la precisión del modelo de regresión múltiple, mientras que técnicas de balanceo de datos podrían corregir el sesgo en la regresión logística.

También sería interesante incorporar nuevas variables, como factores estacionales o eventos locales, que podrían influir tanto en el precio de las habitaciones como en la probabilidad de cancelación.

Finalmente, la implementación de estos modelos en un entorno real permitiría evaluar su impacto a través de pruebas A/B, ajustando las estrategias comerciales en función de las predicciones obtenidas.

6. Anexo (Código)

Origen de los datasets:

1 - https://www.kaggle.com/datasets/aladzuzamar/hotels-accommodation-prices-dataset (Creé las variables de parking y playa. Y eliminé las que no eran necesarias)

2 - https://www.kaggle.com/datasets/arezaei81/hotel-bookingcvs (Eliminé columnas innecesarias)

```
# Cargar librerías necesarias

library(readxl)

library(dplyr)

library(car)

library(ggplot2)

library(GGally)

library(patchwork)

library(Metrics)

library(glmnet)

library(caret)

library(lmtest)

library(knitr)


# ------------------- Carga y Preparación de Datos ------------------- #


# Cargar el dataset

data <- read_excel("Hotels.xlsx")


# Eliminar los nulos

data <- na.omit(data)


# Ver la estructura (tipo de datos en cada columna)

str(data)


# Resumen estadístico

summary(data)


# Número total de filas y columnas

dim(data)  # [filas, columnas]
```

```
# Nombre de las columnas
colnames(data)

# ------------------- Análisis Exploratorio de Datos (EDA) ------------------- #

# ---- HISTOGRAMAS ---- #
# Seleccionar variables numéricas excluyendo "ID"
numeric_vars <- setdiff(names(data)[sapply(data, is.numeric)], "ID")

# Crear lista de histogramas
plots <- lapply(numeric_vars, function(var) {
  ggplot(data, aes_string(x = var)) +
    geom_histogram(bins = 30, fill = "skyblue", color = "black", alpha = 0.7) +
    labs(title = paste("Histograma de", var), x = var, y = "Frecuencia") +
    theme_minimal()
})

# Organizar en 2 filas y 4 columnas
final_plot <- wrap_plots(plots) + plot_layout(ncol = 4, nrow = 2)
print(final_plot)

# ---- Gráficos de Dispersión ---- #
# Seleccionar variables numéricas excluyendo "ID" y "Price" (ya que Price es el eje Y)
numeric_vars <- setdiff(names(data)[sapply(data, is.numeric)], c("ID", "Price"))

# Crear lista de gráficos de dispersión
plots <- lapply(numeric_vars, function(var) {
  ggplot(data, aes_string(x = var, y = "Price")) +
    geom_point(color = "blue", alpha = 0.6) +
    geom_smooth(method = "lm", color = "red", se = FALSE) +  # Línea de tendencia
    labs(title = paste("Price vs", var), x = var, y = "Price") +
    theme_minimal()
})

# Organizar en 2 filas y 4 columnas
```

```
final_plot <- wrap_plots(plots) + plot_layout(ncol = 4, nrow = 2)
print(final_plot)

# --- Matriz de Correlaciones --- #
# Seleccionar solo variables numéricas (excluyendo "ID")
numeric_vars <- setdiff(names(data)[sapply(data, is.numeric)], "ID")

# Calcular la matriz de correlación
cor_matrix <- cor(data[, numeric_vars], use = "complete.obs")

# Mostrar la matriz en consola
print(cor_matrix)

# Graficar la matriz de correlación
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45, addCoef.col = "black")

# ------------------- Modelado y Evaluación de Modelos ------------------- #

# ------ Regresión Lineal Simple ------- #
# Ajustar el modelo de regresión lineal
modelo <- lm(Price ~ Customer_Rating, data = data)

# Mostrar resumen del modelo
summary(modelo)

# Predicciones del modelo
predicciones <- predict(modelo, newdata = data)

# Calcular métricas de evaluación
rmse_value <- rmse(data$Price, predicciones)  # Raíz del error cuadrático medio
mae_value <- mae(data$Price, predicciones)    # Error absoluto medio
r2_value <- summary(modelo)$r.squared         # R-cuadrado
```

```
# Mostrar métricas
cat(sprintf("RMSE: %.2f | MAE: %.2f | R²: %.4f\n", rmse_value, mae_value, r2_value))

# Gráfico de regresión lineal
ggplot(data, aes(x = Customer_Rating, y = Price)) +
  geom_point(color = "blue", alpha = 0.6) +  # Puntos de dispersión
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Línea de regresión
  labs(title = "Regresión Lineal: Price vs Customer_Rating",
       x = "Customer Rating", y = "Price") +
  theme_minimal()

# Gráfico de residuos vs valores ajustados
ggplot(data, aes(x = predicciones, y = resid(modelo))) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuos vs. Valores Ajustados",
       x = "Valores Ajustados", y = "Residuos") +
  theme_minimal()

# Histograma de residuos
ggplot(data, aes(x = resid(modelo))) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Histograma de Residuos", x = "Residuos", y = "Frecuencia") +
  theme_minimal()

# QQ-Plot de residuos
qqnorm(resid(modelo))
