



INFERENCIA Y MODELOS ESTADÍSTICOS

Jacqueline Köhler C. y José Luis Jara V.

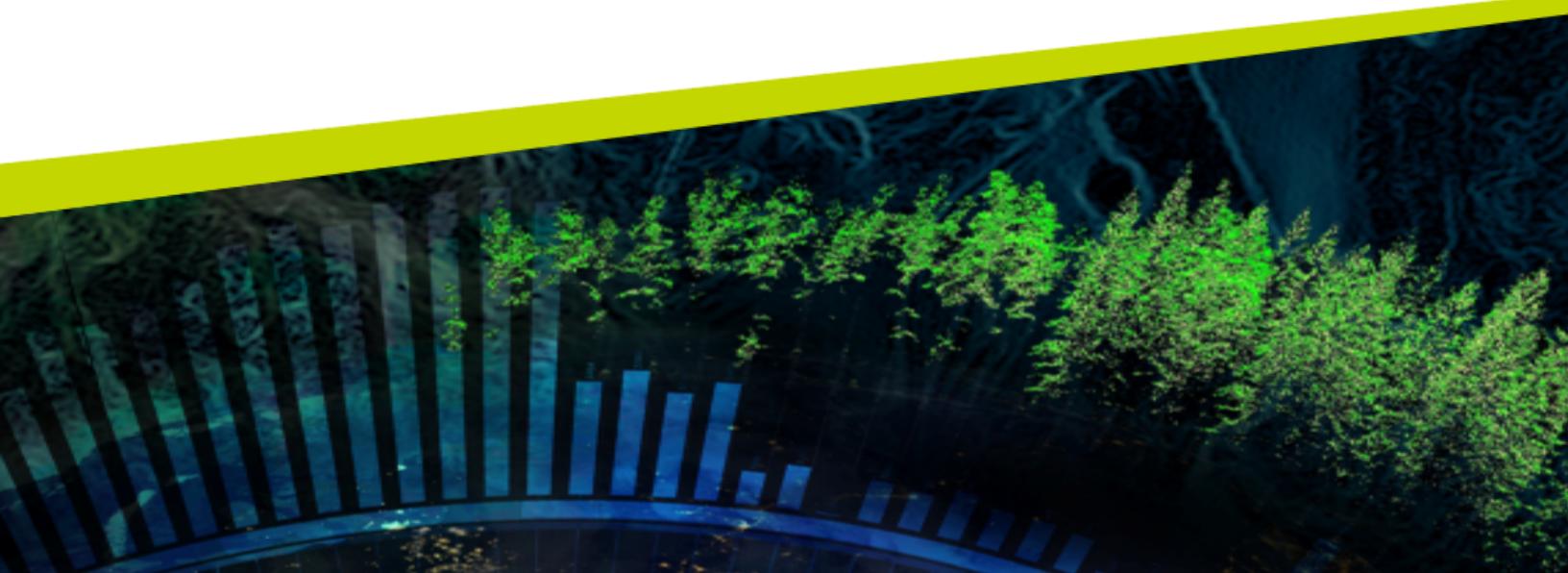


Tabla de contenido

1 Introducción	1
1.1 Inferencia y modelos estadísticos	1
1.2 Variables, parámetros y estadísticos	3
1.3 Conociendo R	5
1.3.1 Importación de datos	5
1.3.2 Importación de paquetes	7
1.3.3 Construcción de una matriz de datos	7
1.3.4 Modificación de una matriz de datos	8
1.3.5 Fórmulas	12
1.4 Ejercicios propuestos	13
2 Exploración de datos	15
2.1 Estadísticas descriptivas	15
2.1.1 Estadísticas descriptivas para datos numéricos	16
2.1.2 Estadísticas descriptivas para datos categóricos	19
2.1.3 Trabajando con datos agrupados	23
2.2 Representación gráfica de datos	23
2.2.1 Una variable numérica	24
2.2.2 Una variable categórica	26
2.2.3 Dos variables numéricas	28
2.2.4 Dos variables categóricas	30
2.2.5 Una variable numérica y otra categórica	32
2.3 Ejercicios propuestos	33
3 Variables aleatorias y distribuciones de probabilidad	37
3.1 Variables aleatorias	37
3.2 Distribuciones continuas	41
3.2.1 Distribución normal	42
3.2.2 Distribución Z	44
3.2.3 Distribución chi-cuadrado	45
3.2.4 Distribución t de Student	45
3.2.5 Distribución F	47
3.3 Distribuciones discretas	48
3.3.1 Distribución de Bernoulli	48
3.3.2 Distribución geométrica	48
3.3.3 Distribución binomial	49
3.3.4 Distribución binomial negativa	51
3.3.5 Distribución de Poisson	52
3.4 Ejercicios propuestos	53
4 Fundamentos para la inferencia	55
4.1 Estimadores puntuales	55
4.2 Modelos estadísticos	57
4.3 Error estándar	58

4.4	Intervalos de confianza	59
4.5	Pruebas de hipótesis	60
4.5.1	Prueba formal de hipótesis con valores p	62
4.5.2	El efecto del nivel de significación	66
4.6	Inferencia para otros estimadores	66
4.6.1	Estimadores puntuales con distribución cercana a la normal	67
4.6.2	Estimadores con otras distribuciones	67
4.7	Ejercicios propuestos	68
5	Inferencia con medias muestrales	71
5.1	Prueba Z	71
5.2	Prueba t de Student	74
5.2.1	Prueba t para una muestra	75
5.2.2	Prueba t para dos muestras pareadas	78
5.2.3	Prueba t para dos muestras independientes	80
5.3	Ejercicios propuestos	83
6	Poder estadístico	85
6.1	Poder, nivel de significación y tamaño de la muestra	85
6.2	Tamaño del efecto	88
6.3	Poder, tamaño del efecto y tamaño de la muestra	90
6.4	Cálculo teórico del poder	91
6.5	Cálculo del poder en R	94
6.6	Ejercicios propuestos	96
7	Inferencia con proporciones muestrales	99
7.1	Método de Wald	99
7.1.1	Método de Wald para una proporción	99
7.1.2	Método de Wald para dos proporciones	101
7.2	Método de Wilson	105
7.3	Poder y pruebas de proporciones	106
7.4	Ejercicios propuestos	107
8	Inferencia no paramétrica con proporciones	109
8.1	Prueba chi-cuadrado de Pearson	109
8.1.1	Prueba chi-cuadrado de homogeneidad	110
8.1.2	Prueba chi-cuadrado de bondad de ajuste	112
8.1.3	Prueba chi-cuadrado de independencia	114
8.2	Pruebas para muestras pequeñas	115
8.2.1	Prueba exacta de Fisher	115
8.2.2	Prueba de mcNemar	118
8.3	Prueba Q de Cochran	120
8.4	Ejercicios propuestos	124
9	ANOVA de una vía para muestras independientes	127
9.1	Condiciones para usar ANOVA de una vía para muestras independientes	128
9.2	Procedimiento ANOVA de una vía para muestras independientes	129
9.2.1	Variabilidad total	130
9.2.2	Variabilidad entre grupos	130
9.2.3	Variabilidad al interior de cada grupo	131
9.2.4	El estadístico de prueba F	131
9.2.5	Resultado del procedimiento ANOVA	132
9.2.6	Resumen del procedimiento ANOVA de una vía para muestras independientes	133
9.3	ANOVA de una vía para muestras independientes en R	133
9.4	Ánalysis post-hoc	136

9.4.1	Correcciones de Bonferroni y Holm	136
9.4.2	Prueba HSD de Tukey	137
9.4.3	Prueba de comparación de Scheffé	140
9.5	Ejercicios propuestos	144
10	ANOVA de una vía para muestras correlacionadas	147
10.1	Condiciones para usar ANOVA de una vía para muestras correlacionadas	148
10.2	Procedimiento ANOVA de una vía para muestras correlacionadas	149
10.2.1	Variabilidad total, entre grupos e intra-grupos	149
10.2.2	Variabilidad entre sujetos	149
10.2.3	El estadístico de prueba F	150
10.2.4	Resultado del procedimiento ANOVA	151
10.2.5	Resumen del procedimiento ANOVA de una vía para muestras correlacionadas	152
10.3	ANOVA de una vía para muestras correlacionadas en R	152
10.4	Procedimientos post-hoc	155
10.5	Ejercicios propuestos	157
11	Inferencia no paramétrica con medianas	159
11.1	Pruebas para una o dos muestras	159
11.1.1	Prueba de suma de rangos de Wilcoxon	159
11.1.2	Prueba de rangos con signo de Wilcoxon	166
11.2	Pruebas para más de dos muestras	169
11.2.1	Prueba de Kruskal-Wallis	169
11.2.2	Prueba de Friedman	173
11.3	Ejercicios propuestos	176
12	Remuestreo	179
12.1	Bootstrapping	179
12.1.1	Bootstrapping para una muestra	179
12.1.2	Bootstrapping para dos muestras independientes	186
12.1.3	Bootstrapping para dos muestras pareadas	189
12.1.4	Bootstrapping para comparar medias de más de dos muestras independientes	192
12.2	Pruebas de permutaciones	196
12.2.1	Prueba de permutaciones para comparar una variable continua en dos muestras independientes	197
12.2.2	Prueba de permutaciones para comparar medias de más de dos muestras correlacionadas	201
12.3	Ejercicios propuestos	204
13	Otras alternativas para datos problemáticos	207
13.1	Transformación de datos	207
13.1.1	Transformación lineal	207
13.1.2	Transformación logarítmica	208
13.1.3	Escalera de potencias de Tukey	211
13.1.4	Transformaciones Box-Cox	216
13.2	Métodos robustos	220
13.2.1	Alternativas robustas para la media	221
13.2.2	Prueba de Yuen para dos muestras independientes	222
13.2.3	Prueba de Yuen para dos muestras pareadas	227
13.2.4	Comparaciones de una vía para múltiples grupos independientes	228
13.2.5	Comparaciones de una vía para múltiples grupos correlacionados	231
13.3	Ejercicios propuestos	233
14	Regresión lineal	235
14.1	Correlación	237
14.2	Regresión lineal mediante mínimos cuadrados	238

14.3 Uso del modelo	243
14.4 Regresión lineal con un predictor categórico	243
14.5 Evaluación de un modelo de RLS	246
14.5.1 Influencia de los valores atípicos	247
14.5.2 Bondad de ajuste	247
14.5.3 Validación cruzada	249
14.5.4 Validación cruzada de k pliegues	250
14.5.5 Validación cruzada dejando uno fuera	252
14.6 Inferencia para regresión lineal	252
14.7 Ejercicios propuestos	254
15 Regresión lineal múltiple	257
15.1 RLM con predictores categóricos	259
15.2 Condiciones para usar RLM	261
15.3 Evaluación del ajuste de una RLM	261
15.4 Comparación de modelos	262
15.5 Selección de predictores	263
15.6 Evaluación de un modelo de RLM	270
15.6.1 Identificación de valores con sobreinfluencia	271
15.6.2 Verificación de las condiciones	277
15.6.3 Validación cruzada	280
15.6.4 Tamaño de la muestra	280
15.7 Ejercicios propuestos	280
16 Regresión logística	283
16.1 Evaluación de un clasificador	285
16.2 Bondad de ajuste del modelo	286
16.3 Regresión logística en R	287
16.4 Condiciones para usar regresión logística	289
16.5 Generalización del modelo	290
16.6 Selección de predictores	291
16.7 Comparación de modelos	291
16.8 Regresión logística en R con selección de predictores	292
16.9 Ejercicios propuestos	299

Índice de tablas

Tabla 1.1	algunas filas de la matriz de datos <code>ffaa</code> .	3
Tabla 1.2	descripción de las variables para el conjunto de datos <code>ffaa</code> .	3
Tabla 1.3	descripción de las variables para el conjunto de datos <code>mtcars</code> .	11
Tabla 2.1	descripción de las variables para el conjunto de datos <code>mtcars</code> .	15
Tabla 2.2	tabla de contingencia para la cantidad de cambios de los automóviles.	20
Tabla 2.3	tabla de contingencia para las variables <code>Cambios</code> y <code>Transmisión</code> .	21
Tabla 2.4	tabla de proporciones con totales por fila para la tabla 2.3.	21
Tabla 2.5	tabla de proporciones con totales por columna para la tabla 2.3.	21
Tabla 2.6	tabla de proporciones con totales por fila y columna para la tabla 2.3.	22
Tabla 2.7	tabla de contingencia para tres variables.	22
Tabla 3.1	distribución de probabilidad para el lanzamiento de un dado adulterado.	37
Tabla 4.1	posibles escenarios para una prueba de hipótesis.	61
Tabla 5.1	muestra para el ejemplo de prueba Z con una muestra.	71
Tabla 5.2	tiempo de ejecución para las instancias de la muestra.	75
Tabla 5.3	tiempos de ejecución de cada algoritmo para las instancias de la muestra.	78
Tabla 5.4	Concentración de anticuerpos de los pacientes vacunados.	81
Tabla 8.1	tabla de frecuencias para el lenguaje de programación favorito de la muestra.	110
Tabla 8.2	frecuencias esperadas si hombres y mujeres tienen las mismas preferencias.	111
Tabla 8.3	valor Z para cada grupo.	111
Tabla 8.4	frecuencias por lenguaje de programación para la toda la nómina y para la muestra.	113
Tabla 8.5	proporciones de la población y valores esperados de la muestra.	113
Tabla 8.6	tabla de contingencia para las características de los hongos.	114
Tabla 8.7	frecuencias esperadas para los hongos.	114
Tabla 8.8	tabla de contingencia para dos variables categóricas con dos niveles cada una.	116
Tabla 8.9	tabla de contingencia con los contagios producidos en el experimento.	116
Tabla 8.10	tablas con los mismos valores marginales que los obtenidos.	117
Tabla 8.11	resultados de la predicción para cada estudiante con ambos modelos.	119
Tabla 8.12	tabla de contingencia con las predicciones de los resultados finales de los estudiantes.	119
Tabla 8.13	resultados de las metaheurísticas para cada instancia con ambos modelos.	120
Tabla 9.1	resultado del procedimiento ANOVA.	133
Tabla 9.2	estimadores y valores críticos para los contrastes de la prueba de comparación de Scheffé.	142
Tabla 10.1	tiempos de ejecución para las diferentes instancias con cada algoritmo del ejemplo.	147
Tabla 10.2	diferencias de las varianza del tiempo de ejecución entre cada par de algoritmos.	149
Tabla 10.3	tiempos de ejecución y tiempo medio de ejecución para las diferentes instancias del ejemplo.	150
Tabla 10.4	resultado del procedimiento ANOVA.	151
Tabla 11.1	evaluación de las interfaces de usuario A y B.	160
Tabla 11.2	muestras combinadas con rango.	161

Tabla 11.3 evaluación de las interfaces de usuario A y B.	166
Tabla 11.4 asignación de rangos con signo.	167
Tabla 11.5 valores que puede adoptar el estadístico W para $n = 3$	167
Tabla 11.6 asignación de rangos a la muestra combinada.	170
Tabla 11.7 resumen de los rangos.	170
Tabla 11.8 evaluación realizada por los usuarios a cada una de las distintas interfaces.	174
Tabla 11.9 ranking de las interfaces por usuario.	174
Tabla 11.10resumen de los rangos.	174
Tabla 12.1 tiempo de ejecución para cada instancia de la muestra.	180
Tabla 12.2 muestra original y remuestreos de bootstrap	180
Tabla 12.3 calificaciones de los estudiantes en la primera y la segunda prueba de un curso inicial de programación.	190
Tabla 12.4 pesos de los huevos (en gramos).	192
Tabla 12.5 tiempos de ejecución para las diferentes instancias con cada algoritmo del ejemplo. . .	201
Tabla 13.1 escalera de transformaciones de Tukey.	211
Tabla 14.1 descripción de las variables para el conjunto de datos <code>mtcars</code> usados en este capítulo. .	237
Tabla 14.2 requisitos funcionales y cantidad de <i>stakeholders</i> para diferentes proyectos desarrollados por la empresa.	252
Tabla 15.1 creación de variables artificiales para una matriz de datos con variables categóricas. .	260
Tabla 16.1 tabla de contingencia para evaluar un clasificador.	285

Índice de figuras

Figura 1.1 ejemplos de modelos.	1
Figura 1.2 formatos de archivo para importar datos en R.	6
Figura 2.1 tres distribuciones de población muy distintas con media $\mu = 0$ y desviación estándar $\sigma = 1$	18
Figura 2.2 dos histogramas.	24
Figura 2.3 gráfico de caja para la variable Potencia.	25
Figura 2.4 gráfico de barras para la variable Cambios.	26
Figura 2.5 gráfico de torta para la variable Cambios.	27
Figura 2.6 gráfico de dispersión para las variables Rendimiento y Peso.	28
Figura 2.7 gráficos de dispersión con diferentes tipos de asociación entre las variables.	29
Figura 2.8 gráficos de barras para las variables Cambios y Motor.	30
Figura 2.9 gráfico de mosaico para las variables Cambios y Motor.	32
Figura 2.10 gráfico de cajas por grupo.	33
Figura 2.11 gráfico de tiras.	34
Figura 2.12 gráficos para los ejercicios propuestos.	34
Figura 3.1 distribución de probabilidad para varios lanzamientos de un dado cargado.	38
Figura 3.2 histograma para el desempeño del programa.	40
Figura 3.3 distribución para el desempeño del programa.	41
Figura 3.4 dos ejemplos superpuestos de distribución normal.	42
Figura 3.5 regla empírica de la distribución normal.	43
Figura 3.6 gráfico cuantil-cuantil.	44
Figura 3.7 ejemplo de distribución χ^2 con 2 grados de libertad.	46
Figura 3.8 ejemplo de distribuciones t.	46
Figura 3.9 ejemplo de una distribución F.	47
Figura 3.10 distribución geométrica para obtener un valor específico lanzando un dado de 20 caras.	49
Figura 3.11 distribución binomial con $\mu = 400$ y $\sigma = 15.4019$	50
Figura 3.12 ejemplo de distribución binomial negativa.	51
Figura 3.13 ejemplo de distribución de Poisson.	52
Figura 4.1 medias obtenidas al agregar a la muestra un elemento cada vez.	55
Figura 4.2 distribución muestral de la media para muestras con 100 observaciones.	57
Figura 4.3 probabilidad de encontrar una media igual o menor que $\bar{x} = 527,9$ [ms] en la distribución muestral con $\mu_{\bar{x}} = 530$ y $\sigma_{\bar{x}} = 1,2$	62
Figura 4.4 cuando la prueba de hipótesis es bilateral, se deben colorear ambas colas.	64
Figura 5.1 gráfico Q-Q para la muestra de la tabla 5.1.	72
Figura 5.2 resultado de la prueba Z para una muestra.	73
Figura 5.3 gráfico para comprobar el supuesto de normalidad.	76
Figura 6.1 poder estadístico para prueba t bilateral.	86
Figura 6.2 poder estadístico para prueba t unilateral.	87
Figura 6.3 poder estadístico para pruebas t.	89

Figura 6.4 aumento del poder estadístico a medida que crece el tamaño de la muestra (manteniendo fijos el tamaño del efecto y el nivel de significación).	90
Figura 6.5 distribución de la diferencia de medias del tiempo de ejecución, señalando zonas de rechazo de la hipótesis nula.	92
Figura 6.6 región de rechazo de la hipótesis nula en la distribución cuando el programa <i>B</i> es, en promedio, 4 milisegundos más rápido que el programa <i>A</i>	92
 Figura 8.1 resultado de la prueba Q de Cochran.	121
Figura 8.2 resultados de los procedimientos post-hoc.	123
 Figura 9.1 gráfico para comprobar el supuesto de normalidad en las tres muestras del ejemplo.	129
Figura 9.2 tamaño del efecto medido.	134
Figura 9.3 valores p obtenidos en las pruebas t para cada par de grupos mediante los métodos de Bonferroni y Holm.	138
Figura 9.4 resultado del procedimiento <i>post-hoc</i> HSD de Tukey.	140
Figura 9.5 valores p e intervalos de confianza para las diferencias de las medias obtenidos mediante la prueba de comparación de Scheffé.	143
 Figura 10.1 gráfico para comprobar el supuesto de normalidad en las muestras del ejemplo.	148
Figura 10.2 resultado del procedimiento ANOVA usando la función <code>aov()</code>	152
Figura 10.3 resultado del procedimiento ANOVA usando la función <code>ezANOVA()</code>	154
Figura 10.4 resultado de la prueba de esfericidad de Mauchly realizada por <code>ezANOVA()</code>	154
Figura 10.5 correcciones de esfericidad realizadas por <code>ezANOVA()</code>	154
Figura 10.6 Tamaño del efecto medido.	155
Figura 10.7 resultados de las pruebas post-hoc para el ejemplo.	158
 Figura 11.1 histogramas de las muestras.	160
Figura 11.2 resultado de la prueba de Mann-Whitney (en rigor, de la prueba para el ejemplo.	165
Figura 11.3 distribución de <i>W</i>	167
Figura 11.4 resultado de la prueba de rangos con signo de Wilcoxon para el ejemplo.	168
Figura 11.5 resultado de la prueba de Kruskal-Wallis y el procedimiento post-hoc de Holm para el ejemplo.	172
Figura 11.6 valores p obtenidos en las pruebas t para cada par de grupos mediante los métodos de Bonferroni y Holm.	176
 Figura 12.1 distribución del tiempo de ejecución para la muestra.	180
Figura 12.2 distribución bootstrap de la media	181
Figura 12.3 distribución bootstrap generada mediante <code>boot()</code> para la media.	183
Figura 12.4 histograma y gráfico Q-Q de la distribución bootstrap generada mediante <code>boot()</code> para la media.	183
Figura 12.5 histograma y gráfico Q-Q de la distribución bootstrap generada mediante <code>bootES()</code> para la media.	184
Figura 12.6 distribución bootstrap e intervalo de confianza para la media de la población generada mediante <code>bootES()</code>	184
Figura 12.7 pruebas de normalidad de Shapiro-Wilk para ambas muestras.	187
Figura 12.8 distribución bootstrap de la diferencia de medias.	187
Figura 12.9 intervalo de confianza BCa para la media de las diferencias.	190
Figura 12.10 pruebas de normalidad para las muestras del ejemplo.	193
Figura 12.11 procedimiento post-hoc para el ejemplo.	193
Figura 12.12 histograma y gráfico Q-Q de la distribución para la diferencia de medias generada mediante permutaciones.	198
Figura 12.13 gráfico Q-Q para comprobar el supuesto de normalidad para el ejemplo.	202
Figura 12.14 resultado del procedimiento post-hoc.	202
 Figura 13.1 resultado de la transformación lineal del script 13.1.	208

Figura 13.2 histogramas del peso cerebral antes y después de la transformación logarítmica.	208
Figura 13.3 gráficos de dispersión para el peso corporal y el peso del cerebro antes y después de las transformaciones logarítmicas.	209
Figura 13.4 histograma de la población histórica de Estados Unidos y gráfico de dispersión de la población por año.	212
Figura 13.5 población de Estados Unidos por año tras aplicar la transformación de Tukey con distintos valores de λ	213
Figura 13.6 histograma de la población de Estados Unidos tras aplicar la transformación de Tukey con distintos valores de λ	214
Figura 13.7 gráficos entregados por <code>transformTukey()</code>	215
Figura 13.8 población de Estados Unidos por año tras aplicar la transformación de Box-Cox con distintos valores de λ	217
Figura 13.9 ejemplos de la transformación Box-Cox versus $\log(x)$	218
Figura 13.10 gráficos de población de Estados Unidos por año usando la transformación de Box-Cox. .	219
Figura 13.11 gráfico Q-Q de las muestras originales.	223
Figura 13.12 gráfico Q-Q de las muestras truncadas.	224
Figura 13.13 resultado de la prueba de Yuen para el ejemplo.	224
Figura 13.14 resultado de la prueba de Yuen con bootstrapping para el ejemplo, usando como estimadores la media y la mediana.	226
Figura 13.15 resultado de la prueba de Yuen para el ejemplo, usando como estimadores la media y la mediana.	227
Figura 13.16 resultado de la comparación entre múltiples grupos independientes usando medias truncadas.	229
Figura 13.17 resultado de la comparación entre múltiples grupos independientes usando medias truncadas con bootstrapping.	230
Figura 13.18 resultado de la alternativa robusta para comparar entre múltiples grupos correlacionados. .	232
Figura 14.1 modelos lineales para cuatro conjuntos de datos.	235
Figura 14.2 residuos para los modelos lineales de la figura 14.1.	236
Figura 14.3 Relación entre el rendimiento y la potencia.	238
Figura 14.4 matriz de correlación para el conjunto de datos <code>mtcars</code>	238
Figura 14.5 modelos lineales (fila superior) que violan alguna condición y sus residuos (fila inferior). .	239
Figura 14.6 regresión lineal simple para predecir el rendimiento de un automóvil a partir de su peso. .	240
Figura 14.7 recta ajustada para el rendimiento de un automóvil de acuerdo a su peso.	241
Figura 14.8 gráficos para evaluar el modelo lineal.	242
Figura 14.9 residuos obtenidos tras usar el modelo para predecir el rendimiento de nuevos automóviles. .	244
Figura 14.10 modelo de regresión lineal y gráfico de residuos para el ejemplo con un predictor dicotómico.	245
Figura 14.11 recta de mínimos cuadrados para el ejemplo con un predictor dicotómico.	246
Figura 14.12 distribución de los residuos.	246
Figura 14.13 seis modelos de regresión lineal con sus respectivos gráficos de residuos.	248
Figura 14.14 recta de mínimos cuadrados usando validación cruzada.	249
Figura 14.15 otra recta de mínimos cuadrados usando validación cruzada.	250
Figura 14.16 regresión lineal para la cantidad de requisitos funcionales de acuerdo a la cantidad de <i>stakeholders</i>	253
Figura 14.17 descripción detallada del modelo obtenido por el gerente para el ejemplo.	254
Figura 15.1 descripción del modelo lineal para predecir el rendimiento de un automóvil a partir de dos variables.	258
Figura 15.2 plano ajustado para la RLM con dos predictores.	259
Figura 15.3 resultado del script 15.2.	261
Figura 15.4 resultado del script 15.4.	265
Figura 15.5 resultado de las llamadas a <code>add1()</code> en el script 15.5	266
Figura 15.6 resultado de la llamada a <code>drop1()</code> en el script 15.5	267

Figura 15.7 modelo nulo.	269
Figura 15.8 modelo completo.	270
Figura 15.9 modelos evaluados por la función <code>step()</code> durante el proceso de selección hacia adelante (parte 1).	271
Figura 15.10modelos evaluados por la función <code>step()</code> durante el proceso de selección hacia adelante (parte 2).	272
Figura 15.11modelo obtenido mediante selección hacia adelante.	273
Figura 15.12modelo obtenido mediante eliminación hacia atrás.	274
Figura 15.13modelo obtenido mediante regresión escalonada.	274
Figura 15.14representación gráfica de los mejores modelos encontrados mediante el método de todos los subconjuntos.	275
Figura 15.15gráficos disponibles en R (base) para evaluar un modelo lineal.	276
Figura 15.16identificación de valores atípicos.	277
Figura 15.17verificación de condición de multicolinealidad.	279
 Figura 16.1 función logística.	283
Figura 16.2 dos curvas ROC.	286
Figura 16.3 ajuste de un modelo de regresión logística.	288
Figura 16.4 curva ROC obtenida al evaluar el modelo con el conjunto de entrenamiento.	289
Figura 16.5 matriz de confusión y medidas de evaluación con el conjunto de entrenamiento para el modelo ajustado.	290
Figura 16.6 curva ROC obtenida al evaluar el modelo con el conjunto de prueba.	291
Figura 16.7 matriz de confusión y medidas de evaluación con el conjunto de prueba para el modelo ajustado.	292
Figura 16.8 modelo de regresión logística obtenido mediante regresión escalonada.	293
Figura 16.9 modelo de regresión logística obtenido mediante regresión escalonada.	293
Figura 16.10factores de inflación de la varianza para los modelos.	293
Figura 16.11modelo de regresión logística con el peso como predictor.	294
Figura 16.12modelo de regresión logística con la potencia como predictor.	295
Figura 16.13comparación de los modelos con un único predictor.	295
Figura 16.14comparación del modelo con dos predictores y el que solo tiene el peso como predictor.	296
Figura 16.15resultado de la prueba de Durbin-Watson para verificar la independencia de los residuos del modelo que solo tiene el peso como predictor.	296
Figura 16.16gráficos para evaluar el modelo de regresión logística.	297
Figura 16.17identificación de posibles valores atípicos.	298

Índice de scripts

1.1	sentencias para importar un conjunto de datos.	6
1.2	instalar y cargar paquetes de R.	7
1.3	construir un dataframe.	8
1.4	modificaciones sencillas de una matriz de datos.	8
1.5	modificación de una matriz de datos con el paquete <code>dplyr</code>	9
1.6	modificación de una matriz de datos con el paquete <code>tidyR</code>	10
1.7	modificación del conjunto de datos mtcars para facilitar su comprensión.	12
2.1	uso de las funciones <code>mean()</code> y <code>sapply()</code>	16
2.2	cálculo de cuantiles con la función <code>quantile()</code>	18
2.3	uso de la función <code>summarise()</code> del paquete <code>dplyr</code>	19
2.4	tabla de contingencia para la variable <code>Cambios</code>	20
2.5	tablas de contingencia y proporciones para dos variables.	21
2.6	matriz de confusión para tres variables.	22
2.7	estadísticas descriptivas para datos agrupados.	23
2.8	histogramas para las variables <code>Rendimiento</code> y <code>Potencia</code>	24
2.9	gráfico de caja para la variable <code>Potencia</code>	26
2.10	gráfico de barras para la variable <code>Cambios</code>	26
2.11	gráfico de torta para la variable <code>Cambios</code>	27
2.12	gráfico de dispersión para las variables <code>Rendimiento</code> y <code>Peso</code>	28
2.13	gráficos de dispersión con diferentes tipos de asociación entre las variables.	29
2.14	gráficos de barras para las variables <code>Cambios</code> y <code>Motor</code>	30
2.15	gráfico de mosaico para las variables <code>Cambios</code> y <code>Motor</code>	31
2.16	gráfico de cajas por grupo.	32
2.17	gráfico de tiras.	33
3.1	variables aleatorias discretas en R.	38
3.2	histogramas de variables aleatorias discretas en R.	38
3.3	combinación lineal de variables aleatorias discretas en R.	40
3.4	graficando dos ejemplos de distribución normal.	42
3.5	creación de un gráfico cuantil-cuantil.	44
4.1	representación gráfica de la media móvil.	55
4.2	distribución de la media muestral.	56
4.3	cálculo del valor p para una prueba de una cola.	63
4.4	cálculo del valor p para una prueba de dos colas.	65
5.1	prueba Z para una muestra.	73
5.2	prueba t para una muestra.	77
5.3	inferencia con la media de las diferencias entre dos muestras pareadas usando la distribución t.	79
5.4	prueba t para dos muestras independientes.	82
6.1	poder estadístico para prueba t bilateral.	86
6.2	aumento del poder estadístico a medida que crece el tamaño de la muestra.	90
6.3	cálculo teórico del poder.	92
6.4	cálculo del poder en R.	95
7.1	método de Wald para una proporción.	101
7.2	método de Wald para la diferencia entre dos proporciones (ejemplo 1).	103
7.3	método de Wald para la diferencia entre dos proporciones (ejemplo 2).	105

7.4	método de Wilson para una proporción.	106
7.5	método de Wilson para la diferencia entre dos proporciones.	106
8.1	prueba chi-cuadrado de homogeneidad.	112
8.2	prueba chi-cuadrado de bondad de ajuste.	113
8.3	prueba chi-cuadrado de independencia.	115
8.4	prueba exacta de Fisher.	117
8.5	prueba de McNemar.	119
8.6	prueba Q de Cochran.	122
9.1	procedimiento ANOVA de una vía para muestras independientes.	134
9.2	procedimientos <i>post-hoc</i> de Bonferroni y Holm en R.	136
9.3	procedimiento <i>post-hoc</i> de Tukey.	139
9.4	prueba de comparación de Scheffé.	143
10.1	procedimiento ANOVA de una vía para muestras correlacionadas.	153
10.2	pruebas post-hoc para el ejemplo.	156
11.1	prueba de Mann-Whitney para el ejemplo.	165
11.2	prueba de rangos con signo de Wilcoxon para el ejemplo.	168
11.3	prueba de Kruskal-Wallis y el procedimiento post-hoc de Holm para el ejemplo.	172
11.4	prueba de Friedman y el procedimiento post-hoc de Holm para el ejemplo.	175
12.1	construcción de un intervalo de confianza para la media poblacional mediante bootstrapping.	184
12.2	inferencia sobre la media de una muestra con bootstrapping.	186
12.3	bootstrapping para la diferencia de medias.	187
12.4	bootstrapping para inferir acerca de la diferencia de medias.	189
12.5	bootstrapping para la media de las diferencias.	190
12.6	bootstrapping para inferir acerca de la media de las diferencias.	191
12.7	prueba de permutaciones para muestras correlacionadas.	192
12.8	pruebas de permutaciones para variables numéricas.	198
12.9	prueba de permutaciones para muestras correlacionadas.	201
13.1	transformación lineal para convertir grados Celcius a grados Fahrenheit.	207
13.2	transformación logarítmica.	209
13.3	transformación de Tukey para la población total de Estados Unidos.	213
13.4	transformación de Box-Cox para la población total de Estados Unidos.	217
13.5	prueba de Yuen para dos muestras independientes.	223
13.6	prueba de Yuen con bootstrapping para dos muestras independientes usando la media y la mediana.	225
13.7	prueba de Yuen para dos muestras pareadas.	227
13.8	alternativas robustas para comparar entre múltiples grupos independientes.	228
13.9	alternativa robusta para comparar entre múltiples grupos correlacionados.	231
14.1	ajuste de una regresión lineal simple.	241
14.2	reemplazar una variable dicotómica por una variable indicadora.	244
14.3	alternativa robusta para comparar entre múltiples grupos correlacionados.	244
14.4	ajuste de una regresión lineal simple usando validación cruzada.	249
14.5	ajuste de una regresión lineal simple usando validación cruzada de 5 pliegues.	251
14.6	regresión lineal para la cantidad de requisitos funcionales de acuerdo a la cantidad de <i>stakeholders</i>	253
15.1	regresión lineal para predecir el rendimiento de un automóvil a partir de dos variables.	257
15.2	creación de variables artificiales para variables categóricas.	260
15.3	comparación de dos modelos lineales.	263
15.4	incorporación y eliminación de variables en un modelo de RLM.	264
15.5	Evaluación de variables a incorporar y eliminar en un modelo de RLM.	266
15.6	selección de predictores a incluir en una RLM.	268
15.7	identificación de valores atípicos.	273
15.8	verificación de condiciones para el modelo.	279
16.1	ajuste de un modelo de regresión logística en R.	288
16.2	ajuste de un modelo de regresión logística usando validación cruzada.	290
16.3	ajuste y evaluación del mejor modelo para predecir el tipo de transmisión de un automóvil.	294

CAPÍTULO 1. INTRODUCCIÓN

Este libro tiene como propósito acompañarte en el aprendizaje de las primeras nociones de inferencia estadística y de creación de modelos estadísticos. En este primer capítulo comenzaremos por buscar definiciones iniciales para los conceptos de inferencia y modelo, para luego abordar algunas nociones iniciales acerca de los datos empleados en estadística y algunas herramientas para que puedas empezar a usar el entorno de programación R, con el cual trabajaremos a lo largo de todo el texto. Te sugerimos, entonces, que lo instales junto con el entorno de desarrollo integrado RStudio.

1.1 INFERENCIA Y MODELOS ESTADÍSTICOS

La Real Academia Española (2014) define **inferencia** como “acción y efecto de inferir”. Esto por sí solo no nos dice mucho, pero si buscamos también la definición de **inferir**, encontraremos que significa “deducir algo o sacarlo como conclusión de otra cosa”. A partir de estas definiciones, y de acuerdo con Devore (2008, p. 5), podemos decir que la **estadística inferencial** es una rama de la estadística que busca obtener una conclusión para un conjunto de individuos o elementos (denominado **población**) a partir de información recolectada de un subconjunto de éste (llamado **muestra**).

Llegar a una definición de **modelo estadístico** puede ser bastante más complejo. Como nos muestra la figura 1.1, ¡un modelo puede ser muchas cosas diferentes! Veamos qué nos dice la Real Academia Española (2014):



Figura 1.1: ejemplos de modelos.

1. Arquetipo o punto de referencia para imitarlo o reproducirlo.
2. En las obras de ingenio y en las acciones morales, ejemplar que por su perfección se debe seguir e imitar.
3. Representación en pequeño de alguna cosa.
4. Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, como la evolución económica de un país, que se elabora para facilitar su comprensión y el estudio de su comportamiento.
5. Objeto, aparato, construcción, etc., o conjunto de ellos realizados con arreglo a un mismo diseño. *Auto modelo 1976. Lavadora último modelo*.
6. Vestido con características únicas, creado por determinado modista, y, en general, cualquier prenda de vestir que esté de moda.
7. En empresas, u. en aposición para indicar que lo designado por el nombre anterior ha sido creado como ejemplar o se considera que puede serlo. *Empresa modelo. Granjas modelo*.
8. Esc. Figura de barro, yeso o cera, que se ha de reproducir en madera, mármol o metal.
9. *Cuba* impreso (||hoja con espacios en blanco).
10. Persona que se ocupa de exhibir diseños de moda.
11. Persona u objeto que copia el artista.

Puede ser de ayuda tener en cuenta algunas definiciones e ideas que nos ofrece la literatura. Kaplan (2009), por ejemplo, señala que un modelo es una representación con un propósito particular. Pero otros contribuyen a enriquecer esta definición:

- Representación simplificada de la realidad en la que aparecen algunas de sus propiedades (Joly, 1988).
- Permiten estudiar de forma simple y comprensible una porción de la realidad (Ríos, 1995).
- Dejan cosas fuera y pueden llevar a conclusiones equivocadas (Kaplan, 2009).
- Resumen de manera conveniente, a juicio de los sus creadores, los aspectos más relevantes del fenómeno estudiado y sus relaciones (Mendez Ramírez, 1998).
- Están mediados por el diseño, es decir la forma de seleccionar y observar (o manipular) la realidad modelada (Mendez Ramírez, 2012).
- Son pequeños, económicos, seguros y fáciles de transportar, copiar y modificar (Kaplan, 2009).

Si a esta concepción del significado de la palabra modelo le agregamos nuevos conceptos fundamentales que nos ofrecen otros autores, podemos acercarnos un poco más a la idea de **modelo estadístico**:

- Modelo matemático de la regularidad estadística de los posibles resultados de la evolución de un fenómeno aleatorio (Mendez Ramírez, 1998).
- Distribución de probabilidad construida para poder hacer inferencias o tomar decisiones desde datos (Freedman, 2009).
- Descripción simple de un proceso probabilístico que puede haber dado origen a un conjunto de datos observados (McCullagh, 2002).
- Modelo estocástico que contiene parámetros desconocidos que deben ser estimados en base a suposiciones acerca del modelo y los datos (SAS Institute Inc., 2008).

Pero, ¿para qué sirven los modelos estadísticos? Diversos autores nos muestran que tales modelos son muy útiles en diversos contextos:

- Para describir (Kaplan, 2009) o resumir datos (Freedman, 2009).
- Para clasificar (Kaplan, 2009) o predecir (Freedman, 2009; Kaplan, 2009).
- Para anticipar el resultado de intervenciones (Freedman, 2009; Kaplan, 2009).

Ahora que hemos definido nuestros conceptos iniciales, veamos algunas definiciones y herramientas que nos permitan comenzar a explorarlos.

1.2 VARIABLES, PARÁMETROS Y ESTADÍSTICOS

Comencemos a partir de un ejemplo para ilustrar algunas ideas iniciales acerca de los datos. La tabla 1.1 muestra las primeras filas de la matriz de datos `ffaa`¹, que almacena información acerca de miembros activos de las Fuerzas Armadas de Chile. Cada columna de la matriz representa una **variable** o característica, mientras que cada fila corresponde a una **unidad de observación** o instancia. Así, cada fila de la tabla 1.1 almacena datos de una misma persona. El uso de **matrices de datos** para almacenar los datos es muy conveniente, pues nos ayuda a acceder a los datos y modificarlos más fácilmente. Por ejemplo, para agregar una nueva observación, basta con añadir una nueva fila a la matriz. Si queremos eliminar una característica, simplemente borramos la columna correspondiente. Para consultar una característica en particular de una observación, solo necesitamos conocer la fila y la columna correspondientes.

id	género	estatura	escalafón	servicio	antigüedad	rama
1	M	1,77	S	89,91	15	E
2	M	1,97	O	65,14	30	C
3	F	1,65	O	97,03	12	A
4	M	1,82	S	76,29	9	A
5	F	1,73	S	69,46	7	M
6	M	1,78	S	97,67	21	E
7	M	1,87	O	72,09	27	C
8	F	1,91	S	94,53	11	A
:	:	:	:	:	:	:
6051	M	1.72	S	86.48	17	E

Tabla 1.1: algunas filas de la matriz de datos `ffaa`.

Antes de comenzar a trabajar con los datos, tenemos que estar seguros de comprender cada uno de sus aspectos. Para ello, las siguientes preguntas pueden ser un excelente punto de partida:

- ¿A qué corresponde cada característica?
- ¿Cuáles son sus unidades de medición?
- ¿Qué valores puede tomar?

La tabla 1.2 muestra la descripción de las características presentes en la matriz de datos de la tabla 1.1. En ella se explica el significado de cada columna de la matriz junto a su rango de valores o a un listado de valores posibles.

Variable	Descripción
id	Identificador de la observación.
género	Género del estudiante (M: masculino, F: femenino).
estatura	Estatura (m).
escalafón	Escalafón al que pertenece (O: oficial, S: suboficial).
servicio	Evaluación de servicio (entre 0 y 100).
antigüedad	Años que lleva en servicio activo.
rama	Rama de las FF.AA. A la que pertenece (E: Ejército, M: Marina, A: Fuerza Aérea, C: Carabineros).

Tabla 1.2: descripción de las variables para el conjunto de datos `ffaa`.

Si estudias la tabla 1.2 con detalle, podemos notar diferencias interesantes entre las variables, más allá de su descripción. Por ejemplo, no todas ellas pueden tomar los mismos valores. Con esto aparece la noción de **tipos de variables**, los cuales podemos jerarquizar:

¹ Los datos aquí presentados son ficticios y han sido creados únicamente con fines pedagógicos.

- **Numéricas:** pueden tomar muchos valores numéricos, y son sensibles a operaciones aritméticas. Pueden separarse en:
 - **Continuas:** pueden tomar cualquier valor (en un intervalo) del conjunto de los reales. Por ejemplo, las variables **estatura** y **servicio** descritas en la tabla 1.2.
 - **Discretas:** no es posible que tomen cualquier valor (en un intervalo). Por ejemplo, podrían tomar únicamente valores enteros no negativos, como la variable **antigüedad** de la matriz de datos **ffaa**.
- **Categóricas:** solo pueden tomar un valor de entre un conjunto acotado. Cada posible valor se denomina **nivel**. Entre las variables categóricas es posible distinguir variables:
 - **Nominales:** no existe un orden natural entre los niveles. Ejemplos de variables nominales son **género** y **rama** de la matriz de datos **ffaa**.
 - **Ordinales:** existe un orden natural entre los niveles. Por ejemplo, la idea de jerarquía es evidente al distinguir entre oficiales y suboficiales en la variable **escalafón** de la tabla 1.1.

Tener diferentes tipos de variables significa que debemos medirlas con distintas clases de escalas, las cuales se distinguen por sus propiedades y los tipos de operaciones que permiten:

- **Escala nominal:** sirve solo para separar un conjunto de elementos en subclases excluyentes entre sí. Los valores no son más que nombres o estados, por lo que no podemos hacer operaciones aritméticas ni podemos establecer relaciones de orden.
- **Escala ordinal o de rangos:** esta escala, al igual que la nominal, permite separar un conjunto de elementos en subclases excluyentes entre sí. Una vez más, los valores son solo nombres o estados, por lo que tampoco podemos hacer operaciones aritméticas. Pero en este caso sí podemos establecer una relación de orden, aunque para ello es necesario que la variable tenga a lo menos tres niveles. A modo de ejemplo, si queremos una variable para medir el nivel de estudios de las personas en un grupo demográfico, podríamos considerar una escala ordinal con los niveles “ninguna”, “básica completa”, “media”, “superior” y “postgrado”. Aquí podemos apreciar claramente que los niveles están ordenados de manera creciente.
- **Escala de intervalo:** sirve para datos continuos o discretos con una gran cantidad de niveles. Además de la noción de orden de la escala ordinal, se cumple que la distancia entre dos valores cualesquiera de la escala es conocida y constante, por lo que podemos emplear operaciones aritméticas. Aunque el punto cero y la unidad de medida son arbitrarios, la razón entre dos intervalos es independiente de ambos elementos. Tomemos, por ejemplo, la escala Celsius de temperatura. El cero está dado por el punto de congelación del agua. La medida o tamaño se calcula en base a los puntos de congelación y ebullición del agua. Sin embargo, a pesar de estos parámetros arbitrarios, el cambio en la cantidad de calor es el mismo si aumentamos la temperatura de 10 a 15 grados Celsius, o de 25 a 30. Si miramos ahora la escala Fahrenheit de temperatura, los puntos fijos son diferentes a los empleados por la escala Celsius, por lo que el cero no significa lo mismo. Sin embargo, existe una transformación lineal que nos permite transformar una medida en una escala a su equivalente en otra escala.
- **Escala de razón:** cumple con todos los atributos de la escala de intervalos, pero además tiene su origen en un cero verdadero. Ejemplos de tales escalas son, por ejemplo, las que permiten medir la masa o la distancia. En una escala de razón, la diferencia entre dos puntos es independiente de la unidad de medida. Por ejemplo, si medimos la masa de dos objetos, la razón es constante independientemente de si empleamos kilogramos, libras u onzas (a diferencia de lo que ocurre con la temperatura usando las escalas Celsius y Fahrenheit).

La estadística usa los datos para responder diversas preguntas, muchas de las cuales se orientan a encontrar relaciones entre variables. Así, dos variables pueden ser:

1. **Independientes:** no existe asociación o relación entre las variables.
2. **Dependientes:** existe una asociación o relación entre las variables. Puede existir:
 - **Asociación positiva:** si una variable crece, la otra también lo hace.
 - **Asociación negativa:** si una variable crece, la otra decrece.

En el contexto de la estadística, decimos que un **parámetro** es cualquier número que describa una población en forma resumida, como por ejemplo la media poblacional. A su vez, un **estadístico** es “cualquier cantidad cuyo valor puede ser calculado a partir de datos muestrales” Devore (2008, p. 204), como por ejemplo la media, la mediana o la desviación estándar de un conjunto de datos observados. Si bien a primera vista

ambos conceptos parecen similares, en realidad existe una diferencia importante entre ellos: el parámetro describe una población, mientras que el estadístico, al ser calculado a partir de una muestra, no es más que una **estimación puntual** del parámetro.

Si necesitas más ejemplos o quieres complementar lo aprendido, puedes consultar los textos de referencia para esta sección. Diez, Barr y Çetinkaya-Rundel (2017, pp. 9-19) describe los principales conceptos relativos a datos, tipos de variables y relaciones entre variables. En Dagnino (2014) puedes aprender más sobre escalas de medición.

1.3 CONOCIENDO R

R es un ambiente de software gratuito para estadística computacional y elaboración de gráficos. En esta sección conoceremos algunas herramientas que nos ayudarán a lo largo de este libro. Desde luego, estas breves páginas no pretenden ser un tutorial completo del lenguaje, sino más bien un punto de partida para que podamos aplicar los contenidos que aquí se abordan. Como ya señalamos, sugerimos el uso del entorno integrado de desarrollo RStudio, cuya documentación e instrucciones de instalación podemos consultar en RStudio (2021). En The R Foundation (s.f.) y Carchedi, De Mesmaeker y Vannoorenberghe (s.f.) podemos encontrar documentación acerca del lenguaje R y sus paquetes.

1.3.1 Importación de datos

Una de las primeras cosas que necesitamos conocer es cómo importar o cargar una matriz de datos (denominada *data frame* en R) desde un archivo de texto plano (.txt) o de valores separados por coma (.csv). Para lograrlo con éxito, debemos tener en cuenta algunas orientaciones para preparar los datos adecuadamente:

- La primera fila se usa para los nombres de las columnas o variables.
- La primera columna contiene los nombres de las observaciones, que deben ser únicos.
- Los nombres de las columnas deben respetar las convenciones de R:
 - No está permitido el uso de espacios ni símbolos especiales (?, \$, *, +, #, (,), -, /, }, {, |, >, <, etc.). Solo se admite el uso de puntos (.) y guiones bajos (_).
 - Los nombres de variables no pueden comenzar con un dígito.
 - Los nombres de las columnas deben ser únicos.
- R es sensible a las mayúsculas.
- No puede haber filas en blanco.
- No debe tener comentarios.
- Los valores faltantes deben ser denotados mediante NA.
- Para columnas con fechas, se usa el formato mm/dd/aaaa.
- El archivo debe tener uno de los siguientes formatos, ejemplificados en la figura 1.2:
 - Extensión .txt con tabulaciones como delimitador y punto decimal para valores flotantes.
 - Extensión .csv en formato inglés, con comas (,) como delimitador y punto decimal para valores flotantes.
 - Extensión .csv en formato español, con punto y comas (;) como delimitador y coma decimal para valores flotantes.

El script 1.1 muestra las diferentes funciones para importar datos en R, donde las líneas que comienzan por # corresponden a comentarios. La línea 2 carga el conjunto de datos `mtcars`, disponible en R, mientras que las líneas 5, 9 y 16 importan datos desde archivos. Tanto `read.delim()` como `read.csv()` y `read.csv2()` se usan

	id	género	estatura	escalafón	servicio	antigüedad	rama
1	M	1.77	S	89.91	15	E	
2	M	1.97	O	65.14	30	C	
3	F	1.65	O	97.03	12	A	
4	M	1.82	S	76.29	9	A	
5	F	1.73	S	69.46	7	M	
6	M	1.78	S	97.67	21	E	
7	M	1.87	O	72.09	27	C	
8	F	1.91	S	94.53	11	A	

(a) Texto plano delimitado por tabulaciones.

```
id,género,estatura,escalafón,servicio,antigüedad,rama
1,M,1.77,S,89.91,15,E
2,M,1.97,O,65.14,30,C
3,F,1.65,O,97.03,12,A
4,M,1.82,S,76.29,9,A
5,F,1.73,S,69.46,7,M
6,M,1.78,S,97.67,21,E
7,M,1.87,O,72.09,27,C
8,F,1.91,S,94.53,11,A
```

(b) Valores separados por comas (inglés).

```
id;género;estatura;escalafón;servicio;antigüedad;rama
1;M;1,77;S;89,91;15;E
2;M;1,97;O;65,14;30;C
3;F;1,65;O;97,03;12;A
4;M;1,82;S;76,29;9;A
5;F;1,73;S;69,46;7;M
6;M;1,78;S;97,67;21;E
7;M;1,87;O;72,09;27;C
8;F;1,91;S;94,53;11;A
```

(c) Valores separados por punto y comas (español).

Figura 1.2: formatos de archivo para importar datos en R.

de la misma forma, pudiendo recibir como argumento una llamada al selector de archivos (`file.choose()`), como en la línea 5, o la ruta completa para el archivo, como en la línea 9. En el caso de la línea 16, basta con proporcionar el nombre de archivo pues la función `setwd()` (línea 12) permite establecer el directorio de trabajo de R para la sesión. Las funciones `head()` y `tail()` (líneas 20 y 24) proporcionan una buena manera de inspeccionar los datos cargados, pues muestran por consola las primeras y últimas filas de la matriz de datos, respectivamente.

Script 1.1: sentencias para importar un conjunto de datos.

```
1 # Cargar un conjunto de datos disponible en R.
2 datos1 <- mtcars
3
4 # Importar desde un archivo de texto plano delimitado por tabuladores.
5 datos2 <- read.delim(file.choose())
6
7 # Importar desde un archivo de valores separados por coma
8 # en formato inglés (figura 1.2 b).
9 datos3 <- read.csv("C:\\\\Inferencia\\\\ejemplo1-csv-eng.csv")
10
11 # Configurar carpeta de trabajo
12 setwd("C:\\\\Inferencia")
13
14 # Importar desde un archivo de valores separados por coma
15 # en formato español (figura 1.2 c).
16 datos4 <- read.csv2("ejemplo1-csv-esp.csv")
17
18 # Mostrar las primeras 6 filas del conjunto de datos
19 # almacenado en datos1.
20 head(datos1)
21
22 # Mostrar las últimas 6 filas del conjunto de datos
23 # almacenado en datos1.
24 tail(datos1)
```

1.3.2 Importación de paquetes

Si bien el entorno R básico incluye muchísimas funcionalidades, existe una enorme variedad de paquetes o colecciones que incorporan otras nuevas o mejoran las ya existentes.

Antes de usar un paquete por primera vez tenemos que instalarlo. Para ello, podemos usar la sentencia que se muestra en la línea 2 del script 1.2. Debemos tener en cuenta que la función `install.packages()` requiere que el nombre del paquete se escriba entre comillas.

Para poder usar un paquete, existen las sentencias `library()` (línea 5 del script 1.2) y `require()` (línea 8), que reciben como argumento el nombre del paquete (sin comillas). Si bien ambas sentencias pueden usarse indistintamente, se diferencian en que `library()` termina la ejecución con un mensaje de error si el paquete no está instalado, mientras que `require()` solo emite una advertencia.

Una forma elegante de evitar errores es verificar si un paquete se encuentra instalado antes de usarlo, para lo que podemos usar una combinación de las sentencias anteriores, como muestran las líneas 11 a 14 del script 1.2. Cabe destacar que la opción `dependencies = TRUE` en la línea 12 asegura que se instalen además aquellos paquetes que son requeridos por el que se desea instalar. Fijémonos que el lenguaje de programación R usa **argumentos con nombre**.

Script 1.2: instalar y cargar paquetes de R.

```
1 # Instalar un paquete.
2 install.packages("ggpubr")
3
4 # Primera forma de importar un paquete.
5 library(ggpubr)
6
7 # Segunda forma de importar un paquete.
8 require(ggplot2)
9
10 # Importar un paquete, instalándolo de ser necesario.
11 if(!require(dplyr)){
12   install.packages("dplyr", dependencies=TRUE)
13   require(dplyr)
14 }
```

1.3.3 Construcción de una matriz de datos

Consideremos la idea de construir una matriz de datos que contenga el nombre, la fecha de nacimiento y las calificaciones de los estudiantes en las tres evaluaciones de una asignatura. El script 1.3 crea esta matriz de datos en R con tres observaciones. En las líneas 2 a 4 crea un vector de strings con los nombres de los estudiantes y lo almacena en la variable `nombre`. De manera similar, en la línea 8 crea un vector de fechas. Debemos notar que para ello construye un vector de tres strings con las fechas en formato aaaa-mm-dd, el cual es entregado como argumento a la función `as.Date()` para que sean convertidos al formato de fecha. Las líneas 12 a 14 crean tres vectores de flotantes para las calificaciones obtenidas por los estudiantes. Hasta este punto, solo se tienen muchas variables con vectores de largo 3, los cuales deben ser combinados para formar una matriz de datos donde cada vector sea una columna. La función `data.frame()`, en las líneas 18 a 22, realiza esta tarea. Dicha función recibe como argumentos tantos vectores como variables tenga el conjunto de datos, y toma los nombres de las variables que los contienen como nombres de las columnas. Cabe destacar que, en la línea 23, `data.frame()` recibe un argumento adicional, el booleano `stringsAsFactors`, con valor

falso. Esto se debe a que, si no se entrega este parámetro, R asume que su valor por defecto es verdadero, por lo que interpreta el vector de strings como una variable categórica y asigna un valor numérico a cada nivel.

La última línea del script 1.3 permite guardar la matriz de datos en un archivo de valores separados por comas (formato español). La función `write.csv2()` recibe como argumentos el nombre de la variable que contiene la matriz de datos y una cadena de caracteres con el nombre del archivo. El argumento `row.names = FALSE` indica que no deseamos guardar los nombres de las filas. Si queremos guardar nuestra matriz de datos en un archivo separado por comas en formato inglés, podemos hacerlo mediante la función `write.csv()`, que funciona del mismo modo que `write.csv2()`.

Script 1.3: construir un dataframe.

```
1 # Crear un vector de strings y guardarlo en la variable nombre.
2 nombre <- c("Alan Brito Delgado",
3           "Zacarias Labarca del Rio",
4           "Elsa Payo Maduro")
5
6 # Crear un vector de fechas y guardarlo en la variable
7 # fecha_nacimiento.
8 fecha_nacimiento <- as.Date(c("2008-1-25", "2006-10-4", "2008-3-27"))
9
10 # Crear tres vectores de reales entre 1.0 y 7.0 y guardarlos
11 # en prueba_i, respectivamente.
12 prueba_1 <- c(5.5, 3.4, 4.5)
13 prueba_2 <- c(3.2, 4.7, 4.1)
14 prueba_3 <- c(4.8, 4.3, 5.1)
15
16 # Construir un data frame a partir de los vectores anteriores y
17 # guardarlos en la variable dataframe.
18 dataframe <- data.frame(nombre,
19                         fecha_nacimiento,
20                         prueba_1,
21                         prueba_2,
22                         prueba_3,
23                         stringsAsFactors = FALSE)
24
25 # Guardar un dataframe en un archivo csv (formato español).
26 write.csv2(dataframe, "C:/Inferencia/Ejemplo.csv", row.names = FALSE)
```

1.3.4 Modificación de una matriz de datos

Muchas veces tendremos la necesidad de modificar la matriz de datos. Algunas tareas, como agregar o quitar una columna o un observación pueden hacerse de manera bastante sencilla, como ilustra el script 1.4.

Script 1.4: modificaciones sencillas de una matriz de datos.

```
1 # Leer un dataframe desde archivo csv.
2 datos <- read.csv2("C:/Inferencia/Ejemplo.csv", stringsAsFactors = FALSE)
3
4 # Eliminar del data frame la columna fecha_nacimiento.
5 dataframe$fecha_nacimiento <- NULL
6
7 # Agregar al data frame la columna edad.
8 dataframe$edad <- c(23, 25, 23)
9
```

```

10 # Crear una nueva observación.
11 nueva <- data.frame(nombre="Elba Calao del Río",
12                       prueba_1 = 6.4,
13                       prueba_2 = 2.3,
14                       prueba_3 = 4.6,
15                       edad = 24)
16
17 # Agregar la nueva observación al data frame.
18 datafram <- rbind(dataframe, nueva)
19
20 # Eliminar las primeras 3 observaciones del data frame.
21 datafram <- datafram[-c(1:3),]
22
23 # Guardar el datafram en un archivo csv .
24 write.csv2(datafram, "C:/Inferencia/Ejemplo_mod.csv", row.names = FALSE)

```

Sin embargo, también podemos vernos en la necesidad de realizar transformaciones más complejas. El paquete `dplyr` ofrece un conjunto de funciones que simplifica esta tarea:

- `filter()`: selecciona instancias (filas) de acuerdo a su valor.
- `arrange()`: modifica el orden de las filas.
- `select()`: permite seleccionar variables (características) por sus nombres, a la vez que las reordena.
- `mutate()`: permite agregar nuevas variables que se obtienen como funciones de otras ya existentes.

Para mostrar el uso de estas funciones (script 1.5) usaremos el conjunto de datos `iris`, disponible en R. Este contiene 150 observaciones pertenecientes a tres especies de una flor llamada `iris`: `setosa`, `versicolor` y `virginica`, para las cuales se registran el largo y ancho de sus sépalos y de sus pétalos (en centímetros). Puedes consultar otras funciones y ejemplos más detallados en Müller (2021) y Wickham y Grolemund (2017, cap. 5).

Script 1.5: modificación de una matriz de datos con el paquete `dplyr`.

```

1 library(dplyr)
2
3 # Cargar datafram iris incluido en R.
4 datos <- iris
5
6 # Seleccionar observaciones correspondientes a la especie versicolor.
7 versicolor <- datos %>% filter(Species == "versicolor")
8
9 # Seleccionar observaciones de la especie versicolor cuyos sépalos tengan una
10 # longitud igual o superior a 6 cm.
11 largas <- datos %>% filter(Species == "versicolor" & Sepal.Length >= 6)
12
13 # Seleccionar la especie y variables relativas a los pétalos.
14 petalos <- datos %>% select(Species, starts_with("Petal"))
15
16 # Seleccionar variables de ancho y la especie.
17 anchos <- datos %>% select(ends_with("Width"), Species)
18
19 # Agregar al conjunto de datos de los pétalos una nueva variable con la razón
20 # entre el largo y el ancho de éstos.
21 petalos <- petalos %>% mutate(Species, Petal.Width,
22                               Petal.Ratio = Petal.Length / Petal.Width)
23
24 # Ordenar el conjunto de datos de pétalos en forma descendente según la razón
25 # de los pétalos.
26 petalos <- petalos %>% arrange(desc(Petal.Ratio))
27
28 # Ordenar el conjunto de datos de pétalos en forma ascendente según el largo de

```

```

29 # los pétalos.
30 petalos <- petalos %>% arrange(Petal.Length)

```

En el script 1.5 aparece frecuentemente el operador `%>%`, llamado *pipe* y definido en el paquete `magrittr`, cuya función es entregar un valor o el resultado de una expresión a la siguiente llamada a una función. En términos sencillos, la expresión `x %>% f` es equivalente a `f(x)`, y su utilidad es que simplifica la lectura de llamadas a funciones anidadas (Bache, 2014).

Otra transformación que se usa a menudo es la de pivotar la matriz de datos, cuyo efecto es el de “alargar” o “ensanchar” la matriz. En el primer caso, se incrementa la cantidad de filas (observaciones) a la vez que se reduce la cantidad de columnas (variables). Para ello se usa la función `pivot_longer(cols, names_to, values_to)` del paquete `tidyverse`, donde:

- `cols`: nombres de las columnas a pivotar.
- `names_to`: especifica el nombre de una nueva columna cuyos valores corresponden a los nombres de las columnas a pivotar.
- `values_to`: especifica el nombre de una nueva columna donde se almacenan los valores de las columnas a pivotar.

En el segundo caso se obtiene como resultado una reducción de la cantidad de filas junto al aumento de la cantidad de columnas. Para ello se usa la función `pivot_wider(names_from, values_from)`, también del paquete `tidyverse`, donde:

- `names_from`: especifica el nombre de una variable desde la que se obtienen los nombres de las nuevas columnas.
- `values_from`: especifica el nombre de una variable desde donde se obtienen los valores de las nuevas columnas.

Veamos con un ejemplo el efecto de estas dos transformaciones. El script 1.6 comienza por crear una matriz de datos en que se registran los tiempos de ejecución (en milisegundos) para seis instancias de un problema con cuatro algoritmos diferentes. Las columnas de la matriz de datos original corresponden al identificador de la instancia y cada uno de los algoritmos. Así, la matriz de datos original tiene 6 filas y 5 columnas.

A continuación, se crea una nueva matriz de datos, `datos_largos`, que resulta de pivotar la original para “alargarla”. Al ejecutar el script 1.6 podemos ver que nuestra nueva matriz de datos tiene solo tres columnas, pero que su cantidad de filas es 24. Si miramos con atención, veremos que ahora tenemos 4 filas por cada instancia, una por cada algoritmo (señalado en la columna `Algoritmo`) con su correspondiente tiempo de ejecución (columna `Tiempo`).

Por último, el script 1.6 crea otro conjunto de datos, `datos_anchos`, a partir de `datos_largos`. Al examinar este nuevo conjunto, se puede apreciar que es idéntico al creado inicialmente.

Script 1.6: modificación de una matriz de datos con el paquete `tidyverse`.

```

1 library(dplyr)
2 library(tidyverse)
3
4 # Crear el data frame.
5 Instancia <- 1:6
6 Quicksort <- c(23.2, 22.6, 23.4, 23.3, 21.8, 23.9)
7 Bubblesort <- c(31.6, 29.3, 30.7, 30.8, 29.8, 30.3)
8 Radixsort <- c(30.1, 28.4, 28.7, 28.3, 29.9, 29.1)
9 Mergesort <- c(25.0, 25.7, 25.7, 23.7, 25.5, 24.7)
10 datos <- data.frame(Instancia, Quicksort, Bubblesort, Radixsort, Mergesort)
11
12 # Mostrar las primeras filas de la matriz de datos.
13 cat("Datos originales\n")
14 print(head(datos))
15 cat("\n")
16

```

```

17 # Convertir la matriz de datos a formato largo.
18 datos_largos <- datos %>% pivot_longer(c("Quicksort", "Bubblesort",
19                                         "Radixsort", "Mergesort"),
20                                         names_to = "Algoritmo",
21                                         values_to = "Tiempo")
22
23 # Mostrar las primeras filas de la matriz de datos largos.
24 cat("Datos largos\n")
25 print(head(datos_largos))
26 cat("\n")
27
28 # Convertir la matriz de datos largos a formato ancho.
29 datos_anchos <- datos_largos %>% pivot_wider(names_from = "Algoritmo",
30                                         values_from = "Tiempo")
31
32 # Mostrar las primeras filas de la matriz de datos largos.
33 cat("Datos anchos\n")
34 print(head(datos_anchos))
35 cat("\n")

```

Habrás notado que para poder usar las funciones de `tidy` se requiere también el paquete `dplyr`. Una alternativa es cargar únicamente el paquete `tidyverse`, el cual los incluye a ambos (entre otros).

Puedes encontrar descripciones más extensas acerca del uso de la funciones del paquete `tidyverse`, junto con ejemplos más avanzados, en Wickham (2021).

En ocasiones puede ser necesario renombrar las columnas para que nos resulte más fácil comprender a qué variable corresponde. La función `rename()` del paquete `dplyr` nos permite hacer esta operación bastante sencilla. Sus argumentos son una lista de elementos de la forma `nuevo nombre = nombre original`. También podemos cambiar el tipo de una variable. Una conversión que nos será muy útil es de variable numérica a categórica, lo que se logra mediante la función `factor(x, levels, labels, ordered)`, donde:

- `x`: nombre de la variable a convertir.
- `levels`: argumento opcional con los posibles valores de la variable categórica.
- `labels`: argumento opcional con las etiquetas asociadas a cada valor.
- `ordered`: valor lógico que especifica si la variable es o no ordinal (falso por defecto).

Tomemos el conjunto de datos `mtcars` (incluido en R) para exemplificar el uso de estas funciones. La tabla 1.3 muestra la descripción de estos datos. El script 1.7 modifica los nombres de las columnas para que sean más representativos y da formato de variable categórica a las variables que así lo requieren, asignando etiquetas adecuadas para cada nivel.

Variable	Descripción
mpg	Rendimiento, en millas / galón (EEUU).
cyl	Número de cilindros.
disp	Desplazamiento, en pulgadas cúbicas.
hp	Potencia, en caballos de fuerza brutos.
drat	Razón del eje trasero.
wt	Peso, en miles de libras.
qsec	Tiempo que tarda en recorrer un cuarto de milla partiendo desde el reposo, en segundos.
vs	Tipo de motor (0: en forma de V, 1: recto).
am	Tipo de transmisión (0: automática, 1: manual).
gear	Número de marchas hacia adelante.
carb	Número de carburadores.

Tabla 1.3: descripción de las variables para el conjunto de datos `mtcars`.

Script 1.7: modificación del conjunto de datos mtcars para facilitar su comprensión.

```
1 library(dplyr)
2
3 # Cargar conjunto de datos.
4 datos <- mtcars
5
6 # Renombrar columnas.
7 datos <- datos %>% rename(Rendimiento = mpg, Cilindrada = cyl,
8                               Desplazamiento = disp, Potencia = hp,
9                               Eje = drat, Peso = wt, Cuarto_milla = qsec,
10                              Motor = vs, Transmision = am, Cambios = gear,
11                              Carburadores = carb)
12
13 # Dar formato categórico a las variables Motor y Transmision, renombrando
14 # sus niveles.
15 datos[["Motor"]] <- factor(datos[["Motor"]], levels = c(0, 1),
16                               labels = c("V", "Recto"))
17
18 datos[["Transmision"]] <- factor(datos[["Transmision"]], levels = c(0, 1),
19                               labels = c("Automático", "Manual"))
20
21 # Dar formato ordinal a las variables Cilindrada y Cambios, renombrando
22 # sus niveles.
23 datos[["Cilindrada"]] <- factor(datos[["Cilindrada"]], levels = c(4, 6, 8),
24                               labels = c("4 cilindros", "6 cilindros",
25                                         "8 cilindros"),
26                               ordered = TRUE)
27
28 datos[["Cambios"]] <- factor(datos[["Cambios"]], levels = c(3, 4, 5),
29                               labels = c("3 cambios", "4 cambios", "5 cambios"),
30                               ordered = TRUE)
31
32 write.csv2(datos, "C:/Inferencia/Mtcars.csv")
```

1.3.5 Fórmulas

Si bien hasta ahora solo tenemos una definición preliminar de lo que es un modelo estadístico, necesitamos conocer una herramienta para representarlos en R, pues son una parte fundamental del funcionamiento de este lenguaje.

Para entender de manera sencilla qué es una fórmula, podemos simplemente decir que permite capturar una expresión no evaluada, y que está asociada a un ambiente. Su sintaxis básica tiene la forma **variable independiente ~ variables dependientes**, lo que nos indica, entonces, que las fórmulas representan una relación entre variables.

Tomemos una vez más el conjunto de datos **iris**. Podríamos representar la asociación entre la especie de **iris** (variable independiente) y las dimensiones de sus pétalos (variables dependientes) como **Species ~ Petal.Length + Petal.Width**.

Extenderemos las nociones acerca del uso de fórmulas a medida que avancemos en nuestro aprendizaje, pero si quieres aprender más puedes consultar Willems (2017).

1.4 EJERCICIOS PROPUESTOS

1. Una encuesta reciente preguntó: “después de la jornada laboral usual, ¿cuántas horas dedica a relajarse o a realizar actividades que disfruta?” a una muestra de 580 chilenas y 575 chilenos. Se encontró que el número promedio de horas era de $1,30 \pm 0,30$ y $1,95 \pm 0,25$ para cada grupo, respectivamente.
 - a) ¿Cómo sería una matriz de datos para este estudio? Muestra algunas filas de ella como ejemplos.
 - b) ¿Cuál podría ser la población objetivo?
 - c) ¿Qué se entendería por unidad de observación?
 - d) ¿Qué tipo de variable sería “el número de horas dedicadas a distraerse después de la jornada laboral usual” que respondió cada persona entrevistada?
 - e) ¿Existe alguna variable categórica? Si es así, ¿de qué tipo? ¿Con qué niveles?
 - f) ¿Qué dato(s) correspondería(n) a un estadístico?
 - g) ¿Cuál(es) sería(n) el(los) parámetro(s) en estudio?
 - h) ¿Logra el estudio establecer que ser mujer chilena ocasiona tener menos horas dedicadas a distraerse después de la jornada laboral usual?
2. Investiga para qué sirven y cómo se usan los argumentos `row.names` y `col.names` en las funciones para importar datos desde archivos y la función `data.frame()`.
3. Construye en R una matriz de datos para almacenar las características de una muestra de servidores. Considera a lo menos una variable categórica y una variable numérica.
4. Investiga qué función (o funciones) ofrece R para guardar una matriz de datos en un archivo y úsala(s) para guardar la matriz de datos del ejercicio anterior.
5. Resuelve en R los siguientes ejercicios. Considera para ello el conjunto de datos nativo de R `chickwts`.
 - a) ¿Cómo se puede cargar el conjunto de datos en la variable `pollos`?
 - b) ¿Cómo se ve la estructura de la matriz de datos almacenada en `pollos`?
6. Muestra ejemplos de las distintas transformaciones que se pueden hacer a una matriz de datos usando para ello conjunto de datos nativo de R `ChickWeight`.

CAPÍTULO 2. EXPLORACIÓN DE DATOS

Siempre es bueno que nos familiaricemos con los datos y algunas de sus características antes de empezar a trabajar con ellos. Esto nos ayuda a decidir qué herramientas son las más adecuadas para dar respuesta a las preguntas que queramos responder. En este capítulo revisaremos las principales estadísticas descriptivas que nos ayudarán a resumir los datos para entenderlos mejor, así como diversos tipos de gráficos que nos permitirán representar los datos de modo que podamos comprenderlos de forma visual. Para ello, tomamos como base los conceptos expuestos en Diez et al. (2017, pp. 26-50), Field et al. (2012, pp. 19-27) y STDHA (s.f.), fuentes que puedes consultar si deseas saber más acerca de estos temas.

Para muchos de los ejemplos de este capítulo usaremos el conjunto de datos `mtcars` con las modificaciones realizadas en el script 1.7, cuyo diccionario de datos se muestra en la tabla 2.1.

Variable	Descripción
Rendimiento	Rendimiento, en millas / galón (EEUU).
Cilindrada	Número de cilindros (4 cilindros, 6 cilindros, 8 cilindros).
Desplazamiento	Desplazamiento, en pulgadas cúbicas.
Potencia	Potencia, en caballos de fuerza brutos.
Eje	Razón del eje trasero.
Peso	Peso, en miles de libras.
Cuarto_milla	Tiempo que tarda en recorrer un cuarto de milla partiendo desde el reposo, en segundos.
Motor	Tipo de motor (V, Recto).
Transmision	Tipo de transmisión (Automático, Manual).
Cambios	Número de cambios hacia adelante (3 cambios, 4 cambios, 5 cambios).
Carburadores	Número de carburadores.

Tabla 2.1: descripción de las variables para el conjunto de datos `mtcars`.

2.1 ESTADÍSTICAS DESCRIPTIVAS

Las estadísticas descriptivas son medidas que nos permiten sintetizar y, como su nombre lo indica, describir los datos. Estas pueden aplicarse tanto a una muestra como a una población. Cuando una de estas medidas se aplica a la muestra, corresponde a un **estimador puntual** de la misma medida para la población. Al ser una estimación, no es exacta, aunque la precisión tiende a aumentar mientras mayor sea el tamaño de la muestra.

Un concepto importante a tener en cuenta es la noción de **distribución**. En este capítulo se considera la **distribución de frecuencia**, que representa cuántas veces aparece cada valor para una variable en un conjunto de datos.

2.1.1 Estadísticas descriptivas para datos numéricos

Una de las estadísticas descriptivas más empleadas es la **media**, conocida en otros contextos como media aritmética o promedio. Denotamos la **media muestral** por \bar{x} , donde x corresponde al nombre de la variable, mientras que para la **media poblacional** empleamos la notación μ_x . Esta medida se calcula como muestra la ecuación 2.1, donde x_i son los n valores observados de la variable. Podemos entender la media como el punto de equilibrio de la distribución (Diez et al., 2017, p. 28). Así, la media corresponde a una **medida de tendencia central**.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.1)$$

El script 2.1 muestra cómo usar la función `mean()` de R para calcular la media de diversas variables del conjunto de datos `mtcars`¹. Como primer ejemplo, se calcula la media de la variable `Rendimiento`. A continuación se muestra cómo realizar esta operación para dos variables, señaladas por el índice de sus respectivas columnas. Luego, de manera similar, se calculan las medias para cuatro columnas consecutivas de la matriz de datos. En estos dos casos hacemos uso de la función `sapply()`, que permite aplicar una misma función (cualquiera) para múltiples columnas.

Script 2.1: uso de las funciones `mean()` y `sapply()`.

```
1 # Cargar conjunto de datos.
2 datos <- read.csv2("C:/ Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
3                     row.names = 1)
4
5 # Calcular la media para la variable Rendimiento.
6 media <- mean(datos[["Rendimiento"]])
7 cat("Rendimiento medio:", media, "\n\n")
8
9 # Calcular la media para la tercera y quinta columnas
10 # (variables Desplazamiento y Eje).
11 cat("Medias\n")
12 print(sapply(datos[c(3, 5)], mean))
13 cat("\n")
14
15 # Calcular la media para las columnas 3 a 6
16 # (variables Desplazamiento, Potencia, Eje y Peso).
17 cat("Medias\n")
18 print(sapply(datos[3:6], mean))
19 cat("\n")
20
21 # Calcular la media para la variable Rendimiento omitiendo valores faltantes.
22 print(mean(datos[["Rendimiento"]], na.rm = TRUE))
```

La función `mean()` devuelve `NA` (*not available*, es decir, no disponible) si existen valores faltantes en los datos de entrada. Para prevenir este error, se puede proporcionar un argumento adicional que descarte los valores faltantes, como muestra la última línea del script 2.1.

Una medida de tendencia central alternativa a la media es la **mediana**, que es, simplemente, el valor central de los valores previamente ordenados. Cuando no existe un valor central, vale decir, cuando el tamaño de la muestra es par, la mediana está dada por el promedio simple de los dos valores centrales. En R, la mediana se calcula con la función `median()`.

¹Todas las demás funciones de R mencionadas en esta sección para las que no se proporcione un script se usan del mismo modo que `mean()`.

La **moda** es, simplemente, el valor más frecuente en el conjunto de datos. No obstante, tiene el problema de que puede haber múltiples modas. Dependiendo de la cantidad de modas, se habla de distribuciones **unimodales**, **bimodales** y **multimodales**.

Si bien R no cuenta con una función nativa para encontrar la moda, el paquete **modeest** ofrece la función **mfv()** que entrega el valor más frecuente de una variable. En caso de que dos (o más) valores sean los más frecuentes con igual cantidad de observaciones, los entrega todos en forma de vector.

Las medidas que hemos estudiado hasta ahora buscan describir el centro del conjunto de datos. No obstante, también es importante conocer su **variabilidad o dispersión**, pues así se puede saber qué tan semejantes (o diferentes) son las observaciones entre sí. Estas suelen calcularse en base a la **desviación** de las observaciones, que se entiende como la distancia entre una observación y la media del conjunto de datos. Las dos principales medidas de dispersión son la **varianza** y la **desviación estándar**, ambas basadas en los cuadrados de las distancias, ya que, por una parte, los valores grandes se incrementan más significativamente y, por otra, se opera solo con valores positivos, pues la dirección de la distancia no es de interés.

La varianza muestral se calcula como muestra la ecuación 2.2, donde x_i son los valores de cada una de las n observaciones. Cabe destacar que puede emplearse un subíndice para indicar el nombre de la variable, al igual que en el caso de la media.

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2.2)$$

La desviación estándar de la muestra se define como la raíz cuadrada de la varianza (2.3), medida que resulta de gran utilidad cuando se necesita saber cuán cercanos son los datos a la media, ya que se encuentra en la misma escala que la variable.

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.3)$$

Al igual que en el caso de la media, podemos usar las fórmulas anteriores para obtener estimaciones puntuales de la varianza y la desviación estándar de la población, denotadas por σ^2 y σ , respectivamente.

Es importante considerar que, si bien la media y la desviación estándar permiten conocer el centro y la dispersión del conjunto de datos, respectivamente, la distribución de los puntos puede ser muy diferente, como ilustra la figura 2.1.

Las funciones de R para calcular la varianza y la desviación estándar son, respectivamente, **var()** y **sd()**.

Aunque menos empleado, el **rango** muestra los valores extremos, es decir, el mínimo y el máximo, de una variable. R ofrece la función **range()** para obtener ambos valores, además de **min()** y **max()** para obtenerlos por separado.

En párrafos anteriores vimos que la mediana es el valor central (o el promedio de los dos valores centrales) del conjunto de datos ordenado, ya sea una población o una muestra. Esto significa, entonces, que esta medida divide el conjunto de datos en dos mitades con igual cantidad de elementos. De manera similar, es posible dividir el conjunto de datos en segmentos más pequeños, por ejemplo en 4, 10 o 100 partes con igual cantidad de elementos. Cada fragmento del conjunto de datos dividido de esta forma recibe el nombre de **cuantil**. Algunas subdivisiones de uso frecuente reciben nombres especiales:

- **Percentiles:** dividen el conjunto de datos en 100 subconjuntos de igual tamaño.
- **Deciles:** dividen el conjunto de datos en 10 subconjuntos de igual tamaño.
- **Quintiles:** dividen el conjunto de datos en 5 subconjuntos de igual tamaño.
- **Cuartiles:** dividen el conjunto de datos en 4 subconjuntos de igual tamaño.

Los cuantiles (al igual que las otras subdivisiones antes mencionadas) se nombran de forma ascendente según el sentido de crecimiento del conjunto de datos. Así, el percentil 1 contiene a los valores más pequeños,

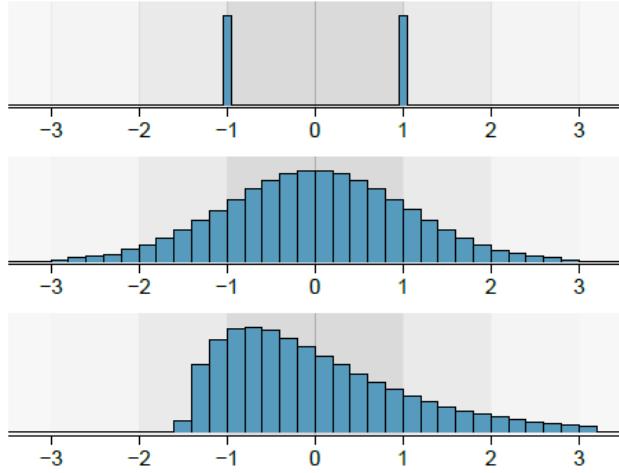


Figura 2.1: tres distribuciones de población muy distintas con media $\mu = 0$ y desviación estándar $\sigma = 1$.

Fuente: Diez et al. (2017, p. 34).

mientras que el percentil 100, a los más grandes. Cabe destacar que la mediana corresponde al percentil 50 o al cuartil 2, y que nombrar al decil 3 es equivalente al percentil 30.

R proporciona la función `quantile()` para calcular cuantiles, que por defecto calcula los cuartiles, aunque su uso puede generalizarse mediante el parámetro adicional `probs`, como muestra el script 2.2. La función `seq()` genera una secuencia de números equiespaciados, y recibe como argumentos el inicio, el término y el incremento de la secuencia.

Script 2.2: cálculo de cuantiles con la función `quantile()`.

```

1 # Cargar conjunto de datos.
2 datos <- read.csv("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
3                     row.names = 1)
4
5 # Cálculo de percentiles para la variable Rendimiento.
6 cat("Cuartiles:\n")
7 print(quantile(datos[["Rendimiento"]]))
8 cat("\n")
9
10 cat("Quintiles:\n")
11 print(quantile(datos[["Rendimiento"]], seq(0, 1, 0.2)))
12 cat("\n")
13
14 cat("Deciles:\n")
15 print(quantile(datos[["Rendimiento"]], seq(0, 1, 0.1)))
16 cat("\n")
17
18 cat("Percentiles:\n")
19 print(quantile(datos[["Rendimiento"]], seq(0, 1, 0.01)))

```

Ahora que conocemos los cuartiles, podemos introducir una nueva medida de variabilidad que usaremos a menudo, llamada **rango intercuartil** o IQR (por su sigla en inglés), dada por la ecuación 2.4, donde Q_1 y Q_3 corresponden a los cuartiles 1 y 3, respectivamente. Al igual que la varianza y la desviación estándar, mientras más disperso sea el conjunto de datos, mayor será el valor del IQR. En R, la función que calcula este estimador es `IQR()`.

$$IQR = Q_3 - Q_1 \quad (2.4)$$

Muchas veces los conjuntos de datos contienen lo que se conoce como **valores atípicos** o *outliers*. Estos corresponden a observaciones que parecen estar fuera de rango o ser muy extremos con respecto al resto de los datos. Medidas como la media o la desviación estándar son muy sensibles a los valores atípicos, por lo que son propensas a errores ante la presencia de este tipo de observaciones. Para reducir el efecto de los valores extremos muchas veces necesitaremos medidas **robustas**, que son aquellas que proporcionan una estimación confiable aún ante la presencia de valores atípicos. En este escenario, la mediana resulta ser una buena medida de tendencia central y el IQR, una buena medida de dispersión.

Nos encontraremos frecuentemente con la necesidad de calcular varias medidas de tendencia central y de dispersión descritas en el apartado anterior. Por esta razón, R, y algunos de sus paquetes, ofrecen algunas funciones que calculan varios de estos estadísticos con una sola llamada. Tal es el caso de la función nativa **summary()**, que entrega la media, la mediana, el primer y el tercer cuartil, el mínimo y el máximo. Otra función que nos puede ser de mucha ayuda es **summarise()**, del paquete **dplyr**. Con ella podemos calcular varias de las medidas en una sola llamada, como muestra el script 2.3.

Script 2.3: uso de la función **summarise()** del paquete **dplyr**.

```

1 library(dplyr)
2
3 datos <- read.csv("C:/ Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
4                     row.names = 1)
5
6 # Cálculo de varias medidas para la variable Potencia.
7 medidas_potencia <- datos %>% summarise(Media = mean(Potencia),
8                                              Mediana = median(Potencia),
9                                              Varianza = var(Potencia),
10                                             IQR = IQR(Potencia))
11
12 print(medidas_potencia)
13 cat("\n")
14
15 # Cálculo de la media y la desviación estándar para las variables Peso y
16 # Cuarto_milla.
17 medidas_varias <- datos %>% summarise(Media_P = mean(Peso),
18                                              Media_C = median(Cuarto_milla),
19                                              SD_P = sd(Peso),
20                                              SD_C = sd(Cuarto_milla))
21
22 print(medidas_varias)
23 cat("\n")

```

2.1.2 Estadísticas descriptivas para datos categóricos

Cuando queremos trabajar con datos categóricos, medidas como la media o la desviación estándar carecen de sentido. En consecuencia, necesitamos otros estadísticos para resumir el conjunto de datos.

Como primer estadístico para variables categóricas podemos mencionar la **frecuencia**, que corresponde a la cantidad de veces que podemos encontrar cada nivel de la variable en los datos. Otro estadístico importante corresponde a la **proporción**, que corresponde a la frecuencia relativa. En otras palabras, la proporción corresponde a frecuencia de un nivel de la variable dividida por la cantidad total de observaciones.

La mejor alternativa para este tipo de datos es la **tabla de contingencia**, también llamada **matriz de confusión** o **tabla de frecuencias**, donde cada fila representa la cantidad de veces en que ocurre una combinación de variables. También es posible usar porcentajes o proporciones en lugar de la cantidad de

ocurrencia, en cuyo caso se habla de una **tabla de frecuencias relativas**. La tabla 2.2 muestra la tabla de contingencia (de frecuencias) para la variable `Cambios`. Se puede observar, por ejemplo, que el conjunto de datos contiene una muestra de 32 automóviles y que 15 de ellos tienen tres cambios.

	3 cambios	4 cambios	5 cambios	Total
	15	12	5	32

Tabla 2.2: tabla de contingencia para la cantidad de cambios de los automóviles.

Desde luego, podemos construir tablas de contingencia de manera bastante sencilla en R. El script 2.4 muestra dos formas de obtener la tabla 2.2. La primera es la función `table()` y la segunda, la función `xtabs()`. El funcionamiento de ambas es equivalente, aunque `xtabs()` muestra el nombre de la variable tabulada al imprimir los resultados y `table()` no lo hace. Las tablas entregadas por estas funciones no incluyen los totales por filas, pero la función `marginSums()` permite calcularlos y mostrarlos como un vector. A su vez, la función `addmargins()` permite calcular dichos totales e incorporarlos a la tabla. Para terminar, el las últimas sentencias del script 2.4 ilustran la manera de obtener las tablas de frecuencias relativas con proporciones y porcentajes, respectivamente.

Podemos ver que las llamadas a `table()` y a `xtabs()` son algo diferentes. La primera recibe como argumento la columna de la matriz de datos, es decir, un vector con los datos a tabular, mientras que la segunda recibe una fórmula en que no existe una variable dependiente y la variable categórica es la independiente.

Script 2.4: tabla de contingencia para la variable `Cambios`.

```

1 # Cargar datos.
2 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
3                      row.names = 1)
4
5 # Crear tabla de contingencia para la variable gear.
6 contingencia <- table(datos[["Cambios"]])
7 cat("Tabla de contingencia generada con table():\n")
8 print(contingencia)
9 cat("\n")
10
11 # Otra forma de crear la misma tabla.
12 contingencia <- xtabs(~ Cambios, data = datos)
13 cat("Tabla de contingencia generada con xtabs():\n")
14 print(contingencia)
15 cat("\n")
16
17 # Calcular totales por fila y mostrarlos por separado.
18 totales <- marginSums(contingencia)
19 cat("Totales por fila:\n")
20 print(totales)
21 cat("\n")
22
23 # Calcular totales por fila y agregarlos a la tabla.
24 con_totales <- addmargins(contingencia, 1)
25 cat("Tabla de contingencia con totales por fila:\n")
26 print(con_totales)
27 cat("\n")
28
29 # Convertir a tabla de proporciones
30 proporciones <- prop.table(contingencia)
31 proporciones <- addmargins(proporciones, 1)
32 cat("Tabla de contingencia con proporciones:\n")
33 print(proporciones)
34 cat("\n")
35

```

```

36 # Convertir a tabla de porcentajes con 2 decimales.
37 porcentajes <- round(prop.table(contingencia), 4) * 100
38 porcentajes <- addmargins(porcentajes)
39 cat("Tabla de contingencia con porcentajes:\n")
40 print(porcentajes)
41 cat("\n")

```

También podemos construir matrices de confusión para dos variables categóricas, como muestra la tabla 2.3 para las variables **Cambios** y **Transmisión**.

		Cambios			Total
		3 cambios	4 cambios	5 cambios	
Transmision	Automático	15	4	0	19
	Manual	0	8	5	13
	Total	15	12	5	32

Tabla 2.3: tabla de contingencia para las variables **Cambios** y **Transmisión**.

En ocasiones resulta útil determinar las proporciones por fila o por columna, que podemos obtener dividiendo el valor de una celda de la matriz por el total de su fila o columna, según corresponda. Así, el total de cada fila (o columna) es igual a 1. Puesto que las proporciones por fila y por columna no son equivalentes, debemos ser cuidadosos al escoger la más adecuada en cada caso. Las tablas 2.4 a 2.6 muestran las proporciones por fila, por columna y generales para la matriz de confusión de la tabla 2.3. La construcción en R de la tabla de contingencia y las tablas de proporciones para dos variables se muestra en el script 2.5.

		Cambios			Total
		3 cambios	4 cambios	5 cambios	
Transmision	Automático	0,7894737	0,2105263	0,0000000	1,0000000
	Manual	0,0000000	0,6153846	0,3846154	1,0000000

Tabla 2.4: tabla de proporciones con totales por fila para la tabla 2.3.

		Cambios			Total
		3 cambios	4 cambios	5 cambios	
Transmision	Automático	1,0000000	0,3333333	0,0000000	1,0000000
	Manual	0,0000000	0,6666667	1,0000000	
	Total	1,0000000	0,0000000	1,0000000	

Tabla 2.5: tabla de proporciones con totales por columna para la tabla 2.3.

Script 2.5: tablas de contingencia y proporciones para dos variables.

```

1 # Cargar datos.
2 datos <- read.csv2("C:/ Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
3                      row.names = 1)
4
5 # Crear tabla de contingencia para las variables Transmision y gear.
6 contingencia <- table(datos[["Transmision"]], datos[["Cambios"]])
7 cat("Tabla de contingencia generada con table():\n")
8 print(contingencia)
9 cat("\n")
10
11 # Otra forma de crear la misma tabla.
12 contingencia <- xtabs(~ Transmision + Cambios, data = datos)
13 cat("Tabla de contingencia generada con xtabs():\n")
14 print(contingencia)
15 cat("\n")

```

		Cambios			
		3 cambios	4 cambios	5 cambios	Total
Transmision	Automático	0,46875	0,12500	0,00000	0,59375
	Manual	0,00000	0,25000	0,15625	0,40625
	Total	0,46875	0,37500	0,15625	1,00000

Tabla 2.6: tabla de proporciones con totales por fila y columna para la tabla 2.3.

```

16
17 # Proporciones con totales por fila.
18 proporciones_fila <- prop.table(contingencia, margin=1)
19 proporciones_fila <- addmargins(proporciones_fila, margin=2)
20 cat("Tabla de contingencia con proporciones totales por fila:\n")
21 print(proporciones_fila)
22 cat("\n")
23
24 # Proporciones con totales por columna.
25 proporciones_columna <- prop.table(contingencia, margin=2)
26 proporciones_columna <- addmargins(proporciones_columna, margin=1)
27 cat("Tabla de contingencia con proporciones totales por columna:\n")
28 print(proporciones_columna)
29 cat("\n")
30
31 # Proporciones con totales.
32 proporciones <- prop.table(contingencia)
33 proporciones <- addmargins(proporciones)
34 cat("Tabla de contingencia con proporciones totales:\n")
35 print(proporciones)
36 cat("\n")

```

Aunque no ocurre con frecuencia, podríamos necesitar una matriz de confusión para más de dos variables. Veamos ahora un ejemplo con tres variables: **Motor**, **Cambios** y **Transmisión**. Para ello, tomamos una de las variables (en este caso, **Motor**) y creamos una subtabla por cada uno de sus niveles. Cada subtabla muestra las frecuencias para la combinación de las dos variables restantes cuando **Motor** tiene el nivel correspondiente, como muestra la tabla 2.7. En R, podemos obtener estas tablas como muestra el script 2.6. Desde luego, esta misma idea puede extenderse para cuatro o más variables categóricas.

Motor = Recto

		Cambios			
		3 cambios	4 cambios	5 cambios	
Transmision	Automático	3	4	0	
	Manual	0	6	1	

Motor = V

		Cambios			
		3 cambios	4 cambios	5 cambios	
Transmision	Automático	12	0	0	
	Manual	0	2	4	

Tabla 2.7: tabla de contingencia para tres variables.

Script 2.6: matriz de confusión para tres variables.

```

1 # Cargar datos.
2 datos <- read.csv("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
3                     row.names = 1)
4

```

```

5 # Convertir la variable Cambios en categórica.
6 datos[["Cambios"]] <- factor(datos[["Cambios"]])
7
8 # Crear tabla de contingencia para las variables Transmision,
9 # Cambios y Motor.
10 contingencia <- ftable(datos[["Transmision"]], datos[["Cambios"]],
11                           datos[["Motor"]])
12
13 cat("Tabla de contingencia generada con ftable():\n")
14 print(contingencia)
15 cat("\n")
16
17 # Otra forma de crear la misma tabla.
18 xtabs(~ Cambios + Transmision + Motor, data = datos)
19 cat("Tabla de contingencia generada con xtabs():\n")
20 print(contingencia)
21 cat("\n")

```

2.1.3 Trabajando con datos agrupados

A menudo nos veremos en la necesidad de obtener estadísticas descriptivas de una variable separando las observaciones en grupos de acuerdo a una variable categórica. Para ello, el paquete `dplyr` ofrece la función `group_by()`, que podemos usar en conjunto con `summarise()`, como muestra el script 2.7. En dicho script, primero se agrupan las observaciones de acuerdo a la variable `Cambios`, y luego se efectúa una llamada a `summarise()` donde el primer argumento cuenta la cantidad de observaciones en el grupo actual y los argumentos restantes (que pueden ser tantos como se desee) corresponden a diferentes estadísticas descriptivas.

Script 2.7: estadísticas descriptivas para datos agrupados.

```

1 library(dplyr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                     row.names = 1)
6
7 resumen <- group_by(datos, Cambios) %>%
8   summarise(count = n(), mean(Rendimiento), median(Rendimiento),
9             sd(Rendimiento), IQR(Rendimiento), mean(Potencia))
10
11 print(resumen)

```

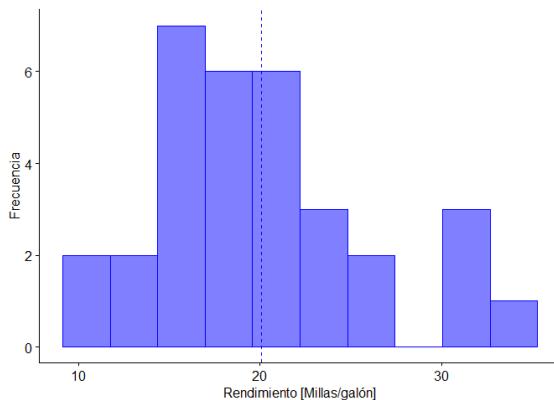
2.2 REPRESENTACIÓN GRÁFICA DE DATOS

En esta sección revisaremos diversos tipos de gráficos que resultan útiles al momento de estudiar un conjunto de datos disponibles, considerando su definición, su utilidad y cómo se construyen en R. Para crear gráficos en R usaremos el paquete `ggpubr`. Algunos de los principales parámetros que usaremos para crear y editar gráficos con este paquete son:

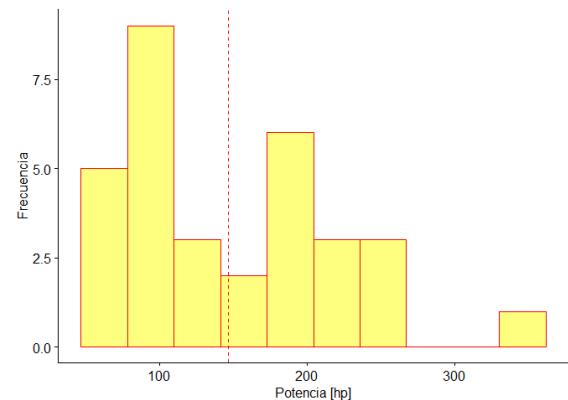
- **data**: un data frame.
- **x**: string con el nombre de la variable **x**.
- **y**: string(s) con el(los) nombre(s) de la(s) variable(s) a graficar.
- **color**: color de delineado.
- **fill**: color de relleno.
- **palette**: paleta de colores cuando existen múltiples grupos.
- **linetype**: tipo de línea a emplear.
- **add**: permite agregar elementos adicionales al gráfico, como barras de error o la media, entre otros.
- **title**: título del gráfico.
- **xlab**: rótulo del eje x. Puede ocultarse usando **xlab = FALSE**.
- **ylab**: rótulo del eje y. Puede ocultarse usando **ylab = FALSE**.

2.2.1 Una variable numérica

El **histograma** resulta muy útil si queremos representar una única variable numérica y la muestra es grande. Podemos decir que este gráfico muestra una aproximación a la **densidad** (o distribución de frecuencias) para la variable, para lo que tenemos que dividir el rango de valores posibles en intervalos (generalmente iguales) y luego contar la cantidad de observaciones en cada intervalo. Para construir el gráfico, creamos una barra por cada intervalo, cuya altura (o longitud) es proporcional a la cantidad de observaciones en el intervalo representado. La figura 2.2 muestra histogramas creados con el script 2.8.



(a) Distribución cercana a la simétrica.



(b) Distribución desviada a la izquierda.

Figura 2.2: dos histogramas.

Script 2.8: histogramas para las variables Rendimiento y Potencia.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                     row.names = 1)
6
7 # Histograma para la variable Rendimiento.
8 g1 <- gghistogram(datos,
9                     x = "Rendimiento",
10                    bins = 10,
11                    add = "mean",

```

```

12      xlab = "Rendimiento [Millas/galón]",
13      ylab = "Frecuencia",
14      color = "blue",
15      fill = "blue")
16
17 print(g1)
18
19 # Histograma para la variable Potencia.
20 g2 <- gghistogram(datos,
21                     x = "Potencia",
22                     bins = 10,
23                     add = "mean",
24                     xlab = "Potencia [hp]",
25                     ylab = "Frecuencia",
26                     color = "red",
27                     fill = "yellow")
28
29 print(g2)

```

A medida que avancemos en este libro, veremos que es muy importante conocer la **distribución de frecuencias** de una variable. Al observar la figura 2.2b, podemos ver que la frecuencia es mayor para potencias más bajas, pues las barras de la izquierda del gráfico son, en general, algo más altas que las de la derecha. Podría decirse que las observaciones se concentran a la izquierda y que hay una cola que se prolonga hacia la derecha. Cuando esto ocurre, decimos que la distribución está **desviada a la izquierda**, o que hay **asimetría negativa**. Análogamente, podría darse que la distribución estuviese desviada a la derecha o, equivalentemente, que presenta asimetría positiva. En el caso de la figura 2.2a, el histograma es más **simétrico**, pues las observaciones se aglomeran hacia el centro y hay colas tanto a la izquierda como a la derecha. Para ilustrar mejor la idea de la simetría, podemos revisar una vez más la figura 2.1, donde la población central es perfectamente simétrica y la inferior presenta asimetría positiva.

Otra ventaja de los histogramas es que permiten identificar modas de una variable, las cuales corresponden a barras que sean más prominentes que las de su entorno. Ambos ejemplos de la figura 2.2 son bimodales, pues tienen dos modas claramente identificables. Si bien es cierto que en ambos casos hay un único valor más frecuente (moda), podemos ver apreciar que existen dos “cumbres” o máximos locales.

Otro gráfico que usaremos a menudo es el de **gráfico de caja**. Es muy útil, pues su construcción considera 5 estadísticos para representar el conjunto de datos y además facilita la identificación de datos atípicos. La figura 2.3 muestra este gráfico para la variable **Potencia**, el cual fue creado con el script 2.9.

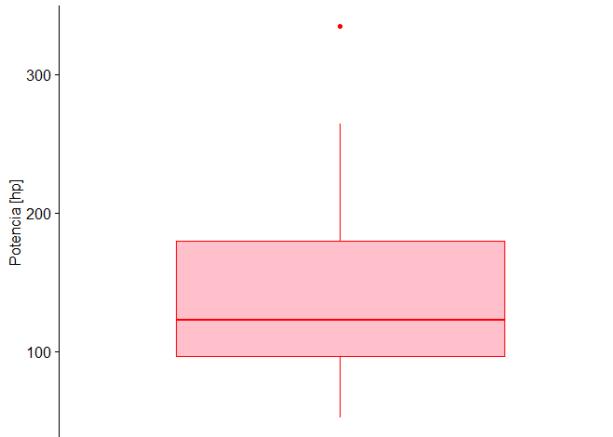


Figura 2.3: gráfico de caja para la variable **Potencia**.

Los extremos inferior y superior del rectángulo o caja de la figura 2.3 corresponden, respectivamente, al

primer y al tercer cuartil, mientras que la línea horizontal al interior de la caja denota la mediana. Así, la caja engloba el 50 % central de los datos, y su altura corresponde al rango intercuartil. Las barras que se extienden por sobre y por debajo de la caja, llamadas bigotes, capturan aquellos datos fuera de la caja central y que estén situados a no más de 1,5 veces el IQR. Cualquier observación que esté más allá de la caja y los bigotes se representa como un punto, el cual podría tratarse de una observación atípica.

Script 2.9: gráfico de caja para la variable Potencia.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/ Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                      row.names = 1)
6
7 g <- ggboxplot(datos[["Potencia"]],
8                  color = "red",
9                  fill = "pink",
10                 ylab = "Potencia [hp]")
11
12 g <- g + rremove("x.ticks")
13 g <- g + rremove("x.text")
14 g <- g + rremove("x.title")
15
16 print(g)

```

2.2.2 Una variable categórica

Si queremos representar una única variable categórica, lo más adecuado es usar un **gráfico de barras**, pues cada barra es tan larga como la proporción de valores presentes en cada nivel de la variable. La figura 2.4 muestra el gráfico de barras correspondiente a la tabla 2.2, elaborado mediante el script 2.10.

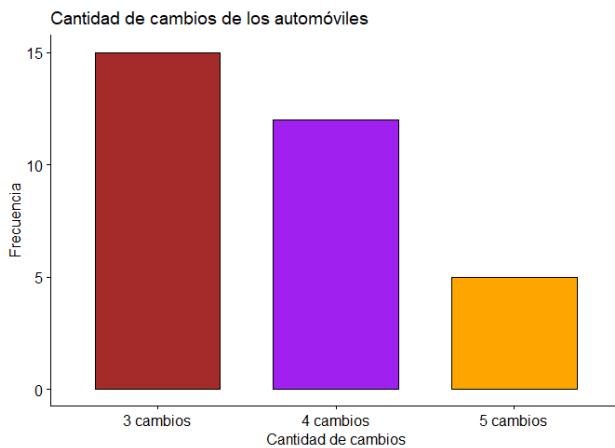


Figura 2.4: gráfico de barras para la variable Cambios.

Script 2.10: gráfico de barras para la variable Cambios.

```

1 library(ggpubr)
2

```

```

3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                      row.names = 1)
6
7 # Crear la tabla de frecuencias para la variable Cambios y convertirla a
8 # data frame.
9 contingencia <- as.data.frame(xtabs(~ Cambios, data = datos))
10
11 # Crear el gráfico de barras.
12 g <- ggbarplot(contingencia,
13                  x = "Cambios",
14                  y = "Freq",
15                  fill = c("brown", "purple", "orange"),
16                  title = "Cantidad de cambios de los automóviles",
17                  xlab = "Cantidad de cambios",
18                  ylab = "Frecuencia")
19
20 print(g)

```

Otra alternativa para representar una única variable categórica es el **gráfico de torta**, que se presenta en la figura 2.5 y se construye en R como muestra el script 2.11.

Cantidad de cambios de los automóviles

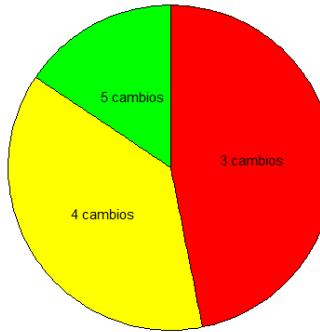


Figura 2.5: gráfico de torta para la variable Cambios.

Script 2.11: gráfico de torta para la variable Cambios.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                      row.names = 1)
6
7 # Crear la tabla de frecuencias y convertirla a data frame.
8 contingencia <- as.data.frame(xtabs(~ Cambios, data = datos))
9
10 # Crear gráfico de torta.
11 g <- ggpie(contingencia,
12              x = "Freq",
13              label = "Cambios",
14              fill = c("red", "yellow", "green"),
15              title = "Cantidad de cambios de los automóviles",
16              lab.pos = "in")
17

```

```
18 print(g)
```

2.2.3 Dos variables numéricas

Los **gráficos de dispersión** son adecuados en este caso. Se caracterizan porque muestran información caso a caso, ya que cada punto del gráfico corresponde a una observación. Por ejemplo, el gráfico de la figura 2.6, creado mediante el script 2.12, muestra este tipo de gráfico para las variables **Rendimiento** y **Peso**.

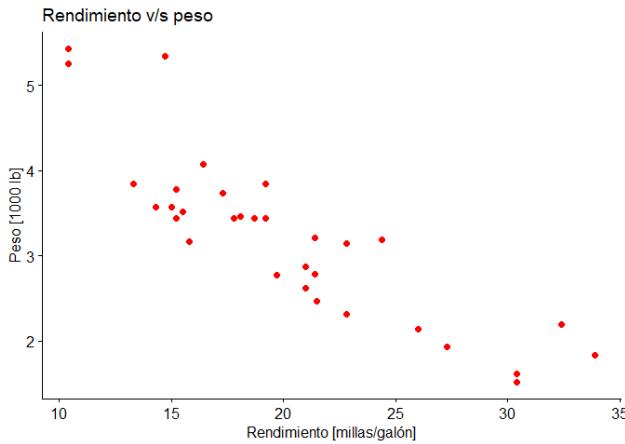


Figura 2.6: gráfico de dispersión para las variables **Rendimiento** y **Peso**.

Script 2.12: gráfico de dispersión para las variables **Rendimiento** y **Peso**.

```
1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/ Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5   row.names = 1)
6
7 # Crear gráfico de dispersión.
8 g <- ggscatter(datos,
9   x = "Rendimiento",
10  y = "Peso",
11  color = "red",
12  title = "Rendimiento v/s peso",
13  xlab = "Rendimiento [millas/galón]",
14  ylab = "Peso [1000 lb]")
15
16 print(g)
```

Los gráficos de dispersión también son muy útiles para identificar si dos (o más) variables están relacionadas. La figura 2.7 (creada mediante el script 2.13) muestra tres gráficos de dispersión diferentes: en el de la izquierda, se aprecia que las variables **Peso** y **Cuarto_milla** son independientes, pues no hay una tendencia definida en la organización de los puntos. En el gráfico del centro, en cambio, podemos ver que la potencia tiende a aumentar a medida que también lo hace el peso, por lo que ambas variables están positivamente asociadas. Por último, el gráfico de la derecha nos muestra que las variables **Peso** y **Rendimiento** presentan asociación negativa, puesto que a medida que la primera aumenta, la segunda disminuye.

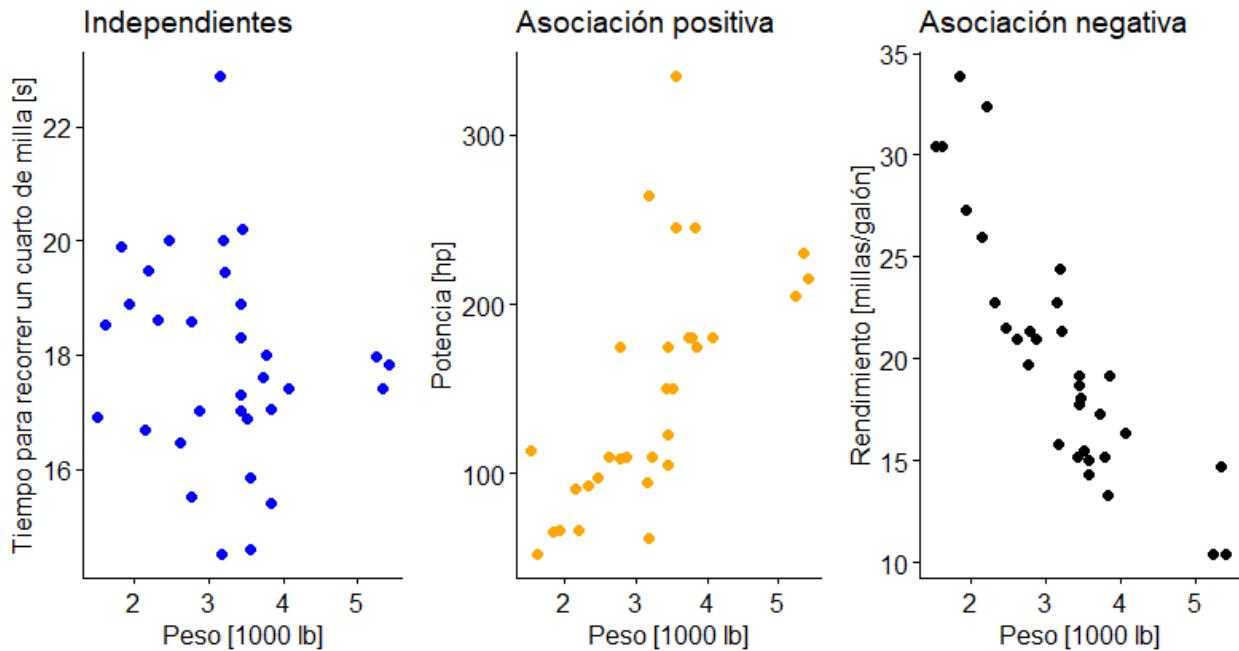


Figura 2.7: gráficos de dispersión con diferentes tipos de asociación entre las variables.

Script 2.13: gráficos de dispersión con diferentes tipos de asociación entre las variables.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                   row.names = 1)
6
7 # Gráfico para variables independientes.
8 g1 <- ggscatter(datos,
9                   x = "Peso",
10                  y = "Cuarto_milla",
11                  color = "blue",
12                  title = "Independientes",
13                  xlab = "Peso [1000 lb]",
14                  ylab = "Tiempo para recorrer un cuarto de milla [s]")
15
16 # Gráfico para variables con asociación positiva.
17 g2 <- ggscatter(datos,
18                   x = "Peso",
19                   y = "Potencia",
20                   color = "orange",
21                   title = "Asociación positiva",
22                   xlab = "Peso [1000 lb]",
23                   ylab = "Potencia [hp]")
24
25 # Gráfico para variables con asociación negativa.
26 g3 <- ggscatter(datos,
27                   x = "Peso",
28                   y = "Rendimiento",
29                   color = "black",
30                   title = "Asociación negativa",
31                   xlab = "Peso [1000 lb]",

```

```

32      ylab = "Rendimiento [millas/galón]")
33
34 # Crear figura con tres gráficos.
35 g <- ggarrange(g1 ,g2 ,g3, ncol = 3, nrow = 1, common.legend = TRUE)
36
37 print(g)

```

2.2.4 Dos variables categóricas

Similares al gráfico de barras para una variable categórica, los **gráficos de barras apiladas, agrupadas y estandarizadas** permiten visualizar la matriz de confusión entre dos variables y encontrar posibles relaciones entre ellas. La figura 2.8, creada con el script 2.14, ejemplifica esta familia de gráficos usando para ello las variables **Cambios** y **Motor**.

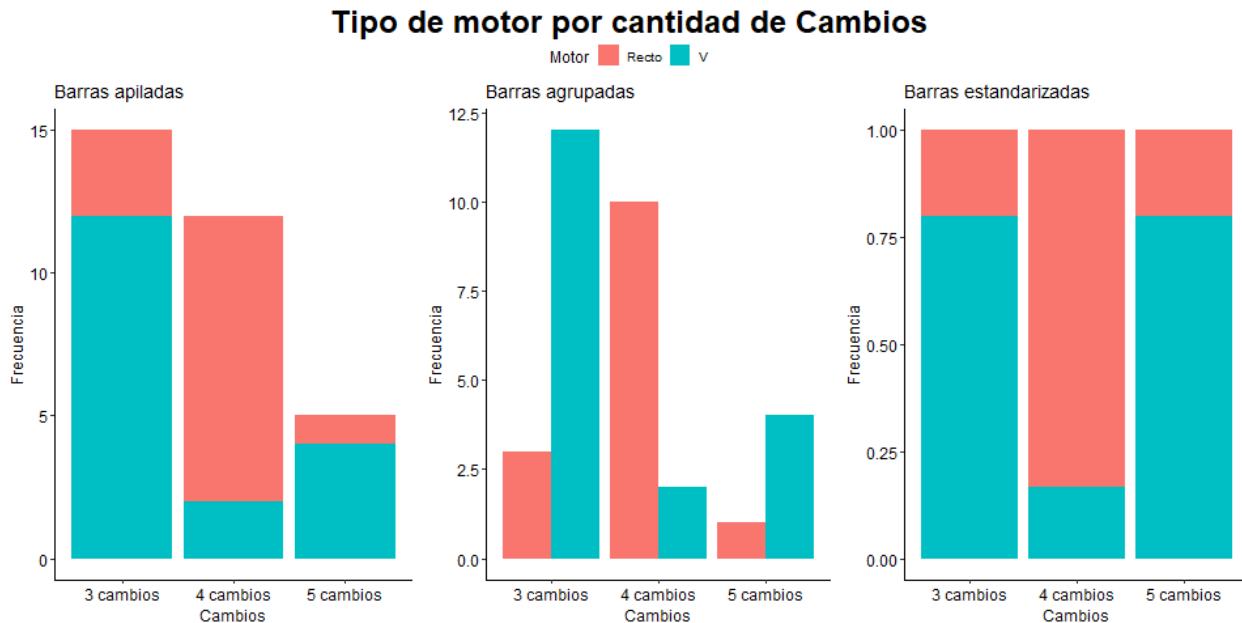


Figura 2.8: gráficos de barras para las variables **Cambios** y **Motor**.

El **gráfico de barras apiladas**, a la izquierda en la figura 2.8, muestra tres barras cuya altura corresponde a la frecuencia de la cantidad de cambios, al igual que en la figura 2.4, pero ahora cada barra está subdividida en secciones de distinto color para cada tipo de motor. La altura de cada sección está dada por la frecuencia del tipo de motor para la cantidad de cambios representada en la barra.

Similar al anterior, el gráfico de la derecha en la figura 2.8, que corresponde a un **gráfico de barras estandarizadas**, muestra barras de igual altura para cada cantidad de cambios representando claramente los cambios en la proporción de cada tipo de motor por la cantidad de cambios. Se puede apreciar que los automóviles con 3 y 5 cambios tienen mayoritariamente motores en forma de V, ambas en igual proporción, mientras que el uso de motores rectos se da principalmente en automóviles de 4 cambios.

El **gráfico de barras agrupadas**, al centro en la figura 2.8, es equivalente al de la izquierda, pero en lugar de dividir una barra en segmentos, muestra barras contiguas para cada tipo de motor.

Script 2.14: gráficos de barras para las variables **Cambios** y **Motor**.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/ Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                      row.names = 1)
6
7 # Crear tabla de contingencia para las variables Motor y Cambios,
8 # y guardarla como data frame.
9 tabla <- xtabs(~ Motor + Cambios, data = datos)
10 contingencia <- as.data.frame(tabla)
11
12 # Crear gráfico de barras segmentadas.
13 g1 <- ggplot(contingencia, aes(fill = Motor, y = Freq, x = Cambios))
14 g1 <- g1 + geom_bar(position = "stack", stat = "identity")
15 g1 <- g1 + labs(y = "Frecuencia") + ggtitle("Barras apiladas")
16 g1 <- g1 + theme_pubr()
17
18 # Crear gráfico de barras agrupadas.
19 g2 <- ggplot(contingencia, aes(fill = Motor, y = Freq, x = Cambios))
20 g2 <- g2 + geom_bar(position = "dodge", stat = "identity")
21 g2 <- g2 + labs(y = "Frecuencia") + ggtitle("Barras agrupadas")
22 g2 <- g2 + theme_pubr()
23
24 # # Crear gráfico de barras segmentadas estandarizado.
25 g3 <- ggplot(contingencia, aes(fill = Motor, y = Freq, x = Cambios))
26 g3 <- g3 + geom_bar(position = "fill", stat = "identity")
27 g3 <- g3 + labs(y = "Frecuencia") + ggtitle("Barras estandarizadas")
28 g3 <- g3 + theme_pubr()
29
30 # Crear una figura que contenga los tres gráficos.
31 g <- ggarrange(g1, g2, g3, nrow = 1, common.legend = TRUE)
32
33 # Agregar un título común en negrita y con fuente de 24 puntos.
34 titulo <- text_grob("Tipo de motor por cantidad de Cambios",
35                      face = "bold", size = 24)
36
37 g <- annotate_figure(g, top = titulo)
38
39 # Guardar la figura en formato png con tamaño 960 x 480 pixeles.
40 ggsave(g, filename = "C:/ Inferencia/f-barras-2.png",
41         height = 480, width = 960)

```

Similar al gráfico de barras para dos variables, el **gráfico de mosaico** permite representar una tabla de contingencia. Para ello, divide un área en regiones y el área de cada región es proporcional al porcentaje de observaciones que representa. La figura 2.9, creada con el script 2.15 ejemplifica el uso de este tipo de gráficos, usando para ello las variables **Cambios** y **Motor**. En ella, el ancho de cada columna es proporcional a la cantidad de automóviles que tienen la correspondiente cantidad de cambios, mientras que la altura de cada barra de las columnas refleja la proporción de automóviles con un determinado tipo de motor.

Si nos fijamos bien en la figura 2.9, podemos ver claramente que los vehículos con 5 cambios son, por mucho, los menos frecuentes y que los con 3 cambios son algo más frecuentes que los que tienen 4 cambios. Del mismo modo, podemos ver que, para vehículos de 3 y 5 cambios, la proporción de vehículos con motor recto es la misma, y mucho menor que la de aquellos con motor en forma de V. Sin embargo, este último no es muy frecuente en automóviles con 4 cambios.

Cabe destacar que, para este tipo de gráfico, se requiere emplear el paquete **ggmosaic**.

Script 2.15: gráfico de mosaico para las variables **Cambios** y **Motor**.

```

1 library(ggmosaic)

```

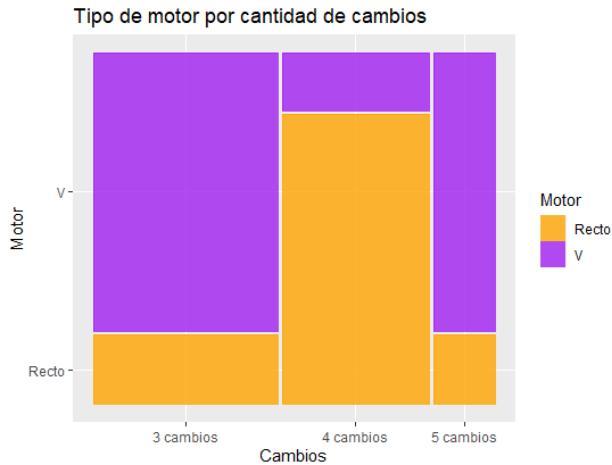


Figura 2.9: gráfico de mosaico para las variables Cambios y Motor.

```

2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5   row.names = 1)
6
7 # Crear tabla de contingencia para las variables gear y vs,
8 # y guardarla como data frame.
9 tabla <- xtabs(~ Cambios + Motor, data = datos)
10 contingencia <- as.data.frame(tabla)
11
12 # Crear gráfico de mosaico.
13 g <- ggplot(data = contingencia)
14 g <- g + geom_mosaic(aes(weight = Freq, x = product(Cambios), fill = Motor))
15
16 g <- g + labs(y = "Motor", x = "Cambios",
17   title = "Tipo de motor por cantidad de cambios")
18
19 g <- g + scale_fill_manual(values=c("orange", "purple"))
20
21 print(g)

```

2.2.5 Una variable numérica y otra categórica

Desde luego, también es importante poder comparar diferentes grupos de observaciones de acuerdo a una característica categórica, para lo cual los gráficos pueden ser de gran ayuda. Por ejemplo, la figura 2.10, creada mediante el script 2.16 muestra un gráfico de cajas para la variable Rendimiento agrupada por el número de cambios de los automóviles.

Script 2.16: gráfico de cajas por grupo.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,

```

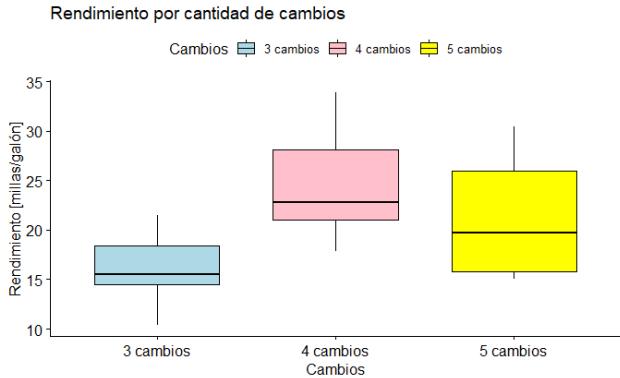


Figura 2.10: gráfico de cajas por grupo.

```

5           row.names = 1)
6
7 g <- ggboxplot(datos, x = "Cambios",
8                 y = "Rendimiento",
9                 palette = c("light blue", "pink", "yellow"),
10                fill = "Cambios",
11                title = "Rendimiento por cantidad de cambios",
12                xlab = "Cambios",
13                ylab = "Rendimiento [millas/galón]")
14
15 print(g)

```

Una buena alternativa, si la cantidad de observaciones es pequeña, es el **gráfico de tiras**, similar al gráfico de dispersión. El script 2.17 construye este gráfico para la variable **Rendimiento** agrupada según los niveles de la variable **Cambios**, obteniéndose como resultado la figura 2.11.

Script 2.17: gráfico de tiras.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                      row.names = 1)
6
7 g <- ggstripchart(datos, x = "Cambios",
8                     y = "Rendimiento",
9                     palette = c("blue", "red", "dark green"),
10                    color = "Cambios",
11                    title = "Rendimiento por cantidad de cambios",
12                    xlab = "Cambios",
13                    ylab = "Rendimiento [millas/galón]")
14
15 print(g)

```

2.3 EJERCICIOS PROPUESTOS

1. ¿Cuándo es apropiado utilizar un gráfico de puntos para revisar datos?

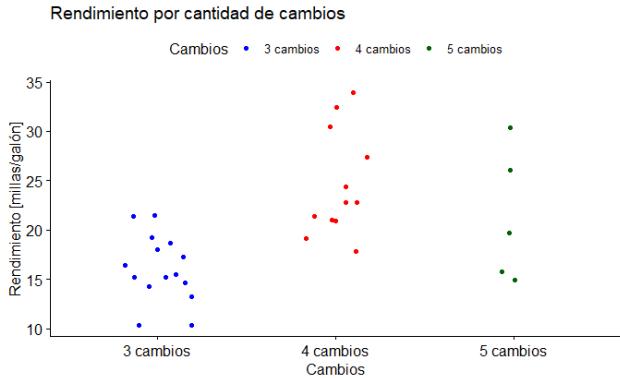
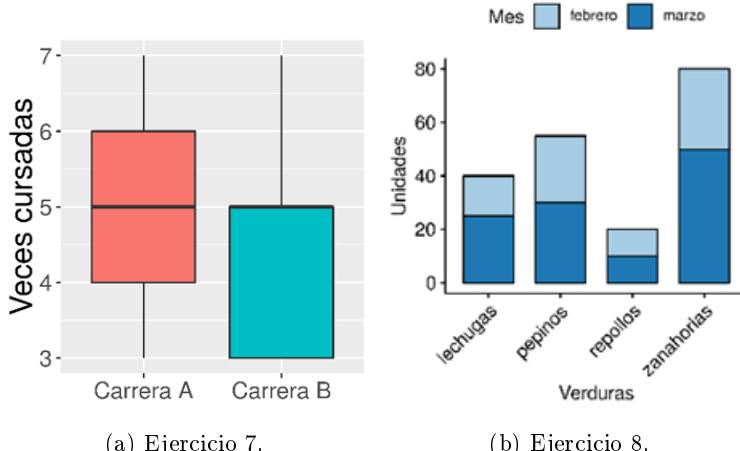


Figura 2.11: gráfico de tiras.

2. ¿Cuándo la mediana caracteriza mejor a un conjunto de datos que la media?
3. Da ejemplos de tres variables que posiblemente tengan una distribución simétrica, con asimetría positiva y con asimetría negativa, respectivamente. Justifique bien cada caso.
4. Describe un estudio en que posiblemente los datos recolectados tengan una distribución bimodal.
5. ¿Qué tipo de información buscada llevaría a utilizar un gráfico de dispersión?
6. ¿Por qué es importante conocer una medida de dispersión (variabilidad) de un conjunto de datos? Dé un ejemplo para clarificar su respuesta.
7. Considera la representación de la figura 2.12a de los datos obtenidos al tomar muestras aleatorias de estudiantes de dos carreras de la Facultad de Ingeniería para estudiar si el número de veces que se cursan las tres asignaturas de física en el Módulo Básico de Ingeniería depende de la carrera de los alumnos. Compara las distribuciones de ambos grupos. ¿En qué se parecen y en qué se diferencian?



(a) Ejercicio 7.

(b) Ejercicio 8.

Figura 2.12: gráficos para los ejercicios propuestos.

8. El gráfico de la figura 2.12b muestra las unidades de verduras vendidas en uno de los kioscos de la Universidad durante los meses anteriores. Construye la tabla de contingencia correspondiente a los datos que se representan. ¿Qué mes tuvo mayores ventas? ¿En qué proporción?
9. ¿Cómo se puede generar la secuencia 0.00, 0.25, 0.50, ..., 2.75, 3.00 en R?
10. Resuelve en R los siguientes ejercicios. Considera para ello el conjunto de datos nativo de R `chickwts`.
 - a) ¿Qué son los cuartiles y cómo se pueden obtener para los pesos de los pollitos reportados en la columna `weight`?
 - b) ¿Cómo obtener los cuartiles del ejercicio anterior por cada tipo de alimento en la columna `feed`?
 - c) ¿Cómo obtener un histograma de los pesos de los pollitos?

- d)* ¿Cómo se obtiene un gráfico de cajas para comparar los pesos de los pollitos por tipo de alimento suministrado?
11. Investiga acerca del uso del paquete de R `ggplot2` para la creación de gráficos.

CAPÍTULO 3. VARIABLES ALEATORIAS Y DISTRIBUCIONES DE PROBABILIDAD

Los conceptos que estudiaremos en este capítulo pueden resultar algo difíciles de entender, por lo que si necesitas más material, puedes consultar las fuentes en que se basa este capítulo: Diez et al. (2017, pp. 104-157) y Freund y Wilson (2003, pp. 104-106).

Definimos como **variable aleatoria** una variable o un proceso cuyo resultado sea numérico. Dichas variables se nombran con letras mayúsculas, y denotamos sus posibles valores por la letra minúscula correspondiente, acompañada de un subíndice. Las variables aleatorias tienen una **distribución de probabilidad**, la cual define la probabilidad de que ocurran los diferentes valores que dicha variable puede tomar.

3.1 VARIABLES ALEATORIAS

La definición de **variable aleatoria continua** es en realidad bastante sencilla: es una variable que puede tomar cualquiera de los infinitos valores posibles dentro de un intervalo.

Una **variable aleatoria discreta**, en cambio, solo puede tomar un conjunto finito de valores. Un ejemplo típico de variable aleatoria puede ser el lanzamiento de un dado. Si el dado está bien balanceado, tendremos igual probabilidad de obtener cualquiera de las caras. Pero es sabido que algunos trampas fabrican dados adulterados para favorecer, por ejemplo, la obtención de valores 1 y 6. Una distribución aleatoria de la variable lanzamiento de un dado adulterado (X) podría ser la que se presenta en la tabla 3.1.

i	1	2	3	4	5	6	Total
x_i	1	2	3	4	5	6	-
$P(X = x_i)$	0.250	0.125	0.125	0.125	0.125	0.250	1.000

Tabla 3.1: distribución de probabilidad para el lanzamiento de un dado adulterado.

El **valor esperado**, denotado como $E(X)$ o μ , corresponde al resultado promedio de una variable aleatoria. Para una variable aleatoria discreta, se calcula sumando los valores posibles ponderados por su probabilidad, como muestra la ecuación 3.1.

$$E(X) \equiv \mu = \sum_{i=1}^n x_i P(X = x_i) \quad (3.1)$$

También podemos calcular qué tan alejado podría estar el valor obtenido del valor esperado por medio de la varianza general, denotada por $Var(X)$ o σ^2 , que se calcula como la media de los cuadrados de la diferencia con respecto a la media ponderada según la probabilidad de ocurrencia, como muestra la ecuación 3.2. Una vez más, la desviación estándar corresponde a la raíz cuadrada de la varianza.

$$Var(X) \equiv \sigma^2 = \sum_{i=1}^n (x_i - \mu)^2 P(X = x_i) \quad (3.2)$$

En R, el paquete **DiscreteRV** permite trabajar con variables aleatorias discretas, como se ejemplifica en el script 3.1 (Cross, 2017).

Script 3.1: variables aleatorias discretas en R.

```

1 library(discreteRV)
2
3 # Crear una variable discreta para representar el dado
4 # adulterado de la tabla 3.1.
5 resultados <- 1:6
6 probabilidad = c(0.25, 0.125, 0.125, 0.125, 0.125, 0.25)
7 X <- RV(outcomes = resultados, probs = probabilidad)
8
9 # Calcular el valor esperado.
10 esperado <- E(X)
11 cat("Valor esperado:", esperado, "\n")
12
13 # Calcular la varianza.
14 varianza <- V(X)
15 cat("Varianza:", varianza, "\n")
16
17 # Calcular la desviación estándar.
18 desviacion <- SD(X)
19 cat("Desviación estándar:", desviacion, "\n")

```

Para ayudarnos a entender mejor la noción de distribución de probabilidad, veamos la figura 3.1 (obtenida mediante el script 3.2). Ella nos muestra, de izquierda a derecha, las distribuciones de probabilidad para el puntaje total obtenido al lanzar 5, 10 y 20 dados, respectivamente.

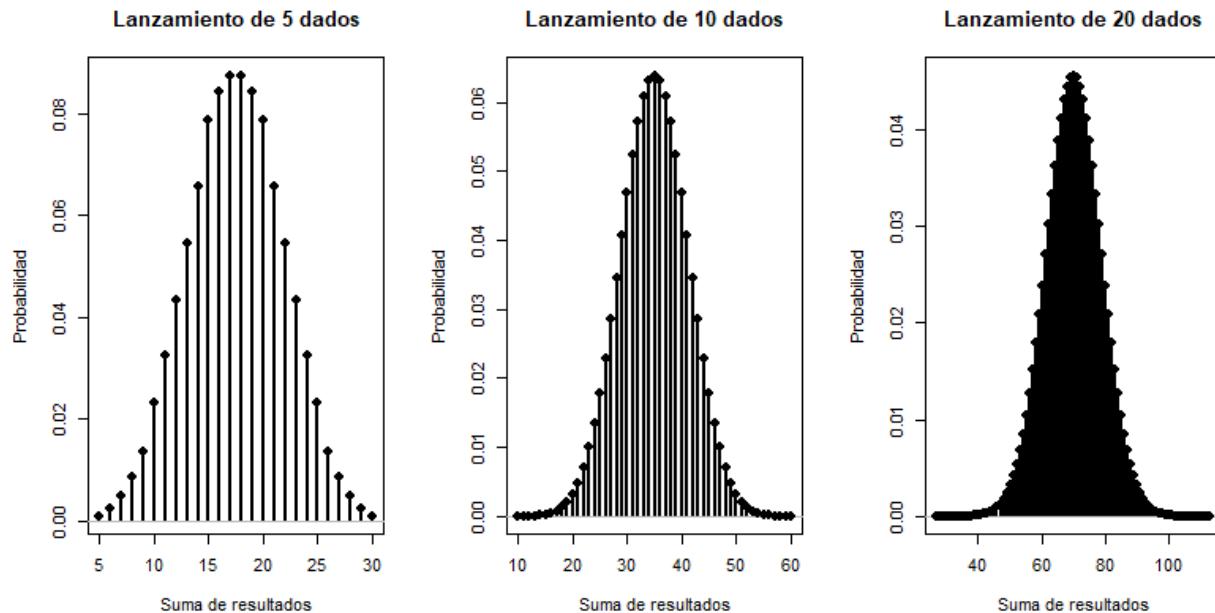


Figura 3.1: distribución de probabilidad para varios lanzamientos de un dado cargado.

Script 3.2: histogramas de variables aleatorias discretas en R.

```

1 library(discreteRV)
2 library(ggpubr)
3
4 # Crear una variable discreta para representar el dado
5 # adulterado de la tabla 4.1.
6 resultados <- 1:6

```

```

7  probabilidades = c(0.25, 0.125, 0.125, 0.125, 0.125, 0.25)
8  X <- RV(outcomes = resultados, probs = probabilidades)
9
10 # Crear vector con los resultados de 5 lanzamientos del dado.
11 lanzar_5 <- SofIID(X, n=5)
12
13 # Crear vector con los resultados de 10 lanzamientos del dado.
14 lanzar_10 <- SofIID(X, n=10)
15
16 # Crear vector con los resultados de 20 lanzamientos del dado.
17 lanzar_20 <- SofIID(X, n=20)
18
19 # Graficar los resultados.
20 par(mfrow=c(1, 3))
21
22 plot(lanzar_5,
23       main="Lanzamiento de 5 dados",
24       xlab="Suma de resultados",
25       ylab="Probabilidad")
26
27 plot(lanzar_10,
28       main="Lanzamiento de 10 dados",
29       xlab="Suma de resultados",
30       ylab="Probabilidad")
31
32 plot(lanzar_20,
33       main="Lanzamiento de 20 dados",
34       xlab="Suma de resultados",
35       ylab="Probabilidad")

```

Conocer la distribución de probabilidad de una variable discreta nos ayuda a hacer estimaciones útiles. A modo de ejemplo, supongamos que un ingeniero de software debe crear un programa que resuelva un problema (siempre con instancias del mismo tamaño) con un tiempo de respuesta no mayor a 25 segundos. El histograma de la figura 3.2 muestra los tiempos de ejecución obtenidos para 500 pruebas de la solución propuesta, donde se observa que 30 de ellas tardaron en realidad más de 25 segundos, con un rango que va de 0 a 30 segundos. Así, podemos estimar la probabilidad de que el tiempo de ejecución sea mayor a 25 segundos dividiendo la cantidad de observaciones que cumplen este criterio por la cantidad total de instancias, como muestra la ecuación 3.3.

$$P(X > 25) = \frac{30}{500} = 0.06 \quad (3.3)$$

Frecuentemente resulta más adecuado expresar o modelar un fenómeno como una combinación de dos o más variables aleatorias. Por ejemplo, un jugador de baloncesto puede anotar canastas de uno, dos o tres puntos dependiendo de si anota con un tiro libre, un lanzamiento desde dentro del área o desde fuera del área. Así, se tienen tres variables aleatorias:

1. X : Anotaciones por tiro libre.
2. Y : Anotaciones desde dentro del área.
3. Z : Anotaciones desde fuera del área.

Podemos representar el total de puntos anotados por el jugador como la suma de los puntos anotados de las tres formas posibles, lo que corresponde a una **combinación lineal** de las variables X , Y y Z . La fórmula general de una combinación lineal de n variables está dada por la ecuación 3.4, donde cada X_i corresponde a una variable aleatoria y cada c_i es una constante conocida.

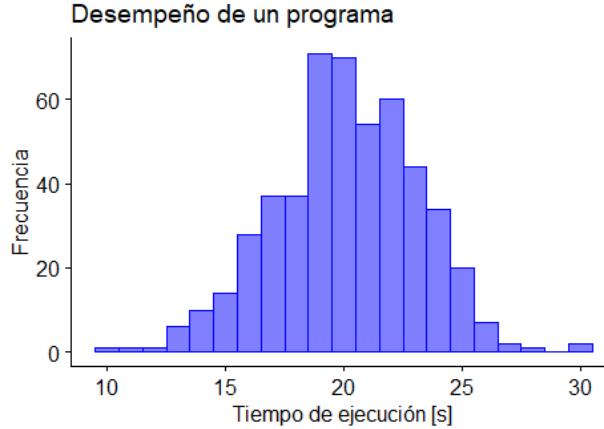


Figura 3.2: histograma para el desempeño del programa.

$$\sum_{i=1}^n c_i X_i \quad (3.4)$$

Cuando las variables de una combinación lineal son independientes¹, podemos calcular el valor esperado y la varianza de la combinación lineal usando las ecuaciones 3.5 y 3.6. Una vez más, la desviación estándar está dada por la raíz cuadrada de la varianza.

$$E\left(\sum_{i=1}^n c_i X_i\right) = \sum_{i=1}^n c_i E(X_i) \quad (3.5)$$

$$Var\left(\sum_{i=1}^n c_i X_i\right) = \sum_{i=1}^n c_i^2 Var(X_i) \quad (3.6)$$

Por supuesto, en R también podemos trabajar con combinaciones lineales de variables aleatorias discretas, como muestra el script 3.3.

Script 3.3: combinación lineal de variables aleatorias discretas en R.

```

1 library(discreteRV)
2
3 # Crear una variable discreta para representar el dado adulterado de la tabla
4 # 3.1, y calcular su valor esperado, varianza y desviación estandar.
5 resultados <- 1:6
6 probabilidades = c(0.25, 0.125, 0.125, 0.125, 0.125, 0.25)
7 X <- RV(outcomes = resultados, probs = probabilidades)
8 esperado_x <- E(X)
9 varianza_x <- V(X)
10 desviacion_x <- SD(X)
11 cat("E(X):", esperado_x, "\n")
12 cat("V(X):", varianza_x, "\n")
13 cat("SD(X):", desviacion_x, "\n\n")
14
15 # Crear una variable aleatoria para un dado balanceado, y calcular su valor
16 # esperado, varianza y desviación estandar.
17 Y <- RV(outcomes = resultados, probs = 1/6)

```

¹Si las variables no son independientes, se requieren métodos más complejos fuera del alcance de este libro.

```

18 esperado_y <- E(Y)
19 varianza_y <- V(Y)
20 desviacion_y <- SD(Y)
21 cat("E(Y):", esperado_y, "\n")
22 cat("V(Y):", varianza_y, "\n")
23 cat("SD(Y):", desviacion_y, "\n\n")
24
25 # Crear una combinación lineal de variables aleatorias, y calcular su valor
26 # esperado, varianza y desviación estandar.
27 Z <- 0.5 * X + 0.5 * Y
28 esperado_z <- E(Z)
29 varianza_z <- V(Z)
30 desviacion_z <- SD(Z)
31 cat("E(Z):", esperado_z, "\n")
32 cat("V(Z):", varianza_z, "\n")
33 cat("SD(Z):", desviacion_z)

```

Al examinar con mayor detención los gráficos de la figura 3.1 podemos apreciar que, a medida que se efectúan más lanzamientos del dado, el histograma se asemeja cada vez más a una curva continua, la cual recibe el nombre de **función de densidad de probabilidad**, o simplemente **distribución o densidad**.

Las distribuciones tienen la propiedad de que el área total bajo la curva siempre es 1, lo que resulta muy útil al momento de calcular probabilidades, pues basta con calcular el área bajo la curva del segmento deseado. Volviendo al ejemplo del desempeño del programa, presentado en la figura 3.2, el tiempo de ejecución es en realidad una variable continua. Así, la probabilidad de que el tiempo de ejecución sea mayor a 25 segundos corresponde al área coloreada en el gráfico de la figura 3.3, con un valor de 0,048².

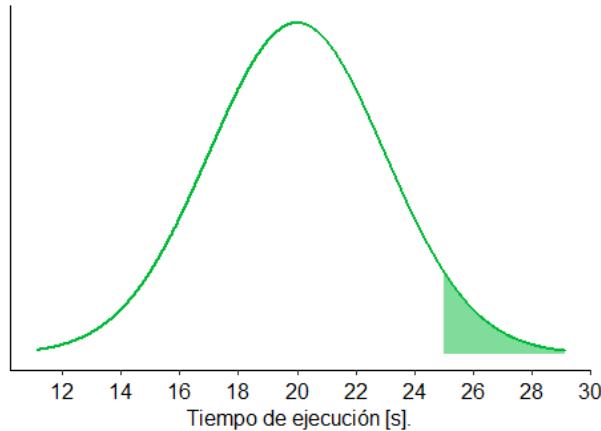


Figura 3.3: distribución para el desempeño del programa.

3.2 DISTRIBUCIONES CONTINUAS

Existen múltiples funciones de distribución continua que son de uso frecuente en estadística, las cuales se describen a continuación.

²El cálculo de esta probabilidad se aborda en el siguiente apartado

3.2.1 Distribución normal

También conocida como **distribución gaussiana**, la **distribución normal** es la más ampliamente empleada en estadística, pues muchas variables se acercan a esta distribución. Se caracteriza por ser unimodal y simétrica, con forma de campana. La figura 3.3 ejemplifica esta distribución.

La distribución normal se usa para modelar diversos fenómenos y podemos ajustarla mediante dos parámetros:

- μ : la media, que desplaza el centro de la curva a lo largo del eje x.
- σ : la desviación estándar, que modifica qué tan dispersos están los datos con respecto a la media.

Así, denotamos este tipo de distribución por $N(\mu, \sigma)$. La figura 3.4, creada mediante el script 3.4, muestra dos ejemplos superpuestos de distribución normal: $N(\mu = 0, \sigma = 1)$ en azul y $N(\mu = 10, \sigma = 6)$ en rojo.

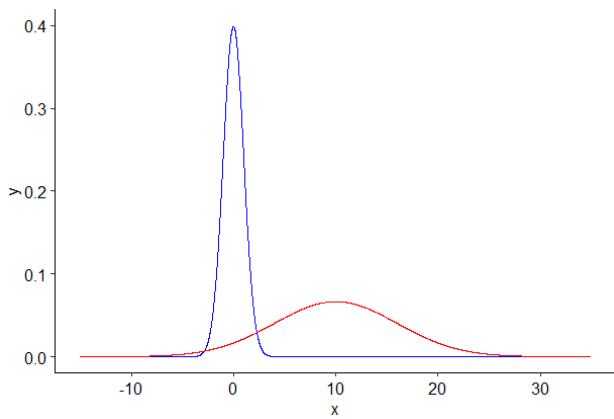


Figura 3.4: dos ejemplos superpuestos de distribución normal.

Script 3.4: graficando dos ejemplos de distribución normal.

```

1 library(ggpubr)
2
3 # Generar valores para una distribución normal con media 0 y
4 # desviación estándar 1.
5 media <- 0
6 desv_est <- 1
7 x <- seq(-15, 35, 0.01)
8 y <- dnorm(x, mean = media, sd = desv_est)
9 normal_1 <- data.frame(x, y)
10
11 # Repetir el proceso para una distribución normal con media 10
12 # y desviación estándar 6.
13 media <- 10
14 desv_est <- 6
15 x <- seq(-15, 35, 0.01)
16 y <- dnorm(x, mean = media, sd = desv_est)
17 normal_2 <- data.frame(x, y)
18
19 # Graficar ambas distribuciones.
20 g <- ggplot(normal_1, aes(x, y)) + geom_line(color = "blue")
21 g <- g + geom_line(data = normal_2, color = "red")
22 g <- g + theme_pubr()
23
24 print(g)

```

Antes de continuar, fijémonos en las líneas 8 y 16 del script 3.4, donde se usa la función `dnorm(x, mean, sd)`. Esta función calcula la densidad de una distribución normal. Además de `dnorm()`, R nos ofrece otras funciones que también resultan de mucha ayuda:

- `pnorm(q, mean, sd, lower.tail)`: permite encontrar percentiles, los cuales corresponden a la **función de distribución acumulada** (es decir, la probabilidad de que la variable tome valores menores o iguales que un valor dado), a partir de las probabilidades.
- `qnorm(p, mean, sd, lower.tail)`: encuentra el percentil para las probabilidades dadas en `p`, por lo que es la función inversa de `pnorm()`.
- `rnorm(n, mean, sd)`: genera aleatoriamente `n` observaciones de la distribución normal especificada.

Los argumentos de esta familia de funciones son:

- `x, q`: vector de cuantiles (percentiles).
- `p`: vector de probabilidades.
- `mean`: media de la distribución normal.
- `sd`: desviación estándar de la distribución normal.
- `lower.tail`: valor lógico que señala cuál de los dos extremos o colas de la distribución emplear.
- `n`: tamaño del vector resultante.

Es importante señalar que, por defecto, `lower.tail` toma el valor verdadero, con lo que `pnorm()` y `qnorm()` operan con la cola inferior de la distribución. Si, en cambio, `lower.tail = FALSE`, dichas funciones operan con la cola superior (es decir, `pnorm()` nos entrega la probabilidad de que la variable tome valores mayores que un valor dado).

Una **regla empírica** muy útil al momento de trabajar con distribuciones normales es la llamada regla 68-95-99.7, ilustrada en la figura 3.5, la cual establece que:

- Cerca de 68 % de las observaciones se encuentran a una distancia de una desviación estándar de la media.
- Alrededor de 95 % de las observaciones se encuentran a una distancia de dos desviación estándar de la media.
- Aproximadamente 99.7 % de las observaciones se encuentran a una distancia de tres desviación estándar de la media.

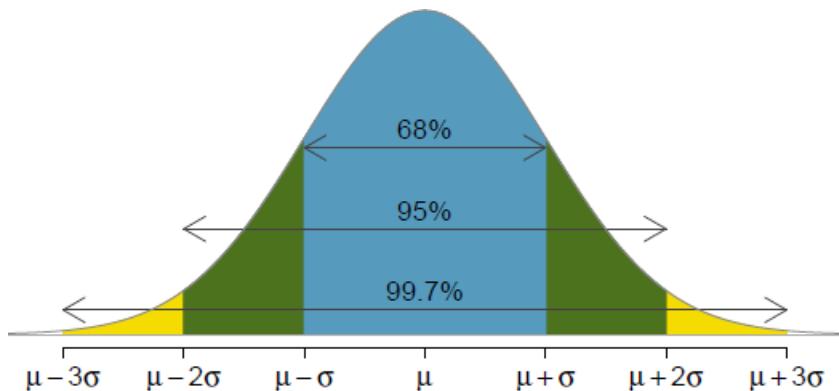


Figura 3.5: regla empírica de la distribución normal. Fuente: Diez et al. (2017, p. 136).

Muchas pruebas estadísticas operan bajo el supuesto de que los datos siguen una distribución normal. Como se insinuó en párrafos anteriores, la normalidad es siempre una aproximación, por lo que debemos verificar que el supuesto de una distribución normal sea aceptable. Una buena herramienta para ello es el **gráfico cuantil-cuantil**, también llamado **gráfico Q-Q**, que se muestra en la figura 3.6 y que podemos construir en R como muestra el script 3.5. En él podemos distinguir los siguientes elementos: un grupo de puntos, una recta y una región coloreada. Los puntos corresponden a las observaciones, mientras que la recta representa la distribución normal. En consecuencia, mientras más se asemeje el patrón que forman los puntos a la recta,

más parecida será la distribución a la normal. La banda coloreada establece el margen aceptable para suponer normalidad en el conjunto de datos. Así, para el conjunto de datos de la figura 3.6 sería imprudente aceptar el supuesto de normalidad.

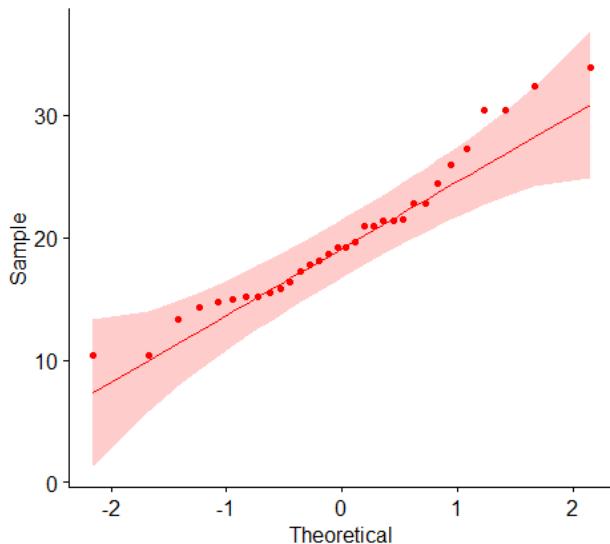


Figura 3.6: gráfico cuantil-cuantil.

Script 3.5: creación de un gráfico cuantil-cuantil.

```

1 library(ggpubr)
2
3 # Cargar datos.
4 datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
5                      row.names = 1)
6
7 # Gráfico Q-Q para la variable Rendimiento.
8 g <- ggqqplot(datos,
9                  x = "Rendimiento",
10                 color = "red")
11
12 print(g)

```

3.2.2 Distribución Z

Al trabajar con distribuciones, especialmente las simétricas, a menudo usaremos **técnicas de estandarización** para determinar qué tan usual o inusual es un determinado valor en una escala única. Así, para la distribución normal usamos como estandarización la **distribución Z** o **distribución normal estándar**, que no es más que una distribución normal centrada en 0 y con desviación estándar 1, que podemos obtener de manera bastante sencilla como muestra la ecuación 3.7.

$$Z = \frac{x - \mu}{\sigma} \quad (3.7)$$

Al aplicar la ecuación 3.7 a una observación x en una distribución normal obtenemos, entonces, su **valor**

z , que determina cuán por encima o por debajo de la media (en términos de la desviación estándar) se encuentra dicha observación x . Así, observaciones cuyos valores z sean negativos estarán por debajo de la media. Análogamente, un valor Z positivo indica que la observación está por sobre la media. Mientras mayor sea el valor absoluto de su valor z ($|z|$), más inusual será la observación.

3.2.3 Distribución chi-cuadrado

También llamada **ji-cuadrado** o χ^2 , la distribución **chi-cuadrado** (Devore, 2008) se usa para caracterizar valores siempre positivos y habitualmente desviados a la derecha. El único parámetro de esta distribución corresponde a los **grados de libertad**, usualmente representada por la letra griega ν , que son una estimación de la cantidad de observaciones empleadas para calcular un estimador. Otra forma de entender esta idea es como la cantidad de valores que pueden cambiar libremente en un conjunto de datos. Como ejemplo, supongamos que necesitamos una muestra de tres elementos cuya media sea 10. Una vez escogidos los primeros dos, solo queda una posibilidad para el tercero de modo que se cumpla con la media deseada. Así, solo los dos primeros valores pueden cambiar libremente, por lo que se tienen dos grados de libertad.

Esta distribución está relacionada con la ya conocida distribución Z , pues si sumamos los cuadrados de k variables aleatorias independientes que siguen una distribución Z , dicha suma sigue una distribución χ^2 con k grados de libertad:

$$\sum_{i=1}^k Z_i^2 \sim \chi^2(\nu = k) \quad (3.8)$$

La media de la distribución χ^2 es $\mu = \nu$, y su desviación estándar, $\sigma = 2\nu$.

Las funciones de R para esta distribución, similares a las descritas para la distribución normal, son:

- `dchisq(x, df)`.
- `pchisq(q, df, lower.tail)`.
- `qchisq(p, df, lower.tail)`.
- `rchisq(n, df)`.

Donde:

- x , q son vectores de cuantiles (enteros no negativos).
- p es un vector de probabilidades.
- n es la cantidad de observaciones.
- df son los grados de libertad.
- `lower.tail` es análogo al de la función `pnorm`.

La figura 3.7 muestra un ejemplo de la distribución χ^2 .

3.2.4 Distribución t de Student

Ampliamente empleada cuando se trabaja con muestras pequeñas, la **distribución t de Student**, o simplemente **distribución t**, tiene, al igual que la distribución χ^2 , los grados de libertad como único parámetro. A

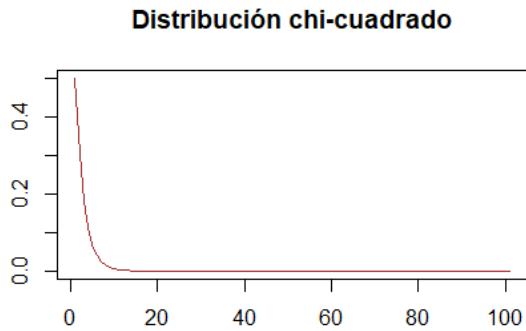


Figura 3.7: ejemplo de distribución χ^2 con 2 grados de libertad.

medida que los grados de libertad aumentan, esta distribución se asemeja cada vez más a la normal, aunque sus colas son más gruesas, como ilustra la figura 3.8.

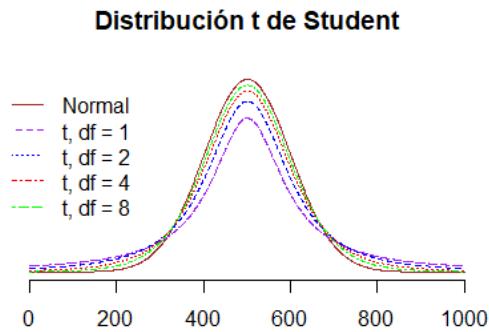


Figura 3.8: ejemplo de distribuciones t.

La distribución t se encuentra relacionada con las distribuciones vistas anteriormente de acuerdo a la ecuación 3.9, donde Z es una distribución normal estándar y $\chi^2(\nu)$ es una distribución χ^2 con ν grados de libertad.

$$Z \sqrt{\frac{\nu}{\chi^2(\nu)}} \sim t(\nu) \quad (3.9)$$

La media de la distribución t, para $\nu > 1$, es $\mu = 0$. Su desviación estándar, para $\nu > 2$, está dada por la ecuación 3.10.

$$\sigma = \sqrt{\frac{\nu}{\nu - 2}} \quad (3.10)$$

Las funciones de R para esta distribución, cuyos argumentos son análogos a los que hemos visto para las distribuciones anteriores, son:

- `dt(x, df)`.
- `pt(q, df, lower.tail)`.

- `qt(p, df, lower.tail)`.
- `rt(n, df)`.

3.2.5 Distribución F

Otra distribución que usaremos a lo largo de este libro es la **distribución F**, ampliamente usada para comparar varianzas. La distribución F se relaciona con las anteriores de acuerdo a la ecuación 3.11, donde $\chi_1^2(\nu_1)$ y $\chi_2^2(\nu_2)$ son dos distribuciones χ^2 con ν_1 y ν_2 grados de libertad, respectivamente. Un ejemplo de una distribución F se puede encontrar en la figura 3.9.

$$\frac{\frac{X_1^2(\nu_1)}{\nu_1}}{\frac{X_2^2(\nu_2)}{\nu_2}} \sim F(\nu_1, \nu_2) \quad (3.11)$$

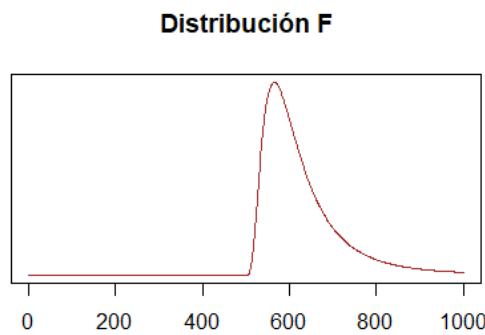


Figura 3.9: ejemplo de una distribución F.

Para $\nu_2 > 2$, la media de esta distribución está dada por la ecuación 3.12, y la desviación estandar corresponde a la ecuación 3.13 para $\nu_2 > 4$.

$$\mu = \frac{\nu_2}{\nu_2 - 2} \quad (3.12)$$

$$\sigma = \sqrt{\frac{2\nu_2^2(\nu_1 + \nu_2 - 2)}{\nu_1(\nu_2 - 2)^2(\nu_2 - 4)}} \quad (3.13)$$

Las funciones de R para esta distribución son:

- `df(x, df1, df2)`.
- `pf(q, df1, df2, lower.tail)`.
- `qf(p, df1, df2, lower.tail)`.
- `rf(n, df1, df2)`.

Donde `df1` como `df2` corresponden a grados de libertad y los argumentos restantes son los mismos que ya hemos visto anteriormente.

3.3 DISTRIBUCIONES DISCRETAS

Al igual que con las variables aleatorias continuas, también existen diversas distribuciones discretas de uso frecuente en estadística.

3.3.1 Distribución de Bernoulli

Una **variable aleatoria de Bernoulli** es aquella en que cada intento individual tiene solo dos resultados posibles: “éxito”, que ocurre con una probabilidad p y se representa habitualmente con un 1, y “fracaso”, que ocurre con probabilidad $q = 1 - p$ y suele representarse por un 0. La selección de qué resultado se considera como éxito o fracaso suele ser arbitraria. Para ilustrar esta idea, si dos personas lanzan una moneda al aire para sortear al ganador, cada una de ellas considerará una cara diferente de la moneda como un éxito.

Otro ejemplo que nos puede ayudar es el de lanzar varios dados de 20 caras, donde el éxito corresponde a obtener un 20 como resultado. Cada uno de ellos tiene una **probabilidad de éxito** (obtener 20) $p = 0.05$ y una **probabilidad de fracaso** (obtener otro valor) $q = 1 - p = 0.95$. Los lanzamientos de los dados son **independientes**, pues un dado no afecta a los demás.

Definimos la **proporción de la muestra** para una distribución de Bernoulli, \hat{p} , como la cantidad de éxitos dividida por la cantidad de intentos. Mientras mayor sea la cantidad de intentos, más cercano será el valor de \hat{p} a la probabilidad real de éxito p .

Al igual que la distribución normal, la distribución de Bernoulli puede resumirse expresando su media ($\mu = p$) y su desviación estándar. Esta última está dada por la ecuación 3.14.

$$\sigma = \sqrt{p(1-p)} \quad (3.14)$$

El paquete **extraDistr** de R ofrece 4 funciones, similares a las ya conocidas, para la distribución de Bernoulli:

- `dbern(x, prob)`.
- `pbern(q, prob, lower.tail)`.
- `qbern(p, prob, lower.tail)`.
- `rbern(n, prob)`.

3.3.2 Distribución geométrica

La **distribución geométrica** describe la cantidad de intentos que debemos realizar hasta obtener un éxito para variables de Bernoulli **independientes e idénticamente distribuidas**, es decir, que no se afectan unas a otras y cada una con igual probabilidad de éxito.

La probabilidad de obtener un éxito al n -ésimo intento está dada por la ecuación 3.15, donde podemos ver que las probabilidades en esta distribución decrecen exponencialmente rápido, como ilustra la figura 3.10. La media y la desviación estándar de la distribución geométrica están dadas, respectivamente, por las ecuaciones 3.16 y 3.17.

$$\Pr(\text{primer éxito al } n\text{-ésimo intento}) = (1 - p)^{n-1}p \quad (3.15)$$

$$\mu = \frac{1}{p} \quad (3.16)$$

$$\sigma = \sqrt{\frac{1-p}{p^2}} \quad (3.17)$$

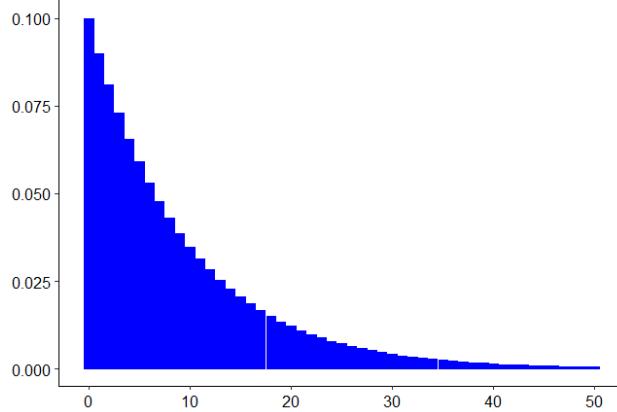


Figura 3.10: distribución geométrica para obtener un valor específico lanzando un dado de 20 caras.

Para entender mejor la utilidad de la distribución geométrica, consideremos la pregunta: ¿cuántas veces tenemos que lanzar un dado de 20 caras para obtener un 1? Anteriormente vimos que la probabilidad de éxito en este caso es $p = 0.05$. El valor esperado, representado por la media, sería en este caso el que se presenta en la ecuación 3.18.

$$\mu = \frac{1}{p} = \frac{1}{0.05} = 20 \quad (3.18)$$

Una vez más, R ofrece funciones similares a las presentadas anteriormente para trabajar con distribuciones geométricas:

- `dgeom(x, prob)`.
- `pgeom(q, prob, lower.tail)`.
- `qgeom(p, prob, lower.tail)`.
- `rbern(n, prob)`.

3.3.3 Distribución binomial

A diferencia de la distribución geométrica, la **distribución binomial** describe la probabilidad de tener exactamente k éxitos en n intentos independientes de Bernoulli con probabilidad de éxito p , cuya función de probabilidad está dada por la ecuación 3.19, donde:

- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ corresponde a la cantidad de formas de obtener k éxitos en un total de n intentos.
- $p^k(1-p)^{n-k}$ es la probabilidad de tener un único éxito en solo una de las $\binom{n}{k}$ maneras posibles.

$$f(k; n, p) = \Pr(k; n, p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (3.19)$$

La media y la desviación estándar de la distribución binomial están dadas por las ecuaciones 3.20 y 3.21, respectivamente. Un ejemplo de esta distribución se presenta en la figura 3.11

$$\mu = np \quad (3.20)$$

$$\sigma = \sqrt{np(1-p)} \quad (3.21)$$

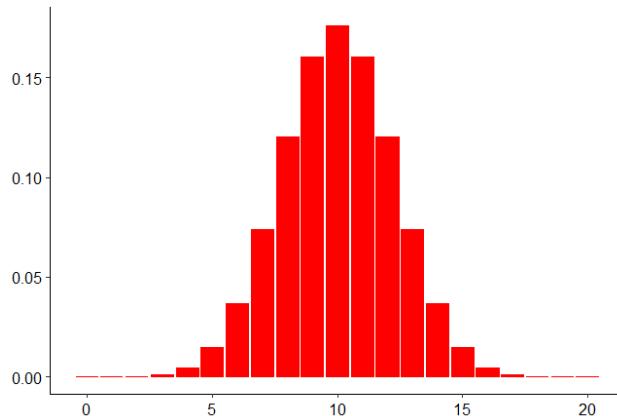


Figura 3.11: distribución binomial con $\mu = 400$ y $\sigma = 15.4019$.

Antes de decidir usar la distribución binomial, tenemos que verificar cuatro condiciones:

1. Los intentos son independientes.
2. La cantidad de intentos (n) es fija.
3. El resultado de cada intento puede ser clasificado como éxito o fracaso.
4. La probabilidad de éxito (p) es la misma para cada intento.

Las funciones que ofrece R para trabajar con esta distribución son:

- `dbinom(x, size, prob)`.
- `pbinom(x, size, prob)`.
- `qbinom(p, size, prob)`.
- `rbinom(n, size, prob)`.

Donde:

- x es un vector numérico.
- p es un vector de probabilidades.
- n es la cantidad de observaciones.
- $size$ corresponde al número de intentos.
- $prob$ es la probabilidad de éxito de cada intento.

En la figura 3.11 podemos observar que, en cierto modo, la distribución binomial se asemeja a la distribución normal: ambas son simétricas, aunque la distribución binomial no tiene la forma de campana de la distribución normal. Esta similitud ofrece una importante ventaja, pues en ocasiones es posible usar la distribución normal para estimar probabilidades binomiales, evitando así el uso de la compleja fórmula de la ecuación 3.19. Formalmente, esta aproximación es válida cuando el tamaño de la muestra, n , es lo suficientemente grande para que tanto np como $n(1-p)$ sean mayores o iguales que 10. En este caso, los parámetros de la distribución normal aproximada son los mismos de la distribución binomial (ecuaciones 3.20 y 3.21).

3.3.4 Distribución binomial negativa

La **distribución binomial negativa** es algo más general que la binomial, pues describe la probabilidad de encontrar el k -ésimo éxito al n -ésimo intento. Como señalan Diez et al. (2017, p. 155), “en el caso binomial, en general se tiene una cantidad fija de intentos y se considera la cantidad de éxitos. En el caso binomial negativo, se examina cuántos intentos se necesitan para observar una cantidad fija de éxitos y se requiere que la última observación sea un éxito”³.

Como adelanta la comparación anterior, antes de decidir usar la distribución binomial negativa tenemos que verificar cuatro condiciones:

1. Los intentos son independientes.
2. El resultado de cada intento puede ser clasificado como éxito o fracaso.
3. La probabilidad de éxito (p) es la misma para cada intento.
4. El último intento debe ser un éxito.

La función de probabilidad para esta distribución, ejemplificada en la figura 3.12, está dada por la ecuación 3.22. La varianza y la desviación estándar están dadas por las ecuaciones 3.23 y 3.24 (Devore, 2008, p. 120).

$$\Pr(k\text{-ésimo éxito al } n\text{-ésimo intento}) = \binom{n-1}{k-1} p^k (1-p)^{n-k} \quad (3.22)$$

$$\mu = \frac{k(1-p)}{p} \quad (3.23)$$

$$\sigma = \sqrt{\frac{k(1-p)}{p^2}} \quad (3.24)$$

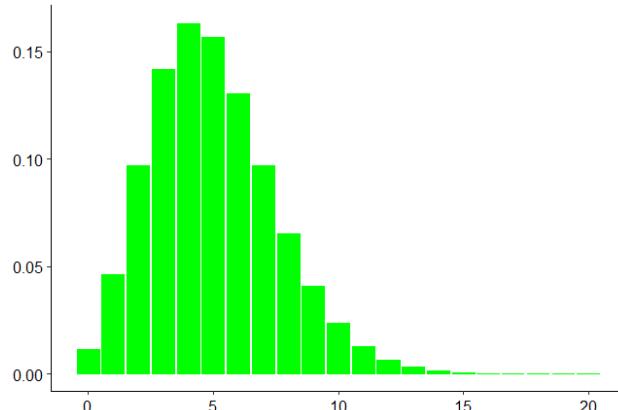


Figura 3.12: ejemplo de distribución binomial negativa.

Nuevamente, R dispone de cuatro funciones que permiten trabajar con esta distribución:

- `dnbino(x, size, prob)`.
- `pnbino(q, size, prob, lower.tail)`.
- `qnbino(p, size, prob, lower.tail)`.
- `rnbino(n, size, prob)`.

Donde:

³Traducción libre de los autores.

- x , q son vectores de cuantiles (enteros no negativos).
- p es un vector de probabilidades.
- n es la cantidad de observaciones.
- `size` corresponde al número (no negativo) de intentos.
- `prob` es la probabilidad de éxito de cada intento.
- `lower.tail` es análogo al de la función `pnorm`.

3.3.5 Distribución de Poisson

Útil para estimar la cantidad de eventos en una población grande en un lapso de tiempo dado, por ejemplo, la cantidad de contagios de influenza entre los habitantes de Santiago en una semana, la **distribución de Poisson** (figura 3.13) tiene una función de probabilidad definida por la ecuación 3.25, donde λ es la tasa o cantidad de eventos que se espera observar en un lapso de tiempo dado y k puede tomar cualquier valor entero no negativo. La media de esta distribución está dada por λ y la desviación estándar, por $\sqrt{\lambda}$.

$$\Pr(\text{observar } k \text{ eventos}) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (3.25)$$

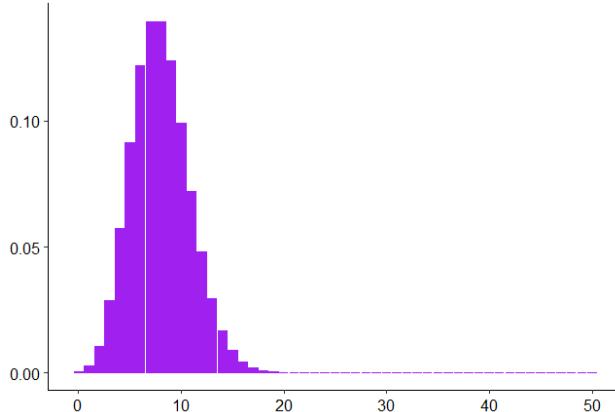


Figura 3.13: ejemplo de distribución de Poisson.

Las funciones de R para esta distribución son:

- `dpois(x, lambda)`.
- `ppois(q, lambda, lower.tail)`.
- `qqpois(p, lambda, lower.tail)`.
- `rpois(n, lambda)`.

Donde:

- x , q son vectores de cuantiles (enteros no negativos).
- p es un vector de probabilidades.
- n es la cantidad de observaciones.
- λ es un vector no negativo de medias.
- `lower.tail` es análogo al de la función `pnorm`.

3.4 EJERCICIOS PROPUESTOS

1. Da un ejemplo de variable aleatoria (novedosa) que puedas observar en tus compañeros y que tenga una función de densidad de probabilidad discreta.
2. Para la variable anterior, ¿cuál sería el valor esperado? ¿Cuál sería la varianza? ¿Cómo te imaginas su función de densidad de probabilidad?
3. Lista tres nombres distintos con que también se llama a las funciones de densidad de probabilidad.
4. Si una variable aleatoria tiene una función de densidad de probabilidad con media igual a 30 y desviación estándar de 3, ¿por qué podría ocurrir que la probabilidad de que la variable tome el valor 30 sea nula, es decir, $P(X = 30) = 0$?
5. Según el Reporte Mensual de Empleo, las siguientes son las estadísticas (media \pm desviación estándar) para las seis variables relevantes que se han estudiado en los últimos cinco años:
 - a) Número de personas despedidas: 64.675 ± 8.321 .
 - b) Número de personas renunciadas: 118.543 ± 17.936 .
 - c) Número de personas jubiladas: 97.092 ± 11.147 .
 - d) Número de empleos creados: 24.715 ± 10.832 .
 - e) Número de personas contratadas: 301.345 ± 27.261 .
 - f) Número de personas entrando a la fuerza de trabajo: 26.444 ± 29.440 .

Con esta información, calcula la media y la desviación estándar de:

- a) Caída neta del empleo: $(d) - (a) - (b) - (c)$.
 - b) Subida neta del empleo: $(e) - (a) - (b) - (c)$.
 - c) Caída neta del desempleo: $(a) + (b) + (e) + (f)$.
 - d) Vacancia del empleo: $(d) - (e)$.
6. ¿Qué significa que cierto valor de una variable aleatoria, usualmente con distribución normal, tenga valor $Z = 1,5$?
 7. Según la regla empírica, ¿entre qué estaturas se podría encontrar al 95 % de los estudiantes varones del Departamento de Ingeniería Informática de la Universidad de Santiago de Chile, si esta variable sigue una distribución $N(\mu = 171, \sigma = 3)$?
 8. ¿Qué información entrega un Gráfico Q-Q? ¿Para qué se usa?
 9. La probabilidad de que un estudiante universitario chileno seleccionado al azar sea VIH positivo es 0,013. ¿Cuáles serían la media y la desviación estándar de esta variable?
 10. En promedio, ¿a cuántos estudiantes universitarios se debería revisar hasta encontrar a un VIH positivo?
 11. Si el Departamento de Salud de una Universidad chilena controla a 50 estudiantes por día durante una semana de clases (lunes a viernes), ¿cuál sería el número promedio de VIH positivos detectados cada día? ¿Con qué varianza?
 12. Si la Universidad del ejercicio anterior dispone de 10 paquetes de tratamiento de VIH para estudiantes, ¿cómo podría saber a cuántos estudiantes debería examinar para poder asignarlos (suponiendo que todo estudiante VIH positivo acepta el tratamiento)?
 13. Muestra un ejemplo novedoso de una variable aleatoria relacionada que podría seguir una distribución de Poisson.

CAPÍTULO 4. FUNDAMENTOS PARA LA INFERENCIA

En el capítulo 1 se definen los conceptos de población, entendido como todo el conjunto de interés, y muestra, que es un subconjunto de la población. También se introducen las nociones de parámetro, correspondiente a un valor que resume la población (por ejemplo la media de la población, μ), y de estadístico, como valor que resume una muestra (por ejemplo, la media muestral, \bar{x}). La **inferencia estadística** tiene por objeto entender cuán cerca está el estadístico del parámetro real de la población. En este capítulo conoceremos los principios necesarios para la inferencia estadística, con base en Diez et al. (2017, pp. 168-202) y Field et al. (2012, pp. 40-47).

4.1 ESTIMADORES PUNTUALES

Como ya dijimos, los parámetros y los estadísticos son valores que resumen, respectivamente, una población y una muestra. En consecuencia, podemos decir que un estadístico corresponde a un **estimador puntual** de un parámetro. El valor de un estimador puntual cambia dependiendo de la muestra que usemos para obtenerlo. Así, por más que su valor se acerque al parámetro de la población, difícilmente será igual a este último. Sin embargo, el estimador tiende a mejorar a medida que aumentamos el tamaño de la muestra, por efecto de la **ley de los grandes números**. Para ilustrar este fenómeno, consideremos la **media móvil**, que es una secuencia de medias muestrales en que cada una de ellas toma un elemento más de la población que su antecesora. La figura 4.1, elaborada con el script 4.1, ejemplifica este fenómeno.

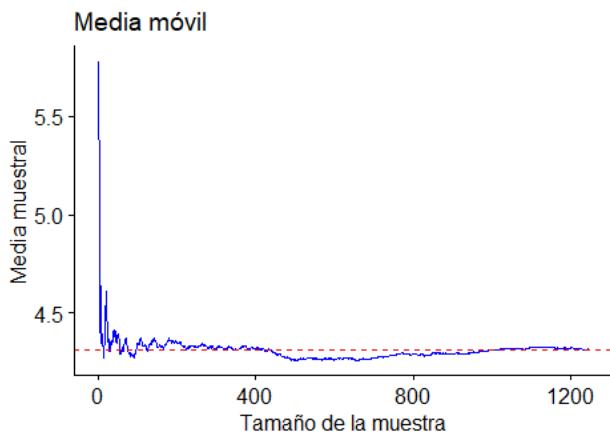


Figura 4.1: medias obtenidas al agregar a la muestra un elemento cada vez.

Script 4.1: representación gráfica de la media móvil.

```
1 library(ggpubr)
2
3 # Establecer la semilla para generar números aleatorios.
4 set.seed(9437)
5
6 # Generar aleatoriamente una población de tamaño 1500
7 # (en este caso, con una distribución cercana a la normal).
```

```

8 poblacion <- rnorm(n = 1500, mean = 4.32, sd = 0.98)
9
10 # Calcular la media de la población.
11 media_poblacion <- mean(poblacion)
12 cat("Media de la población:", media_poblacion, "\n")
13
14 # Tomar una muestra de tamaño 1250.
15 tamano_muestra <- 1250
16 muestra <- sample(poblacion, tamano_muestra)
17
18 # Calcular las medias acumuladas (es decir, con muestras de
19 # 1, 2, 3, ... elementos).
20 n <- seq(along = muestra)
21 media <- cumsum(muestra) / n
22
23 # Crear una matriz de datos con los tamaños y las medias muestrales.
24 datos <- data.frame(n, media)
25
26 # Graficar las medias muestrales.
27 g <- ggline(data = datos,
28               x = "n",
29               y = "media",
30               plot_type = "l",
31               color = "blue",
32               main = "Media móvil",
33               xlab = "Tamaño de la muestra",
34               ylab = "Media muestral")
35
36 # Añadir al gráfico una recta con la media de la población.
37 g <- g + geom_hline(aes(yintercept = media_poblacion),
38                      color = "red", linetype = 2)
39
40 print(g)

```

Para determinar qué tan adecuado es un estimador, necesitamos saber cuánto cambia de una muestra a otra. Si esta variabilidad es pequeña, es muy probable que la estimación sea buena. Podemos estudiar la variabilidad de la muestra con ayuda de la **distribución muestral**, que representa la distribución de estimadores puntuales obtenidos con **todas** las diferentes muestras de igual tamaño de una misma población. La figura 4.2 (construida con el script 4.2) representa las medias para diferentes muestras de una población, aunque solo una selección aleatoria de todas las posibles muestras, incluyendo además una línea vertical roja que señala la media de la población. Podemos destacar que las medias muestrales tienden a aglutinarse en torno a la media poblacional, pues de acuerdo al **teorema del límite central**, la distribución de \bar{x} se approxima a la normalidad. Esta aproximación mejora a medida que aumenta el tamaño de la muestra.

Script 4.2: distribución de la media muestral.

```

1 library(ggpubr)
2
3 # Establecer la semilla para generar números aleatorios.
4 set.seed(94)
5
6 # Generar aleatoriamente una población de tamaño 1500
7 # (en este caso, con una distribución cercana a la normal).
8 poblacion <- rnorm(n = 1500, mean = 4.32, sd = 0.98)
9
10 # Calcular la media de la población.
11 media_poblacion <- mean(poblacion)
12 cat("Media de la población:", media_poblacion, "\n")
13

```

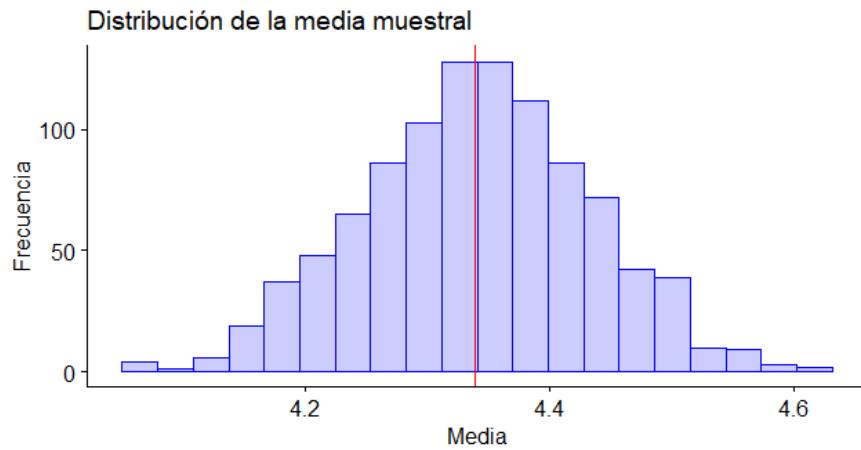


Figura 4.2: distribución muestral de la media para muestras con 100 observaciones.