qqline(resid(modelo), col = "red", lwd = 2)

# ---- Lineal Polinómico ---- #
# Ajustar modelo de regresión polinómica (grado 2)
modelo_poly2 <- lm(Price ~ poly(Customer_Rating, 2), data = data)

# Mostrar resumen del modelo
summary(modelo_poly2)
```

```
# ------------------- Regresión Múltiple ------------------- #
# Ajustar el modelo de regresión múltiple
modelo_multi <- lm(Price ~ Distance + Rooms + Room_Squares + Beach + Parking, data = data)

# Mostrar resumen del modelo
summary(modelo_multi)

# Predicciones del modelo
predicciones_multi <- predict(modelo_multi, newdata = data)

# Calcular métricas de error
rmse_value <- rmse(data$Price, predicciones_multi)  # Error cuadrático medio
mae_value <- mae(data$Price, predicciones_multi)    # Error absoluto medio
r2_value <- summary(modelo_multi)$r.squared         # R²

# Mostrar métricas en una sola fila
cat(sprintf("RMSE: %.2f | MAE: %.2f | R²: %.4f\n", rmse_value, mae_value, r2_value))

# Gráfico de valores verdaderos vs predicciones
ggplot(data, aes(x = predicciones_multi, y = Price)) +
  geom_point(color = "blue", alpha = 0.6) +  # Puntos de dispersión
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed", size = 1.2) +  # Línea ideal
  labs(title = "Valores Verdaderos vs. Predicciones (Regresión Múltiple)",
       x = "Precio Predicho", y = "Precio Real") +
  theme_minimal()

# Gráfico de residuos vs valores ajustados
ggplot(data, aes(x = predicciones_multi, y = resid(modelo_multi))) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuos vs. Valores Ajustados",
       x = "Valores Ajustados", y = "Residuos") +
  theme_minimal()
```

```
# Histograma de residuos
ggplot(data, aes(x = resid(modelo_multi))) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Histograma de Residuos", x = "Residuos", y = "Frecuencia") +
  theme_minimal()

# QQ-Plot de residuos
qqnorm(resid(modelo_multi))
qqline(resid(modelo_multi), col = "red", lwd = 2)

# ------------------- Regularización Ridge-Lasso ------------------- #
# Definir matriz de predictores (X) y variable objetivo (y)
X <- as.matrix(data[, c("Distance", "Rooms", "Room_Squares")])
y <- data$Price

# Definir una secuencia de valores de lambda (10^ de -4 a 10^4)
lambda_seq <- 10^seq(-4, 4, length = 100)

# Ajustar modelo Ridge
modelo_ridge <- glmnet(X, y, alpha = 0, lambda = lambda_seq)

# Validación cruzada para encontrar el mejor lambda
cv_ridge <- cv.glmnet(X, y, alpha = 0, lambda = lambda_seq)

# Obtener el mejor lambda
best_lambda_ridge <- cv_ridge$lambda.min

# Ajustar modelo final con el mejor lambda
modelo_ridge_final <- glmnet(X, y, alpha = 0, lambda = best_lambda_ridge)

# Mostrar coeficientes del modelo final
coef(modelo_ridge_final)

# Ajustar modelo Lasso
modelo_lasso <- glmnet(X, y, alpha = 1, lambda = lambda_seq)
```

```
# Validación cruzada para encontrar el mejor lambda
cv_lasso <- cv.glmnet(X, y, alpha = 1, lambda = lambda_seq)

# Obtener el mejor lambda
best_lambda_lasso <- cv_lasso$lambda.min

# Ajustar modelo final con el mejor lambda
modelo_lasso_final <- glmnet(X, y, alpha = 1, lambda = best_lambda_lasso)

# Mostrar coeficientes del modelo final
coef(modelo_lasso_final)

# Predicciones con los mejores modelos
predicciones_ridge <- predict(modelo_ridge_final, s = best_lambda_ridge, newx = X)
predicciones_lasso <- predict(modelo_lasso_final, s = best_lambda_lasso, newx = X)

# Calcular métricas de error
rmse_ridge <- RMSE(predicciones_ridge, y)
rmse_lasso <- RMSE(predicciones_lasso, y)

mae_ridge <- MAE(predicciones_ridge, y)
mae_lasso <- MAE(predicciones_lasso, y)

# Función para calcular R^2
r2 <- function(y_true, y_pred) {
  ss_res <- sum((y_true - y_pred)^2)
  ss_tot <- sum((y_true - mean(y_true))^2)
  return(1 - ss_res / ss_tot)
}

r2_ridge <- r2(y, predicciones_ridge)
r2_lasso <- r2(y, predicciones_lasso)

cat(sprintf("Ridge -> RMSE: %.2f | MAE: %.2f | R^2: %.2f\n", rmse_ridge, mae_ridge, r2_ridge))
```

```
cat(sprintf("Lasso -> RMSE: %.2f | MAE: %.2f | R^2: %.2f\n", rmse_lasso, mae_lasso, r2_lasso))


# ------------------- Regresión Logística ------------------- #

# Cargar dataset de cancelaciones
data_cancellation <- read.csv("Hotel_Cancellation.csv", stringsAsFactors = FALSE)

# Seleccionar solo variables numéricas
numeric_vars <- select(data_cancellation, where(is.numeric))

# Calcular la correlación de is_canceled con las demás variables
correlations <- cor(numeric_vars, use = "pairwise.complete.obs")["is_canceled", ]

# Mostrar las correlaciones en orden descendente
correlations_sorted <- sort(correlations, decreasing = TRUE)
print(correlations_sorted)

# Preparar datos para regresión logística
data_logistic <- select(data_cancellation, is_canceled, adults, children, babies,
                        is_repeated_guest, previous_cancellations,