```

14 # Tomar 1000 muestras de tamaño 100. Quedan almacenadas
15 # como una matriz donde cada columna es una muestra.
16 tamano_muestra <- 100
17 repeticiones <- 1000
18
19 muestras <- replicate(repeticiones,
20                         sample(poblacion, tamano_muestra))
21
22 # Calcular medias muestrales y almacenar los resultados
23 # en forma de data frame.
24 medias <- colMeans(muestras)
25 medias <- as.data.frame(medias)
26
27 # Construir un histograma de las medias muestrales.
28 g <- gghistogram(data = medias,
29                     x = "medias",
30                     bins = 20,
31                     title = "Distribución de la media muestral",
32                     xlab = "Media",
33                     ylab = "Frecuencia",
34                     color = "blue",
35                     fill = "blue",
36                     alpha = 0.2)
37
38 # Agregar línea vertical con la media de la población.
39 g <- g + geom_vline(aes(xintercept = media_poblacion),
40                      color = "red", linetype = 1)
41
42 print(g)

```

4.2 MODELOS ESTADÍSTICOS

Ahora que hemos conocido más conceptos, podemos definir con precisión qué es un **modelo estadístico**. En el capítulo 1 dijimos que un modelo es simplemente una representación y que los modelos estadísticos pueden

emplearse para diversos propósitos:

- Describir o resumir datos.
- Clasificar objetos o predecir resultados.
- Anticipar los resultados de intervenciones (en ocasiones).

Más formalmente, un modelo estadístico es una descripción de un **proceso probabilístico** con **parámetros desconocidos** que deben ser **estimados** en base a **suposiciones** y un conjunto de datos **observados**. En general, tiene la forma dada en la ecuación 4.1:

$$y_i = (\text{modelo}) + \varepsilon_i \quad (4.1)$$

Donde:

- y_i es el i -ésimo valor observado de la variable respuesta Y (también llamada variable de salida o variable dependiente).
- modelo es el resultado de una función determinista basada en un conjunto de argumentos.
- ε_i es el error, correspondiente a la **variación natural**, y no a una equivocación, existente entre los valores observados y los valores pronosticados por el modelo. También recibe los nombres de variación no sistemática, variación aleatoria, residuos o incluso, residuales.

El error ε_i en la ecuación 4.1 se relaciona entonces con la calidad del modelo. Mientras menor sea el error, mejor será el modelo. Por el contrario, un error grande es señal de un modelo fallido, que no describe bien los datos, no ayuda a predecirlos bien, o no ayuda a su correcta clasificación.

La media y la proporción, y cualquier estadístico en general, son, en sí mismos, modelos estadísticos, aunque bastante simples.

4.3 ERROR ESTÁNDAR

En el capítulo 2 conocimos la desviación estándar como medida que estima la distancia de las observaciones respecto de la media. El **error estándar**, denotado usualmente por $SE_{\hat{\theta}}$ o $\sigma_{\hat{\theta}}$, corresponde a la desviación estándar de la distribución de un estimador muestral $\hat{\theta}$ de un parámetro θ . Por ejemplo, el error estándar de la media, es decir la desviación estándar de la distribución de las medias de todas las posibles muestras de n observaciones independientes, se calcula de acuerdo a la ecuación 4.2.

$$SE_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \quad (4.2)$$

Donde σ es la desviación estándar de la población y n corresponde al tamaño de la muestra. En esta ecuación queda en evidencia que el error estándar de la media disminuye a medida que el tamaño de la muestra aumenta. Un método confiable que podemos usar para asegurar que las observaciones sean independientes es realizar un muestreo aleatorio simple¹ que abarque menos del 10 % de la población.

Volviendo a la ecuación para calcular el error estándar de la media muestral (ecuación 4.2), ¡debemos tener cuidado antes de usarla! Ya hemos mencionado antes que la distribución de las medias muestrales tiende a ser cercana a la normal, por lo que en dicho caso es posible usar el **modelo normal**, sustentado en el teorema del límite central. Las condiciones que deben cumplirse para usar este modelo y que, en consecuencia, el error estándar sea preciso, son:

1. Las observaciones de la muestra son independientes.

¹ Es decir, una muestra en que todos los elementos de la población tengan igual probabilidad de ser escogidos. Las técnicas de muestreo se abordan con más detalle en capítulos posteriores.

2. La muestra es grande (en general $n \geq 30$).
3. La distribución de la muestra no es significativamente asimétrica. Esto último suele además relacionarse con la presencia de valores atípicos. Mientras mayor sea el tamaño de la muestra, más se puede relajar esta condición.

Si no se cumplen las condiciones anteriores, debemos considerar otras opciones: para muestras pequeñas, se deben considerar métodos alternativos, y si la distribución de la muestra presenta una asimetría significativa, entonces tendremos que incrementar el tamaño de la muestra para compensar el efecto de la desviación.

4.4 INTERVALOS DE CONFIANZA

Hasta ahora sabemos que un estimador puntual es un único valor (obtenido a partir de una muestra) que, como su nombre indica, estima un parámetro de la población. Por ende, dicho valor rara vez es exacto. En consecuencia, lo lógico sería establecer un rango de valores plausibles para el parámetro estimado, que llamaremos **intervalo de confianza**, y que se construye en torno al estimador puntual. Dado que el error estándar representa la desviación estándar asociada al estimador, tiene sentido que lo usemos como guía en este proceso.

Recordemos que en el capítulo 3 vimos una regla empírica para la distribución normal (figura 3.5), la cual señala que (para distribuciones normales) alrededor de 95 % de las veces el estimador puntual se encontrará en un rango de 2 errores estándar del parámetro. Es decir, al considerar un intervalo de confianza de dos errores estándar (4.3), tendremos 95 % de **confianza** de haber capturado el parámetro real.

$$\bar{x} \pm 2 \cdot SE_{\bar{x}} \quad (4.3)$$

Podemos generalizar la ecuación 4.3 para calcular el intervalo de confianza para la media con cualquier **nivel de confianza** como muestra la ecuación 4.4.

$$\bar{x} \pm z^* \cdot SE_{\bar{x}} \quad (4.4)$$

El término z^* en la ecuación 4.4 corresponde, usualmente, al valor z tal que el área bajo la curva normal estándar comprendida entre $-z^*$ y z^* es igual al nivel de confianza deseado. La expresión $z^* \cdot SE$ recibe el nombre de **margen de error**.

Tomemos como ejemplo un **nivel de confianza** (que, por razones que veremos en la sección siguiente, denotaremos por $1 - \alpha$) de 90 % (es decir, $1 - \alpha = 0,9$). Eso significa, entonces, que nuestro intervalo de confianza excluye el 5 % del área correspondiente a la cola inferior (es decir, el percentil con valor 0,05) e igual porcentaje del área correspondiente a la cola superior (que, como la distribución Z es simétrica, es igual al área anterior). Puesto que conocemos el percentil, $(1 - \alpha)/2 = 0,05$, en R podemos usar la llamada `qnorm(0.05, mean = 0, sd = 1, lower.tail = FALSE)` y obtenemos $z^* = 1,64$. Es importante indicar que en esta llamada estamos en realidad trabajando con la cola superior para que z^* sea positivo. Si hacemos la llamada para la cola inferior, obtenemos $z^* = -1,64$.

Es importante destacar que, una vez más, debemos ser cuidadosos al interpretar un intervalo de confianza del $x\%$ ($x = 1 - \alpha$). Su significado es, sencillamente, “se tiene $x\%$ de certeza de que el parámetro de la población se encuentra entre...” (Diez et al., 2017, p. 180), es decir, que, en promedio, $x\%$ de los intervalos de confianza que se construyan en torno a un estadístico, con muestras de un tamaño fijo, capturarán el verdadero valor del parámetro. Esto **no es equivalente** a decir que el valor del parámetro tiene una “probabilidad de $x\%$ ” de estar entre los valores del intervalo calculado, lo que sería incorrecto. Por otra parte, los intervalos de confianza no dicen nada acerca de observaciones individuales, sino que solo hablan del parámetro en cuestión.

4.5 PRUEBAS DE HIPÓTESIS

Supongamos que un banco ha desarrollado un nuevo sistema computacional para gestionar sus transacciones. El nuevo sistema (N) se ha puesto a prueba durante un mes, funcionando (con iguales condiciones de hardware) en paralelo con el sistema antiguo (A) y el banco ha llevado un registro del tiempo que tarda cada sistema en efectuar cada transacción. El gerente ha determinado que autorizará la migración al nuevo sistema únicamente si este es más rápido que el antiguo para procesar las transacciones. Se sabe que el sistema antiguo tarda en promedio $\mu_A = 530$ milisegundos en procesar una transacción. Para el sistema nuevo se han registrado $n = 1.600$ transacciones, realizadas en un tiempo promedio de $\bar{x}_N = 527,9$ [ms] con desviación estándar $s_N = 48$ [ms].

Una primera aproximación para tomar la decisión puede ser investigar si existe diferencia en los tiempos de ejecución de ambos sistemas, lo que puede expresarse en torno a dos **hipótesis** (palabra que la Real Academia Española (2014) define como “Suposición de algo posible o imposible para sacar de ello una consecuencia”) que compiten entre sí:

H_0 : El nuevo sistema, en promedio, tarda lo mismo que el antiguo en procesar las transacciones, es decir:

$$\mu_N = \mu_A.$$

H_A : Los sistemas requieren, en promedio, cantidades de tiempo diferentes para procesar las transacciones, es decir: $\mu_N \neq \mu_A$

La primera hipótesis, H_0 , recibe el nombre de **hipótesis nula** y suele representar una postura escéptica, es decir, que no hay cambios, por lo que **la hipótesis nula siempre se formula como una igualdad!**. La segunda (H_A), llamada **hipótesis alternativa**, representa en cambio una nueva perspectiva. Esta primera aproximación corresponde a una **prueba bilateral** o de dos colas, pues la diferencia puede ser en ambos sentidos: H_0 no parece correcta si $\mu_N < \mu_A$ o si $\mu_N > \mu_A$.

Como en este caso conocemos el valor de $\mu_A = 530$ [ms], también podríamos escribir la formulación matemática de las hipótesis de la siguiente manera:

$$H_0: \mu_N = 530$$

$$H_A: \mu_N \neq 530$$

En este planteamiento, “530” recibe el nombre de **valor nulo**, pues representa el valor del parámetro cuando se cumple la hipótesis nula.

Una aproximación más cercana al problema descrito puede ser investigar si el nuevo sistema es efectivamente **más rápido** que el antiguo. En este caso, se habla de una **prueba unilateral** o de una cola, pues solo interesa saber si el tiempo promedio empleado por el nuevo sistema es menor que el empleado por el sistema antiguo. Las hipótesis, en este caso, serían:

H_0 : El nuevo sistema tarda, en promedio, lo mismo que el antiguo en procesar las transacciones, es decir:

$$\mu_N = \mu_A.$$

H_A : El nuevo sistema tarda, en promedio, menos que el antiguo en procesar las transacciones, es decir:
 $\mu_N < \mu_A$

Obviamente en otros casos podría interesar solamente si valor alternativo es mayor que el valor nulo.

Teniendo las hipótesis planteadas, sigue decidir si la hipótesis nula parece o no plausible a través de una **prueba de hipótesis**. El marco para la prueba de hipótesis es **escéptico**: no se rechaza la hipótesis nula a menos que haya suficiente evidencia para rechazarla en favor de la hipótesis alternativa. Esta idea es muy parecida a la expresada en la expresión de uso común “se presume inocente mientras no se demuestre lo contrario”. Sin embargo, el que no se logre rechazar H_0 **no significa aceptarla** como verdadera o como correcta sin más. Por eso se usa un lenguaje bastante peculiar, señalando que *se falla al rechazar H_0* o bien que *se rechaza H_0 en favor de H_A* . Retomando la analogía con la expresión anterior, que no haya pruebas suficientes para la culpabilidad, no significa que una persona sea en verdad inocente.

Volvamos al escenario del ejemplo para la prueba de hipótesis bilateral (es decir, aquella en que solo queremos

ver si hay diferencias en el tiempo de procesamiento de transacciones entre ambos sistemas del banco). El valor de $\bar{x}_N = 527,9$ [ms] es, en efecto, distinto de $\mu_A = 530$ [ms]. No obstante, al ser una estimación puntual, como ya hemos estudiado, esta diferencia podría deberse simplemente a la muestra escogida, por lo que el parámetro real μ_N podría ser igual a μ_A [ms]. En consecuencia, resulta útil calcular el intervalo de confianza para \bar{x}_N .

Comencemos por determinar el error estándar:

$$SE_{\bar{x}} = \frac{s_N}{\sqrt{n}} = \frac{48}{\sqrt{1600}} = 1,2$$

Ahora fijemos un nivel de confianza, por ejemplo 95 %, y usemos el valor z^* correspondiente para calcular el intervalo de confianza:

$$\bar{x}_N \pm z^* \cdot SE_{\bar{x}} = 527,9 \pm 1,96 \cdot 1,2 = [525,548; 530,252]$$

Como el parámetro del sistema antiguo ($\mu_A = 530$ [ms]) cae (a penas) dentro de este intervalo, se puede suponer que no existe una diferencia significativa entre los tiempos promedio requeridos por ambos sistemas, por lo que no se rechaza H_0 . Así, tenemos un 95 % de confianza en que no existe una diferencia entre los tiempos que requieren ambos sistemas para procesar transacciones. Sin embargo, esta decisión es un tanto apresurada ya que el resultado está cerca del borde de rechazo y, en este caso, lo lógico sería investigar más (hacer crecer la muestra).

Revisemos ahora el caso planteado con hipótesis alternativa unilateral (es decir, queremos ver si el nuevo sistema es, en efecto, más rápido). Manteniendo nuestro nivel de confianza $1-\alpha = 0,95$, en este caso debemos considerar los valores menores a $\mu_A = 530$ [ms] para el cálculo de z^* . En otras palabras, el 5 % que descartamos corresponde únicamente a la cola superior. Así, nuestro valor para z^* está dado por la llamada `qnorm(0.05, mean = 0, sd = 1, lower.tail = FALSE)`, obteniéndose $z^* = 1,64$ (aprox.) por lo que se tiene que la cota superior es:

$$\bar{x}_N - z^* \cdot SE_{\bar{x}} = 527,9 - 1,64 \cdot 1,2 = 529,874$$

Luego, el intervalo de confianza va desde “cualquier valor” bajo la media observada en la muestra hasta el valor calculado arriba, por lo que el intervalo con 95 % confianza sería: $[-\infty; 529,874]$.

Ahora el valor $\mu_A = 530$ [ms] cae (apenas) fuera del intervalo y podemos decir que existe evidencia de que el nuevo sistema tarda en promedio menos tiempo que el antiguo en procesar las transacciones.

Ahora bien, siempre que se prueban hipótesis podemos cometer un error al momento de decidir si rechazar o no la hipótesis nula. Afortunadamente, la estadística ofrece herramientas para cuantificar cuán frecuentes son dichos errores. Existen cuatro posibles escenarios, los cuales se presentan en la tabla 4.1. El **error tipo I** corresponde a rechazar H_0 cuando en realidad es verdadera, mientras que el **error tipo II** corresponde a no rechazarla cuando en realidad H_A es verdadera.

Conclusión de la prueba		
	No rechazar H_0	Rechazar H_0 en favor de H_A
Verdad	H_0 verdadera	Decisión correcta
	H_A verdadera	Error tipo II
		Decisión correcta

Tabla 4.1: posibles escenarios para una prueba de hipótesis.

Como ya hemos señalado, la prueba de hipótesis se basa en no rechazar H_0 a menos que se tenga evidencia contundente. Por regla general, no se desea cometer el error de rechazar incorrectamente la hipótesis nula (error tipo I) en más de 5 % de los casos. Esto corresponde a un **nivel de significación** de 0,05, denotado por $\alpha = 0,05$. Si usamos un intervalo de confianza de 95 % para evaluar una prueba de hipótesis en que la

hipótesis nula es verdadera, cometemos un error tipo I cada vez que el estimador puntual esté a 1,96 o más errores estándar del parámetro de la población. Esto puede ocurrir un 5 % de las veces (2,5 % en cada cola de la distribución para el caso bilateral). Del mismo modo, un intervalo de confianza del 99 % es equivalente a un nivel de significación $\alpha = 0,01$.

El intervalo de confianza es de mucha ayuda para decidir si rechazar o no H_0 . No obstante, no aporta información directa acerca de cuán fuerte es la evidencia para la decisión tomada.

4.5.1 Prueba formal de hipótesis con valores p

Antes de que la computación se hiciera masiva, las personas tenían dos procedimientos posibles para decidir una prueba de hipótesis. El primero es el realizado en la sección anterior, esto es, calcular el intervalo con $(1 - \alpha) \%$ de confianza de acuerdo a los estadísticos observados en una muestra y revisar si el valor nulo cae o no dentro de este intervalo. El otro procedimiento clásico, que podemos encontrar en muchos libros y sitios en Internet, es estimar a qué valor z corresponde la media observada en la distribución normal estandarizada que define el valor nulo y el error estándar: si este estadístico z es mayor que z^* , entonces el estadístico cae en una “zona de rechazo” de H_0 ; en caso contrario ($|z| < z^*$), se falla en rechazar la hipótesis nula.

Si bien estos procedimientos siguen siendo útiles, su diseño respondía a la existencia de **tablas de probabilidad** en que se tabulaban probabilidades para algunos valores de percentiles de uso común, como 90 %, 95 %, 0,975 % o 0,99 %.

Con la llegada de los computadores, y en particular de entornos como R, es posible obtener probabilidades (casi) exactas para cualquier percentil. Esto hizo que un tercer método para decidir una prueba de hipótesis haya ido ganando popularidad: el uso del **valor p**, también llamado **p-valor**, que es definido por Diez et al. (2017, p. 186) como “la probabilidad de observar datos al menos tan favorables como la muestra actual para la hipótesis alternativa, si la hipótesis nula es verdadera”. De esta forma, un p-valor permite cuantificar cuán fuerte es la evidencia en contra de la hipótesis nula (y en favor de la hipótesis alternativa).

Consideremos ahora el escenario de la hipótesis unilateral del ejemplo, con un nivel de significación $\alpha = 0,05$, bajo el supuesto de que H_0 es verdadera y que la muestra a su vez tiene una distribución cercana a la normal. Recordemos que $\bar{x}_N = 527,9$ [ms] y $s_N = 48$ [ms] en $n = 1600$ observaciones. Esta distribución se vería como muestra la figura 4.3, creada mediante el script 4.3.

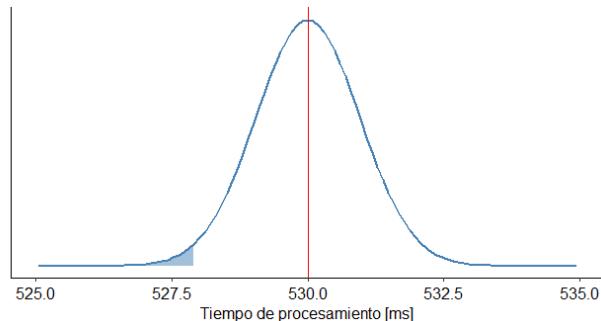


Figura 4.3: probabilidad de encontrar una media igual o menor que $\bar{x} = 527,9$ [ms] en la distribución muestral con $\mu_{\bar{x}} = 530$ y $\sigma_{\bar{x}} = 1,2$.

En este punto, resulta importante hacer una aclaración en relación al valor p. El área bajo la sección de la curva con valores menores o iguales a un estimador se calcula usando para ello el **valor z**, definido en la ecuación 4.5, como **estadístico de prueba**.

$$z = \frac{\text{estimador puntual} - \text{valor nulo}}{SE_{\text{estimador puntual}}} = \frac{\hat{\theta} - \theta_0}{SE_{\hat{\theta}}} \quad (4.5)$$

Un **estadístico de prueba** es un estadístico de resumen que resulta especialmente útil para evaluar hipótesis o calcular el valor p. El valor z se usa cuando el estimador puntual se acerca a la normalidad, aunque existen otros estadísticos de prueba adecuados para otros escenarios.

Script 4.3: cálculo del valor p para una prueba de una cola.

```

1 library(ggpubr)
2
3 # Generar una muestra donde la media cumpla con la hipótesis nula.
4 set.seed(872)
5
6 media_poblacion_antiguo <- 530
7 media_muestra_nuevo <- 527.9
8 desv_est <- 48
9 n <- 1600
10 error_est <- desv_est / sqrt(n)
11
12 x <- seq(media_poblacion_antiguo - 5.2 * error_est,
13             media_poblacion_antiguo + 5.2 * error_est,
14             0.01)
15
16 y <- dnorm(x, mean = media_poblacion_antiguo, sd = error_est)
17
18 datos <- data.frame(x, y)
19
20 # Graficar la muestra.
21 g <- ggplot(data = datos, aes(x))
22
23 g <- g + stat_function(fun = dnorm,
24                         args = list(mean = media_poblacion_antiguo,
25                                     sd = error_est),
26                         colour = "steelblue", size = 1)
27
28 g <- g + ylab("")
29 g <- g + scale_y_continuous(breaks = NULL)
30 g <- g + scale_x_continuous(name = "Tiempo de procesamiento [ms]")
31 g <- g + theme_pubr()
32
33 # Colorear el área igual o menor que la media observada.
34 g <- g + geom_area(data = subset(datos,
35                         x < media_muestra_nuevo),
36                         aes(y = y),
37                         colour = "steelblue",
38                         fill = "steelblue",
39                         alpha = 0.5)
40
41 # Agregar una línea vertical para el valor nulo.
42 g <- g + geom_vline(aes(xintercept = media_poblacion_antiguo),
43                         color = "red", linetype = 1)
44
45 print(g)
46
47 # Calcular el valor Z para la muestra.
48 Z <- (media_muestra_nuevo - media_poblacion_antiguo) / error_est
49
```

```

50 # Calcular el valor p.
51 p_1 <- pnorm(z, lower.tail = TRUE)
52
53 cat("Valor p: ", p_1, "\n")
54
55 # También se puede calcular el valor p directamente a partir de la
56 # distribución muestral definida por el valor nulo y el error
57 # estándar.
58 p_2 <- pnorm(media_muestra_nuevo, mean = media_poblacion_antiguo,
59             sd = est_err)
60
61 cat("Valor p: ", p_2)

```

El valor p , en este caso $p = 0,040$, corresponde al área coloreada en la figura 4.3, y se calcula en la línea 51 del script 4.3. Esto nos indica, en este caso, que si H_0 fuera verdadera y el nuevo sistema tarda en promedio lo mismo que el antiguo en procesar las transacciones, la probabilidad de encontrar una media de a lo más 527,9 [ms] para una muestra de 1.600 transacciones es de 4 %, lo que sería bastante poco frecuente.

Cuanto menor sea el valor p , más fuerte será la evidencia en favor de H_A por sobre H_0 . Y aquí la ventaja de usar este método para decidir: el valor p se puede **comparar directamente** con el nivel de significación α , y si p es menor que el nivel de significación se considera evidencia suficiente para rechazar la hipótesis nula en favor de la hipótesis alternativa. En este ejemplo, $p = 0,040 < \alpha = 0,05$, por lo que se rechaza H_0 en favor de H_A . Pero como se dijo cuando usamos intervalos de confianza, el valor p está cerca del valor α y convendría ser menos tajante en la decisión y evaluar la posibilidad de ampliar la muestra para conseguir evidencia más definitiva.

Siempre es recomendable formular la conclusión de la prueba de hipótesis en lenguaje llano, para facilitar su comprensión. Así, en este caso concluimos que los datos sugieren que el nuevo sistema tarda menos que el antiguo en procesar transacciones, pero que es necesario hacer un estudio con más observaciones para tener un diagnóstico más definitivo.

Volvamos nuevamente al escenario de la prueba de hipótesis bilateral para el ejemplo, manteniendo el nivel de significación $\alpha = 0,05$. Puesto que en este caso nos interesa la diferencia en ambas direcciones, ya que la evidencia en ambas direcciones es favorable para H_A , debemos considerar el área bajo las dos colas de la curva normal, a diferencia del caso de la prueba de hipótesis unilateral en que solo se consideramos la cola correspondiente a la dirección de interés de la diferencia. Dado que el modelo normal es simétrico, el área bajo ambas colas es la misma (figura 4.4, script 4.4). El valor p , entonces, ahora es igual a dos veces el área de la cola inferior, es decir, $p = 0,080$. Puesto que $p > \alpha$, se falla en rechazar H_0 . Es decir, no hay evidencia suficiente para concluir que existe una diferencia entre los tiempos promedio requeridos por ambos sistemas para procesar transacciones.

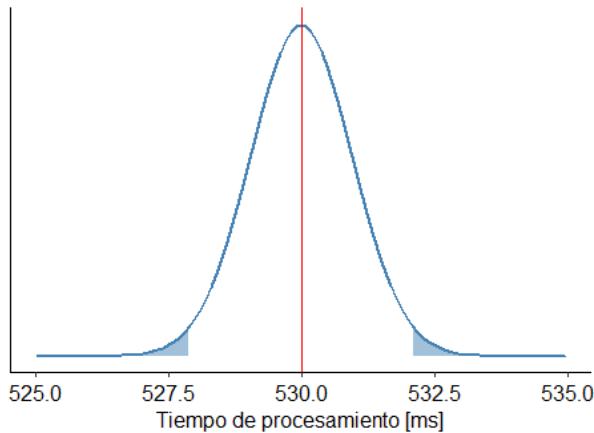


Figura 4.4: cuando la prueba de hipótesis es bilateral, se deben colorear ambas colas.

Script 4.4: cálculo del valor p para una prueba de dos colas.

```
1 library(ggpubr)
2
3 # Generar una muestra donde la media cumpla con la hipótesis nula.
4 set.seed(208)
5
6 media_poblacion_antiguo <- 530
7 media_muestra_nuevo <- 527.9
8 desv_est <- 48
9 n <- 1600
10 error_est <- desv_est / sqrt(n)
11
12 x <- seq(media_poblacion_antiguo - 5.2 * error_est,
13             media_poblacion_antiguo + 5.2 * error_est,
14             0.01)
15
16 y <- dnorm(x,
17               mean = media_poblacion_antiguo,
18               sd = error_est)
19
20 dataframe <- data.frame(x, y)
21
22 # Graficar la muestra.
23 g <- ggplot(data = dataframe, aes(x))
24
25 g <- g + stat_function(fun = dnorm,
26                         args = list(mean = media_poblacion_antiguo,
27                                     sd = error_est),
28                         colour = "steelblue", size = 1)
29
30 g <- g + ylab("")
31 g <- g + scale_y_continuous(breaks = NULL)
32 g <- g + scale_x_continuous(name = "Tiempo de procesamiento [ms]")
33 g <- g + theme_pubr()
34
35 # Colorear el área igual o menor que la media observada.
36 g <- g + geom_area(data = subset(dataframe,
37                           x < media_muestra_nuevo),
38                           aes(y = y),
39                           colour = "steelblue",
40                           fill = "steelblue",
41                           alpha = 0.5)
42
43 # Calcular el área bajo la cola inferior.
44 area_inferior <- pnorm(media_muestra_nuevo,
45                         mean = media_poblacion_antiguo,
46                         sd = desv_est)
47
48
49 # Colorear igual área en la cola restante.
50 corte_x <- qnorm(1 - area_inferior,
51                   mean = media_poblacion_antiguo,
52                   sd = desv_est)
53
54 g <- g + geom_area(data = subset(dataframe,
55                           x > corte_x),
56                           aes(y = y),
57                           colour = "steelblue",
58                           fill = "steelblue"),
```

```

59         alpha = 0.5)
60
61 # Agregar una línea vertical para el valor nulo.
62 g <- g + geom_vline(aes(xintercept = media_poblacion_antiguo),
63                       color = "red", linetype = 1)
64
65 print(g)
66
67 # Calcular el valor Z para la muestra.
68 Z <- (media_muestra_nuevo - media_poblacion_antiguo) / error_est
69
70 # Calcular el valor p (recordando ahora que la hipótesis es bilateral).
71 p <- 2 * pnorm(Z, lower.tail = TRUE)
72
73 cat("Valor p: ", p)

```

Un punto importante que debemos tener en cuenta es que **las pruebas unilaterales** se usan cuando se desea verificar un incremento o un decremento, pero no ambas. No obstante, esta decisión debe tomarse siempre **antes de examinar los datos**, pues de lo contrario se duplica la probabilidad de cometer errores de tipo I y se está cayendo en **prácticas poco éticas**.

4.5.2 El efecto del nivel de significación

Hemos visto que el nivel de significación (α) representa la proporción de veces en que se cometería un error de tipo I (es decir, rechazar H_0 en favor de H_A , cuando H_0 es en realidad verdadera). Si resulta costoso o peligroso cometer un error de este tipo, debemos requerir evidencia más fuerte para rechazar la hipótesis nula (es decir, reducir la probabilidad de que esto ocurra), lo que podemos lograr usando un valor más pequeño para el nivel de significación, por ejemplo, $\alpha = 0,01$. Sin embargo, esto necesariamente **aumentará** la probabilidad de cometer un error de tipo II.

Si, por el contrario, el costo o el peligro de cometer un error de tipo II (no rechazar H_0 cuando en realidad H_A es verdadera) es mayor, debemos escoger un nivel de significación más elevado (por ejemplo, $\alpha = 0,10$).

Así, el nivel de significación seleccionado para una prueba siempre debe reflejar las consecuencias de cometer errores de tipo I o de tipo II.

4.6 INFERENCIA PARA OTROS ESTIMADORES

Hasta ahora, solo hemos considerado la media como estimador para la inferencia. No obstante, muchos de los conceptos que hemos visto en este capítulo pueden aplicarse, con algunas ligeras modificaciones, usando otros estimadores.

4.6.1 Estimadores puntuales con distribución cercana a la normal

En realidad existen múltiples estimadores puntuales, además de la media, cuya distribución muestral es cercana a la normal si las muestras son lo suficientemente grandes, tales como las proporciones y la diferencia de medias. Si bien veremos con detalle la prueba de hipótesis con estos estimadores puntuales en capítulos posteriores, es importante contar con algunas orientaciones generales.

Un supuesto importante que debemos tener en cuenta es que el estimador puntual $\hat{\theta}$ debe ser **insesgado**. Esto significa que la distribución muestral de $\hat{\theta}$ tiene su centro en el valor del parámetro θ que estima. En otras palabras, un estimador insesgado (como la media) tiende a proveer una estimación cercana al parámetro real.

En términos generales, el intervalo de confianza para un estimador puntual insesgado cuya distribución es cercana a la normal (como la media, las proporciones o la diferencia de medias) está dado por la ecuación 4.6, donde z^* se escoge de manera tal que se condiga con el nivel de confianza seleccionado y la lateralidad de la hipótesis alternativa. Como se dijo anteriormente, el valor $z^* \cdot SE_{\hat{\theta}}$ se denomina “margen de error”. Debemos recordar que la ecuación 4.2 corresponde al error estándar de la media, pero los errores estándar para otros estimadores puntuales se estiman de manera diferente a partir de los datos.

$$\hat{\theta} \pm z^* \cdot SE_{\hat{\theta}} \quad (4.6)$$

El método de prueba de hipótesis usando valores p puede generalizarse para otros estimadores puntuales con distribución cercana a la normal. Para ello, Diez et al. (2017, p. 199) señalan que se deben considerar los siguientes pasos:

Prueba de hipótesis usando el modelo normal:

1. Formular las hipótesis nula (H_0) y alternativa (H_A) en lenguaje llano y luego en notación matemática.
2. Identificar un estimador puntual (estadístico) adecuado e insesgado para el parámetro de interés.
3. Verificar las condiciones para garantizar que la estimación del error estándar sea razonable y que la distribución muestral del estimador puntual siga aproximadamente una distribución normal.
4. Calcular el error estándar. Luego, graficar la distribución muestral del estadístico bajo el supuesto de que H_0 es verdadera y sombrear las áreas que representan el valor p.
5. Usando el gráfico y el modelo normal, calcular el valor p para evaluar las hipótesis y escribir la conclusión en lenguaje llano.

4.6.2 Estimadores con otras distribuciones

Existen métodos de construcción de intervalos de confianza y prueba de hipótesis adecuados para aquellos casos en que el estimador puntual o el estadístico de prueba no son cercanos a la normal (por ejemplo, si la muestra es pequeña, se tiene una mala estimación del error estándar o el estimador puntual tiene una distribución distinta a la normal). No obstante, la selección de métodos alternativos debe hacerse siempre teniendo en cuenta la distribución muestral del estimador puntual o del estadístico de prueba.

Una consideración importante es que **siempre debemos verificar el cumplimiento de las condiciones requeridas por una herramienta estadística**, pues de lo contrario las conclusiones pueden ser erradas y carecerán de validez.

4.7 EJERCICIOS PROPUESTOS

1. ¿Es correcto afirmar que, si se lanza un dado una y otra vez, la media móvil simple del número de puntos que aparecen en la cara superior crece monótonamente? Justifica tu respuesta.
2. ¿Es correcto afirmar que, si se lanza un dado una y otra vez, la proporción de veces que aparece un número impar de puntos (1, 3 o 5) en la cara superior es siempre 0,5? Justifica su respuesta.
3. Si se calcula la media de diez muestras distintas extraídas de la misma población, ¿se espera ver el mismo valor cada vez? ¿Cómo se llama a este fenómeno?
4. Completa las siguientes oraciones:
 - a) Una estimación _____ es un _____ calculado con datos de una muestra como aproximación del valor desconocido de un _____ de la población en estudio.
 - b) \bar{X} o \bar{x} se usan para denotar la _____, que es una estimación puntual de μ , la _____.
5. Se sabe que una prueba para medir el coeficiente intelectual de jóvenes de 18 años produce puntuaciones que siguen una distribución $\mathcal{N}(\mu = 100, \sigma^2 = 100)$.
 - a) Dibuja el histograma de la distribución muestral de medias para muestras de tamaño 25 de esta población.
 - b) Una de las muestras anteriores presentó $\bar{x} = 95$ y $s = 15$. Determina el intervalo con 95 % de confianza para este caso.
 - c) Con otra de las muestras se pudo determinar que su intervalo con 99 % de confianza era [90, 26; 105, 74]. ¿Qué significa esto?
 - d) El intervalo anterior, ¿es más grande o más pequeño que uno con 90 % de confianza?
6. Una empresa de tecnología quiere promocionar un software especializado para almacenar y recuperar imágenes médicas digitales. Con esta idea, está financiando un estudio para determinar el tiempo (en segundos) que necesita un grupo de médicos para recuperar imágenes desde sus propios registros en sus portátiles personales y desde la base de datos central con el software ofrecido y una conexión a la Web.
 - a) Enuncia las hipótesis nula y alternativa (en castellano común).
 - b) Identifica la variable aleatoria que se va a estudiar, el parámetro de interés y el correspondiente estadístico.
 - c) Enuncia, más formalmente, las hipótesis nula y alternativa para este caso.
 - d) Supón que el intervalo con 95 % de confianza para el tiempo de recuperación promedio de una imagen digital desde la base de datos central resultó ser [24; 36] [s]. ¿Qué decisión tomarías ante la hipótesis nula: la media del tiempo de recuperación de una imagen digital con el nuevo software es de 25 segundos? En este caso, ¿cuál podría ser la hipótesis alternativa?
 - e) Para el intervalo de confianza anterior, ¿cuál sería un error de tipo I?
 - f) Conociendo el intervalo de confianza anterior, ¿es posible cometer un error de tipo II? Explica.
7. Si una hipótesis nula es falsa, aumentar el nivel de significación para un tamaño de muestra dado, ¿reduce la probabilidad de rechazarla?
8. ¿Qué significa que un estadístico tenga un valor p de 0,025?
9. Si una hipótesis nula es rechazada a un nivel de significación de 0,01, ¿será rechazada a un nivel de significación 0,05? Explica.
10. Si una hipótesis nula es rechazada por una prueba unilateral (una cola), ¿será también rechazada por una prueba bilateral (dos colas)? Explica.
11. Acabas de leer un artículo que hace la siguiente aseveración: “*a 95 % confidence interval for mean reaction time is from 0.25 to 0.29 seconds. Thus, about 95 % of individuals will have reaction times in this interval.*” Comenta.
12. Da el ejemplo de un estudio en que es más dañino cometer un error tipo II que un error tipo I.
13. Lista las condiciones que deben verificarse para asegurar que el TLC (teorema del límite central) está rigiendo y es posible hacer una prueba de hipótesis o calcular un intervalo de confianza.
14. Si para un estudio de una determinada variable aleatoria numérica es igualmente dañino cometer errores de tipo I como errores tipo II:
 - a) Dibuja la distribución de una muestra de tamaño 16 (un diagrama de caja, por ejemplo) para la que el contraste de hipótesis con nivel de significación 0,05 sea confiable.

- b)* Dibuja la distribución de una muestra de tamaño 30 en que se requiera de un nivel de significación más exigente ($\alpha < 0,05$) para hacer el contraste de hipótesis más confiable.
- c)* Dibuja la distribución de una muestra en que es mejor no confiar en el contraste de hipótesis con métodos estudiados hasta ahora.
15. Si un estudio sobre el tiempo promedio de búsqueda y recuperación de imágenes médicas con dos tecnologías distintas reporta: “existe una diferencia significativa ($p < 0,02$) entre el tiempo invertido con la tecnología A ($33 \pm 4[s]$) que con la tecnología B ($30 \pm 6[s]$)”, ¿significa que se debe adoptar la tecnología B? ¿Por qué?
16. Explica por qué se incrementa la probabilidad de cometer errores tipo I al cambiar de una prueba de hipótesis bilateral a otra unilateral.

CAPÍTULO 5. INFERENCIA CON MEDIAS MUESTRALES

En el capítulo 4 conocimos los principios de la inferencia y definimos los principales conceptos involucrados. En dicho capítulo conocimos el modelo normal, es decir, que la distribución muestral de la media sigue aproximadamente una distribución normal, supuesto que en general se cumple si la muestra tiene a lo menos 30 observaciones.

Veremos que diversas pruebas estadísticas consideran el modelo normal, aunque otras consideran estadísticos (estimaciones puntuales) diferentes que siguen otras distribuciones que ya conocimos en el capítulo 3.

En este capítulo veremos nuestras primeras pruebas estadísticas, las cuales nos permitirán inferir acerca de una o dos medias muestrales. Para ello nos basaremos principalmente en las explicaciones que ofrecen Diez et al. (2017, pp. 219-239) y Meena (2020).

5.1 PRUEBA Z

Como ya adelantamos, la prueba Z es adecuada para inferir acerca de las medias con una o dos muestras, aunque aquí solo veremos el primer caso. Para poder usarla, debemos **verificar el cumplimiento** de algunas condiciones, muchas de las cuales están asociadas al modelo normal que conocimos en el capítulo anterior:

- La muestra debe tener al menos 30 observaciones. Si la muestra tiene menos de 30 observaciones, se debe conocer la varianza de la población.
- Las observaciones deben ser independientes, es decir que la elección de una observación para la muestra no influye en la selección de las otras.
- La población de donde se obtuvo la muestra sigue aproximadamente una distribución normal.

Esta prueba resulta adecuada si queremos **asegurar** o **descartar** que la media de la población tiene un cierto **valor hipotético**. Esteban Quito es gerente de un grupo de inversiones que actualmente brinda apoyo financiero a más de 300 pequeñas empresas. El Sr. Quito desea saber si, en promedio, las utilidades obtenidas el mes pasado por las empresas a las que brinda apoyo fueron de 20 millones de pesos. Para ello, nos ha informado que la desviación estándar para las utilidades de las empresas durante el mes pasado es de 2,32 millones de pesos y nos ha proporcionado una muestra, obtenida mediante muestreo aleatorio simple, con las utilidades (en millones de pesos) reportadas por 20 de las empresas durante dicho periodo, que se muestra en la tabla 5.1.

Empresa	Utilidad [M\$]						
1	19,33	6	22,22	11	22,55	16	29,68
2	29,37	7	31,26	12	20,69	17	29,27
3	29,14	8	26,92	13	24,68	18	26,72
4	32,10	9	31,40	14	28,74	19	27,08
5	25,04	10	17,66	15	26,85	20	20,62

Tabla 5.1: muestra para el ejemplo de prueba Z con una muestra.

El Sr. Quito nos ha dicho que debemos ser muy exigentes con respecto a nuestras conclusiones, por lo que se decide usar un nivel de significación $\alpha = 0,01$ (es decir, un nivel de confianza de 99 %).

Comencemos por formular nuestras hipótesis:

H_0 : la media de las utilidades obtenidas por las empresas el mes pasado (μ) es de 20 millones de pesos, es decir: $\mu = 20$ [M\$].

H_A : las utilidades obtenidas el mes pasado por las empresas son, en promedio, distintas de 20 millones de pesos, es decir: $\mu \neq 20$ [M\$].

Ahora debemos verificar el cumplimiento de las condiciones para poder usar la prueba Z. En cuanto a la primera condición, el enunciado nos indica que, si bien la muestra tiene solo 20 observaciones, la desviación estándar de la población es conocida, por lo que se verifica su cumplimiento.

También podemos comprobar en el enunciado que las observaciones son independientes entre sí, pues fueron obtenidas mediante muestreo aleatorio simple y corresponden a menos del 10 % de la población.

En cuanto a la distribución de la muestra, el gráfico Q-Q de la figura 5.1 (obtenido mediante el script 5.1) nos muestra que no se observan valores atípicos. Otra forma de comprobar esta condición es mediante la prueba de Shapiro-Wilk (Parada, 2019), que podemos realizar en R mediante la función `shapiro.test(x)`, donde x es un vector con las observaciones de la muestra. La hipótesis nula de esta prueba es que la muestra fue extraída desde una distribución normal (por ende, la hipótesis alternativa es que la distribución detrás de la muestra es diferente a la normal). Al ejecutar el script, podemos ver que el valor p obtenido es $p = 0,244$, muy superior a nuestro nivel de significación, por lo que podemos suponer con relativa confianza que la población de donde proviene la muestra sigue una distribución muestral.

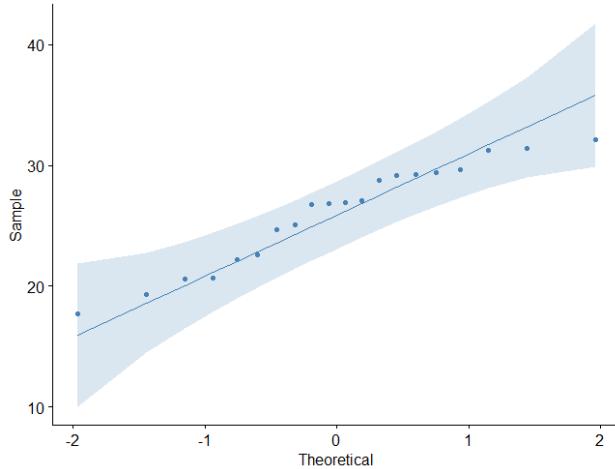


Figura 5.1: gráfico Q-Q para la muestra de la tabla 5.1.

Puesto que hemos comprobado que se cumplen todas las condiciones, podemos hacer una prueba Z para una muestra. Comencemos por calcular ahora el **estadístico de prueba** como ya hemos estudiado, usando para ello la ecuación 3.7:

$$Z = \frac{\bar{x} - \mu}{SE} = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}} = \frac{26,066 - 20}{\frac{2,32}{\sqrt{20}}} = 11.69309$$

Con este resultado calculamos el valor p. Debemos recordar que las funciones de R (al igual que las antiguas tablas de probabilidades) nos entregan la probabilidad asociada al área correspondiente a una sola cola de la distribución, por lo que debemos multiplicar el resultado por 2 para considerar ambas colas si, como en este caso, se trata de una prueba bilateral. Al hacer la llamada `2 * pnorm(2.6147, lower.tail = FALSE)`, obtenemos que $p = 1,382574 \times 10^{-31} < 0.01^1$, con lo que se rechaza la hipótesis nula en favor de la hipótesis alternativa. En este caso, el valor p obtenido es mucho menor que el nivel de significación establecido. Así,

¹Por convención, los valores p suelen reportarse con tres decimales, pero en este caso presentamos el resultado con detalle por claridad.

concluimos que los datos **sugieren** con 99 % de confianza que, en promedio, las utilidades obtenidas por las empresas durante el mes pasado difieren de los 20 millones de pesos establecidos.

Un comentario importante es que, si hubiésemos obtenido, por ejemplo, $p = 0,009$, deberíamos ser cuidadosos puesto que dicho valor es bastante cercano al nivel de significación establecido, por lo que sería prudente evaluar los resultados con una muestra más grande.

Desde luego, gracias a R podemos realizar esta prueba simplemente con una llamada a la función `z.test(x, mu, stdev, alternative, conf.level)`, disponible en el paquete `TeachingDemos`, donde:

- `x`: vector con las observaciones de la muestra.
- `mu`: valor nulo.
- `stdev`: desviación estándar de la población.
- `alternative`: tipo de hipótesis alternativa. Puede tomar los valores “`two.sided`” (hipótesis bilateral), “`less`” (hipótesis unilateral que la media de la población es menor que el valor nulo) o “`greater`” (hipótesis unilateral que la media de la población es mayor que el valor nulo).
- `conf.level`: nivel de confianza.

El script 5.1 muestra el desarrollo de este ejemplo en forma manual y luego, en las líneas 42 y 49, dos alternativas equivalentes usando la función `z.test()`. El resultado que se obtiene al usar esta función es el que se muestra en la figura 5.2, idéntico al obtenido en nuestro desarrollo previo.

One Sample z-test

```
data: media
z = 11.693, n = 20.00000, Std. Dev. = 2.32000, Std. Dev. of the sample mean = 0.51877,
p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 20
99 percent confidence interval:
24.72974 27.40226
sample estimates:
mean of media
26.066
```

Figura 5.2: resultado de la prueba Z para una muestra.

Script 5.1: prueba Z para una muestra.

```
1 library(TeachingDemos)
2 library(ggpubr)
3
4 # Ingresar los datos.
5 muestra <- c(19.33, 29.37, 29.14, 32.10, 25.04, 22.22, 31.26, 26.92,
6           31.40, 17.66, 22.55, 20.69, 24.68, 28.74, 26.85, 29.68,
7           29.27, 26.72, 27.08, 20.62)
8
9 # Establecer los datos conocidos.
10 desv_est <- 2.32
11 n <- length(muestra)
12 valor_nulo <- 20
13
14 # Crear gráfico Q-Q para verificar la distribución de la muestra,
15 datos <- data.frame(muestra)
16
17 g <- ggqqplot(datos, x = "muestra", color = "SteelBlue")
18 print(g)
19
20 # Verificar distribución muestral usando la prueba de normalidad
```

```

21 # de Shapiro-Wilk.
22 normalidad <- shapiro.test(muestra)
23 print(normalidad)
24
25 # Fijar un nivel de significación.
26 alfa <- 0.01
27
28 # Calcular la media de la muestra.
29 cat("\tPrueba Z para una muestra\n\n")
30 media <- mean(muestra)
31 cat("Media =", media, "M$\n")
32
33 # Calcular el estadístico de prueba.
34 Z <- (media - valor_nulo) / (desv_est / sqrt(n))
35 cat("Z =", Z, "\n")
36
37 # Calcular el valor p.
38 p <- 2 * pnorm(Z, lower.tail = FALSE)
39 cat("p =", p, "\n")
40
41 # Hacer la prueba Z con R.
42 # Una alternativa es usando la media muestral y el tamaño de la muestra.
43 prueba1 <- z.test(media, mu = valor_nulo, n = 20, alternative = "two.sided",
44 stdev = desv_est, conf.level = 1-alfa)
45
46 print(prueba1)
47
48 # Otra opción es usando la muestra directamente.
49 prueba2 <- z.test(muestra, mu = valor_nulo, alternative = "two.sided",
50 stdev = desv_est, conf.level = 1-alfa)
51
52 print(prueba2)

```

5.2 PRUEBA T DE STUDENT

En la práctica, rara vez podemos conocer la desviación estándar de la población y a menudo nos encontraremos con muestras pequeñas, por lo que la prueba Z no es muy utilizada.

En el caso de la media, el teorema del límite central se cumple para datos normales, es decir, independientemente del tamaño de la muestra, la media muestral tendrá una distribución cercana a la normal siempre que las observaciones sean independientes y provengan de una distribución cercana a la normal. Sin embargo, cuando el conjunto de datos es pequeño, resulta muy difícil comprobar el cumplimiento de estas condiciones.

En el capítulo 3 conocimos la distribución t de Student, o simplemente distribución t. Vimos que un aspecto destacado de esta distribución, siempre centrada en 0 y definida únicamente por los grados de libertad (ν) como parámetro, es su semejanza con la distribución normal pese a que sus colas son algo más gruesas. Este grosor adicional de las colas tiene como consecuencia que, para la distribución t, es más probable que una observación esté a más de dos desviaciones estándares de la media que en el caso de la distribución normal. Este fenómeno permite que la estimación del error estándar sea más certera que al usar la distribución normal cuando el conjunto de datos es pequeño.

La prueba t de Student, basada en la distribución t, es en consecuencia la alternativa más ampliamente empleada para inferir acerca de una o dos medias muestrales.

5.2.1 Prueba t para una muestra

Aunque la prueba t no opera bajo el supuesto de normalidad, aún así requiere verificar algunas condiciones para poder usarla:

1. Las observaciones son independientes entre sí.
2. Las observaciones provienen de una distribución cercana a la normal.

Podemos ver que estas condiciones son casi las mismas que para la prueba Z, excepto por el hecho de que no limitan el tamaño de la muestra para que sea mayor a 30. La ventaja evidente de eliminar esta restricción es que la distribución t permite su uso para muestras pequeñas, pero es igualmente adecuada cuando la muestra es grande. Esto se debe a que la forma de la distribución t es regulada por los grados de libertad y, a medida que aumentan, más se parece a una distribución normal. Este parámetro, al trabajar con medias de muestras de tamaño n , siempre estará dado por $\nu = n - 1$.

Tomemos el siguiente problema para ilustrar la prueba de hipótesis para la media de una muestra usando el modelo t: un ingeniero en Informática necesita determinar si el tiempo promedio que tarda una implementación dada de un algoritmo en resolver un problema, sabiendo que el algoritmo siempre se ejecuta en las mismas condiciones (misma máquina, igual disponibilidad de recursos de hardware y tamaño constante de las instancias), es inferior a 500 milisegundos. Para ello, ha seleccionado aleatoriamente 15 instancias del problema y registrado el tiempo de ejecución del algoritmo (en milisegundos) para cada una de ellas, como muestra la tabla 5.2.

Obs.	t [ms]	Obs.	t [ms]	Obs.	t [ms]
1	411,5538	6	388,6731	11	418,1169
2	393,2753	7	430,0382	12	408,4110
3	445,8905	8	469,4734	13	463,3733
4	411,4022	9	409,5844	14	407,0908
5	498,8969	10	442,0800	15	516,5222

Tabla 5.2: tiempo de ejecución para las instancias de la muestra.

El primer paso es formular las hipótesis:

H_0 : el tiempo promedio que tarda el algoritmo en resolver una instancia del problema es igual a 500 milisegundos.

H_A : el tiempo promedio que tarda el algoritmo en resolver una instancia del problema es inferior a 500 milisegundos.

Recordemos que μ_0 es el valor nulo, por lo que en este caso $\mu_0 = 500$ [ms]. Matemáticamente, las hipótesis anteriores pueden formularse como:

Denotando como μ al tiempo medio que tarda la implementación del algoritmo en resolver una instancia cualquiera del problema:

$H_0: \mu = \mu_0$, esto es $\mu = 500$

$H_A: \mu < \mu_0$, es decir $\mu < 500$

Ahora debemos verificar que se cumplen las condiciones necesarias para usar la distribución t:

- Como las muestras fueron elegidas al azar, se puede asumir que son independientes.
- El gráfico de la figura 5.3 muestra que es válido suponer una distribución cercana a la normal. Si bien los puntos de la muestra no forman una recta, no se observan valores atípicos que se alejen de la región aceptable.

La media de la muestra es de $\bar{x} = 434,2921$, y la desviación estándar, $s = 38,0963$.

En este caso, el estadístico de prueba es el estadístico T, el cual sigue una distribución t con $\nu = n - 1$ grados de libertad y está dado por la ecuación 5.1, donde la subexpresión (s/\sqrt{n}) corresponde al error estándar de la media (cuando no se conoce la desviación estándar de la población, σ).

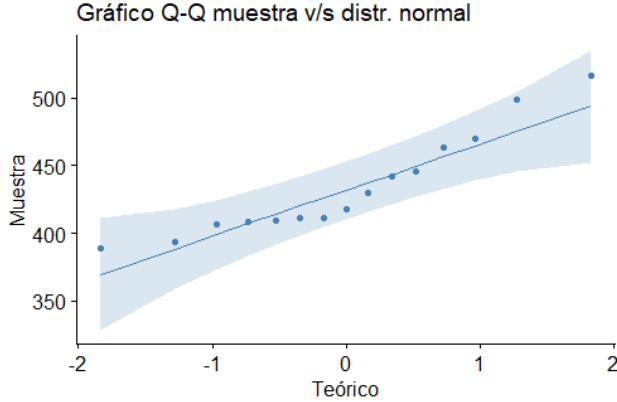


Figura 5.3: gráfico para comprobar el supuesto de normalidad.

$$T = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \quad (5.1)$$

Así, para el ejemplo tenemos que:

$$T = \frac{434,2921 - 500}{\frac{38,0963}{\sqrt{15}}} = -6,6801$$

A partir de este resultado, obtenemos el valor p con ayuda de la función `pt()`, obteniéndose $p = 5,219 \cdot 10^{-6}$, o simplemente, como dicta la convención, $p < 0,001$.

La fórmula para construir el intervalo de confianza usando la distribución t es ligeramente diferente al caso normal, como muestra la ecuación 5.2. Para este ejemplo consideraremos un nivel de confianza de 97,5 % (es decir, un nivel de significación $\alpha = 0,025$).

$$\bar{x} \pm t_{\nu}^* \cdot SE \quad (5.2)$$

Fijémonos en que en la ecuación 5.2 aparece el nuevo valor t_{ν}^* , el cual se obtiene a partir del nivel de confianza y la distribución t con ν grados de libertad (en este caso, $\nu = 14$), usando para ello una tabla de distribución t o la función `qt()` en R. Como puede verse al ejecutar el script 5.2, en este caso $t_{\nu}^* = 2,1448$.

Para el cálculo del error estándar, nuevamente se emplea la ecuación 4.2:

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}} = \frac{38,0963}{\sqrt{15}} = 9,8364$$

Así, el intervalo de confianza está dado por:

$$(-\infty, t_{nu}^* * SE_{\bar{x}}] = (-\infty, 2,1448 * 9,8364] = (-\infty, 455,3892]$$

Una vez más, R permite realizar esta prueba de manera rápida y sencilla, gracias a la función `t.test(x, alternative, mu, conf.level)`, donde:

- `x`: vector no vacío de valores numéricos (la muestra).
- `alternative`: tipo de prueba de hipótesis. Los posibles valores son “`two.sided`” (prueba bilateral), “`greater`” (hipótesis unilateral que la media de la población es mayor que el valor nulo) o “`less`” (hipótesis unilateral que la media de la población es menor que el valor nulo).

- `mu`: valor nulo.
- `conf.level`: nivel de confianza.

El script 5.2 muestra el desarrollo en R para este ejemplo, incluyendo la construcción del gráfico de la figura 5.3, con iguales resultados al realizar la prueba paso a paso y con la función `t.test()`.

A partir de los resultados podemos observar que el valor p obtenido es muy pequeño, dando a entender que, si se cumple el supuesto de que la verdadera media es $\mu = 500$ [ms] (hipótesis nula), sería muy improbable obtener una media muestral de $\bar{x} = 434,2921$. Además, el valor p es muchísimo menor que el nivel de significación, por lo que la evidencia a favor de H_A es muy fuerte. En consecuencia, se rechaza H_0 en favor de H_A . Se puede afirmar, con 97,5 % de confianza, que el tiempo promedio que tarda el algoritmo en resolver una instancia del problema es inferior a 500 milisegundos.

Script 5.2: prueba t para una muestra.

```

1 library(ggpubr)
2
3 # Cargar los datos.
4 tiempo <- c(411.5538, 393.2753, 445.8905, 411.4022, 498.8969,
5           388.6731, 430.0382, 469.4734, 409.5844, 442.0800,
6           418.1169, 408.4110, 463.3733, 407.0908, 516.5222)
7
8 # Establecer los datos conocidos.
9 n <- length(tiempo)
10 grados_libertad <- n - 1
11 valor_nulo <- 500
12
13
14 # Verificar si la distribución se acerca a la normal.
15 g <- ggqqplot(data = data.frame(tiempo),
16                 x = "tiempo",
17                 color = "steelblue",
18                 xlab = "Teórico",
19                 ylab = "Muestra",
20                 title = "Gráfico Q-Q muestra v/s distr. normal")
21
22 print(g)
23
24 # Fijar un nivel de significación.
25 alfa <- 0.025
26
27 # Calcular el estadístico de prueba.
28 cat("\tPrueba t para una muestra\n\n")
29 media <- mean(tiempo)
30 cat("Media =", media, "M$\n")
31 desv_est <- sd(tiempo)
32 error <- desv_est / sqrt(n)
33 t <- (media - valor_nulo) / error
34 cat("t =", t, "\n")
35
36 # Calcular el valor p.
37 p <- pt(t, df = grados_libertad, lower.tail = TRUE)
38 cat("p =", p, "\n")
39
40 # Construir el intervalo de confianza.
41 t_critico <- qt(alfa, df = grados_libertad, lower.tail = FALSE)
42 superior <- media + t_critico * error
43 cat("Intervalo de confianza = (-Inf, ", superior, "]\n", sep = "")
44
45 # Aplicar la prueba t de Student con la función de R.

```

```

46 prueba <- t.test(tiempo,
47             alternative = "less",
48             mu = valor_nulo,
49             conf.level = 1 - alfa)
50
51 print(prueba)

```

5.2.2 Prueba t para dos muestras pareadas

Para esta prueba, supongamos ahora que el ingeniero en Informática del ejemplo anterior tiene dos algoritmos diferentes (A y B) que, en teoría, deberían tardar lo mismo en resolver un problema. Para ello, probó ambos algoritmos con 35 instancias del problema (elegidas al azar) de igual tamaño y registró los tiempos de ejecución (en milisegundos) de ambos algoritmos bajo iguales condiciones para cada una de ellas, además de calcular la diferencia en los tiempos de ejecución, como muestra la tabla 5.3. El ingeniero desea comprobar si efectivamente el rendimiento de ambos algoritmos es equivalente.

instancia	t_A [ms]	t_B [ms]	dif [ms]	instancia	t_A [ms]	t_B [ms]	dif [ms]
1	436,5736	408,5142	28,0594	19	438,5959	458,2536	-19,6577
2	470,7937	450,1075	20,6862	20	439,7409	474,9863	-35,2454
3	445,8354	490,2311	-44,3957	21	464,5916	496,0153	-31,4237
4	470,9810	513,6910	-42,7100	22	467,9926	485,8112	-17,8186
5	485,9394	467,6467	18,2927	23	415,3252	457,4253	-42,1001
6	464,6145	484,1897	-19,5752	24	495,4094	483,3700	12,0394
7	466,2139	465,9334	0,2805	25	493,7082	510,7131	-17,0049
8	468,9065	502,6670	-33,7605	26	433,1082	467,5739	-34,4657
9	473,8778	444,9693	28,9085	27	445,7433	482,5621	-36,8188
10	413,0639	456,3341	-43,2702	28	515,2049	453,5986	61,6063
11	496,8705	501,1443	-4,2738	29	441,9420	385,9391	56,0029
12	450,6578	471,7833	-21,1255	30	472,1396	548,7884	-76,6488
13	502,9759	441,1206	61,8553	31	451,2234	467,2533	-16,0299
14	465,6358	544,1575	-78,5217	32	476,5149	494,7049	-18,1900
15	437,6397	447,8844	-10,2447	33	440,7918	451,9716	-11,1798
16	458,8806	432,4108	26,4698	34	460,1070	522,3699	-62,2629
17	503,1435	477,1712	25,9723	35	450,1008	444,1270	5,9738
18	430,0524	482,4828	-52,4304				

Tabla 5.3: tiempos de ejecución de cada algoritmo para las instancias de la muestra.

Para este ejemplo, tenemos dos tiempos de ejecución diferentes para cada instancia del problema: uno con cada algoritmo. En consecuencia, los datos están **pareados**. Es decir, cada observación de un conjunto tiene una correspondencia o conexión especial con exactamente una observación del otro. Una forma de uso común para examinar datos pareados es usar la diferencia entre cada par de observaciones, para lo cual podemos usar la técnica de la distribución t (también llamada prueba t de Student) vista en la sección anterior.

La media de las diferencias es $\bar{x}_{dif} = -12,08591$ y la desviación estándar es $s_{dif} = 36,08183$.

Una vez más, comenzamos por formular las hipótesis:

H_0 : la media de las diferencias en los tiempos de ejecución es igual a 0.

H_A : la media de las diferencias en los tiempos de ejecución es distinta de 0.

Que matemáticamente se expresan como:

Denotando la media de las diferencias en los tiempos de ejecución necesitados por ambos algoritmos para cualquier instancia del problema como μ_{dif} :

$$H_0: \mu_{dif} = 0$$

$$H_A: \mu_{dif} \neq 0$$

Como siguiente paso, verificamos el cumplimiento de las condiciones. Como las instancias fueron escogidas al azar, se puede suponer razonablemente que las observaciones son independientes, pues además el conjunto de instancias posibles es muy grande (o infinito) y las 35 seleccionadas no superan el 10 % de la población. Además, al aplicar una prueba de normalidad de Shapiro-Wilk (ver script 5.3, línea 23) se obtiene $p = 0,357$, con lo que podemos concluir que la diferencia en los tiempos de ejecución se acerca razonablemente a una distribución normal. En consecuencia, podemos proceder con la prueba t de Student. El ingeniero no necesita ser especialmente riguroso, por lo que usaremos un nivel de confianza del 95 %.

En este caso, la función `t.test()` de R permite efectuar la prueba de dos maneras diferentes (con idéntico resultado), como muestra el script 5.3. La primera de ellas (línea 32) es aplicar la prueba t directamente a las diferencias, tal como en la sección anterior (es decir, una prueba t para una muestra). La segunda (línea 41) consiste en entregar a la función ambas muestras por separado e indicarle que están pareadas. En este caso, la llamada tiene la forma `t.test(x, y, paired, alternative, mu, conf.level)`, donde los argumentos son:

- `x`: vector de valores numéricos para la primera muestra.
- `y`: vector de valores numéricos para la segunda muestra.
- `paired`: booleano (por defecto falso) que, cuando es verdadero, indica que ambas muestras están pareadas.
- `alternative`: tipo de prueba de hipótesis.
- `mu`: valor nulo.
- `conf.level`: nivel de confianza.

Script 5.3: inferencia con la media de las diferencias entre dos muestras pareadas usando la distribución t.

```

1 # Cargar los datos.
2 instancia <- seq(1, 35, 1)
3
4 t_A <- c(436.5736, 470.7937, 445.8354, 470.9810, 485.9394,
5      464.6145, 466.2139, 468.9065, 473.8778, 413.0639,
6      496.8705, 450.6578, 502.9759, 465.6358, 437.6397,
7      458.8806, 503.1435, 430.0524, 438.5959, 439.7409,
8      464.5916, 467.9926, 415.3252, 495.4094, 493.7082,
9      433.1082, 445.7433, 515.2049, 441.9420, 472.1396,
10     451.2234, 476.5149, 440.7918, 460.1070, 450.1008)
11
12 t_B <- c(408.5142, 450.1075, 490.2311, 513.6910, 467.6467,
13     484.1897, 465.9334, 502.6670, 444.9693, 456.3341,
14     501.1443, 471.7833, 441.1206, 544.1575, 447.8844,
15     432.4108, 477.1712, 482.4828, 458.2536, 474.9863,
16     496.0153, 485.8112, 457.4253, 483.3700, 510.7131,
17     467.5739, 482.5621, 453.5986, 385.9391, 548.7884,
18     467.2533, 494.7049, 451.9716, 522.3699, 444.1270)
19
20 diferencia <- t_A - t_B
21
22 # Verificar si la distribución se acerca a la normal.
23 normalidad <- shapiro.test(diferencia)
24 print(normalidad)
25
26 # Fijar un nivel de significación.
27 alfa <- 0.05
28
29 # Aplicar la prueba t de Student a la diferencia de medias.

```

```

30 valor_nulo <- 0
31
32 prueba_1 <- t.test(diferencia,
33                         alternative = "two.sided",
34                         mu = valor_nulo,
35                         conf.level = 1 - alfa)
36
37 print(prueba_1)
38
39 # Otra alternativa puede ser aplicar la prueba t de Student
40 # para dos muestras pareadas.
41 prueba_2 <- t.test(x = t_A,
42                     y = t_B,
43                     paired = TRUE,
44                     alternative = "two.sided",
45                     mu = valor_nulo,
46                     conf.level = 1 - alfa)
47
48 print(prueba_2)

```

Los resultados para esta prueba son:

- El valor para el estadístico de prueba T es $t = -1,9816$.
- Se consideran $df = 34$ grados de libertad para la distribución t.
- El valor p obtenido es $p = 0,05565$.
- El intervalo de confianza obtenido es $[-24,4804542; 0,3086313]$.
- La media de la muestra es $\bar{x} = -12,08591$.

En este caso, la media de las diferencias está dentro del intervalo de confianza, y además el valor p es mayor que el nivel de significación, por lo que se falla al rechazar la hipótesis nula. Pero, nuevamente, el resultado está cerca del borde de significación. En consecuencia, se puede afirmar con 95 % de confianza que pareciera no haber suficiente evidencia para descartar que ambos algoritmos tardan, en promedio, lo mismo en procesar las instancias del problema, aunque sería necesario conseguir una muestra más grande para tener mayor certeza.

5.2.3 Prueba t para dos muestras independientes

En este caso, la prueba t se usa para comparar las medias de dos poblaciones en que las observaciones con que se cuenta no tienen relación con ninguna de las otras observaciones, ni influyen en su selección, ni en la misma ni en la otra muestra. En este caso la inferencia se hace sobre la diferencia de las medias: $\mu_1 - \mu_2 = d_0$, donde d_0 es un valor hipotético fijo para la diferencia. Usualmente se usa $d_0 = 0$, en cuyo caso las muestras podrían provenir de dos poblaciones distintas con igual media, o desde la misma población. Para ello, la prueba usa como estimador puntual la diferencia de las medias muestrales ($\bar{x}_1 - \bar{x}_2$). Así, el estadístico T en este caso toma la forma de la ecuación 5.3.

$$T = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{SE_{(\bar{x}_1 - \bar{x}_2)}} \quad (5.3)$$

Al usar la distribución t de Student para la diferencia de medias, se deben cumplir los siguientes requisitos:

1. Cada muestra cumple las condiciones para usar la distribución t.
2. Las muestras son independientes entre sí.

Veamos el funcionamiento de esta prueba con un ejemplo. El doctor E. L. Matta Sanno desea determinar si una nueva vacuna A es más efectiva que otra vacuna B, a fin de inmunizar a la población mundial contra una terrible enfermedad. Para ello, ha reclutado a un grupo de 28 voluntarios en diferentes países, 15 de los cuales (seleccionados al azar) recibieron la vacuna A y los 13 restantes, la vacuna B. La tabla 5.4 muestra, para cada voluntario, la concentración de anticuerpos (en microgramos por cada mililitro de sangre) al cabo de un mes de recibir la vacuna.

Anticuerpos [mg/ml]	
Vacuna A	Vacuna B
6,04	5,32
19,84	3,31
8,62	5,68
13,02	5,73
12,20	4,86
14,78	5,68
4,53	2,93
26,67	5,48
3,14	6,10
19,14	2,56
10,86	7,52
13,13	7,41
6,34	4,02
11,16	
7,62	

Tabla 5.4: Concentración de anticuerpos de los pacientes vacunados.

Las hipótesis a formular en este caso son:

H_0 : no hay diferencia entre la efectividad promedio de ambas vacunas.

H_A : la vacuna A es, en promedio, más efectiva que la B.

En lenguaje matemático:

Si μ_A y μ_B son la concentraciones medias de anticuerpos presentes en personas luego de un mes de recibir la vacuna A y B, respectivamente, entonces:

$H_0: \mu_A = \mu_B$

$H_A: \mu_A > \mu_B$

Como es habitual, debemos ahora verificar el cumplimiento de las condiciones. Ambas muestras son independientes entre sí, pues son diferentes voluntarios y fueron designados aleatoriamente a cada grupo. Además, se puede asumir que las observaciones son independientes, pues cada muestra es significativamente menor a la población total a vacunar. En cuanto al supuesto de normalidad para cada muestra, al aplicar a cada una la prueba de Shapiro-Wilk (script 5.4, líneas 13 y 15) se obtiene, respectivamente, $p = 0,428$ y $p = 0,445$. En ambos casos el valor p es bastante alto, por lo que podemos concluir que ambas muestras provienen de poblaciones que se distribuyen de forma aproximadamente normal. Puesto que hemos verificado las condiciones, podemos llevar a cabo la prueba t para dos muestras independientes.

Ahora bien, como las muestras son algo pequeñas, sería prudente proceder con algo más de cautela. Además, en este escenario, un error tipo I (rechazar H_0 cuando es verdadera) implicaría reducir innecesariamente la cantidad de vacunas disponibles y retrasar el proceso de vacunación, poniendo en riesgo a todos los habitantes del planeta. Un error tipo II, en cambio, podría causar que se continúe el uso indistinto de ambas vacunas retrasando ligeramente el efecto inmune en la población. En consecuencia, el error tipo I es más grave, por lo que el nivel de significación debiese ser aún más exigente. En consecuencia, optaremos por $\alpha = 0,01$.

Al aplicar la prueba t (script 5.4), obtenemos que la diferencia entre las medias es 6,683 [mg/ml] y que el intervalo de confianza es $[2,2739; \infty)$. Además, el valor p es $p < 0.001$, muy inferior al nivel de significación

$\alpha = 0,01$. Esto significa que la evidencia en favor de H_A es muy fuerte, por lo rechazamos la hipótesis nula. En consecuencia, podemos concluir con 99 % de confianza que la vacuna A es, en promedio, mejor que la vacuna B (produce una mayor concentración media de anticuerpos en las personas vacunadas con ella que la producida por la vacuna B).

Script 5.4: prueba t para dos muestras independientes.

```

1 library(ggpubr)
2
3 # Cargar los datos.
4 vacuna_A <- c(6.04, 19.84, 8.62, 13.02, 12.20, 14.78, 4.53, 26.67,
5           3.14, 19.14, 10.86, 13.13, 6.34, 11.16, 7.62)
6
7 vacuna_B <- c(5.32, 3.31, 5.68, 5.73, 4.86, 5.68, 2.93, 5.48, 6.10,
8           2.56, 7.52, 7.41, 4.02)
9
10 # Verificar si las muestras se distribuyen de manera cercana
11 # a la normal.
12 normalidad_A <- shapiro.test(vacuna_A)
13 print(normalidad_A)
14 normalidad_B <- shapiro.test(vacuna_B)
15 print(normalidad_B)
16
17 # Fijar un nivel de significación.
18 alfa <- 0.01
19
20 # Aplicar la prueba t para dos muestras independientes.
21 prueba <- t.test(x = vacuna_A,
22                   y = vacuna_B,
23                   paired = FALSE,
24                   alternative = "greater",
25                   mu = 0,
26                   conf.level = 1 - alfa)
27
28 print(prueba)
29
30 # Calcular la diferencia entre las medias.
31 media_A <- mean(vacuna_A)
32 media_B <- mean(vacuna_B)
33 diferencia <- media_A - media_B
34 cat("Diferencia de las medias =", diferencia, "[mg/ml]\n")

```

Si estás leyendo atentamente, te habrás dado cuenta que ¡no hemos definido el error estándar para cuando tenemos dos muestras! En este caso, SE se construye a partir del error estándar de cada muestra, como se aprecia en la ecuación 5.4. En este escenario, la determinación de los grados de libertad es más compleja, por lo que se recomienda usar programas estadísticos o, en su defecto, escoger el menor valor entre $n_1 - 1$ y $n_2 - 1$.

$$SE_{\bar{x}_1 - \bar{x}_2} = \sqrt{SE_{\bar{x}_1}^2 + SE_{\bar{x}_2}^2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \quad (5.4)$$

Se puede lograr un mejor ajuste de la distribución t si se sabe con certeza que las desviaciones estándares de ambas poblaciones son casi iguales. En este caso, se puede usar una **varianza agrupada** (s_p^2 , del inglés *pooled variance*) que reemplaza tanto a s_1^2 como a s_2^2 en la ecuación 5.4. Esta varianza agrupada se calcula como muestra la ecuación 5.5 y, en este caso, se consideran $n_1 + n_2 - 2$ grados de libertad.

$$s_p^2 = \frac{s_1^2 \cdot (n_1 - 1) + s_2^2 \cdot (n_2 - 1)}{n_1 + n_2 - 2} \quad (5.5)$$

Por defecto, R utiliza la corrección de Welch para la prueba t de Student de la diferencia de dos medias, variante considerada más segura, que en general entrega resultados muy similares a la versión original de la prueba cuando las muestras tienen varianzas similares. No obstante, los resultados son bastante mejores cuando los tamaños de las muestras y sus desviaciones estándares son muy diferentes (Kassambara, 2019a). La corrección de Welch calcula el error estándar como muestra la ecuación 5.4, pero ajusta los grados de libertad de acuerdo a la ecuación 5.6.

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2} \right)^2}{\frac{s_1^4}{n_1^2(n_1-1)} + \frac{s_2^4}{n_2^2(n_2-1)}} \quad (5.6)$$

5.3 EJERCICIOS PROPUESTOS

1. Investiga acerca de la prueba de Kolmogorov-Smirnov y explica cómo puede usarse para verificar si una distribución se asemeja a la normal. Compara esta prueba con la de Shapiro-Wilk.
2. Para confirmar que el tiempo que requieren los estudiantes de ingeniería para desarrollar una guía de ejercicios de Cálculo I es de dos horas, se eligió aleatoriamente a 16 estudiantes de esta asignatura y se les pidió anotar el tiempo [min.] invertido en la tarea. Los resultados fueron los siguientes: 140,6; 133,3; 142,4; 86,4; 129,9; 110,8; 133,2; 129,1; 142,5; 150,2; 141,6; 111,0; 127,2; 137,9; 131,9; 121,9.
 - a) Enuncia las hipótesis nula y alternativa a contrastar.
 - b) Analiza si es razonable en este caso considerar que los datos cumplen las condiciones para usar una prueba t de Student.
 - c) Independientemente del resultado anterior, aplica la prueba propuesta y obtenga un intervalo de confianza y un valor p.
 - d) Usando un nivel de significación adecuado, entrega una conclusión para la cuestión planteada.
3. El departamento de control de calidad de un importante laboratorio requiere analizar la concentración de ingredientes activos presente en una muestra de 10 botellas diferentes de detergente líquido que ellos seleccionaron aleatoriamente en el último mes. Como se sospecha que esta concentración depende del catalizador que se use, la mitad del contenido de cada botella fue sometida a un catalizador, y la otra mitad a otro catalizador. En orden por botella seleccionada, los resultados fueron:
 - Catalizador 1: 62,9; 67,2; 67,4; 67,4; 67,2; 64,6; 69,6; 65,7; 68,2; 72,0.
 - Catalizador 2: 66,8; 69,3; 69,6; 67,3; 68,8; 68,4; 68,6; 70,3; 69,6; 71,7.
 - a) Como primer paso, el departamento de control de calidad necesita saber si la concentración media de concentraciones de ingredientes activos depende del catalizador elegido.
 - b) Propón las hipótesis nula y alternativa que permitan responder el problema planteado con una prueba t de Student.
 - c) Muestra que es razonable considerar que estos datos cumplen las condiciones para usar la prueba propuesta y fija un nivel de significación apropiado.
 - d) Aplica la prueba propuesta y obtenga un intervalo de confianza y un valor p.
 - e) ¿Cuál sería tu respuesta al departamento de control mencionado?
4. Una fábrica de detectores de radón recibió consultas de sus clientes sobre si era conveniente comprar su nuevo modelo de detectores Radolmes+® para reemplazar los antiguos aparatos Radolmes® en su poder. Si bien los técnicos están seguros que la inversión es conveniente, la gerencia decidió hacer un estudio previo a la recomendación. Para esto, se introdujeron en una tómbola oscura los números de serie de los aparatos producidos en los últimos meses de ambos modelos y se seleccionaron 26 números sin mirar y girando la tómbola cinco veces entre cada selección, resultando escogidos 12 aparatos Radolmes y 14 aparatos Radolmes+. Luego, cada detector seleccionado se expuso a 100 pCi/l de radón. Las lecturas resultantes fueron las siguientes:
 - Radolmes: 105,6; 100,1; 90,9; 105,0; 91,2; 99,6; 96,9; 107,7; 96,5; 103,3; 91,3; 92,4.

- Radolmes+: 98,9; 94,3; 95,9; 107,7; 102,0; 94,2; 100,6; 98,5; 99,1; 101,3; 94,4; 103,6; 95,3; 106,7.
- a) ¿Qué hipótesis nula y alternativa se deberían docimar² con una prueba t de Student para responder a la inquietud planteada?
- b) ¿Cumplen los datos obtenidos las condiciones para usar esta prueba t de Student?
- c) Aplicando la prueba t de Student para este caso, obtén un intervalo de confianza y un valor p.
- d) ¿Qué aconsejarías a los directivos de la fábrica?

²Término que suele ocuparse en estadística como sinónimo de “probar”.

CAPÍTULO 6. PODER ESTADÍSTICO

En el capítulo 4 estudiamos el procedimiento para someter hipótesis a prueba, junto con los errores de decisión que podríamos cometer:

- Error tipo I: rechazar H_0 en favor de H_A cuando H_0 es en realidad verdadera.
- Error tipo II: no rechazar H_0 en favor de H_A cuando H_A es en realidad verdadera.

Allí conocimos el nivel de significación, α , como herramienta para representar y, de alguna manera, controlar la probabilidad de cometer un error de tipo I, con lo que la preocupación se centra en controlar la ocurrencia de esta clase de errores, desviando la atención de los errores de tipo II. Esto se debe a que la hipótesis nula representa el *status quo*, es decir, mantener las cosas y creencias tal como están y, por ende, cuando no se rechaza H_0 , no suele requerirse tomar ninguna acción. En contraste, la hipótesis alternativa describe un cambio de condiciones, por lo que rechazar H_0 en favor de H_A usualmente conlleva un esfuerzo, mayor costo, para adaptarse o aprovechar las nuevas condiciones.

Sin embargo, en el capítulo 4 también vimos que el valor de α debe ser acorde con las consecuencias de cometer errores tanto de tipo I como de tipo II, ¡pero no sabemos cómo se relaciona el nivel de significación con los errores de tipo II!

Así como el nivel de significación α corresponde a la probabilidad de cometer errores de tipo I, definimos ahora β como la probabilidad de cometer errores de tipo II. α y β están relacionados: **para un tamaño fijo de la muestra: al reducir β , α aumenta, y viceversa**. Este fenómeno se evidencia con mayor fuerza mientras más pequeña sea la muestra. No obstante, en la práctica resulta más interesante conocer la probabilidad de **no** cometer errores de tipo II. Esto nos lleva a un nuevo concepto: el **poder estadístico** de una prueba de hipótesis, dado por $1 - \beta$, que se define como **la probabilidad de correctamente rechazar H_0 cuando es falsa**.

Otra forma de entender la noción de poder de una prueba es qué tan propensa es esta para distinguir un efecto real de una simple casualidad, lo que nos lleva a la noción de **tamaño del efecto**, que corresponde a una cuantificación de la diferencia entre dos grupos, o del valor observado con respecto al valor nulo.

En el capítulo 5 conocimos la prueba t para inferir acerca de dos medias. En este contexto, el tamaño del efecto corresponde a qué tan grande es la diferencia real entre ambas. Si quieres aprender más sobre estos conceptos, puedes consultar las fuentes en las que se basa este capítulo: Diez et al. (2017, pp. 239-245) y Freund y Wilson (2003, pp. 123-138).

6.1 PODER, NIVEL DE SIGNIFICACIÓN Y TAMAÑO DE LA MUESTRA

En la introducción de este capítulo vimos que el poder corresponde a la probabilidad de **no** cometer un error de tipo II, y que está muy relacionado con el tamaño de la muestra. También mencionamos que existe una relación entre el poder y el nivel de significación, la cual exploraremos en esta sección.

La figura 6.1 (creada mediante el script 6.1) muestra cuatro curvas de poder para la prueba t de Student de una muestra con desviación estándar $s = 1$ y valor nulo $\mu_0 = 0$. En ella, el tamaño del efecto está representada en la misma escala de la variable, aunque en la sección siguiente veremos otra alternativa. La curva roja considera $\alpha = 0,05$ y $n = 6$; la azul, $\alpha = 0,01$ y $n = 6$; la verde, $\alpha = 0,05$ y $n = 10$, y la naranja, $\alpha = 0,01$ y $n = 10$. En ella podemos observar que:

- El poder de la prueba aumenta mientras mayor es el tamaño del efecto (en este caso, la distancia entre el valor nulo y la media de la muestra).
- A medida que el tamaño del efecto disminuye (es decir, el estimador se acerca al valor nulo), el poder se aproxima al nivel de significación.
- Usar un valor de α más exigente (menor), manteniendo constante el tamaño de la muestra, hace que la curva de poder sea más baja para cualquier tamaño del efecto (lo que verifica la relación entre α y β).
- Usar una muestra más grande aumenta el poder de la prueba para cualquier tamaño del efecto distinto de 0.

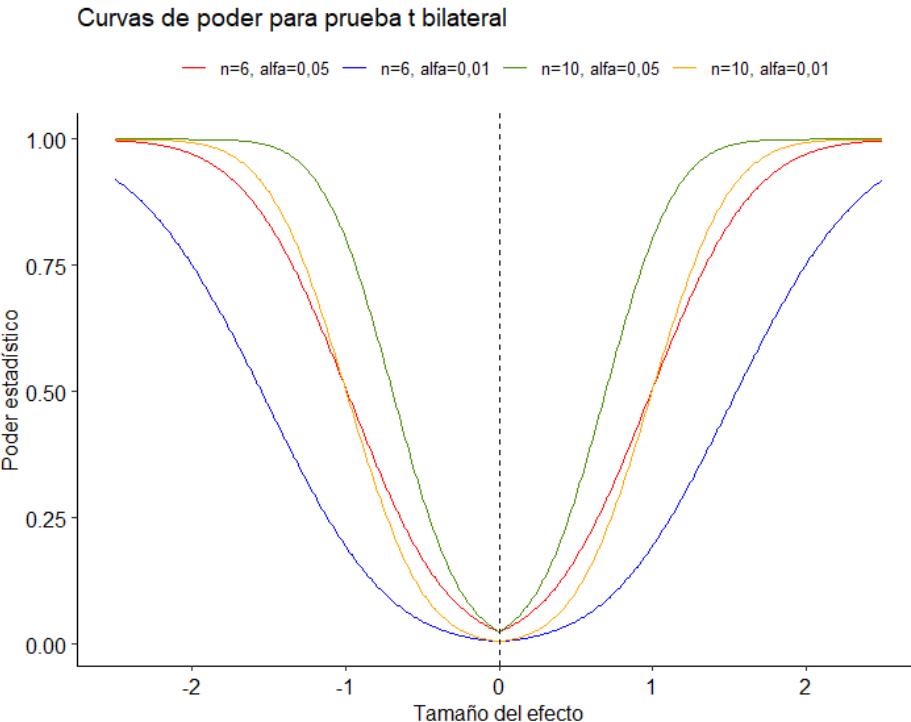


Figura 6.1: poder estadístico para prueba t bilateral.

De manera similar, la figura 6.2 considera las mismas muestras y los mismos niveles de significación que la figura 6.1, pero ahora para una prueba t unilateral. En ella se aprecia que la gran desventaja de las pruebas unilaterales es que el poder tiende a cero a medida que el tamaño del efecto aumenta en sentido contrario a la hipótesis alternativa, por lo que no sería posible detectar una diferencia en el sentido opuesto aunque fuese muy grande (pues no hay una región de rechazo en dicho sentido). El script empleado para la construcción de la figura 6.2 es idéntico al script 6.1, excepto porque el argumento `alternative` toma como valor “`one.sided`” en las llamadas a `power.t.test()`.

Script 6.1: poder estadístico para prueba t bilateral.

```

1 library(ggpubr)
2 library(tidyverse)
3
4 # Generar un vector con un rango de valores para la efecto
5 # de medias.
6 efecto <- seq(-2.5, 2.5, 0.01)
7
8 # Calcular el poder para una prueba t bilareral, para cada tamaño
9 # del efecto, asumiendo una muestra con desviación estàndar igual a 1.
10 # Se consideran 4 escenarios para calcular el poder:
11 # 1. Una muestra de tamaño 6 y nivel de significación 0.05.

```

Curvas de poder para prueba t bilateral

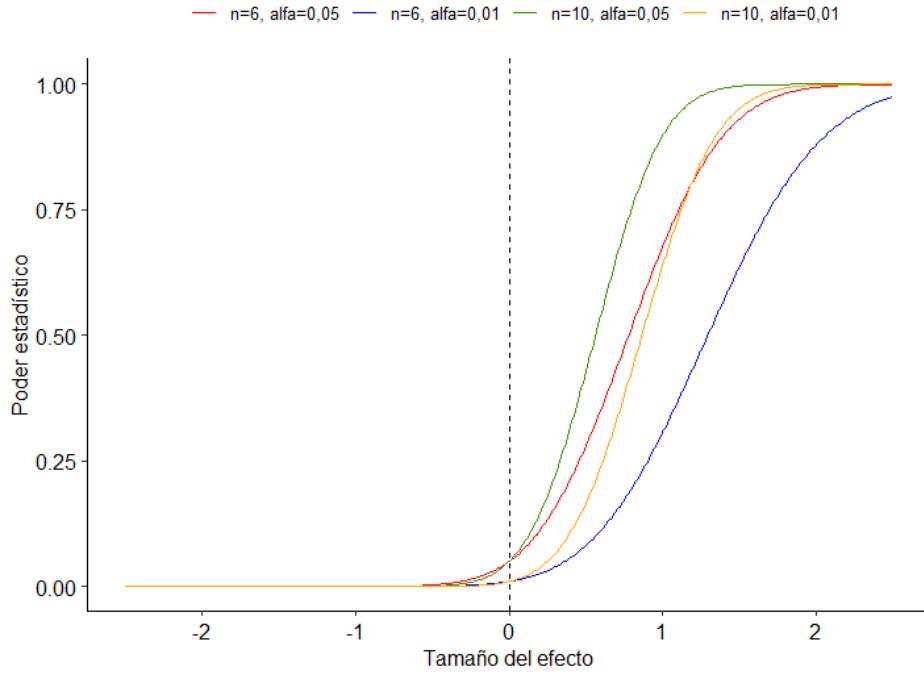


Figura 6.2: poder estadístico para prueba t unilateral.

```

12 # 2. Una muestra de tamaño 6 y nivel de significación 0.01.
13 # 3. Una muestra de tamaño 10 y nivel de significación 0.05.
14 # 4. Una muestra de tamaño 10 y nivel de significación 0.01.
15 n_6_alfa_05 <- power.t.test(n = 6,
16                               delta = efecto,
17                               sd = 1,
18                               sig.level = 0.05,
19                               type = "one.sample",
20                               alternative = "two.sided")$power
21
22 n_6_alfa_01 <- power.t.test(n = 6,
23                               delta = efecto,
24                               sd = 1,
25                               sig.level = 0.01,
26                               type = "one.sample",
27                               alternative = "two.sided")$power
28
29 n_10_alfa_05 <- power.t.test(n = 10,
30                               delta = efecto,
31                               sd = 1,
32                               sig.level = 0.05,
33                               type = "one.sample",
34                               alternative = "two.sided")$power
35
36 n_10_alfa_01 <- power.t.test(n = 10,
37                               delta = efecto,
38                               sd = 1,
39                               sig.level = 0.01,
40                               type = "one.sample",
41                               alternative = "two.sided")$power

```

```

42
43 # Construir matriz de datos en formato ancho.
44 datos <- data.frame(efecto, n_6_alfa_05, n_6_alfa_01,
45                      n_10_alfa_05, n_10_alfa_01)
46
47 # Llevar a formato largo.
48 datos <- datos %>% pivot_longer(!"efecto",
49                                     names_to = "fuente",
50                                     values_to = "poder")
51
52 # Formatear fuente como variable categórica.
53 niveles <- c("n_6_alfa_05", "n_6_alfa_01", "n_10_alfa_05",
54             "n_10_alfa_01")
55
56 etiquetas <- c("n=6, alfa=0,05", "n=6, alfa=0,01", "n=10, alfa=0,05",
57                 "n=10, alfa=0,01")
58
59 datos[["fuente"]] <- factor(datos[["fuente"]], levels = niveles,
60                               labels = etiquetas)
61
62 # Graficar las curvas de poder.
63 g <- ggplot(datos, aes(efecto, poder, colour = factor(fuente)))
64 g <- g + geom_line()
65 g <- g + labs(colour = "")
66 g <- g + ylab("Poder estadístico")
67 g <- g + xlab("Tamaño del efecto")
68
69 g <- g + scale_color_manual(values=c("red", "blue", "chartreuse4",
70                                 "orange"))
71
72 g <- g + theme_pubr()
73 g <- g + ggtitle("Curvas de poder para prueba t bilateral")
74 g <- g + geom_vline(xintercept = 0, linetype = "dashed")
75
76 print(g)

```

La figura 6.3 muestra las curvas de poder para una prueba t unilateral y otra bilateral, ambas para una muestra de tamaño 6, desviación estándar $s = 1$ y $\alpha = 0,05$. En ella se evidencia claramente la ventaja de las pruebas unilaterales: cuando el tamaño del efecto aumenta en el sentido de la hipótesis alternativa, el poder es mayor que para una prueba bilateral.

Es deseable que las pruebas que se empleen para docir hipótesis tengan un alto poder y, si hay más de una prueba disponible, se debe escoger la más poderosa. No obstante, los cálculos del poder suelen ser altamente complejos. Afortunadamente, la teoría permite en muchos casos conocer la prueba con mayor poder posible ante cualquier hipótesis alternativa, nivel de significación y tamaño de muestra (siempre que se cumplan las condiciones de base). Estas pruebas reciben el nombre de **uniformemente más poderosas**, y tal es el caso de la prueba t de Student.

6.2 TAMAÑO DEL EFECTO

El problema que podríamos tener al considerar el tamaño del efecto en la misma escala de la variable estudiada, como hemos hecho hasta ahora, es que esta escala varía de variable en variable. Para poder hacer comparaciones con mayor libertad, existen diferentes **medidas estandarizadas de efecto** que podemos

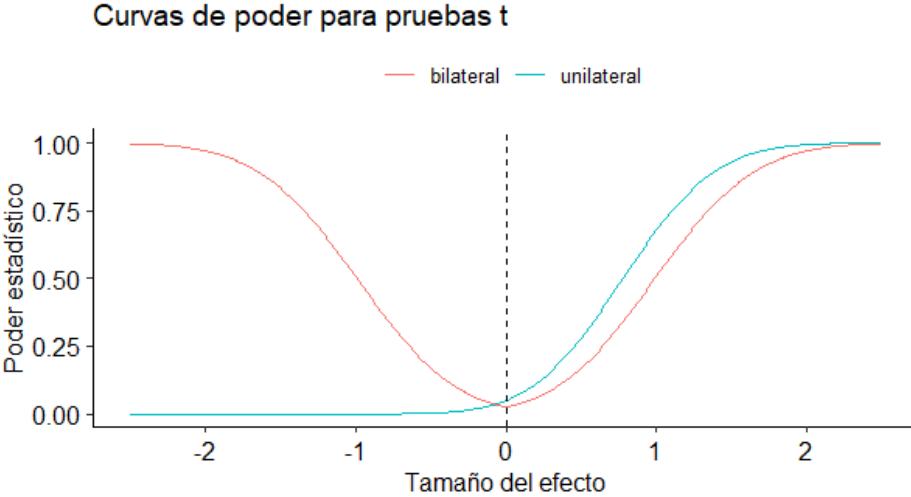


Figura 6.3: poder estadístico para pruebas t.

usar. Puesto que hasta ahora solo hemos estudiado la prueba t de Student, en esta sección conoceremos la llamada **d de Cohen** (Kassambara, 2019b), una medida estándar ampliamente empleada para el tamaño del efecto con esta prueba.

En términos generales, se considera que $d = 0,2$ es un efecto pequeño (imperceptible a simple vista), $d = 0,5$ es un efecto mediano (probablemente perceptible a simple vista) y $d = 0,8$, un efecto grande (definitivamente perceptible a simple vista).

En el caso de la prueba t de una muestra, la d de Cohen se calcula como muestra la ecuación 6.1, donde:

- \bar{x} : media muestral.
- μ_0 : media teórica para el contraste (valor nulo).
- s : desviación estándar de la muestra con $n - 1$ grados de libertad.

$$d = \frac{\bar{x} - \mu_0}{s} \quad (6.1)$$

Para la prueba t de diferencia de dos medias (también llamada prueba t para dos muestras independientes o, simplemente, prueba t independiente), si el tamaño de la muestra es mayor a 50 elementos, se calcula como muestra la ecuación 6.2, y para muestras pequeñas se aplica un factor de corrección, como indica la ecuación 6.3, donde:

- \bar{x}_1, \bar{x}_2 : medias muestrales de cada grupo.
- n_1 y n_2 son los tamaños de ambas muestras.
- s_p : desviación estándar agrupada, dada por la ecuación 6.4¹.

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_p} \quad (6.2)$$

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_p} \cdot \frac{n_1 + n_2 - 3}{n_1 + n_2 - 2, 25} \quad (6.3)$$

$$s_p = \sqrt{\frac{\sum(x - \bar{x}_1)^2 + \sum(x - \bar{x}_2)^2}{n_1 + n_2 - 2}} \quad (6.4)$$

¹ Note que esta corresponde a la raíz de la varianza agrupada descrita en 5.5

En el caso de la variante de Welch para la prueba t independiente, la fórmula para el cálculo de la d de Cohen es ligeramente diferente, como puede apreciarse en la ecuación 6.5.

$$d = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2 + s_2^2}{2}}} \quad (6.5)$$

Por último, las ecuaciones 6.6 y 6.7 muestran cómo se calcula la d de Cohen en el caso de la prueba t con muestras pareadas grandes ($n > 50$) y pequeñas, respectivamente, donde D corresponde a las diferencias entre las observaciones pareadas.

$$d = \frac{\bar{x}_D}{s_D} \quad (6.6)$$

$$d = \frac{\bar{x}_D}{s_D} \cdot \frac{n_1 - 2}{n_1 - 1,25} \quad (6.7)$$

6.3 PODER, TAMAÑO DEL EFECTO Y TAMAÑO DE LA MUESTRA

Mencionamos en páginas anteriores que el poder puede también entenderse como qué tan propensa es una prueba estadística para distinguir un efecto real de una simple casualidad, y que podemos cuantificar este efecto.

Una gran ventaja del poder estadístico es que nos sirve para determinar el tamaño adecuado de la muestra para detectar un cierto tamaño del efecto. La figura 6.4, elaborada con el script 6.2, muestra el aumento del poder estadístico a medida que el tamaño de la muestra aumenta (para un tamaño del efecto y nivel de significación fijos). En ella se aprecia que, a medida que el tamaño de la muestra crece, el poder estadístico también crece asintóticamente a 1, valor que equivale a tener la certeza de rechazar la hipótesis nula si esta es falsa.

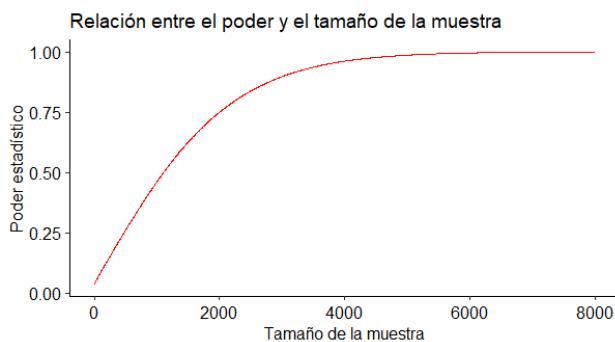


Figura 6.4: aumento del poder estadístico a medida que crece el tamaño de la muestra (manteniendo fijos el tamaño del efecto y el nivel de significación).

Script 6.2: aumento del poder estadístico a medida que crece el tamaño de la muestra.

```

1 library(ggpubr)
2
3 # Generar un vector con un rango para el tamaño de la muestra.
4 n <- seq(5, 8000, 5)

```

```

5
6 # Definir constantes
7 desv_est <- 6
8 alfa <- 0.05
9 tam_efecto <- 0.5
10
11 # Se calcula el poder con que se detecta el tamaño del efecto para
12 # cada tamaño de la muestra, asumiendo una prueba bilateral para
13 # una sola muestra.
14 poder <- power.t.test(n = n,
15                         delta = tam_efecto,
16                         sd = desv_est,
17                         sig.level = alfa,
18                         type = "two.sample",
19                         alternative = "two.sided")$power
20
21 # Crear un data frame.
22 datos <- data.frame(n, poder)
23
24 # Graficar la curva de poder.
25 g <- ggplot(datos, aes(n, poder))
26 g <- g + geom_line(colour = "red")
27 g <- g + ylab("Poder estadístico")
28 g <- g + xlab("Tamaño de la muestra")
29 g <- g + theme_pubr()
30 g <- g + ggtitle("Relación entre el poder y el tamaño de la muestra")
31
32 print(g)

```

6.4 CÁLCULO TEÓRICO DEL PODER

Como ya hemos mencionado a lo largo de este capítulo, el poder es la probabilidad de correctamente rechazar H_0 cuando es falsa, lo que equivale a la probabilidad de distinguir un efecto real de una mera casualidad. Ahora veremos algunos ejemplos de cómo podemos usar el poder.

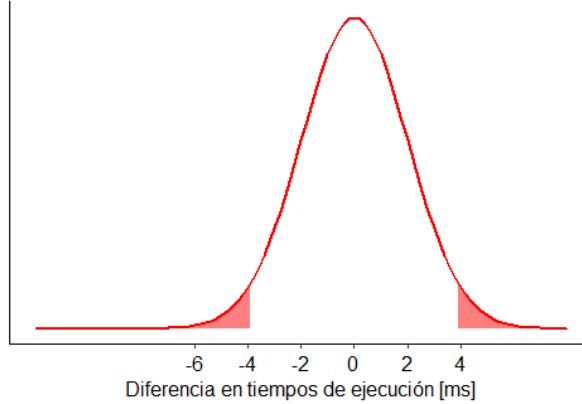
Lola Drones, estudiante de computación, ha diseñado dos nuevos algoritmos (A y B) que resuelven un mismo problema como parte de su trabajo de titulación. Lola desea saber si existe diferencia entre los tiempos de ejecución de ambos algoritmos. Para ello, ha decidido realizar una prueba t con muestras pareadas, con un nivel de significación $\alpha = 0,05$, usando para ello 36 instancias del problema de tamaño fijo que se ejecutan bajo iguales condiciones con cada algoritmo. Además, Lola ya sabe que la diferencia en el tiempo de ejecución sigue una distribución normal con desviación estándar $\sigma = 12$ milisegundos. Así, Lola ha formulado las siguientes hipótesis:

$H_0: \mu_{(A_i - B_i)} = 0$, es decir que la media de las diferencias en el tiempo de ejecución necesitado por los algoritmos A y B , para cada posible instancia i , es cero

$H_A: \mu_{(A_i - B_i)} \neq 0$

La figura 6.5 muestra cómo sería la distribución de la muestra (media de las diferencias en los tiempos de ejecución) si la hipótesis nula (H_0) fuese cierta, con las áreas correspondientes a la región de rechazo de H_0 coloreadas.

Supongamos por un momento que, en realidad, el algoritmo B es en promedio 4 milisegundos más rápido que el algoritmo A . En este caso tendríamos que la media de las diferencias es de -4 [ms], correspondiente



al tamaño del efecto. En este caso, su distribución sería como muestra la figura 6.6 (ver script 6.3) en color azul. Al superponer esta nueva curva a la que ya teníamos bajo el supuesto de que la hipótesis nula fuera verdadera, vemos que el área de la curva real que se situaría dentro de la región de rechazo de la curva teórica es aquella coloreada en azul. Esta área corresponde al poder de la prueba t, que en este caso es de 0,516 de acuerdo al análisis teórico (ver script 6.3, líneas 77–86). Puesto que el poder corresponde a la probabilidad de **no** cometer un error de tipo II, de acuerdo al resultado obtenido se tiene que $\beta = 0,484$. ¡Lola no sería capaz de detectar una diferencia de -4 [ms] casi la mitad de las veces!

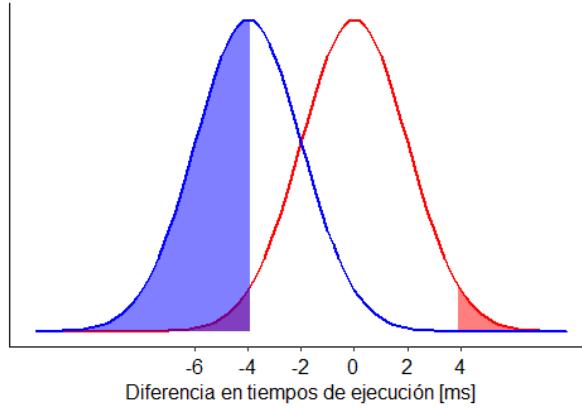


Figura 6.6: región de rechazo de la hipótesis nula en la distribución cuando el programa *B* es, en promedio, 4 milisegundos más rápido que el programa *A*.

Script 6.3: cálculo teórico del poder.

```

1 library(ggpubr)
2 library(pwr)
3
4 # Fijar valores conocidos.
5 sigma <- 12
6 alfa <- 0.05
7 n <- 36
8 media_nula <- 0
9
10 # Calcular el error estándar.
11 SE <- sigma / sqrt(n)
12

```

```

13 # Gráficoar la distribución muestral de la media de las diferencias si
14 # la hipótesis nula fuera verdadera.
15 media_efecto <- -4
16 x <- seq(-6 * SE, 4 * SE, 0.01)
17 y <- dnorm(x, mean = media_nula, sd = SE)
18 df <- data.frame(x, y)
19 g <- ggplot(data = data.frame(x, y), aes(x))
20
21 g <- g + stat_function(
22   fun = dnorm,
23   args = list(mean = media_nula, sd = SE),
24   colour = "red", size = 1)
25
26 g <- g + ylab("")
27 g <- g + scale_y_continuous(breaks = NULL)
28 g <- g + scale_x_continuous(name = "Diferencia en tiempos de ejecución [ms]",
29                             breaks = seq(-6, 4, 2))
30
31 g <- g + theme_pubr()
32
33 # Colorear la región de rechazo de la hipótesis nula.
34 media_nula <- 0
35 Z_critico <- qnorm(alfa/2, mean = media_nula, sd = SE, lower.tail = FALSE)
36 q_critico_inferior <- media_nula - Z_critico
37 q_critico_superior <- media_nula + Z_critico
38
39 g <- g + geom_area(data = subset(df, x < q_critico_inferior),
40                      aes(y = y),
41                      colour = "red",
42                      fill = "red",
43                      alpha = 0.5)
44
45 g <- g + geom_area(data = subset(df, x > q_critico_superior),
46                      aes(y = y),
47                      colour = "red",
48                      fill = "red",
49                      alpha = 0.5)
50
51 print(g)
52
53 # Superponer la distribución muestral de la media de las diferencias
54 # si la la diferencia de medias fuera -4.
55 g <- g + stat_function(
56   fun = dnorm,
57   args = list(mean = media_efecto, sd = SE),
58   colour = "blue", size = 1)
59
60 # Colorear la región de la nueva curva situada en la región de
61 # rechazo de la curva original.
62 x1 <- seq(-6 * SE, 4 * SE, 0.01)
63 y1 <- dnorm(x, mean = media_efecto, sd = SE)
64 g <- g + geom_area(data = subset(data.frame(x1, y1),
65                      x < q_critico_inferior),
66                      aes(x = x1, y = y1),
67                      colour = "blue",
68                      fill = "blue",
69                      alpha = 0.5)
70
71 g <- g + geom_area(data = subset(data.frame(x1, y1),

```

```

72           x > q_critico_superior),
73   aes(x = x1, y = y1),
74   colour = "blue",
75   fill = "blue",
76   alpha = 0.5)
77 print(g)
78
79 # Calcular el poder de acuerdo al análisis teórico.
80 poder <- pnorm(q_critico_inferior,
81                   mean = media_efecto,
82                   sd = SE,
83                   lower.tail = TRUE)
84 + pnorm(q_critico_superior,
85           mean = media_efecto,
86           sd = SE,
87           lower.tail = FALSE)
88
89 cat("Poder = ", poder, "\n")
90
91 # Calcular la probabilidad de cometer un error tipo II.
92 beta <- 1 - poder
93 cat("Beta = ", beta, "\n")

```

6.5 CÁLCULO DEL PODER EN R

Desde luego, si trabajamos con R, podemos usar funciones para calcular el poder. Como primera alternativa, R trae incorporada la función `power.t.test(n, delta, sd, sig.level, power, type, alternative)` (empleada en los scripts 6.1 y 6.2), donde:

- `n`: tamaño de la muestra (por cada grupo, si corresponde).
- `delta`: diferencia observada entre las medias, o entre la media muestral y el valor nulo, no estandarizada.
- `sd`: desviación estándar observada.
- `sig.level`: nivel de significación.
- `power`: poder de la prueba.
- `type`: tipo de prueba t de Student (“`two.sample`” para diferencia de medias, “`one.sample`” para una sola muestra o “`paired`” para dos muestras pareadas).
- `alternative`: tipo de hipótesis alternativa (“`one.sided`” si es unilateral, “`two.sided`” si es bilateral).

Esta función entrega como resultado un objeto con diversos elementos (que podemos indexar del mismo modo que las columnas de una matriz de datos), entre los que se incluyen los 5 primeros argumentos definidos para la función.

Si revisamos con detenimiento los argumentos de la función `power.t.test()`, veremos que ¡recibe el poder como uno de sus argumentos! Esto no parece tener sentido... ¿o sí?

Como ya hemos visto existe una relación entre: poder, tamaño de la muestra, tamaño del efecto y nivel de significación. A esta combinación de elementos debemos añadir también la desviación estándar, aunque no estudiaremos las matemáticas subyacentes.

En realidad, para usar la función `power.t.test()` siempre debemos señalar el tipo de prueba t con el que estamos trabajando y si la hipótesis alternativa es de una o dos colas. Esta función nos permite calcular cualquiera de los demás argumentos (tamaño de la muestra, tamaño del efecto, desviación estándar, nivel de

significación o poder estadístico) para la prueba en cuestión a partir de los 4 argumentos restantes. Así, al argumento que queremos calcular se le asigna el valor NULL en la llamada.

Recordemos que en el ejemplo de la sección anterior, Lola Drones desea usar una prueba t bilateral para dos muestras pareadas a fin de determinar si hay diferencia entre los tiempos de ejecución promedio de ambos algoritmos. Para ello, ha considerado $n = 36$ y $\alpha = 0,05$, sabiendo que $sd = 12$ [ms]. Las líneas 4 a 14 del script 6.4 muestran cómo calcular el poder para este ejemplo si se desea detectar un tamaño del efecto (δ) de 4 [ms], obteniéndose como resultado que el poder es de 0.494 (y $\beta = 1 - \text{poder} = 0,506$), ligeramente diferente al obtenido en forma teórica debido a errores de redondeo.

¿Cuántas instancias debería usar Lola para lograr un poder de 0,9, manteniendo $\alpha = 0,05$, $sd = 12$ [ms] y $\delta = 4$ [ms]? Las líneas 17–28 del script 6.4 muestran cómo hacer este cálculo, obteniéndose como resultado $n = 97$. Como el tamaño de la muestra siempre debe ser un entero positivo, la línea 27 aproxima el resultado al entero superior.

Otra alternativa es usar la función `pwr.t.test(n, d, sig.level, power, type, alternative)` (ver script 6.4, líneas 37–63), incluida en el paquete `pwr`, donde:

- `n`: tamaño de la muestra (por cada grupo, si corresponde).
- `d`: tamaño del efecto (`d` de Cohen).
- `sig.level`: nivel de significación.
- `power`: poder de la prueba.
- `type`: tipo de prueba t de Student (“`two.sample`” para diferencia de medias, “`one.sample`” para una sola muestra o “`paired`” para dos muestras pareadas).
- `alternative`: tipo de hipótesis alternativa (“`greater`” o “`less`” si es unilateral, “`two.sided`” si es bilateral).

Debemos fijarnos en que, si bien esta función opera de manera similar a `power.t.test()`, en este caso la desviación estándar y la diferencia son reemplazadas por el tamaño del efecto que podemos cuantificar, como ya vimos, mediante la `d` de Cohen. Sin embargo, debemos tener cuidado, pues la función `pwr.t.test()` solo es adecuada para una muestra, dos muestras pareadas o cuando ambas muestras tienen igual tamaño. En el caso de la prueba t para dos muestras independientes con diferentes tamaños, debemos usar, en cambio, la función `pwr.t2n.test(n1, n2, d, sig.level, power, alternative)`.

Script 6.4: cálculo del poder en R.

```

1 library(pwr)
2
3 # Fijar valores conocidos.
4 n <- 36
5 diferencia <- 4
6 desv_est <- 12
7 alfa <- 0.05
8 poder <- 0.9
9
10 # Calcular el poder usando la función power.t.test().
11 cat("Cálculo del poder con power.t.test()\n")
12
13 resultado <- power.t.test(n = n,
14                             delta = diferencia,
15                             sd = desv_est,
16                             sig.level = alfa,
17                             power = NULL,
18                             type = "paired",
19                             alternative = "two.sided")
20
21 print(resultado)
22
23 # Cálculo del tamaño de la muestra usando la función power.t.test().
24 cat("Cálculo del tamaño de la muestra con power.t.test()\n")

```

```

25
26 resultado <- power.t.test(n = NULL,
27                             delta = diferencia,
28                             sd = desv_est,
29                             sig.level = alfa,
30                             power = poder,
31                             type = "paired",
32                             alternative = "two.sided")
33
34 n <- ceiling(resultado[["n"]])
35 cat("n = ", n, "\n")
36
37 # Calcular el tamaño del efecto (d de Cohen).
38 d <- (4 / desv_est) * ((n - 2) / (n - 1.25))
39
40 # Calcular el poder usando la función pwr.t.test().
41 cat("\n\nCálculo del poder con pwr.t.test()\n")
42
43 resultado <- pwr.t.test(n = n,
44                         d = d,
45                         sig.level = alfa,
46                         power = NULL,
47                         type = "paired",
48                         alternative = "two.sided")
49
50 print(resultado)
51
52 # Cálculo del tamaño de la muestra usando la función pwr.t.test().
53 cat("\nCálculo del tamaño de la muestra con pwr.t.test()\n")
54
55 resultado <- pwr.t.test(n = NULL,
56                         d = d,
57                         sig.level = alfa,
58                         power = poder,
59                         type = "paired",
60                         alternative = "two.sided")
61
62 n <- ceiling(resultado[["n"]])
63 cat("n = ", n, "\n")

```

6.6 EJERCICIOS PROPUESTOS

1. Define con tus propias palabras lo que es el tamaño del efecto.
2. Un estudio sobre en el tiempo que necesitan los estudiantes para resolver una guía de ejercicios de Cálculo I, comparó un grupo de estudiantes que cursaban la asignatura por primera vez con un grupo que la cursaba en segunda ocasión. Sabiendo que este tiempo se distribuye normalmente en ambos casos, con varianza similar, dibuja cómo se verían los datos si el efecto de repetir la asignatura sobre el tiempo requerido para resolver la guía fuera “grande” y si este efecto fuera “pequeño, pero positivo”.
3. Investiga cómo se calcula y cómo se interpreta la medida d de Hedges para el tamaño del efecto, e indica en qué casos es adecuada.
4. ¿Por qué se necesita conocer el tamaño del efecto?
5. ¿Cómo se relaciona el tamaño del efecto con la significación estadística?

6. ¿Por qué sería útil determinar un tamaño de muestra apropiado?
7. Explica en tus palabras lo que se muestra en la figura 6.4.
8. Ante algunas acusaciones de colusión, el Tribunal de la Libre Competencia quiere estudiar dos compañías del mercado de los seguros de automóviles. En base a datos del gremio de las aseguradoras, se puede asumir que el precio de las primas estándares para diferentes marcas de vehículos sigue una distribución aproximadamente normal con desviación estándar de \$16.000. Fija los otros parámetros del estudio y determina qué tamaño debería tener la muestra de automóviles para detectar una diferencia de \$10.000 en el precio medio de las compañías bajo sospecha.

CAPÍTULO 7. INFERENCIA CON PROPORCIONES MUESTRALES

En el capítulo 5 conocimos las pruebas Z y t de Student para contrastar hipótesis con una y dos medias. Ahora estudiaremos los métodos de Wald y de Wilson para inferir acerca de una y dos proporciones, basándonos para ello en los textos de Diez et al. (2017, pp. 274-286), NIST/SEMATECH (2013, pp. 7.2.4, 7.2.4.1), Pértega y Pita (2004), Champely, Ekstrom, Dalgaard, Gill, Weibelzahl, Anandkumar, Ford, Volcic y de Rosario (2020) y Kabacoff (2017).

7.1 MÉTODO DE WALD

En el capítulo 3 vimos que, cuando queremos responder preguntas del tipo “¿qué proporción de la ciudadanía apoya al gobierno actual?”, estamos hablando de una variable aleatoria que sigue una distribución binomial. En general, no conocemos la **probabilidad de éxito p** de la población, por lo que tenemos que usar el estimador puntual (correspondiente a la proporción de éxito de la muestra), denotado por \hat{p} . Este estimador se distribuye de manera cercana a la normal cuando se cumplen las siguientes condiciones:

1. Las observaciones de la muestra son independientes.
2. Se cumple la **condición de éxito-fracaso**, que establece que se espera observar al menos 10 observaciones correspondientes a éxito y al menos 10, correspondientes a fracasos. Matemáticamente, $np \geq 10$ y $n(1-p) \geq 10$.

Así, si la distribución muestral de \hat{p} cumple con las condiciones anteriores, se dice que es cercana a la normalidad con media $\mu = p$, desviación estándar $\sigma = \sqrt{p(1-p)}$ y error estándar dado por la ecuación 7.1.

$$SE_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}} \quad (7.1)$$

7.1.1 Método de Wald para una proporción

El **método de Wald** permite construir intervalos de confianza y contrastar hipótesis bajo el supuesto de normalidad para una proporción. Consideremos el siguiente ejemplo: Aquiles Baeza, ingeniero en informática, desea conocer qué proporción de las ejecuciones de un algoritmo de ordenamiento para instancias con 100.000 elementos (bajo iguales condiciones de hardware y sistema) tardan menos de 25 segundos. Para ello, registró los tiempos de ejecución para 150 instancias generadas de manera aleatoria, encontrando que 64 % de dichas instancias fueron resueltas en un tiempo menor al señalado.

Si bien no conocemos la probabilidad real de éxito para la población, sabemos que $\hat{p} = 0,64$. Así, si se cumplen las condiciones para que la distribución de \hat{p} sea cercana a la normal, podemos construir un intervalo de confianza para la verdadera proporción muestral.

En el enunciado del ejemplo nos indican que las instancias del problema fueron escogidas de manera aleatoria y sabemos que éstas representan menos del 10 % del total de instancias posibles, con lo que se verifica la

independencia de las observaciones. Por otra parte, nos dicen que la proporción de éxito es $\hat{p} = 0,64$, por lo que esperamos encontrar $0,64 \cdot 150 = 96$ instancias que tardan menos de 25 segundos y $(1 - 0,64) \cdot 150 = 54$ fracasos (instancias que tardan 25 segundos o más), con lo que se cumple la condición de éxito-fracaso. En consecuencia, podemos asumir que la distribución muestral de \hat{p} sigue aproximadamente a la normal.

Podemos estimar el error estándar usando la ecuación 7.1, reemplazando p por el estadístico \hat{p} :

$$SE = \sqrt{\frac{\hat{p}(1 - \hat{p})}{n}} = \sqrt{\frac{0,64(1 - 0,64)}{150}} = 0,0392$$

Con ello, construimos el intervalo de confianza para un nivel de significación $\alpha = 0,05$ usando la ecuación general (4.6) con \hat{p} como estimador puntual:

$$\hat{p} \pm z^* \cdot SE \rightarrow 0,64 \pm 1,96 \cdot 0,0392 \rightarrow [0,5632; 0,7168]$$

Este intervalo significa que tenemos 95 % de confianza que la proporción de instancias (de 100.000 elementos) del problema que el algoritmo ordena en menos de 25 segundos se encuentra entre 56,32 % y 71,6 %.

Desde luego, también podemos usar el modelo normal en el contexto de la prueba de hipótesis para una proporción. Para ello, se deben cumplir las condiciones de independencia y éxito-fracaso que ya verificamos para construir el intervalo de confianza, pero en este caso tenemos que verificar la segunda condición con el valor nulo, denotado p_0 . Una vez verificadas ambas condiciones, el error estándar y el estadístico Z que permiten determinar el p-valor se calculan usando las ecuaciones 7.2 y 7.3, respectivamente.

$$SE = \sqrt{\frac{p_0(1 - p_0)}{n}} \quad (7.2)$$

$$Z = \frac{\hat{p} - p_0}{SE} \quad (7.3)$$

Supongamos ahora, volviendo a nuestro ejemplo, que Baeza afirma que más del 70 % de las instancias de tamaño 100.000 se ejecutan en menos de 25 segundos. Sin embargo, su jefe no está seguro, por lo que decide comprobarlo mediante una prueba de hipótesis con un nivel de significación $\alpha = 0,05$ (recordemos que $n = 150$ y $\hat{p} = 0,64$):

H_0 : el 70 % de las instancias se ejecutan en menos de 25 segundos.

H_A : más del 70 % de las instancias se ejecutan en menos de 25 segundos.

De acuerdo a las hipótesis formuladas por el jefe de Baeza, el valor nulo es $p_0 = 0,7$, con lo que estas pueden formularse matemáticamente como:

Denotando como p a la proporción de todas las instancias de tamaño 100.000 que se ejecutan en menos de 25 segundos y considerando el valor hipotético $p_0 = 0,7$ para este parámetro:

H_0 : $p = p_0$

H_A : $p > p_0$

Ya antes habíamos comprobado que se verifica la independencia de las observaciones. Además, considerando que el valor nulo fuese verdadero esperaríamos encontrar $0,7 \cdot 150 = 105$ éxitos y $(1 - 0,7) \cdot 150 = 45$ fracasos, ambos valores mayores que 10, por lo que la condición de éxito-fracaso se verifica.

Con ello, podemos calcular el estadístico de prueba:

$$SE = \sqrt{\frac{p_0(1 - p_0)}{n}} = \sqrt{\frac{0,7(1 - 0,7)}{150}} = 0,0374$$

$$Z = \frac{\hat{p} - p_o}{SE} = \frac{0,64 - 0,7}{0,0374} = -1,6043$$

El valor p asociado, calculado en R mediante la llamada a la función `2 * pnorm(-1.6042)`, es $p = 0,109$. En consecuencia, la evidencia no es suficiente para rechazar la hipótesis nula, por lo que se concluye, con 95 % de confianza, que no es posible aceptar que el algoritmo se ejecute en menos de 25 segundos para más del 70 % de las instancias de tamaño 100.000.

R no ofrece esta prueba, como función. Sin embargo, podemos hacerla como muestra el script 7.1 para nuestro ejemplo.

Script 7.1: método de Wald para una proporción.

```

1 # Fijar valores conocidos
2 n <- 150
3 p_exito <- 0.64
4 alfa <- 0.05
5 valor_nulo <- 0.7
6
7 # Construcción del intervalo de confianza.
8 error_est <- sqrt((p_exito * (1 - p_exito)) / n)
9 Z_critico <- qnorm(alfa / 2, lower.tail = FALSE)
10 inferior <- p_exito - Z_critico * error_est
11 superior <- p_exito + Z_critico * error_est
12 cat("Intervalo de confianza = [", inferior, ", ", superior, "]\n", sep = "")
13
14 # Prueba de hipótesis.
15 error_est_hip <- sqrt((valor_nulo * (1 - valor_nulo)) / n)
16 Z <- (p_exito - valor_nulo) / error_est_hip
17 p <- pnorm(Z, lower.tail = FALSE)
18 cat("Hipótesis alternativa unilateral\n")
19 cat("Z =", Z, "\n")
20 cat("p =", p)

```

7.1.2 Método de Wald para dos proporciones

También podemos usar el método de Wald para estudiar la **diferencia entre las proporciones** de dos poblaciones, considerando para ello como estimador puntual la diferencia $\hat{p}_1 - \hat{p}_2$.

De manera similar a lo que ya vimos para una única proporción, también en este caso debemos verificar ciertas condiciones antes de poder aplicar el modelo normal:

1. Cada proporción, por separado, sigue el modelo normal.
2. Las dos muestras son independientes una de la otra.

El error estándar para la diferencia entre dos proporciones muestrales está dado por la ecuación 7.4, donde p_1 y p_2 corresponden a las proporciones de las poblaciones, y n_1 y n_2 , a los tamaños de las muestras. La construcción del intervalo de confianza se realiza, una vez más, con la ecuación general 4.6.

$$SE_{\hat{p}_1 - \hat{p}_2} = \sqrt{SE_{\hat{p}_1}^2 + SE_{\hat{p}_2}^2} = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}} \quad (7.4)$$

A modo de ejemplo, supongamos que la Facultad de Ingeniería de una prestigiosa universidad desea determinar si la tasa de reprobación de estudiantes que rinden la asignatura de programación por primera vez es igual para hombres y mujeres. Para ello, se examina la situación final de los estudiantes que rindieron la asignatura durante el segundo semestre de 2017. Para una muestra de 48 hombres (de un total de 632), se encontró que 26 de ellos reprobaron la asignatura. De manera similar, para una muestra de 42 mujeres (de un total de 507), se encontraron 20 reprobaciones¹, con ambas muestras tomadas de manera aleatoria.

Como ya es habitual, comencemos por verificar las condiciones de normalidad para cada una de las muestras. En ambos casos, las observaciones son independientes entre sí, pues provienen de personas diferentes que representan a menos del 10 % de la población. Además, los datos entregados evidencian que en ambos casos se cumple la condición de éxito-fracaso. Adicionalmente, ambas muestras son independientes entre sí, pues ambas categorías se excluyen mutuamente. Con esto último se verifican entonces las condiciones de normalidad para la diferencia de proporciones.

Sean \hat{p}_1 y \hat{p}_2 las proporciones de éxito muestrales (considerando en este contexto la reprobación como éxito) para hombres y mujeres, respectivamente:

$$\hat{p}_1 = 26/48 = 0,5417$$

$$\hat{p}_2 = 20/42 = 0,4762$$

$$\hat{p}_1 - \hat{p}_2 = 0,5417 - 0,4762 = 0,0655$$

El error estándar puede estimarse como:

$$SE_{\hat{p}_1 - \hat{p}_2} = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}} = \sqrt{\frac{0,5417(1-0,5417)}{48} + \frac{0,4762(1-0,4762)}{42}} = 0,1054$$

Suponiendo un nivel de significación $\alpha = 0,05$, el intervalo de confianza corresponde a:

$$\hat{p}_1 - \hat{p}_2 \pm z^* SE_{\hat{p}_1 - \hat{p}_2} \rightarrow 0,0655 \pm 1,96 \cdot 0,1054 \rightarrow [-0,1411; 0,2721]$$

En consecuencia, podemos afirmar con 95 % de confianza que la diferencia en la tasa de reprobación de la asignatura de programación para hombres y mujeres varía entre -14,11 % y 27,21 %.

Desde luego, también podemos realizar pruebas de hipótesis en este escenario. Para el ejemplo tenemos que:

H_0 : no hay diferencia en la tasa de reprobación de hombres y mujeres.

H_A : las tasas de reprobación son diferentes para hombres y mujeres.

Matemáticamente:

Denotando como p_1 y p_2 a las proporciones de hombres y mujeres, respectivamente, que reprueban la asignatura de programación la primera vez que la cursan:

$H_0: p_1 - p_2 = 0$

$H_A: p_1 - p_2 \neq 0$

Ya verificamos las condiciones para operar bajo el supuesto de normalidad cuando construimos el intervalo de confianza. Sin embargo, **cuando la hipótesis nula supone que no hay diferencia entre las proporciones**, la verificación de la condición de éxito-fracaso y la estimación del error estándar se realizan usando para ello la **proporción agrupada**, dada por la ecuación 7.5, donde $\hat{p}_1 n_1$ y $\hat{p}_2 n_2$ representan la cantidad de éxitos en la primera y segunda muestra, respectivamente.

¹ Los datos aquí presentados son ficticios, creados únicamente con fines pedagógicos.

$$\hat{p} = \frac{\text{número de éxitos}}{\text{número de casos}} = \frac{\hat{p}_1 n_1 + \hat{p}_2 n_2}{n_1 + n_2} \quad (7.5)$$

Así, en este caso tenemos:

$$\hat{p} = \frac{\hat{p}_1 n_1 + \hat{p}_2 n_2}{n_1 + n_2} = \frac{0,5417 \cdot 48 + 0,4762 \cdot 42}{48 + 42} = 0,5111$$

En consecuencia, en el caso de los hombres esperamos encontrar $\hat{p}n_1 = 24,5328 > 10$ éxitos (reprobaciones) y $(1 - \hat{p})n_1 = 23,4672 > 10$ fracasos. Del mismo modo, para las mujeres esperamos $\hat{p}n_2 = 21,4662 > 10$ éxitos y $(1 - \hat{p})n_2 = 20,5338 > 10$ fracasos, con lo que se verifican las condiciones para emplear el modelo normal.

El error estándar se calcula, como ya mencionamos, usando la proporción agrupada:

$$SE = \sqrt{\frac{\hat{p}(1 - \hat{p})}{n_1} + \frac{\hat{p}(1 - \hat{p})}{n_2}} = \sqrt{\frac{0,5111 \cdot (1 - 0,5111)}{48} + \frac{0,5111 \cdot (1 - 0,5111)}{42}} = 0,1056$$

El estimador puntual para la diferencia es $\hat{p}_1 - \hat{p}_2 = 0,0655$, con lo cual el estadístico de prueba está dado por:

$$Z = \frac{\text{estimador puntual} - \text{valor nulo}}{SE} = \frac{0,0655 - 0}{0,1056} = 0,6203$$

En consecuencia, el valor p correspondiente es $p = 0,5351$. Puesto que el valor p es mayor que $\alpha = 0,05$, se falla en rechazar la hipótesis nula. Así, podemos decir con 95 % de confianza que no existe evidencia suficiente para concluir que hay diferencia en la tasa de reprobación de hombres y mujeres para el primer curso de programación.

El script 7.2 muestra el desarrollo de este ejemplo en R.

Script 7.2: método de Wald para la diferencia entre dos proporciones (ejemplo 1).

```

1 # Fijar valores conocidos
2 n_hombres <- 48
3 n_mujeres <- 42
4 exitos_hombres <- 26
5 exitos_mujeres <- 20
6 alfa <- 0.05
7 valor_nulo <- 0
8
9 # Calcular probabilidades de éxito.
10 p_hombres <- exitos_hombres / n_hombres
11 p_mujeres <- exitos_mujeres / n_mujeres
12
13 # Estimar la diferencia.
14 diferencia <- p_hombres - p_mujeres
15
16 # Construcción del intervalo de confianza.
17 error_hombres <- (p_hombres * (1 - p_hombres)) / n_hombres
18 error_mujeres <- (p_mujeres * (1 - p_mujeres)) / n_mujeres
19 error_est <- sqrt(error_hombres + error_mujeres)
20 Z_critico <- qnorm(alfa / 2, lower.tail = FALSE)
21 inferior <- diferencia - Z_critico * error_est
22 superior <- diferencia + Z_critico * error_est
23 cat("Intervalo de confianza = [", inferior, ", ", superior, "]\n", sep = "")
24
25 # Prueba de hipótesis.

```

```

26 p_agrupada <- (exitos_hombres + exitos_mujeres) / (n_hombres + n_mujeres)
27 error_hombres <- (p_agrupada * (1 - p_agrupada)) / n_hombres
28 error_mujeres <- (p_agrupada * (1 - p_agrupada)) / n_mujeres
29 error_est_ hip <- sqrt(error_hombres + error_mujeres)
30 Z <- (diferencia - valor_nulo) / error_est_ hip
31 p <- 2 * pnorm(Z, lower.tail = FALSE)
32 cat("Hipótesis alternativa bilateral\n")
33 cat("Z =", Z, "\n")
34 cat("p =", p)

```

Cuando contrastamos hipótesis para la **diferencia entre dos proporciones con un valor nulo distinto de 0**, el procedimiento es ligeramente diferente. En este caso, la comprobación de la condición de éxito-fracaso se realiza de manera independiente para ambas muestras y el error estándar se calcula, como ya se estudió para los intervalos de confianza, mediante la ecuación 7.4.

Supongamos ahora que la Facultad de Ingeniería de la Universidad anterior ha decidido replicar el estudio realizado para el curso de programación, esta vez para una asignatura de física. No obstante, las autoridades están convencidas de que la tasa de reprobación es 10 % mayor para los hombres y que, incluso, la diferencia podría ser mayor. Desean comprobar con un nivel de confianza de 95 % y para ello, seleccionaron aleatoriamente a 89 de los 1.023 hombres y a 61 de las 620 mujeres de la cohorte correspondiente al primer semestre de 2019. En las muestras se encuentran, respectivamente, 45 y 21 reprobaciones.

Las hipótesis son, en este caso:

H_0 : la tasa de reprobación de los hombres es exactamente 10 % más alta que la de las mujeres.
 H_A : la tasa de reprobación de los hombres es más de 10 % más alta que la de las mujeres.

Matemáticamente:

Denotando como p_1 y p_2 a las proporciones de hombres y mujeres, respectivamente, que reprueban la asignatura de física estudiada la primera vez que la cursan:

$H_0: p_1 - p_2 = 0,1$

$H_A: p_1 - p_2 > 0,1$

Al igual que en los ejemplos previos, las observaciones de cada muestra son independientes entre sí pues corresponden a menos del 10 % de la población y fueron escogidos aleatoriamente. A su vez, los datos proporcionados indican que se cumple la condición de éxito-fracaso para cada muestra. Como ambas muestras pertenecen a grupos diferentes de estudiantes, son independientes entre sí. En consecuencia, se cumplen las condiciones para operar bajo el modelo normal.

En el caso de los hombres, la tasa de éxito se estima como:

$$\hat{p}_1 = \frac{45}{89} = 0,5056$$

Análogamente, para las mujeres tenemos:

$$\hat{p}_2 = \frac{21}{61} = 0,3443$$

Con lo que el estimador puntual para la diferencia es:

$$\hat{p}_1 - \hat{p}_2 = 0,5056 - 0,3443 = 0,1613$$

Ahora calculamos el error estándar:

$$SE = \sqrt{\frac{0,5056(1 - 0,5056)}{89} + \frac{0,3443(1 - 0,3443)}{61}} = 0,0807$$

Con lo cual podemos calcular el estadístico de prueba:

$$Z = \frac{\text{estimador puntual} - \text{valor nulo}}{SE} = \frac{0,1613 - 0,1}{0,0807} = 0,7596$$

Con lo que se puede obtener el valor p, correspondiente a $p = 0.2237 > \alpha = 0,05$.

En consecuencia, se falla en rechazar H_0 en favor de H_A , por lo que concluimos, con 95 % de confianza, que no es posible descartar que la tasa de reprobación de los hombres es 10% superior a la de las mujeres para el curso de física.

En R, esta prueba puede realizarse como muestra el script 7.3.

Script 7.3: método de Wald para la diferencia entre dos proporciones (ejemplo 2).

```
1 # Fijar valores conocidos
2 n_hombres <- 89
3 n_mujeres <- 61
4 exitos_hombres <- 45
5 exitos_mujeres <- 21
6 alfa <- 0.05
7 valor_nulo <- 0.1
8
9 # Calcular probabilidades de éxito.
10 p_hombres <- exitos_hombres / n_hombres
11 p_mujeres <- exitos_mujeres / n_mujeres
12
13 # Estimar la diferencia.
14 diferencia <- p_hombres - p_mujeres
15
16 # Prueba de hipótesis.
17 p_agrupada <- (exitos_hombres + exitos_mujeres) / (n_hombres + n_mujeres)
18 error_hombres <- (p_hombres * (1 - p_hombres)) / n_hombres
19 error_mujeres <- (p_mujeres * (1 - p_mujeres)) / n_mujeres
20 error_est <- sqrt(error_hombres + error_mujeres)
21 Z <- (diferencia - valor_nulo) / error_est
22 p <- pnorm(Z, lower.tail = FALSE)
23 cat("Hipótesis alternativa bilateral\n")
24 cat("Z =", Z, "\n")
25 cat("p =", p)
```

7.2 MÉTODO DE WILSON

El método de Wald, tratado en la sección anterior, es el método que tradicionalmente se ha usado y el que aparece en la mayoría de los libros clásicos de inferencia estadística. Sin embargo, el método está siendo muy criticado hoy en día debido a que hace importantes simplificaciones matemáticas en su procedimiento y ya hay evidencia empírica que ha demostrado sus limitaciones (Agresti & Coull, 1998).

Gracias al aumento del poder de cómputo y la disponibilidad de software estadístico, han surgido diversas alternativas, entre las cuales destaca el **método de Wilson** (junto con algunas variaciones), considerado el más robusto por diversos autores (Agresti & Coull, 1998; Brown et al., 2001; Devore, 2008; Wallis, 2013). Este método opera del mismo modo que el de Wald, aunque las fórmulas empleadas para estimar la proporción en la muestra y el error estándar son diferentes.

En R, podemos hacer esta prueba usando la función `prop.test(x, n, p, alternative, conf.level, ...)`, cuyos principales parámetros son:

- `x`: cantidad de éxitos en la muestra.
- `n`: tamaño de la muestra.
- `p`: valor nulo (por defecto, `p=NULL`).
- `alternative`: tipo de hipótesis alternativa, por defecto bilateral (`alternative="two.sided"`), y valores “`less`” y “`greater`” para hipótesis unilaterales.
- `conf.level`: nivel de confianza (`conf.level=0.95` por defecto).

El script 7.4 muestra el uso de esta función con el mismo ejemplo que usamos para presentar la prueba de Wald para una proporción. Del mismo modo, el script 7.5 usa la función `prop.test()` para el primer ejemplo del método de Wald para la diferencia entre dos proporciones. Sin embargo, esta función tiene la limitante de que, al trabajar con dos proporciones, no permite establecer un valor nulo distinto de cero para la diferencia.

Script 7.4: método de Wilson para una proporción.

```

1 # Fijar valores conocidos
2 n <- 150
3 p_exito <- 0.64
4 alfa <- 0.05
5 valor_nulo <- 0.7
6
7 # Calcular cantidad de éxitos.
8 exitos <- p_exito * n
9
10 # Prueba de Wilson en R.
11 prueba <- prop.test(exitos, n = n, p = valor_nulo,
12                      alternative = "greater", conf.level = 1 - alfa)
13
14 print(prueba)

```

Script 7.5: método de Wilson para la diferencia entre dos proporciones.

```

1 # Fijar valores conocidos (hombres, mujeres)
2 n <- c(48, 42)
3 exitos <- c(26, 20)
4 alfa <- 0.05
5
6 # Prueba de Wilson en R.
7 prueba <- prop.test(exitos, n = n, alternative = "two.sided",
8                      conf.level = 1 - alfa)
9
10 print(prueba)

```

7.3 PODER Y PRUEBAS DE PROPORCIONES

En el capítulo anterior conocimos el poder estadístico y vimos que está relacionado con el nivel de significación, el tamaño de la muestra y el tamaño del efecto que queremos detectar.

R base nos ofrece la función `power.prop.test(n, p1, p2, sig.level, power, alternative)`, donde:

- `n`: número de observaciones por cada grupo.
- `p1`: probabilidad de éxito en un grupo.
- `p2`: probabilidad de éxito en otro grupo.

- `sig.level`: nivel de significación.
- `power`: poder de la prueba.
- `alternative`: tipo de hipótesis alternativa (“`one.sided`” si es unilateral, “`two.sided`” si es bilateral).

Al igual que vimos en el capítulo anterior para la función `power.t.test()`, recibe cuatro de los primeros argumentos y al restante debe asignársele el valor `NULL`. Como resultado, retorna un objeto que incluye el valor calculado para el argumento faltante.

Una vez más, el paquete `pwr` de R nos ofrece varias funciones que podemos usar como alternativa:

- `pwr.p.test(h, n, sig.level, power, alternative)`: para pruebas con una única proporción.
- `pwr.2p.test(h, n, sig.level, power, alternative)`: para pruebas con dos proporciones donde ambas muestras son de igual tamaño.
- `pwr.2p2n.test(h, n1, n2, sig.level, power, alternative)`: para pruebas con dos proporciones y muestras de diferente tamaño.

Donde:

- `h`: tamaño de efecto.
- `n, n1, n2`: tamaño(s) de la(s) muestra(s).
- `sig.level`: nivel de significación.
- `power`: poder.
- `alternative`: tipo de hipótesis alternativa (“`two.sided`”, “`less`” o “`greater`”).

El funcionamiento de esta familia de funciones es igual al que ya conocimos en el capítulo anterior para la función `pwr.t.test()`. Se entrega el parámetro `alternative` y todos los demás excepto uno (al cual debe asignarse explícitamente el valor `NULL`). Como resultado, la función calcula dicho valor.

El tamaño del efecto puede calcularse como muestra la ecuación 7.6, implementada en R en la función `ES.h(p1, p2)` del paquete `pwr`.

$$h = 2 \arcsin(\sqrt{p_1}) - 2 \arcsin(\sqrt{p_2}) \quad (7.6)$$

En el caso de una única proporción, los autores del paquete `pwr` sugieren usar $p_2 = 0,5$ (Champely et al., 2020).

Otra función que nos puede ser de ayuda es `bsamsize(p1, p2, fraction, alpha, power)`, del paquete `Hmisc`. En el caso de una prueba de Wilson con dos muestras, calcula los tamaños de cada grupo dados los siguientes argumentos:

- `p1`: probabilidad de la población para el grupo 1.
- `p2`: probabilidad del grupo 2.
- `fraction`: fracción de las observaciones en el grupo 1 ($n_1/(n_1 + n_2)$).
- `alpha`: nivel de significación.
- `power`: poder deseado.

7.4 EJERCICIOS PROPUESTOS

1. ¿En qué condiciones la distribución muestral de una proporción tiene comportamiento aproximadamente normal?
2. ¿Cómo se calcula la desviación estándar de la distribución muestral de las proporciones bajo estas condiciones (según el método de Wald)?
3. ¿Cómo se calcula un intervalo de confianza para la verdadera proporción (según el método de Wald)?

4. El patrón de un gran fundo de nogales está preocupado porque se ha detectado la presencia de una plaga en varios árboles. Si bien existe un pesticida para el parásito, este es bastante caro y su aplicación solo se justifica económicamente si más del 20 % de los árboles está infectado. En consecuencia, el patrón ha decidido estimar la extensión de la infestación revisando una muestra aleatoria de 200 nogales (una porción bastante pequeña de los más de 20.000 árboles en el fundo). En base a lo anterior, determina:
 - a) ¿Cuál es la variable dicotómica (experimento Bernulli) en este caso?
 - b) ¿Cuál es el parámetro de interés?
 - c) ¿Qué estimador existe para este parámetro?
 - d) ¿Qué hipótesis respondería las dudas del patrón del fundo?
5. En el experimento del ejercicio anterior se encontró que 45 árboles de la muestra estaban infectados:
 - a) ¿Se puede asumir que esta proporción muestral sigue el modelo normal?
 - b) Independientemente de la respuesta anterior, obtén un intervalo con 95 % confianza para la verdadera proporción de árboles infectados en el fundo.
 - c) ¿Qué recomendarías al patrón del fundo?
6. Como el patrón sigue con dudas, ahora pregunta: ¿cuántos árboles debería revisar en una muestra para estar 99 % confiado que más del 20 % de los árboles están infectados, con solo 10 % de probabilidades de equivocarse si la verdadera proporción fuera 18 %? ¿Cómo se puede calcular esto? ¿Cuál debiera ser la respuesta a la pregunta del patrón?
7. ¿En qué condiciones la distribución muestral de la diferencia de dos proporciones tiene comportamiento aproximadamente normal?
8. ¿Cómo se calcula el error estándar de la diferencia entre dos proporciones (según el método de Wald)?
9. ¿Cómo se calcula un intervalo de confianza para la verdadera diferencia entre dos proporciones (según el método de Wald)?
10. Un laboratorio homeopático acaba de lanzar un tónico que asegura que ayuda a prevenir el resfriado durante el periodo invernal, con igual eficacia tanto en mujeres como en hombres. Para comprobar esta promesa, el laboratorio está realizando un estudio de la eficacia del producto en una muestra aleatoria de 100 mujeres y 200 hombres:
 - a) ¿Cuál es el parámetro de interés y qué estimador se podría usar?
 - b) ¿Qué hipótesis se deberían docir para comprobar o refutar la homogeneidad de la eficacia del tónico para el resfriado?
11. El estudio anterior encontró que, durante las semanas de prueba, 38 mujeres y 102 hombres presentaron síntomas de resfriado. ¿Es homogénea la eficacia del producto con un nivel de significación de 0,05?
12. ¿Qué poder tuvo la prueba anterior?
13. ¿Qué tamaño deberían tener las muestras aleatorias de mujeres y hombres (manteniendo la proporción del ejemplo) para conseguir un poder de 0,85 con 99 % de confianza?
14. Las fórmulas presentadas en la sección 7.1, se conocen colectivamente como “Método de Wald”, el que ya no es recomendado por académicos del área. Usando la bibliografía citada, ¿cuáles son las fórmulas del método de Wilson para estimar el error estándar de la proporción y su extensión a la prueba de hipótesis e intervalos de confianza?
15. Investiga para qué sirve y cómo funciona el parámetro **correct** (verdadero por defecto) de la función `prop.test()` de R.

CAPÍTULO 8. INFERENCIA NO PARAMÉTRICA CON PROPORCIONES

Si eres una persona observadora, habrás notado que el título de este capítulo lleva la frase **no paramétrica** para referirse a inferencias con proporciones, pero ¿qué significa esto?

En el capítulo 5 conocimos las pruebas Z y t de Student. Ambas formulan hipótesis relativas al parámetro μ de una distribución normal (o la diferencia $\mu_1 - \mu_2$ de dos distribuciones normales). Así estas pruebas (y otras que se verán más adelante) hacen una fuerte suposición acerca de la distribución que subyace a las poblaciones estudiadas, lo que permite inferir sobre los parámetros de esas distribuciones. Lo mismo ocurre con las pruebas de Wald y Wilson estudiadas en el capítulo 7, las cuales contrastan hipótesis en torno a un cierto valor para el parámetro p de una población que sigue una distribución binomial (o la diferencia de los parámetros $p_1 - p_2$ de dos de estas poblaciones).

En este capítulo conoceremos algunas pruebas para inferir acerca de proporciones cuyas hipótesis nula y alternativa **no mencionan parámetro alguno**. Es más, **ninguna de ellas hace alguna suposición sobre la distribución de la población** desde donde proviene la muestra analizada. Es por esta razón que a estas pruebas (y a otras que se abordan en capítulos posteriores) se les denomina **no paramétricas o libres de distribución**.

Las pruebas no paramétricas nos ofrecen una ventaja evidente: **son menos restrictivas** que las pruebas paramétricas, porque imponen menos supuestos a las poblaciones para poder trabajar con ellas. Asegurar que una población sigue una distribución normal o binomial, por ejemplo, puede ser una tarea difícil y, en la práctica, no es infrecuente encontrarse con conjuntos de datos que no parecen seguir alguna de estas distribuciones. Pero... si las pruebas no paramétricas parecen tan ventajosas, ¿por qué no usarlas siempre? Por dos grandes razones:

- Las pruebas no paramétricas **nos entregan menos información**. Como veremos en este capítulo para el caso de las proporciones, estas pruebas se limitan a trabajar con hipótesis del tipo “las poblaciones muestran las mismas proporciones” versus “las poblaciones muestran proporciones distintas”, pero **ninguna indica cuáles serían esas proporciones** en realidad, ni siquiera si es mayor en una o en la otra.
- Cuando sí se cumplen las condiciones para aplicar una prueba paramétrica, las versiones no paramétricas presentan **menor poder estadístico** y, en consecuencia, suelen necesitar muestras de mayor tamaño para detectar diferencias significativas que pudieran existir entre las poblaciones comparadas.

Como ya hemos dicho, en este capítulo conoceremos algunas pruebas no paramétricas para estudiar la relación entre dos variables categóricas, con base en Diez et al. (2017, pp. 286-302), Pértega y Pita (2004), Glen (2016b) y Mangiafico (2016).

8.1 PRUEBA CHI-CUADRADO DE PEARSON

Conocida también como **Prueba χ^2 de Asociación**, la **prueba chi-cuadrado de Pearson** sirve para inferir con proporciones cuando disponemos de dos variables categóricas y una de ellas es dicotómica (es decir, tiene solo dos niveles). En este caso, podemos registrar las frecuencias observadas para las posibles combinaciones de ambas variables mediante una tabla de contingencia o matriz de confusión, como ya estudiamos en el capítulo 2. En adelante, nos referiremos a cada una de estas combinaciones como un grupo.

Debemos verificar algunas condiciones antes de poder usar la prueba chi-cuadrado:

1. Las observaciones deben ser independientes entre sí.
2. Debe haber al menos 5 observaciones esperadas en cada grupo.

La primera de estas condiciones ya la hemos encontrado antes, mientras que explicaremos la segunda a medida que avancemos en el estudio de la prueba chi-cuadrado.

Si bien en esta sección estamos hablando de una única prueba, que sigue siempre el mismo procedimiento, es común encontrarla como tres pruebas diferentes:

- Prueba χ^2 de homogeneidad.
- Prueba χ^2 de bondad de ajuste
- Prueba χ^2 de independencia.

La diferencia entre ellas es **conceptual** (no matemática) y tiene relación con cómo se miren las variables y las poblaciones involucradas en el problema.

8.1.1 Prueba chi-cuadrado de homogeneidad

Esta prueba resulta adecuada si queremos determinar si **dos poblaciones** (la variable dicotómica) presentan **las mismas proporciones en los diferentes niveles de una variable categórica**.

Por ejemplo, supongamos que la Sociedad Científica de Computación (SCC) ha realizado una encuesta a 300 programadores con más de 3 años de experiencia de todo el país, escogidos al azar, y les ha preguntado cuál es su lenguaje de programación favorito. La tabla 8.1 muestra las preferencias para cada lenguaje, separadas en programadores (varones) y programadoras (mujeres). ¿Son similares las preferencias de lenguaje de programación entre hombres y mujeres?

Lenguaje	C	Java	Python	Ruby	Otro	Total
Programadores	42	56	51	27	24	200
Programadoras	25	24	27	15	9	100
Total	67	80	78	42	33	300

Tabla 8.1: tabla de frecuencias para el lenguaje de programación favorito de la muestra.

Si fuera cierto que ambas poblaciones tienen las mismas preferencias, esperaríamos encontrar proporciones similares en las muestras, pese a la variabilidad. En consecuencia, necesitamos determinar si las diferencias entre las cantidades observadas y las esperadas son lo suficientemente grandes como para proporcionar evidencia convincente de que las preferencias son disímiles. La tabla 8.2 muestra las frecuencias esperadas para cada lenguaje de programación bajo este supuesto, calculadas mediante la ecuación 8.1, donde:

- n_i : total de observaciones en la fila i .
- n_j : total de observaciones en la columna j .
- n : tamaño de la muestra.

$$E_{i,j} = \frac{n_i \cdot n_j}{n} \quad (8.1)$$

Ahora que ya sabemos cómo determinar la cantidad de observaciones esperadas en cada grupo, podemos verificar que, para cada caso, este valor es mayor que 5. Adicionalmente, es razonable suponer la muestra representa menos del 10 % de los programadores del país y sabemos que fue seleccionada de manera aleatoria, por lo que podemos proceder con la prueba χ^2 de homogeneidad.

Lenguaje	C	Java	Python	Ruby	Otro	Total
Programadores	44,7	53,3	52,0	28,0	22,0	200
Programadoras	22,3	26,7	26,0	14,0	11,0	100
Total	67	80	78	42	33	300

Tabla 8.2: frecuencias esperadas si hombres y mujeres tienen las mismas preferencias.

Las hipótesis a contrastar son:

H_0 : programadores hombres y mujeres tienen las mismas preferencias en lenguaje de programación favorito (ambas poblaciones muestran las mismas proporciones para cada lenguaje estudiado).

H_A : programadores hombres y mujeres tienen preferencias distintas en lenguajes de programación favorito.

Recordemos que la primera aproximación para construir un estadístico de prueba adecuado está dada por la ecuación 4.5, que reproducimos aquí:

$$Z = \frac{\text{estimador puntual} - \text{valor nulo}}{SE_{\text{estimador puntual}}}$$

Podríamos usar esta fórmula de la diferencia estandarizada para cada uno de los grupos, donde:

- El estimador puntual corresponde a la frecuencia observada para el grupo.
- El valor nulo corresponde a la frecuencia esperada para el grupo.
- El error estándar del estimador puntual es la raíz cuadrada del valor nulo.

Así, para los programadores (varones) en C se tiene:

$$Z_{H_C} = \frac{42 - 44,7}{\sqrt{44,7}} = -0,404$$

Al repetir el procedimiento para cada grupo, se obtienen los valores Z presentados en la tabla 8.3.

Lenguaje	C	Java	Python	Ruby	Otro
Programadores	-0,404	0,370	-0,139	-0,189	0,426
Programadoras	0,572	-0,523	0,196	0,267	-0,603

Tabla 8.3: valor Z para cada grupo.

Pero necesitamos transformar estos estadísticos por cada grupo en un único estadístico de prueba. Para ello, se considera la suma de sus cuadrados, pues así todos los valores son positivos y las diferencias significativas se incrementan aún más (como en el caso de la varianza). Así, se define el estadístico de prueba χ^2 , definido en la ecuación 8.2, donde m y n son, respectivamente, la cantidad de filas y la cantidad de columnas de la tabla de frecuencias, sin considerar los totales (puede ser útil en este punto repasar lo que aprendimos en el capítulo 3 sobre la distribución χ^2).

$$\chi^2 = \sum_{i=1}^m \sum_{j=1}^n Z_{ij}^2 = \sum_{i=1}^m \sum_{j=1}^n \frac{(\text{cantidad observada} - \text{cantidad esperada})^2}{\text{cantidad esperada}} \quad (8.2)$$

Para el ejemplo tenemos entonces:

$$\begin{aligned} \chi^2 = & (-0.404)^2 + (0.370)^2 + (-0.139)^2 + (-0.189)^2 + (0.426)^2 + (0.572)^2 + \\ & + (-0.523)^2 + (0.196)^2 + (0.267)^2 + (-0.603)^2 = 1,611 \end{aligned}$$

Como estamos sumando $m \cdot n$ valores Z al cuadrado, el estadístico χ^2 sigue una distribución chi-cuadrado, con $\nu = (m - 1) \cdot (n - 1)$ grados de libertad. En el ejemplo, $\nu = (2 - 1) \cdot (5 - 1) = 4$.

El valor p para la prueba chi-cuadrado está dado por el área bajo la curva de la distribución chi-cuadrado con valores mayores al obtenido para el estadístico de prueba. En este caso, gracias a la llamada en R `pchisq(1.611, df = 4, lower.tail = FALSE)`, obtenemos que $p = 0,807$. Suponiendo un nivel de significación $\alpha = 0,05$, $p > \alpha$, por lo que se falla al rechazar la hipótesis nula. Es decir, no hay evidencia suficientemente fuerte que sugiera, con 95 % de confianza, que programadores hombres y mujeres prefieran lenguajes de programación distintos.

En R, podemos realizar la prueba chi-cuadrado de homogeneidad como muestra el script 8.1, usando para ello la función `chisq.test(x)`, donde x corresponde a la matriz de confusión.

Al ejecutar el script, debemos tener en cuenta que el valor p obtenido usando R es ligeramente diferente debido a los redondeos aplicados en la tabla 8.2 y al resolver la ecuación 8.2.

Script 8.1: prueba chi-cuadrado de homogeneidad.

```
1 # Crear tabla de contingencia.
2 programadores <- c(42, 56, 51, 27, 24)
3 programadoras <- c(25, 24, 27, 15, 9)
4
5 tabla <- as.table(rbind(programadores, programadoras))
6
7 dimnames(tabla) <- list(sexo = c("programadores", "programadoras"),
8                           lenguajes = c("C", "Java", "Python", "Ruby", "Otro"))
9
10 print(tabla)
11
12 # Hacer prueba chi-cuadrado de homogeneidad.
13 prueba <- chisq.test(tabla)
14 print(prueba)
```

8.1.2 Prueba chi-cuadrado de bondad de ajuste

Esta prueba permite comprobar si una distribución de frecuencias observada se asemeja a una distribución esperada. Usualmente se emplea para comprobar si una muestra es representativa de la población (NIST/SEMATECH, 2013, p. 1.3.5.15).

Para entender mejor esta idea, supongamos ahora que una gran empresa de desarrollo de software cuenta con una nómina de 660 programadores, especialistas en diferentes lenguajes de programación. El gerente ha seleccionado un subconjunto de 55 programadores, supuestamente de forma aleatoria, para enviarlos a cursos de perfeccionamiento en sus respectivos lenguajes, pero el sindicato lo ha acusado de “seleccionar estas personas a conveniencia de los intereses mezquinos de la gerencia, impidiendo que el grupo sea representativo a fin de asegurar una mejora en la productividad de toda la empresa”. Ante el inminente riesgo de movilizaciones, el gerente necesita demostrar que el grupo seleccionado es una muestra representativa de sus programadores.

La tabla 8.4 muestra la cantidad de especialistas en cada lenguaje, tanto para la nómina de la empresa como para la muestra seleccionada.

Como ya es habitual, comenzemos por verificar las condiciones. Puesto que la muestra representa menos del 10 % de la población y fue elegida de manera aleatoria, las observaciones son independientes entre sí.

La segunda condición resulta algo más compleja. Comencemos por calcular la proporción de programadores de la nómina (población) especialista en cada lenguaje. Para el caso de C, tenemos:

Lenguaje	C	Java	Python	Ruby	Otro
Nómina	236	78	204	76	66
Muestra	17	9	14	10	5

Tabla 8.4: frecuencias por lenguaje de programación para la toda la nómina y para la muestra.

$$P_C = \frac{n_C}{n} = \frac{236}{660} = 0,358$$

En consecuencia, esperaríamos la misma proporción de especialistas en C en la muestra, es decir:

$$E_C = P_C \cdot n = 0,358 \cdot 55 = 19,690$$

Repitiendo este proceso para los lenguajes restantes, obtenemos las proporciones para la población y valores esperados para la muestra que se presentan en la tabla 8.5. En ella podemos ver que para cada grupo se esperan más de 5 observaciones, por lo que se verifica la segunda condición.

Lenguaje	C	Java	Python	Ruby	Otro
Proporciones nómina	0,358	0,118	0,309	0,115	0,100
Valores esperados muestra	19,690	6,490	16,995	6,325	5,500

Tabla 8.5: proporciones de la población y valores esperados de la muestra.

En este ejemplo, las hipótesis a contrastar son:

- H_0 : las proporciones de especialistas en cada lenguaje son las mismas para la nómina y la muestra.
 H_A : las proporciones de especialistas en cada lenguaje son diferentes en la nómina que en la muestra.

En este caso, se puede proceder de igual manera que para la prueba de homogeneidad, como muestra el script 8.2. Para este ejemplo, el valor p resultante es $p = 0,461$, por lo que se falla al rechazar la hipótesis nula con un nivel de significación $\alpha = 0,05$. En consecuencia, podemos concluir con 95 % de confianza que no hay evidencia de que la muestra seleccionada no sea representativa de la nómina de programadores de la empresa, por lo que la acusación del sindicato no tiene fundamentos.

Script 8.2: prueba chi-cuadrado de bondad de ajuste.

```

1 # Crear tabla de contingencia.
2 nomina <- c(236, 78, 204, 76, 66)
3 muestra <- c(17, 9, 14, 10, 5)
4
5 tabla <- as.table(rbind(nomina, muestra))
6
7 dimnames(tabla) <- list(grupo = c("Nómina", "Muestra"),
8                           lenguajes = c("C", "Java", "Python", "Ruby", "Otro"))
9
10 print(tabla)
11
12 # Verificar si se esperan más de 5 observaciones por cada grupo.
13 n_nomina <- sum(nomina)
14 n_muestra <- 55
15 proporciones <- round(nomina/n_nomina, 3)
16 esperados <- round(proporciones * n_muestra, 3)
17 print(esperados)
18
19 # Hacer prueba chi-cuadrado de homogeneidad.
20 prueba <- chisq.test(tabla, correct = FALSE)
21 print(prueba)

```

8.1.3 Prueba chi-cuadrado de independencia

Esta prueba permite **determinar si dos variables categóricas, de una misma población, son estadísticamente independientes** o si, por el contrario, están relacionadas.

Tomemos en este caso como ejemplo que un micólogo desea determinar si existe relación entre la forma del sombrero de los hongos y si estos son o no comestibles. Para ello, tras recolectar una muestra de 8.120 hongos, obtiene la tabla de contingencia que se muestra en la tabla 8.6¹.

		Forma del sombrero					
		Campana	Convexo	Hundido	Nudoso	Plano	Total
Clase	Comestible	404	1.948	32	228	1.596	4.208
	Venenoso	48	1.708	0	600	1.556	3.912
	Total	452	3.656	32	828	3.152	8.120

Tabla 8.6: tabla de contingencia para las características de los hongos.

Una vez más, comenzemos por verificar las condiciones. Podemos suponer que la muestra fue obtenida de manera aleatoria, ya que se trata de un estudio publicado en una revista científica, y, desde luego, representa menos del 10 % de la población mundial de hongos. En consecuencia, se verifica la condición de independencia de las observaciones en las muestras.

Ahora debemos determinar cuántas observaciones esperaríamos tener en cada grupo si las variables fueran independientes. En este caso, la frecuencia esperada para cada celda está dado por la ecuación 8.3, donde:

- n_i : total de observaciones en la fila i .
- n_j : total de observaciones en la columna j .
- n : tamaño de la muestra.

$$E_{i,j} = \frac{n_i \cdot n_j}{n} \quad (8.3)$$

De acuerdo a esto, la cantidad de hongos comestibles con sombrero en forma de campana que esperaríamos encontrar es:

$$E_{1,1} = \frac{4.208 \cdot 452}{8.120} = 234,238$$

Si replicamos este cálculo para cada celda de nuestra matriz de confusión, se obtienen los valores esperados que se presentan en la tabla 8.7. Podemos ver que todos los valores esperados superan las 5 observaciones, por lo que podemos proceder con la prueba χ^2 de independencia.

		Forma del sombrero					
		Campana	Convexo	Hundido	Nudoso	Plano	
Clase	Comestible	234,238	1.894,636	16,583	429,092	1.633,450	
	Venenoso	217,762	1.761,364	15,417	398,908	1.518,550	
	Total	452	3.656	32	828	3.152	8.120

Tabla 8.7: frecuencias esperadas para los hongos.

En este caso, las hipótesis a docimar son:

H_0 : las variables clase y forma del sombrero son independientes.

H_A : las variables clase y forma del sombrero están relacionadas.

¹Datos obtenidos desde el conjunto de datos Mushroom, disponible en <https://archive.ics.uci.edu/ml/datasets/mushroom> (última visita: 26-04-2021).

Al ejecutar la prueba en R (script 8.3) obtenemos que el valor para el estadístico de prueba es $\chi^2 = 485,64$, con $\nu = 4$ grados de libertad y un valor $p < 2 \cdot 10^{-16}$. Aún para un nivel de significación muy exigente, como $\alpha = 0,01$, el valor p obtenido nos permite rechazar la hipótesis nula en favor de la hipótesis alternativa. En consecuencia, concluimos con 99 % de confianza que las variables clase y forma del sombrero están relacionadas (son dependientes).

Script 8.3: prueba chi-cuadrado de independencia.

```

1 # Crear tabla de contingencia.
2 comestible <- c(404, 1948, 32, 228, 1596)
3 venenoso <- c(48, 1708, 0, 600, 1556)
4
5 tabla <- as.table(rbind(comestible, venenoso))
6
7 dimnames(tabla) <- list(tipo = c("comestible", "venenoso"),
8                           sombrero = c("campana", "convexo", "hundido",
9                                       "nudoso", "plano")))
10
11
12 print(tabla)
13
14 # Hacer prueba chi-cuadrado de independencia.
15 prueba <- chisq.test(tabla)
16 cat("\nLa prueba internamente calcula los valores esperados:\n")
17 esperados <- round(prueba[["expected"]], 3)
18 print(esperados)
19
20 cat("\nResultado de la prueba:\n")
21 print(prueba)

```

8.2 PRUEBAS PARA MUESTRAS PEQUEÑAS

Hemos visto que la prueba χ^2 nos pide que las observaciones esperadas para cada grupo sean a lo menos 5. Sin embargo, hay escenarios donde esta condición no se cumple, por lo que debemos recurrir a alguna alternativa.

8.2.1 Prueba exacta de Fisher

La **prueba exacta de Fisher** es una alternativa a la prueba χ^2 de independencia en el caso de que **ambas variables sean dicotómicas**. Así, las hipótesis a contrastar son:

- H_0 : las variables son independientes.
- H_A : las variables están relacionadas.

En este escenario, las frecuencias de la muestra pueden resumirse en una tabla de contingencia de 2×2 , como muestra la tabla 8.8.

Si se asume independencia entre ambas variables y los totales por filas y columnas son fijos, la **probabilidad exacta de observar el conjunto de frecuencias de la tabla 8.8** está dada por la ecuación 8.4,

		Variable 1		Total
		Presente	Ausente	
Variable 2	Presente	a	b	
	Ausente	c	d	
	Total	a+c	b+d	n

Tabla 8.8: tabla de contingencia para dos variables categóricas con dos niveles cada una.

correspondiente a la función de distribución hipergeométrica.

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!} \quad (8.4)$$

La prueba lleva en su nombre la palabra **exacta** porque internamente construye todas las tablas posibles con los mismos totales marginales que recibe como entrada y, para cada una de ellas, determina la probabilidad exacta de observarla. El valor p corresponde en este caso a la suma de las probabilidades de todas las tablas con probabilidad menor o igual que la tabla dada.

Para entender mejor esta prueba, supongamos que un controvertido estudio desea determinar si dos vacunas, Argh y Grrr, son igualmente efectivas para inmunizar a la población ante una mordida de vampiro. Para ello, los investigadores reclutaron a 17 voluntarios de todo el mundo, de los cuales 6 recibieron la vacuna Argh y los 11 restantes, la Grrr. Al cabo de tres meses, sometieron a cada uno de los participantes a una mordida de vampiro y observaron que ninguno de los voluntarios que recibieron la vacuna Argh resultó afectado, mientras que 5 de los que recibieron la vacuna Grrr se convirtieron en vampiros, como resume la tabla 8.9.

		Vacuna		Total
		Argh	Grrr	
Resultado	Vampiro	0	5	
	Humano	6	6	
	Total	6	11	17

Tabla 8.9: tabla de contingencia con los contagios producidos en el experimento.

La probabilidad de observar un conjunto de frecuencias con los mismos totales por fila y por columna, si las variables son realmente independientes está dada por:

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!} = \frac{5!12!6!11!}{17!0!5!6!6!} = 0,075$$

Son cinco las posibles tablas (además de la obtenida) con iguales valores marginales, como podemos ver en la tabla 8.10.

Calculando las probabilidades para cada una de ellas de acuerdo a la ecuación 8.4, se tiene que:

- Probabilidad para la tabla 8.10a: 0,001.
- Probabilidad para la tabla 8.10b: 0,320.
- Probabilidad para la tabla 8.10c: 0,027.
- Probabilidad para la tabla 8.10d: 0,400.
- Probabilidad para la tabla 8.10e: 0,178.

Así, el valor p está dado por la suma de las probabilidades de las tablas con probabilidad menor o igual a la de los datos observados:

$$p = 0,075 + 0,001 + 0,027 = 0,103$$

		Vacuna		Total
		Argh	Grrr	
Resultado	Infectado	5	0	5
	Sano	1	11	12
	Total	6	11	17

(a)

		Vacuna		Total
		Argh	Grrr	
Resultado	Infectado	1	4	5
	Sano	5	7	12
	Total	6	11	17

(b)

		Vacuna		Total
		Argh	Grrr	
Resultado	Infectado	4	1	5
	Sano	2	10	12
	Total	6	11	17

(c)

		Vacuna		Total
		Argh	Grrr	
Resultado	Infectado	2	3	5
	Sano	4	8	12
	Total	6	11	17

(d)

		Vacuna		Total
		Argh	Grrr	
Resultado	Infectado	3	2	5
	Sano	3	9	12
	Total	6	11	17

(e)

Tabla 8.10: tablas con los mismos valores marginales que los obtenidos.

Considerando un nivel de significación $\alpha = 0,05$, se falla al rechazar la hipótesis nula. En consecuencia, se concluye con 95 % de confianza que no existe evidencia de que exista una asociación entre la cantidad de nuevos vampiros y la vacuna recibida.

En R, podemos llevar a cabo esta prueba mediante la función `fisher.test(x, conf.level)`, donde `x` corresponde a la tabla de contingencia y `conf.level`, al nivel de confianza. El script 8.4 muestra el su uso para el ejemplo (con una pequeña diferencia en el valor p obtenido debido a los redondeos efectuados en el cálculo anterior).

Script 8.4: prueba exacta de Fisher.

```

1 # Construir la tabla de contingencia.
2 vacuna <- c(rep("Argh", 6), rep("Grrr", 11))
3 resultado <- c(rep("Humano", 12), rep("Vampiro", 5))
4 datos <- data.frame(resultado, vacuna)
5 tabla <- xtabs(~., datos)
6 print(tabla)
7
8 # Aplicar prueba exacta de Fisher.
9 alfa <- 0.05
10 prueba <- fisher.test(tabla, 1-alfa)

```

```
11 print(prueba)
```

8.2.2 Prueba de mcNemar

Esta prueba resulta apropiada cuando una misma característica, con respuesta dicotómica, se mide en dos ocasiones diferentes para los mismos sujetos (muestras pareadas) y queremos determinar si se produce o no un cambio significativo entre ambas mediciones. Una vez más, podemos registrar las frecuencias en una matriz de confusión como la que vimos en 8.8. En ella, podemos ver que las celdas a y d corresponde a instancias en que no hay cambios. La celda b en dicha tabla representa a las instancias que cambian de **Presente** a **Ausente** y la celda c , a instancias que cambian de **Ausente** a **Presente**.

Las hipótesis asociadas a la prueba de mcNemar son:

H_0 : **no** hay cambios significativos en las respuestas.

H_A : **sí** hay cambios significativos en las respuestas.

Puesto que nos interesa medir los cambios, solo nos sirven las celdas b y c de la tabla de contingencia. La cantidad de instancias en que se producen cambios es $b + c$ y, de acuerdo a la hipótesis nula, se esperaría que $(b+c)/2$ cambien en un sentido y que las $(b+c)/2$ restantes lo hicieran en sentido contrario. Así, b y c cuentan respectivamente los éxitos y los fracasos de una distribución binomial de $b + c$ intentos con probabilidad de éxito igual a $1/2$. Cuando $(b+c) > 10$, esta distribución binomial se asemeja a una distribución normal con la misma media $((b+c)/2)$ y desviación estándar $\sqrt{(b+c)/4}$, a partir de la cual se puede obtener un estadístico z . Sin embargo, la mayoría de los paquetes de software para estadística (incluido R) reportan el cuadrado de dicho estadístico (e ignoran completamente los casos en que hay 10 o menos cambios entre las mediciones), el cual sigue una distribución χ^2 con un grado de libertad y se calcula como muestra la ecuación 8.5 (Agresti, 2019).

$$\chi^2 = \frac{(b - c)^2}{b + c} \quad (8.5)$$

Puesto que los datos siguen una distribución binomial (discreta), pero se está usando como aproximación la distribución chi-cuadrado (continua), suele emplearse un **factor de corrección de continuidad** propuesta por Frank Yates en 1934. El estadístico de prueba con la corrección de Yates se calcula en realidad como muestra la ecuación 8.6.

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c} \quad (8.6)$$

Para ilustrar el funcionamiento de la prueba de mcNemar, suponga que un cientista de datos ha construido dos modelos para predecir, a partir de las notas obtenidas en cursos previos, si sus estudiantes aprobarán o no la asignatura de aprendizaje automático. Al probar sus modelos con los 25 estudiantes del semestre anterior, observó que predijeron el resultado final de cada estudiante como muestra la tabla 8.11 y se resume en la matriz de confusión de la tabla 8.12.

El cientista de datos desea saber si existe diferencia entre el desempeño de ambos algoritmos, por lo que decide emplear la prueba de mcNemar. Al calcular el estadístico de prueba (con el factor de corrección), obtiene:

$$\chi^2 = \frac{(|5 - 7| - 1)^2}{5 + 7} = \chi^2 = \frac{(5 - 7)^2}{5 + 7} = 0,083$$

Alumno	Modelo 1	Modelo 2
1	Correcto	Correcto
2	Correcto	Correcto
3	Correcto	Correcto
4	Correcto	Correcto
5	Correcto	Correcto
6	Correcto	Correcto
7	Correcto	Correcto
8	Correcto	Correcto
9	Correcto	Correcto
10	Correcto	Incorrecto
11	Correcto	Incorrecto
12	Correcto	Incorrecto
13	Correcto	Incorrecto
14	Correcto	Incorrecto
15	Correcto	Incorrecto
16	Correcto	Incorrecto
17	Incorrecto	Incorrecto
18	Incorrecto	Incorrecto
19	Incorrecto	Incorrecto
20	Incorrecto	Incorrecto
21	Incorrecto	Correcto
22	Incorrecto	Correcto
23	Incorrecto	Correcto
24	Incorrecto	Correcto
25	Incorrecto	Correcto

Tabla 8.11: resultados de la predicción para cada estudiante con ambos modelos.

		Modelo 1		Total
		Correcto	Incorrecto	
Modelo 2	Correcto	9	5	14
	Incorrecto	7	4	11
	Total	16	9	25

Tabla 8.12: tabla de contingencia con las predicciones de los resultados finales de los estudiantes.

El valor p está dado por el área bajo la cola superior de la distribución chi-cuadrado, que en R puede calcularse como `pchisq(0.083, 1, lower.tail = FALSE)`, obteniéndose que $p = 0,773$. En consecuencia, se falla al rechazar la hipótesis nula (para un nivel de significación $\alpha = 0,05$) y se concluye que no hay evidencia suficiente para creer que existe una diferencia en el desempeño de ambos clasificadores.

La función de R para esta prueba, que por defecto aplica el factor de corrección, es `mcNemar.test(x)`, donde x corresponde a la tabla de contingencia. El script 8.5 muestra su aplicación para el ejemplo dado.

Script 8.5: prueba de McNemar.

```

1 # Construir la tabla de contingencia.
2 alumno <- seq(1:25)
3 modelo_1 <- c(rep("Correcto", 16), rep("Incorrecto", 9))
4 modelo_2 <- c(rep("Correcto", 9), rep("Incorrecto", 11), rep("Correcto", 5))
5 datos <- data.frame(alumno, modelo_2, modelo_1)
6 tabla <- table(modelo_2, modelo_1)
7 print(tabla)
8
9 # Aplicar prueba de McNemar.

```

```

10 prueba <- mcnemar.test(tabla)
11 print(prueba)

```

8.3 PRUEBA Q DE COCHRAN

La **prueba Q de Cochran** es una extensión de la prueba de McNemar, adecuada cuando la variable de respuesta es dicotómica y la variable independiente tiene más de dos observaciones pareadas (cuando ambas variables son dicotómicas, esta prueba es equivalente a la de McNemar). Como tal, debería estar incluida en la sección precedente, pero le dedicaremos una sección aparte pues la explicación requiere de algunos conceptos importantes que no hemos estudiado aún.

Veamos esta prueba por medio de un ejemplo. Elsa Capunta, estudiante de un curso de algoritmos, tiene como tarea determinar si existe una diferencia significativa en el desempeño de tres metaheurísticas que buscan resolver el problema del vendedor viajero. Para ello, el profesor le ha proporcionado los datos presentados en la tabla 8.13, donde la columna `instancia` identifica cada instancia del problema empleada para evaluar las metaheurísticas y las restantes columnas indican si la metaheurística en cuestión encontró (1) o no (0) la solución óptima para dicha instancia.

Instancia	Simulated Annealing	Colonia de hormigas	Algoritmo genético
1	0	0	1
2	1	0	0
3	0	1	1
4	0	0	1
5	0	0	1
6	0	1	1
7	0	0	0
8	1	0	1
9	0	0	0
10	0	1	1
11	0	0	1
12	0	0	0
13	1	0	0
14	0	0	1
15	0	1	1

Tabla 8.13: resultados de las metaheurísticas para cada instancia con ambos modelos.

Las hipótesis contrastadas por la prueba Q de Cochran son:

- H_0 : la proporción de “éxitos” es la misma para todos los grupos.
 H_A : la proporción de “éxitos” es distinta para al menos un grupo.

Como ya debemos suponer, esta prueba también requiere que se cumplan algunas condiciones:

1. La variable de respuesta es dicotómica.
2. La variable independiente es categórica.
3. Las observaciones son independientes entre sí.
4. El tamaño de la muestra es lo suficientemente grande. Glen (2016b) sugiere que $n \cdot k \geq 24$, donde n es el tamaño de la muestra (la cantidad de instancias, para el ejemplo) y k , la cantidad de niveles en la variable independiente.

El estadístico de prueba se calcula como muestra la ecuación 8.7, donde:

- b : cantidad de bloques.
- k : cantidad de bloques (niveles de la variable independiente).
- x_j : total de éxitos en la columna j .
- x_i : total de éxitos en la fila i .
- N : número total de éxitos.

$$Q = k(k-1) \frac{\sum_{j=1}^k (x_j - \frac{N}{k})^2}{\sum_{i=1}^b x_i(k-x_i)} \quad (8.7)$$

Podemos ver que los cálculos que se llevan a cabo para esta prueba son complejos, por lo que suele hacerse mediante software. En R, esta prueba está implementada en la función `cochran.qtest(formula, data, alpha = 0.05)` del paquete `RVAideMemoire`, donde:

- `formula`: fórmula de la forma `respuesta ~ independiente | bloques`.
- `data`: matriz de datos en formato largo.
- `alpha`: nivel de significación.

Al ejecutar el script 8.6, obtenemos el resultado que se muestra en la figura 8.1. Tenemos que el valor p es $p = 0,028$, menor que el nivel de significación $\alpha = 0,05$, por lo que rechazamos la hipótesis nula en favor de la hipótesis alternativa. En consecuencia, Elsa concluye con 95 % de confianza que al menos una de las metaheurísticas tiene un desempeño diferente a las demás.

```
Cochran's Q test

data: resultado by metaheuristica, block = instancia
Q = 7.1667, df = 2, p-value = 0.02778
alternative hypothesis: true difference in probabilities is not equal to 0
sample estimates:
proba in group annealing proba in group genetico proba in group hormigas
0.2000000          0.6666667          0.2666667

Pairwise comparisons using Wilcoxon sign test

annealing genetico
genetico   0.09814   -
hormigas  1.00000  0.09375

P value adjustment method: fdr
```

Figura 8.1: resultado de la prueba Q de Cochran.

Todo buen estudiante sabe que Elsa debe entregar en su tarea una respuesta más detallada que la que hemos obtenido hasta ahora, pues el profesor esperaría un análisis de las diferencias.

En este punto, debemos mencionar que la hipótesis nula de la prueba Q de Cochran no es específica, sino que comprueba la igualdad de todas las proporciones. Esta clase de hipótesis nula suele llamarse **ómnibus** (en ocasiones también colectiva o global). Así, se dice que la prueba Q de Cochran es una prueba ómnibus porque tiene esta clase de hipótesis nula, con la dificultad de que solo detecta si existe al menos bloque con una proporción de “éxito” diferente. Sin embargo, de ser afirmativa la respuesta, no nos dice qué grupos presentan diferencias (Lane, s.f.). Desde luego, existen métodos para responder a esta última pregunta, llamados **pruebas post-hoc**, o también *a posteriori*. Reciben este nombre porque se realizan una vez que se ha concluido gracias a la prueba ómnibus que existen diferencias significativas.

Algo importante que debemos recordar: **solo haremos un procedimiento post-hoc si la prueba ómnibus rechaza la hipótesis nula** en favor de la hipótesis alternativa. Además, el procedimiento post-hoc

realizado debe considerar el mismo nivel de significación que la prueba ómnibus.

En el caso de la prueba Q de Cochran, el procedimiento post-hoc consiste en efectuar pruebas de McNemar entre cada par de bloques. R nos permite hacer esto mediante la función `pairwiseMcNemar(formula, data, method)` del paquete `rcompanion`, donde `formula` y `data` son las mismas que para la prueba Q de Cochran y `method` nos permite determinar el método para ajustar los valores p de las comparaciones. Pero... ¿por qué queríamos ajustar los valores p?

Como explican Goeman y Solari (2014), cuando contrastamos hipótesis acotamos la probabilidad de cometer errores tipo I por medio del nivel de significación α . Sin embargo, cuando hacemos múltiples contrastes de hipótesis simultáneamente, cada uno de ellos tendrá una probabilidad α de cometer un error de tipo I. Esto se traduce en un **incremento de la probabilidad de cometer este tipo de errores** a medida que aumenta la cantidad de hipótesis contrastadas y, en consecuencia, en una reducción del poder estadístico.

Muchos factores de corrección tienen por objeto distribuir el nivel de significación empleado para la prueba ómnibus en cada prueba de pares de bloques. El método más sencillo para ajustar los valores p es la **corrección de Bonferroni**. Como explica la ayuda de R, esta corrección simplemente multiplica el valor p obtenido en cada prueba por la cantidad de pruebas realizadas. En general, no se recomienda el uso del método de Bonferroni, especialmente si el número de grupos es alto, pues es considerado muy **conservador**, lo que significa que mantiene la probabilidad de cometer un error tipo I más baja que el nivel de significación establecido (y es, por ende, más propensa a cometer errores tipo II).

Otra alterativa es la **corrección de Holm** (Glen, 2016c), con mayor poder estadístico que la de Bonferroni. Esta corrección comienza por efectuar las pruebas entre pares de bloques y luego ordena los valores p en forma creciente. A continuación, se calcula el factor de Holm, HB , para cada par de bloques, dado por la ecuación 8.8, donde:

- α : nivel de significación.
- N : cantidad de comparaciones efectuadas.
- i : importancia de la comparación (posición en la lista de valores p ordenados).

$$HB_i = \frac{\alpha}{N - i + 1} \quad (8.8)$$

Luego, se compara el valor p con su respectivo factor de Holm y, si el valor p es menor, se considera que existe una diferencia significativa. R implementa esta corrección de manera ligeramente diferente, de modo que el valor p ajustado pueda ser comparado con el nivel de significación original.

Si has estado leyendo de manera atenta, habrás notado que en el resultado entregado por `cochran.qtest()` para el ejemplo (figura 8.1) aparece otro procedimiento post-hoc adecuado para la prueba Q de Cochran, aunque no lo presentaremos aquí pues se basa en una prueba que estudiaremos en capítulos posteriores.

El script 8.6 incluye también los procedimientos *post-hoc* mediante pruebas de McNemar usando las correcciones de Holm y Bonferroni, obteniéndose los resultados que se muestran en la figura 8.2.

Podemos ver en la figura 8.2 que, aún cuando la prueba Q de Cochran indica que existen diferencias significativas entre las metaheurísticas, ninguno de los procedimientos post-hoc ha detectado diferencias significativas entre pares de bloques. En consecuencia, la respuesta que Elsa debe dar a su profesor es que la evidencia no es lo suficientemente fuerte para poder afirmar que existen diferencias entre las metaheurísticas, pero que podría ser adecuado hacer un estudio con una muestra mayor puesto que los resultados de la prueba Q de Cochran y de los procedimientos post-hoc son contradictorios.

Script 8.6: prueba Q de Cochran.

```
1 library(tidyverse)
2 library(RVAideMemoire)
3 library(rcompanion)
4
5 # Crear matriz de datos.
6 instancia <- 1:15
```

Cochran's Q test

```

Procedimiento post-hoc con corrección de Bonferroni
$Test.method
  Test
  1 exact

$Adustment.method
  Method
  1 bonferroni

$Pairwise
  Comparison Successes Trials p.value p.adjust
  1 annealing - genetico = 0      2     11  0.0654  0.1960
  2 annealing - hormigas = 0      3      7   1  1.0000
  3 genetico - hormigas = 0      6      6  0.0313  0.0939

```

Procedimiento post-hoc con corrección de Holm

```

$Test.method
  Test
  1 exact

$Adustment.method
  Method
  1 holm

$Pairwise
  Comparison Successes Trials p.value p.adjust
  1 annealing - genetico = 0      2     11  0.0654  0.1310
  2 annealing - hormigas = 0      3      7   1  1.0000
  3 genetico - hormigas = 0      6      6  0.0313  0.0939

```

Figura 8.2: resultados de los procedimientos post-hoc.