                        previous_bookings_not_canceled, booking_changes,
                        adr,  lead_time)

# Ver estructura del dataset
str(data_logistic)

# Resumen estadístico
summary(data_logistic)

# Balance de cancelaciones
data_balance <- data_logistic %>%
 group_by(is_canceled) %>%
 summarise(count = n()) %>%
 mutate(percentage = round(count / sum(count) * 100, 1))  # Calcular %

# Gráfico de barras con porcentajes
```

```
ggplot(data_balance, aes(x = factor(is_canceled), y = count, fill = factor(is_canceled))) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(percentage, "%")), vjust = -0.5, size = 5) +  # Mostrar %
  scale_fill_manual(values = c("blue", "red"), labels = c("No Cancelado", "Cancelado")) +
  labs(title = "Balance de Cancelaciones", x = "Estado de Reserva", y = "Cantidad") +
  theme_minimal()

# Histogramas de variables
vars_to_plot <- c("adults", "children", "babies", "is_repeated_guest",
                  "previous_cancellations", "previous_bookings_not_canceled",
                  "booking_changes", "adr", "lead_time")

# Crear una lista de histogramas
hist_plots <- lapply(vars_to_plot, function(var) {
  ggplot(data_logistic, aes(x = .data[[var]], fill = factor(is_canceled))) +
    geom_histogram(alpha = 0.6, bins = 30, position = "identity") +
    scale_fill_manual(values = c("blue", "red"), labels = c("No Cancelado", "Cancelado")) +
    labs(title = paste("Distribución de", var), x = var, y = "Frecuencia") +
    theme_minimal()
})

# Unir los gráficos en un solo panel
wrap_plots(hist_plots, ncol = 2)  # 2 columnas para organizar mejor

# Ajustar la regresión logística
logistic_model <- glm(is_canceled ~ ., data = data_logistic, family = binomial)

# Resumen del modelo
summary(logistic_model)

# Predicciones en probabilidades
predicted_probs <- predict(logistic_model, type = "response")

# Convertir probabilidades a clases (umbral 0.5)
```

```
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)

# Crear matriz de confusión
conf_matrix <- table(Predicho = predicted_classes, Real = data_logistic$is_canceled)

# Calcular porcentajes de acierto y error
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix) * 100
error_rate <- 100 - accuracy

# Convertir la matriz de confusión a formato data frame para ggplot2
conf_matrix_df <- as.data.frame(as.table(conf_matrix))
colnames(conf_matrix_df) <- c("Predicho", "Real", "Frecuencia")

# Agregar porcentaje de cada celda
conf_matrix_df$Porcentaje <- round(conf_matrix_df$Frecuencia / sum(conf_matrix_df$Frecuencia) * 100, 2)

# Gráfico de matriz de confusión
ggplot(conf_matrix_df, aes(x = Real, y = Predicho, fill = Tipo)) +
 geom_tile(color = "black") +
 geom_text(aes(label = paste0(Frecuencia, " (", Porcentaje, "%)")), size = 6) +
 scale_fill_manual(values = c("Acierto" = "lightgreen", "Error" = "lightcoral")) +
 theme_minimal() +
 labs(title = "Matriz de Confusión", x = "Real", y = "Predicho")

# Calcular métricas con confusionMatrix
conf_matrix <- confusionMatrix(predicted_classes, data_logistic$is_canceled, positive = "1")

accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Pos Pred Value"]  # Precisión
recall <- conf_matrix$byClass["Sensitivity"]  # Recall
f1_score <- 2 * (precision * recall) / (precision + recall)  # F1 Score

# Crear tabla con los resultados
metricas_df <- data.frame(
 Métrica = c("Accuracy", "Precision", "Recall", "F1 Score"),
 Valor = c(accuracy, precision, recall, f1_score)
```

```
)

# Mostrar la tabla formateada
kable(metricas_df, digits = 2, col.names = c("Métrica", "Valor"), align = "c")

# ------------------- Regresión Logística con Stepwise Forward ------------------- #

# Modelo base vacío (solo intercepto)
base_model <- glm(is_canceled ~ 1, data = data_logistic, family = binomial)

# Modelo completo con todas las variables
full_model <- glm(is_canceled ~ ., data = data_logistic, family = binomial)

# Stepwise Forward: Selección de variables basada en AIC
stepwise_forward_model <- step(base_model,
                               scope = list(lower = base_model, upper = full_model),
                               direction = "forward",
                               trace = TRUE)

# Resumen del modelo seleccionado
summary(stepwise_forward_model)

# Predicciones en probabilidades
pred_probs <- predict(stepwise_forward_model, type = "response")

# Convertir probabilidades a clases (umbral 0.5)
pred_classes <- ifelse(pred_probs > 0.5, 1, 0)

# Convertir variables a factor para la matriz de confusión
data_logistic$is_canceled <- as.factor(data_logistic$is_canceled)
pred_classes <- as.factor(pred_classes)

# Crear matriz de confusión
conf_matrix <- confusionMatrix(pred_classes, data_logistic$is_canceled, positive = "1")
```