```

7 annealing <- c(0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)
8 hormigas <- c(0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1)
9 genetico <- c(1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1)
10 datos <- data.frame(instancia, annealing, hormigas, genetico)
11
12 # Llevar matriz de datos a formato largo.
13 datos <- datos %>% pivot_longer(c("annealing", "hormigas", "genetico"),
14                                     names_to = "metaheuristica",
15                                     values_to = "resultado")
16
17 datos[["instancia"]] <- factor(datos[["instancia"]])
18 datos[["metaheuristica"]] <- factor(datos[["metaheuristica"]])
19
20 # Hacer prueba Q de Cochran.
21 prueba <- cochrans.qtest(resultado ~ metaheuristica | instancia,
22                           data = datos, alpha = 0.05)
23
24 print(prueba)
25
26 # Procedimiento post-hoc con corrección de Bonferroni.

```

```

27 post_hoc_1 <- pairwiseMcNemar(resultado ~ metaheuristica | instancia,
28                                     data = datos, method = "bonferroni")
29
30 cat("\nProcedimiento post-hoc con corrección de Bonferroni\n")
31 print(post_hoc_1)
32
33 # Procedimiento post-hoc con corrección de Holm.
34 post_hoc_2 <- pairwiseMcNemar(resultado ~ metaheuristica | instancia,
35                                     data = datos, method = "holm")
36
37 cat("\nProcedimiento post-hoc con corrección de Holm\n")
38 print(post_hoc_2)

```

8.4 EJERCICIOS PROPUESTOS

1. Explica cómo se calcula el estadístico χ^2 .
2. Menciona las condiciones para que una prueba de hipótesis χ^2 sea válida.
3. Da un ejemplo en que se requiera utilizar una prueba χ^2 de homogeneidad. ¿Qué hipótesis nula y alternativa serían docimadas?
4. Da un ejemplo en que se requiera utilizar una prueba χ^2 de bondad de ajuste. ¿Qué hipótesis nula y alternativa serían docimadas?
5. Da un ejemplo en que se requiera utilizar una prueba χ^2 de independencia. ¿Qué hipótesis nula y alternativa serían docimadas?
6. Un estudio clínico reclutó a 32 pacientes con fatiga crónica para determinar si un tratamiento basado en inyecciones intramusculares de magnesio es efectivo para esta condición. De los 15 pacientes que recibieron estas inyecciones, seleccionados de manera aleatoria, 12 reportaron sentirse mejor (80 %), mientras que solo 3 pacientes de los 17 que recibieron placebo (18 %) reportaron mejorías.
 - a) ¿Se cumplen las condiciones para aplicar una prueba exacta de Fisher al problema enunciado?
 - b) ¿Cuáles serían las hipótesis nula y alternativa para esta prueba?
 - c) Independientemente de la respuesta anterior, aplica la prueba usando R y luego de forma manual (Ayuda: hay 16 tablas que mantienen los totales marginales en el enunciado).
 - d) ¿A qué conclusión lleva este procedimiento?
7. Antes del debate de candidatos presidenciales, una encuesta consultó a 1.000 auditores si apoyaban o no una reforma constitucional para permitir matrimonio igualitario, encontrando 705 personas a favor y 295 en contra. Luego de que estas personas escucharon el debate, 663 se manifestaron a favor y 337 en contra de la reforma. 73 encuestados cambiaron de opinión de en contra a en apoyo de la medida, mientras que 115 cambiaron su opinión a favor para estar en contra.
 - a) ¿Se cumplen las condiciones para aplicar una prueba de McNemar al problema enunciado?
 - b) ¿Cuáles serían las hipótesis nula y alternativa si usamos esta prueba?
 - c) Independientemente de la respuesta anterior, aplica la prueba usando R y luego de forma manual.
 - d) ¿A qué conclusión lleva este procedimiento?
8. Con palabras propias ¿qué es una prueba ómnibus? ¿Con qué otros nombres se les conoce?
9. Con palabras propias ¿qué es una prueba post-hoc? ¿Cuándo se aplican?
10. Con palabras propias, cuando hay más de dos grupos ¿por qué es problemático hacer múltiples pruebas entre pares de esos grupos?
11. Las autoridades de la universidad desean conocer si las semanas de receso (sin actividades docentes) ayuda o no al descanso del estudiantado. Para eso seleccionaron 20 estudiantes de forma aleatoria y les consultaron si se sentían “cansada/o” o “descansada/o” en tres ocasiones: el lunes, miércoles y viernes de la primera semana de receso del semestre. Los resultados se muestran en la siguiente tabla, donde 0 representa cansancio y 1 descanso.

Estudiante	Lunes	Miércoles	Viernes
1	1	1	1
2	0	1	1
3	0	0	1
4	0	1	0
5	1	0	0
6	0	1	1
7	0	1	1
8	0	0	1
9	0	1	1
10	0	1	0
11	1	1	0
12	1	1	1
13	0	0	0
14	1	0	1
15	0	1	1
16	0	1	0
17	0	0	1
18	0	1	1
19	1	0	1
20	0	1	1

- a) ¿Hay diferencias entre los tres periodos de tiempo sin actividades? No olvide enunciar las hipótesis, seleccionar una prueba para docimiarlas y verificar si se cumplen las condiciones necesarias para realizar la prueba seleccionada.
- b) Si hay diferencias, ¿entre qué periodos se encuentran? No olvide justificar el procedimiento post-hoc seguido si corresponde.

CAPÍTULO 9. ANOVA DE UNA VÍA PARA MUESTRAS INDEPENDIENTES

En el capítulo 5 conocimos la prueba t de Student que permite, entre otras funciones, inferir acerca de la diferencia entre las medias de dos poblaciones a partir de dos muestras. Sin embargo, muchas veces necesitaremos realizar un procedimiento similar para $k \geq 3$ grupos. En el capítulo 8 nos enfrentamos a un escenario similar para cuando trabajamos con proporciones, para lo cual conocimos la prueba Q de Cochran. Aprendimos que esta última prueba es de tipo ómnibus: es decir, comprueba la igualdad de todos los grupos, pero que si encuentra diferencias no nos indica dónde están. En consecuencia, conocimos los procedimientos post-hoc para identificar entre qué grupos existen estas diferencias.

En el caso de las medias, cuando tenemos más de dos grupos también podemos usar una prueba ómnibus y algunos procedimientos post-hoc, para lo cual nos basaremos en las ideas presentadas por Lowry (1999, caps. 13-14); Glen (2021d); IBM (1989); Meier (2021, p. 4) y Berman (2000).

Intuitivamente, podríamos abordar este problema efectuando pruebas t independientes para cada pareja de grupos con un nivel de significación α . Por ejemplo, para tres muestras A , B y C con medias \bar{x}_A , \bar{x}_B y \bar{x}_C , respectivamente, se tendrían tres pruebas t de Student para diferencia de medias:

1. $\bar{x}_A - \bar{x}_B$
2. $\bar{x}_A - \bar{x}_C$
3. $\bar{x}_B - \bar{x}_C$

Sin embargo, este enfoque presenta un grave inconveniente: por cada una de las pruebas t anteriores, se tiene una probabilidad α de cometer un error tipo I. Al efectuar cada una de las pruebas anteriores, la probabilidad total de que en alguna de ellas se cometiera un error tipo I se acerca a $3 \cdot \alpha$, bastante superior al nivel de significación nominal establecido para la prueba¹. El método de **análisis de varianza**, comúnmente conocido como **ANOVA** o AoV (del inglés Analysis of Variance), surge, en esencia, como un método para combatir este problema al comparar simultáneamente tres o más medias muestrales.

De manera similar, también existe el procedimiento ANOVA para muestras correlacionadas (que se aborda en el capítulo siguiente), semejante a la prueba t de Student con muestras pareadas. Los procedimientos ANOVA para muestras independientes y muestras correlacionadas corresponden al **análisis de varianza de una vía**, pues solo consideran una única variable independiente (de tipo categórica, un **factor**) cuyos niveles definen los grupos que se están comparando.

Existe además el **análisis de varianza de dos vías**, no abordado en el presente texto, el cual permite examinar simultáneamente los efectos de dos variables independientes e, incluso, determinar si ambas interactúan. De hecho, existen métodos para el análisis con más factores, que también están fuera del alcance de este curso.

Para explicar en detalle el procedimiento ANOVA de una vía para muestras independientes, consideremos el siguiente ejemplo: un ingeniero cuenta con tres algoritmos (A, B y C) para resolver un determinado problema (en iguales condiciones y para instancias de tamaño fijo, digamos con E elementos) y desea comparar su eficiencia. Para cada algoritmo, selecciona una muestra aleatoria independiente de instancias y registra el tiempo de ejecución (en milisegundos) para cada una de las instancias de la muestra correspondiente, obteniendo las siguientes observaciones:

- Algoritmo A: 23, 19, 25, 23, 20

¹ Aunque el cálculo exacto de esta probabilidad disjunta escapa a los alcances de este curso, es intuitivo ver que la probabilidad de no cometer un error de tipo I en cada prueba es $(1 - \alpha)$. Así, la probabilidad de no cometer un error de tipo I en las tres comparaciones es $(1 - \alpha)^3$. Luego, la probabilidad de cometer un error de tipo I para la hipótesis global (no hay diferencias entre los grupos) es $1 - (1 - \alpha)^3$. Si, por ejemplo, nominalmente $\alpha = 0,05$, el nivel de significación para la hipótesis global sería aproximadamente 0.143

- Algoritmo B: 26, 24, 28, 23, 29
- Algoritmo C: 19, 24, 20, 21, 17

La pregunta detrás de ANOVA para este ejemplo es: ¿se diferencian los tiempos medios que requieren los algoritmos para resolver todas las posibles instancias del problema de tamaño E ? De donde se desprende que:

H_0 : el tiempo de ejecución promedio para instancias de tamaño E es igual para los tres algoritmos.

H_A : el tiempo de ejecución promedio para instancias de tamaño E es diferente para al menos un algoritmo.

Notemos que, como en el caso de la prueba Q de Cochran, la hipótesis nula no es específica, sino que comprueba la igualdad de todas las medias, por lo que ANOVA es una prueba omnibus.

9.1 CONDICIONES PARA USAR ANOVA DE UNA VÍA PARA MUESTRAS INDEPENDIENTES

Al igual que otras pruebas estudiadas en capítulos anteriores, el procedimiento ANOVA requiere que se cumplan algunas condiciones:

1. La escala con que se mide la variable dependiente tiene las propiedades de una escala de intervalos iguales.
2. Las k muestras son obtenidas de manera aleatoria e independiente desde la(s) población(es) de origen.
3. Se puede suponer razonablemente que la(s) población(es) de origen sigue(n) una distribución normal.
4. Si las muestras provienen de más de una población, estas tienen la misma varianza.

En nuestro ejemplo con los algoritmos, la primera condición se verifica, puesto que si para una instancia i un algoritmo tarda 20 [ms] mientras que otro algoritmo tarda 30 [ms], esa es la misma diferencia (10 milisegundos) que se presenta para una instancia j en que uno tarda 35 [ms] y el otro 45 [ms]. A su vez, el enunciado señala que el proceso seguido por el ingeniero garantiza el cumplimiento de la segunda condición.

La figura 9.1 (creada mediante el script 9.1, líneas 20–29) muestra gráficos Q-Q para cada muestra. Como se observan algunos valores que podrían ser atípicos y las muestras son pequeñas, es mejor que procedamos con cautela y usemos un nivel de significación $\alpha = 0,025$.

Una regla sencilla para comprobar la cuarta condición, llamada también **homogeneidad de las varianzas** u **homocedasticidad**, es comprobar que la razón entre la máxima y la mínima varianza muestral de los grupos no sea superior a 1,5.

$$\begin{aligned}\bar{x}_A &= \frac{23 + 19 + 25 + 23 + 20}{5} = 22,0 \\ s_A^2 &= \frac{(23 - 22)^2 + (19 - 22)^2 + (25 - 22)^2 + (23 - 22)^2 + (20 - 22)^2}{4} = 6,0 \\ \bar{x}_B &= \frac{26 + 24 + 28 + 23 + 29}{5} = 26,0 \\ s_B^2 &= \frac{(26 - 26)^2 + (24 - 26)^2 + (28 - 26)^2 + (23 - 26)^2 + (29 - 26)^2}{4} = 6,5 \\ \bar{x}_C &= \frac{19 + 24 + 20 + 21 + 17}{5} = 20,2 \\ s_C^2 &= \frac{(19 - 20,2)^2 + (24 - 20,2)^2 + (20 - 20,2)^2 + (21 - 20,2)^2 + (17 - 20,2)^2}{4} = 6,7\end{aligned}$$

En el caso del ejemplo, la muestra obtenida para el algoritmo A tiene la menor varianza, mientras que la muestra del algoritmo C tiene la mayor:

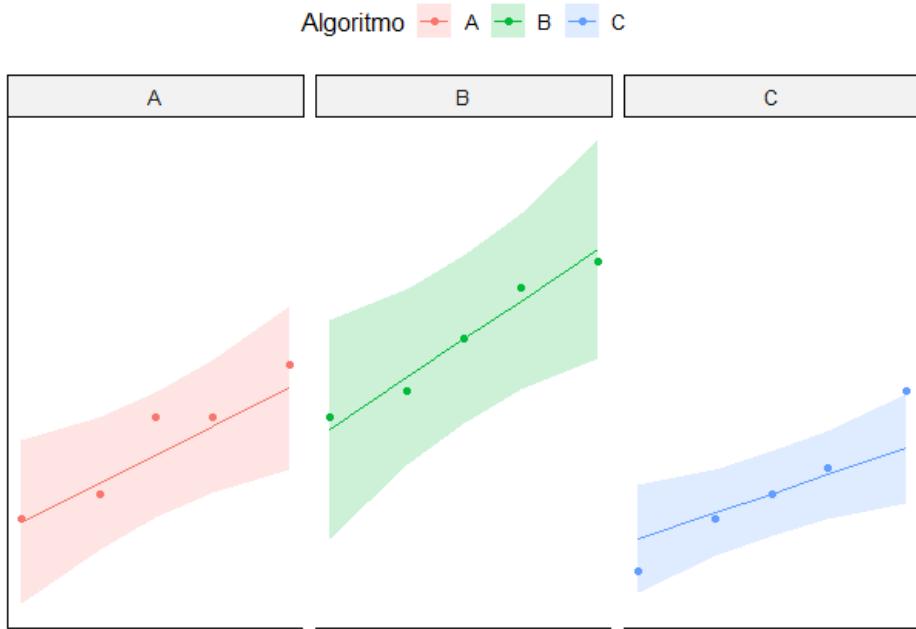


Figura 9.1: gráfico para comprobar el supuesto de normalidad en las tres muestras del ejemplo.

$$\frac{s_C^2}{s_A^2} = \frac{6,7}{6,0} = 1,117 \quad (9.1)$$

En consecuencia, la condición de homocedasticidad se verifica para el ejemplo.

Se ha encontrado que ANOVA es una prueba **robusta**, que resiste razonablemente bien a desviaciones en las condiciones de normalidad o de homocedasticidad, especialmente cuando las muestras tienen el mismo tamaño. Pero estas suposiciones sí están detrás de la lógica y matemática de la prueba, por lo que **no debemos ignorar** violaciones importantes a estas condiciones.

9.2 PROCEDIMIENTO ANOVA DE UNA VÍA PARA MUESTRAS INDEPENDIENTES

Como su nombre indica, ANOVA se centra en la **variabilidad** de las muestras, una generalización de la varianza, que se calcula en base a la suma de los cuadrados de las desviaciones, como muestra la ecuación 9.2, donde n corresponde al tamaño de la muestra, y \bar{x} a la media muestral.

$$SS = \sum_{i=1}^n (x_i - \bar{x})^2 \quad (9.2)$$

9.2.1 Variabilidad total

La variabilidad total, SS_T , se calcula mediante la ecuación 9.2 considerando la totalidad de observaciones, vale decir, combinando las muestras correspondientes a los diferentes grupos.

$$\begin{aligned}\bar{x}_T &= \frac{23 + 19 + 25 + 23 + 20 + 26 + 24 + 28 + 23 + 29 + 19 + 24 + 20 + 21 + 17}{15} \\ &= 22,733\end{aligned}$$

$$\begin{aligned}SS_T &= (23 - 22,733)^2 + (19 - 22,733)^2 + (25 - 22,733)^2 + (23 - 22,733)^2 + (20 - 22,733)^2 + \\ &\quad (26 - 22,733)^2 + (24 - 22,733)^2 + (28 - 22,733)^2 + (23 - 22,733)^2 + (29 - 22,733)^2 + \\ &\quad (19 - 22,733)^2 + (24 - 22,733)^2 + (20 - 22,733)^2 + (21 - 22,733)^2 + (12 - 22,733)^2 \\ &= 164,933\end{aligned}$$

La variabilidad total puede descomponerse en dos partes: una de ellas corresponde a la variabilidad existente al interior de cada uno de los grupos (o variabilidad intra-grupos), *within groups* en inglés, denotada por SS_{wg} ; la otra corresponde a la variabilidad entre los diferentes grupos, *between groups* en inglés, denotada como SS_{bg} . La ecuación 9.3 muestra una **identidad importante** que relaciona ambas componentes.

$$SS_T = SS_{bg} + SS_{wg} \quad (9.3)$$

9.2.2 Variabilidad entre grupos

La **variabilidad entre grupos** nos permite medir de manera agregada la magnitud de las diferencias entre las distintas medias muestrales. Se calcula como muestra la ecuación 9.4, donde:

- k : cantidad de grupos.
- n_i : cantidad de observaciones en el i -ésimo grupo.
- \bar{x}_i : media del i -ésimo grupo.
- \bar{x}_T : media de la muestra combinada.

$$SS_{bg} = \sum_{i=1}^k n_i(\bar{x}_i - \bar{x}_T)^2 \quad (9.4)$$

Esta medida corresponde a la suma de las desviaciones cuadradas de la media de cada grupo con respecto a la media combinada, donde la diferencia de cada grupo se pondera por la cantidad de observaciones que este contiene, a fin de mantener la representatividad de cada grupo. Mide el grado en que los grupos difieren unos de otros. Para el ejemplo tenemos:

$$SS_{bg} = 5(22,0 - 22,733)^2 + 5(26,0 - 22,733)^2 + 5(20,2 - 22,733)^2 = 88,133$$

9.2.3 Variabilidad al interior de cada grupo

La **variabilidad intra-grupos**, a su vez, corresponde a la suma total de las desviaciones cuadradas al interior de cada grupo, por lo que representa la variabilidad aleatoria de cada uno de los diferentes grupos. Esta medida se calcula de acuerdo a la ecuación 9.5, donde SS_i corresponde a la variabilidad del i -ésimo grupo, calculada mediante la ecuación 9.2.

$$SS_{wg} = \sum_{i=1}^k SS_i \quad (9.5)$$

Para el ejemplo:

$$SS_A = (23 - 22)^2 + (19 - 22)^2 + (25 - 22)^2 + (23 - 22)^2 + (20 - 22)^2 = 24,0$$

$$SS_B = (26 - 26)^2 + (24 - 26)^2 + (28 - 26)^2 + (23 - 26)^2 + (29 - 26)^2 = 26,0$$

$$SS_C = (19 - 20,2)^2 + (24 - 20,2)^2 + (20 - 20,2)^2 + (21 - 20,2)^2 + (17 - 20,2)^2 = 26,8$$

$$SS_{wg} = SS_A + SS_B + SS_C = 24,0 + 26,0 + 26,8 = 76,8$$

9.2.4 El estadístico de prueba F

En el capítulo 2 vimos que la varianza se calcula como:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Podemos generalizar esta ecuación como muestra la ecuación 9.6, donde ν corresponde a los grados de libertad.

$$s^2 = \frac{1}{\nu} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (9.6)$$

En el contexto de análisis de varianza, llameremos MS , del inglés *mean square*, a la media de las desviaciones cuadradas. Para el caso de la variabilidad entre grupos, se tienen $\nu_{bg} = k - 1$ grados de libertad, donde k corresponde a la cantidad de grupos (para el ejemplo con los tres algoritmos, $\nu_{bg} = 3 - 1 = 2$). Con ello, la media cuadrada entre grupos queda dada por la ecuación 9.7.

$$MS_{bg} = \frac{SS_{bg}}{\nu_{bg}} \quad (9.7)$$

Entonces, en nuestro ejemplo:

$$MS_{bg} = \frac{88,133}{2} = 44,067$$

De manera similar, los grados de libertad para la componente de la variabilidad al interior de los grupos está dada por la suma de los grados de libertad en cada grupo, como se ve en la ecuación 9.8 (siendo k la cantidad

de grupos), y su media de las desviaciones cuadradas se normaliza con estos grados de libertad (ecuación 9.9).

$$\nu_{wg} = \sum_{i=1}^k (n_k - 1) \quad (9.8)$$

$$MS_{wg} = \frac{SS_{wg}}{\nu_{wg}} \quad (9.9)$$

Así, para el ejemplo tenemos:

$$\begin{aligned}\nu_{wg} &= (5 - 1) + (5 - 1) + (5 - 1) = 12 \\ MS_{wg} &= \frac{76,8}{12} = 6,4\end{aligned}$$

En ocasiones resulta útil conocer también la cantidad total de grados de libertad, que podemos obtener mediante la ecuación 9.10, donde n_T es el tamaño de la muestra combinada.

$$\nu_T = n_T - 1 = \nu_{bg} + \nu_{wg} \quad (9.10)$$

Si bien la relación entre MS_{bg} y MS_{wg} es compleja, en general se cumple que:

- Si la hipótesis nula es verdadera, MS_{bg} tiende a ser menor o igual que MS_{wg} .
- Si la hipótesis nula es falsa, MS_{bg} tiende a ser mayor que MS_{wg} .

Representamos esta relación mediante la razón F (el estadístico de prueba para ANOVA), que se calcula como muestra la ecuación 9.11, donde MS_{efecto} corresponde a una estimación de la varianza del efecto que se desea medir y MS_{error} , a la variabilidad aleatoria pura asociada a la situación. En este punto puede ser útil revisar nuevamente lo que ya hemos aprendido de la distribución F (capítulo 3).

$$F = \frac{MS_{efecto}}{MS_{error}} \quad (9.11)$$

En el ejemplo queremos estudiar si existe diferencia entre las medias de los grupos, por lo que $MS_{efecto} = MS_{bg}$. Asimismo, la variabilidad aleatoria está dada por la variabilidad al interior de los grupos, por lo que $MS_{error} = MS_{wg}$. Así:

$$F = \frac{44,067}{6,4} = 6,885$$

De manera similar a lo que hemos visto en otras pruebas, el p-valor corresponde al área bajo la cola superior de la distribución F (en este caso con 2 y 12 grados de libertad) mayor o igual al estadístico obtenido, que en R puede calcularse mediante la llamada `pf(6,885, 2, 12, lower.tail = FALSE)`, obteniéndose $p = 0,010$.

9.2.5 Resultado del procedimiento ANOVA

El resultado del procedimiento ANOVA suele representarse en forma tabular, como muestra la tabla 9.1.

Fuente	ν	SS	MS	F	p
Entre grupos (efecto)	2	88,133	44,067	6,885	0,010
Intra-grupos (error)	12	76,800	6,400		
TOTAL	14	164,933			

Tabla 9.1: resultado del procedimiento ANOVA.

Como es usual, la conclusión de esta prueba se efectúa comparando el valor p con el nivel de significación. En el ejemplo, $\alpha = 0,025$ y $p < \alpha$, por lo que rechazamos la hipótesis nula en favor de la hipótesis alternativa. En consecuencia, podemos concluir con 97,5 % de confianza que el tiempo de ejecución promedio es diferente para al menos uno de los algoritmos comparados.

Una observación importante que debemos tener en cuenta es que, si usamos ANOVA para casos con **solo dos grupos** (en su correspondientes versiones pareada o independiente), los resultados son equivalentes a los que obtendríamos con una **prueba t de Student**, y el estadístico F sería igual al cuadrado del estadístico t. No obstante, la prueba t puede ser unilateral o bilateral, mientras que el análisis de varianza es intrínsecamente unidireccional, pues la distribución F solo está definida para valores no negativos.

9.2.6 Resumen del procedimiento ANOVA de una vía para muestras independientes

El procedimiento ANOVA de una vía para variables independientes puede resumirse en los siguientes pasos:

1. Calcular la suma de los cuadrados de las desviaciones para la muestra combinada (SS_T).
2. Para cada grupo g , calcular la suma de los cuadrados de las desviaciones dentro de dicho grupo (SS_g).
3. Calcular la variabilidad entre grupos (SS_{bg}).
4. Calcular la variabilidad al interior de los grupos (SS_{wg}).
5. Calcular los grados de libertad (ν_T , ν_{bg} y ν_{wg}).
6. Calcular las medias de las desviaciones cuadradas (MS_{bg} y MS_{wg}).
7. Calcular el estadístico de prueba (F).
8. Obtener el valor p.

9.3 ANOVA DE UNA VÍA PARA MUESTRAS INDEPENDIENTES EN R

Desde luego, R nos ofrece funciones para realizar diferentes pruebas ANOVA, incluyendo la de una vía para muestras independientes.

La primera alternativa que conoceremos es la función `aov(formula, data)`, donde:

- **formula:** se escribe de la forma `variable_dependiente ~ variable_independiente`.
- **data:** data frame que contiene las variables especificadas en la fórmula.

Otra opción es usar la función `ezANOVA(data, dv, wid, between, return_aov)` del paquete `ez`, donde:

- **data:** data frame con los datos.
- **dv:** variable dependiente (numérica con escala de igual intervalo).
- **wid:** variable (factor) con el identificador de cada instancia.
- **between:** variable independiente (factor).

- `return_aov`: si es verdadero, devuelve un objeto de tipo `aov` para uso posterior.

Un parámetro adicional que no hemos mencionado es `type`, el cual no estudiaremos en detalle porque escapa a los alcances de este libro. Sin embargo, se debe tener en cuenta que este argumento permite incorporar algunas modificaciones y resguardos en caso que las muestras tengan diferentes tamaños o se tengan datos incompletos. Al trabajar con R, para la mayoría de los casos se recomienda mantener el valor por defecto (`type = 2`).

Una ventaja de `ezANOVA()` por sobre `aov()` es que, además de ejecutar la prueba ANOVA, realiza también la **prueba de homocedasticidad de Levene** (NIST/SEMATECH, 2013). Si bien no estudiaremos esta prueba en detalle, es pertinente mencionar que sirve para comprobar si k muestras tienen igual varianza, por lo que su resultado nos ayuda a verificar las condiciones requeridas para poder aplicar el procedimiento ANOVA de una vía para muestras independientes. Las hipótesis detrás de esta prueba son:

H_0 : las varianzas de las k poblaciones desde donde se obtuvieron las muestras son iguales.

H_A : al menos una de las poblaciones de origen tiene una varianza diferente a alguna de las otras poblaciones.

El paquete `ez` contiene también la función `ezPlot(data, dv, wid, between, x)`, la cual nos permite ver gráficamente el tamaño del efecto medido. En general, los argumentos son los mismos que para `ezANOVA()`, con la salvedad del nuevo argumento `x`, que señala la variable que va en el eje horizontal del gráfico.

El script 9.1 muestra el procedimiento ANOVA de una vía para muestras independientes usando ambas funciones, con igual resultado al obtenido de manera manual. Además, genera el gráfico del tamaño del efecto medido, presentado en la figura 9.2.

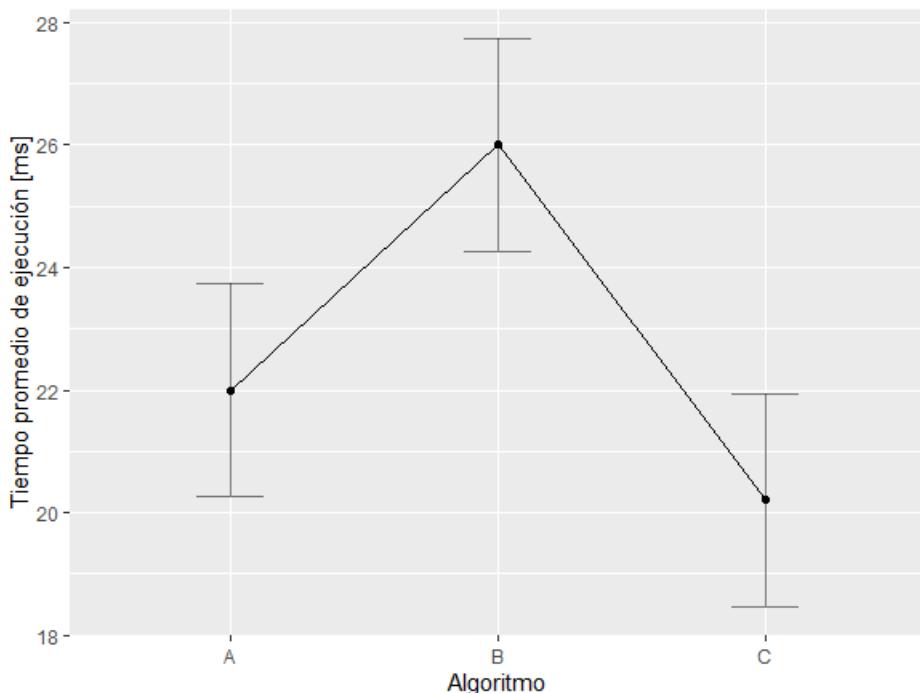


Figura 9.2: tamaño del efecto medido.

Script 9.1: procedimiento ANOVA de una vía para muestras independientes.

```

1 library(tidyverse)
2 library(ggpubr)
3 library(ez)
4
5 # Crear el data frame en formato ancho.
6 A <- c(23, 19, 25, 23, 20)

```

```

7 B <- c(26, 24, 28, 23, 29)
8 C <- c(19, 24, 20, 21, 17)
9 datos <- data.frame(A, B, C)
10
11 # Llevar data frame a formato largo.
12 datos <- datos %>% pivot_longer(c("A", "B", "C"),
13                                     names_to = "algoritmo",
14                                     values_to = "tiempo")
15
16 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
17 datos[["instancia"]] <- factor(1:nrow(datos))
18
19 # Comprobación de normalidad.
20 g <- ggqqplot(datos,
21                 x = "tiempo",
22                 y = "algoritmo",
23                 color = "algoritmo")
24
25 g <- g + facet_wrap(~ algoritmo)
26 g <- g + rremove("x.ticks") + rremove("x.text")
27 g <- g + rremove("y.ticks") + rremove("y.text")
28 g <- g + rremove("axis.title")
29 print(g)
30
31 # Procedimiento ANOVA con aov().
32 cat("Procedimiento ANOVA usando aov\n\n")
33 prueba <- aov(tiempo ~ algoritmo, data = datos)
34 print(summary(prueba))
35
36 # Procedimiento ANOVA con ezANOVA().
37 cat("\n\nProcedimiento ANOVA usando ezANOVA\n\n")
38 prueba2 <- ezANOVA(
39   data = datos,
40   dv = tiempo,
41   between = algoritmo,
42   wid = instancia,
43   return_aov = TRUE)
44
45 print(prueba2)
46
47 # Gráfico del tamaño del efecto.
48 g2 <- ezPlot(
49   data = datos,
50   dv = tiempo,
51   wid = instancia,
52   between = algoritmo,
53   y_lab = "Tiempo promedio de ejecución [ms]",
54   x = algoritmo
55 )
56
57 print(g2)

```

9.4 ANÁLISIS POST-HOC

Al aplicar el procedimiento ANOVA de una vía para muestras independientes a nuestro ejemplo pudimos concluir que **existe al menos un algoritmo cuyo tiempo promedio de ejecución es diferente al de los demás**. Ahora bien, si los algoritmos del ejemplo tienen por objeto resolver un problema crítico, de cuya rápida solución depende aumentar la productividad de una empresa o prevenir una situación de mucho riesgo, desde luego nos interesaría conocer cuál es el mejor (o el peor) de los algoritmos comparados a fin de poder garantizar un menor tiempo de respuesta. En consecuencia, necesitamos contar con algún método que permita determinar si los tiempos de ejecución de los algoritmos A y B difieren significativamente, o los de B y C, o bien los de A y C. En el contexto general, si tenemos k grupos, la cantidad de comparaciones (N) que deberíamos efectuar está dada por la ecuación 9.12.

$$N = \binom{k}{2} = \frac{k(k-1)}{2} \quad (9.12)$$

Al igual que aprendimos en el capítulo anterior para la prueba Q de Cochran, existen diversas pruebas *post-hoc* que podemos usar para este fin, algunas de las cuales exploraremos a continuación.

9.4.1 Correcciones de Bonferroni y Holm

Como ya estudiamos en el capítulo anterior, los factores de corrección de Bonferroni y Holm distribuyen el nivel de significación cuando se realizan múltiples comparaciones entre pares de grupos. Las fórmulas para calcularlos son las mismas que ya conocemos, pero ahora se realizan pruebas t para muestras independientes para cada par. R dispone de la función `pairwise.t.test(x, g, p.adjust.method, pool.sd, paired, alternative, ...)`, donde:

- `x`: vector con la variable dependiente.
- `g`: factor o vector de agrupamiento.
- `p.adjust.method`: señala qué método emplear para ajustar los valores p resultantes.
- `pool.sd`: valor booleano que indica si se usa o no varianza combinada.
- `paired`: valor booleano que indica si las pruebas t son pareadas (verdadero) o no.
- `alternative`: indica si la prueba es bilateral ("two.sided") o unilateral ("greater" o "less").
- `...`: argumentos adicionales que se pasan a la función `t.test()` que es llamada internamente

El script 9.2 muestra la realización de pruebas t para cada par de grupos usando tanto la corrección de Bonferroni como la de Holm, obteniéndose los resultados que se muestran en la figura 9.3. Debemos recordar que el nivel de significación que se entrega como argumento es el mismo que usamos en el procedimiento ANOVA.

Script 9.2: procedimientos *post-hoc* de Bonferroni y Holm en R.

```
1 library(tidyverse)
2
3 # Crear el data frame en formato ancho.
4 A <- c(23, 19, 25, 23, 20)
5 B <- c(26, 24, 28, 23, 29)
6 C <- c(19, 24, 20, 21, 17)
7 datos <- data.frame(A, B, C)
8
9 # Llevar data frame a formato largo.
10 datos <- datos %>% pivot_longer(c("A", "B", "C")),
```

```

11             names_to = "algoritmo",
12             values_to = "tiempo")
13
14 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
15 datos[["instancia"]] <- factor(1:nrow(datos))
16
17 # Establecer nivel de significación (el mismo usado en ANOVA).
18 alfa <- 0.025
19
20 # Procedimiento post-hoc de Bonferroni.
21 cat("Procedimiento post-hoc de Bonferroni\n\n")
22
23 bonferroni <- pairwise.t.test(datos[["tiempo"]],
24                                 datos[["algoritmo"]],
25                                 p.adj = "bonferroni",
26                                 pool.sd = TRUE,
27                                 paired = FALSE,
28                                 conf.level = 1 - alfa)
29
30 print(bonferroni)
31
32 # Procedimiento post-hoc de Holm.
33 cat("\n\nProcedimiento post-hoc de Holm\n\n")
34
35 holm <- pairwise.t.test(datos[["tiempo"]],
36                           datos[["algoritmo"]],
37                           p.adj = "holm",
38                           pool.sd = TRUE,
39                           paired = FALSE,
40                           conf.level = 1 - alfa)
41
42 print(holm)

```

Los valores p obtenidos con ambos métodos son diferentes (un lector atento recordará que la corrección de Bonferroni es considerada muy conservadora). Sin embargo, en ambos casos podemos ver que únicamente los algoritmos B y C presentan una diferencia significativa al comparar el valor p ajustado que entrega R con el nivel de significación ($\alpha = 0,025$). Si miramos el gráfico del tamaño del efecto obtenido para el procedimiento ANOVA (figura 9.2), podemos concluir entonces con 97,5% de confianza que el algoritmo C es más rápido que el algoritmo B.

9.4.2 Prueba HSD de Tukey

La prueba **HSD de Tukey** es más poderosa que los factores de corrección de Bonferroni y Holm. Se asemeja a estas últimas en que también busca diferencias significativas (de hecho, el nombre HSD se debe a las siglas inglesas para “diferencia honestamente significativa”) entre los diferentes pares de medias, aunque usa un enfoque muy diferente: para ello emplea el estadístico Q , el cual sigue una distribución de rango estudiantizado², que para cualquier par de medias en los k grupos se calcula según la ecuación 9.13, donde:

- \bar{x}_g es la mayor de las dos medias comparadas.
- \bar{x}_p es la menor de las dos medias comparadas.
- MS_{wg} corresponde la media cuadrada intra-grupos (entregada por el procedimiento ANOVA).

²El detalle de la distribución de rango estudiantizado escapa a los alcances de este texto.

Procedimiento post-hoc de Bonferroni

```

Pairwise comparisons using t tests with pooled SD

data:  datos[["tiempo"]] and datos[["algoritmo"]]

      A      B
B 0.084 -
C 0.848 0.010

P value adjustment method: bonferroni

```

Procedimiento post-hoc de Holm

```

Pairwise comparisons using t tests with pooled SD

data:  datos[["tiempo"]] and datos[["algoritmo"]]

      A      B
B 0.056 -
C 0.283 0.010

P value adjustment method: holm

```

Figura 9.3: valores p obtenidos en las pruebas t para cada par de grupos mediante los métodos de Bonferroni y Holm.

- n_m es la cantidad de observaciones por cada muestra. Si las k muestras tienen tamaños diferentes, se calcula mediante la fórmula presentada en la ecuación 9.14.

$$Q = \frac{\bar{x}_g - \bar{x}_p}{\sqrt{\frac{MS_{wg}}{n_m}}} \quad (9.13)$$

$$n_m = \frac{k}{\sum_{i=1}^k \frac{1}{n_i}} \quad (9.14)$$

En la práctica, sin embargo, no necesitamos calcular el estadístico Q para cada par de medias, sino que basta con conocer el valor crítico de este estadístico para el nivel de significación α establecido (denotado por Q_α), el cual depende de la cantidad de grupos (k) y de los grados de libertad del error, ν_{wg} , en el caso de ANOVA de una vía para muestras independientes. La llamada `qtukey(α , n_m , ν_{wg} , lower.tail = FALSE)` en R entrega el valor de Q_α .

El valor crítico Q_α nos permite determinar cuán grande debe ser la diferencia entre las medias de dos grupos para ser considerada significativa, lo cual se logra mediante la ecuación 9.15.

$$HSD_\alpha = Q_\alpha \cdot \sqrt{\frac{MS_{wg}}{n_m}} \quad (9.15)$$

Así, una diferencia entre las medias de dos grupos únicamente es significativa si es mayor o igual que HSD_α .

Para el ejemplo, se tenemos que $Q_\alpha = 4,324$, de donde:

$$HSD_{0,025} = 4,324 \cdot \sqrt{\frac{6,4}{5}} = 4,892$$

Recordando del procedimiento ANOVA que $\bar{x}_A = 22$, $\bar{x}_B = 26$ y $\bar{x}_C = 20,2$; tenemos:

$$\bar{x}_B - \bar{x}_A = 26 - 22 = 4$$

$$\bar{x}_A - \bar{x}_C = 22 - 20,2 = 1,8$$

$$\bar{x}_B - \bar{x}_C = 26 - 20,2 = 5,8$$

Así, la tercera diferencia es la única que supera $HSD_{0,025}$, con lo que solo existe diferencia significativa entre los tiempos promedio de ejecución de los algoritmos B y C, y se puede concluir que el algoritmo C es más rápido que el algoritmo B, lo que se condice con los resultados presentados en la figura 9.2.

R también permite realizar la prueba HSD de Tukey, como muestra el script 9.3. La función para ello es `TukeyHSD(x, which, ordered, conf.level)`, donde:

- `x`: un modelo ANOVA (objeto de tipo `aov`).
- `which`: string con el nombre de la variable para la que se calculan las diferencias.
- `ordered`: valor lógico que, cuando es verdadero, hace que los grupos se ordenen de acuerdo a sus medias a fin de obtener diferencias positivas.
- `conf.level`: nivel de confianza.

La figura 9.4 muestra el resultado obtenido para la prueba HSD de Tukey mediante el script 9.3.

Script 9.3: procedimiento *post-hoc* de Tukey.

```

1 library(tidyverse)
2
3 # Crear el data frame en formato ancho.
4 A <- c(23, 19, 25, 23, 20)
5 B <- c(26, 24, 28, 23, 29)
6 C <- c(19, 24, 20, 21, 17)
7 datos <- data.frame(A, B, C)
8
9 # Llevar data frame a formato largo.
10 datos <- datos %>% pivot_longer(c("A", "B", "C"),
11                                     names_to = "algoritmo",
12                                     values_to = "tiempo")
13
14 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
15 datos[["instancia"]] <- factor(1:nrow(datos))
16
17 # Establecer nivel de significación (el mismo usado en ANOVA).
18 alfa <- 0.025
19
20 # Procedimiento ANOVA.
21 anova <- aov(tiempo ~ algoritmo, data = datos)
22
23 # Prueba HSD de Tukey.
24 post_hoc <- TukeyHSD(anova,
25                       "algoritmo",
26                       ordered = TRUE,
```

```

27   conf.level = 1 - alfa)
28
29 print(post_hoc)

Tukey multiple comparisons of means
 97.5% family-wise confidence level
 factor levels have been ordered

Fit: aov(formula = tiempo ~ algoritmo, data = datos)

$algoritmo
    diff      lwr      upr     p adj
A-C  1.8 -3.0923417  6.692342 0.5176889
B-C  5.8  0.9076583 10.692342 0.0090297
B-A  4.0 -0.8923417  8.892342 0.0670199

```

Figura 9.4: resultado del procedimiento *post-hoc* HSD de Tukey.

En la figura 9.4 podemos apreciar que la columna **diff** muestra las diferencias de las medias entre grupos, obteniéndose resultados idénticos a los teóricos, y la columna **p.adj** entrega valores *p* asociados a cada diferencia, **ajustados** para compararlos con el nivel de significación original. Cabe destacar que el único valor *p* menor a este nivel ($\alpha = 0,025$) corresponde a la diferencia B-C, siendo esta última la única significativa, lo cual una vez más coincide con el resultado del procedimiento manual. También debemos notar que las columnas **lwr** y **upr** muestran el límite inferior y superior, respectivamente, del intervalo de $(1 - \alpha) \cdot 100\%$ confianza para la verdadera diferencia entre las medias de los grupos.

9.4.3 Prueba de comparación de Scheffé

Otra alternativa para hacer un análisis *post-hoc* es la **prueba de Scheffé**. Al igual que la corrección de Bonferroni, este método también es muy conservador al momento de efectuar comparaciones entre pares. No obstante, tiene la ventaja de que permite hacer comparaciones adicionales, además de todos los pares de grupos: por ejemplo, podríamos preguntar si un grupo es mejor que todos los demás. El ingeniero del ejemplo podría, tras encontrar mediante el procedimiento ANOVA que existen diferencias significativas, plantearse preguntas del siguiente tipo:

1. ¿Existe diferencia entre los tiempos de ejecución de los algoritmos A y B?
2. ¿Es el tiempo promedio de ejecución del algoritmo A distinto al tiempo de ejecución promedio de los algoritmos B y C?

La primera pregunta corresponde a una comparación entre pares, pero la segunda resulta más compleja. En realidad, podemos modelar escenarios para múltiples preguntas, usando para ello **contrastes**, que son **combinaciones lineales** de las medias de cada grupo. Para entender mejor esta idea, veamos la primera pregunta. Matemáticamente, puede formularse como las siguientes hipótesis:

$$H_0: \mu_A - \mu_B = 0$$

$$H_A: \mu_A - \mu_B \neq 0$$

La hipótesis nula puede expresarse, entonces, como una combinación lineal de la forma:

$$c_A \cdot \mu_A + c_B \cdot \mu_B + c_C \cdot \mu_C = 0$$

Que puede, a su vez, representarse vectorialmente como:

$$[c_A, c_B, c_C]$$

En este caso, resulta evidente que la combinación lineal es:

$$1 \cdot \mu_A - 1 \cdot \mu_B + 0 \cdot \mu_C = 0$$

Que corresponde al vector:

$$[1, -1, 0]$$

La segunda pregunta es algo más compleja, pero las hipótesis asociadas son:

$$\begin{aligned} H_0: \mu_A - \frac{\mu_B + \mu_C}{2} &= 0 \\ H_A: \mu_A - \frac{\mu_B + \mu_C}{2} &\neq 0 \end{aligned}$$

Vectorialmente dada por:

$$\left[1, -\frac{1}{2}, -\frac{1}{2} \right]$$

Ahora que hemos establecido qué es un contraste, podemos comenzar a explicar el procedimiento *post-hoc* de Scheffé, el cual ocupa el mismo nivel de significación empleado para el procedimiento ANOVA. Recordemos que, para el ejemplo, $\alpha = 0,025$, $\bar{x}_A = 22$, $\bar{x}_B = 26$ y $\bar{x}_C = 20,2$.

El primer paso consiste en determinar los contrastes a realizar. Supongamos que el ingeniero desea hacer todas las comparaciones entre pares y, además, comparar cada algoritmo contra los dos restantes. Podemos representar esto en forma matricial, donde cada fila de la matriz corresponde a un contraste:

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & -0,5 & -0,5 \\ -0,5 & 1 & -0,5 \\ -0,5 & -0,5 & 1 \end{bmatrix}$$

Luego calculamos los estimadores para cada contraste C_i :

$$\begin{aligned} C_1 &= |\bar{x}_A - \bar{x}_B| = 4,0 \\ C_2 &= |\bar{x}_A - \bar{x}_C| = 1,8 \\ C_3 &= |\bar{x}_B - \bar{x}_C| = 5,8 \\ C_4 &= |\bar{x}_A - \frac{\bar{x}_B + \bar{x}_C}{2}| = 1,1 \\ C_5 &= |-\frac{\bar{x}_A}{2} + \bar{x}_B - \frac{\bar{x}_C}{2}| = 4,9 \\ C_6 &= |-\frac{\bar{x}_A}{2} - \bar{x}_B + \bar{x}_C| = 3,8 \end{aligned}$$

El tercer paso consiste en calcular los valores críticos para la prueba de comparación de Scheffé, dados por la ecuación 9.16, donde:

- i es el número de fila del contraste.
- ν_{efecto} , ν_{error} y MS_{error} se obtienen desde la tabla ANOVA (tabla 9.1).
- F^* corresponde al percentil $1 - \alpha$ de la distribución $F(\nu_{\text{efecto}}, \nu_{\text{error}})$.
- c_j es el peso del grupo j en la comparación i .
- n_j es el tamaño de la muestra para el grupo j .

$$VC_i = \sqrt{\nu_{\text{efecto}} \cdot F^* \cdot MS_{\text{error}} \cdot \sum_{j=1}^k \frac{c_j^2}{n_j}} \quad (9.16)$$

Así, para el ejemplo tenemos que $\nu_{\text{efecto}} = 2$, $\nu_{\text{error}} = 12$ y $MS_{\text{error}} = 6,4$. Podemos calcular F^* en R con la llamada `qf(1 - 0.025, 2, 12, lower.tail = TRUE)`, obteniéndose $F^* = 5,0959$. Debemos notar que en la ecuación 9.16, $\nu_{\text{efecto}} \cdot F^* \cdot MS_{\text{error}} = 2 \cdot 5,0959 \cdot 6,4 = 65,2275$ es constante para todos los contrastes. Así:

$$\begin{aligned} VC_1 &= \sqrt{65,2275 \cdot \left(\frac{1^2}{5} + \frac{(-1)^2}{5} + \frac{0^2}{5} \right)} = 4,9891 \\ VC_2 &= \sqrt{65,2275 \cdot \left(\frac{1^2}{5} + \frac{0^2}{5} + \frac{(-1)^2}{5} \right)} = 4,9891 \\ VC_3 &= \sqrt{65,2275 \cdot \left(\frac{0^2}{5} + \frac{1^2}{5} + \frac{(-1)^2}{5} \right)} = 4,9891 \\ VC_4 &= \sqrt{65,2275 \cdot \left(\frac{1^2}{5} + \frac{(-0,5)^2}{5} + \frac{(-0,5)^2}{5} \right)} = 2,4945 \\ VC_5 &= \sqrt{65,2275 \cdot \left(\frac{(-0,5)^2}{5} + \frac{1^2}{5} + \frac{(-0,5)^2}{5} \right)} = 2,4945 \\ VC_6 &= \sqrt{65,2275 \cdot \left(\frac{(-0,5)^2}{5} + \frac{(-0,5)^2}{5} + \frac{1^2}{5} \right)} = 2,4945 \end{aligned}$$

Tabulemos los resultados obtenidos hasta ahora, como muestra la tabla 9.2.

i	C_i	VC_i
1	4,0	4,9891
2	1,8	4,9891
3	5,8	4,9891
4	1,1	2,4945
5	4,9	2,4945
6	3,8	2,4945

Tabla 9.2: estimadores y valores críticos para los contrastes de la prueba de comparación de Scheffé.

Finalmente evaluamos cada contraste, comparando el estimador C_i con el valor crítico correspondiente, VC_i . Si $C_i > VC_i$, la comparación es estadísticamente significativa. Podemos ver, entonces, que las comparaciones 3, 5 y 6 son significativas. Esto quiere decir que:

- Existe una diferencia significativa entre las eficiencias de los algoritmos B y C.
- El tiempo promedio de ejecución del algoritmo B es distinto del tiempo promedio de ejecución (combinado) de los algoritmos A y C.
- El tiempo promedio de ejecución del algoritmo C es distinto del tiempo promedio de ejecución (combinado) de los algoritmos A y B.

En R, este procedimiento puede hacerse mediante la función `ScheffeTest(x, which, contrasts, conf.level)` del paquete `DescTools`, donde:

- `x`: objeto `aov` con el resultado de ANOVA.
- `which`: variable independiente en la prueba.
- `contrasts`: matriz con los contrastes (cada contraste es una columna).
- `conf.level`: nivel de confianza.

El script 9.4 muestra el ejemplo en R, cuyo resultado se presenta en la figura 9.5. A diferencia del proceso manual, la función `ScheffeTest()` nos entrega un valor `p` ajustado para cada contraste e identifica aquellos que son relevantes para diferentes niveles de significación. Debemos tener en cuenta que el resultado es ligeramente diferente debido a errores de redondeo. Aquí, al igual que en el caso de la prueba HSD de Tukey, las columnas `lwr` y `upr` señalan los límites del intervalo de confianza para la verdadera diferencia entre las medias de los grupos.

```
Posthoc multiple comparisons of means: Scheffe Test
97.5% family-wise confidence level

$algoritmo
      diff     lwr.ci     upr.ci   pval
A-B    -4.0 -9.1079193  1.1079193 0.0808 .
A-C     1.8 -3.3079193  6.9079193 0.5479
B-C     5.8  0.6920807 10.9079193 0.0118 *
A-B,C -1.1 -5.5235879  3.3235879 0.7356
B-A,C  4.9  0.4764121  9.3235879 0.0138 *
C-A,B -3.8 -8.2235879  0.6235879 0.0540 .

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figura 9.5: valores `p` e intervalos de confianza para las diferencias de las medias obtenidos mediante la prueba de comparación de Scheffé.

Script 9.4: prueba de comparación de Scheffé.

```
1 library(tidyverse)
2 library(DescTools)
3
4 # Crear el data frame en formato ancho.
5 A <- c(23, 19, 25, 23, 20)
6 B <- c(26, 24, 28, 23, 29)
7 C <- c(19, 24, 20, 21, 17)
8 datos <- data.frame(A, B, C)
9
10 # Llevar data frame a formato largo.
11 datos <- datos %>% pivot_longer(c("A", "B", "C"),
12                                     names_to = "algoritmo",
13                                     values_to = "tiempo")
14
15 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
16 datos[["instancia"]] <- factor(1:nrow(datos))
17
18 # Establecer nivel de significación (el mismo usado en ANOVA).
19 alfa <- 0.025
20
21 # Procedimiento ANOVA.
22 anova <- aov(tiempo ~ algoritmo, data = datos)
```

```

23
24 # Crear matriz de contrastes.
25 contrastes <- matrix(c(1, -1, 0,
26                         1, 0, -1,
27                         0, 1, -1,
28                         1, -0.5, -0.5,
29                         -0.5, 1, -0.5,
30                         -0.5, -0.5, 1),
31                         nrow=6,
32                         byrow = TRUE
33 )
34
35 # Trasponer matriz de contrastes (para que cada contraste sea una columna).
36 contrastes <- t(contrastes)
37
38 # Hacer prueba de Scheffé.
39 scheffe <- SchefféTest(x = anova,
40                         which = "algoritmo",
41                         contrasts = contrastes,
42                         conf.level = 1 - alfa
43 )
44
45 print(scheffe)

```

Un detalle importante a tener en cuenta es que podemos hacer la llamada a `SchefféTest()` sin entregar los argumentos `which` y `contrasts`, en cuyo caso únicamente se contrastan todos los pares, como en las pruebas *post-hoc* precedentes.

9.5 EJERCICIOS PROPUESTOS

1. Si se tienen datos de tres grupos (A, B y C), ¿por qué no se pueden hacer tres comparaciones con pares de grupos con la prueba t de Student (A-B, A-C, B-C) vista en el capítulo 5?
2. Si SS es la suma de desviaciones cuadradas, ¿qué son SS entre grupos, SS al interior de los grupos y SS total?
3. Define la razón F. ¿Por qué se espera que sea cercana a 1 si es que las poblaciones tienen medias similares?
4. ¿Qué significa que un procedimiento ANOVA sea de una vía?
5. ¿Cuándo un procedimiento ANOVA de una vía es equivalente a una prueba T de Student?
6. ¿Qué sería ANOVA de dos vías?
7. ¿Cuándo aplica el procedimiento ANOVA de una vía para muestras independientes?
8. ¿Cuáles son las hipótesis contrastadas en el procedimiento ANOVA de una vía para muestras independientes?
9. Enumera las condiciones (o supuestos) del procedimiento ANOVA de una vía para muestras independientes para tener confiabilidad.
10. Investiga con más detalle por qué se dice que un procedimiento ANOVA es una prueba omnibus.
11. Investiga en qué consiste, para qué sirve y cómo se aplica en R la prueba de Levene.
12. Investiga algún procedimiento *post-hoc* no abordado en este capítulo, junto con la forma de aplicarlo en R.
13. El conjunto de datos `chickwts`, disponible en R, registra el peso de 71 pollitos a las seis semanas de nacidos y el tipo de alimento que cada pollito recibió. Para este conjunto de datos:
 - a) Verifica si se cumplen las condiciones para efectuar un procedimiento ANOVA de una vía para

muestras independientes.

- b) Independientemente del resultado anterior, efectúa el procedimiento ANOVA de una vía para muestras independientes a fin de determinar si existen diferencias en el peso de los pollitos de acuerdo al tipo de alimento recibido.
- c) En caso de identificar que existen diferencias significativas, lleva a cabo los análisis post-hoc y determina qué tipos de alimento presentan dichas diferencias. Compara los resultados obtenidos con los diferentes métodos.

CAPÍTULO 10. ANOVA DE UNA VÍA PARA MUESTRAS CORRELACIONADAS

En el capítulo 9 conocimos el procedimiento ANOVA de una vía para muestras independientes, que podemos entender como una extensión de la prueba t de Student para muestras independientes. De manera similar, ahora abordaremos el **procedimiento ANOVA de una vía para muestras correlacionadas** (también llamado **ANOVA para medidas repetidas** o **ANOVA intra-sujetos**) que puede asociarse a la prueba t con muestras pareadas, pero ahora con tres o más mediciones (o condiciones) en lugar de dos. Para ello tomaremos como base la explicación que ofrece Lowry (1999, cap. 15).

En este caso podemos distinguir entre dos escenarios:

- **Diseño con medidas repetidas:** a cada sujeto se le toman medidas en las diferentes condiciones, por ejemplo, registrar los tiempos de ejecución para una misma instancia de un problema con k algoritmos diferentes.
- **Diseño con bloques aleatorios:** cada bloque contiene diferentes sujetos agrupados según una determinada característica, por ejemplo, registrar tiempos de ejecución usando instancias de grafos diferentes, pero similares (como que tengan el mismo número de vértices y aristas), para los k algoritmos.

El método es el mismo en ambos casos e intenta controlar estadísticamente la variación introducida por factores distintos al que se desea estudiar, usando para ello varias mediciones de un sujeto (o grupos de sujetos parecidos). Si bien el diseño con bloques aleatorios es común, especialmente en medicina, este apunte usa las medidas repetidas en su discusión, ya que son más comunes en el área de la informática.

Como es habitual, usemos un ejemplo para ver cómo se lleva a cabo el procedimiento ANOVA de una vía para muestras correlacionadas. Supongamos que un estudiante de un curso de programación debe comparar la eficiencia de cuatro algoritmos de ordenamiento: *quicksort*, *bubblesort*, *radixsort* y *mergesort*. Para ello, ha seleccionado aleatoriamente 6 arreglos de igual tamaño y registrado, para cada uno de ellos, el tiempo de ejecución utilizado por cada algoritmo (en milisegundos), como muestra la tabla 10.1¹.

Instancia	Quicksort	Bubblesort	Radixsort	Mergesort
1	23,2	31,6	30,1	25,0
2	22,6	29,3	28,4	25,7
3	23,4	30,7	28,7	25,7
4	23,3	30,8	28,3	23,7
5	21,8	29,8	29,9	25,5
6	23,9	30,3	29,1	24,7

Tabla 10.1: tiempos de ejecución para las diferentes instancias con cada algoritmo del ejemplo.

En este caso, la lógica es muy similar a la que ya conocimos para ANOVA con muestras independientes. Sin embargo, existe una diferencia importante al trabajar con muestras correlacionadas: no toda la variabilidad es pura e inevitable, sino que una parte de ella se debe a diferencias individuales preexistentes entre los sujetos (por ejemplo, un arreglo puede estar ordenado desde el inicio, mientras otro podría estar en orden inverso).

Recordemos que la pregunta detrás de ANOVA es: ¿se diferencian las medias poblacionales?, por lo que nuestras hipótesis son:

H_0 : El tiempo de ejecución promedio es igual para los cuatro algoritmos.

H_A : El tiempo de ejecución promedio es diferente para al menos un algoritmo.

¹ Los valores aquí expuestos son ficticios.

10.1 CONDICIONES PARA USAR ANOVA DE UNA VÍA PARA MUESTRAS CORRELACIONADAS

Al igual que otras pruebas que hemos conocido en capítulos anteriores, este procedimiento requiere que se cumplan algunas condiciones:

1. La escala con que se mide la variable dependiente tiene las propiedades de una escala de intervalos iguales.
2. Las mediciones son independientes al interior de cada grupo.
3. Se puede suponer razonablemente que la(s) población(es) de origen sigue(n) una distribución normal.
4. La matriz de varianzas-covarianzas es esférica. Como explica Horn (2008, p. 1), esta condición establece que las varianzas entre los diferentes niveles de las medidas repetidas deben ser iguales.

Veamos si nuestro ejemplo cumple con las condiciones. La primera se verifica, puesto que el tiempo, como toda magnitud física, tiene una escala de intervalos iguales (de hecho tiene escala de razón). A su vez, el enunciado señala que el proceso seguido por el ingeniero garantiza el cumplimiento de la segunda condición.

La figura 10.1 (creada mediante el script 10.1, líneas 20–26) muestra gráficos Q-Q para cada grupo, donde se puede apreciar que no se observan valores que pudieran ser considerados atípicos y se puede suponer razonablemente que las distribuciones se asemejan a la normal.

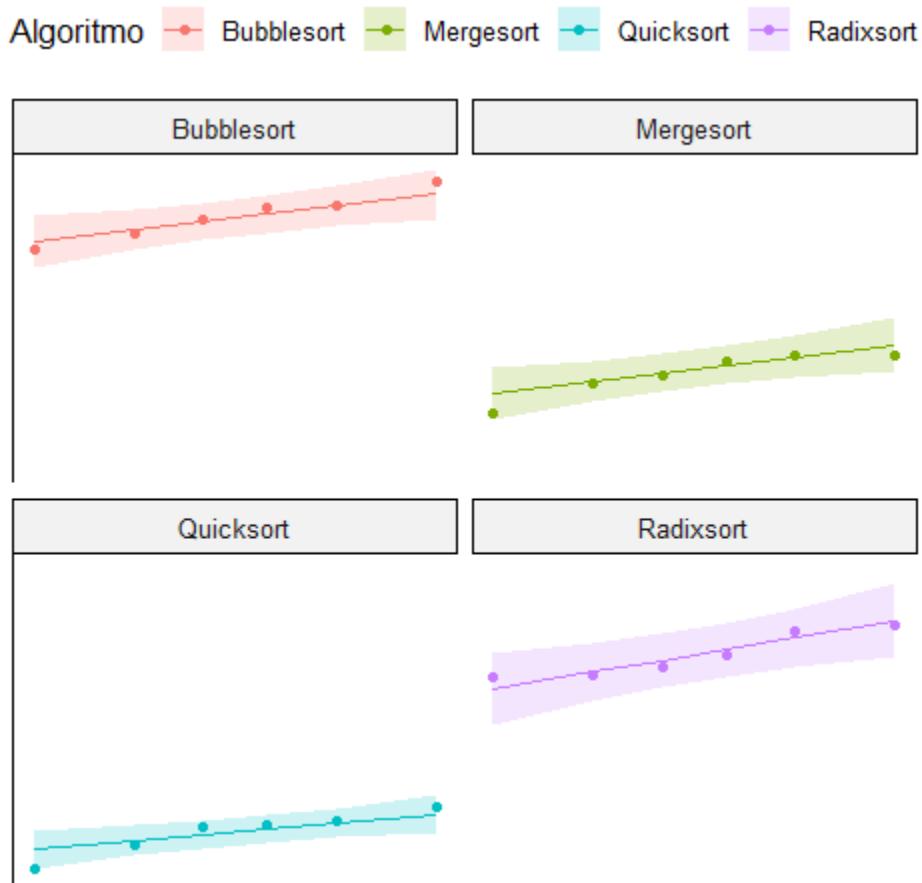


Figura 10.1: gráfico para comprobar el supuesto de normalidad en las muestras del ejemplo.

La prueba de esfericidad es más compleja, por lo que no se aborda en este texto. En principio, podemos examinar las diferencias entre las varianzas registradas para cada algoritmo (que se muestran en la tabla

10.2). Podemos ver que las diferencias parecen más bien “pequeñas” si se considera que los tiempos promedio están el rango de 23 a 30 [ms], por lo que podríamos asumir que son iguales. No obstante, la función de `R ezANOVA()` incluye una prueba para verificar esta condición: la **prueba de esfericidad de Mauchly**, e incluso proporciona un método para controlar posibles violaciones, como veremos más adelante en este capítulo.

	Mergesort	Quicksort	Radixsort
Bubblesort	0.05	0.12	0.07
Mergesort		0.06	0.01
Quicksort			-0.05

Tabla 10.2: diferencias de las varianza del tiempo de ejecución entre cada par de algoritmos.

10.2 PROCEDIMIENTO ANOVA DE UNA VÍA PARA MUESTRAS CORRELACIONADAS

Al igual que para el caso de muestras independientes, el procedimiento ANOVA para muestras correlacionadas opera en base a la variabilidad, calculada en base a la suma de los cuadrados de las desviaciones. Recordemos la forma general de este cálculo:

$$SS = \sum_{i=1}^n (x_i - \bar{x})^2$$

10.2.1 Variabilidad total, entre grupos e intra-grupos

Los primeros pasos son idénticos a los que ya estudiamos para ANOVA de una vía con muestras independientes, y consisten en calcular la variabilidad total (es decir, para las muestras combinadas), la variabilidad entre grupos y la variabilidad intra-grupos, denotadas por SS_T , SS_{bg} y SS_{wg} , respectivamente.

$$\begin{aligned} SS_T &= 224,930 \\ SS_{bg} &= 213,045 \\ SS_{wg} &= 11,885 \end{aligned}$$

10.2.2 Variabilidad entre sujetos

Como mencionamos en la introducción de este capítulo, al trabajar con muestras correlacionadas es necesario descartar la variabilidad debida a las diferencias preexistentes entre los diferentes sujetos ($SS_{sujetos}$), pues

estas son ajenas al factor en estudio, y solo nos interesa conservar la variabilidad pura (SS_{error}). Así, en ANOVA para muestras correlacionadas aparece una nueva identidad, dada por la ecuación 10.1.

$$SS_{wg} = SS_{sujetos} + SS_{error} \quad (10.1)$$

De manera similar a la que ya empleamos en cálculos previos, la variabilidad entre sujetos está dada por la ecuación 10.2, donde:

- k corresponde a la cantidad de observaciones (medidas) por cada sujeto.
- \bar{x}_i es la media de las observaciones del i -ésimo sujeto.
- \bar{x}_T es la media combinada de las mediciones.

$$SS_{sujetos} = k \cdot \sum_{i=1}^n (\bar{x}_i - \bar{x}_T)^2 \quad (10.2)$$

La tabla 10.3 muestra una vez más las observaciones del ejemplo, incluyendo el tiempo promedio de ejecución para cada instancia.

Instancia	Quicksort	Bubblesort	Radixsort	Mergesort	\bar{x}
1	23,2	31,6	30,1	25,0	27,475
2	22,6	29,3	28,4	25,7	26,500
3	23,4	30,7	28,7	25,7	27,125
4	23,3	30,8	28,3	23,7	26,525
5	21,8	29,8	29,9	25,5	26,750
6	23,9	30,3	29,1	24,7	27,000

Tabla 10.3: tiempos de ejecución y tiempo medio de ejecución para las diferentes instancias del ejemplo.

$$SS_{sujetos} = 4 \cdot [(27,475 - 26,896)^2 + (26,500 - 26,896)^2 + (27,125 - 26,896)^2 + (26,525 - 26,896)^2 + (26,750 - 26,896)^2 + (27,000 - 26,896)^2] = 2,857$$

$$SS_{error} = SS_{wg} - SS_{sujetos} = 11,885 - 2,857 = 9,028$$

10.2.3 El estadístico de prueba F

Al igual que en el capítulo 9, calculamos ahora los grados de libertad ya conocidos (recordemos que ahora k corresponde a la cantidad de mediciones por sujeto):

$$\begin{aligned} \nu_T &= n_T - 1 = 24 - 1 = 23 \\ \nu_{bg} &= k - 1 = 4 - 1 = 3 \\ \nu_{wg} &= n_T - k = 24 - 4 = 20 \end{aligned}$$

Puesto que anteriormente descompusimos la variabilidad intra-grupos en variabilidad intra-sujetos y variabilidad del error, necesitamos también identificar los grados de libertad correspondientes a cada componente, dados por las ecuaciones 10.3 y 10.4, respectivamente.

$$\nu_{sujetos} = n_{sujetos} - 1 \quad (10.3)$$

$$\nu_{error} = \nu_{wg} - \nu_{sujetos} \quad (10.4)$$

Para el ejemplo, tenemos:

$$\begin{aligned}\nu_{sujetos} &= 6 - 1 = 5 \\ \nu_{error} &= 20 - 5 = 15\end{aligned}$$

Las medias cuadradas del procedimiento ANOVA de una vía para muestras correlacionadas son, respectivamente, la media cuadrada entre grupos para el efecto (igual que para muestras independientes) y la media cuadrada del error, dada por la ecuación 10.5.

$$MS_{error} = \frac{SS_{error}}{\nu_{error}} \quad (10.5)$$

Así, tenemos que:

$$\begin{aligned}MS_{efecto} &= MS_{bg} = \frac{213,045}{3} = 71,015 \\ MS_{error} &= \frac{9,028}{15} = 0,602\end{aligned}$$

En consecuencia:

$$F = \frac{MS_{efecto}}{MS_{error}} = \frac{71,015}{0,602} = 117,992$$

Al hacer la llamada `pf(117.992, 3, 15, lower.tail = FALSE)` para calcular el valor p, obtenemos $p = 1,177 \cdot 10^{-10}$.

10.2.4 Resultado del procedimiento ANOVA

Una vez más, el resultado del procedimiento se representa en la forma tabular, como muestra la tabla 10.4.

Fuente	ν	SS	MS	F	p
Efecto	3	213,045	71,015	117,991	$1,177 \cdot 10^{-10}$
Error	15	9,028	0,602		
TOTAL	23	224,930			

Tabla 10.4: resultado del procedimiento ANOVA.

El valor p obtenido es muy menor a cualquier nivel de significación típico que podamos considerar, por lo que rechazamos la hipótesis nula en favor de la hipótesis alternativa. Así, el estudiante del ejemplo concluye con más de 99 % de confianza que existen diferencias significativas entre al menos dos de los algoritmos de ordenamiento comparados.

10.2.5 Resumen del procedimiento ANOVA de una vía para muestras correlacionadas

El procedimiento ANOVA de una vía para variables independientes puede resumirse en los siguientes pasos:

1. Calcular la suma de los cuadrados de las desviaciones para la muestra combinada (SS_T).
2. Para cada grupo g , calcular la suma de los cuadrados de las desviaciones dentro de dicho grupo (SS_g).
3. Calcular la variabilidad entre grupos (SS_{bg}).
4. Calcular la variabilidad al interior de los grupos (SS_{wg}).
5. Calcular la variabilidad intra-sujetos y la variabilidad del error ($SS_{sujetos}$ y SS_{error}).
6. Calcular los grados de libertad relevantes (ν_T , $\nu_{efecto} = \nu_{bg}$ y ν_{error}).
7. Calcular las medias cuadradas ($MS_{efecto} = MS_{bg}$ y MS_{error}).
8. Calcular el estadístico de prueba (F).
9. Obtener el valor p.

10.3 ANOVA DE UNA VÍA PARA MUESTRAS CORRELACIONADAS EN R

Para efectuar el procedimiento ANOVA de una vía para muestras correlacionadas en R, podemos usar las mismas funciones ya estudiadas en el capítulo 9: `aov()` y `ezANOVA()`, como ilustra el script 10.1.

En el caso de `aov()`, podemos apreciar que la fórmula entregada en la llamada (líneas 31–32 del script 10.1) es bastante diferente a la del capítulo 9. Esto se debe a que esta función realiza por defecto un procedimiento ANOVA para muestras independientes, por lo que se debe explicitar en la fórmula que se requiere descartar la variabilidad entre sujetos. La figura 10.2 muestra el resultado obtenido, idéntico al presentado en la tabla 10.4 salvo por ligeras diferencias de redondeo.

```
Resultado de la prueba ANOVA para muestras correlacionadas con aov

Error: Instancia
  Df Sum Sq Mean Sq F value Pr(>F)
Residuals  5  2.857  0.5714

Error: Instancia:Algoritmo
  Df Sum Sq Mean Sq F value    Pr(>F)
Algoritmo  3 213.04   71.01     118 1.18e-10 ***
Residuals 15   9.03    0.60
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figura 10.2: resultado del procedimiento ANOVA usando la función `aov()`.

La llamada a `ezANOVA()`, en cambio, es muy similar a la ya conocida, como se puede apreciar en las líneas 39–40 del script 10.1. Es importante destacar que la única diferencia con respecto a la llamada realizada en el capítulo 9 es que ahora el **argumento between** empleado en el capítulo 9 ha sido **reemplazado por within** para la variable independiente. Esta diferencia indica a `ezANOVA()` que se trata de un procedimiento ANOVA con muestras correlacionadas. La figura 10.3 muestra, una vez más, que se obtiene el mismo resultado.

Script 10.1: procedimiento ANOVA de una vía para muestras correlacionadas.

```
1 library(tidyverse)
2 library(ggpubr)
3 library(ez)
4
5 # Crear el data frame.
6 instancia <- factor(1:6)
7 Quicksort <- c(23.2, 22.6, 23.4, 23.3, 21.8, 23.9)
8 Bubblesort <- c(31.6, 29.3, 30.7, 30.8, 29.8, 30.3)
9 Radixsort <- c(30.1, 28.4, 28.7, 28.3, 29.9, 29.1)
10 Mergesort <- c(25.0, 25.7, 25.7, 23.7, 25.5, 24.7)
11 datos <- data.frame(instancia, Quicksort, Bubblesort, Radixsort, Mergesort)
12
13 # Llevar data frame a formato largo.
14 datos <- datos %>% pivot_longer(c("Quicksort", "Bubblesort", "Radixsort",
15                                     "Mergesort"),
16                                     names_to = "algoritmo", values_to = "tiempo")
17
18 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
19
20 # Comprobación de normalidad.
21 g <- ggqqplot(datos, x = "tiempo", y = "algoritmo", color = "algoritmo")
22 g <- g + facet_wrap(~ algoritmo)
23 g <- g + rremove("x.ticks") + rremove("x.text")
24 g <- g + rremove("y.ticks") + rremove("y.text")
25 g <- g + rremove("axis.title")
26 print(g)
27
28 # Procedimiento ANOVA con aov.
29 cat("Procedimiento ANOVA usando aov\n\n")
30
31 prueba <- aov(tiempo ~ algoritmo + Error(instancia/(algoritmo)),
32                 data = datos)
33
34 print(summary(prueba))
35
36 # Procedimiento ANOVA con ezANOVA().
37 cat("\n\nProcedimiento ANOVA usando ezANOVA\n\n")
38
39 prueba2 <- ezANOVA(data = datos, dv = tiempo, within = algoritmo,
40                      wid = instancia, return_aov = TRUE)
41
42 print(summary(prueba2$aov))
43 cat("\n\nPero ezANOVA entrega más información.\n")
44 cat("El resultado de la prueba de esfericidad de Mauchly:\n\n")
45 print(prueba2[["Mauchly's Test for Sphericity"]])
46
47 cat("\n\nY factores de corrección para cuando no se cumple la\n")
48 cat("condición de esfericidad:\n\n")
49 print(prueba2$`Sphericity Corrections`)
50
51 # Gráfico del tamaño del efecto.
52 g2 <- ezPlot(data = datos, dv = tiempo, wid = instancia, within = algoritmo,
53               y_lab = "Tiempo promedio de ejecución [ms]", x = algoritmo)
54
55 print(g2)
```

Resultado de la prueba ANOVA para muestras correlacionadas con ezANOVA

Error: Instancia

Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	5	2.857	0.5714	

Error: Instancia:Algoritmo

Df	Sum Sq	Mean Sq	F value	Pr(>F)
Algoritmo	3	213.04	71.01	118 1.18e-10 ***
Residuals	15	9.03	0.60	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Figura 10.3: resultado del procedimiento ANOVA usando la función ezANOVA().

Habíamos mencionado que otra ventaja de ezANOVA() es que verifica la condición de esfericidad mediante la prueba de esfericidad de Mauchly, cuyo resultado se muestra en la figura 10.4. Podemos apreciar que el valor p obtenido en esta prueba es muy alto ($p = 0,555$), de lo que se desprende que los datos del ejemplo sí cumplen con la condición de esfericidad (hipótesis nula de la prueba de Mauchly).

Resultado de la prueba de esfericidad de Mauchly

Effect	W	p	p<.05
2 Algoritmo	0.3367911	0.5545469	

Figura 10.4: resultado de la prueba de esfericidad de Mauchly realizada por ezANOVA().

Ahora bien, existen dos correcciones que suelen emplearse cuando se producen violaciones a la condición de esfericidad: la de **Greenhouse-Geisser** y la de **Huynd-Feldt**. Ambas estiman la esfericidad, denotada por ϵ , y corrigen el valor p de ANOVA en base a dicha estimación (ajustando los grados de libertad de la distribución F usada en el cálculo). La corrección de Greenhouse-Geisser es más conservadora y tiende a subestimar ϵ cuando esta es cercana a 1, por lo que se recomienda su uso para $\epsilon < 0,75$. Para $\epsilon \geq 0,75$ (estimada con el método Greenhouse-Geisser, de acuerdo a Karadimitriou y Marshall (2016)) suele emplearse la estimación de Huynd-Feldt, algo más liberal (Lærd Statistics, 2020b). ezANOVA() lleva a cabo ambas correcciones y reporta para cada una de ellas tanto la estimación de la esfericidad como el valor p corregido, como se aprecia en la figura 10.5:

- GGe: estimación de ϵ con el método de Greenhouse-Geisser.
- p[gg]: valor p tras la corrección de Greenhouse-Geisser.
- HFe: estimación de ϵ con el método de Huynd-Feldt.
- p[HF]: valor p tras la corrección de Huynd-Feldt.

Factores de corrección para cuando no se cumple la condición de esfericidad

Effect	GGe	p[GG]	p[GG]<.05	HFe	p[HF]	p[HF]<.05
2 Algoritmo	0.6803135	8.377723e-08	*	1.154155	1.177725e-10	*

Figura 10.5: correcciones de esfericidad realizadas por ezANOVA().

Si los datos del ejemplo no cumplieran con la esfericidad, deberíamos considerar p[GG] como p valor de la prueba, y no el valor (sin corregir) de la tabla entregada por ezANOVA() de la figura 10.3. Una vez más, podemos graficar el tamaño del efecto medido (script 10.1, líneas 52–55), obteniéndose como resultado el gráfico de la figura 10.6.

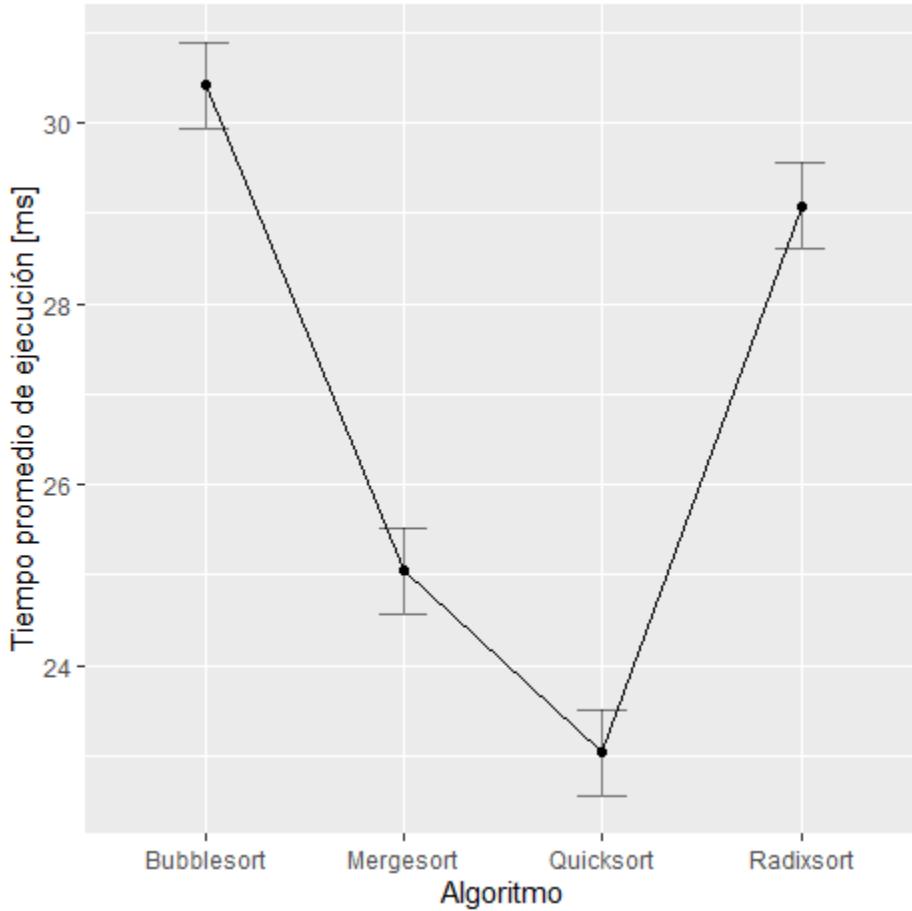


Figura 10.6: Tamaño del efecto medido.

10.4 PROCEDIMIENTOS POST-HOC

Podemos ocupar los mismos procedimientos post-hoc estudiados en el capítulo 9 tras realizar un procedimiento ANOVA de una vía con muestras correlacionadas.

En el caso de las correcciones de Bonferroni y Holm, lo único que cambia es que ahora debemos asignar el valor TRUE al argumento `paired` de la función `pairwise.t.test()`.

En cuanto a las pruebas HSD de Tukey y de Scheffé, su realización se dificulta al usar la tabla ANOVA resultante de un proceso de una vía para muestras correlacionadas. Esto se debe a que el formato del objeto `aov` resultante difiere al que se obtiene al realizar un procedimiento ANOVA para muestras independientes, por lo que las funciones del paquete `DescTools` arrojan un error. No obstante, existe una alternativa. El primer paso consiste en crear un modelo mixto (concepto que va más allá de los alcances de este documento) mediante la función `lme(formula, data, random)` del paquete `nlme`, donde:

- `formula`: tiene la forma `<variable dependiente> ~ <variable categórica>`.
- `data`: matriz de datos en formato largo.
- `random`: fórmula de la forma `~1|<identificador del sujeto>`.

Como segundo paso, estimamos la media de la variable dependiente, con su respectivo intervalo de confianza, para cada nivel de la variable categórica. Para esto usamos la función `emmeans(object, specs)` del paquete homónimo, donde:

- **object**: modelo mixto construido en el paso previo.
- **specs**: nombre del factor en estudio, delimitado por comillas.

Por último, estimamos las medias de las diferencias para los contrastes entre pares, con su error estándar y los valores p correspondientes, mediante la función `pairs(x, adjust)`, donde:

- **x**: diferencias obtenidas en el párrafo precedente.
- **adjust**: método para ajustar los valores p. “tukey” para el método HSD de Tukey, “scheffe” para el método de Scheffé.

Los mecanismos para estimar las medias marginales (`emmeans`) y para construir otros contrastes con el método de Scheffé van más allá de los alcances de este libro, pero pueden encontrarse en la documentación de los paquetes R involucrados.

El script 10.2 efectúa las pruebas post-hoc para el ejemplo, obteniéndose los resultados de la figura 10.7. Considerando el ajuste para múltiples pruebas de Tukey, podemos concluir con 99 % de confianza que todos los algoritmos tienen tiempos de ejecución distintos, con la excepción del par *bubblesort/radixsort*, para el que la evidencia no es suficiente para descartar que presentan el mismo tiempo de ejecución medio.

Script 10.2: pruebas post-hoc para el ejemplo.

```

1 library(tidyverse)
2 library(nlme)
3 library(emmeans)
4 library(ez)
5
6 # Crear el data frame.
7 instancia <- factor(1:6)
8 Quicksort <- c(23.2, 22.6, 23.4, 23.3, 21.8, 23.9)
9 Bubblesort <- c(31.6, 29.3, 30.7, 30.8, 29.8, 30.3)
10 Radixsort <- c(30.1, 28.4, 28.7, 28.3, 29.9, 29.1)
11 Mergesort <- c(25.0, 25.7, 25.7, 23.7, 25.5, 24.7)
12 datos <- data.frame(instancia, Quicksort, Bubblesort, Radixsort, Mergesort)
13
14 # Llevar data frame a formato largo.
15 datos <- datos %>% pivot_longer(c("Quicksort", "Bubblesort", "Radixsort",
16                                     "Mergesort"),
17                                     names_to = "algoritmo", values_to = "tiempo")
18
19 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
20
21 # Nivel de significación.
22 alfa <- 0.01
23
24 # Procedimiento ANOVA.
25 anova <- ezANOVA(data = datos, dv = tiempo, within = algoritmo,
26                    wid = instancia, return_aov = TRUE)
27
28 # Procedimiento post-hoc de Bonferroni.
29 bonferroni <- pairwise.t.test(datos[["tiempo"]], datos[["algoritmo"]],
30                                p.adj = "bonferroni", paired = TRUE)
31
32 cat("Corrección de Bonferroni\n")
33 print(bonferroni)
34
35 # Procedimiento post-hoc de Holm.
36 holm <- pairwise.t.test(datos[["tiempo"]], datos[["algoritmo"]],
37                           p.adj = "holm", paired = TRUE)
38
39 cat("\n\nCorrección de Holm\n")

```

```

40 print(holm)
41
42 # Procedimiento post-hoc HSD de Tukey.
43 mixto <- lme(tiempo ~ algoritmo, data = datos, random = ~1| instancia)
44 medias <- emmeans(mixto, "algoritmo")
45 tukey <- pairs(medias, adjust = "tukey")
46
47 cat("\n\nPrueba HSD de Tukey\n\n")
48 print(tukey)
49
50 # Procedimiento post-hoc de Scheffé
51 cat("\n\nComparación de Scheffé\n")
52 scheffe <- pairs(medias, adjust = "scheffe")
53 print(scheffe)

```

10.5 EJERCICIOS PROPUESTOS

1. ¿Cuándo aplica el procedimiento ANOVA de una vía para muestras correlacionadas? ¿Con qué otros nombres se encuentra?
2. Explica a qué se refiere “eliminar la variabilidad de los sujetos”.
3. ¿Cuáles son las hipótesis contrastadas en el procedimiento ANOVA de una vía para muestras correlacionadas?
4. Enumera las condiciones (o supuestos) del procedimiento ANOVA de una vía para muestras correlacionadas para tener confiabilidad.
5. El conjunto de datos ChickWeight registra el peso (en gramos) de 50 pollitos al momento de nacer y al cabo de varios días después de nacidos. Identifica si existen diferencias significativas en el peso de los pollitos al momento de nacer, al cabo de 4 días y luego de 8 días. En caso de detectarse tales diferencias, indica cuáles son.

Corrección de Bonferroni

Pairwise comparisons using paired t tests

data: Tiempo and Algoritmo

	Bubblesort	Mergesort	Quicksort
Mergesort	0.00112	-	-
Quicksort	1.4e-05	0.07196	-
Radixsort	0.09232	0.00088	0.00039

P value adjustment method: bonferroni

Corrección de Holm

Pairwise comparisons using paired t tests

data: Tiempo and Algoritmo

	Bubblesort	Mergesort	Quicksort
Mergesort	0.00059	-	-
Quicksort	1.4e-05	0.02399	-
Radixsort	0.02399	0.00059	0.00033

P value adjustment method: holm

Prueba HSD de Tukey

contrast	estimate	SE	df	t.ratio	p.value
Bubblesort - Mergesort	5.37	0.445	15	12.058	<.0001
Bubblesort - Quicksort	7.38	0.445	15	16.589	<.0001
Bubblesort - Radixsort	1.33	0.445	15	2.996	0.0403
Mergesort - Quicksort	2.02	0.445	15	4.531	0.0020
Mergesort - Radixsort	-4.03	0.445	15	-9.062	<.0001
Quicksort - Radixsort	-6.05	0.445	15	-13.594	<.0001

Degrees-of-freedom method: containment

P value adjustment: tukey method for comparing a family of 4 estimates

Comparación de Scheffé

contrast	estimate	SE	df	t.ratio	p.value
Bubblesort - Mergesort	5.37	0.445	15	12.058	<.0001
Bubblesort - Quicksort	7.38	0.445	15	16.589	<.0001
Bubblesort - Radixsort	1.33	0.445	15	2.996	0.0642
Mergesort - Quicksort	2.02	0.445	15	4.531	0.0040
Mergesort - Radixsort	-4.03	0.445	15	-9.062	<.0001
Quicksort - Radixsort	-6.05	0.445	15	-13.594	<.0001

Degrees-of-freedom method: containment

P value adjustment: scheffe method with rank 3

Figura 10.7: resultados de las pruebas post-hoc para el ejemplo.

CAPÍTULO 11. INFERENCIA NO PARAMÉTRICA CON MEDIANAS

En el capítulo 8 conocimos algunos métodos no paramétricos que podemos usar para inferir sobre frecuencias cuando nuestro conjunto de datos no cumple con las condiciones para poder usar, por ejemplo, las pruebas de Wald o de Wilson. Mencionamos también que este problema también puede ocurrir para el caso de inferir con medias, por lo que en este capítulo conoceremos alternativas no paramétricas para las pruebas t de Student (para una y dos medias) y ANOVA (para más de dos medias). Para ello nos basaremos principalmente en Lowry (1999, caps. 11a, 12a, 14a, 15a), Glen (2021c) y Lærd Statistics (2020a).

11.1 PRUEBAS PARA UNA O DOS MUESTRAS

En el capítulo 5 aprendimos que la prueba t de Student es adecuada para inferir acerca de una o dos medias muestrales, siempre y cuando se verifiquen algunas condiciones. En el caso de la prueba t de una muestra (o de la diferencia de dos muestras pareadas):

1. Las observaciones son independientes entre sí.
2. Las observaciones provienen de una distribución cercana a la normal.

En el caso de dos muestras independientes:

1. Cada muestra cumple las condiciones para usar la distribución t.
2. Las muestras son independientes entre sí.

Es importante mencionar también que la distribución normal es continua, de donde se desprende que la escala de medición empleada para la medición de las muestras debe ser de intervalos iguales.

Como ya vimos en el capítulo 8, si usamos la prueba t en un escenario en que no se cumple alguna de estas condiciones, el resultado no sería válido pues carecería de sentido y, en consecuencia, también lo harían las conclusiones que se obtengan a partir de él.

11.1.1 Prueba de suma de rangos de Wilcoxon

La **prueba de suma de rangos de Wilcoxon**, también llamada **prueba U de Mann-Whitney** o **prueba de Wilcoxon-Mann-Whitney**, es una alternativa no paramétrica a la prueba t de Student con muestras independientes. Pese a ser no paramétrica, requiere verificar el cumplimiento de las siguientes condiciones:

1. Las observaciones de ambas muestras son independientes.
2. La escala de medición empleada debe ser a lo menos ordinal, de modo que tenga sentido hablar de relaciones de orden (“igual que”, “menor que”, “mayor o igual que”).

Consideremos el siguiente contexto para estudiar la aplicación de esta prueba: una empresa de desarrollo de software desea evaluar la usabilidad de dos interfaces alternativas, *A* y *B*, para un nuevo producto de software. Con este fin, la empresa ha seleccionado al azar a 23 voluntarias y voluntarios, quienes son asignados de manera

aleatoria a dos grupos, cada uno de los cuales debe probar una de las interfaces ($n_A = 12$, $n_B = 11$). Cada participante debe evaluar 6 aspectos de usabilidad de la interfaz, cada uno de los cuales se mide con una escala Likert de 7 puntos, donde 1 significa “muy malo” y 7, “muy bueno”. La valoración que cada participante da a la interfaz evaluada corresponde al promedio simple de las puntuaciones de los 6 aspectos evaluados. La tabla 11.1 muestra las evaluaciones realizadas por cada participante.

	Interfaz A	Interfaz B
	2,7	5,0
	6,6	1,4
	1,6	5,6
	5,1	4,6
	3,7	6,7
	6,1	2,7
	5,0	1,3
	1,4	6,3
	1,8	3,7
	1,5	1,3
	3,0	6,8
	5,3	
Media	3,65	4,13

Tabla 11.1: evaluación de las interfaces de usuario A y B.

En este caso, si bien se cumple la condición de independencia de la prueba t de Student, no podemos usar esta prueba por dos razones: primero, no todas las escalas Likert pueden asegurar que son de igual intervalo. En el ejemplo, si dos participantes califican un aspecto de la interfaz A con notas 3 y 5, mientras que dos participantes califican esos aspectos con notas 4 y 6 para la interfaz B, ¿se podría asegurar que en ambos casos existe la misma diferencia de usabilidad (2 puntos)? Pocas escalas Likert tienen estudios de reproducibilidad que aseguren esta consistencia, por lo que no podríamos asumir que la escala es de intervalos iguales en este ejemplo. En segundo lugar, al revisar los histogramas para las muestras (figura 11.1) podemos observar que las distribuciones no se asemejan a una normal.

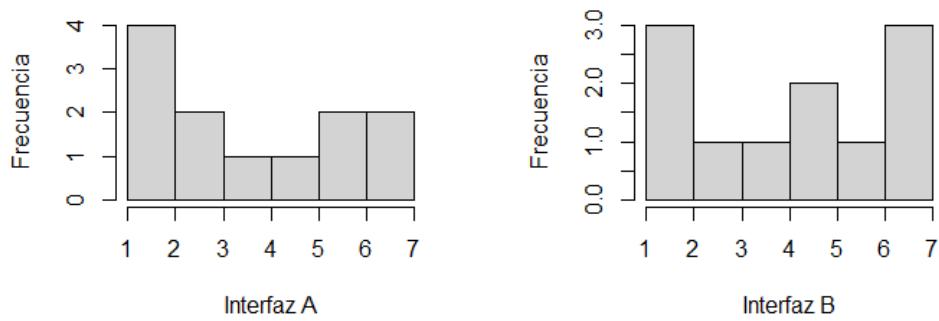


Figura 11.1: histogramas de las muestras.

Como alternativa, podemos usar la prueba no paramétrica de Wilcoxon-Mann-Whitney, cuyas hipótesis para el ejemplo son:

H_0 : no hay diferencia en la usabilidad de ambas interfaces (se distribuyen de igual forma).

H_A : sí hay diferencia en la usabilidad de ambas interfaces (distribuciones distintas).

Al igual que en el caso de la prueba χ^2 de Pearson, estas hipótesis no hacen referencia a algún parámetro de

una supuesta distribución para las poblaciones de puntuaciones de usabilidad, es decir, nos entregan menos información que la prueba paramétrica equivalente.

El primer paso de la prueba consiste en combinar todas las observaciones en un único conjunto de tamaño $n_T = n_A + n_B$ y ordenarlo de menor a mayor. A cada elemento se le asigna un **valor de rango** (*rank* en inglés) de 1 a n_T , de acuerdo a la posición que ocupa en el conjunto ordenado. En caso de que un valor aparezca más de una vez, cada repetición toma como valor el rango promedio de todas las ocurrencias del valor. La tabla 11.2 muestra el resultado de este proceso. Podemos notar que hay dos observaciones con valor 1, 3 a las que le corresponderían los rangos 1 y 2, por lo que, en consecuencia, ambas reciben el mismo valor de rango, igual al promedio 1,5. Esto también ocurre para las puntuaciones 1,4; 2,7; 3,7 y 5,0.

Observación	Muestra	Rango
1,3	B	1,5
1,3	B	1,5
1,4	A	3,5
1,4	B	3,5
1,5	A	5,0
1,6	A	6,0
1,8	A	7,0
2,7	A	8,5
2,7	B	8,5
3,0	A	10,0
3,7	A	11,5
3,7	B	11,5
4,6	B	13,0
5,0	A	14,5
5,0	B	14,5
5,1	A	16,0
5,3	A	17,0
5,6	B	18,0
6,1	A	19,0
6,3	B	20,0
6,6	A	21,0
6,7	B	22,0
6,8	B	23,0

Tabla 11.2: muestras combinadas con rango.

A continuación, se suman los rangos asociados a las observaciones de cada muestra, y para la muestra combinada. Así, para la muestra A obtenemos:

$$T_A = 3,5 + 5,0 + 6,0 + 7,0 + 8,5 + 10,0 + 11,5 + 14,5 + 16,0 + 17,0 + 19,0 + 21,0 = 139$$

De manera análoga, para la muestra B se tiene:

$$T_B = 1,5 + 1,5 + 3,5 + 8,5 + 11,5 + 13,0 + 14,5 + 18,0 + 20,0 + 22,0 + 23,0 = 137$$

La suma de rangos para la muestra combinada siempre está dada por la ecuación 11.1.

$$T_T = \frac{n_T \cdot (n_T + 1)}{2} \quad (11.1)$$

Para el ejemplo:

$$T_T = \frac{23 \cdot (23 + 1)}{2} = 276$$

Trabajar con los rangos en lugar de las observaciones nos ofrece dos ventajas: la primera es que el foco solo está en las relaciones de orden entre las observaciones, sin necesidad de que estas provengan de una escala de intervalos iguales. La segunda es que esta transformación facilita conocer de manera sencilla algunas propiedades del conjunto de datos. Por ejemplo, la suma de rangos de la muestra se determina siempre mediante la ecuación 11.1 y la media de rangos de la muestra combinada es siempre como muestra la ecuación 11.2.

$$\mu = \frac{n_T \cdot (n_T + 1)}{2} \cdot \frac{1}{n_T} = \frac{n_T + 1}{2} \quad (11.2)$$

Para el ejemplo:

$$\mu = \frac{23 + 1}{2} = 12$$

En consecuencia, la hipótesis nula en el dominio de los rangos es que las medias de los rangos de las dos muestras son iguales. Si la hipótesis nula fuera cierta, las observaciones en ambas muestras serían similares, por lo que, al ordenar la muestra combinada, ambas muestras se mezclarían de manera homogénea. En consecuencia, deberíamos esperar que los promedios de rangos para cada muestra se aproximen al rango promedio de la muestra combinada, es decir, que T_A y T_B se aproximen a los siguientes valores:

$$\begin{aligned}\mu_A &= n_A \cdot \frac{(n_T) + 1}{2} = 12 \cdot \frac{(23 + 1)}{2} = 144 \\ \mu_B &= n_B \cdot \frac{(n_T) + 1}{2} = 11 \cdot \frac{(23 + 1)}{2} = 132\end{aligned}$$

La prueba de Wilcoxon-Mann-Whitney tiene dos variantes, una para muestras grandes y otra para muestras pequeñas, que se diferencian a partir de este punto.

11.1.1.1 Prueba de suma de rangos de Wilcoxon para muestras grandes

Hasta ahora, hemos determinado que:

- El valor observado $T_A = 139$ pertenece a una distribución muestral con media $\mu_A = 144$.
- El valor observado $T_B = 137$ pertenece a una distribución muestral con media $\mu_B = 132$.

Bajo el supuesto de que la hipótesis nula sea verdadera, podríamos demostrar que las distribuciones muestrales de T_A y T_B tienen la misma desviación estándar, dada por la ecuación 11.3.

$$\sigma_T = \sqrt{\frac{n_A \cdot n_B \cdot (n_T + 1)}{12}} \quad (11.3)$$

Con lo que:

$$\sigma_T = \sqrt{\frac{12 \cdot 11 \cdot (23 + 1)}{12}} = 16,248$$

Cuando **ambas muestras tienen tamaño mayor o igual a 5**, siguiendo un procedimiento similar al descrito en la primera sección del capítulo 4, podemos demostrar que las distribuciones muestrales de T_A y T_B tienden a aproximarse a la distribución normal. En consecuencia, una vez conocidas la media y la desviación estándar de una distribución normal para la muestra, podemos calcular el estadístico z para T_A o T_B , dado por la ecuación 11.4, donde:

- T_{obs} es cualquiera de los valores observados, T_A o T_B .
- μ_{obs} es la media de la distribución muestral de T_{obs} .
- σ_T es la desviación estándar de la distribución muestral de T_{obs} (es decir, el error estándar).

$$z = \frac{(T_{obs} - \mu_{obs}) \pm 0,5}{\sigma_T} \quad (11.4)$$

Puesto que las distribuciones muestrales de T son intrínsecamente discretas (solo pueden asumir valores con decimales cuando existen rangos empataos), debemos emplear un factor de corrección de continuidad:

- $-0,5$ si $T_{obs} > \mu_{obs}$.
- $0,5$ si $T_{obs} < \mu_{obs}$.

Volviendo al ejemplo, tenemos:

$$\begin{aligned} z_A &= \frac{(139 - 144) + 0,5}{16,248} = -0,277 \\ z_B &= \frac{(137 - 132) - 0,5}{\sigma_T} = 0,277 \end{aligned}$$

Los valores z obtenidos a partir de T_A y T_B siempre tienen igual valor absoluto y signos opuestos, por lo que no importa cuál de ellos usemos para la prueba de significación estadística. No obstante, debemos tener muy claro el significado del signo de z : si para el ejemplo tuviésemos como hipótesis alternativa que la interfaz A es mejor que la interfaz B, entonces esperaríamos que las observaciones de mayor rango estuvieran en el grupo A, por lo que z_A tendría que ser positivo.

El valor z obtenido permite calcular el valor p para una hipótesis alternativa unilateral (pues solo delimita la región de rechazo en una de las colas de la distribución normal subyacente). Así, para el ejemplo, cuya hipótesis alternativa es bilateral, podemos calcular el valor p en R mediante la llamada `2 * pnorm(-0.277, mean = 0, sd = 1, lower.tail = TRUE)`, obteniéndose como resultado $p = 0,782$.

Evidentemente, el valor p obtenido es muy alto, por lo que fallamos al rechazar la hipótesis nula. En consecuencia, podemos concluir que no es posible descartar que las dos interfaces tienen niveles de usabilidad similares.

11.1.1.2 Prueba de suma de rangos de Wilcoxon para muestras pequeñas

Cuando las muestras son pequeñas (menos de 5 observaciones), no podemos usar el supuesto de normalidad del apartado anterior, por lo que necesitamos una vía alternativa. Este método sirve también para muestras más grandes, con resultados equivalentes a los ya obtenidos.

Aprovechando una vez más las ventajas de considerar los rangos en lugar de las observaciones originales, podemos calcular el máximo valor posible para la suma de rangos de cada muestra como indica la ecuación 11.5. Fijémonos en que el valor máximo para la suma de rangos de una muestra se produce cuando esta contiene los n_x rangos mayores de la muestra combinada.

$$T_{x[\max]} = n_x \cdot n_y + \frac{n_x \cdot (n_x + 1)}{2} \quad (11.5)$$

Así, para el ejemplo:

$$\begin{aligned} T_{A[\max]} &= 12 \cdot 11 + \frac{12 \cdot (12 + 1)}{2} = 210 \\ T_{B[\max]} &= 11 \cdot 12 + \frac{11 \cdot (11 + 1)}{2} = 198 \end{aligned}$$

Con esto podemos definir un nuevo estadístico de prueba U , como muestra la ecuación 11.6.

$$U_x = T_{x[\max]} - T_x \quad (11.6)$$

Por lo que:

$$\begin{aligned} U_A &= 210 - 139 = 71 \\ U_B &= 198 - 137 = 61 \end{aligned}$$

El valor del estadístico de prueba es el mínimo entre U_A y U_B , por lo que $U = 61$.

Debemos notar que siempre se cumple la identidad presentada en la ecuación 11.7, por lo que podemos escoger cualquiera de los valores U obtenidos para realizar el resto del procedimiento.

$$U_A + U_B = n_A \cdot n_B \quad (11.7)$$

Si la hipótesis nula fuese cierta, esperaríamos que:

$$\begin{aligned} U_A &= T_{A[\max]} - \mu_A = 210 - 144 = 66 \\ U_B &= T_{B[\max]} - \mu_B = 198 - 132 = 66 \end{aligned}$$

Formalmente, entonces, si la hipótesis nula fuera verdadera, esperaríamos que:

$$U_A = U_B = \frac{n_A \cdot n_B}{2}$$

En consecuencia, la pregunta asociada a la prueba de hipótesis es: si la hipótesis nula es verdadera (no hay diferencias significativas en la usabilidad de ambas interfaces), ¿qué tan probable es obtener un valor de U al menos tan pequeño como el observado ($U = 61$)? Para responder a esta pregunta, seguimos un procedimiento similar al que ya conocimos para la prueba exacta de Fisher (capítulo 8): se calculan todas las formas en que n_T rangos podrían combinarse en dos grupos de tamaños n_A y n_B , y luego se determina la proporción de las combinaciones que produzcan un valor de U al menos tan pequeño como el encontrado. Pero ¡existen 676.039 combinaciones posibles!

Aunque R no ofrece herramientas para calcular el valor p a partir del estadístico U (pues utiliza el estadístico W , propuesto por Frank Wilcoxon en 1945, que lleva a los mismos resultados), afortunadamente existen tablas que permiten conocer el máximo valor de U para el cual se rechaza la hipótesis nula para un nivel de significación dado sin tener que revisar todas las combinaciones. Considerando $\alpha = 0,05$ para una prueba bilateral, el valor crítico es $U = 33$ (Real Statistics Using Excel, s.f.). Puesto que $61 > 33$, fallamos al rechazar la hipótesis nula, por lo que concluimos con 95 % de confianza que no se puede descartar que la usabilidad de ambas interfaces sea la misma.

11.1.1.3 Prueba de suma de rangos de Wilcoxon en R

Como ya dijimos, la implementación de esta prueba en R usa el estadístico W (introducido por Wilcoxon) en lugar del estadístico U empleado por Mann y Whitney. Es por ello que esta prueba se realiza mediante la función `wilcox.test(x, y, paired = FALSE, alternative, mu, conf.level)`, donde:

- `x, y`: vectores numéricos con las observaciones. Para aplicar la prueba con una única muestra, `y` debe ser nulo (por defecto, lo es).
- `paired`: booleano con valor falso para indicar que las muestras son independientes (se asume por defecto).
- `alternative`: señala el tipo de hipótesis alternativa: bilateral ("two.sided") o unilateral ("less" o "greater").
- `mu`: valor nulo¹.
- `conf.level`: nivel de confianza.

El script 11.1 muestra la aplicación de esta prueba para el ejemplo, obteniéndose los resultados que se presentan en la figura 11.2.

```
Wilcoxon rank sum test with continuity correction

data: a and b
W = 61, p-value = 0.7816
alternative hypothesis: true location shift is not equal to 0
```

Figura 11.2: resultado de la prueba de Mann-Whitney (en rigor, de la prueba para el ejemplo).

Script 11.1: prueba de Mann-Whitney para el ejemplo.

```
1 # Ingresar los datos.
2 a <- c(2.7, 6.6, 1.6, 5.1, 3.7, 6.1, 5.0, 1.4, 1.8, 1.5, 3.0, 5.3)
3 b <- c(5.0, 1.4, 5.6, 4.6, 6.7, 2.7, 1.3, 6.3, 3.7, 1.3, 6.8)
4
5 # Establecer nivel de significación.
6 alfa <- 0.05
7
8 # Hacer la prueba de Mann-Whitney.
9 prueba <- wilcox.test(a, b, alternative = "two.sided", conf.level = 1 - alfa)
10 print(prueba)
```

¹ Hay un poco de confusión (especialmente en Internet) respecto a las hipótesis que son contrastadas por estas pruebas. El argumento `mu` de la función `wilcox.test()` define el valor nulo de la prueba. Cuando se trabaja con una muestra (no ejemplificado en este capítulo) o la diferencia de dos muestras pareadas (como se discute en la siguiente sección), se prueba la hipótesis nula que la distribución de origen es simétrica en torno al valor `mu`. Esto equivale a decir que `mu` es el valor nulo para la **mediana** de la distribución de origen, en el primer caso, o de la distribución de las diferencias de las variables de origen, en el segundo. Cuando se comparan dos grupos independientes, se prueba la hipótesis que los parámetros de localización de las distribuciones de `x` e `y` difieren en `mu`. Solo cuando estas distribuciones de origen tienen la **misma forma** (igual simetría y varianza), esto es equivalente a verificar que poblaciones tienen las **mismas medianas**. Cuando la prueba es unilateral, solo se revisa si el parámetro de localización de la distribución de `x` está a la izquierda (`alternative = "less"`) o a la derecha (`alternative = "greater"`) del de la distribución de `y`.

11.1.2 Prueba de rangos con signo de Wilcoxon

La **prueba de rangos con signo de Wilcoxon** es, conceptualmente, parecida a la prueba de suma de rangos de Wilcoxon presentada en la sección anterior. Sin embargo, en este caso es la alternativa no paramétrica a la prueba t de Student con muestras pareadas. Las condiciones que se deben cumplir para usar esta prueba son:

1. Los pares de observaciones son independientes.
2. La escala de medición empleada para las observaciones es intrínsecamente continua.
3. La escala de medición empleada para ambas muestras debe ser a lo menos ordinal.

Consideremos ahora un nuevo contexto para la aplicación de esta prueba. Una empresa de desarrollo desea evaluar la usabilidad de dos interfaces alternativas, *A* y *B*, para un nuevo producto de software, a fin de determinar si, como asegura el departamento de diseño, es mejor la interfaz *A*. Para ello, la empresa ha seleccionado a 10 participantes al azar, quienes deben evaluar 6 aspectos de usabilidad de cada interfaz, cada uno de los cuales se mide con una escala Likert de 7 puntos, donde 1 significa “muy malo” y 7, “muy bueno”. La valoración que un participante da a la interfaz evaluada corresponde al promedio simple de las puntuaciones de los 6 aspectos evaluados. La tabla 11.3 muestra las evaluaciones realizadas por cada participante a cada una de las interfaces.

Participante	Interfaz A	Interfaz B
1	2,9	6,0
2	6,1	2,8
3	6,7	1,3
4	4,7	4,7
5	6,4	3,1
6	5,7	1,8
7	2,7	2,9
8	6,9	4,0
9	1,7	2,3
10	6,4	1,6

Tabla 11.3: evaluación de las interfaces de usuario A y B.

Formalmente, las hipótesis son:

H_0 : las mismas personas no perciben diferencia en la usabilidad de ambas interfaces.

H_A : las mismas personas consideran que la interfaz A tiene mejor usabilidad que la interfaz B.

La mecánica inicial para esta prueba consiste en calcular las diferencias entre cada par de observaciones y obtener luego su valor absoluto. Generalmente se descartan aquellas instancias con diferencia igual a cero, pues no aportan información relevante al procedimiento. A continuación se ordenan las diferencias absolutas en orden creciente y se les asignan rangos de manera correlativa del mismo modo que en la prueba de Wilcoxon-Mann-Whitney. Una vez asignados los rangos, se les incorpora el signo asociado a la diferencia. La tabla 11.4 ilustra el proceso descrito.

Una vez realizado este proceso, se calcula el estadístico de prueba W , correspondiente a la suma de los rangos con signo. Debemos notar que, tras eliminar aquellas observaciones con diferencia igual a 0, el tamaño de las muestras para el ejemplo es $n = 9$. Así:

$$W = -1 + -2 + 3 + -4 + 5, 5 + 5, 5 + 7 + 8 + 9 = 31$$

Desde luego, el máximo valor posible para W , W_{max} corresponde a la suma de los n rangos sin signo (todos positivos) (ecuación 11.2) y, además, $W_{min} = -|W_{max}|$.

Participante	Interfaz A	Interfaz B	A-B	A-B	Rango absoluto	Rango con signo
4	4,7	4,7	0,0	0	-	-
7	2,7	2,9	-0,2	0,2	1	-1
9	1,7	2,3	-0,6	0,6	2	-2
8	6,9	4,0	2,9	2,9	3	+3
1	2,9	6,0	-3,1	3,1	4	-4
2	6,1	2,8	3,3	3,3	5,5	+5,5
5	6,4	3,1	3,3	3,3	5,5	+5,5
6	5,7	1,8	3,9	3,9	7	+7
10	6,4	1,6	4,8	4,8	8	+8
3	6,7	1,3	5,4	5,4	9	+9

Tabla 11.4: asignación de rangos con signo.

Para entender mejor la distribución de W , una muestra de tamaño n genera n rangos no empatados sin signo (columna “Rango absoluto” de la tabla 11.4). A su vez, cada uno de dichos rangos podría tomar valores positivos o negativos, por lo que para W se tienen 2^n combinaciones para los signos de los rangos. La tabla 11.5 muestra todas las posibles combinaciones para $n = 3$. Si la hipótesis nula fuese cierta, los rangos positivos y negativos se distribuirían de manera homogénea, por lo que esperaríamos que el valor de W fuese cercano a 0 (hipótesis nula en el dominio de los rangos).

La figura 11.3 muestra la distribución de W para distintos valores de n . En ella podemos apreciar que, a medida que n aumenta, la distribución de W se aproxima cada vez más a una distribución normal centrada en $\mu_W = 0$.

Rango			
1	2	3	W
+	+	+	6
+	+	-	0
+	-	+	2
+	-	-	-4
-	+	+	4
-	+	-	-2
-	-	+	0
-	-	-	-6

Tabla 11.5: valores que puede adoptar el estadístico W para $n = 3$.

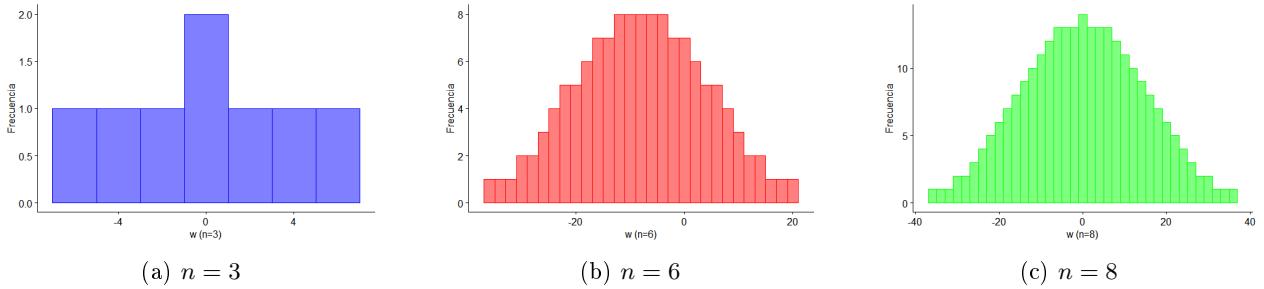


Figura 11.3: distribución de W .

La desviación estándar de la distribución muestral de W está dada por la ecuación 11.8.

$$\sigma_W = \sqrt{\frac{n \cdot (n+1) \cdot (2n+1)}{6}} \quad (11.8)$$

Para el ejemplo:

$$\sigma_W = \sqrt{\frac{9 \cdot (9 + 1) \cdot (2 \cdot 9 + 1)}{6}} = 16,882$$

Puesto que estamos trabajando bajo el supuesto de normalidad, calculamos el estadístico de prueba z , dado por la ecuación 11.9.

$$z = \frac{W - 0,5}{\sigma_W} \quad (11.9)$$

Así, para el ejemplo tenemos que:

$$z = \frac{31 - 0,5}{16,882} = 1,807$$

Una vez conocido el estadístico de prueba, podemos obtener el valor p mediante la llamada `pnorm(1.807, mean = 0, sd = 1, lower.tail = FALSE)` (no multiplicamos por 2, pues es una prueba unilateral), obteniendo como resultado $p = 0,035$. Considerando un nivel de significación $\alpha = 0,05$, rechazamos la hipótesis nula en favor de la hipótesis alternativa. En consecuencia, concluimos con 95 % de confianza que la usabilidad de la interfaz A es mejor que la de la interfaz B .

Siempre debemos tener en cuenta que el supuesto de normalidad es válido únicamente para $n > 10$, por lo que en caso de que las muestras sean más pequeñas, tenemos que consultar la tabla de valores críticos para la distribución W .

En R, la prueba de rangos con signo de Wilcoxon está implementada en la misma función que en el caso de muestras independientes, pero ahora la llamada es `wilcox.test(x, y, paired = TRUE, alternative, conf.level)`, donde:

- `x, y`: vectores numéricos con las observaciones.
- `paired`: booleano con valor verdadero para indicar que las muestras son pareadas.
- `alternative`: señala el tipo de hipótesis alternativa: bilateral ("two.sided") o unilateral ("less" o "greater"). `mu`: valor nulo de la prueba.
- `conf.level`: nivel de confianza.

Así, el valor por defecto para el parámetro `paired` es `FALSE`, en cuyo caso se efectúa la prueba de suma de rangos de Wilcoxon; mientras que si explícitamente indicamos `paired = TRUE`, se aplica la prueba de rangos con signo de Wilcoxon.

El script 11.2 muestra la aplicación de la prueba de rangos con signo de Wilcoxon para el ejemplo, obteniéndose los resultados que se presentan en la figura 11.4. Es importante tener en cuenta en cuenta que R usa una variante ligeramente diferente. En lugar del estadístico de prueba W , calcula el estadístico V , correspondiente a la suma de los rangos con signo positivo.

```
Wilcoxon signed rank test with continuity correction

data: a and b
V = 38, p-value = 0.03778
alternative hypothesis: true location shift is greater than 0
```

Figura 11.4: resultado de la prueba de rangos con signo de Wilcoxon para el ejemplo.

Script 11.2: prueba de rangos con signo de Wilcoxon para el ejemplo.

```
1 # Ingresar los datos.
2 a <- c(2.9, 6.1, 6.7, 4.7, 6.4, 5.7, 2.7, 6.9, 1.7, 6.4)
```

```

3 b <- c(6.0, 2.8, 1.3, 4.7, 3.1, 1.8, 2.9, 4.0, 2.3, 1.6)
4
5 # Establecer nivel de significación.
6 alfa <- 0.05
7
8 # Hacer la prueba de rangos con signo de Wilcoxon.
9 prueba <- wilcox.test(a, b, alternative = "greater", paired = TRUE,
10                         conf.level = 1 - alfa)
11
12 print(prueba)

```

11.2 PRUEBAS PARA MÁS DE DOS MUESTRAS

Al igual que existen alternativas no paramétricas para inferir con una o dos medias muestrales, también las hay para cuando se tienen más de dos muestras. Conoceremos ahora alternativas no paramétricas para el procedimiento ANOVA de una vía, tanto para muestras independientes como para muestras correlacionadas.

11.2.1 Prueba de Kruskal-Wallis

En el capítulo 9 estudiamos el procedimiento ANOVA de una vía para $k > 2$ muestras independientes, el cual requiere el cumplimiento de los siguientes supuestos:

1. La escala con que se mide la variable dependiente tiene las propiedades de una escala de intervalos iguales.
2. Las k muestras son obtenidas de manera aleatoria e independiente desde la(s) población(es) de origen.
3. Se puede suponer razonablemente que la(s) población(es) de origen sigue(n) una distribución normal.
4. Las k muestras tienen varianzas aproximadamente iguales.

Si bien ANOVA es usualmente robusto ante desviaciones leves de las condiciones (excepto la segunda) cuando las muestras son de igual tamaño, no ocurre lo mismo cuando los tamaños de las muestras difieren. En este caso, una alternativa es emplear la **prueba de Kruskal-Wallis**, cuyas condiciones son:

1. La variable independiente debe tener a lo menos dos niveles (aunque, para dos niveles, se suele usar la prueba de Wilcoxon-Mann-Whitney).
2. La escala de la variable dependiente debe ser, a lo menos, ordinal.
3. Las observaciones son independientes entre sí.

Para ilustrar esta prueba, tomemos el ejemplo de un ingeniero que cuenta con cuatro algoritmos (A, B, C y D) para resolver un determinado problema (en iguales condiciones y para instancias de tamaño fijo) y desea comparar su eficiencia. Para cada algoritmo, selecciona una muestra aleatoria independiente de instancias y registra el tiempo de ejecución (en milisegundos) del algoritmo en cuestión para cada una de las instancias de la muestra correspondiente, obteniendo las siguientes mediciones:

- Algoritmo A: 21, 22, 22, 23, 23, 23, 23, 24, 24, 24, 25, 26
- Algoritmo B: 15, 17, 18, 18, 19, 19, 20, 20, 21
- Algoritmo C: 9, 10, 10, 10, 11, 11, 12, 12, 12, 13, 14, 15
- Algoritmo D: 15, 15, 16, 16, 16, 18, 18, 18, 18

Las hipótesis a contrastar son, entonces:

H_0 : todos los algoritmos son igual de eficientes (o, de manera similar, ningún algoritmo es menos ni más eficiente que los demás).

H_A : al menos uno de los algoritmos presenta una eficiencia diferente a al menos algún otro algoritmo.

El procedimiento de la prueba de Kruskal-Wallis tiene elementos similares a los descritos en las pruebas no paramétricas para una y dos medias. El primer paso consiste en combinar las muestras y luego asignar el rango a cada elemento, obteniéndose para el ejemplo el resultado de la tabla 11.6.

Observaciones				Ranking de obs.			
A	B	C	D	A	B	C	D
21	15	9	15	31,5	15,5	1,0	15,5
22	17	10	15	33,5	21,0	3,5	15,5
22	18	10	16	33,5	24,0	3,5	19,0
23	18	10	16	36,5	24,0	3,5	19,0
23	19	10	16	36,5	27,5	3,5	19,0
23	19	11	18	36,5	27,5	8,5	24,0
23	20	11	18	36,5	29,5	8,5	24,0
24	20	12	18	40,0	29,5	8,5	24,0
24	21	12		40,0	31,5	8,5	
24		12		40,0		8,5	
25		12		42,0		8,5	
26		13		43,0		12,0	
		14				13,0	
		15				15,5	

Tabla 11.6: asignación de rangos a la muestra combinada.

A continuación se calcula la suma (T_x) y la media (M_x) de los rangos en cada grupo y en la muestra combinada. La tabla 11.7 presenta los valores obtenidos para el ejemplo, incluyendo además el tamaño muestral (n_x).

	A	B	C	D	Combinada
n	12	9	14	8	43
T	449,50	230,00	106,50	160,00	946,00
M	37,46	25,56	7,61	20,00	22,00

Tabla 11.7: resumen de los rangos.

De manera similar a ANOVA, se requiere determinar la diferencia entre las medias grupales. Para ello se calculan las desviaciones cuadradas de las medias grupales de los rangos con respecto a la media total de los rangos. Así, la variabilidad entre grupos está dada por la ecuación 11.10.

$$SS_{bg(R)} = \sum_{i=1}^k n_i \cdot (M_i - M_T)^2 \quad (11.10)$$

Para el ejemplo, entonces:

$$\begin{aligned} SS_{bg(R)} &= n_A \cdot (M_A - M_T)^2 + n_B \cdot (M_B - M_T)^2 + n_C \cdot (M_C - M_T)^2 + n_D \cdot (M_D - M_T)^2 = \\ &12 \cdot (37,46 - 22)^2 + 9 \cdot (25,56 - 22)^2 + 14 \cdot (7,61 - 22)^2 + 8 \cdot (20 - 22)^2 = 5.913,21 \end{aligned}$$

La hipótesis nula, llevada al dominio de los rangos, es que los rangos medios de los distintos grupos no serán muy diferentes entre sí. Podría esperarse que el valor nulo para $SS_{bg(R)}$ fuera 0, no obstante, no es

así. Supongamos por un momento que tenemos 3 muestras con dos observaciones cada una, con lo que tendríamos un total de 6 rangos. Dichos rangos pueden combinarse de 90 maneras distintas para formar tres grupos con dos elementos. La distribución muestral de $SS_{bg(R)}$ estaría dada, entonces, por los valores de $SS_{bg(R)}$ obtenidos para cada una de las 90 combinaciones, de los cuales únicamente 6 son iguales a 0 y todos los restantes, mayores que 0 (recuerde que es matemáticamente imposible obtener desviaciones cuadradas con valor negativo). La media de la distribución muestral para $SS_{bg(R)}$ está dada por la ecuación 11.11.

$$\mu_{SS} = (k - 1) \frac{n_T \cdot (n_T + 1)}{12} \quad (11.11)$$

Para el ejemplo, entonces, tenemos que el valor nulo es:

$$\mu_{SS} = (4 - 1) \frac{43 \cdot (43 + 1)}{12} = 473$$

Llegado este punto, se define el estadístico de prueba H , el cual se construye en torno al valor obtenido para $SS_{bg(R)}$ y parte de la fórmula empleada para calcular el valor nulo, como muestra la ecuación 11.12.

$$H = \frac{SS_{bg(R)}}{\frac{n_T \cdot (n_T + 1)}{12}} = \frac{12 \cdot SS_{bg(R)}}{n_T \cdot (n_T + 1)} \quad (11.12)$$

En consecuencia, el valor del estadístico de prueba para el ejemplo es:

$$H = \frac{12 \cdot 5.913,21}{43 \cdot (43 + 1)} = 37.5$$

Cuando cada uno de los k grupos tiene a lo menos 5 observaciones, el estadístico de prueba H sigue una distribución χ^2 con $\nu = k - 1$ grados de libertad. Así, podemos calcular el valor p para el ejemplo (en R) mediante la llamada `pchisq(37.5, 3, lower.tail = FALSE)`, obteniéndose como resultado $p = 3.606 \cdot 10^{-8}$. Este valor indica que la evidencia es suficientemente fuerte como para rechazar la hipótesis nula en favor de la hipótesis alternativa, incluso para un nivel de significación $\alpha = 0,01$. En consecuencia, podemos concluir con 99% de confianza que existen diferencias significativas entre los tiempos promedio de ejecución de los algoritmos A , B , C y D .

Fijémonos en que, al igual que ANOVA, la prueba de Kruskal-Wallis es de tipo ómnibus, por lo que no entrega información en relación a cuáles grupos presentan diferencias. En consecuencia, una vez más es necesario efectuar un análisis post-hoc cuando se detectan diferencias significativas. De manera similar a la estudiada en el capítulo 9, podemos hacer comparaciones entre pares de grupos con la prueba de Wilcoxon-Mann-Whitney (equivalentes a las realizadas con la prueba t de Student para ANOVA de una vía para muestras independientes), usando alguno de los factores de corrección que ya conocimos en el capítulo 8 (por ejemplo, Holm y Bonferroni) (Amat Rodrigo, 2016c).

En R, podemos ejecutar la prueba de Kruskal-Wallis mediante la función `kruskal.test(formula, data)`, donde:

- **formula:** tiene la forma <variable dependiente> ~ <variable independiente (factor)>.
- **data:** matriz de datos en formato largo.

Para los procedimientos post-hoc, las pruebas de Bonferroni y Holm pueden realizarse mediante la función `pairwise.wilcox.test(x, g, p.adjust.method, paired = FALSE)`, donde:

- **x:** vector con la variable dependiente.
- **g:** factor o agrupamiento.
- **p.adjust.method:** puede ser “holm” o “bonferroni”, entre otras alternativas.
- **paired:** valor booleano que indica si la prueba es pareada (verdadero) o no. Para la prueba de Kruskal-Wallis debe ser FALSE.

El script 11.3 muestra la realización de la prueba de Kruskal-Wallis para el ejemplo e incorpora el procedimiento post-hoc de Holm. Los resultados se presentan en la figura 11.5. Podemos ver que el valor p difiere ligeramente al obtenido anteriormente, debido a errores de redondeo. A partir de los resultados del procedimiento post-hoc, considerando un nivel de significación $\alpha = 0,01$, podemos concluir con 99 % de confianza que existen diferencias significativas entre los tiempos promedio de ejecución de todos los pares de algoritmos con excepción de los algoritmos *B* y *D*.

```
Kruskal-Wallis rank sum test

data: Tiempo by Algoritmo
Kruskal-Wallis chi-squared = 37.714, df = 3,
p-value = 3.249e-08

Pairwise comparisons using Wilcoxon rank sum test with continuity correction

data: datos$Tiempo and datos$Algoritmo

  A      B      C
B 0.00060 -     -
C 9.3e-05 0.00042 -
D 0.00060 0.02738 0.00060

P value adjustment method: holm
```

Figura 11.5: resultado de la prueba de Kruskal-Wallis y el procedimiento post-hoc de Holm para el ejemplo.

Script 11.3: prueba de Kruskal-Wallis y el procedimiento post-hoc de Holm para el ejemplo.

```
1 # Construir la matriz de datos.
2 A <- c(24, 23, 26, 21, 24, 24, 25, 22, 23, 22, 23, 23)
3 B <- c(17, 15, 18, 20, 19, 21, 20, 18, 19)
4 C <- c(10, 11, 14, 11, 15, 12, 12, 10, 9, 13, 12, 12, 10, 10)
5 D <- c(18, 16, 18, 15, 16, 15, 18, 16)
6 Tiempo <- c(A, B, C, D)
7
8 Algoritmo <- c(rep("A", length(A)),
9                 rep("B", length(B)),
10                rep("C", length(C)),
11                rep("D", length(D)))
12
13 Algoritmo <- factor(Algoritmo)
14
15 datos <- data.frame(Tiempo, Algoritmo)
16
17 # Establecer nivel de significación
18 alfa <- 0.01
19
20 # Hacer la prueba de Kruskal-Wallis.
21 prueba <- kruskal.test(Tiempo ~ Algoritmo, data = datos)
22 print(prueba)
23
24 # Efectuar procedimiento post-hoc de Holm si se encuentran diferencias
25 # significativas.
26 if(prueba$p.value < alfa) {
27   post_hoc <- pairwise.wilcox.test(datos$Tiempo,
28                                     datos$Algoritmo,
```

```

29      p.adjust.method = "holm",
30      paired = FALSE)
31
32  print(post_hoc)
33 }

```

Notemos que `pairwise.wilcox.test()` solo reporta los p valores ajustados. Si queremos conocer el tamaño del efecto de las diferencias detectadas, debemos realizar las correspondientes pruebas de Wilcoxon-Mann-Whitney para todos los pares de grupos que presenten diferencias significativas.

11.2.2 Prueba de Friedman

Como es natural suponer, podemos considerar la **prueba de Friedman** como una alternativa no paramétrica al procedimiento ANOVA de una vía con muestras correlacionadas descrito en el capítulo 10. Sin embargo, debemos saber que no es exactamente una extensión de esta prueba, puesto que no considera las diferencias relativas entre sujetos (como lo hace ANOVA y la prueba de rangos con signo de Wilcoxon), y en consecuencia, como señala Baguley (2012), el poder estadístico es bastante menor.

Recordemos las condiciones que se deben verificar para poder aplicar la prueba ANOVA para muestras correlacionadas:

1. La escala con que se mide la variable dependiente tiene las propiedades de una escala de intervalos iguales.
2. Las mediciones son independientes al interior de cada grupo.
3. Se puede suponer razonablemente que la(s) población(es) de origen sigue(n) una distribución normal.
4. La matriz de varianzas-covarianzas es esférica. Como explica Horn (2008, p. 1), esta condición establece que las varianzas entre los diferentes niveles de las medidas repetidas deben ser iguales.

Existen situaciones en las que no podemos comprobar que la escala de medición de la variable dependiente sea de intervalos iguales:

- Cuando las observaciones se miden en una escala logarítmica (por ejemplo, la escala de pH para medir la acidez o la escala de Richter para medir la intensidad de los sismos).
- Cuando las mediciones provienen de una escala ordinal, por ejemplo, un orden de preferencia.
- Cuando las mediciones de base provienen de una escala ordinal. Por ejemplo, cuando se suman o promedian puntajes de diversos elementos evaluados con una escala Likert.

Las condiciones requeridas por la prueba de Friedman son las siguientes:

1. La variable independiente debe ser categórica y tener a lo menos tres niveles.
2. La escala de la variable dependiente debe ser, a lo menos, ordinal.
3. Los sujetos son una muestra aleatoria e independiente de la población.

Como ejemplo para esta prueba, supongamos ahora que un equipo de desarrolladores desea establecer qué interfaz gráfica (*A*, *B* o *C*) resulta más atractiva para los usuarios de un nuevo sistema, por lo que han seleccionado una muestra aleatoria representativa de los distintos tipos de usuarios y les han solicitado evaluar 6 aspectos de cada interfaz con una escala Likert de 5 puntos, donde el valor 1 corresponde a una valoración muy negativa y 5, a una muy positiva. La tabla 11.8 muestra las puntuaciones totales asignadas por cada participante a las diferentes interfaces. En consecuencia, las hipótesis a contrastar son:

H_0 : las interfaces tienen preferencias similares.

H_A : al menos una interfaz obtiene una preferencia distinta a las demás.

El primer paso del proceso consiste en asignar rangos a las observaciones de cada sujeto. La interfaz con puntuación más baja recibe un rango de 1 y la más alta, un rango de 3 (generalizando, si se tienen *k*

Usuario	A	B	C
1	21	6	13
2	10	21	25
3	7	18	18
4	21	7	20
5	24	24	24
6	27	13	8
7	17	13	29

Tabla 11.8: evaluación realizada por los usuarios a cada una de las distintas interfaces.

observaciones pareadas, se asignan rangos con valores 1 a k). En caso de empate, se asigna el promedio de los rangos correspondientes. La tabla 11.9 muestra el resultado de este proceso.

Usuario	Originales			Rangos		
	A	B	C	A	B	C
1	21	6	13	3	1	2
2	10	21	25	1	2	3
3	7	18	18	1	2,5	2,5
4	21	7	20	3	1	2
5	24	24	24	2	2	2
6	27	13	8	3	2	1
7	17	13	29	2	1	3

Tabla 11.9: ranking de las interfaces por usuario.

La hipótesis nula para la prueba de Friedman es que, los rangos promedio de cada interfaz serán muy similares. Si denotamos el rango promedio de un grupo (interfaz) por M_x , para cada grupo esperamos, entonces, que se cumpla la igualdad de la ecuación 11.13, donde k es la cantidad de grupos.

$$M_x = \frac{k+1}{2} \quad (11.13)$$

A continuación se calculan algunas estadísticas de resumen, donde n corresponde al tamaño de cada muestra y M , a la media de los rangos (tabla 11.10).

	A	B	C	Combinada
n	7	7	7	21
M	2,14	1,64	2,21	2

Tabla 11.10: resumen de los rangos.

Con estos valores, podemos definir una medida para la variabilidad de los grupos agregados, dada por la ecuación 11.14.

$$SS_{bg(R)} = \sum_{i=1}^k n_i \cdot (M_i - M_T)^2 \quad (11.14)$$

Haciendo el cálculo para el ejemplo, tenemos:

$$SS_{bg(R)} = 7 \cdot [(2,14 - 2)^2 + (1,64 - 2)^2 + (2,21 - 2)^2] = 1,357$$

Con el resultado anterior, podemos ahora calcular el estadístico de prueba (11.15), que sigue una distribución χ^2 con $k - 1$ grados de libertad.

$$\chi^2 = \frac{SS_{bg(R)}}{\frac{k \cdot (k+1)}{12}} = \frac{12 \cdot SS_{bg(R)}}{k \cdot (k+1)} \quad (11.15)$$

Para el ejemplo:

$$\chi^2 = \frac{12 \cdot 1,357}{3 \cdot (3 + 1)} = 1,357$$

Una vez más, calculamos el valor p mediante la llamada `pchisq(1.357, 2, lower.tail = FALSE)`, obteniéndose $p = 0,507$. Considerando un nivel de significación $\alpha = 0,05$, se falla al rechazar la hipótesis nula. En consecuencia, concluimos con 95 % de confianza que no hay evidencia suficiente para descartar que las preferencias entre las distintas interfaces sean las mismas.

En este caso no es necesario realizar un procedimiento post-hoc, pues la prueba ómnibus no encontró diferencias estadísticamente significativas. No obstante, si fuese necesario, podemos efectuar una prueba de rangos con signo de Wilcoxon por cada par de grupos y aplicar algún factor de corrección.

Para hacer la prueba de Friedman en R, podemos usar la función `friedman.test(formula, data)`, donde:

- **formula:** tiene la forma <variable dependiente> \sim <variable independiente>| <identificador de sujeto o bloque>.
- **data:** matriz de datos en formato largo.

Para los procedimientos post-hoc, podemos aplicar los ajustes de Bonferroni y Holm mediante la función `pairwise.wilcox.test()`, del mismo modo descrito para la prueba de Kruskal-Wallis, cuidando en este caso que el argumento `paired` debe tomar forzosamente el valor TRUE. Si además queremos conocer el tamaño del efecto detectado para aquellos pares identificados como relevantes, debemos realizar las correspondientes pruebas de rangos con signo de Wilcoxon para todos los pares de grupos que presenten diferencias significativas (Amat Rodrigo, 2016b).

El script 11.4 muestra la realización de la prueba de Friedman para el ejemplo, cuyo resultado se presenta en la figura 11.6, e incorpora el procedimiento post-hoc de Holm por fines académicos, ya que solo debería realizarse si la prueba ómnibus encuentra diferencias significativas.

Script 11.4: prueba de Friedman y el procedimiento post-hoc de Holm para el ejemplo.

```

1 # Construir la matriz de datos.
2 A <- c(21, 10, 7, 21, 24, 27, 17)
3 B <- c(6, 21, 18, 7, 24, 13, 13)
4 C <- c(13, 25, 18, 20, 24, 8, 29)
5
6 Puntuacion <- c(A, B, C)
7
8 Interfaz <- c(rep("A", length(A)),
9                 rep("B", length(B)),
10                rep("C", length(C)))
11
12 Sujeto <- rep(1:7, 3)
13
14 Interfaz <- factor(Interfaz)
15
16 datos <- data.frame(Sujeto, Puntuacion, Interfaz)
17
18 # Establecer nivel de significación
19 alfa <- 0.05
20
```

```

21 # Hacer la prueba de Friedman.
22 prueba <- friedman.test(Puntuacion ~ Interfaz | Sujeto, data = datos)
23 print(prueba)
24
25 # Efectuar procedimiento post-hoc de Holm si se encuentran diferencias
26 # significativas.
27 if(prueba$p.value < alfa) {
28   post_hoc <- pairwise.wilcox.test(datos$Puntuacion,
29                                     datos$Interfaz,
30                                     p.adjust.method = "holm",
31                                     paired = TRUE)
32
33   print(post_hoc)
34 }

```

Friedman rank sum test

data: Puntuacion and Interfaz and Sujeto
Friedman chi-squared = 1.6522, df = 2, p-value = 0.4378

Figura 11.6: valores p obtenidos en las pruebas t para cada par de grupos mediante los métodos de Bonferroni y Holm.

Por último, debemos saber que va tomando fuerza la idea de no usar la prueba de Friedman. En reemplazo, se está recomendando transformar los datos en rangos y luego aplicar directamente el análisis de varianza sobre los datos *rankeados* (Zimmerman & Zumbo, 1993). Es más, esta idea va ganando adeptos incluso para muestras independientes y el análisis de dos muestras.

11.3 EJERCICIOS PROPUESTOS

1. ¿Qué riesgos se corren si se aplica la prueba t de Student con dos muestras que no cumplen con las suposiciones que hace esta prueba?
2. La prueba de Wilcoxon-Mann-Whitney es una alternativa no paramétrica ¿para qué versión de la prueba t de Student?
3. ¿Qué suposiciones hace la prueba de Wilcoxon-Mann-Whitney?
4. Explica cómo la prueba de Wilcoxon-Mann-Whitney construye el ranking de los datos.
5. ¿Qué estadístico usa la prueba de Wilcoxon-Mann-Whitney y cómo se calcula?
6. ¿Por qué a la prueba de Wilcoxon-Mann-Whitney también se le conoce como U-test?
7. La prueba de los rangos con signo de Wilcoxon es una alternativa no paramétrica ¿para qué versión de la prueba t de Student?
8. ¿Qué suposiciones hace la prueba de los rangos con signo de Wilcoxon?
9. Explica cómo la prueba de los rangos con signo de Wilcoxon construye el ranking de los datos.
10. ¿Qué estadístico usa la prueba de los rangos con signo de Wilcoxon y cómo se calcula?
11. ¿Cuándo es más relevante preocuparse de las violaciones de las condiciones del procedimiento ANOVA para muestras independientes?
12. Explica cómo la prueba de Kruskal-Wallis construye el ranking de los datos.
13. ¿Qué estadístico usa la prueba de Kruskal-Wallis y cómo se calcula? ¿Qué distribución sigue dicho estadístico?
14. ¿Cuál es la hipótesis nula de la prueba de Kruskal-Wallis?
15. ¿Qué suposiciones hace la prueba de Kruskal-Wallis?

16. Explique cómo la prueba de Friedman construye el ranking de los datos.
17. ¿Qué estadístico usa la prueba de Friedman y cómo se calcula? ¿Qué distribución sigue dicho estadístico?
18. ¿Cuál es la hipótesis nula de la prueba de Friedman?
19. ¿Qué suposiciones hace la prueba de Friedman?

CAPÍTULO 12. REMUESTREO

Los **métodos basados en remuestreo** son una buena alternativa a emplear cuando necesitamos inferir sobre parámetros distintos a la media o la proporción, o bien cuando no se cumplen las condiciones requeridas por las pruebas ya conocidas. Además, algunos de estos métodos son más precisos que los tradicionales. Pese a estas ventajas, los métodos basados en remuestreo realizan enormes cantidades de cómputos, por lo que en la práctica requieren de herramientas de software para su aplicación. Si bien existen métodos de remuestreo paramétricos y semiparamétricos, en este capítulo abordaremos las principales técnicas de remuestreo no paramétricas, basándonos en las ideas descritas por Amat Rodrigo (2016a) y Hesterberg et al. (2003).

12.1 BOOTSTRAPPING

A partir de lo que hemos aprendido hasta ahora, ya tenemos bastante claro que, en estadística, el ideal es contar con varias muestras grandes. Pero muchas veces solo disponemos de una muestra bastante pequeña. Sin embargo, si esta muestra es representativa de la población, esperaríamos que las observaciones que ella contiene aparecieran con frecuencias similares a las de la población. El método de **bootstrapping** se construye en torno a esta idea y, en términos generales, sigue los siguientes pasos:

1. Crear una gran cantidad B de nuevas muestras (cientos o miles) a partir de la muestra original. Cada muestra debe tener el mismo tamaño que la original y se construye mediante **muestreo con reposición**. Esto quiere decir que, al seleccionar un elemento de la muestra original, se devuelve a ella antes de tomar el siguiente, por lo que podría ser reelegido.
2. Calcular la distribución bootstrap y obtener el estadístico de interés para cada una de las muestras.
3. Usar la distribución bootstrap, la cual entrega información acerca de la forma, el centro y la variabilidad de la distribución muestral del estadístico de interés.

A los lectores atentos les habrá llamado la atención que, a diferencia de las pruebas y procedimientos anteriores, el segundo paso del método de bootstrapping habla de un **estadístico de interés** en lugar de la media o la proporción (como las pruebas estudiadas hasta ahora). Esto se debe a que, en general, puede aplicarse para casi **cualquier estadístico**.

Esta técnica, además de contrastar hipótesis, permite construir intervalos de confianza para el parámetro estimado de la población.

12.1.1 Bootstrapping para una muestra

Supongamos que la investigadora Helen Chufe desea evaluar un nuevo algoritmo de clasificación y determinar el tiempo promedio de ejecución (en milisegundos) para instancias de tamaño fijo del problema. Para ello ha realizado pruebas con 10 instancias del problema y registrado los tiempos de ejecución, presentados en la tabla 12.1. La figura 12.1 muestra la distribución del tiempo de ejecución para la muestra.

Evidentemente, la muestra es pequeña ($n = 10$) y su distribución está fuertemente desviada hacia la izquierda, por lo que Chufe ha decidido emplear bootstrapping como alternativa para enfrentar estos datos problemáticos.

Instancia	1	2	3	4	5	6	7	8	9	10
Tiempo (ms)	79	75	84	75	94	82	76	90	79	88

Tabla 12.1: tiempo de ejecución para cada instancia de la muestra.

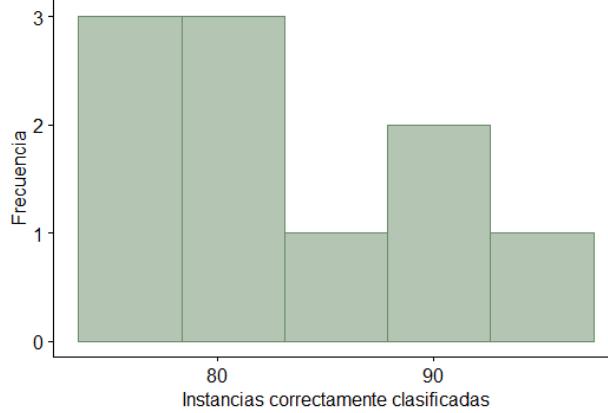


Figura 12.1: distribución del tiempo de ejecución para la muestra.

Para ilustrar el proceso paso a paso, consideremos inicialmente $B = 10$ remuestreos y calculemos la media para cada uno. La tabla 12.2 muestra en cada columna una de las muestras obtenidas, con sus respectivas medias en la última fila.

Original	Bt 1	Bt 2	Bt 3	Bt 4	Bt 5	Bt 6	Bt 7	Bt 8	Bt 9	Bt 10
79	84	84	84	94	94	94	76	82	75	79
75	94	75	82	88	75	75	88	88	82	79
84	79	82	84	90	90	94	79	79	94	94
75	79	88	79	90	82	94	76	75	94	82
94	88	79	79	90	82	76	84	75	79	84
82	75	84	79	75	76	75	75	79	76	82
76	88	82	84	75	76	79	75	90	88	94
90	88	79	79	75	82	75	79	88	88	79
79	84	79	90	94	90	88	75	75	79	75
88	75	94	88	82	88	76	94	90	82	82
82,2	83,4	82,6	82,8	85,3	83,5	82,6	80,1	82,1	83,7	83,0

Tabla 12.2: muestra original y remuestreos de bootstrap

La figura 12.2 muestra la distribución bootstrap de la media para los 10 remuestreos del ejemplo (figura 12.2a) y para 2.000 remuestreos (figura 12.2b). En ella podemos ver claramente que, a medida que la cantidad de muestras bootstrap crece, la distribución bootstrap de la media se asemeja cada vez más a la distribución normal, por lo que se acerca a la forma que esperaríamos para la distribución muestral. La figura 12.2b también nos da una idea acerca de la variabilidad de las medias de los diferentes remuestreos. Sin embargo, podemos conocer mejor esta variabilidad calculando el error estándar, dado por la ecuación 12.1, donde \bar{x}_i^* es la media del i -ésimo remuestreo y B es la cantidad de remuestreos realizados.

$$SE_{b,\bar{x}} = \sqrt{\frac{1}{B-1} \cdot \sum_{i=1}^B \left(\bar{x}_i^* - \frac{1}{B} \cdot \sum_{i=1}^B \bar{x}_i^* \right)^2} \quad (12.1)$$

Fijémonos en que la subexpresión $\frac{1}{B} \cdot \sum_{i=1}^B \bar{x}_i^*$ de la ecuación 12.1 corresponde al promedio de las medias de

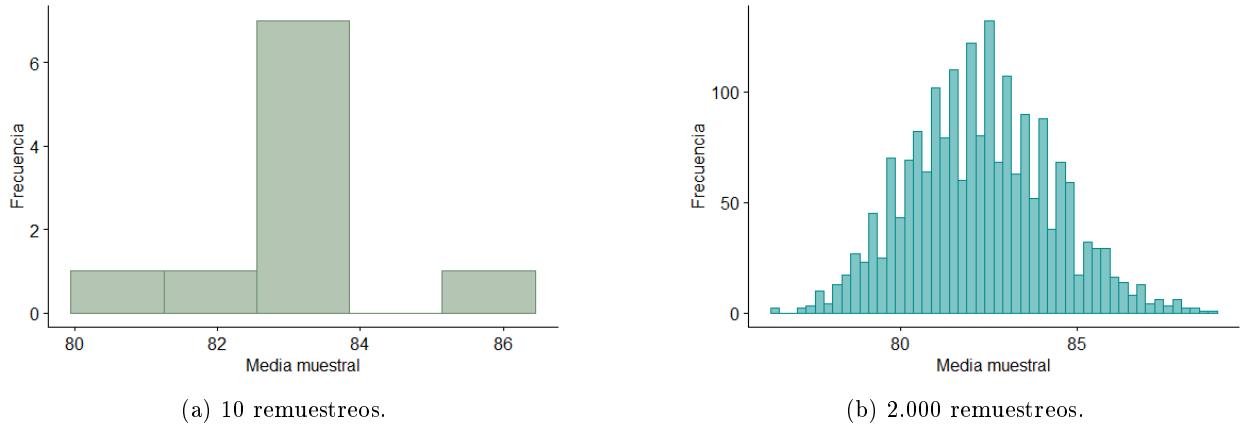


Figura 12.2: distribución bootstrap de la media

los remuestreos (media de la distribución bootstrap), por lo que la subexpresión $\sum_{i=1}^B \left(\bar{x}_i^* - \frac{1}{B} \cdot \sum_{i=1}^B \bar{x}_i^* \right)^2$ es, a su vez, la suma de las desviaciones cuadradas, con lo que resulta evidente la semejanza con el cálculo del error estándar para la media de una muestra presentado en el capítulo 4:

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$

Para el ejemplo con $B = 10$, entonces, tenemos que la media de la distribución bootstrap es:

$$\frac{1}{B} \cdot \sum_{i=1}^B \bar{x}_i^* = 83,4 + 82,6 + 82,8 + 85,3 + 83,5 + 82,6 + 80,1 + 82,1 + 83,7 + 83,0 = 82,91$$

Con lo que la suma de las desviaciones cuadradas es:

$$\begin{aligned} \sum_{i=1}^B (\bar{x}_i^* - 82,91)^2 &= (83,4 - 82,91)^2 + (82,6 - 82,91)^2 + (82,8 - 82,91)^2 + \\ &+ (85,3 - 82,91)^2 + (83,5 - 82,91)^2 + (82,6 - 82,91)^2 + (80,1 - 82,91)^2 + \\ &+ (82,1 - 82,91)^2 + (83,7 - 82,91)^2 + (83,0 - 82,91)^2 = 15,689 \end{aligned}$$

En consecuencia, el error estándar de la distribución bootstrap es:

$$SE_{B,\bar{x}} = \sqrt{\frac{1}{B-1} \cdot 15,689} = 1,320$$

Otra medida que suele emplearse para la distribución bootstrap es el **sesgo**, que indica cuánto se aleja el estadístico de interés de la muestra original (θ) de la media de la distribución bootstrap, como muestra la ecuación 12.2.

$$sesgo = \theta - \frac{1}{B} \cdot \sum_{i=1}^B \bar{x}_i^* \tag{12.2}$$

Para el ejemplo:

$$sesgo = \bar{x} - \frac{1}{B} \cdot \sum_{i=1}^B \bar{x}_i^* = 82,2 - 82,91 = -0,71$$

Ahora que ya conocemos la distribución bootstrap para la media, podemos entonces **construir un intervalo de confianza para la media de la población**, para lo que abordaremos diferentes alternativas.

Cuando la distribución bootstrap se asemeja a la normal y el sesgo es pequeño en comparación con el estimador calculado (como en este caso), podemos construir el intervalo de confianza usando la distribución t, del mismo modo que conocimos en el capítulo 4, como muestra la ecuación 12.3, usando el error estándar de la distribución bootstrap. El valor crítico de t , t^* , se obtiene para el nivel de significación establecido para el estudio y $\nu = n - 1$ grados de libertad (no olvidemos que n es el tamaño de la muestra original).

$$\theta \pm t^* \cdot SE_{B,\bar{x}} \quad (12.3)$$

Así, si consideramos para este ejemplo un nivel de significación $\alpha = 0,01$, el valor crítico de t (para dos colas) con 9 grados de libertad es $t^* = 3,25$. En consecuencia, el intervalo de confianza resultante para la media de la población es:

$$82,2 \pm 3,25 \cdot 1,320 = (77,910; 86,490)$$

Otra alternativa cuando la distribución bootstrap se asemeja a la normal, que tiene en cuenta posibles asimetrías, es construir el intervalo de confianza en base a cuantiles. En este caso, para $\alpha = 0,01$, los límites del intervalo están dados por los percentiles 1 y 99 de la distribución bootstrap:

$$(80,280; 85,156)$$

Cuando los intervalos de confianza obtenidos por ambos métodos son muy diferentes, es clara señal de que no podemos asumir que la distribución bootstrap se asemeja a la normal. En general, lo más recomendable es usar otro esquema, llamado BCa (del inglés *bias-corrected accelerated*), es decir, con sesgo corregido y acelerado. No se detalla aquí el procedimiento, pues requiere el empleo de software.

Desde luego, es inviable usar bootstrapping sin software. R ofrece el paquete `boot`, con las funciones `boot(data, statistic, R)` para generar la distribución bootstrap y `boot.ci(boot.out, conf, type)` para calcular los intervalos de confianza, donde:

- `data`: el conjunto de datos. En caso de matrices y data frames, se considera cada fila como una observación con múltiples variables.
- `statistic`: función que se aplica a los datos y devuelve un vector con el (o los) estadístico(s) de interés.
- `R`: cantidad de remuestreos bootstrap (B).
- `boot.out`: objeto de la clase `boot`, generado por la función `boot()`.
- `conf`: nivel de confianza ($1 - \alpha$).
- `type`: string o vector que indica los tipos de intervalo de confianza a construir (“`norm`” para el basado en la distribución normal, “`perc`” para el basado en los percentiles y “`bca`” para el método recomendado).

Debemos mencionar que la función `boot()` puede recibir otros muchos argumentos, los cuales escapan al alcance de los contenidos aquí expuestos. El script 12.1 construye intervalos de confianza mediante bootstrapping para el ejemplo, con $B = 2000$ y manteniendo el nivel de significación $\alpha = 0,01$. En las líneas 15–17 se construye la función para el estadístico de interés (en este caso la media), que luego usa la función `boot()` para generar la distribución bootstrap (líneas 19–20), obteniéndose el resultado que se presenta en la figura 12.3.

Podemos ver gráficamente esta distribución mediante un histograma y un gráfico Q-Q (figura 12.4), gracias a la llamada a la función `plot()` con el resultado entregado por `boot()` como argumento (línea 23).

ORDINARY NONPARAMETRIC BOOTSTRAP

```

Call:
boot(data = muestra, statistic = media, R = B)

Bootstrap Statistics :
      original   bias    std. error
t1*     82.2  0.06125    1.98329

```

Figura 12.3: distribución bootstrap generada mediante `boot()` para la media.

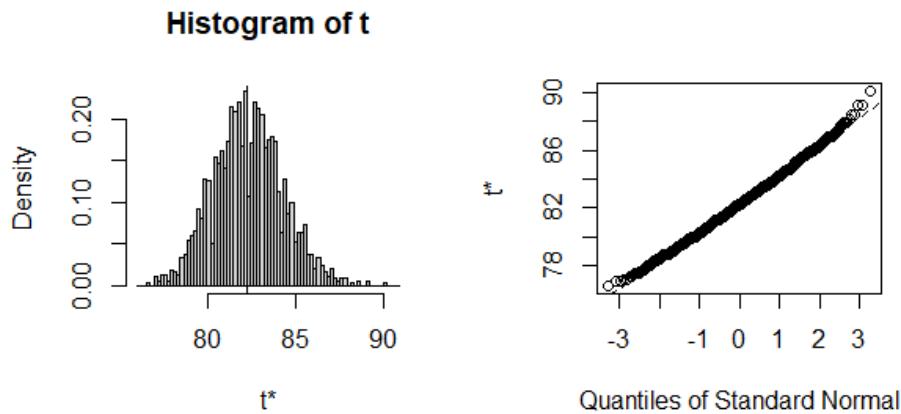


Figura 12.4: histograma y gráfico Q-Q de la distribución bootstrap generada mediante `boot()` para la media.

En las líneas 26–39 se muestra el uso de `boot.ci()` para construir los intervalos de confianza mediante diferentes métodos, obteniéndose los siguientes resultados:

- Intervalo de confianza usando aproximación normal: (77,03; 87,25).
- Intervalo de confianza usando percentiles: (77,4; 87,7).
- Intervalo de confianza BCa: (77,48; 87,90).

Las líneas 42–45 muestran otra alternativa para construir la distribución bootstrap por medio del paquete `bootES`, que ofrece la función `bootES(data, R, ci.type, ci.conf, plot, ...)`. Esta función realiza internamente una llamada a la función `boot()` descrita en los párrafos precedentes, pero no requiere implementar previamente la función para el cálculo de la media. Debemos tener en cuenta que aquí solo se muestran algunos de los argumentos, a saber:

- `data`: conjunto de datos.
- `R`: cantidad de remuestreos bootstrap (B).
- `ci.type`: tipo de intervalo de confianza a construir (opcional), con las mismas opciones descritas para `boot.ci()`.
- `ci.conf`: nivel de significación para el intervalo de confianza (opcional, por defecto 0.95).
- `plot`: por defecto con valor `FALSE`, cuando es `TRUE` genera una figura con el histograma y el gráfico Q-Q de la ditribución bootstrap.
- `...`: permite pasar otros argumentos para la función `boot()` subyacente.

Las figuras 12.5 y 12.6 muestran los resultados obtenidos, ligeramente diferentes a los anteriores. A partir de

estos últimos podemos concluir que tenemos 95 % de confianza de que el algoritmo tarda entre 77,48 ms y 87,90 ms en ejecutar las instancias del tamaño seleccionado.

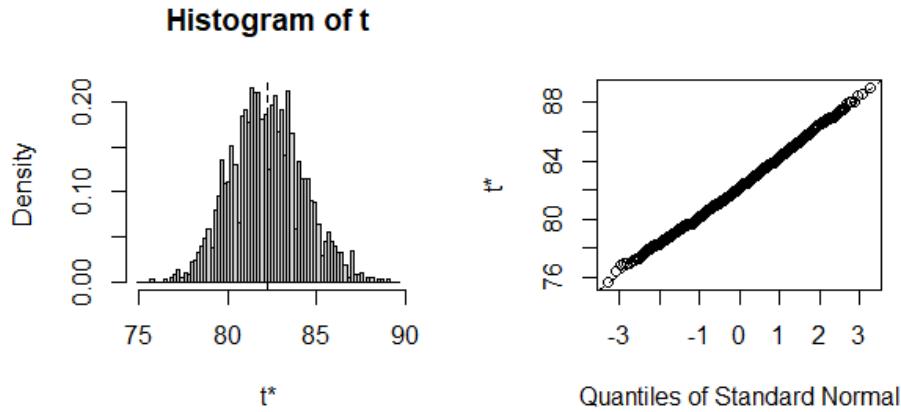


Figura 12.5: histograma y gráfico Q-Q de la distribución bootstrap generada mediante `bootES()` para la media.

99.00% bca Confidence Interval, 2000 replicates				
Stat	CI (Low)	CI (High)	bias	SE
82.200	77.482	87.900	0.061	1.983

Figura 12.6: distribución bootstrap e intervalo de confianza para la media de la población generada mediante `bootES()`.

Script 12.1: construcción de un intervalo de confianza para la media poblacional mediante bootstrapping.

```

1 library(boot)
2 library(bootES)
3
4 # Crear muestra inicial, mostrar su histograma y calcular la media.
5 muestra <- c(79, 75, 84, 75, 94, 82, 76, 90, 79, 88)
6 datos <- data.frame(muestra)
7
8 # Establecer cantidad de remuestreos y nivel de significación.
9 B = 2000
10 alfa <- 0.01
11
12 cat("Paquete boot\n")
13
14 # Construir distribución bootstrap usando el paquete boot.
15 media <- function(valores, i) {
16   mean(valores[i])
17 }
18
19 set.seed(432)
20 distribucion_b <- boot(muestra, statistic = media, R = B)
21 print(distribucion_b)
22
23 # Graficar distribución bootstrap.
24 print(plot(distribucion_b))
25

```

```

26 # Construir intervalos de confianza.
27 intervalo_t <- boot.ci(distribucion_b, conf = 1 - alfa, type = "norm")
28
29 cat("\n\nIntervalo de confianza usando distribución t:\n")
30 print(intervalo_t)
31
32 intervalo_perc <- boot.ci(distribucion_b, conf = 1 - alfa, type = "perc")
33
34 cat("\n\nIntervalo de confianza usando percentiles:\n")
35 print(intervalo_perc)
36
37 intervalo_bca <- boot.ci(distribucion_b, conf = 1 - alfa, type = "bca")
38
39 cat("\n\nIntervalo de confianza BCa:\n")
40 print(intervalo_bca)
41
42 # Construir distribución bootstrap usando el paquete bootES.
43 set.seed(432)
44
45 distribucion_bootstrapES <- bootES(muestra, R = B, ci.type = "bca",
46                                     ci.conf = 1 - alfa, plot = TRUE)
47
48 print(distribucion_bootstrapES)

```

Supongamos ahora que Helen desea hacer una prueba de hipótesis para ver si el tiempo promedio de ejecución del algoritmo para instancias del tamaño seleccionado es mayor a 75 milisegundos. Así, tenemos que:

Denotando como μ al tiempo medio que tarda el algoritmo de Helen para resolver instancias de tamaño fijo del problema, entonces:

$$H_0: \mu = 75 \text{ [ms]}$$

$$H_A: \mu > 75 \text{ [ms]}$$

El contraste de hipótesis requiere siempre generar la distribución centrada en el valor nulo para, a partir de ella, obtener el valor p. Sabemos que la distribución bootstrap se centra *alrededor* del valor observado, por lo que debemos desplazarla para cumplir con esta condición. Para lograrlo, simplemente necesitamos restar a cada observación de la distribución bootstrap la diferencia entre su valor promedio y el valor nulo.

Para calcular el valor p, seguimos la fórmula señalada en la ecuación 12.4, donde:

- r : cantidad de observaciones en la distribución bootstrap (desplazada) a lo menos tan extremas como el estadístico observado.
- B : cantidad de repeticiones bootstrap consideradas en la simulación.

$$p = \frac{r + 1}{B + 1} \tag{12.4}$$

Tras hacer la prueba (script 12.2), obtenemos que $p = 0,001$, menor que el nivel de significación, por lo que la evidencia es suficientemente fuerte para rechazar la hipótesis nula en favor de la hipótesis alternativa. En consecuencia, concluimos con 99,5 % de confianza que el tiempo de ejecución promedio del algoritmo para instancias del tamaño seleccionado supera los 75 milisegundos.

Script 12.2: inferencia sobre la media de una muestra con bootstrapping.

```
1 library(boot)
2
3 set.seed(432)
4
5 # Crear muestra inicial, mostrar su histograma y calcular la media.
6 muestra <- c(79, 75, 84, 75, 94, 82, 76, 90, 79, 88)
7 valor_observado <- mean(muestra)
8 datos <- data.frame(muestra)
9
10 # Construir distribución bootstrap.
11 B <- 2000
12
13 media <- function(valores, i) {
14   mean(valores[i])
15 }
16
17 distribucion_b <- boot(muestra, statistic = media, R = B)
18
19 # Desplazar la distribución bootstrap para que se centre en
20 # el valor nulo.
21 valor_nulo <- 75
22 desplazamiento <- mean(distribucion_b[["t"]]) - valor_nulo
23 distribucion_nula <- distribucion_b[["t"]] - desplazamiento
24
25 # Determinar el valor p.
26 p <- (sum(distribucion_nula > valor_observado) + 1) / (B + 1)
27 cat("Valor p:", p)
```

12.1.2 Bootstrapping para dos muestras independientes

El proceso para comparar dos poblaciones mediante bootstraping es similar al que ya conocimos para una única población. Si tenemos dos muestras independientes A y B provenientes de dos poblaciones diferentes, de tamaños n_A y n_B respectivamente, los pasos a seguir son:

1. Fijar la cantidad B de repeticiones bootstrap.
2. En cada repetición, hacer un remuestreo con reposición de tamaño n_A a partir de la muestra A y otro de tamaño n_B a partir de la muestra B .
3. En cada repetición, calcular el estadístico de interés para generar la distribución bootstrap.
4. Construir el intervalo de confianza para el estadístico de interés.

Supongamos que una Universidad desea estudiar la diferencia entre las calificaciones finales de hombres y mujeres que rinden una asignatura inicial de programación por primera vez. Para ello, disponen de las notas (en escala de 1,0 a 7,0) de 27 hombres y 19 mujeres:

- Hombres: 1,3; 1,5; 1,6; 1,7; 1,7; 1,9; 2,3; 2,4; 2,6; 2,6; 2,6; 2,7; 2,8; 3,2; 3,7; 4,1; 4,4; 4,5; 4,8; 5,2; 5,2; 5,3; 5,5; 5,5; 5,6; 5,6; 5,7; 5,7
- Mujeres: 3,5; 3,6; 3,8; 4,3; 4,5; 4,5; 4,9; 5,1; 5,3; 5,3; 5,5; 5,8; 6,0; 6,3; 6,3; 6,4; 6,4; 6,6; 6,7

Tras aplicar pruebas de Shapiro-Wilk (figura 12.7), los investigadores han comprobado que las notas de los varones no siguen una distribución normal, por lo que han decidido usar bootstrapping para la prueba de hipótesis, con un nivel de significación $\alpha = 0,05$ y $B = 9999$ repeticiones.

```

Shapiro-Wilk normality test

data: hombres
W = 0.88357, p-value = 0.005742

Shapiro-Wilk normality test

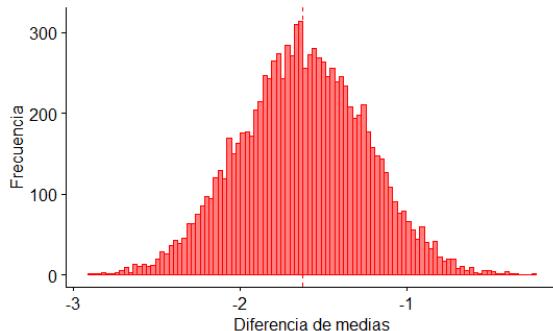
data: mujeres
W = 0.93022, p-value = 0.1748

```

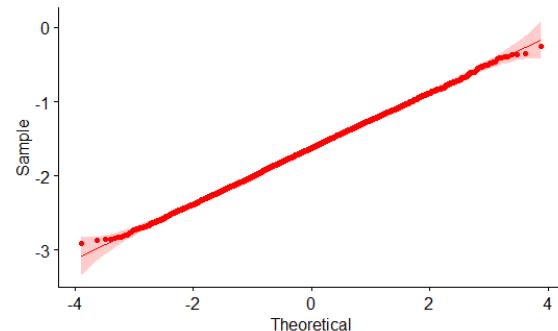
Figura 12.7: pruebas de normalidad de Shapiro-Wilk para ambas muestras.

La media observada (en la muestra original) para la calificación final de las mujeres es $\bar{x}_m = 5,305$, mientras que para los hombres es $\bar{x}_h = 3,670$. Así, la diferencia observada es $\bar{x}_h - \bar{x}_m = -1,635$.

La distribución bootstrap de la diferencia de medias se asemeja a la normal (figura 12.8), con media $\bar{x} = -1,628$ y desviación estándar $s = 0,377$.



(a) histograma con la media añadida.



(b) gráfico Q-Q.

Figura 12.8: distribución bootstrap de la diferencia de medias.

Al construir el intervalo de confianza mediante el método BCa para la distribución bootstrap, R nos entrega como resultado el intervalo $(-2,372; -0,894)$. En consecuencia, concluimos con 95 % de confianza que no es posible descartar que la diferencia en la calificación final entre hombres y mujeres es de 1,5 puntos.

El script 12.3 muestra el desarrollo de este ejemplo en R, el cual usa la función `two.boot(sample1, sample2, FUN, R)` del paquete `simpleboot`, donde:

- `sample1, sample2`: muestras originales.
- `FUN`: función que, para cada muestra, calcula el estadístico de interés θ .
- `R`: cantidad de remuestreos con repetición.

Esta función opera generando remuestreos para cada una de las muestras originales, y calculando en cada iteración el estadístico $\theta_1 - \theta_2$, donde los subíndices señalan la muestra correspondiente.

Script 12.3: bootstrapping para la diferencia de medias.

```

1 library(simpleboot)
2 library(boot)
3 library(ggpubr)
4
5 set.seed(432)
6
7 # Ingresar datos originales

```

```

8 hombres <- c(1.3, 1.5, 1.6, 1.7, 1.7, 1.9, 2.3, 2.4, 2.6, 2.6, 2.7,
9           2.8, 3.2, 3.7, 4.1, 4.4, 4.5, 4.8, 5.2, 5.2, 5.3, 5.5,
10          5.5, 5.6, 5.6, 5.7, 5.7)
11
12 mujeres <- c(3.5, 3.6, 3.8, 4.3, 4.5, 4.5, 4.9, 5.1, 5.3, 5.3, 5.5,
13           5.8, 6.0, 6.3, 6.3, 6.4, 6.4, 6.6, 6.7)
14
15 n_hombres <- length(hombres)
16 n_mujeres <- length(mujeres)
17
18 sexo <- c(rep("Hombre", n_hombres), rep("Mujer", n_mujeres))
19 nota <- c(hombres, mujeres)
20 datos <- data.frame(nota, sexo)
21
22 # Comprobar normalidad de las muestras.
23 print(shapiro.test(hombres))
24 print(shapiro.test(mujeres))
25
26 # Calcular la diferencia observada entre las medias muestrales.
27 media_hombres <- mean(hombres)
28 media_mujeres <- mean(mujeres)
29 diferencia_observada <- media_hombres - media_mujeres
30
31 cat("diferencia observada:", media_hombres - media_mujeres, "\n\n")
32
33 # Establecer el nivel de significación.
34 alfa <- 0.05
35
36 # Crear la distribución bootstrap.
37 B <- 9999
38 distribucion_bootstrap <- two.boot(hombres, mujeres, FUN = mean, R = B)
39
40 # Examinar la distribución bootstrap.
41 valores <- data.frame(distribucion_bootstrap$t)
42 colnames(valores) <- "valores"
43
44 histograma <- gghistogram(valores, x = "valores", color = "red",
45                           fill = "red", bins = 100,
46                           xlab = "Diferencia de medias",
47                           ylab = "Frecuencia", add = "mean")
48
49 print(histograma)
50
51 qq <- ggqqplot(valores, x = "valores", color = "red")
52 print(qq)
53
54 cat("Distribución bootstrap:\n")
55 cat("\tMedia:", mean(valores$valores), "\n")
56 cat("\tDesviación estándar:", sd(valores$valores), "\n\n")
57
58 # Construir el intervalo de confianza.
59 intervalo_bca <- boot.ci(distribucion_bootstrap, conf = 1 - alfa,
60                           type = "bca")
61
62 print(intervalo_bca)

```

Supongamos ahora que el estudio del ejemplo desea determinar, con un nivel de significación $\alpha = 0,05$, si la diferencia entre las calificaciones finales de hombres y mujeres es igual a 1,5 puntos. Para ello, formulamos las siguientes hipótesis:

Sean μ_h y μ_m las calificaciones finales de hombres y mujeres, respectivamente, que rinden una asignatura inicial de programación por primera vez en la Universidad en estudio, entonces:

$$H_0: \mu_h - \mu_m = 1,5$$

$$H_A: \mu_h - \mu_m \neq 1,5$$

Tras aplicar bootstrapping para la prueba de hipótesis (script 12.4), obtenemos un valor p de $p = 0,364$, superior al nivel de significación, por lo que fallamos al rechazar la hipótesis nula. En consecuencia, concluimos con 95 % de confianza que la diferencia en la calificación final entre hombres y mujeres es de 1,5 puntos.

Script 12.4: bootstrapping para inferir acerca de la diferencia de medias.

```

1 library(simpleboot)
2 library(boot)
3 library(ggpubr)
4
5 set.seed(432)
6
7 # Ingresar datos originales
8 hombres <- c(1.3, 1.5, 1.6, 1.7, 1.7, 1.9, 2.3, 2.4, 2.6, 2.6, 2.7,
9           2.8, 3.2, 3.7, 4.1, 4.4, 4.5, 4.8, 5.2, 5.2, 5.3, 5.5,
10          5.5, 5.6, 5.6, 5.7, 5.7)
11
12 mujeres <- c(3.5, 3.6, 3.8, 4.3, 4.5, 4.5, 4.9, 5.1, 5.3, 5.3, 5.5,
13            5.8, 6.0, 6.3, 6.3, 6.4, 6.4, 6.6, 6.7)
14
15 n_hombres <- length(hombres)
16 n_mujeres <- length(mujeres)
17
18 sexo <- c(rep("Hombre", n_hombres), rep("Mujer", n_mujeres))
19 nota <- c(hombres, mujeres)
20 datos <- data.frame(nota, sexo)
21
22 # Calcular la diferencia observada entre las medias muestrales.
23 media_hombres <- mean(hombres)
24 media_mujeres <- mean(mujeres)
25 valor_observado <- media_hombres - media_mujeres
26
27 # Crear la distribución bootstrap.
28 B <- 9999
29 valor_nulo <- 1.5
30 distribucion_bootstrap <- two.boot(hombres, mujeres, FUN = mean, R = B)
31 desplazamiento <- mean(distribucion_bootstrap[["t"]]) - valor_nulo
32 distribucion_nula <- distribucion_bootstrap[["t"]] - desplazamiento
33
34 # Determinar el valor p.
35 p <- (sum(abs(distribucion_nula) > abs(valor_observado)) + 1) / (B + 1)
36 cat("Valor p:", p)

```

12.1.3 Bootstrapping para dos muestras pareadas

En este caso, el procedimiento resulta muy sencillo. A partir de las dos muestras originales, se crea una nueva muestra con la diferencia entre ambas, y luego se realiza el proceso especificado para la construcción de un intervalo de confianza que ya conocimos para el caso de una única muestra.

Supongamos ahora, que la Universidad del ejemplo anterior desea saber si existe diferencia entre las calificaciones obtenidas en la primera y la segunda prueba de un curso inicial de programación. Para ello, dispone de las calificaciones (en escala de 1,0 a 7,0) obtenidas en ambas pruebas para una muestra de 20 estudiantes, como muestra la tabla 12.3. Han decidido llevar a cabo el estudio mediante bootstrapping con $B = 3999$ repeticiones y un nivel de significación $\alpha = 0,05$, para lo cual han creado en R el script 12.5, obteniendo los resultados que se presentan en la figura 12.9.

Alumno	Prueba 1	Prueba 2
1	3,5	5,2
2	2,7	5,1
3	1,0	5,9
4	1,8	4,8
5	1,6	1,4
6	4,3	2,3
7	5,8	6,8
8	6,4	5,3
9	3,9	3,1
10	4,3	3,8
11	3,4	4,6
12	5,3	1,2
13	5,8	3,9
14	5,3	2,0
15	2,0	1,7
16	1,3	3,3
17	4,0	6,0
18	5,3	4,8
19	1,6	6,9
20	3,6	1,3

Tabla 12.3: calificaciones de los estudiantes en la primera y la segunda prueba de un curso inicial de programación.

```
95.00% bca Confidence Interval, 3999 replicates
Stat          CI (Low)      CI (High)    bias       SE
0.325        -0.656       1.439       0.001     0.541
```

Figura 12.9: intervalo de confianza BCa para la media de las diferencias.

A partir del resultado anterior, concluimos con 95 % de confianza que, en promedio, la diferencia de las medias para las calificaciones de la primera y la segunda evaluación se encuentra en el intervalo $(-0,656; 1,439)$, por lo que no podemos desechar la idea de que no existe una diferencia estadísticamente significativa entre ellas..

Script 12.5: bootstrapping para la media de las diferencias.

```
1 library(bootES)
2
3 set.seed(432)
4
5 # Ingresar datos originales.
6 alumno <- 1:20
7
8 prueba_1 <- c(3.5, 2.7, 1.0, 1.8, 1.6, 4.3, 5.8, 6.4, 3.9, 4.3, 3.4,
9           5.3, 5.8, 5.3, 2.0, 1.3, 4.0, 5.3, 1.6, 3.6)
10
11 prueba_2 <- c(5.2, 5.1, 5.9, 4.8, 1.4, 2.3, 6.8, 5.3, 3.1, 3.8, 4.6,
12           1.2, 3.9, 2.0, 1.7, 3.3, 6.0, 4.8, 6.9, 1.3)
```

```

13
14 # Establecer nivel de significación.
15 alfa <- 0.05
16
17 # Calcular la diferencia entre ambas observaciones.
18 diferencia <- prueba_2 - prueba_1
19
20 # Generar la distribución bootstrap y su intervalo de confianza.
21 B <- 3999
22
23 distribucion_bootstrapES <- bootES(diferencia, R = B, ci.type = "bca",
24                                     ci.conf = 1 - alfa, plot = FALSE)
25
26 print(distribucion_bootstrapES)

```

Ahora la Universidad del ejemplo desea saber si la diferencia entre las calificaciones obtenidas en la primera y la segunda prueba de un curso inicial de programación es de 5 décimas. Así, considerando un nivel de significación $\alpha = 0,05$, los investigadores formulan las siguientes hipótesis:

$$H_0: \mu_{dif} = 0,5$$

$$H_A: \mu_{dif} \neq 0,5$$

Tras efectuar la prueba de hipótesis mediante bootstrapping (script 12.6) obtienen un valor p de $p = 0,573$, por lo que la evidencia no es suficientemente fuerte como para rechazar la hipótesis nula. En consecuencia, los investigadores concluyen con 95 % de confianza que no hay razones para no pensar que la diferencia de las calificaciones obtenidas en ambas evaluaciones es de 5 décimas.

Script 12.6: bootstrapping para inferir acerca de la media de las diferencias.

```

1 library(bootES)
2
3 set.seed(432)
4
5 # Ingresar datos originales.
6 alumno <- 1:20
7
8 prueba_1 <- c(3.5, 2.7, 1.0, 1.8, 1.6, 4.3, 5.8, 6.4, 3.9, 4.3, 3.4,
9           5.3, 5.8, 5.3, 2.0, 1.3, 4.0, 5.3, 1.6, 3.6)
10
11 prueba_2 <- c(5.2, 5.1, 5.9, 4.8, 1.4, 2.3, 6.8, 5.3, 3.1, 3.8, 4.6,
12           1.2, 3.9, 2.0, 1.7, 3.3, 6.0, 4.8, 6.9, 1.3)
13
14 # Establecer nivel de significación.
15 alfa <- 0.05
16
17 # Calcular la diferencia entre ambas observaciones.
18 diferencia <- prueba_2 - prueba_1
19
20 # Calcular la media observada de las diferencias.
21 valor_observado <- mean(diferencia)
22
23 # Generar la distribución bootstrap y su intervalo de confianza.
24 B <- 3999
25 valor_nulo <- 0.5
26
27 distribucion_bootstrapES <- bootES(diferencia, R = B, ci.type = "bca",
28                                     ci.conf = 1 - alfa, plot = FALSE)
29
30 distribucion_nula <- distribucion_bootstrapES[["t"]] - valor_nulo
31

```

```

32
33 # Determinar el valor p.
34 p <- (sum(abs(distribucion_nula) > abs(valor_observado)) + 1) / (B + 1)
35 cat("Valor p:", p)

```

12.1.4 Bootstrapping para comparar medias de más de dos muestras independientes

World Egg Alliance (WEA), importante empresa avícola, trabaja con 3 razas de gallinas: Castellana, Leghorn y Marans. La sede chilena de dicha empresa cuenta con más de 500 ejemplares de cada raza. El jefe de producción requiere determinar si alguna de estas especies produce huevos con un peso promedio diferente al de las demás. Para ello, ha recolectado 18 huevos, cada uno de una gallina diferente seleccionada de manera aleatoria, y registrado su peso (en gramos), como muestra la tabla 12.4.

Castellana	Leghorn	Marans
71,61	57,97	56,45
59,49	59,88	51,00
52,05	59,15	53,73
62,14	59,08	57,87
65,77	51,58	56,32
70,25	58,94	
67,66		

Tabla 12.4: pesos de los huevos (en gramos).

Dado que las muestras por cada raza de gallinas tienen tamaños diferentes y que no todas cumplen el supuesto de normalidad, como muestra el resultado de las pruebas de Shapiro-Wilk (Figura 12.10), el jefe ha decidido usar bootstrapping para efectuar el estudio, con un nivel de significación de 0,05 y 3999 repeticiones.

Las hipótesis a contrastar para el procedimiento ómnibus son:

H_0 : No existe diferencia en el peso promedio de los huevos puestos por gallinas Castellanas, Leghorn y Marans.

H_A : Los huevos de al menos una raza de gallinas (Castellanas, Leghorn y Marans) tienen, en promedio, un peso diferente.

Como resultado, ha obtenido un valor p de 0,015, menor que el nivel de significación. Así, concluye con 95 % de confianza que el peso promedio de los huevos es significativamente diferente para al menos una de las razas de gallinas.

Para determinar entre qué razas existen diferencias, efectúa un procedimiento post-hoc para comparar las diferencias de medias entre pares de razas, ajustando los valores p con el método de Benjamini y Hochberg. La Figura 12.11 muestra los resultados obtenidos.

El jefe concluye, con 95 % de confianza, que existen diferencias significativas entre el peso de los huevos que ponen las gallinas Castellana y Marans. Al estudiar las diferencias observadas, puede ver que los huevos de las gallinas Castellana tienen mayor peso.

El script 12.7 corresponde a la solución desarrollada por el jefe.

Script 12.7: prueba de permutaciones para muestras correlacionadas.

```
1 LIBRARY (DPLYR)
```

```

datos$Raza: Castellana

Shapiro-Wilk normality test

data: dd[x, ]
W = 0.94057, p-value = 0.6438

-----
datos$Raza: Leghorn

Shapiro-Wilk normality test

data: dd[x, ]
W = 0.67504, p-value = 0.003338

-----
datos$Raza: Marans

Shapiro-Wilk normality test

data: dd[x, ]
W = 0.91937, p-value = 0.5259

```

Figura 12.10: pruebas de normalidad para las muestras del ejemplo.

```

2 LIBRARY(EZ)
3 LIBRARY(GGPUBR)
4
5 # FIJAR UNA SEMILLA PARA ASEGURAR REPLICABILIDAD.
6 SET.SEED(123)
7
8 # CONSTRUIR LA MATRIZ DE DATOS.
9 PESO <- c(71.61, 59.49, 52.05, 62.14, 65.77, 70.25, 67.66, 57.97, 59.88, 59.15,
10      59.08, 51.58, 58.94, 56.45, 51.00, 53.73, 57.87, 56.32)
11
12 RAZA <- FACTOR(c(rep("CASTELLANA", 7), rep("LEGHORN", 6), rep("MARANS", 5)))
13 DATOS <- DATA.FRAME(RAZA, PESO)
14
15 # ESTABLECER NIVEL DE SIGNIFICACIÓN.

```

Resultado del procedimiento post-hoc:

Diferencia observada entre pares de razas:

Castellana - Leghorn: 6
 Castellana - Marans: 9
 Leghorn - Marans: 3

P-valores entre pares de razas:

Castellana - Leghorn: 0.057
 Castellana - Marans: 0.012
 Leghorn - Marans: 0.409

Figura 12.11: procedimiento post-hoc para el ejemplo.