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Pos Pred Value"]  # Precisión
recall <- conf_matrix$byClass["Sensitivity"]  # Recall
f1_score <- 2 * (precision * recall) / (precision + recall)  # F1 Score

# Mostrar métricas en consola
cat(sprintf("Accuracy: %.2f\n", accuracy))
cat(sprintf("Precision: %.2f\n", precision))
cat(sprintf("Recall: %.2f\n", recall))
cat(sprintf("F1 Score: %.2f\n", f1_score))

# Convertir la matriz de confusión a formato data frame para ggplot2
conf_matrix_df <- as.data.frame(conf_matrix$table)
colnames(conf_matrix_df) <- c("Predicho", "Real", "Frecuencia")

# Agregar porcentaje de cada celda
conf_matrix_df$Porcentaje <- round(conf_matrix_df$Frecuencia / sum(conf_matrix_df$Frecuencia) * 100, 2)

# Agregar una columna para identificar aciertos y errores
conf_matrix_df$Tipo <- ifelse(conf_matrix_df$Real == conf_matrix_df$Predicho, "Acierto", "Error")

# Gráfico de matriz de confusión con colores personalizados
ggplot(conf_matrix_df, aes(x = Real, y = Predicho, fill = Tipo)) +
  geom_tile(color = "black") +
  geom_text(aes(label = paste0(Frecuencia, " (", Porcentaje, "%)")), size = 6) +
  scale_fill_manual(values = c("Acierto" = "lightgreen", "Error" = "lightcoral")) +
  theme_minimal() +
  labs(title = "Matriz de Confusión", x = "Real", y = "Predicho")