```

16 ALFA <- 0.05
17
18 # VERIFICAR CONDICIÓN DE NORMALIDAD.
19 NORMALIDAD <- BY(DATA = DATOS$PESO, INDICES = DATOS$RAZA, FUN = SHAPIRO.TEST)
20 PRINT(NORMALIDAD)
21
22 # SEPARAR ÍNDICES DE CADA RAZA PARA MANTENER LOS TAMAÑOS DE LAS MUESTRAS AL
23 # HACER REMUESTREO.
24 IND_CAS <- WHICH(DATOS$RAZA == "CASTELLANA")
25 IND_LEG <- WHICH(DATOS$RAZA == "LEGHORN")
26 IND_MAR <- WHICH(DATOS$RAZA == "MARANS")
27
28
29
30 # FUNCIÓN QUE CALCULA EL ESTADÍSTICO F USADO POR ANOVA PARA MUESTRAS
31 # INDEPENDIENTES.
32 # ENTRADAS: DATAFRAME, ÍNDICES DE LAS OBSERVACIONES PERTENECIENTES A CADA
33 # UNA DE LAS TRES MUESTRAS.
34 # SALIDA: ESTADÍSTICO F.
35 CALCULAR_F <- FUNCTION(DF, IND_1, IND_2, IND_3) {
36   INDICES <- c(IND_1, IND_2, IND_3)
37   IDS <- FACTOR(1:LENGTH(INDICES))
38   DF_NUEVO <- CBIND(ID = IDS, DF[INDICES, ])
39   PRUEBA <- EZANOVA(DF_NUEVO, PESO, ID, BETWEEN = RAZA, TYPE = 2)
40   RETURN(PRUEBA$ANOVA$F)
41 }
42
43
44
45 # CALCULAR ESTADÍSTICO PARA LOS DATOS ORIGINALES.
46 F_OBSERVADO <- SUPPRESSWarnings(CALCULAR_F(DATOS, IND_CAS, IND_LEG, IND_MAR))
47 CAT("VALOR OBSERVADO: F =", F_OBSERVADO, "\n")
48
49 # CABE RECORDAR QUE AL REMUESTREAR CON REEMPLAZO LA MUESTRA ORIGINAL,
50 # LAS REMUESTRAS NO REPRESENTAN LA HIPÓTESIS NULA. CON DOS MUESTRAS,
51 # BASTA CON "RECENTRAR" LA DISTRIBUCIÓN BOOTSTRAP EN EL VALOR
52 # NULO, PERO ESTE PROCEDIMIENTO ES DIFERENTE PARA EL CASO DEL ESTADÍSTICO F.
53 # SI LA HIPÓTESIS NULA SE CUMPLE, LOS TRES GRUPOS TIENEN LAS MISMAS MEDIAS Y
54 # VARIANZAS. ES MÁS FÁCIL HACER ESTOS AJUSTES *ANTES* DEL REMUESTREO (VER:
55 # - FISHER, N. I., & HALL, P. (1990). ON BOOTSTRAP HYPOTHESIS TESTING.
56 # AUSTRALIAN JOURNAL OF STATISTICS, 32(2), 177-190.
57 # - HALL, P., & WILSON, S. R. (1991). TWO GUIDELINES FOR BOOTSTRAP HY-
58 # POSESIS TESTING. BIOMETRICS, 757-762.
59 # - MARTIN, M. A. (2007). BOOTSTRAP HYPOTHESIS TESTING FOR SOME COMMON
60 # STATISTICAL PROBLEMS: A CRITICAL EVALUATION OF SIZE AND POWER PROP-
61 # ERTIES. COMPUTATIONAL STATISTICS & DATA ANALYSIS, 51(12), 6321-6342.)
62
63 # CALCULAR MEDIDAS DE TENDENCIA CENTRAL GENERALES (AGRUPADAS).
64 MEDIA_GENERAL <- MEAN(DATOS$PESO)
65 SD_GENERAL <- SD(DATOS$PESO)
66
67 # CALCULAR MEDIDAS DE TENDENCIA CENTRAL POR GRUPO.
68 GRUPOS <- DATOS %>% GROUP_BY(RAZA) %>%
69   SUMMARISE(MEDIA = MEAN(PESO), SD = SD(PESO)) %>%
70   AS.DATA.FRAME()
71
72 # DESPLAZAR LOS VALORES ORIGINALES PARA QUE LOS TRES GRUPOS TENGAN
73 # LA MISMA MEDIA E IGUAL DESVIACIÓN ESTÁNDAR.
74 DATOS_CENTRADOS <- DATOS

```

```

75
76 DATOS_CENTRADOS[IND_CAS, "PESO"] <- MEDIA_GENERAL +
77   (DATOS_CENTRADOS[IND_CAS, "PESO"] - GRUPOS[1, "MEDIA"]) *
78   (SD_GENERAL / GRUPOS[1, "SD"])
79
80 DATOS_CENTRADOS[IND_LEG, "PESO"] <- MEDIA_GENERAL +
81   (DATOS_CENTRADOS[IND_LEG, "PESO"] - GRUPOS[2, "MEDIA"]) *
82   (SD_GENERAL / GRUPOS[2, "SD"])
83
84 DATOS_CENTRADOS[IND_MAR, "PESO"] <- MEDIA_GENERAL +
85   (DATOS_CENTRADOS[IND_MAR, "PESO"] - GRUPOS[3, "MEDIA"]) *
86   (SD_GENERAL / GRUPOS[3, "SD"])
87
88 # REMUESTREAR LOS DIFERENTES GRUPOS.
89 B <- 3999
90 REM_CAS <- LAPPLY(1:B, FUNCTION(i) SAMPLE(IND_CAS, REPLACE = TRUE))
91 REM_LEG <- LAPPLY(1:B, FUNCTION(i) SAMPLE(IND_LEG, REPLACE = TRUE))
92 REM_MAR <- LAPPLY(1:B, FUNCTION(i) SAMPLE(IND_MAR, REPLACE = TRUE))
93
94 # OBTENER DISTRIBUCIÓN BOOTSTRAP.
95 OBTENER_F_BOOT <- FUNCTION(i, DF) {
96   SUPPRESSWARNINGS(CALCULAR_F(DF, REM_CAS[[i]], REM_LEG[[i]], REM_MAR[[i]]))
97 }
98
99 DISTRIBUCION <- SAPPLY(1:B, FUNCTION(i) OBTENER_F_BOOT(i, DATOS_CENTRADOS))
100
101 # CALCULAR EL VALOR P PARA UNA PRUEBA BILATERAL.
102 NUMERADOR <- SUM(ABS(DISTRIBUCION) > ABS(F_OBSERVADO))
103 VALOR_P <- (NUMERADOR + 1) / (B + 1)
104 CAT("RESULTADO PROCEDIMIENTO ÓMNIBUS:\n")
105 CAT("P =", ROUND(VALOR_P, 3), "\n")
106
107
108
109 # SI EL VALOR P ES MENOR QUE EL NIVEL DE SIGNIFICACIÓN, ES NECESARIO HACER UN
110 # PROCEDIMIENTO POST-HOC.
111 IF(VALOR_P < ALFA) {
112   CAT("RESULTADO DEL PROCEDIMIENTO POST-HOC:\n")
113   # SE EFECTÚAN COMPARACIONES ENTRE PARES DE RAZAS. SE PUEDEN USAR LAS MISMAS
114   # REMUESTRAS OBTENIDAS PARA EL PROCEDIMIENTO ÓMNIBUS. EN ESTE CASO, COMO SE
115   # TRATA DE MEDIAS DE MUESTRAS INDEPENDIENTES, EL ESTADÍSTICO PERTINENTE ES
116   # LA DIFERENCIA DE LAS MEDIAS.
117
118 # FUNCIÓN QUE CALCULA LA DIFERENCIA DE MEDIAS.
119 CALCULAR_DIFERENCIA_MEDIAS <- FUNCTION(DF, IND_1, IND_2) {
120   MEDIA_1 <- MEAN(DF[IND_1, "PESO"])
121   MEDIA_2 <- MEAN(DF[IND_2, "PESO"])
122   RETURN(MEDIA_1 - MEDIA_2)
123 }
124
125 # CALCULAR DIFERENCIAS OBSERVADAS ENTRE CADA PAR DE RAZAS.
126 DIF_OBS_CAS_LEG <- CALCULAR_DIFERENCIA_MEDIAS(DATOS, IND_CAS, IND_LEG)
127 DIF_OBS_CAS_MAR <- CALCULAR_DIFERENCIA_MEDIAS(DATOS, IND_CAS, IND_MAR)
128 DIF_OBS_LEG_MAR <- CALCULAR_DIFERENCIA_MEDIAS(DATOS, IND_LEG, IND_MAR)
129 CAT("\n")
130 CAT("DIFERENCIA OBSERVADA ENTRE PARES DE RAZAS:\n")
131 CAT("CASTELLANA - LEGHORN:", ROUND(DIF_OBS_CAS_LEG), "\n")
132 CAT("CASTELLANA - MARANS:", ROUND(DIF_OBS_CAS_MAR), "\n")
133 CAT("LEGHORN - MARANS:", ROUND(DIF_OBS_LEG_MAR), "\n")

```

```

134
135 # OBTENER DISTRIBUCIONES BOOTSTRAP PARA CADA DIFERENCIA.
136 D_DIF_CAS_LEG <- SAPPY(1:B,
137   FUNCTION(i) CALCULAR_DIFERENCIA_MEDIAS(
138     DATOS_CENTRADOS, REM_CAS[[i]], REM_LEG[[i]]))
139
140 D_DIF_CAS_MAR <- SAPPY(1:B,
141   FUNCTION(i) CALCULAR_DIFERENCIA_MEDIAS(
142     DATOS_CENTRADOS, REM_CAS[[i]], REM_MAR[[i]]))
143
144 D_DIF_LEG_MAR <- SAPPY(1:B,
145   FUNCTION(i) CALCULAR_DIFERENCIA_MEDIAS(
146     DATOS_CENTRADOS, REM_LEG[[i]], REM_MAR[[i]]))
147
148 # CALCULAR P-VALORES AJUSTADOS.
149 DEN <- B + 1
150 VALOR_P_CAS_LEG <- (SUM(ABS(D_DIF_CAS_LEG)) > ABS(DIF_OBS_CAS_LEG)) + 1) / DEN
151 VALOR_P_CAS_MAR <- (SUM(ABS(D_DIF_CAS_MAR)) > ABS(DIF_OBS_CAS_MAR)) + 1) / DEN
152 VALOR_P_LEG_MAR <- (SUM(ABS(D_DIF_LEG_MAR)) > ABS(DIF_OBS_LEG_MAR)) + 1) / DEN
153 VALORES_P <- c(VALOR_P_CAS_LEG, VALOR_P_CAS_MAR, VALOR_P_LEG_MAR)
154 VALORES_P_AJUSTADOS <- P.ADJUST(VALORES_P, METHOD = "BH")
155 CAT("\n")
156 CAT("P-VALORES ENTRE PARES DE RAZAS:\n")
157 CAT("CASTELLANA - LEGHORN:", round(VALORES_P_AJUSTADOS[1], 3), "\n")
158 CAT("CASTELLANA - MARANS:", round(VALORES_P_AJUSTADOS[2], 3), "\n")
159 CAT("LEGHORN - MARANS:", round(VALORES_P_AJUSTADOS[3], 3), "\n")
160 }

```

12.2 PRUEBAS DE PERMUTACIONES

En el capítulo 8 conocimos la prueba exacta de Fisher, la cual obtiene un valor p exacto tras calcular todas las permutaciones de los datos con iguales valores marginales en una tabla de contingencia como alternativa para muestras pequeñas de la prueba y considerar únicamente aquellas permutaciones que ocurren con igual o menor probabilidad que la obtenida para los datos del estudio.

La prueba exacta de Fisher es lo que se conoce como una **prueba exacta de permutaciones**, cuyo único requisito es la **intercambiabilidad**: si se cumple la hipótesis nula, todas las permutaciones pueden ocurrir con igual probabilidad. En la práctica, este tipo de métodos puede emplearse para diversos estadísticos, tales como la proporción, la media y la varianza. Puesto que el valor p entregado por las pruebas de permutaciones es exacto, no es posible obtener un intervalo de confianza.

En términos generales, las pruebas exactas de permutaciones para la diferencia entre dos grupos A y B (puede extenderse esta idea para más grupos) de tamaños n_A y n_B , respectivamente, sigue los siguientes pasos:

1. Calcular la diferencia entre el estadístico de interés observado para ambos grupos.
2. Juntar ambas muestras en una muestra combinada.
3. Obtener todas las permutaciones de la muestra combinada en que se pueden distribuir las observaciones en dos grupos de tamaños n_A y n_B .
4. Construir la distribución de las posibles diferencias, calculando la diferencia entre el estadístico de interés obtenido para ambos grupos en cada una de las permutaciones.
5. Calcular el valor p exacto, dado por la proporción de permutaciones en que el valor (absoluto, si es bilateral) de la diferencia calculada es menor/mayor o igual al valor (absoluto si es bilateral) de la diferencia observada.

Puesto que las pruebas exactas de permutaciones requieren calcular todas las permutaciones, solo resultan adecuadas para muestras pequeñas, pues requieren de una enorme cantidad de cómputos. En consecuencia, si la muestra es grande, suele tomarse una muestra aleatoria de las permutaciones posibles, procedimiento que suele denominarse **simulación de Monte Carlo**, y a partir de ella calcular un valor p aproximado dado por la ecuación 12.4.

Podemos ver que en la ecuación 12.4 se suma 1 tanto al numerador como al denominador. Esto corresponde a una corrección que debemos aplicar puesto que el método de Monte Carlo no es insesgado.

De los párrafos anteriores se desprende que las pruebas de permutaciones (exactas o no) son adecuadas para el contraste de hipótesis con dos o más muestras, pues determinan una significación estadística (valor p).

En términos generales, el procedimiento para efectuar una prueba de permutaciones usando simulaciones de Monte Carlo no es muy distinto al de bootstrapping, aunque hay algunas diferencias fundamentales en el trasfondo:

1. Formular las hipótesis a contrastar (e identificar el estadístico de interés θ).
2. Crear una gran cantidad P de permutaciones (generalmente terminada en 9 para simplificar los cálculos) a partir de las muestras originales, usando **muestreo sin reposición sobre la muestra combinada**, y obtener el estadístico θ para cada una de las muestras.
3. Generar la distribución que el estadístico θ tendría si la hipótesis nula fuese cierta.
4. Determinar la probabilidad de encontrar un valor de θ al menos tan extremo como el observado en la distribución generada.

Debemos fijarnos en que, a diferencia de bootstrapping, las pruebas de permutaciones usan muestreo sin reposición puesto que, si la hipótesis nula fuera cierta, cada permutación de los valores obtenidos en la muestra combinada sería igualmente probable. Así, lo que se hace en cada repetición es tomar una muestra sin repetición de la muestra original (es decir, “reordenar” las observaciones) y asignar aleatoriamente cada observación a uno de los grupos, respetando los tamaños n_A y n_B de las muestras originales.

12.2.1 Prueba de permutaciones para comparar una variable continua en dos muestras independientes

El profesor de una asignatura inicial de programación, que se imparte para estudiantes de primer año de Ingeniería y estudiantes de último año de otras carreras que pueden cursar dicha asignatura como electivo, desea estudiar si existen diferencias en el rendimiento académico de ambos grupos. Para ello, considera una muestra de $n_A = 20$ estudiantes de primer año de Ingeniería y $n_B = 12$ estudiantes de último año de otras carreras.

El profesor ha decidido comparar el promedio de calificaciones finales de ambos grupos, usando para ello una prueba de permutaciones con $P = 5999$ repeticiones y un nivel de significación $\alpha = 0,05$. La diferencia observada para las muestras originales es $\bar{x}_A - \bar{x}_B = -0,017$, sugiriendo que los estudiantes de Ingeniería tienen peores calificaciones. Así, las hipótesis a contrastar son:

Denotando como μ_A al promedio de calificaciones finales de estudiantes de primer año de Ingeniería en el curso inicial de programación bajo estudio, y como μ_B al promedio de calificaciones finales de estudiantes de último año de otras carreras en el mismo curso, entonces:

$$H_0: \mu_A - \mu_B = 0$$

$$H_A: \mu_A - \mu_B \neq 0$$

Tras hacer la prueba, la distribución generada se asemeja bastante a la normal, aunque con una ligera asimetría hacia la derecha (figura 12.12), y el valor p obtenido para el contraste de hipótesis es $p = 0,969$, por lo que concluye con 95 % de confianza que no hay evidencia suficiente para creer que existe diferencia en entre los promedios de las calificaciones finales de ambos grupos de estudiantes.

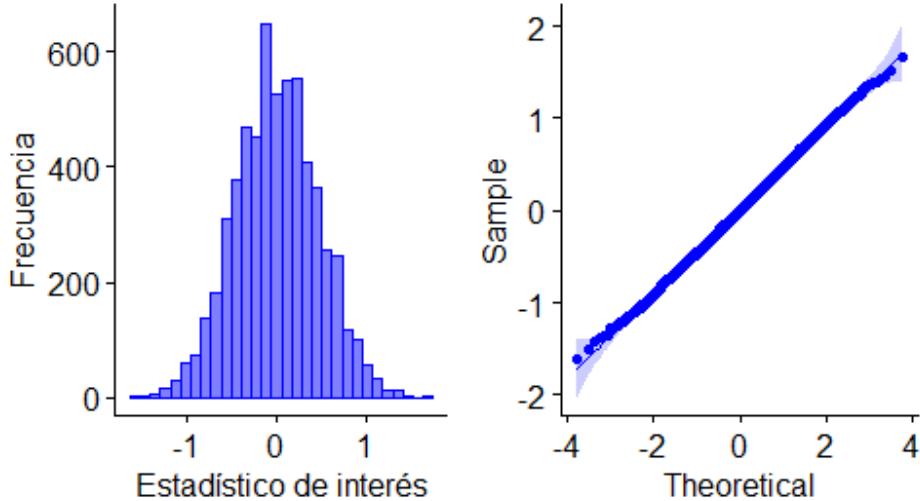


Figura 12.12: histograma y gráfico Q-Q de la distribución para la diferencia de medias generada mediante permutaciones.

Intrigado por este resultado, pues el profesor tiene la fuerte sensación de que, en general, los estudiantes de Ingeniería tienen más calificaciones deficientes que los estudiantes de otras carreras, ha decidido hacer un nuevo estudio con las mismas muestras, comparando ahora la diferencia en la variabilidad (manteniendo la misma cantidad de repeticiones e igual nivel de significación). Así:

Denotando como σ_A a la varianza de las calificaciones finales de estudiantes de primer año de Ingeniería en el curso inicial de programación bajo estudio, y como σ_B a la varianza de las calificaciones finales de estudiantes de último año de otras carreras en el mismo curso, entonces:

$$H_0: \sigma_A - \sigma_B = 0$$

$$H_A: \sigma_A - \sigma_B \neq 0$$

La diferencia observada entre las varianzas de la muestra original es $\sigma_{x_A} - \sigma_{x_B} = 2,560$, sugiriendo que la variabilidad de las calificaciones obtenidas por los estudiantes de ingeniería es mayor. Tras efectuar el contraste de hipótesis, obtiene como resultado $p = 0,003$, evidencia suficiente para rechazar la hipótesis nula en favor de la hipótesis alternativa. Así, el profesor concluye que su percepción no es del todo errada, puesto que la variabilidad de las calificaciones es significativamente mayor para los estudiantes de Ingeniería.

Para hacer estos estudios, el profesor desarrolló en R el script 12.8. A pesar de que existen algunos paquetes de R para realizar pruebas de permutaciones, hemos decidido en esta ocasión implementar el procedimiento creando la función `contrastar_hipotesis_permutaciones()`, cuya especificación puede leerse en el script 12.8, la cual realiza el proceso y arroja como resultado el valor p resultante. Podemos ver que esta función opera usando como estadístico de interés la diferencia de un estadístico θ entre dos muestras y que la función que calcula dicho estadístico θ para una muestra se entrega como argumento.

Script 12.8: pruebas de permutaciones para variables numéricas.

```

1 library(ggpubr)
2
3 # Crear muestras iniciales.
4 a <- c(5.4, 4.7, 6.3, 2.9, 5.9, 5.1, 2.1, 6.2, 1.6, 6.7, 3.0, 3.3,
5     5.0, 4.1, 3.3, 3.4, 1.2, 3.8, 5.8, 4.2)
6
7 b <- c(4.0, 4.1, 4.3, 4.3, 4.3, 4.2, 4.3, 4.3, 4.4, 4.1, 4.3, 4.0)
8
9 # Establecer semilla y cantidad de repeticiones.
10 R = 5999
11 set.seed(432)
```

```

12
13 # Función para obtener una permutación.
14 # Argumentos:
15 # - i: iterador (para llamadas posteriores).
16 # - muestra_1, muestra_2: muestras.
17 # Valor:
18 # - lista con las muestras resultantes tras la permutación.
19 obtiene_permutacion <- function(i, muestra_1, muestra_2) {
20   n_1 <- length(muestra_1)
21   combinada <- c(muestra_1, muestra_2)
22   n <- length(combinada)
23   permutacion <- sample(combinada, n, replace = FALSE)
24   nueva_1 <- permutacion[1:n_1]
25   nueva_2 <- permutacion[(n_1+1):n]
26   return(list(nueva_1, nueva_2))
27 }
28
29 # Función para calcular la diferencia de un estadístico de interés entre las
30 # dos muestras.
31 # Argumentos:
32 # - muestras: lista con las muestras.
33 # - FUN: nombre de la función que calcula el estadístico de interés.
34 # Valor:
35 # - diferencia de un estadístico para dos muestras.
36 calcular_diferencia <- function(muestras, FUN) {
37   muestra_1 <- muestras[[1]]
38   muestra_2 <- muestras[[2]]
39   diferencia <- FUN(muestra_1) - FUN(muestra_2)
40   return(diferencia)
41 }
42
43 # Función para calcular el valor p.
44 # Argumentos:
45 # - distribucion: distribución nula del estadístico de interés.
46 # - valor_observado: valor del estadístico de interés para las muestras
47 # originales.
48 # - repeticiones: cantidad de permutaciones a realizar.
49 # - alternative: tipo de hipótesis alternativa. "two.sided" para
50 # hipótesis bilateral, "greater" o "less" para hipótesis unilaterales.
51 # Valor:
52 # - el valor_p calculado.
53 calcular_valor_p <- function(distribucion, valor_observado,
54                               repeticiones, alternative) {
55   if(alternative == "two.sided") {
56     numerador <- sum(abs(distribucion) > abs(valor_observado)) + 1
57     denominador <- repeticiones + 1
58     valor_p <- numerador / denominador
59   }
60   else if(alternative == "greater") {
61     numerador <- sum(distribucion > valor_observado) + 1
62     denominador <- repeticiones + 1
63     valor_p <- numerador / denominador
64   }
65   else {
66     numerador <- sum(distribucion < valor_observado) + 1
67     denominador <- repeticiones + 1
68     valor_p <- numerador / denominador
69   }
70 }
```

```

71     return(valor_p)
72 }
73
74 # Función para graficar una distribución.
75 # Argumentos:
76 # - distribucion: distribución nula del estadístico de interés.
77 # - ...: otros argumentos a ser entregados a gghistogram y ggqqplot.
78 graficar_distribucion <- function(distribucion, ...) {
79   observaciones <- data.frame(distribucion)
80
81   histograma <- gghistogram(observaciones, x = "distribucion",
82                             xlab = "Estadístico de interés",
83                             ylab = "Frecuencia", bins = 30, ...)
84
85   qq <- ggqqplot(observaciones, x = "distribucion", ...)
86
87   # Crear una única figura con todos los gráficos de dispersión.
88   figura <- ggarrange(histograma, qq, ncol = 2, nrow = 1)
89   print(figura)
90 }

91
92 # Función para hacer la prueba de permutaciones.
93 # Argumentos:
94 # - muestra_1, muestra_2: vectores numéricos con las muestras a comparar.
95 # - repeticiones: cantidad de permutaciones a realizar.
96 # - FUN: función del estadístico E para el que se calcula la diferencia.
97 # - alternative: tipo de hipótesis alternativa. "two.sided" para
98 #   hipótesis bilateral, "greater" o "less" para hipótesis unilaterales.
99 # - plot: si es TRUE, construye el gráfico de la distribución generada.
100 # - ...: otros argumentos a ser entregados a graficar_distribucion.
101 contrastar_hipotesis_permutaciones <- function(muestra_1, muestra_2,
102                                                 repeticiones, FUN,
103                                                 alternative, plot, ...) {
104   cat("Prueba de permutaciones\n\n")
105   cat("Hipótesis alternativa:", alternative, "\n")
106   observado <- calcular_diferencia(list(muestra_1, muestra_2), FUN)
107   cat("Valor observado:", observado, "\n")
108
109   n_1 <- length(muestra_1)
110
111   # Generar permutaciones.
112   permutaciones <- lapply(1:repeticiones, obtiene_permutacion, muestra_1,
113                           muestra_2)
114
115   # Generar la distribución.
116   distribucion <- sapply(permutoaciones, calcular_diferencia, FUN)
117
118   # Graficar la distribución.
119   if(plot) {
120     graficar_distribucion(distribucion, ...)
121   }
122
123   # Calcular el valor p.
124   valor_p <- calcular_valor_p(distribucion, observado, repeticiones,
125                                 alternative)
126
127   cat("Valor p:", valor_p, "\n\n")
128 }
129

```

```

130
131
132 # Hacer pruebas de permutaciones para la media y la varianza.
133 contrastar_hipotesis_permutaciones(a, b, repeticiones = R, FUN = mean,
134                                         alternative = "two.sided", plot = TRUE,
135                                         color = "blue", fill = "blue")
136
137 contrastar_hipotesis_permutaciones(a, b, repeticiones = R, FUN = var,
138                                         alternative = "two.sided", plot = FALSE)

```

12.2.2 Prueba de permutaciones para comparar medias de más de dos muestras correlacionadas

Supongamos ahora que un estudiante de un curso de programación necesita comparar la eficiencia de tres algoritmos de ordenamiento: *quicksort*, *bubblesort* y *mergesort*. Para ello, ha seleccionado aleatoriamente 6 arreglos de igual tamaño y registrado para cada uno de ellos el tiempo de ejecución utilizado por cada algoritmo (en milisegundos) bajo iguales condiciones, como muestra la tabla 12.5.

Instancia	Quicksort	Bubblesort	Mergesort
1	11,2	15,7	12,0
2	22,6	29,3	25,7
3	23,4	30,7	25,7
4	23,3	30,8	23,7
5	21,8	29,8	25,5
6	40,1	50,3	44,7

Tabla 12.5: tiempos de ejecución para las diferentes instancias con cada algoritmo del ejemplo.

Tras comprobar mediante la figura 12.13 que no se cumple la condición de normalidad, el estudiante ha decidido usar permutaciones para resolver su problema. Para ello, ha considerado un nivel de significación $\alpha = 0,01$ y un total de 2999 repeticiones, obteniendo como resultado un valor $p = 0,0003$, mucho menor que el nivel de significación. En consecuencia, concluye con 99 % de confianza que el tiempo de ejecución promedio es significativamente diferente para al menos uno de los algoritmos.

A fin de determinar qué algoritmos difieren en su tiempo promedio de ejecución, ha decidido llevar a cabo un procedimiento post-hoc, calculando los valores p para las medias de las diferencias entre cada par de grupos para las diferentes permutaciones, obteniendo los resultados que se presentan en la figura 12.14. En consecuencia, el estudiante concluye con 99 % de confianza, que existen diferencias significativas en el tiempo promedio de ejecución entre los algoritmos Quicksort y Bubblesort y los algoritmos Bubblesort y Mergesort. Al estudiar las diferencias observadas, puede ver que Bubblesort es menos eficiente que los dos algoritmos restantes.

El script 12.9 corresponde a la solución desarrollada por el estudiante.

Script 12.9: prueba de permutaciones para muestras correlacionadas.

```

1 library(ggpubr)
2 library(ez)
3 library(tidyverse)
4
5 # Crear el data frame.
6 Quicksort <- c(11.2, 22.6, 23.4, 23.3, 21.8, 40.1)
7 Bubblesort <- c(15.7, 29.3, 30.7, 30.8, 29.8, 50.3)
8 Mergesort <- c(12.0, 25.7, 25.7, 23.7, 25.5, 44.7)

```

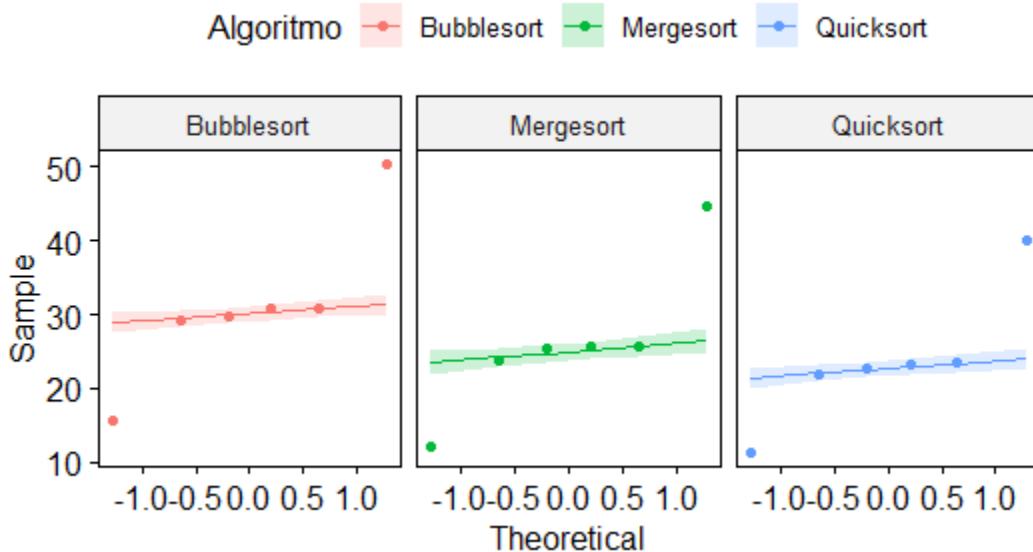


Figura 12.13: gráfico Q-Q para comprobar el supuesto de normalidad para el ejemplo.

```
Análisis post-hoc (permutaciones) para la diferencia de las medias
-----
Valores p:
Quicksort - Bubblesort: 0.000
Quicksort - Mergesort: 0.116
Bubblesort - Mergesort: 0.003

Diferencias observadas:
Quicksort - Bubblesort: -7.367
Quicksort - Mergesort: -2.483
Bubblesort - Mergesort: 4.883
```

Figura 12.14: resultado del procedimiento post-hoc.

```

9 Instancia <- factor(1:6)
10 datos_anchos <- data.frame(Instancia, Quicksort, Bubblesort, Mergesort)
11
12 datos_largos <- datos_anchos %>% pivot_longer(c("Quicksort", "Bubblesort",
13                                         "Mergesort"),
14                                         names_to = "Algoritmo",
15                                         values_to = "Tiempo")
16
17 datos_largos[["Algoritmo"]] <- factor(datos_largos[["Algoritmo"]])
18
19 # Verificar condición de normalidad.
20 g <- ggqqplot(datos_largos, "Tiempo", facet_by = "Algoritmo",
21                 color = "Algoritmo")
22
23 print(g)
24
25 # Establecer nivel de significación.
26 alfa <- 0.01
27
```

```

28 # Obtener el valor observado , correspondiente al estadístico F entregado
29 # por ANOVA para la muestra original.
30 anova <- ezANOVA(datos_largos, dv = Tiempo, within = Algoritmo,
31                   wid = Instancia, return_aov = TRUE)
32
33 valor_observado <- anova[["ANOVA"]][["F"]]
34
35 # Generar permutaciones.
36 R = 2999
37 # copia_ancha <- data.frame(datos_anchos)
38
39 set.seed(432)
40
41 # Función para obtener una permutación.
42 # Devuelve una matriz de datos con formato ancho.
43 obtiene_permutacion <- function(i, df_ancho) {
44   df_ancho[, 2:4] <- t(apply(df_ancho[, 2:4], 1, sample))
45   return(df_ancho)
46 }
47
48 # Obtiene permutaciones
49 permutaciones <- lapply(1:R, obtiene_permutacion, datos_anchos)
50
51 # Función para obtener el estadístico F para una matriz de datos con formato
52 # ancho.
53 obtiene_F <- function(df_ancho) {
54   df_largo <- df_ancho %>% pivot_longer(c("Quicksort", "Bubblesort",
55                                             "Mergesort"),
56                                             names_to = "Algoritmo",
57                                             values_to = "Tiempo")
58
59   df_largo[["Algoritmo"]] <- factor(df_largo[["Algoritmo"]])
60
61   anova <- ezANOVA(df_largo, dv = Tiempo, within = Algoritmo, wid = Instancia,
62                      return_aov = TRUE)
63   return(anova[["ANOVA"]][["F"]])
64 }
65
66 # Genera distribución de estadísticos F con las permutaciones.
67 distribucion <- sapply(permutoaciones, obtiene_F)
68
69 # Obtener valor p.
70 p <- (sum(distribucion > valor_observado) + 1) / (R + 1)
71 cat("ANOVA de una vía para muestras pareadas con permutaciones\n")
72 cat("p =", p, "\n\n")
73
74 # Análisis post-hoc.
75
76 # Función para calcular la media de las diferencias para dos columnas de una
77 # matriz de datos en formato ancho.
78 obtiene_media_difs <- function(df_ancho, columna_1, columna_2) {
79   media <- mean(df_ancho[[columna_1]] - df_ancho[[columna_2]])
80   return(media)
81 }
82
83 # Obtiene las las medias de las diferencias observadas
84 dif_obs_quick_bubble <- obtiene_media_difs(datos_anchos, "Quicksort",
85                                              "Bubblesort")
86

```

```

87 dif_obs_quick_merge <- obtiene_media_difs(datos_anchos, "Quicksort",
88                                         "Mergesort")
89
90 dif_obs_bubble_merge <- obtiene_media_difs(datos_anchos, "Bubblesort",
91                                         "Mergesort")
92
93 # Obtiene las distribuciones de las medias de las diferencias permutadas
94 dist_medias_difs_quick_bubble <- sapply(permutaciones, obtiene_media_difs,
95                                         "Quicksort", "Bubblesort")
96
97 dist_medias_difs_quick_merge <- sapply(permutaciones, obtiene_media_difs,
98                                         "Quicksort", "Mergesort")
99
100 dist_medias_difs_bubble_merge <- sapply(permutaciones, obtiene_media_difs,
101                                         "Bubblesort", "Mergesort")
102
103 # Obtener valores p.
104 num <- sum(abs(dist_medias_difs_quick_bubble) > abs(dif_obs_quick_bubble)) + 1
105 den <- R + 1
106 p_quick_bubble <- num / den
107
108 num <- sum(abs(dist_medias_difs_quick_merge) > abs(dif_obs_quick_merge)) + 1
109 den <- R + 1
110 p_quick_merge <- num / den
111
112 num <- sum(abs(dist_medias_difs_bubble_merge) > abs(dif_obs_bubble_merge)) + 1
113 den <- R + 1
114 p_bubble_merge <- num / den
115
116 cat("\n\n")
117 cat("Análisis post-hoc (permutaciones) para la diferencia de las medias\n")
118 cat("-----\n")
119 cat("Valores p:\n")
120
121 cat(sprintf("Quicksort - Bubblesort: %.3f\n", p_quick_bubble))
122 cat(sprintf("Quicksort - Mergesort: %.3f\n", p_quick_merge))
123 cat(sprintf("Bubblesort - Mergesort: %.3f\n", p_bubble_merge))
124
125 cat("\nDiferencias observadas:\n")
126 cat(sprintf("Quicksort - Bubblesort: %.3f\n", dif_obs_quick_bubble))
127 cat(sprintf("Quicksort - Mergesort: %.3f\n", dif_obs_quick_merge))
128 cat(sprintf("Bubblesort - Mergesort: %.3f\n", dif_obs_bubble_merge))

```

12.3 EJERCICIOS PROPUESTOS

1. En tus palabras, ¿qué son las técnicas de remuestreo?
2. Explica si ¿podría considerarse que la prueba exacta de Fisher usa técnicas de remuestreo?
3. ¿En qué se parecen y en qué se diferencian las técnicas de bootstrapping y permutación?
4. En tus palabras, ¿qué es una simulación Monte Carlo?
5. ¿Cómo realizarías bootstrapping para determinar si la estatura media de estudiantes de Las Condes es igual a la estatura media de estudiantes de La Pintana?
6. ¿Cómo usarías Monte Carlo para verificar que un medicamento para bajar el colesterol funciona en un grupo de 30 personas escogidas al azar?

7. ¿Cómo usarías bootstrapping para analizar si tres algoritmos necesitan tiempos similares en procesar 25 instancias de prueba del problema de la mochila?
8. ¿Cómo usarías Monte Carlo para conocer si un medicamento para bajar la presión funciona al administrarlo a un grupo de personas hipertensas, comparando con un grupo de personas hipertensas recibiendo placebo y un grupo de control de personas no hipertensas?

CAPÍTULO 13. OTRAS ALTERNATIVAS PARA DATOS PROBLEMÁTICOS

Como ya sabemos, muchos procedimientos estadísticos requieren que los datos cumplan con ciertas propiedades o condiciones, lo que no siempre ocurre. En capítulos anteriores hemos visto que, ante este escenario, podemos usar como alternativa métodos no paramétricos (capítulos 8 y 11) o remuestreo (capítulo 12), aunque no son las únicas opciones. En este capítulo abordaremos otras estrategias que podemos usar cuando necesitamos analizar datos problemáticos.

13.1 TRANSFORMACIÓN DE DATOS

Otro fenómeno que ocurre a menudo en los estudios, además del incumplimiento de ciertas condiciones, es la necesidad de convertir los datos de una escala a otra diferente. Para hacer tales transformaciones, debemos aplicar una determinada función a una variable aleatoria X , lo que nos entrega como resultado una nueva variable aleatoria Y .

Como explica Lane (s.f., pp. 1, 16), existen diversos métodos que podemos usar para transformar datos, dependiendo de la forma que tengan los datos originales y la que deseemos obtener como resultado. En esta sección conoceremos, entonces, algunas transformaciones de uso frecuente en estadística.

13.1.1 Transformación lineal

Las **transformaciones lineales** son las más sencillas de las transformaciones, y para hacerlas nos basta con aplicar una función lineal, es decir, de la forma presentada en la ecuación 13.1, donde m y n son constantes.

$$y_i = m \cdot x_i + n \quad (13.1)$$

La física nos ofrece muchos escenarios en que es necesario aplicar este tipo de transformaciones, pues es el tipo de operación que realizamos cuando convertimos de una unidad a otra. A modo de ejemplo, consideremos la conversión de grados Celsius a grados Fahrenheit:

$$F = 1,8 \cdot C + 32$$

En R, podemos hacer este tipo de transformaciones de forma muy sencilla mediante operaciones aritméticas que se aplican vectorialmente, como muestra el script 13.1. En él se transforma un vector con 4 temperaturas en grados Celsius a grados Fahrenheit. El resultado se muestra en la figura 13.1.

Script 13.1: transformación lineal para convertir grados Celcius a grados Fahrenheit.

```
1 # Crear un vector con cuatro observaciones en grados Celsius.  
2 Celsius <- c(-8, 0, 29.8, 100)  
3
```

```

4 # Aplicar transformación lineal para convertir a grados Fahrenheit.
5 Fahrenheit <- 1.8 * Celsius + 32
6
7 # Mostrar los resultados.
8 cat("Temperaturas en grados Celsius\n")
9 print(Celsius)
10 cat("\nTemperaturas en grados Fahrenheit\n")
11 print(Fahrenheit)

```

Temperaturas en grados Celsius
[1] -8.0 0.0 29.8 100.0

Temperaturas en grados Fahrenheit
[1] 17.60 32.00 85.64 212.00

Figura 13.1: resultado de la transformación lineal del script 13.1.

13.1.2 Transformación logarítmica

La transformación logarítmica nos puede servir cuando tenemos distribuciones muy asimétricas, pues ayuda a reducir la desviación y así facilita el cumplimiento de la condición de normalidad requerida por muchas de las pruebas estadísticas que ya conocemos. Para ver este efecto de manera más clara, usaremos un conjunto de datos que registra el peso corporal (en kilogramos) y el peso del cerebro (en gramos) de diversos animales, algunos de ellos extintos (Rousseeuw & Leroy, 1987, p. 57). En R, esta transformación puede hacerse gracias a la función `log(x, base)`, aunque debemos tener cuidado con posibles valores iguales a 0. El script 13.2 aplica esta transformación al peso corporal y al peso cerebral de los animales (líneas 22–23). La figura 13.2 (script 13.2, líneas 28–43) muestra gráficamente el resultado de esta transformación para el peso cerebral de los animales.

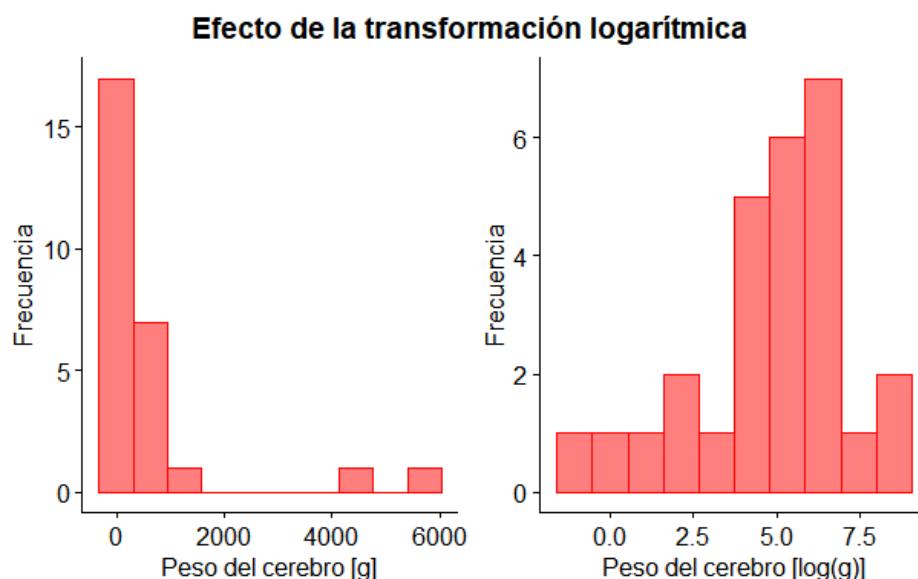


Figura 13.2: histogramas del peso cerebral antes y después de la transformación logarítmica.

Muchas veces la transformación logarítmica hace que nos sea más fácil interpretar los datos, evidenciando patrones más claros para la relación entre variables. En la figura 13.3 (script 13.2, líneas 47–60), a la derecha, se evidencia una fuerte relación entre el peso corporal y el peso del cerebro tras transformar ambas variables, relación que no podemos percibir con los datos originales (izquierda).

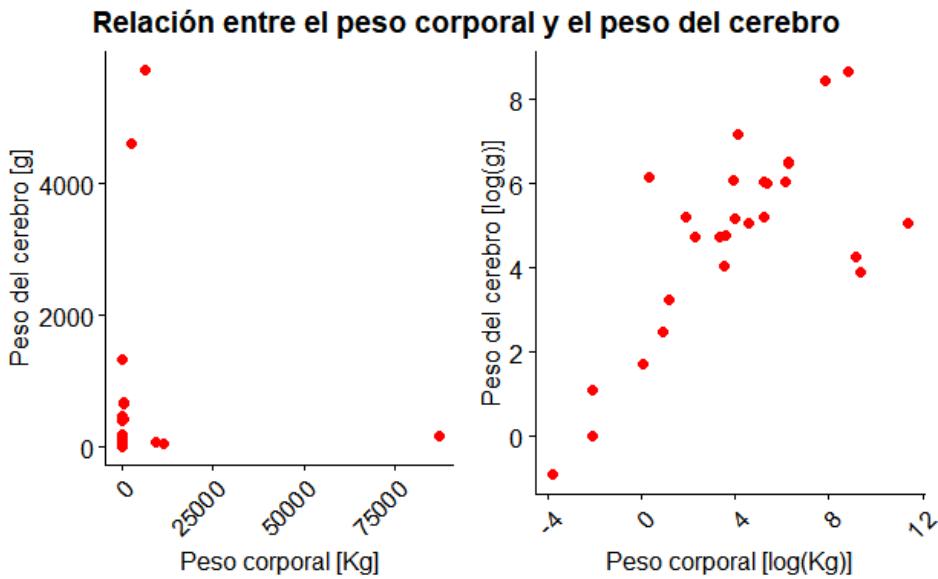


Figura 13.3: gráficos de dispersión para el peso corporal y el peso del cerebro antes y después de las transformaciones logarítmicas.

Script 13.2: transformación logarítmica.

```

1 library(ggpubr)
2
3 # Cargar datos
4 animal <- c("Mountain beaver", "Cow", "Grey wolf", "Goat", "Guinea pig",
5   "Diplodocus", "Asian elephant", "Donkey", "Horse",
6   "Potar monkey", "Cat", "Giraffe", "Gorilla", "Human",
7   "African elephant", "Triceratops", "Rhesus monkey", "Kangaroo",
8   "Golden hamster", "Mouse", "Rabbit", "Sheep", "Jaguar",
9   "Chimpanzee", "Brachiosaurus", "Mole", "Pig")
10
11 body_weight <- c(1.35, 465, 36.33, 27.66, 1.04, 11700, 2547, 187.1, 521, 10,
12   3.3, 529, 207, 62, 6654, 9400, 6.8, 35, 0.12, 0.023, 2.5,
13   55.5, 100, 52.16, 87000, 0.122, 192)
14
15 brain_weight <- c(465, 423, 119.5, 115, 5.5, 50, 4603, 419, 655, 115, 25.6,
16   680, 406, 1320, 5712, 70, 179, 56, 1, 0.4, 12.1, 175, 157,
17   440, 154.5, 3, 180)
18
19 datos <- data.frame(animal, body_weight, brain_weight)
20
21 # Aplicar transformación logarítmica
22 log_cuerpo <- log(body_weight)
23 log_cerebro <- log(brain_weight)
24 datos <- data.frame(datos, log_cuerpo, log_cerebro)
25
26 # Histogramas para el peso cerebral antes y después de la transformación
27 # logarítmica.
28 g3 <- gghistogram(datos, x = "brain_weight", bins = 10,
```

```

29         xlab = "Peso del cerebro [g]", ylab = "Frecuencia",
30         color = "red", fill = "red")
31
32 g4 <- gghistogram(datos, x = "log_cerebro", bins = 10,
33                     xlab = "Peso del cerebro [log(g)]", ylab = "Frecuencia",
34                     color = "red", fill = "red")
35
36 # Crear una única figura con ambos histogramas.
37 histograma <- ggarrange(g3, g4, ncol = 2, nrow = 1)
38
39 titulo <- text_grob("Efecto de la transformación logarítmica",
40                       face = "bold", size = 14)
41
42 histograma <- annotate_figure(histograma, top = titulo)
43 print(histograma)
44
45 # Gráficos de dispersión para la relación entre peso corporal y peso del
46 # cerebro, antes y después de aplicar la transformación logarítmica.
47 g1 <- ggscatter(datos, x = "body_weight", y = "brain_weight",
48                   color = "red", xlab = "Peso corporal [Kg]",
49                   ylab = "Peso del cerebro [g]") + rotate_x_text(45)
50
51 g2 <- ggscatter(datos, x = "log_cuerpo", y = "log_cerebro",
52                   color = "red", xlab = "Peso corporal [log(Kg)]",
53                   ylab = "Peso del cerebro [log(g)]") + rotate_x_text(45)
54
55 # Crear una única figura con los gráficos de dispersión.
56 dispersion <- ggarrange(g1, g2, ncol = 2, nrow = 1)
57 texto <- "Relación entre el peso corporal y el peso del cerebro"
58 titulo <- text_grob(texto, face = "bold", size = 14)
59 dispersion <- annotate_figure(dispersion, top = titulo)
60 print(dispersion)

```

Esos sí, tenemos que ser muy cuidadosos al usar esta transformación, porque cuando comparamos medias de datos tras una transformación logarítmica, ¡en realidad estamos comparando **medias geométricas**! Recorremos que la media geométrica se calcula de acuerdo a la ecuación 13.2 y suele ocuparse para representar tasas de crecimiento o de interés (Glen, 2021b).

$$\left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}} = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n} \quad (13.2)$$

Para ilustrar esta idea, supongamos que aplicamos una transformación logarítmica con base 10 al vector [1, 10, 100], obteniendo como resultado [0, 1, 2].

La media aritmética del vector transformado es:

$$\frac{0 + 1 + 2}{3} = 1$$

Al revertir la transformación para la media, tenemos que:

$$10^1 = 10$$

Lo que es distinto a la media aritmética de los datos originales:

$$\frac{1 + 10 + 100}{3} = 37$$

A su vez, la media geométrica del vector original es:

$$\sqrt[3]{1 \cdot 10 \cdot 100} = 10$$

Así, si dos variables a las que se ha aplicado la transformación logarítmica tienen igual media, entonces **las medias geométricas de las variables originales son iguales**.

13.1.3 Escalera de potencias de Tukey

Más general que la transformación logarítmica, la **escalera de potencias de Tukey** nos ayuda a cambiar la forma de una distribución asimétrica para que se asemeje a la normal. Este método consiste en explorar relaciones de la forma que muestra la ecuación 13.3, donde λ puede tomar cualquier valor real y se escoge de modo que la distribución de los datos transformados sea lo más cercana a la normal posible. También es útil al explorar la relación entre dos variables, en cuyo caso se busca obtener un gráfico de dispersión en que los puntos se asemejen a una recta.

$$y = x^\lambda \quad (13.3)$$

Formalmente, la transformación de Tukey se define según la ecuación 13.4, aunque (por la falta de computadores) suelen usarse únicamente aquellas que se muestran en la tabla 13.1. Fijémonos en que si $\lambda = 1$, no se realiza transformación alguna, y que para el caso de $\lambda = 0$, se tiene que $x^0 = 1$, por lo que se reemplaza en este caso por la transformación logarítmica.

$$\tilde{x}_\lambda = \begin{cases} x^\lambda & \lambda > 0 \\ \log(x) & \lambda = 0 \\ -(x^\lambda) & \lambda < 0 \end{cases} \quad (13.4)$$

λ	-2	-1	$-\frac{1}{2}$	0	$\frac{1}{2}$	1	2
\tilde{x}	$-\frac{1}{x^2}$	$-\frac{1}{x}$	$-\frac{1}{\sqrt{x}}$	$\log(x)$	\sqrt{x}	x	x^2

Tabla 13.1: escalera de transformaciones de Tukey.

Usemos ahora la población total de Estados Unidos entre los años 1610 y 1850 (United States Census Bureau, 2004, 2021) como ejemplo para entender mejor esta transformación. La figura 13.4 (líneas 18–30 del script 13.3) muestra, a la izquierda, el histograma para la población (en millones de habitantes) y, a la derecha, un gráfico de dispersión para la población por año. Podemos ver claramente que la distribución de la población presenta una fuerte asimetría hacia la izquierda y que la población parece aumentar de manera exponencial durante ese periodo.

La figura 13.5 (líneas 46–75 del script 13.3) muestra los gráficos de dispersión para la población por año tras aplicar la transformación de Tukey con diferentes valores de λ a la población. En ella podemos observar que la curva cambia gradualmente de convexa a cóncava a medida que aumenta el valor de λ , lo que deja entrever que, para obtener un resultado que sea lo más cercano posible a una línea recta, tenemos que resolver un problema de optimización que minimice los valores residuales tras ajustar una recta a los puntos transformados. De las transformaciones presentadas en la figura 13.5, la más cercana a una recta se obtiene para $\lambda = 0$.

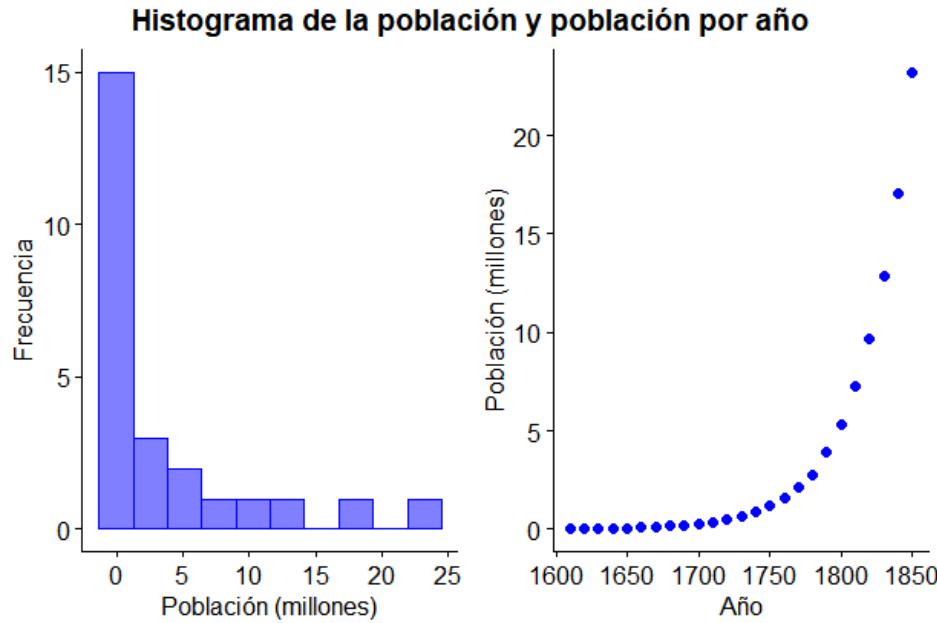


Figura 13.4: histograma de la población histórica de Estados Unidos y gráfico de dispersión de la población por año.

En párrafos anteriores mencionamos que la transformación de Tukey también permite reducir la asimetría en la distribución de los datos, como muestra la figura 13.6 (líneas 79–108 del script 13.3). En ella podemos notar que, a medida que λ aumenta, se reduce la asimetría negativa.

Como podemos suponer, reducir la asimetría nos ayuda a cumplir el requisito de normalidad que imponen muchas pruebas estadísticas, permitiéndonos así lograr resultados teóricamente más precisos. Sin embargo, una vez más tenemos que ser cuidadosos y tener en cuenta la transformación realizada al momento de interpretar los resultados. Si bien tenemos certeza que si se encuentran diferencias significativas en la variable transformada, estas diferencias **también existen en la variable original**, los estadísticos y los intervalos de confianza **¡no son los mismos!** que arrojarían las pruebas con los datos originales!

En R, el paquete `rcompanion` incluye la función `transformTukey(x, start, end, int, plotit, verbose, quiet, statistic, returnLambda)`, donde:

- `x`: vector de valores a transformar.
- `start`: valor inicial de λ a evaluar.
- `end`: valor final de λ a evaluar.
- `int`: intervalo entre los valores de λ a evaluar.
- `plotit`: si toma valor `TRUE`, entrega los siguientes gráficos:
 - Estadístico de la prueba de normalidad versus λ .
 - Histograma de los valores transformados.
 - Gráfico Q-Q de los valores transformados.
- `verbose`: si toma valor `TRUE`, muestra información adicional sobre la prueba de normalidad con respecto a λ .
- `quiet`: si toma valor `TRUE`, no muestra información alguna por pantalla.
- `statistic`: si toma valor 1, usa la prueba de normalidad de Shapiro-Wilk. Con valor 2, usa la prueba de Anderson-Darling.
- `returnLambda`: si toma valor `TRUE`, devuelve el valor de λ . Si toma valor `FALSE`, devuelve los datos transformados.

Tras llamar a la función `transformTukey()` con los datos de la población de Estados Unidos (script 13.3, líneas 111–112), obtenemos que el valor óptimo de λ es 0,12 y los gráficos de la figura 13.7. El gráfico 13.7a nos

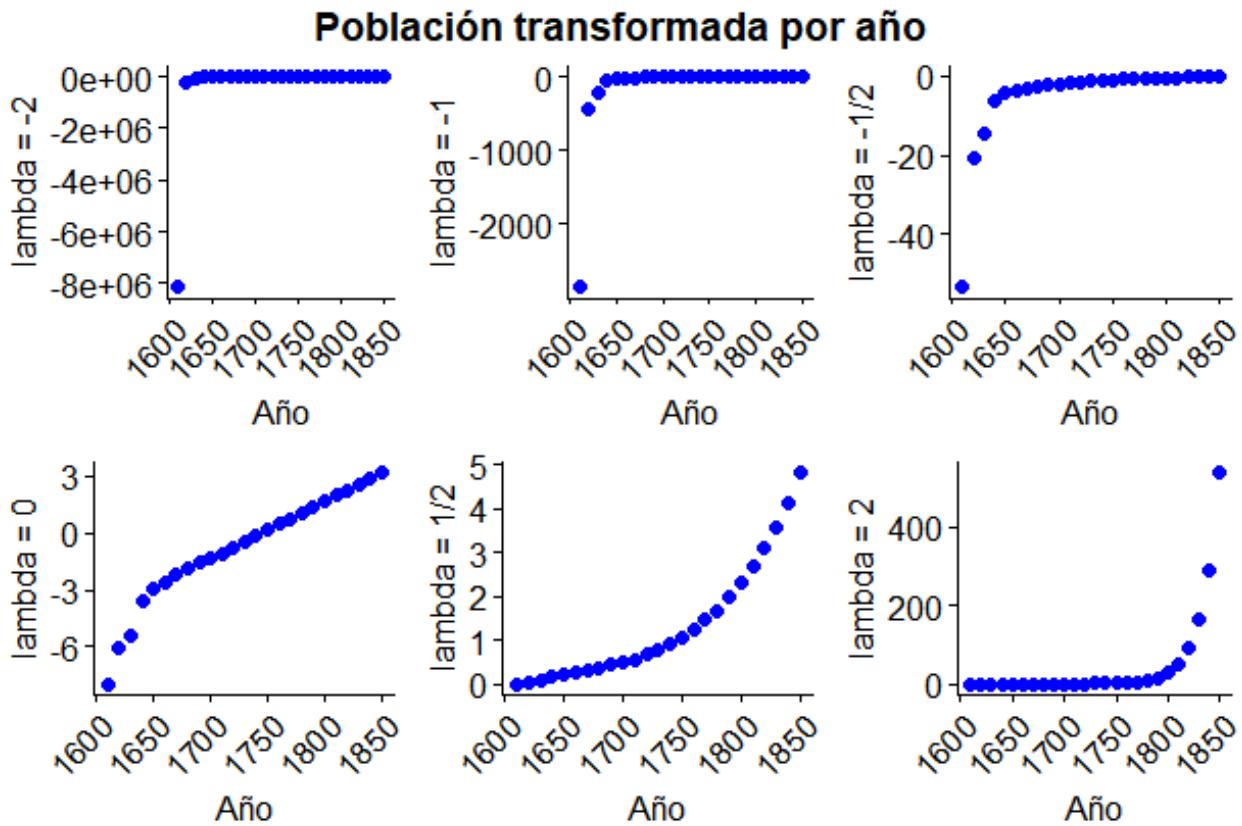


Figura 13.5: población de Estados Unidos por año tras aplicar la transformación de Tukey con distintos valores de λ .

muestra que el valor óptimo de λ es aquel que maximiza el estadístico entregado por la prueba de normalidad. A su vez, en la figura 13.7b vemos que la distribución obtenida con $\lambda = 0,12$ se asemeja mucho más a la normal que la de los datos originales o que cualquiera de las presentadas en la figura 13.6, lo que se ve confirmado por el gráfico Q-Q de la figura 13.7c.

Script 13.3: transformación de Tukey para la población total de Estados Unidos.

```

1 library(ggpubr)
2 library(rcompanion)
3
4 # Cargar datos
5 Year <- c(1610, 1620, 1630, 1640, 1650, 1660, 1670, 1680, 1690, 1700, 1710,
6   1720, 1730, 1740, 1750, 1760, 1770, 1780, 1790, 1800, 1810, 1820,
7   1830, 1840, 1850)
8
9 Population <- c(0.00035, 0.002302, 0.004646, 0.026634, 0.050368, 0.075058,
10   0.111935, 0.151507, 0.210372, 0.250888, 0.331711, 0.466185,
11   0.629445, 0.905563, 1.17076, 1.593625, 2.148076, 2.780369,
12   3.929214, 5.308483, 7.239881, 9.638453, 12.86602, 17.069453,
13   23.191876)
14
15 datos <- data.frame(Year, Population)
16
17 # Gráfico de dispersión e histograma.
18 g1 <- gghistogram(datos, x = "Population", bins = 10,
19   xlab = "Población (millones)", ylab = "Frecuencia",

```

Histogramas de la población transformada

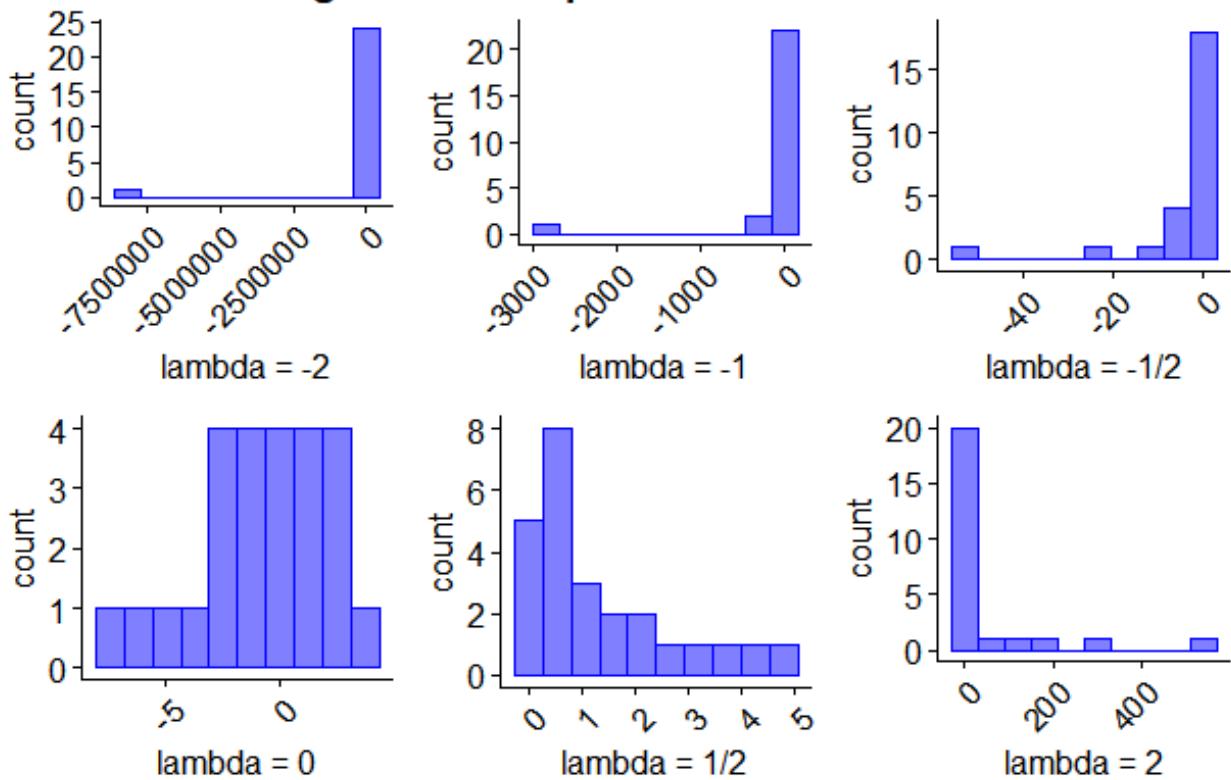
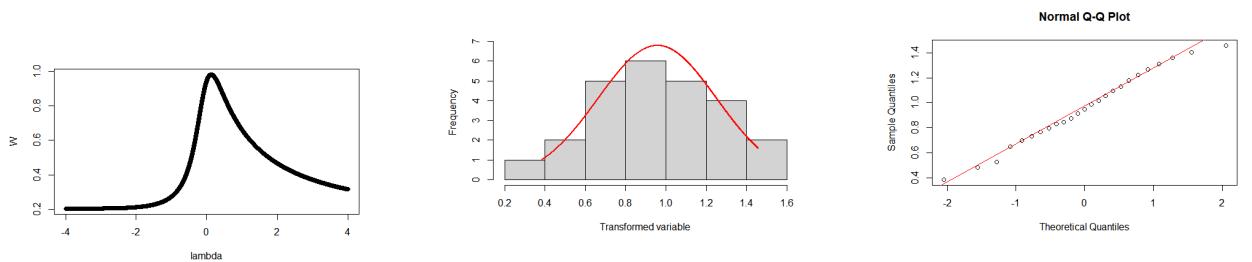


Figura 13.6: histograma de la población de Estados Unidos tras aplicar la transformación de Tukey con distintos valores de λ .

```

20                                     color = "blue", fill = "blue")
21
22 g2 <- ggscatter(datos, x = "Year", y = "Population", color = "blue",
23                   xlab = "Año", ylab = "Población (millones)")
24
25 # Histograma de la población y población por año
26 original <- ggarrange(g1, g2, ncol = 2, nrow = 1)
27 texto <- "Histograma de la población y población por año"
28 titulo <- text_grob(texto, face = "bold", size = 14)
29 original <- annotate_figure(original, top = titulo)
30 print(original)
31
32 # Transformaciones de la población
33 lambda_menos_dos <- -1 / (datos$Population ** 2)
34 lambda_menos_uno <- -1 / datos$Population
35 lambda_menos_un_medio <- -1 / sqrt(datos$Population)
36 lambda_cero <- log(datos$Population)
37 lambda_un_medio <- sqrt(datos$Population)
38 lambda_dos <- datos$Population ** 2
39
40 transformaciones <- data.frame(datos, lambda_menos_dos, lambda_menos_uno,
41                                 lambda_menos_un_medio, lambda_cero,
42                                 lambda_un_medio, lambda_dos)
43
44 # Gráficos de dispersión para la transformación de Tukey de la población y el

```



(a) estadístico W de la prueba de Shapiro-Wilk por cada valor de λ .

(b) histograma de la población transformada con el valor óptimo de λ .

(c) gráfico Q-Q de la población transformada con el valor óptimo de λ .

Figura 13.7: gráficos entregados por `transformTukey()`.

```

45 # año, usando distintos valores de lambda.
46 gt1 <- ggscatter(transformaciones, x = "Year", y = "lambda_menos_dos",
47                   color = "blue", xlab = "Año",
48                   ylab = "lambda = -2") + rotate_x_text(45)
49
50 gt2 <- ggscatter(transformaciones, x = "Year", y = "lambda_menos_uno",
51                   color = "blue", xlab = "Año",
52                   ylab = "lambda = -1") + rotate_x_text(45)
53
54 gt3 <- ggscatter(transformaciones, x = "Year", y = "lambda_menos_un_medio",
55                   color = "blue", xlab = "Año",
56                   ylab = "lambda = -1/2") + rotate_x_text(45)
57
58 gt4 <- ggscatter(transformaciones, x = "Year", y = "lambda_cero",
59                   color = "blue", xlab = "Año",
60                   ylab = "lambda = 0") + rotate_x_text(45)
61
62 gt5 <- ggscatter(transformaciones, x = "Year", y = "lambda_un_medio",
63                   color = "blue", xlab = "Año",
64                   ylab = "lambda = 1/2") + rotate_x_text(45)
65
66 gt6 <- ggscatter(transformaciones, x = "Year", y = "lambda_dos",
67                   color = "blue", xlab = "Año",
68                   ylab = "lambda = 2") + rotate_x_text(45)
69
70 # Crear una única figura con todos los gráficos de dispersión.
71 dispersion <- ggarrange(gt1, gt2, gt3, gt4, gt5, gt6, ncol = 3, nrow = 2)
72 texto <- "Población transformada por año"
73 titulo <- text_grob(texto, face = "bold", size = 14)
74 dispersion <- annotate_figure(dispersion, top = titulo)
75 print(dispersion)
76
77 # Histogramas para la transformación de Tukey de la población y el año,
78 # usando distintos valores de lambda.
79 h1 <- gghistogram(transformaciones, bins = 10, x = "lambda_menos_dos",
80                     color = "blue", fill = "blue",
81                     xlab = "lambda = -2") + rotate_x_text(45)
82
83 h2 <- gghistogram(transformaciones, bins = 10, x = "lambda_menos_uno",
84                     color = "blue", fill = "blue",
85                     xlab = "lambda = -1") + rotate_x_text(45)
86
87 h3 <- gghistogram(transformaciones, bins = 10, x = "lambda_menos_un_medio",

```

```

88         color = "blue", fill = "blue",
89         xlab = "lambda = -1/2") + rotate_x_text(45)
90
91 h4 <- gghistogram(transformaciones, bins = 10, x = "lambda_cero",
92                     color = "blue", fill = "blue",
93                     xlab = "lambda = 0") + rotate_x_text(45)
94
95 h5 <- gghistogram(transformaciones, bins = 10, x = "lambda_un_medio",
96                     color = "blue", fill = "blue",
97                     xlab = "lambda = 1/2") + rotate_x_text(45)
98
99 h6 <- gghistogram(transformaciones, bins = 10, x = "lambda_dos",
100                     color = "blue", fill = "blue",
101                     xlab = "lambda = 2") + rotate_x_text(45)
102
103 # Crear una única figura con todos los gráficos de dispersión.
104 histograma <- ggarrange(h1, h2, h3, h4, h5, h6, ncol = 3, nrow = 2)
105 texto <- "Histogramas de la población transformada"
106 titulo <- text_grob(texto, face = "bold", size = 14)
107 histograma <- annotate_figure(histograma, top = titulo)
108 print(histograma)
109
110 # Buscar la mejor transformación de Tukey usando una función de R.
111 transformacion <- transformTukey(datos$Population, start = -4, end = 4,
112                                     int = 0.001, returnLambda = TRUE)

```

13.1.4 Transformaciones Box-Cox

La **transformación Box-Cox** es una versión escalada de la transformación de Tukey, dada por la ecuación 13.5:

$$x'_\lambda = \frac{x^\lambda - 1}{\lambda} \quad (13.5)$$

Si bien a primera vista parece muy diferente a la ecuación 13.4, podemos reescribirla como:

$$x'_\lambda = \frac{x^\lambda - 1}{\lambda} \approx \frac{(1 + \lambda \log(x) + \frac{1}{2}\lambda^2 \log(x)^2 + \dots) - 1}{\lambda}$$

De donde:

$$\lim_{\lambda \rightarrow 0} \frac{(1 + \lambda \log(x) + \frac{1}{2}\lambda^2 \log(x)^2 + \dots) - 1}{\lambda} = 0$$

Tras aplicar la regla de l'Hôpital, obtenemos finalmente que:

$$\lim_{\lambda \rightarrow 0} x'_\lambda = \log(x)$$

Por lo que, al igual que en la escalera de potencias de Tukey, pero de forma más natural, empleamos la transformación logarítmica para $\lambda = 0$.

La figura 13.8 (creada con el script 13.4, líneas 39–64) muestra los gráficos de dispersión para la población total de Estados Unidos por año tras aplicar la transformación de Box-Cox con diferentes valores de λ . Podemos ver que el resultado se parece al que obtuvimos con la transformación de Tukey (figura 13.5).

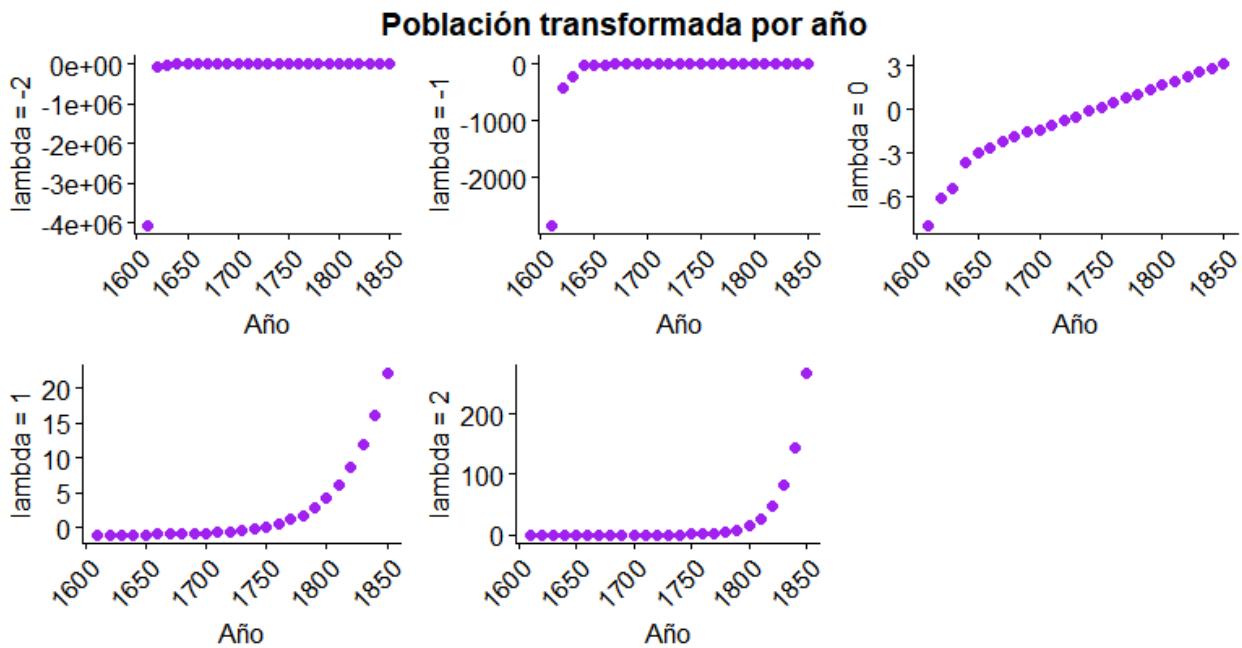


Figura 13.8: población de Estados Unidos por año tras aplicar la transformación de Box-Cox con distintos valores de λ .

Una característica interesante de esta transformación es que, para cualquier valor de λ , $x'_\lambda = 0$ cuando $x = 1$. Podemos observar esto claramente en la figura 13.9, que compara transformaciones Box-Cox con distintos valores de λ con $\log(x)$.

El paquete **DescTools** de R incluye varias funciones que permiten efectuar la transformación Box-Cox (Carchedi et al., s.f.). Destacan entre ellas:

- **BoxCoxLambda(x, lower, upper)**: devuelve el valor óptimo de λ para la transformación Box-Cox del vector x .
- **BoxCox(x, lambda)**: devuelve un vector correspondiente a la transformación Box-Cox de x con parámetro λ .
- **BoxCoxInv(x, lambda)**: revierte la transformación Box-Cox del vector x con parámetro λ .

Donde:

- x : vector numérico.
- λ : límite inferior para los posibles valores de λ .
- $upper$: límite superior para los posibles valores de λ .
- λ : parámetro de la transformación.

En las líneas 67–70 del script 13.4 se determina el valor óptimo del parámetro λ , obteniéndose como resultado $\lambda = 0,09942656$, para luego efectuar la transformación Box-Cox correspondiente de la población de Estados Unidos. La figura 13.10 (script 13.4, líneas 84–87) muestra gráficamente el resultado de la transformación. En el gráfico 13.10a podemos ver que la relación entre la población transformada y el año se asemeja a una recta, mientras que el histograma de la figura 13.10b se parece bastante a una distribución normal, hecho que vemos confirmado por el gráfico Q-Q de la figura 13.10c.

Script 13.4: transformación de Box-Cox para la población total de Estados Unidos.

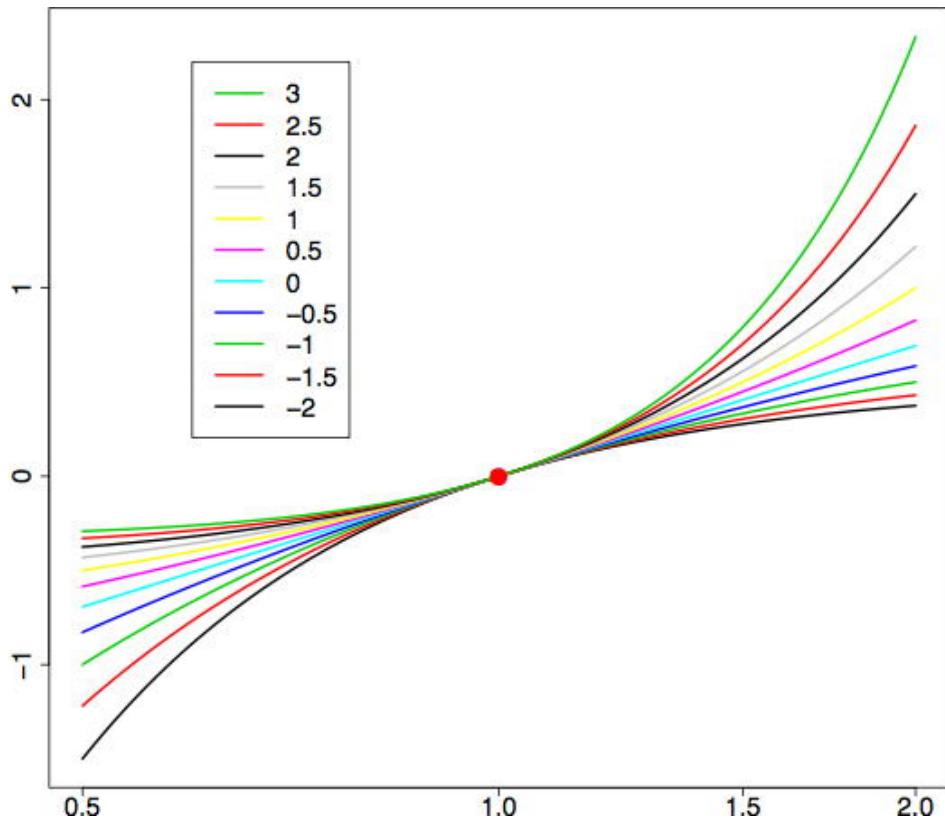
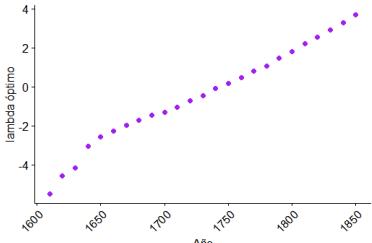


Figura 13.9: ejemplos de la transformación Box-Cox versus $\log(x)$. Fuente: (Lane, s.f., p. 16).

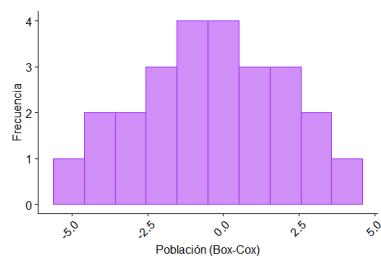
```

1 library(ggpubr)
2 library(DescTools)
3
4 # Cargar datos
5 Year <- c(1610, 1620, 1630, 1640, 1650, 1660, 1670, 1680, 1690, 1700, 1710,
6     1720, 1730, 1740, 1750, 1760, 1770, 1780, 1790, 1800, 1810, 1820,
7     1830, 1840, 1850)
8
9 Population <- c(0.00035, 0.002302, 0.004646, 0.026634, 0.050368, 0.075058,
10     0.111935, 0.151507, 0.210372, 0.250888, 0.331711, 0.466185,
11     0.629445, 0.905563, 1.17076, 1.593625, 2.148076, 2.780369,
12     3.929214, 5.308483, 7.239881, 9.638453, 12.86602, 17.069453,
13     23.191876)
14
15 datos <- data.frame(Year, Population)
16
17 # Transformación de Box-cox
18 box_cox <- function(x, lambda) {
19     if(lambda == 0) {
20         return(log(x))
21     }
22
23     resultado <- (x ** lambda - 1) / lambda
24     return(resultado)
25 }
26
27 # Transformaciones de la población

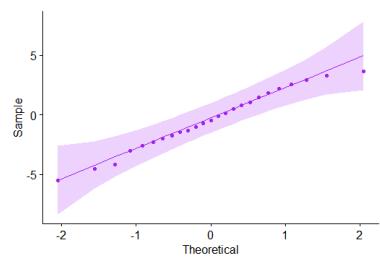
```



(a) población de Estados Unidos por año tras aplicar la transformación de Box-Cox con λ óptimo.



(b) histograma de la población transformada con el valor óptimo de λ .



(c) gráfico Q-Q de la población transformada con el valor óptimo de λ .

Figura 13.10: gráficos de población de Estados Unidos por año usando la transformación de Box-Cox.

```

28 lambda_menos_dos <- box_cox(datos$Population, -2)
29 lambda_menos_uno <- box_cox(datos$Population, -1)
30 lambda_cero <- box_cox(datos$Population, 0)
31 lambda_uno <- box_cox(datos$Population, 1)
32 lambda_dos <- box_cox(datos$Population, 2)
33
34 transformaciones <- data.frame(datos, lambda_menos_dos, lambda_menos_uno,
35                               lambda_cero, lambda_uno, lambda_dos)
36
37 # Gráficos de dispersión para la transformación de Box-Cox de la población y
38 # el año, usando distintos valores de lambda.
39 gt1 <- ggscatter(transformaciones, x = "Year", y = "lambda_menos_dos",
40                   color = "purple", xlab = "Año",
41                   ylab = "lambda = -2") + rotate_x_text(45)
42
43 gt2 <- ggscatter(transformaciones, x = "Year", y = "lambda_menos_uno",
44                   color = "purple", xlab = "Año",
45                   ylab = "lambda = -1") + rotate_x_text(45)
46
47 gt3 <- ggscatter(transformaciones, x = "Year", y = "lambda_cero",
48                   color = "purple", xlab = "Año",
49                   ylab = "lambda = 0") + rotate_x_text(45)
50
51 gt4 <- ggscatter(transformaciones, x = "Year", y = "lambda_uno",
52                   color = "purple", xlab = "Año",
53                   ylab = "lambda = 1") + rotate_x_text(45)
54
55 gt5 <- ggscatter(transformaciones, x = "Year", y = "lambda_dos",
56                   color = "purple", xlab = "Año",
57                   ylab = "lambda = 2") + rotate_x_text(45)
58
59 # Crear una única figura con todos los gráficos de dispersión.
60 dispersion <- ggarrange(gt1, gt2, gt3, gt4, gt5, ncol = 3, nrow = 2)
61 texto <- "Población transformada por año"
62 titulo <- text_grob(texto, face = "bold", size = 14)
63 dispersion <- annotate_figure(dispersion, top = titulo)
64 print(dispersion)
65
66 # Buscar la mejor transformación Box-Cox usando funciones de R.
67 lambda <- BoxCoxLambda(datos$Population, lower = -4, upper = 4)
68 cat("Lambda óptimo:", lambda)
69 transformacion <- BoxCox(datos$Population, lambda)

```

```

70 datos <- data.frame(datos, transformacion)
71
72 # Graficar los datos transformados.
73 g1 <- ggqqplot(transformacion, color = "purple")
74 print(g1)
75
76 g2 <- gghistogram(datos, bins = 10, x = "transformacion", color = "purple",
77                      fill = "purple", xlab = "Población (Box-Cox)",
78                      ylab = "Frecuencia") + rotate_x_text(45)
79
80 print(g2)
81
82 # Gráfico de dispersión para la transformación de Box-Cox de la población y
83 # el año, usando lambda óptimo.
84 g3 <- ggscatter(datos, x = "Year", y = "transformacion", color = "purple",
85                  xlab = "Año", ylab = "lambda óptimo") + rotate_x_text(45)
86
87 print(g3)

```

13.2 MÉTODOS ROBUSTOS

Hemos mencionado varias veces en este libro que muchas de las pruebas estadísticas clásicas requieren que los datos sigan una distribución cercana a la normal, pero que es frecuente que los datos disponibles no cumplan esta u otras condiciones.

Pensemos como ejemplo en la prueba t de Student (capítulo 5), que se usa para inferir acerca de la media de una población. Sin embargo, como mencionamos en el capítulo 2, esta medida de tendencia central tiene el problema de ser sensible a la presencia de valores atípicos, a distribuciones asimétricas o a muestras muy pequeñas. En términos generales, el incumplimiento de las condiciones del supuesto de normalidad puede causar diversos problemas:

- Resultados sesgados.
- Intervalos de confianza calculados de manera inadecuada.
- Reducción del poder estadístico de la prueba.

Hemos visto que muchas veces las alternativas no paramétricas resultan útiles cuando las muestras son pequeñas y presentan distribuciones asimétricas, aunque estas pruebas igualmente requieren verificar ciertas condiciones.

Otra alternativa es usar métodos de remuestreo para establecer la distribución muestral y poder emplear métodos paramétricos. Sin embargo, sabemos que el remuestreo suele ser muy costoso en términos de la cantidad de cálculos realizados.

La sección anterior, por su parte, nos ofrece la opción de aplicar ciertas transformaciones a los datos de modo que se asemejen más a la distribución normal, suponiendo que el tamaño de la(s) muestra(s) sea adecuado. No obstante, este enfoque tiene la dificultad de que, tras la transformación, en realidad estamos infiriendo acerca de un nuevo parámetro, lo cual podría alejarse significativamente de la pregunta de investigación original. ¡Recordemos que al aplicar la transformación logarítmica, por ejemplo, la prueba t de Student infiere sobre la media geométrica en lugar de la media aritmética!

En el capítulo 2 mencionamos la existencia de estimadores robustos, poco sensibles a asimetrías muestrales o valores atípicos. No obstante, el paradigma estadístico tradicional no suele considerarlos. En esta sección, basada en las ideas expuestas por Mair y Wilcox (2020), aborda, en consecuencia, alternativas robustas para muchas de las pruebas estudiadas hasta ahora, disponibles en el paquete **WRS2** de R.

13.2.1 Alternativas robustas para la media

En el capítulo 2 conocimos distintas medidas de tendencia central. Entre ellas vimos que la mediana, correspondiente al valor central (o el promedio de los dos valores centrales) de la muestra ordenada, es una alternativa robusta a la media. No obstante, existen otras opciones que nos pueden ser útiles.

13.2.1.1 Media truncada

La **media truncada** es bastante similar a la media aritmética que ya conocemos, con la diferencia de que se calcula descartando un determinado porcentaje (γ) de los valores en ambos extremos (colas) de la distribución. Tomemos como ejemplo una muestra x con 10 elementos, los cuales han sido ordenados por simplicidad:

$$x = [1, 4, 37, 38, 40, 43, 43, 45, 87, 91]$$

Si calculamos la media para la muestra anterior, tenemos que es $\bar{x} = 42,9$. No obstante, si observamos la muestra del ejemplo con detención, podemos darnos cuenta de que los valores extremos parecen ser atípicos y, en consecuencia, pueden tener una gran influencia en el valor resultante para la media. Así, podría ser más adecuado calcular la media truncada con $\gamma = 0,2$, es decir, podando el 20 % de los valores más pequeños y el 20 % de los valores más grandes, con lo que obtendremos:

$$\bar{x}_t = \frac{37 + 38 + 40 + 43 + 43 + 45}{6} = 41$$

Así, si $\gamma = 0,5$, se obtiene la mediana.

En R, podemos calcular la media truncada mediante la ya conocida función `mean()` del paquete base, agregando el argumento adicional `trim` con la proporción γ de los datos extremos a descartar, esto es `mean(x, trim = 0.2)` para el ejemplo.

13.2.1.2 Media Winsorizada

Un problema de la media truncada es que, al usarla, descartamos muchos datos, lo que puede causar problemas en algunos casos. Otra opción puede ser, en lugar de descartar los valores extremos en cada cola, reemplazarlos por los valores extremos que no serían descartados al usar la media truncada y luego calcular la media con la muestra modificada. A esta medida se le conoce como **media Winsorizada**. Si retomamos nuestro ejemplo para la media truncada, los valores extremos tras la operación de truncado son 37 y 45. Así, reemplazamos los valores truncados en la muestra original por estos nuevos extremos, con lo que nuestra muestra Winsorizada es:

$$x = [37, 37, 37, 38, 40, 43, 43, 45, 45, 45]$$

Con lo que la media Winsorizada es:

$$\bar{x}_W = \frac{37 + 37 + 37 + 38 + 40 + 43 + 43 + 45 + 45 + 45}{10} = 41$$

En R, podemos hacer este cálculo mediante la función `winmean(x, tr)` del paquete `WRS2`, donde:

- `x`: vector con la muestra original.
- `tr`: proporción de los datos en cada cola a Winsorizar.

Así, la llamada para el ejemplo sería `winmean(x, tr = 0.2)`.

13.2.2 Prueba de Yuen para dos muestras independientes

La **prueba de Yuen** es una buena alternativa a la prueba t de Student para muestras independientes cuando las varianzas de ambas muestras son muy diferentes o los tamaños de las muestras son muy dispares. Utiliza las medias truncadas y las medias Winsorizadas, aunque no se recomienda usar esta prueba si las muestras se truncan cerca del nivel de medianas ($\gamma \approx 0,5$).

Cuando tenemos dos muestras independientes, el estadístico de prueba está dado por la ecuación 13.6, donde \bar{x}_{ti} son las medias truncadas de cada muestra.

$$T_y = \frac{\bar{x}_{t1} - \bar{x}_{t2}}{\sqrt{d_1 + d_2}} \quad (13.6)$$

El denominador de la ecuación 13.6 corresponde al error estándar, donde d_i se calcula como señala la ecuación 13.7, con:

- s_{w1} : desviación estándar de la muestra Winsorizada.
- n_i : tamaño de la muestra original.
- h_i : tamaño de la muestra truncada.

$$d_i = \frac{(n_i - 1)s_{wi}^2}{h_i(h_i - 1)} \quad (13.7)$$

El estadístico T_y sigue una distribución t cuyos grados de libertad se calculan mediante la ecuación 13.8.

$$\nu_y = \frac{(d_1 + d_2)^2}{\frac{d_1^2}{h_1-1} + \frac{d_2^2}{h_2-1}} \quad (13.8)$$

A su vez, la ecuación 13.9 muestra cómo se construye el intervalo de confianza, donde t corresponde al cuantil $1 - \alpha/2$ de la distribución t con ν_y grados de libertad.

$$(\bar{x}_{t1} - \bar{x}_{t2}) \pm t \sqrt{d_1 + d_2} \quad (13.9)$$

Para una prueba de hipótesis bilateral, las hipótesis son:

$$\begin{aligned} H_0: \mu_{t1} &= \mu_{t2} \\ H_A: \mu_{t1} &\neq \mu_{t2} \end{aligned}$$

En R, podemos aplicar la prueba de Yuen para muestras independientes mediante la función `yuen(formula, data, tr)` del paquete `WRS2`, donde:

- **formula:** tiene la forma <variable dependiente> \sim <variable independiente>. Note que la variable independiente debe tener dos niveles, a fin de determinar a qué muestra pertenece cada observación de la variable dependiente.
- **data:** matriz de datos.
- **tr:** parámetro γ de la poda.

Para pruebas unilaterales, sin embargo, se recomienda usar la variante con bootstrap, implementada en la función `yuen(formula, data, tr, nboot)`, donde `nboot` señala la cantidad de muestras a obtener mediante bootstrapping.

El script 13.5 muestra un ejemplo de uso de la prueba de Yuen. Como contexto, se desea comparar el tiempo promedio de ejecución (en milisegundos) de dos algoritmos. Se han seleccionado aleatoriamente 70 instancias de igual tamaño del problema, las cuales han sido asignadas al azar a cada uno de los algoritmos. En consecuencia, contamos con $n_a = 40$ observaciones para el algoritmo A y $n_b = 30$ observaciones para el algoritmo B. Se ha establecido para este estudio un nivel de significación $\alpha = 0,05$.

Las líneas 19–20 del script 13.5 construyen gráficos Q-Q para comprobar el supuesto de normalidad de la prueba t de Student, obteniéndose como resultado la figura 13.11, donde podemos observar que las muestras obtenidas no se distribuyen normalmente, especialmente para el caso del algoritmo A.

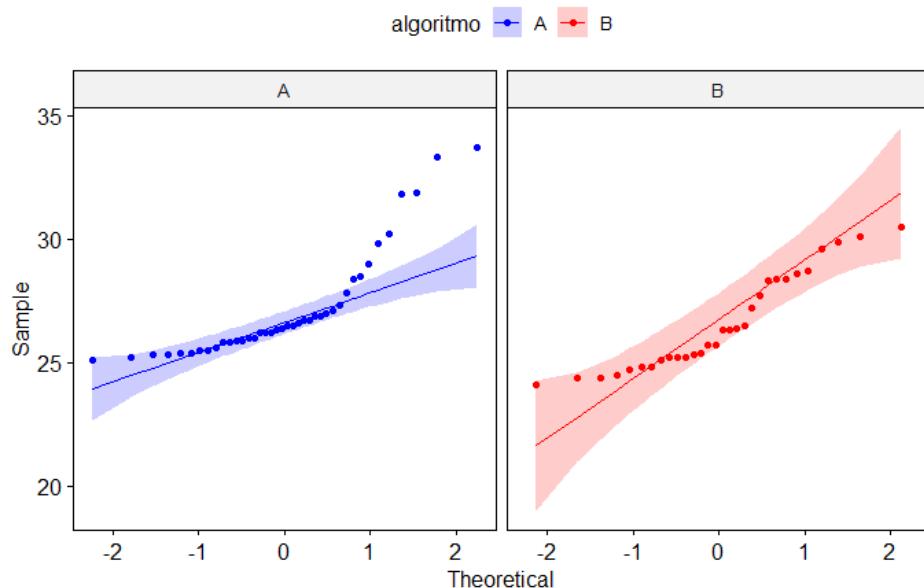


Figura 13.11: gráfico Q-Q de las muestras originales.

Para ilustrar los conceptos asociados a la prueba de Yuen, las líneas 28–45 del script 13.5 truncan ambas muestras considerando $\gamma = 0,2$ y construyen gráficos Q-Q para las muestras podadas (figura 13.12). Podemos apreciar que, tras la poda, la distribución de los datos se approxima más a la normal.

Finalmente, las líneas 48–49 del script 13.5 efectúan la prueba de Yuen para ambas muestras, obteniéndose como resultado (figura 13.13) una diferencia entre las medias truncadas de 0,246, con intervalo de 95 % de confianza ($-0,859; 1,351$) y tamaño del efecto de 0,090. El valor p obtenido, $p = 0,653$, no es significativo al nivel de significación establecido, por lo que concluimos con 95 % de confianza que no es posible descartar que ambos algoritmos tienen, en promedio, igual tiempo de ejecución.

Script 13.5: prueba de Yuen para dos muestras independientes.

```

1 library(WRS2)
2 library(ggpubr)
3

```

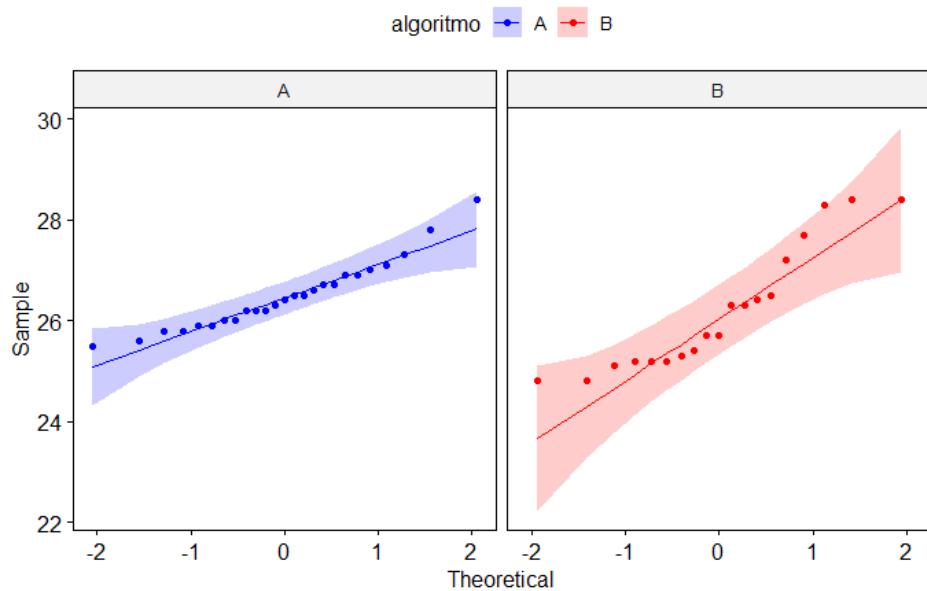


Figura 13.12: gráfico Q-Q de las muestras truncadas.

```

Call:
yuen(formula = tiempo ~ algoritmo, data = datos, tr = gamma)

Test statistic: 0.455 (df = 29.05), p-value = 0.65252

Trimmed mean difference: 0.24583
95 percent confidence interval:
-0.8592    1.3509

Explanatory measure of effect size: 0.09

```

Figura 13.13: resultado de la prueba de Yuen para el ejemplo.

```

4 # Construir data frame.
5 a <- c(25.1, 25.2, 25.3, 25.3, 25.4, 25.4, 25.4, 25.5, 25.5, 25.5, 25.6,
6     25.9, 25.9, 26.0, 26.0, 26.2, 26.2, 26.2, 26.3, 26.4, 26.5, 26.5,
7     26.6, 26.7, 26.7, 26.9, 26.9, 27.0, 27.1, 27.3, 27.8, 28.4, 28.5,
8     29.0, 29.8, 30.2, 31.8, 31.9, 33.3, 33.7)
9
10 b <- c(24.1, 24.4, 24.4, 24.5, 24.7, 24.8, 24.8, 25.1, 25.2, 25.2, 25.2,
11     25.3, 25.4, 25.7, 25.7, 26.3, 26.3, 26.4, 26.5, 27.2, 27.7, 28.3,
12     28.4, 28.4, 28.6, 28.7, 29.6, 29.9, 30.1, 30.5)
13
14 tiempo <- c(a, b)
15 algoritmo <- c(rep("A", length(a)), rep("B", length(b)))
16 datos <- data.frame(tiempo, algoritmo)
17
18 # Comprobar normalidad.
19 g <- ggqqplot(datos, x = "tiempo", facet.by = "algoritmo",
20                 palette = c("blue", "red"), color = "algoritmo")
21
22 print(g)

```

```

23
24 # Establecer nivel de significación.
25 alfa <- 0.05
26
27 # Ver poda del 20%.
28 gamma <- 0.2
29 n_a <- length(a)
30 n_b <- length(b)
31
32 poda_a <- n_a * gamma
33 poda_b <- n_b * gamma
34
35 a_truncada <- a[poda_a:(n_a - poda_a)]
36 b_truncada <- b[poda_b:(n_b - poda_b)]
37
38 tiempo <- c(a_truncada, b_truncada)
39 algoritmo <- c(rep("A", length(a_truncada)), rep("B", length(b_truncada)))
40 datos_truncados <- data.frame(tiempo, algoritmo)
41
42 g <- ggqqplot(datos_truncados, x = "tiempo", facet.by = "algoritmo",
43                  palette = c("blue", "red"), color = "algoritmo")
44
45 print(g)
46
47 # Aplicar prueba de Yuen.
48 prueba <- yuen(tiempo ~ algoritmo, data = datos, tr = gamma)
49 print(prueba)

```

El paquete WRS2 incluye también la función pb2gen(formula, data, est, nboot), que usa bootstrapping para aplicar la prueba de Yuen usando otras medidas robustas de tendencia central, donde:

- **formula:** tiene la misma forma descrita para la prueba de Yuen.
- **data:** matriz de datos.
- **est:** medida a emplear. Puede tomar las opciones “**mean**” para la media y “**median**” (mediana), entre otras opciones que escapan a los alcances de este curso.
- **nboot:** cantidad de muestras a generar mediante bootstrapping.

El script 13.6 muestra cómo aplicar la prueba de Yuen para el ejemplo, usando como estimadores la media y la mediana, obteniéndose los resultados de la figura 13.14. Podemos ver que, en ambos casos, el valor p obtenido es más bajo que al usar la media podada.

Script 13.6: prueba de Yuen con bootstrapping para dos muestras independientes usando la media y la mediana.