# Crear tabla con los resultados de métricas
metricas_df <- data.frame(
  Métrica = c("Accuracy", "Precision", "Recall", "F1 Score"),
  Valor = c(accuracy, precision, recall, f1_score)
)
```

```
# Mostrar la tabla formateada
kable(metricas_df, digits = 2, col.names = c("Métrica", "Valor"), align = "c")

# ---------- Regresión Logística Simple ---------- #

# Convertir la variable objetivo a factor
data_logistic$is_canceled <- as.factor(data_logistic$is_canceled)

# Lista para almacenar los resultados
resultados <- data.frame(Variable = character(), Accuracy = numeric(), stringsAsFactors = FALSE)

# Seleccionar todas las variables predictoras
variables <- setdiff(names(data_logistic), "is_canceled")

# Iterar sobre todas las variables para ajustar regresiones logísticas simples
for (var in variables) {
  formula <- as.formula(paste("is_canceled ~", var))
  modelo <- glm(formula, data = data_logistic, family = binomial)

  pred_probs <- predict(modelo, type = "response")
  pred_classes <- ifelse(pred_probs > 0.5, 1, 0)

  pred_classes <- as.factor(pred_classes)

  conf_matrix <- confusionMatrix(pred_classes, data_logistic$is_canceled, positive = "1")
  accuracy <- conf_matrix$overall["Accuracy"]

  resultados <- rbind(resultados, data.frame(Variable = var, Accuracy = accuracy))
}

# Seleccionar la mejor variable según la precisión
mejor_variable <- resultados[which.max(resultados$Accuracy), "Variable"]
mejor_precision <- max(resultados$Accuracy)
```

```
cat(sprintf("La mejor variable es '%s' con una precisión de %.2f%%\n", mejor_variable, mejor_precision * 100))

# Ajustar el modelo final con la mejor variable
mejor_formula <- as.formula(paste("is_canceled ~", mejor_variable))
mejor_modelo <- glm(mejor_formula, data = data_logistic, family = binomial)

# Predicciones finales
pred_probs_final <- predict(mejor_modelo, type = "response")
pred_classes_final <- ifelse(pred_probs_final > 0.5, 1, 0)

# Matriz de confusión final
conf_matrix_final <- confusionMatrix(as.factor(pred_classes_final), data_logistic$is_canceled, positive = "1")

# Extraer métricas
accuracy_final <- conf_matrix_final$overall["Accuracy"]
precision_final <- conf_matrix_final$byClass["Pos Pred Value"]
recall_final <- conf_matrix_final$byClass["Sensitivity"]
f1_score_final <- 2 * (precision_final * recall_final) / (precision_final + recall_final)

# Convertir matriz de confusión a data frame para ggplot2
conf_matrix_df <- as.data.frame(conf_matrix_final$table)
colnames(conf_matrix_df) <- c("Predicho", "Real", "Frecuencia")

# Agregar porcentaje de cada celda
conf_matrix_df$Porcentaje <- round(conf_matrix_df$Frecuencia / sum(conf_matrix_df$Frecuencia) * 100, 2)

# Agregar columna para diferenciar aciertos y errores
conf_matrix_df$Tipo <- ifelse(conf_matrix_df$Real == conf_matrix_df$Predicho, "Acierto", "Error")

# Gráfico de matriz de confusión con colores
ggplot(conf_matrix_df, aes(x = Real, y = Predicho, fill = Tipo)) +
  geom_tile(color = "black") +
  geom_text(aes(label = paste0(Frecuencia, " (", Porcentaje, "%)")), size = 6) +
  scale_fill_manual(values = c("Acierto" = "lightgreen", "Error" = "lightcoral")) +
  theme_minimal() +
```

```
  labs(title = "Matriz de Confusión - Mejor Modelo", x = "Real", y = "Predicho")

# Tabla con métricas
metricas_df <- data.frame(
  Métrica = c("Accuracy", "Precision", "Recall", "F1 Score"),
  Valor = c(accuracy_final, precision_final, recall_final, f1_score_final)
)

# Mostrar la tabla formateada
kable(metricas_df, digits = 2, col.names = c("Métrica", "Valor"), align = "c")
```