```

1 library(WRS2)
2
3 # Construir data frame.
4 a <- c(25.1, 25.2, 25.3, 25.3, 25.4, 25.4, 25.5, 25.5, 25.5, 25.6, 25.8, 25.8,
5      25.9, 25.9, 26.0, 26.0, 26.2, 26.2, 26.2, 26.3, 26.4, 26.5, 26.5,
6      26.6, 26.7, 26.7, 26.9, 26.9, 27.0, 27.1, 27.3, 27.8, 28.4, 28.5,
7      29.0, 29.8, 30.2, 31.8, 31.9, 33.3, 33.7)
8
9 b <- c(24.1, 24.4, 24.4, 24.5, 24.7, 24.8, 24.8, 25.1, 25.2, 25.2, 25.2,
10     25.3, 25.4, 25.7, 25.7, 26.3, 26.3, 26.4, 26.5, 27.2, 27.7, 28.3,
11     28.4, 28.4, 28.6, 28.7, 29.6, 29.9, 30.1, 30.5)
12
13 tiempo <- c(a, b)
14 algoritmo <- c(rep("A", length(a)), rep("B", length(b)))
15 datos <- data.frame(tiempo, algoritmo)
16

```

```

Resultado al usar la media como estimador

Call:
pb2gen(formula = tiempo ~ algoritmo, data = datos, est = "mean",
nboot = bootstrap)

Test statistic: 0.61, p-value = 0.21321
95% confidence interval:
-0.3008    1.5617

Resultado al usar la mediana como estimador

Call:
pb2gen(formula = tiempo ~ algoritmo, data = datos, est = "median",
nboot = bootstrap)

Test statistic: 0.45, p-value = 0.47147
95% confidence interval:
-0.95    1.35

```

Figura 13.14: resultado de la prueba de Yuen con bootstrapping para el ejemplo, usando como estimadores la media y la mediana.

```

17 # Establecer nivel de significación y cantidad de muestras a generar
18 # con bootstrapping.
19 alfa <- 0.05
20 bootstrap <- 999
21
22 # Aplicar prueba con la media
23 set.seed(135)
24
25 prueba_media <- pb2gen(tiempo ~ algoritmo ,
26                         data = datos,
27                         est = "mean",
28                         nboot = bootstrap)
29
30 cat("\n\nResultado al usar la media como estimador\n\n")
31 print(prueba_media)
32
33 # Aplicar prueba con la mediana
34 set.seed(135)
35
36 prueba_mediana <- pb2gen(tiempo ~ algoritmo ,
37                           data = datos,
38                           est = "median",
39                           nboot = bootstrap)
40
41 cat("Resultado al usar la mediana como estimador\n\n")
42 print(prueba_mediana)

```

13.2.3 Prueba de Yuen para dos muestras pareadas

Para el caso de dos muestras pareadas, podemos generalizar el estadístico de la prueba de Yuen con medias truncadas como muestra la ecuación 13.10, donde el nuevo término d_{12} está dado por la ecuación 13.11. En este caso, ambas muestras tienen igual tamaño n y h corresponde al tamaño de la muestra combinada tras la poda.

$$T_y = \frac{\bar{x}_{t1} - \bar{x}_{t2}}{\sqrt{d_1 + d_2 - 2 \cdot d_{12}}} \quad (13.10)$$

$$d_{12} = \frac{1}{h(h-1)} \sum_{i=1}^n (x_{i1} - \bar{x}_1) \cdot (x_{i2} - \bar{x}_2) \quad (13.11)$$

Supongamos ahora que queremos comparar el rendimiento de dos algoritmos X e Y, para lo cual hemos seleccionado aleatoriamente 25 instancias del problema y registrado su tiempo de ejecución en milisegundos con cada uno de los algoritmos. El script 13.7 ilustra el uso de la función `yuend(x, y, tr)` del paquete WRS2, que compara las medias truncadas, obteniéndose, para este ejemplo, el resultado que muestra la figura 13.15.

```
Call:
yuend(x = x, y = y, tr = gamma)

Test statistic: -4.5915 (df = 14), p-value = 0.00042

Trimmed mean difference: -0.76
95 percent confidence interval:
-1.115      -0.405

Explanatory measure of effect size: 0.44
```

Figura 13.15: resultado de la prueba de Yuen para el ejemplo, usando como estimadores la media y la mediana.

Script 13.7: prueba de Yuen para dos muestras pareadas.

```
1 library(WRS2)
2
3 # Construir data frame.
4 x <- c(32.0, 32.0, 32.0, 32.0, 32.1, 32.1, 32.1, 32.1, 32.2, 32.3, 32.3, 32.5,
5     32.7, 32.7, 32.7, 33.1, 33.4, 33.9, 34.1, 34.2, 34.5, 36.0, 36.6,
6     36.7, 37.2, 38.0)
7
8 y <- c(33.0, 33.0, 33.0, 33.0, 33.0, 33.0, 33.3, 33.3, 33.3, 33.3, 33.5,
9     33.6, 33.7, 33.9, 33.9, 34.2, 34.2, 34.3, 34.3, 34.4, 34.5, 34.6,
10    36.4, 38.9, 40.2)
11
12 # Fijar nivel de significación.
13 alfa <- 0.05
14
15 # Aplicar prueba de Yuen para muestras pareadas.
16 gamma <- 0.2
17 prueba <- yuend(x = x, y = y, tr = gamma)
18 print(prueba)
```

Puesto que el valor p obtenido, $p < 0.00042$, es menor que el nivel de significación, la evidencia es suficientemente fuerte como para rechazar la hipótesis nula en favor de la hipótesis alternativa. En consecuencia,

podemos afirmar con 95 % de confianza que existe una diferencia estadísticamente significativa en el desempeño de ambos algoritmos, siendo el algoritmo X el más eficiente (puesto que la diferencia estimada entre las medias tiene signo negativo).

13.2.4 Comparaciones de una vía para múltiples grupos independientes

El paquete WRS2 ofrece diferentes alternativas a ANOVA de una vía para muestras independientes que podemos usar cuando los tamaños muestrales son muy diferentes o no se cumple la condición de homocedasticidad.

La función `t1way(formula, data, tr, alpha)` efectúa un procedimiento similar a ANOVA usando medias truncadas. A su vez, la función `lincon(formula, data, tr, alpha)` permite realizar el procedimiento post-hoc correspondiente.

De manera similar, `t1waybt(formula, data, tr, nboot)` realiza un procedimiento análogo al anterior incorporando bootstrapping. En este caso, el procedimiento post-hoc puede realizarse mediante la función `mcppb20(formula, data, tr, nboot)`.

Una tercera opción es la función `mediway(formula, data, iter)`, que emplea la mediana y sigue un proceso iterativo. No obstante, en este caso el paquete no ofrece funciones que permitan realizar el procedimiento post-hoc.

Los argumentos asociados a las funciones mencionadas en los párrafos anteriores son:

- `formula`: de la forma `<variable dependiente>~<variable independiente>`.
- `data`: matriz de datos.
- `tr`: parámetro γ de la poda.
- `alpha`: nivel de significación.
- `nboot`: cantidad de muestras a generar mediante bootstrapping.
- `iter`: cantidad de iteraciones a realizar.

El script 13.8 ilustra el funcionamiento de algunas de las funciones descritas en los párrafos precedentes, suponiendo que ahora deseamos comparar el tiempo promedio de ejecución (en milisegundos) de tres algoritmos, contando con $n_a = 40$ observaciones para el algoritmo A, $n_b = 30$ observaciones para el algoritmo B y $n_c = 35$ observaciones para el algoritmo C. Se ha establecido para este estudio un nivel de significación $\alpha = 0,05$. Podemos ver en las figuras 13.16 y 13.17 que las funciones `t1way()` y `t1waybt()` arrojan el mismo resultado. Puesto que el valor p obtenido, $p = 0,00021$, es menor que el nivel de significación, rechazamos la hipótesis nula en favor de la hipótesis alternativa. Concluimos, entonces, con 95 % de confianza, que existe una diferencia estadísticamente significativa entre los tiempos promedio de ejecución de los algoritmos.

Al efectuar los procedimientos post-hoc respectivos, los valores p obtenidos son ligeramente diferentes para ambos métodos (figuras 13.16 y 13.17). No obstante, en ambos casos podemos concluir que el algoritmo C presenta un tiempo de ejecución promedio diferente.

Si examinamos las medias de cada grupo, tenemos que $\bar{x}_A = 27,19$ [ms], $\bar{x}_B = 26,58$ [ms] y $\bar{x}_C = 25,52$ [ms], por lo que el algoritmo C es más rápido.

Script 13.8: alternativas robustas para comparar entre múltiples grupos independientes.

```

1 library(WRS2)
2
3 # Construir data frame.
4 a <- c(25.1, 25.2, 25.3, 25.3, 25.4, 25.4, 25.5, 25.5, 25.6, 25.8, 25.8,
5      25.9, 25.9, 26.0, 26.0, 26.2, 26.2, 26.2, 26.3, 26.4, 26.5, 26.5,
6      26.6, 26.7, 26.7, 26.9, 26.9, 27.0, 27.1, 27.3, 27.8, 28.4, 28.5,
7      29.0, 29.8, 30.2, 31.8, 31.9, 33.3, 33.7)

```

Comparación entre grupos usando medias truncadas

Call:

```
t1way(formula = tiempo ~ algoritmo, data = datos, tr = gamma,  
      alpha = alfa)
```

Test statistic: F = 10.9813

Degrees of freedom 1: 2

Degrees of freedom 2: 34.39

p-value: 0.00021

Explanatory measure of effect size: 0.48

Bootstrap CI: [0.28; 0.68]

Procedimiento post-hoc

Call:

```
lincon(formula = tiempo ~ algoritmo, data = datos, tr = gamma,  
       alpha = alfa)
```

	psihat	ci.lower	ci.upper	p.value
A vs. B	0.24583	-1.11867	1.61033	0.65252
A vs. C	1.49583	0.65571	2.33596	0.00022
B vs. C	1.25000	-0.02757	2.52757	0.03909

Figura 13.16: resultado de la comparación entre múltiples grupos independientes usando medias truncadas.

```
8  
9 b <- c(24.1, 24.4, 24.4, 24.5, 24.7, 24.8, 24.8, 25.1, 25.2, 25.2,  
10    25.3, 25.4, 25.7, 25.7, 26.3, 26.3, 26.4, 26.5, 27.2, 27.7, 28.3,  
11    28.4, 28.4, 28.6, 28.7, 29.6, 29.9, 30.1, 30.5)  
12  
13 c <- c(24.5, 24.5, 24.5, 24.5, 24.5, 24.5, 24.6, 24.6, 24.6, 24.6,  
14    24.6, 24.7, 24.7, 24.7, 24.7, 24.7, 24.8, 25.0, 25.0, 25.0, 25.2, 25.2,  
15    25.2, 25.2, 25.5, 25.7, 25.9, 26.2, 26.5, 26.5, 26.7, 27.0, 29.2,  
16    29.9, 30.1)  
17  
18 tiempo <- c(a, b, c)  
19 algoritmo <- c(rep("A", length(a)), rep("B", length(b)), rep("C", length(c)))  
20 datos <- data.frame(tiempo, algoritmo)  
21  
22 # Fijar nivel de significación.  
23 alfa <- 0.05  
24  
25 # Comparar los diferentes algoritmos usando medias truncadas.  
26 cat("Comparación entre grupos usando medias truncadas\n\n")  
27 gamma <- 0.2  
28  
29 set.seed(666)  
30  
31 medias_truncadas <- t1way(tiempo ~ algoritmo, data = datos, tr = gamma,  
32                               alpha = alfa)  
33  
34 print(medias_truncadas)
```

Comparación entre grupos usando bootstrap

```
Call:  
t1way(formula = tiempo ~ algoritmo, data = datos, tr = gamma,  
      alpha = alfa)  
  
Test statistic: F = 10.9813  
Degrees of freedom 1: 2  
Degrees of freedom 2: 34.39  
p-value: 0.00021  
  
Explanatory measure of effect size: 0.48  
Bootstrap CI: [0.28; 0.68]
```

Procedimiento post-hoc

```
Call:  
mcppb20(formula = tiempo ~ algoritmo, data = datos, tr = gamma,  
         nboot = muestras)  
  
      psihat ci.lower ci.upper p-value  
A vs. B 0.24583 -0.91667  1.61389 0.55656  
A vs. C 1.49583  0.73690  2.44464 0.00000  
B vs. C 1.25000  0.16270  2.34206 0.00801
```

Figura 13.17: resultado de la comparación entre múltiples grupos independientes usando medias truncadas con bootstrapping.

```
35  
36 if(medias_truncadas$p.value < alfa) {  
37   cat("\nProcedimiento post-hoc\n\n")  
38  
39   set.seed(666)  
40  
41   post_hoc <- lincon(tiempo ~ algoritmo, data = datos, tr = gamma,  
42                       alpha = alfa)  
43  
44   print(post_hoc)  
45 }  
46  
47 # Comparar los diferentes algoritmos usando bootstrap.  
48 cat("Comparación entre grupos usando bootstrap\n\n")  
49 muestras <- 999  
50  
51 set.seed(666)  
52  
53 bootstrap <- t1waybt(tiempo ~ algoritmo, data = datos, tr = gamma,  
54                         nboot = muestras)  
55  
56 print(medias_truncadas)  
57  
58 if(medias_truncadas$p.value < alfa) {  
59   cat("\nProcedimiento post-hoc\n\n")  
60 }
```

```

61   set.seed(666)
62
63 post_hoc <- mcppb20(tiempo ~ algoritmo, data = datos, tr = gamma,
64                      nboot = muestras)
65
66 print(post_hoc)
67 }

```

13.2.5 Comparaciones de una vía para múltiples grupos correlacionados

Desde luego, el paquete WRS2 también ofrece opciones robustas para reemplazar el procedimiento ANOVA de una vía para muestras correlacionadas, que podemos usar cuando los datos disponibles violan la condición de esfericidad.

La función `ranova(y, groups, blocks, tr)` efectúa un procedimiento similar a ANOVA usando medias truncadas, mientras que la función `rmmcp(y, groups, blocks, tr, alpha)` implementa el procedimiento post-hoc para dicha prueba. Por otra parte, `rmanovab(y, groups, blocks, tr, nboot)` realiza la misma tarea que `ranova()`, incorporando bootstrapping. En este caso, el procedimiento post-hoc está dado por la función `pairdepb(y, groups, blocks, tr, nboot)`. Los argumentos para esta familia de funciones son:

- `formula`: de la forma `<variable dependiente>~<variable independiente>`.
- `y`: vector con la variable dependiente.
- `groups`: vector que indica los grupos.
- `blocks`: vector que identifica los sujetos o bloques.
- `tr`: parámetro γ de la poda.
- `alpha`: nivel de significación.
- `nboot`: cantidad de muestras a generar mediante bootstrapping.

El script 13.9 muestra el uso de las funciones robustas sin bootstrapping para ANOVA de una vía con muestras correlacionadas. El ejemplo presentado aborda, una vez más, la comparación del desempeño de tres algoritmos, X, Y y Z. Para ello, se han seleccionado aleatoriamente 25 instancias del problema y se registra su tiempo de ejecución (en milisegundos) con cada uno de los algoritmos. Para este estudio consideraremos un nivel de significación $\alpha = 0,05$. Podemos ver los resultados obtenidos en la figura 13.18.

El valor p resultante, $p = 1 \cdot 10^{-5}$, indica que la evidencia es suficientemente fuerte para rechazar la hipótesis nula en favor de la hipótesis alternativa, por lo que realizamos el procedimiento post-hoc correspondiente. La conclusión, con 95 % de confianza, es que no todos los algoritmos tienen el mismo rendimiento promedio, siendo el algoritmo X más eficiente que los algoritmos Y y Z.

Script 13.9: alternativa robusta para comparar entre múltiples grupos correlacionados.

```

1 library(WRS2)
2 library(tidyverse)
3
4 # Construir data frame.
5 X <- c(32.0, 32.0, 32.0, 32.0, 32.1, 32.1, 32.1, 32.2, 32.3, 32.3, 32.5,
6       32.7, 32.7, 32.7, 33.1, 33.4, 33.9, 34.1, 34.2, 34.5, 36.0, 36.6,
7       36.7, 37.2, 38.0)
8
9 Y <- c(33.0, 33.0, 33.0, 33.0, 33.0, 33.0, 33.3, 33.3, 33.3, 33.3, 33.5,
10      33.6, 33.7, 33.9, 33.9, 34.2, 34.2, 34.3, 34.3, 34.4, 34.5, 34.6,
11      36.4, 38.9, 40.2)
12
13 Z <- c(32.0, 32.2, 32.5, 32.6, 32.7, 32.7, 32.7, 33.0, 33.2, 33.4, 33.6,

```

```

Call:
rmanova(y = tiempo, groups = algoritmo, blocks = instancia, tr = gamma)

Test statistic: F = 24.1706
Degrees of freedom 1: 1.5
Degrees of freedom 2: 20.96
p-value: 1e-05

Procedimiento post-hoc

Call:
rmmcp(y = tiempo, groups = algoritmo, blocks = instancia, tr = gamma,
       alpha = alfa)

      psihat ci.lower ci.upper p.value p.crit    sig
X vs. Y -0.85333 -1.16837 -0.53830 0.00000 0.0169 TRUE
X vs. Z -0.68667 -0.98245 -0.39089 0.00002 0.0250 TRUE
Y vs. Z -0.00667 -0.26776  0.25443 0.94566 0.0500 FALSE

```

Figura 13.18: resultado de las alternativa robusta para comparar entre múltiples grupos correlacionados.

```

14      33.6, 33.9, 34.1, 34.2, 34.4, 34.4, 34.5, 34.6, 34.7, 36.3, 36.6,
15      36.7, 38.9, 39.2)
16
17 instancia <- 1:length(X)
18 datos <- data.frame(instancia, X, Y, Z)
19
20 # Llevar data frame a formato largo.
21 datos <- datos %>% pivot_longer(c("X", "Y", "Z"), names_to = "algoritmo",
22                                     values_to = "tiempo")
23
24 datos[["algoritmo"]] <- factor(datos[["algoritmo"]])
25
26 # Fijar nivel de significación.
27 alfa <- 0.05
28
29 # Aplicar alternativa robusta para ANOVA de una vía con
30 # muestras correlacionadas.
31 gamma <- 0.2
32
33 prueba <- rmanova(y = datos[["tiempo"]], groups = datos[["algoritmo"]],
34                      blocks = datos[["instancia"]], tr = gamma)
35
36 print(prueba)
37
38 if(prueba$p.value < alfa) {
39   cat("\nProcedimiento post-hoc\n\n")
40
41   post_hoc <- rmmcp(y = datos[["tiempo"]], groups = datos[["algoritmo"]],
42                      blocks = datos[["instancia"]], tr = gamma, alpha = alfa)
43
44   print(post_hoc)
45 }

```

13.3 EJERCICIOS PROPUESTOS

1. ¿Para qué se usa la transformación logarítmica?
2. Explica por qué comparar medias aritméticas de datos en escala logarítmica compara las medias geométricas en la escala normal de la variable.
3. El paquete **rcompanion** proporciona una función para aplicar la escala de potencias de Tukey. Experimenta con su uso.
4. Explica la relación entre la escala de potencias de Tukey y la transformación Box-Cox.
5. El paquete **DescTools** proporciona funciones para aplicar la transformación Box-Cox. Experimenta con su uso.
6. En tus palabras, ¿qué es un estadístico robusto?
7. Menciona tres situaciones que complican las pruebas de hipótesis paramétricas tradicionales.
8. ¿Qué alternativas se mencionan para tratar datos problemáticos? En particular, ¿qué recomienda el capítulo para muestras pequeñas que muestran desviaciones de normalidad?
9. Explica, en tus propias palabras, las dos medidas de tendencia central robustas presentadas en este capítulo.
10. ¿Cómo funciona y para qué sirve la prueba de Yuen?
11. ¿Se pueden comparar dos medianas en vez de dos medias de grupos independientes?
12. Describe las funciones disponibles en el paquete **WRS2** para hacer análisis de varianza robusta (con medias y con medianas, e incluyendo procedimientos post-hoc) para grupos independientes.
13. ¿Existe una alternativa para medidas repetidas de la prueba de Yuen?
14. ¿Existe una alternativa robusta para hacer un análisis de varianza con medidas repetidas?

CAPÍTULO 14. REGRESIÓN LINEAL

En el capítulo 2 introdujimos los gráficos de dispersión como una herramienta que nos permite identificar posibles relaciones entre dos variables cuantitativas. En este capítulo estudiaremos la **regresión lineal simple** (RLS), herramienta que sistematiza esta idea, basándonos en los textos de Diez et al. (2017, pp. 331-355), Field et al. (2012, pp. 245-311) e Irizarry (2019, pp. 535-545).

La RLS asume que la relación entre dos variables, x e y , puede ser modelada mediante **una recta** de la forma que se presenta en la ecuación 14.1, donde:

- β_0 y β_1 son los parámetros del modelo lineal.
- x es la variable explicativa o **predictor** (variable independiente).
- y es la variable **de respuesta** o **de salida** (variable dependiente).

$$\hat{y} = \beta_0 + \beta_1 x \quad (14.1)$$

Llamamos **intercepción** (*intercept*, en inglés) al parámetro β_0 , que corresponde al punto en que la recta corta el eje y . A su vez, denominamos **pendiente** al parámetro β_1 , el cual determina la inclinación de la recta del modelo.

Si tuviéramos una relación lineal perfecta entre ambas variables, significaría que se podríamos conocer el valor exacto de y con solo conocer el valor de x . Sin embargo, como podemos apreciar en la figura 14.1, rara vez los datos se ajustan al modelo con exactitud.

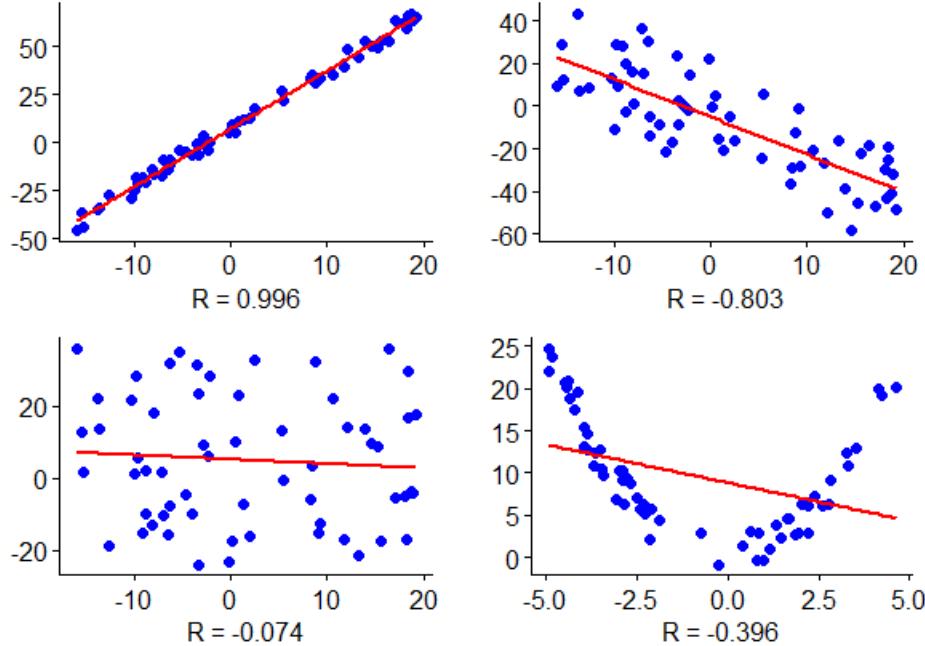


Figura 14.1: modelos lineales para cuatro conjuntos de datos.

Los gráficos de la fila superior de la figura 14.1 muestran dos tendencias lineales, siendo la izquierda una relación directa y muy fuerte, y la de la derecha, inversa y algo más débil. En el caso de los gráficos de la fila inferior, los datos en el de la izquierda no se aglutan en torno a la recta marcada, y los de la derecha, presentan un escenario donde ambas variables se relacionan clara y fuertemente, pero de manera no lineal.

Siempre tenemos que tener en cuenta que, si los datos presentan una tendencia no lineal, debemos usar herramientas más avanzadas que la regresión lineal simple.

Fijémonos en el siguiente modelo lineal, que corresponde a la línea roja en el gráfico de arriba a la izquierda de la figura 14.1:

$$\hat{y} = 7 + 3x$$

En él, si $x = 5$, entonces $\hat{y} = 22$. \hat{y} es un estimador que podemos entender de la siguiente manera: dado un valor de x , el valor de y es, en promedio, \hat{y} . En otras palabras, \hat{y} corresponde al valor esperado de y para un determinado valor de x . En la práctica, existe una diferencia entre el valor esperado \hat{y} y el valor observado de y . Esta diferencia se denomina **residuo** y se denota e . Así, tenemos que el valor observado de y está dado por la ecuación 14.2.

$$y = \hat{y} + e \quad (14.2)$$

Otra forma de entender el residuo es como la distancia que separa a la observación de la recta. Si la observación se encuentra por sobre esta última, entonces $e > 0$. En caso contrario, $e < 0$. Puesto que los residuos sirven para evaluar qué tan bien se ajusta un modelo lineal al conjunto de datos, suelen mostrarse en un **gráfico de residuos**, el cual es sencillamente un gráfico de dispersión donde la variable predictora se representa en su escala original y el eje y muestra el residuo para cada observación. La figura 14.2 muestra los residuos para los modelos lineales de la figura 14.1.

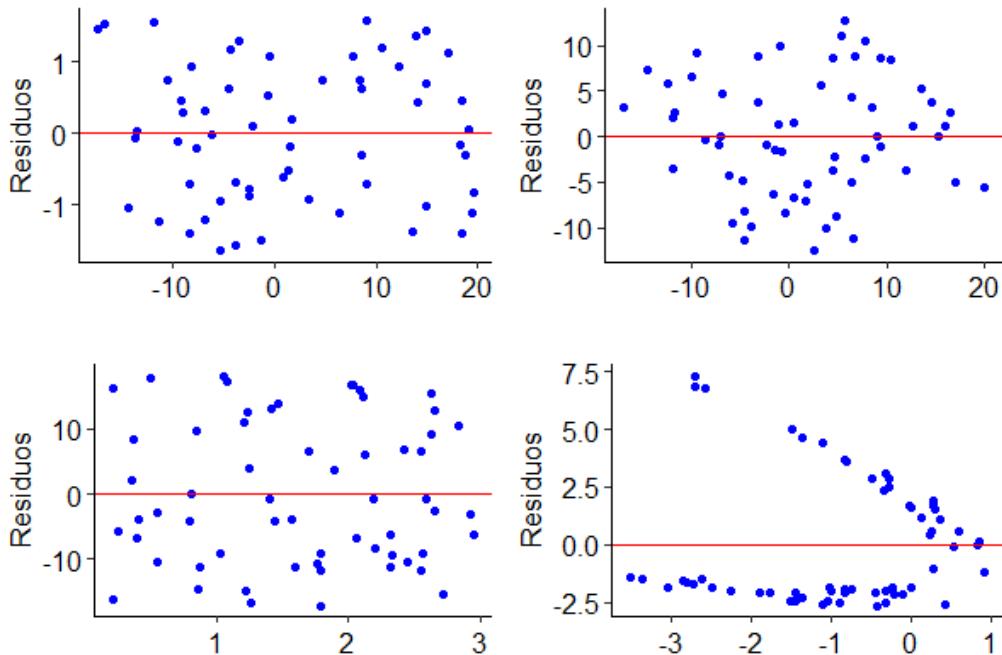


Figura 14.2: residuos para los modelos lineales de la figura 14.1.

14.1 CORRELACIÓN

Hasta ahora hemos hablado de la fuerza de una relación lineal entre dos variables, concepto que hemos asociado implícitamente a la magnitud de los residuos. Formalmente, podemos medir la fuerza de una relación lineal mediante la **correlación**. Una de las formas más sencillas para calcularla es el coeficiente de correlación de Pearson, dado por la ecuación 14.3, donde:

- \bar{x}, \bar{y} son las medias de las variables x e y en la muestra.
- s_x, s_y corresponden a las desviaciones estándar de las de las variables x e y en la muestra.
- n es el tamaño de la muestra.

$$R = \frac{1}{n-1} \cdot \sum_{i=1}^n \frac{x_i - \bar{x}}{s_x} \cdot \frac{y_i - \bar{y}}{s_y} \quad (14.3)$$

La correlación siempre toma un valor entre -1 y 1. Mientras más débil sea la relación entre dos variables, su valor será más cercano a 0. El signo de la correlación indica si la relación es directa ($R > 0$) o inversa ($R < 0$). Para comprender mejor esta idea, fíjemonos los coeficientes de correlación obtenidos para cada modelo lineal de la figura 14.1, indicados en las etiquetas del eje x . Podemos ver que, en el caso de abajo a la derecha, la relación es muy fuerte, pero no lineal, por lo que R , al considerar solo una recta, toma un valor relativamente bajo.

Para los ejemplos de este capítulo usaremos el conjunto de datos `mtcars`, disponible en R, que contiene diversas características para $n = 32$ modelos de automóviles de los años 1973 y 1974. La tabla 14.1 describe brevemente cada una de las variables de dicho conjunto.

Columna	Descripción
mpg	Rendimiento, en millas (EEUU) por galón [millas/galón].
cyl	Cantidad de cilindros del motor.
disp	Volumen útil de los cilindros de un motor, en centímetros cúbicos [cc].
hp	Potencia del motor, en caballos de fuerza [hp].
drat	Relación del eje trasero (proporción).
wt	Peso total, en miles de libras.
qsec	Tiempo mínimo para recorrer un cuarto de milla (desde el reposo), en segundos [s].
vs	Tipo de motor (0 = en forma de V, 1 = recto).
am	Transmisión (0 = automática, 1 = manual).
gear	Número de marchas hacia adelante.
carb	Número de carburadores.

Tabla 14.1: descripción de las variables para el conjunto de datos `mtcars` usados en este capítulo.

Si consideramos a x como el rendimiento del vehículo y a y como la potencia del motor, cuya relación se muestra gráficamente en la figura 14.3, tenemos que: $\bar{x} = 20,091$, $s_x = 6,027$, $\bar{y} = 146,688$ y $s_y = 68,563$. En consecuencia, la correlación es:

$$R = \frac{1}{32-1} \cdot \sum_{i=1}^n \frac{x_i - 20,091}{6,027} \cdot \frac{y_i - 146,688}{68,563} = -0,776$$

En R, podemos calcular la correlación entre dos variables usando la función `cor(x, y)`, donde x es el predictor e y la respuesta. Para el ejemplo obtenemos que $R = -0,7761684$, lo que coincide con el resultado teórico teniendo en cuenta que la diferencia se debe únicamente al redondeo.

Adicionalmente, cuando x es una matriz de datos, la función `cor(x)` nos entrega una **matriz de correlación**, que contiene las correlaciones entre todos los pares de variables. La figura 14.4 muestra la matriz de correlación

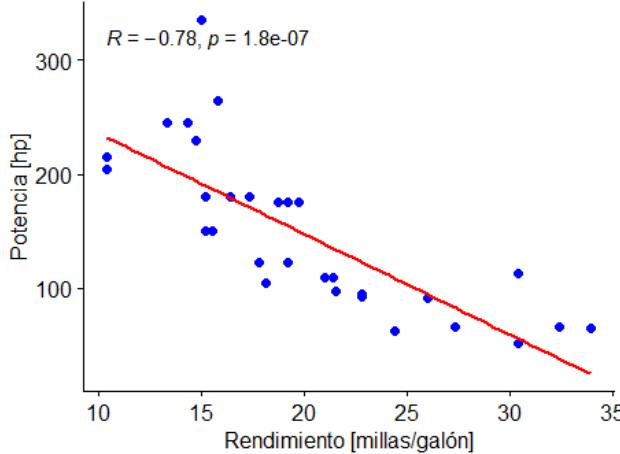


Figura 14.3: Relación entre el rendimiento y la potencia.

(redondeada al segundo decimal) para el conjunto de datos `mtcars`. Podemos ver que, naturalmente, la matriz de correlación es simétrica y que su diagonal solo contiene unos.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1.00	-0.85	-0.85	-0.78	0.68	-0.87	0.42	0.66	0.60	0.48	-0.55
cyl	-0.85	1.00	0.90	0.83	-0.70	0.78	-0.59	-0.81	-0.52	-0.49	0.53
disp	-0.85	0.90	1.00	0.79	-0.71	0.89	-0.43	-0.71	-0.59	-0.56	0.39
hp	-0.78	0.83	0.79	1.00	-0.45	0.66	-0.71	-0.72	-0.24	-0.13	0.75
drat	0.68	-0.70	-0.71	-0.45	1.00	-0.71	0.09	0.44	0.71	0.70	-0.09
wt	-0.87	0.78	0.89	0.66	-0.71	1.00	-0.17	-0.55	-0.69	-0.58	0.43
qsec	0.42	-0.59	-0.43	-0.71	0.09	-0.17	1.00	0.74	-0.23	-0.21	-0.66
vs	0.66	-0.81	-0.71	-0.72	0.44	-0.55	0.74	1.00	0.17	0.21	-0.57
am	0.60	-0.52	-0.59	-0.24	0.71	-0.69	-0.23	0.17	1.00	0.79	0.06
gear	0.48	-0.49	-0.56	-0.13	0.70	-0.58	-0.21	0.21	0.79	1.00	0.27
carb	-0.55	0.53	0.39	0.75	-0.09	0.43	-0.66	-0.57	0.06	0.27	1.00

Figura 14.4: matriz de correlación para el conjunto de datos `mtcars`

14.2 REGRESIÓN LINEAL MEDIANTE MÍNIMOS CUADRADOS

Si bien existen diversos métodos para ajustar un modelo lineal, el más empleado es el de la **línea de mínimos cuadrados**, que minimiza la suma de los cuadrados de los residuos (ecuación 14.4).

$$\min \sum_{i=1}^n e_i^2 \quad (14.4)$$

El método de mínimos cuadrados tiene las ventajas de ser fácil de calcular y de tomar en cuenta la discrepancia entre la magnitud del residuo y su efecto. Como señalan Diez et al. (2017, p. 341), “por ejemplo, desviarse por 4 suele ser más de dos veces peor que desviarse por 2”. No obstante, para aplicar este método debemos verificar que se cumplan algunas condiciones:

1. Los datos deben presentar una relación lineal.
2. La distribución de los residuos debe ser cercana a la normal.
3. La variabilidad de los puntos en torno a la línea de mínimos cuadrados debe ser aproximadamente constante.
4. Las observaciones deben ser independientes entre sí. Esto significa que no se puede usar regresión lineal con series de tiempo (tema que va más allá de los alcances de este texto).

Los gráficos de residuos reflejan cuando no se cumplen las condiciones anteriores. Por ejemplo, el gráfico inferior derecho de la figura 14.1 aplica regresión lineal entre un par de variables cuya relación es, en realidad, cuadrática. Esta relación se puede ver en la forma en que se distribuyen los puntos en el gráfico de residuos correspondiente de la figura 14.2.

La figura 14.5 muestra, en la fila superior, dos modelos lineales en los que los datos no cumplen las condiciones, con sus respectivos gráficos de residuos en la fila inferior. A la izquierda, se viola la condición de normalidad de los residuos, que no se distribuyen aleatoriamente en torno a la línea 0. A la derecha, no se respeta la condición de homocedasticidad, y la variabilidad de los puntos en torno a la línea de mínimos cuadrados no es aproximadamente constante, lo que genera una característica forma de embudo en el gráfico de residuos.

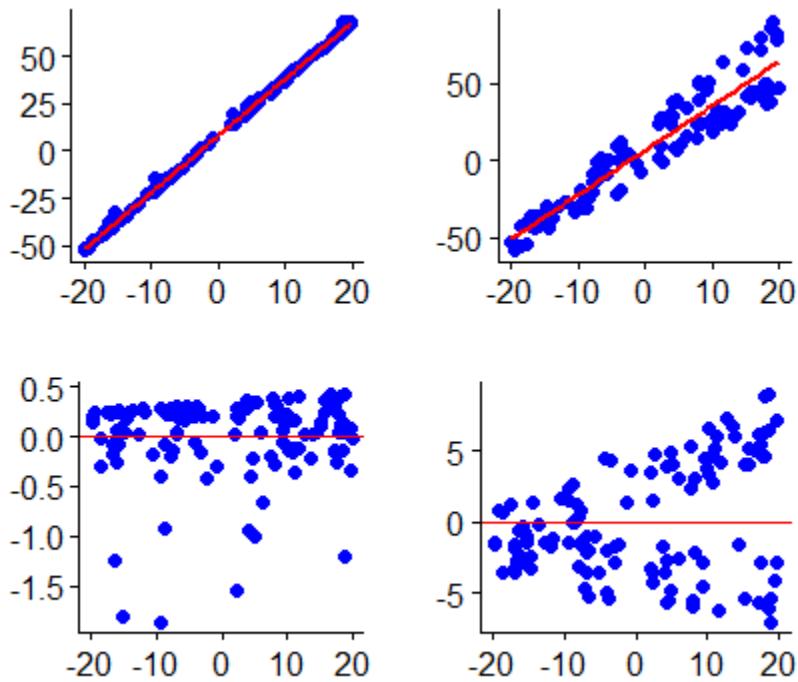


Figura 14.5: modelos lineales (fila superior) que violan alguna condición y sus residuos (fila inferior).

El primer paso que debemos seguir cuando queremos determinar la recta de mínimos cuadrados para un conjunto de datos consiste en estimar la pendiente (β_1) mediante la ecuación 14.5, donde:

- s_x y s_y son las desviaciones estándares muestrales de las variables x e y , respectivamente.
- R corresponde a la correlación entre ambas variables.

$$b_1 = \frac{s_y}{s_x} \cdot R \quad (14.5)$$

El punto (\bar{x}, \bar{y}) , donde \bar{x} e \bar{y} son las medias muestrales para las variables representadas en los ejes x e y respectivamente, siempre pertenece a la recta de mínimos cuadrados, por lo que podemos calcular la intercepción mediante la ecuación 14.6 (Winner, 2021, p. 4).

$$b_0 = \bar{y} - b_1 \cdot \bar{x} \quad (14.6)$$

Cuando contamos con más de una variable para construir una regresión lineal simple (RLS), lo más adecuado es que escogamos como predictor aquella variable que tenga la correlación más fuerte con la variable de respuesta. Para el caso del conjunto de datos `mtcars`, la variable que presenta la correlación más fuerte con el rendimiento (`mpg`) es el peso del automóvil (`wt`), con $R = -0,87$, como podemos ver en la figura 14.4. Así, si usamos como predictor la variable peso, tenemos que la pendiente de la recta es:

$$b_1 = \frac{6,027}{0,978} \cdot -0,868 = -5,344$$

A su vez, la intercepción está dada por:

$$b_0 = 20,091 + 5,344 \cdot 3,217 = 37,285$$

Por lo que la recta ajustada mediante mínimos cuadrados es:

$$\widehat{\text{mpg}} = 37,285 - 5,344 \cdot \text{wt}$$

Desde luego, R ofrece una función que permite ajustar la recta de mínimos cuadrados para un par de variables: `lm(formula, data)`, donde:

- **formula:** tiene la forma <variable de respuesta>~<variable predictora>.
- **data:** matriz de datos.

El script 14.1 ajusta la línea de mínimos cuadrados para la variable de respuesta rendimiento (`mpg`), con la variable peso (`wt`) como predictor, mediante el uso de `lm()`. En la figura 14.6, bajo el encabezado `Coefficients`, podemos ver que los valores estimados para los parámetros de la RLS coinciden con los obtenidos previamente.

Un detalle interesante es que, en la línea 8 del script 14.1, usamos la llamada `print(summary(modelo))` en lugar de `print(modelo)`, lo que nos entrega información más detallada del modelo ajustado (figura 14.6). La segunda solo muestra los coeficientes obtenidos.

```

Call:
lm(formula = mpg ~ wt, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.5432 -2.3647 -0.1252  1.4096  6.8727 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 37.2851    1.8776 19.858 < 2e-16 ***
wt          -5.3445    0.5591 -9.559 1.29e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446 
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10

```

Figura 14.6: regresión lineal simple para predecir el rendimiento de un automóvil a partir de su peso.

La figura 14.7 muestra la recta ajustada para el rendimiento de un automóvil de acuerdo a su peso (script 14.1, líneas 11–15), donde podemos observar que los datos presentan una relación lineal, aunque algunos puntos parecen estar algo alejados de la recta ajustada. A su vez, la figura 14.8 muestra diversos gráficos obtenidos mediante la línea 18 del script 14.1, donde vemos que la variabilidad de los residuos no es muy grande (figura 14.8a) y, en general, sigue una distribución razonablemente cercana a la normal (figura 14.8b), aunque en ambas figuras se aprecian unos pocos modelos que se comportan como valores atípicos. Además, podemos suponer que las observaciones son independientes entre sí y, evidentemente, no corresponden a una serie de tiempo. Con este análisis verificamos las condiciones 1 y 4 para emplear la regresión lineal de mínimos cuadrados, aunque las dos condiciones restantes parecen no cumplirse.

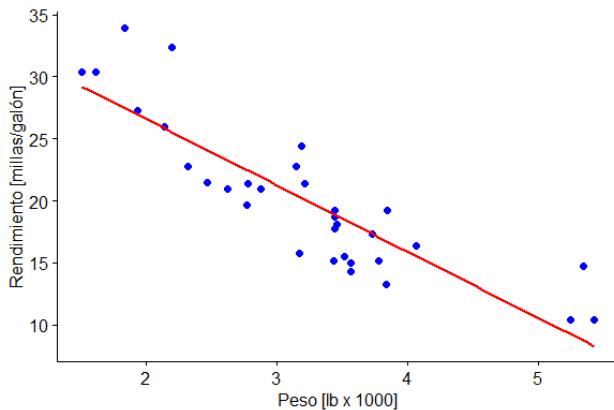


Figura 14.7: recta ajustada para el rendimiento de un automóvil de acuerdo a su peso.

Si bien para fines del ejercicio supondremos que las condiciones se cumplen, vale la pena que revisemos algunas características que, según Pardoe et al. (2018), se observan en el gráfico de los residuos cuando sí se verifican todas las condiciones o, en otras palabras, cuando el modelo de RLS es apropiado:

1. Un gráfico en que los residuos se distribuyen aleatoriamente en torno a la línea de valor 0, sugiere que es razonable suponer que las variables presentan una relación lineal.
2. Cuando los residuos forman una “banda horizontal” en torno a la línea de valor 0, sugiere una variabilidad aproximadamente constante de los residuos.
3. La ausencia de residuos que se alejen del patrón que forman los demás sugiere la ausencia de valores atípicos.

Script 14.1: ajuste de una regresión lineal simple.

```

1 library(ggpubr)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Ajustar modelo con R.
7 modelo <- lm(mpg ~ wt, data = datos)
8 print(summary(modelo))
9
10 # Graficar el modelo.
11 p <- ggscatter(datos, x = "wt", y = "mpg", color = "blue", fill = "blue",
12                  xlab = "Peso [lb x 1000]", ylab = "Rendimiento [millas/galón]")
13
14 p <- p + geom_smooth(method = lm, se = FALSE, colour = "red")
15 print(p)
16
17 # Crear gráficos para evaluar el modelo.
18 plot(modelo)
19
```

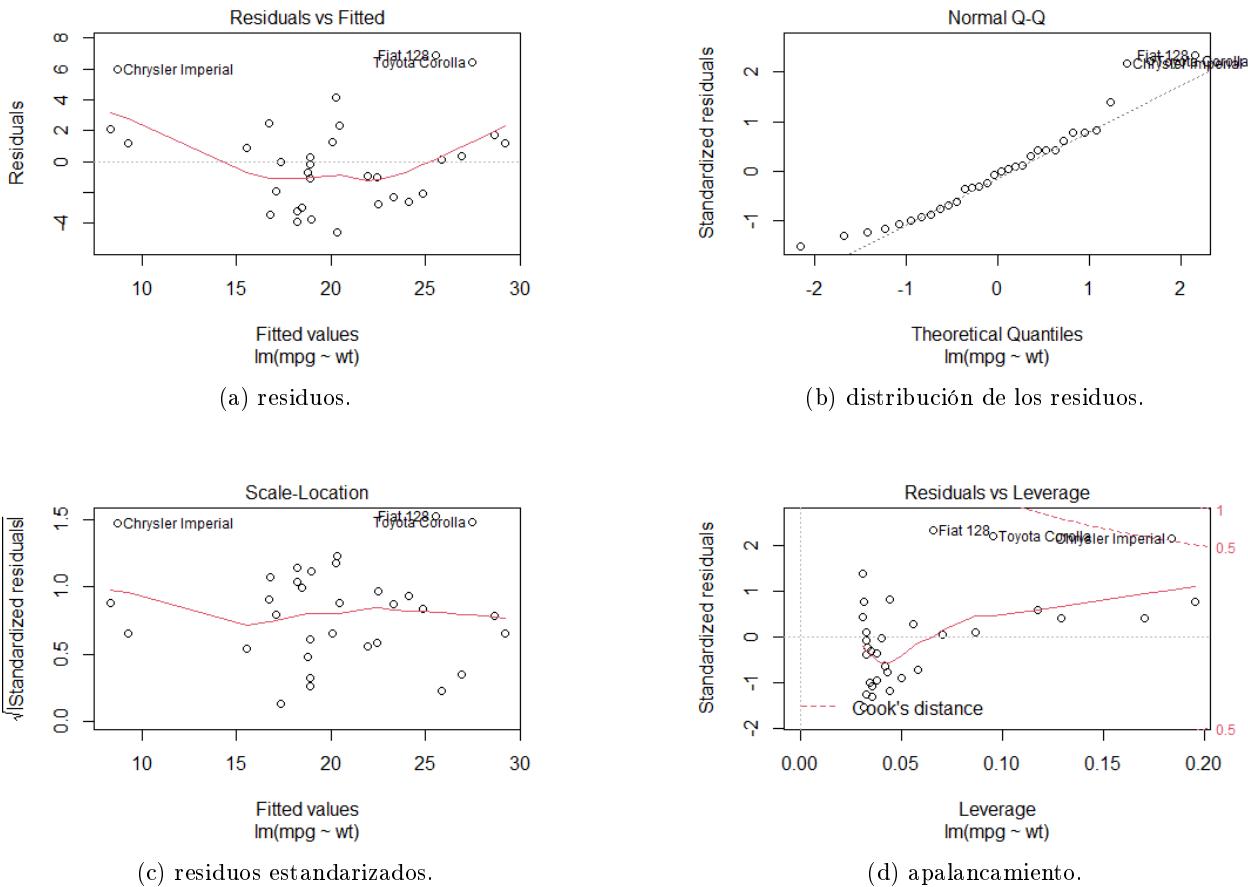


Figura 14.8: gráficos para evaluar el modelo lineal.

```

20 # Ingresar algunas instancias artificiales.
21 mpg <- c(23.714, 19.691, 19.242, 12.430, 10.090, 9.565, 18.171, 26.492, 7.054,
22           24.447, 15.683, 17.403, 13.465, 18.850, 29.493)
23
24 wt <- c(2.973, 4.532, 2.332, 3.016, 4.220, 4.286, 2.580, 3.084, 3.816, 2.775,
25           3.251, 3.013, 4.951, 2.644, 2.218)
26
27 nuevos <- data.frame(mpg, wt)
28
29 # Usar el modelo para predecir el rendimiento de los nuevos y ver los
30 # residuos resultantes.
31 predicciones <- predict(modelo, nuevos)
32 residuos <- nuevos$mpg - predicciones
33 nuevos <- data.frame(nuevos, residuos)
34
35 r <- ggscatter( nuevos, x = "wt", y = "residuos", color = "blue",
36                 fill = "blue", xlab = "Peso [lb * 1000]", ylab = "Residuo")
37
38 r <- r + geom_hline(yintercept = 0, colour = "red")
39 print(r)

```

Una de las etapas más importantes en un proceso de análisis es la **interpretación de los parámetros** del modelo. La pendiente explica la diferencia esperada en el valor de la respuesta y si el predictor x se incrementa

en una unidad. Así, para el ejemplo se espera que, al incrementar en el peso del automóvil en 1.000 libras, el rendimiento se reduzca en 5,344 millas por galón de combustible. A su vez, la intercepción corresponde a la respuesta que se obtendría en promedio si x fuese igual a 0, suponiendo que el modelo fuese válido para $x = 0$, lo que no siempre ocurre. De hecho, para el ejemplo, es imposible que un automóvil carezca de masa.

El párrafo anterior ilustra una limitación propia de cualquier modelo: este, al ser una simplificación de la realidad, tiene validez únicamente en el rango de valores de los datos originales, por lo que la **extrapolación** (es decir, estimar valores fuera del rango de los datos originales) puede conllevar a errores al asumir que el modelo es válido donde aún no ha sido analizado. El peso de los automóviles del ejemplo varía entre 1.500 y 5.500 libras aproximadamente, por lo que si lo usáramos para predecir el rendimiento de un vehículo de 7.000 libras o de solo 980, el resultado podría carecer de validez.

Desde luego, debemos tener en cuenta también las condiciones de diseño del modelo. El resultado podría ser equivocado si intentáramos predecir, por ejemplo, el rendimiento de un automóvil moderno (recordemos que el conjunto de datos solo contiene vehículos de los años 1973 y 1974). Más aún, la variable de respuesta carece absolutamente de sentido si pensamos, por ejemplo, en un automóvil eléctrico.

14.3 USO DEL MODELO

Supongamos que queremos predecir el rendimiento de un auto norteamericano (modelo 1974) cuyo peso es de 4.260 libras (es decir, $wt = 4,260$). Para ello, basta con reemplazar el valor del predictor en el modelo:

$$\widehat{mpg} = 37,285 - 5,344 \cdot 4,260 = 14,520$$

En R, la función `predict(object, newdata)` nos permite usar un modelo (en este caso, una RLS) para predecir una respuesta. Los argumentos de esta función son:

- **object:** el modelo a emplear.
- **newdata:** matriz de datos con las nuevas instancias para las que se desea efectuar la predicción, la cual debe tener todas las columnas presentes en la fórmula del modelo (para el ejemplo, `mpg` y `wt`).

La línea 31 del script 14.1 ilustra el uso de esta función para un conjunto de 15 instancias generadas artificialmente.

En la línea 32 se calculan los residuos para posteriormente graficarlos (líneas 35–39) a fin de tener una idea preliminar acerca de la calidad de las predicciones. Como resultado obtenemos la figura 14.9, donde podemos observar que los residuos varían en un rango bastante más amplio que para el conjunto de datos original (figura 14.8a).

14.4 REGRESIÓN LINEAL CON UN PREDICTOR CATEGÓRICO

Las variables categóricas también nos pueden servir para predecir una respuesta. En este capítulo solo estudiaremos el caso de una variable dicotómica (es decir, con solo dos niveles), idea que profundizaremos en el siguiente capítulo.

Para usar una variable categórica con dos niveles, tenemos que convertirla a formato numérico, para lo cual creamos una nueva **variable indicadora** que toma los valores 0 y 1. Hacer este proceso en R es bastante

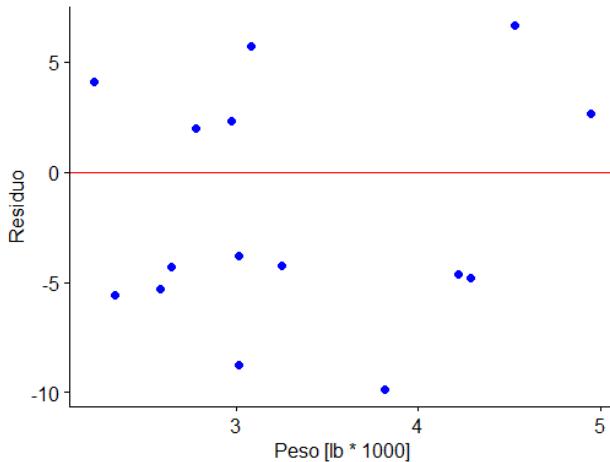


Figura 14.9: residuos obtenidos tras usar el modelo para predecir el rendimiento de nuevos automóviles.

sencillo, como muestra el script 14.2. En la práctica, rara vez tendremos que realizar este paso, pues las funciones de R que ajustan modelos lo hacen automáticamente cuando encuentran predictores categóricicos.

Script 14.2: reemplazar una variable dicotómica por una variable indicadora.

```

1 # Crear un data frame con una variable dicotómica.
2 alumno <- 1:5
3 sexo <- factor(c("F", "M", "F", "F", "M"))
4 datos <- data.frame(alumno, sexo)
5
6 # Crear una variable indicadora para sexo, con valor 0
7 # para hombres y 1, para mujeres.
8 es_mujer <- rep(1, length(sexo))
9 es_mujer[sexo == "M"] <- 0
10
11 # Reemplazar la variable sexo por la variable indicadora.
12 datos <- cbind(datos, es_mujer)
13 datos[["sexo"]] <- NULL

```

El conjunto de datos **mtcars** ya cuenta con un par de variables que cumplen con esta característica: la transmisión (**am**) y la forma del motor (**vs**). De estas dos variables, la forma del motor tiene una correlación más fuerte con el rendimiento, por lo que la usaremos como ejemplo para crear un modelo RLS. Al crear el modelo (script 14.3) obtenemos como resultado la recta representada en el gráfico superior de la figura 14.10, con los residuos del gráfico inferior en la misma figura. A su vez, la figura 14.11 muestra los valores obtenidos para los parámetros del modelo.

Cuando usamos un predictor dicotómico siempre se cumple la condición de que los datos presentan una relación lineal. Sin embargo, debemos verificar que la distribución de los residuos de ambos grupos se asemeje a la normal y que tengan varianzas similares. El panel superior de la figura 14.10 muestra que, en efecto, las variabilidades de los residuos de ambos grupos son independientes y la figura 14.12 muestra que la distribución de los residuos se acerca a la normal para ambos tipos de transmisión, por lo que se verifican las condiciones.

Script 14.3: alternativa robusta para comparar entre múltiples grupos correlacionados.

```

1 library(ggpubr)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Ajustar modelo con R.

```

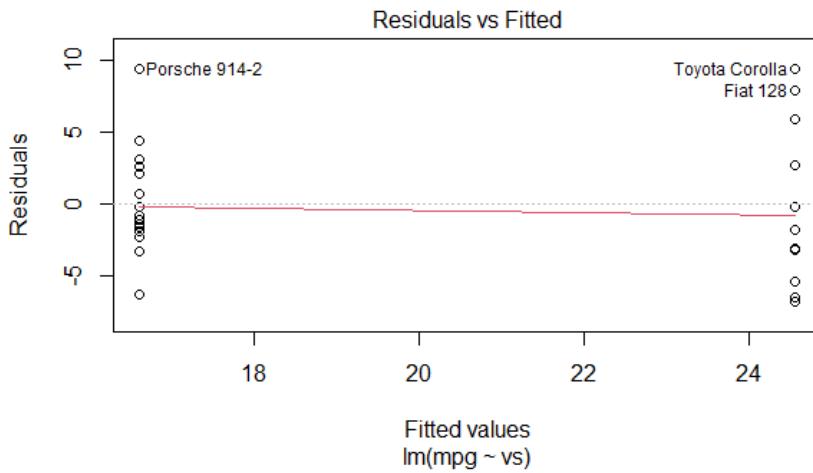
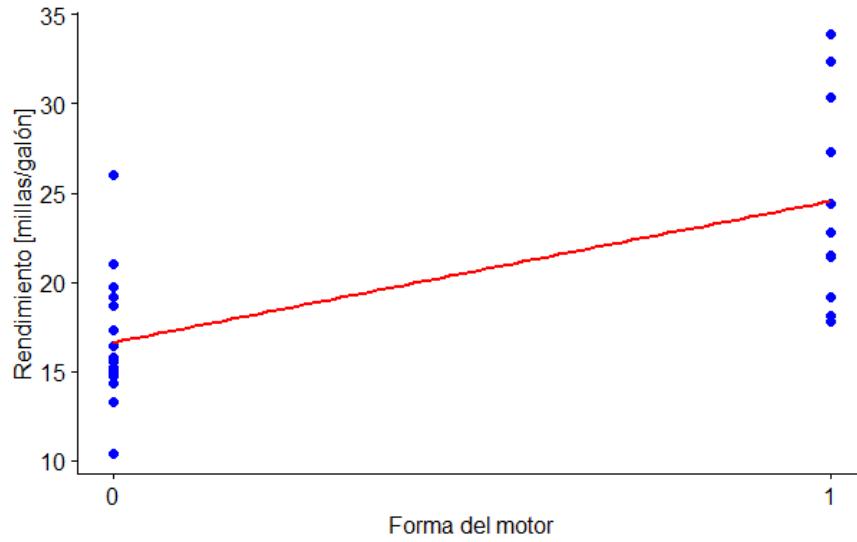


Figura 14.10: modelo de regresión lineal y gráfico de residuos para el ejemplo con un predictor dicotómico.

```

7 modelo <- lm(mpg ~ vs, data = datos)
8 print(summary(modelo))
9
10 # Graficar el modelo.
11 p <- ggscatter(datos, x = "vs", y = "mpg", color = "blue", fill = "blue",
12                 xlab = "Forma del motor", ylab = "Rendimiento [millas/galón]",
13                 xticks.by = 1)
14
15 p <- p + geom_smooth(method = lm, se = FALSE, colour = "red")
16 print(p)
17
18 # Crear gráficos para evaluar el modelo.
19 plot(modelo)
20
21 # Graficar residuos.
22 residuos <- modelo$residuals
23 datos <- cbind(datos, residuos)
24 datos[["vs"]] <- factor(datos[["vs"]])

```

```

Call:
lm(formula = mpg ~ vs, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.757 -3.082 -1.267  2.828  9.383 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 16.617     1.080   15.390 8.85e-16 ***  
vs           7.940     1.632    4.864 3.42e-05 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.581 on 30 degrees of freedom
Multiple R-squared:  0.4409, Adjusted R-squared:  0.4223 
F-statistic: 23.66 on 1 and 30 DF,  p-value: 3.416e-05

```

Figura 14.11: recta de mínimos cuadrados para el ejemplo con un predictor dicotómico.

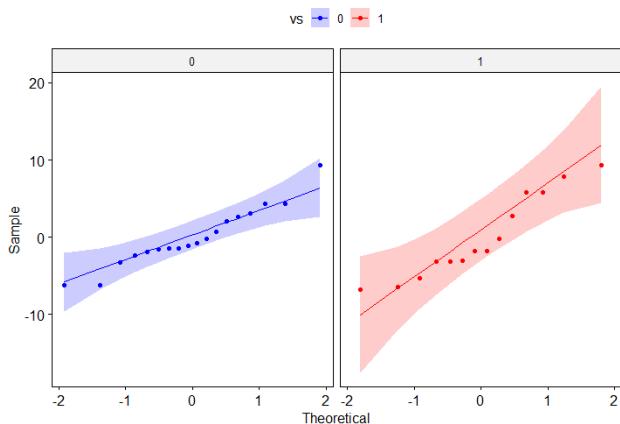


Figura 14.12: distribución de los residuos.

```

25
26 r <- ggqqplot(datos, x = "residuos", facet.by = "vs", color = "vs",
27                 palette = c("blue", "red"))
28
29 print(r)

```

14.5 EVALUACIÓN DE UN MODELO DE RLS

Hasta ahora hemos visto cómo ajustar y usar un modelo de RLS. Sin embargo, no hemos realizado algunas verificaciones importantes antes de hacer predicciones, pues puede ocurrir que la recta ajustada esté fuertemente influenciada por un pequeño grupo de valores atípicos o que no pueda generalizarse para otras muestras.

14.5.1 Influencia de los valores atípicos

Hemos dicho que, en muchas ocasiones, los valores atípicos influyen significativamente en el incumplimiento de las condiciones que debemos verificar para poder usar una regresión lineal. Sin embargo, no todos los valores atípicos son perjudiciales. La figura 14.13 muestra, para seis conjuntos de datos, los gráficos de dispersión (incluyendo la línea de regresión) y sus respectivos gráficos de los residuos. En cada uno de ellos se evidencia la presencia de al menos un valor atípico. Como señalan Diez et al. (2017, p. 349):

- En (1) hay un valor atípico que se aleja mucho de la nube de puntos, pero que no parece tener mucha influencia en la línea de regresión.
- En (2), se observa un valor atípico, a la derecha y bastante cercano a la línea de regresión, que no parece tener gran influencia.
- Nuevamente aparece un valor atípico a la derecha en (3), el cual parece ser el causante de que la línea de regresión no se ajuste muy bien a la nube principal de puntos.
- En (4), los datos se agrupan en dos nubes, una principal y la otra (secundaria) con cuatro valores atípicos. La nube secundaria parece influenciar fuertemente la línea de regresión, haciendo que se ajuste pobremente a los datos de la nube principal.
- La nube principal no evidencia tendencia alguna (pendiente cercana a cero) en (5), y el valor atípico a la derecha parece ejercer una gran influencia en la línea de regresión.
- En (6) se observa un valor atípico a la izquierda que se aleja bastante de la nube principal. Sin embargo, no parece ejercer mucha influencia en la línea de regresión y se sitúa cerca de ella.

Los valores atípicos que se alejan horizontalmente del centro de la nube principal de puntos pueden, potencialmente, tener una gran influencia en el ajuste de la línea de regresión. Este fenómeno se conoce como **apalancamiento** (*leverage* en inglés), pues dichos puntos parecen tirar de la línea hacia ellos. Cuando un valor atípico ejerce efectivamente esta influencia, decimos que es un **punto influyente**. Una forma de saber si un punto es o no influyente es determinar la línea de regresión sin considerar dicho punto y ver cuánto se aleja este último de la nueva línea. Si miramos la figura 14.8d, podemos detectar dos observaciones atípicas que influyen en el ajuste del modelo.

Si bien puede resultar tentador descartar los valores atípicos antes de ajustar un modelo, no es pertinente llevar a cabo esta acción sin hacer un riguroso análisis previo. En muchos casos los valores atípicos resultan ser las observaciones más interesantes. Diez et al. (2017, p. 349) ilustran esta idea con el ejemplo de acciones de la bolsa con valores excepcionalmente altos. Si omitieran estos valores atípicos, los agentes de bolsa perderían los mejores negocios.

Un buen método para identificar valores atípicos es usar los residuos estandarizados (figura 14.8c) (es decir, divididos por la estimación de su desviación estándar), pues esto nos permite establecer un rango fijo de valores aceptables y, en consecuencia, fijar un criterio para comparar residuos de distintos modelos.

Debemos ser cuidadosos cuando usemos como predictores variables categóricas que tengan pocas observaciones en alguno de sus niveles, pues cuando esto ocurre, dichas observaciones se convierten en puntos influyentes.

14.5.2 Bondad de ajuste

Una medida muy útil que podemos usar para evaluar la **bondad de ajuste** de un modelo de regresión lineal con respecto a las observaciones es el **coeficiente de determinación**, que corresponde al cuadrado de la correlación, por lo que suele también denominarse R-cuadrado (R^2) (Glen, 2021a). Esta medida, cuyo valor varía entre 0 y 1, corresponde al porcentaje de la variabilidad de la respuesta que es explicado por el predictor, dado por la ecuación 14.7, donde s_e^2 es la varianza de los residuos.

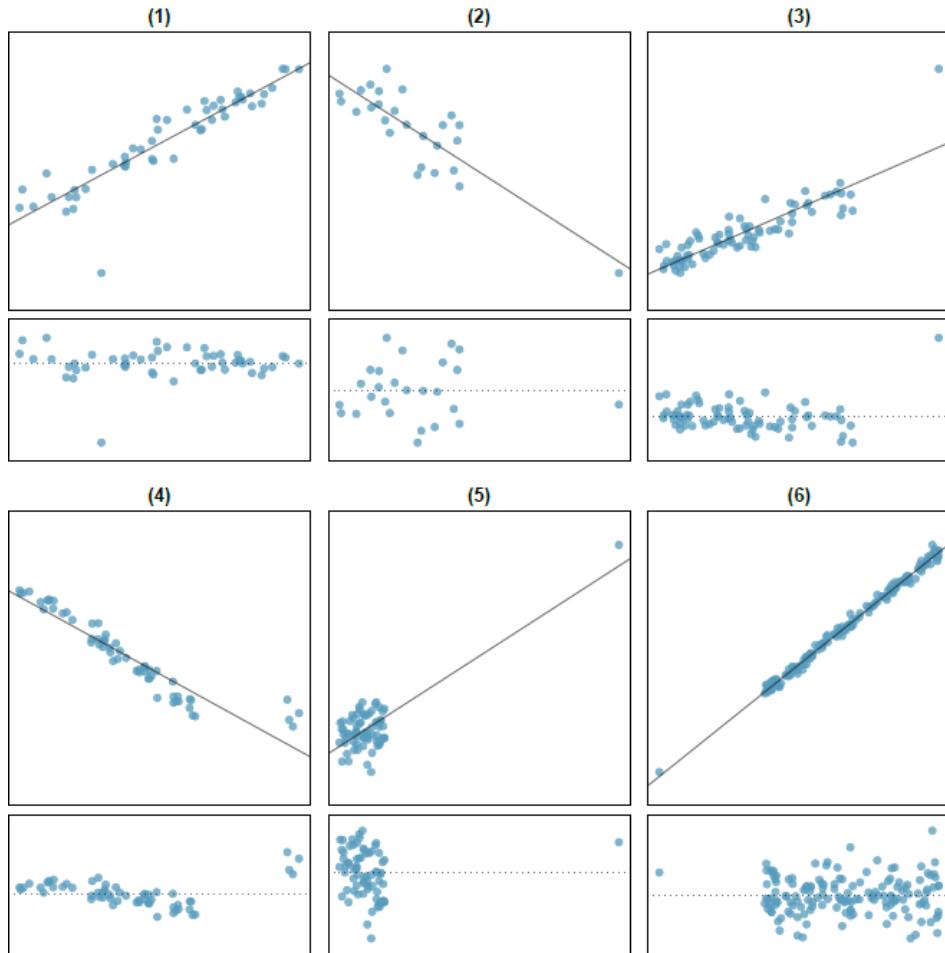


Figura 14.13: seis modelos de regresión lineal con sus respectivos gráficos de residuos. Fuente: Diez et al. (2017, p. 350).

$$R^2 = \frac{s_y^2 - s_e^2}{s_y^2} \quad (14.7)$$

Para el ejemplo, como podemos verificar en la penúltima línea de la descripción del modelo obtenido (figura 14.6) bajo el nombre de **Multiple R-squared**, tenemos:

$$R^2 = -0,868^2 = 0,753$$

En consecuencia, la recta de regresión lineal, construida con el peso del vehículo como predictor, explica 75,3 % de la variabilidad en el rendimiento.

14.5.3 Validación cruzada

Hasta ahora hemos visto cómo detectar valores atípicos que influyan en la recta y cómo determinar si la recta se ajusta bien a la muestra. Sin embargo, nos falta verificar si el modelo puede generalizarse. Una estrategia frecuente para esto es la **validación cruzada**, en la que el conjunto de datos se separa en dos fragmentos:

- **Conjunto de entrenamiento:** suele contener entre el 80 % y el 90 % de las observaciones (aunque es frecuente encontrar que solo contenga el 70 % de ellas), escogidas de manera aleatoria, y se emplea para ajustar la recta con el método de mínimos cuadrados.
- **Conjunto de prueba:** contiene el 10 % a 30 % restante de las instancias, y se usa para evaluar el modelo con datos nuevos.

Estos porcentajes se definen con el propósito de contar con la mayor cantidad de datos posible para ajustar el modelo, resguardando que el conjunto de prueba sea lo suficientemente grande como para obtener una buena estimación de la calidad del modelo.

La idea detrás de este método es evaluar cómo se comporta el modelo con datos que no ha visto previamente, en comparación al comportamiento con el conjunto de entrenamiento. Una buena métrica que podemos usar para esta tarea es el **error cuadrático medio**, o MSE por sus siglas en inglés, pues es lo que el método de mínimos cuadrados busca minimizar.

El script 14.4 aborda, una vez más, el ajuste de una RLS para predecir el rendimiento de un automóvil a partir de su peso, pero esta vez usando validación cruzada. Como resultado, obtenemos el modelo de la figura 14.14. Fijémonos en que, para el conjunto de entrenamiento, el error cuadrático medio es $MSE_e = 5,652$, mientras que para el conjunto de prueba obtenemos $MSE_p = 17,516$, bastante más elevado (¡más del triple!). Esto sugiere que el modelo puede estar sobreajustado, es decir, que se adapta bien a los datos del conjunto de entrenamiento pero no tanto al conjunto de prueba, por lo que podría ser imprudente suponer que puede ser generalizado. Sin embargo, esto puede deberse a la separación aleatoria de los datos. Al ejecutar el script 14.4 reemplazando la semilla aleatoria por 125, obtenemos el resultado de la figura 14.15. Podemos notar que los parámetros del modelo son algo diferentes a los obtenidos con la semilla 101. Además, ahora el error cuadrático medio para el conjunto de entrenamiento es $MSE_e = 8,596$ y para el conjunto de prueba, $MSE_p = 9,122$. Estos últimos valores son muy parecidos, por lo que este segundo modelo sí podría ser generalizable.

```
Call:  
lm(formula = mpg ~ wt, data = entrenamiento)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-3.602 -1.854 -0.212  1.590  4.684  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 34.4745    2.1780 15.829 8.89e-13 ***  
wt          -4.5761    0.6566 -6.969 9.16e-07 ***  
---  
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
Residual standard error: 2.494 on 20 degrees of freedom  
Multiple R-squared:  0.7083, Adjusted R-squared:  0.6938  
F-statistic: 48.57 on 1 and 20 DF,  p-value: 9.159e-07
```

Figura 14.14: recta de mínimos cuadrados usando validación cruzada.

Script 14.4: ajuste de una regresión lineal simple usando validación cruzada.

```
1 # Cargar los datos.
```

```

2 datos <- mtcars
3
4 # Crear conjuntos de entrenamiento y prueba.
5 set.seed(101)
6 n <- nrow(datos)
7 n_entrenamiento <- floor(0.7 * n)
8 muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
9 entrenamiento <- datos[muestra, ]
10 prueba <- datos[-muestra, ]
11
12 # Ajustar modelo con el conjunto de entrenamiento.
13 modelo <- lm(mpg ~ wt, data = entrenamiento)
14 print(summary(modelo))
15
16 # Calcular error cuadrado promedio para el conjunto de entrenamiento.
17 mse_entrenamiento <- mean(modelo$residuals ** 2)
18 cat("MSE para el conjunto de entrenamiento:", mse_entrenamiento, "\n")
19
20 # Hacer predicciones para el conjunto de prueba.
21 predicciones <- predict(modelo, prueba)
22
23 # Calcular error cuadrado promedio para el conjunto de prueba.
24 error <- prueba[["mpg"]] - predicciones
25 mse_prueba <- mean(error ** 2)
26 cat("MSE para el conjunto de prueba:", mse_prueba)

```

Call:

lm(formula = mpg ~ wt, data = entrenamiento)

Residuals:

Min	1Q	Median	3Q	Max
-3.7972	-2.7338	-0.0359	1.3380	6.5505

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	38.1048	2.3760	16.037	6.97e-13 ***
wt	-5.6044	0.7381	-7.593	2.59e-07 ***

Signif. codes:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’
	0.1 ‘ ’	1		

Residual standard error: 3.075 on 20 degrees of freedom

Multiple R-squared: 0.7425, Adjusted R-squared: 0.7296

F-statistic: 57.66 on 1 and 20 DF, p-value: 2.586e-07

Figura 14.15: otra recta de mínimos cuadrados usando validación cruzada.

14.5.4 Validación cruzada de k pliegues

Una buena manera de mejorar la estimación del error cuadrático medio es obtener más observaciones, de acuerdo al ya conocido teorema del límite central. Para esto, se puede usar una nueva manera de remuestreo: la **validación cruzada de k pliegues** (en inglés *k-fold cross validation*). La idea de fondo es la misma de

la validación cruzada expuesta en el apartado anterior: usar un conjunto de entrenamiento para ajustar el modelo y otro de prueba para evaluarlo. Sin embargo, esta variante modifica este proceso a fin de obtener k estimaciones del error. Para ello se separa el conjunto de datos en k subconjuntos de igual tamaño y, como explica Amat Rodrigo (2016d), realizamos k estimaciones del error cuadrático medio de la siguiente manera:

1. Para cada uno de los k subconjuntos:
 - a) Tomar uno de los k subconjuntos del conjunto de entrenamiento y reservarlo como conjunto de prueba.
 - b) Ajustar la recta de mínimos cuadrados usando para ello los $k - 1$ subconjuntos restantes.
 - c) Estimar el error cuadrático medio usando para ello el conjunto de prueba.
2. Estimar el error cuadrático medio del modelo, correspondiente a la media de los k errores cuadrados medios obtenidos en el paso 1.

En R, podemos realizar este proceso de forma bastante sencilla gracias a la función `train(formula, method = "lm", trControl = trainControl(method = "cv", number))` del paquete `caret`, donde:

- `formula`: fórmula que se emplea en las llamadas internas a `lm()`.
- `number`: cantidad de pliegues (k).

El lector atento habrá notado que hemos asignado valores fijos a algunos de los argumentos de la función `train()`. Esto se debe a que este método sirve para ajustar muchos otros modelos además de la RLS. El script 14.5 crea, una vez más, una RLS para predecir el rendimiento de un automóvil a partir de su peso, usando ahora validación cruzada de 5 pliegues.

Script 14.5: ajuste de una regresión lineal simple usando validación cruzada de 5 pliegues.

```

1 library(caret)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Crear conjuntos de entrenamiento y prueba.
7 set.seed(101)
8 n <- nrow(datos)
9 n_entrenamiento <- floor(0.7 * n)
10 muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
11 entrenamiento <- datos[muestra, ]
12 prueba <- datos[-muestra, ]
13
14 # Ajustar modelo usando validación cruzada de 5 pliegues.
15 modelo <- train(mpg ~ wt, data = entrenamiento, method = "lm",
16                   trControl = trainControl(method = "cv", number = 5))
17
18 print(summary(modelo))
19
20 # Hacer predicciones para el conjunto de entrenamiento.
21 predicciones_entrenamiento <- predict(modelo, entrenamiento)
22
23 # Calcular error cuadrado promedio para el conjunto de prueba.
24 error_entrenamiento <- entrenamiento[["mpg"]] - predicciones_entrenamiento
25 mse_entrenamiento <- mean(error_entrenamiento ** 2)
26 cat("MSE para el conjunto de entrenamiento:", mse_entrenamiento, "\n")
27
28 # Hacer predicciones para el conjunto de prueba.
29 predicciones_prueba <- predict(modelo, prueba)
30
31 # Calcular error cuadrado promedio para el conjunto de prueba.
32 error_prueba <- prueba[["mpg"]] - predicciones_prueba
33 mse_prueba <- mean(error_prueba ** 2)
34 cat("MSE para el conjunto de prueba:", mse_prueba)
```

Un aspecto importante a tener en cuenta es que la función `train()` ajusta el modelo final con la totalidad del conjunto de entrenamiento, por lo que el error cuadrático medio para el conjunto de prueba y los parámetros del modelo son los mismos que ya habíamos obtenido. Sin embargo, la estimación del error cuadrático medio para el conjunto de entrenamiento es diferente: al usar la semilla 101 se obtiene $MSE_e = 2,785$ y para la semilla 125, $MSE_e = 3,482$, valores bastante más parecidos entre sí.

14.5.5 Validación cruzada dejando uno fuera

Cuando la muestra disponible es pequeña, tema que reforzaremos en el capítulo siguiente, una buena alternativa es usar **validación cruzada dejando uno fuera** (*leave-one-out cross validation* en inglés). El esquema es el mismo que para validación cruzada con k pliegues, pero ahora usaremos tantos pliegues como observaciones tenga el conjunto de entrenamiento. En otras palabras, hacemos una iteración por cada elemento del conjunto de entrenamiento, reservando una única observación para validación. En R, la llamada a `train()` es muy similar a la que hicimos para validación cruzada con k pliegues: solo cambia el argumento `trControl`, cuyo valor ahora debe ser `trainControl(method = "LOOCV")`.

14.6 INFERENCIA PARA REGRESIÓN LINEAL

También podemos usar los modelos de RLS para hacer inferencia, procedimiento que ilustraremos mediante el siguiente ejemplo: el gerente de una empresa de desarrollo de software cree que, mientras más *stakeholders* tiene un proyecto, menos requisitos funcionales tiene el software a desarrollar. Para llevar a cabo el estudio pertinente, seleccionó aleatoriamente los datos de 15 proyectos de entre los 200 que ha desarrollado la empresa hasta la fecha, los cuales se muestran en la tabla 14.2.

requisitos	stakeholders
11	8
10	8
12	6
14	6
8	8
13	7
18	3
15	1
20	3
16	4
21	5
13	4
10	4
9	9
21	2

Tabla 14.2: requisitos funcionales y cantidad de *stakeholders* para diferentes proyectos desarrollados por la empresa.

Para llevar a cabo su estudio, el gerente ha ajustado un modelo de regresión lineal usando para ello el script

14.6. La recta ajustada y el gráfico de residuos se muestran en la figura 14.16.

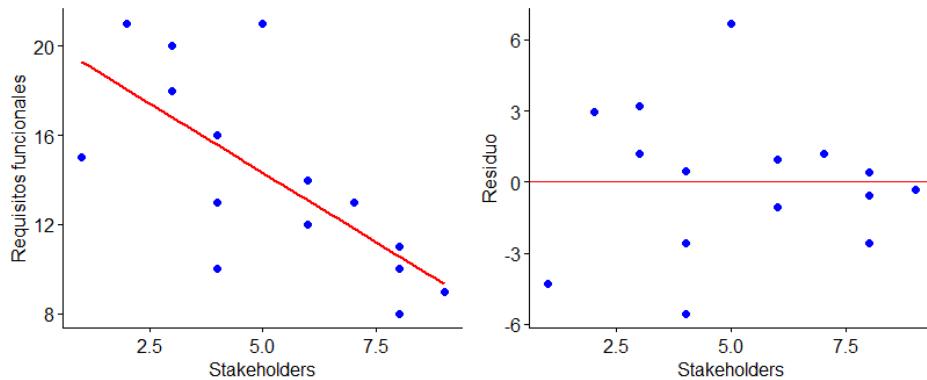


Figura 14.16: regresión lineal para la cantidad de requisitos funcionales de acuerdo a la cantidad de *stakeholders*.

Script 14.6: regresión lineal para la cantidad de requisitos funcionales de acuerdo a la cantidad de *stakeholders*.

```

1 library(ggpubr)
2
3 # Crear los datos originales.
4 requisitos <- c(11, 10, 12, 14, 8, 13, 18, 15, 20, 16, 21, 13, 10, 9, 21)
5 stakeholders <- c(8, 8, 6, 6, 8, 7, 3, 1, 3, 4, 5, 4, 4, 9, 2)
6 datos <- data.frame(requisitos, stakeholders)
7
8 # Ajustar modelo.
9 modelo <- lm(requisitos ~ stakeholders, data = datos)
10 print(summary(modelo))
11
12 # Graficar el modelo.
13 p <- ggscatter(
14   datos, x = "stakeholders", y = "requisitos", color = "blue", fill = "blue",
15   xlab = "Stakeholders", ylab = "Requisitos funcionales")
16
17 p <- p + geom_smooth(method = lm, se = FALSE, colour = "red")
18
19 # Graficar los residuos.
20 b_1 <- modelo$coefficients[2]
21 b_0 <- modelo$coefficients[1]
22 residuos <- datos[["requisitos"]] - (b_1 * datos[["stakeholders"]] + b_0)
23 datos <- data.frame(datos, residuos)
24
25 r <- ggscatter(datos, x = "stakeholders", y = "residuos", color = "blue",
26                  fill = "blue", xlab = "Stakeholders", ylab = "Residuo")
27
28 r <- r + geom_hline(yintercept = 0, colour = "red")
29
30 g <- ggarrange(p, r, ncol = 2, nrow = 1)
31 print(g)
32
33 # Verificar normalidad de los residuos.
34 cat("Prueba de normalidad para los residuos\n")
35 print(shapiro.test(datos$residuos))

```

Puesto que la correlación entre ambas variables es relativamente fuerte ($R = -0,706$), podemos comprobar

que los datos siguen una tendencia lineal. Al aplicar la prueba de normalidad de Shapiro-Wilk a los residuos, concluimos que estos siguen una distribución cercana a la normal ($p = 0,924$). Podemos apreciar en la figura 14.16 que la variabilidad de los residuos es relativamente constante. Por otra parte, las observaciones son independientes entre sí, pues han sido seleccionadas de manera aleatoria y corresponden a menos del 10 % de la población. En consecuencia, se verifica el cumplimiento de todas las condiciones necesarias para emplear un modelo de RLS ajustado mediante mínimos cuadrados.

```

Call:
lm(formula = requisitos ~ stakeholders, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5624 -1.8160  0.4234  1.1840  6.6840 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 20.5482    1.9810 10.373 1.17e-07 ***
stakeholders -1.2464    0.3466 -3.596  0.00326 **  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.184 on 13 degrees of freedom
Multiple R-squared:  0.4987, Adjusted R-squared:  0.4601 
F-statistic: 12.93 on 1 and 13 DF,  p-value: 0.003255

```

Figura 14.17: descripción detallada del modelo obtenido por el gerente para el ejemplo.

En la descripción del modelo (figura 14.17) podemos notar, bajo el encabezado **Coefficients**, una tabla con dos filas: una por cada parámetro del modelo, donde la primera corresponde a la intercepción y la segunda, a la pendiente. A su vez, la primera columna identifica los parámetros del modelo y la segunda presenta sus valores estimados. Como toda estimación tiene asociado un margen de error, la tercera columna muestra el error estándar para cada parámetro. Las dos columnas restantes requieren de una explicación algo más detallada, por lo que no las describiremos aquí.

El gerente, con la intención de evaluar a una abatida estudiante en práctica que aún no ha cursado su asignatura de estadística, le ha entregado los resultados obtenidos y le ha preguntado si los datos sustentan su teoría de que la cantidad de requisitos funcionales disminuye a medida que la cantidad de *stakeholders* aumenta. Tras muchas horas buscando información, la estudiante ha formulado las siguientes hipótesis:

H_0 : $\beta_1 = 0$. La pendiente del modelo es igual a 0 o, lo que es lo mismo, la cantidad de *stakeholders* no explica en absoluto la cantidad de requisitos funcionales.

H_A : $\beta_1 < 0$.

Puesto que el valor p entregado por R corresponde a una prueba bilateral (fijarse en el valor absoluto que incluye el título de la columna: $Pr(>|t|)$), en el caso unilateral se debe considerar la mitad de este valor. En consecuencia, el gerente concluye, con 99 % de confianza ($p < 0.002$), que en efecto la cantidad de requisitos funcionales disminuye a medida que la cantidad de *stakeholders* aumenta.

14.7 EJERCICIOS PROPUESTOS

1. ¿Qué asume la regresión lineal, qué variables involucra y con qué parámetros trabaja?

2. ¿Cómo lucen los gráficos de dispersión de una relación lineal fuerte, de una débil y de una nula?
3. ¿Por qué hay que mirar un gráfico de dispersión de los datos al pensar en regresión lineal?
4. Describe cómo se usa la línea de regresión para predecir.
5. ¿Qué son los residuos? ¿Qué valores pueden tomar? ¿Qué utilidad tienen?
6. Explica qué mide la correlación, cómo se calcula y qué valores puede tomar.
7. Explica cómo funciona el método de los mínimos cuadrados.
8. ¿Qué condiciones necesita el método de los mínimos cuadrados para ser confiable?
9. ¿Cómo lucen los gráficos de residuos que no cumplen con alguno de los requisitos enunciados en el ejercicio anterior?
10. Explica cómo se interpretan los parámetros estimados con regresión por mínimos cuadrados.
11. Explica qué mide el coeficiente de determinación R^2 , cómo se calcula y qué valores puede tomar.
12. Explica cómo se interpretan los parámetros de la regresión lineal cuando la variable predictora es categórica.
13. Explica qué es apalancamiento y por qué es importante detectarlo.
14. ¿Cómo lucen los gráficos de datos (y residuos) que pueden tener problemas de apalancamiento?
15. ¿Cuáles son las hipótesis que se contrastan al hacer inferencia con la regresión lineal?
16. Investiga cómo se usa la función `lm()` de R y qué información entrega.

CAPÍTULO 15. REGRESIÓN LINEAL MÚLTIPLE

En el capítulo anterior conocimos los principios detrás de la regresión lineal, considerando para ello una única variable predictora y una variable de respuesta. Sin embargo, en la vida real es más frecuente que un fenómeno sea explicado por muchas variables. En consecuencia, en este capítulo presentaremos un nuevo modelo lineal más complejo: la regresión lineal múltiple (RLM), correspondiente al caso de una única respuesta con múltiples predictores. Para ello tomaremos como base los textos de Field et al. (2012, pp. 245-311) y Diez et al. (2017, pp. 372-385).

Una regresión lineal con múltiples variables tiene la forma que se presenta en la ecuación 15.1, donde:

- Cada x_i es un predictor.
- Cada β_i corresponde a un parámetro del modelo.
- k es la cantidad de predictores.
- \hat{y} es una estimación de la respuesta.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k \quad (15.1)$$

Una vez más, al ajustar el modelo mediante el método de mínimos cuadrados, buscamos minimizar la suma de los cuadrados de los residuos (ecuación 15.2), proceso que se vuelve más complejo a medida que aumenta la cantidad de variables por lo que suele hacerse mediante el uso de software.

$$\min \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (15.2)$$

Al igual que en el caso de la regresión lineal simple (RLS), la RLM requiere verificar algunas condiciones:

1. La distribución de los residuos debe ser cercana a la normal.
2. La variabilidad de los residuos debe ser aproximadamente constante.
3. Los residuos deben ser independientes entre sí.
4. Cada variable se relaciona linealmente con la respuesta.

Para los ejemplos de este capítulo usaremos una vez más el conjunto de datos `mtcars`, cuyas variables describimos en la tabla 14.1. Como punto de partida, recordemos la RLS para predecir el rendimiento de un automóvil (usando el mismo conjunto de datos) a partir de su peso y ajustada con todo el conjunto de entrenamiento (figura 14.7).

Ahora consideraremos una RLM con dos predictores para el rendimiento: el peso (columna `wt`) y el tiempo mínimo requerido para recorrer un cuarto de milla (columna `qsec`), modelo que podemos obtener mediante el script 15.1 y que se muestra en la figura 15.1. El procedimiento para ajustar la RLM en R es el mismo que usamos en el capítulo anterior, pero ahora en el lado derecho de la fórmula para ajustar el modelo tenemos que combinar ambos predictores.

Script 15.1: regresión lineal para predecir el rendimiento de un automóvil a partir de dos variables.

```
1 library(scatterplot3d)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Ajustar modelo usando validación cruzada de 5 pliegues.
7 modelo <- lm(mpg ~ wt + qsec, data = datos)
8 print(summary(modelo))
9
```

```

Call:
lm(formula = mpg ~ wt + qsec, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3962 -2.1431 -0.2129  1.4915  5.7486 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 19.7462    5.2521   3.760 0.000765 ***
wt          -5.0480    0.4840 -10.430 2.52e-11 ***
qsec         0.9292    0.2650   3.506 0.001500 **  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.596 on 29 degrees of freedom
Multiple R-squared:  0.8264, Adjusted R-squared:  0.8144 
F-statistic: 69.03 on 2 and 29 DF,  p-value: 9.395e-12

```

Figura 15.1: descripción del modelo lineal para predecir el rendimiento de un automóvil a partir de dos variables.

```

10 # Graficar modelo ajustado.
11 g <- scatterplot3d(datos$wt, datos$qsec, datos$mpg, type = "p",
12                      highlight.3d = TRUE, pch = 20, xlab = "Peso [lb x 1000]",
13                      ylab = "Rendimiento [millas/galón]",
14                      zlab = "1/4 de milla [s]")
15
16 g$plane3d(modelo, draw_polygon = TRUE, draw_lines = TRUE)
17 print(g)

```

Para usar este modelo a fin de predecir valores para la respuesta a partir de un nuevo conjunto de datos, usamos una vez más la función `predict()`, del mismo modo que vimos en el capítulo 14 para la RLS.

Como en este caso tenemos dos predictores, lo que se ajusta ya no es una recta, sino un plano, como muestra la figura 15.2. Así, ya no tiene sentido hablar de la pendiente de la recta al momento de interpretar los parámetros del modelo. Un análisis de regresión lineal con múltiples variables busca aislar la relación entre cada predictor y la respuesta, por lo que el coeficiente β_i , asociado al i -ésimo predictor, representa el cambio esperado que se produce en la respuesta al incrementar dicho predictor en una unidad, **manteniendo constantes todos los demás predictores**. Si b_1 es el parámetro ajustado para el peso y b_2 es el parámetro ajustado para el cuarto de milla, b_1 puede entenderse como la pendiente del plano de la figura 15.1 con respecto al eje y , mientras que b_2 puede entenderse como la pendiente del mismo plano con respecto al eje z . A su vez, la intercepción fija la posición del plano con respecto al origen.

Desde luego, podemos extender esta idea para más de dos predictores. En tal caso ajustamos un hiperplano cuya forma y ubicación están dadas por los parámetros del modelo.

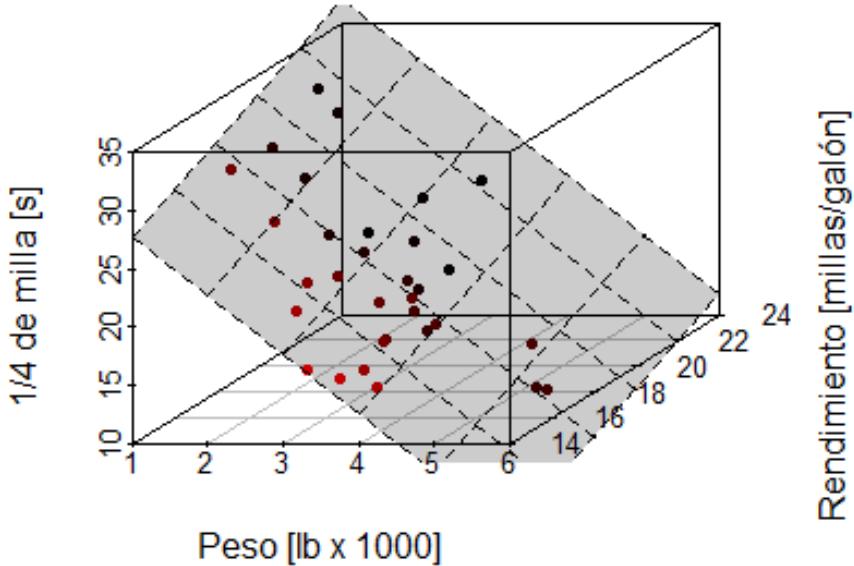


Figura 15.2: plano ajustado para la RLM con dos predictores.

15.1 RLM CON PREDICTORES CATEGÓRICOS

En el capítulo 14 habíamos señalado que para usar una variable categórica con dos niveles como predictor, esta debe ser transformada en una variable indicadora. Desde luego, podemos extender la misma idea para el caso de variables categóricas con más niveles.

Imaginemos que tenemos una variable categórica con k niveles. El primer paso consiste en crear $k - 1$ nuevas variables artificiales. A continuación, para cada una de estas nuevas variables, escogemos un nivel diferente de la variable original y asignamos un 1 a todas las observaciones que tengan ese nivel y un 0 a las restantes. Para entender mejor esta idea, supongamos que un conjunto de datos tiene la variable categórica `tipo`, con cuatro niveles: A, B, C y D. Creamos tres nuevas variables, por ejemplo, `tipo_A`, `tipo_B` y `tipo_C`. La variable `tipo_A` contiene tantas observaciones como la variable original, con un 1 para aquellas observaciones en que `tipo` toma el valor A y un 0 para las restantes. Para las variables `tipo_B` y `tipo_C` se procede de manera análoga. Puesto que solo se crean $k - 1$ variables artificiales, se descarta aquella correspondiente al nivel D de la variable `tipo`.

En R, podemos hacer esta tarea de manera sencilla mediante la función `dummy.data.frame(data, names, drop)` del paquete `dummies`, donde:

- `data`: matriz de datos.
- `names`: nombres de las columnas para las que se desea crear variables artificiales. Si se omite este argumento, se crean variables artificiales para todas las variables categóricas y de tipo string.
- `drop`: indicador booleano que, cuando es verdadero, descarta la variable original del resultado.

Es importante señalar que esta función crea una variable artificial por cada nivel de la variable categórica,

por lo que debemos descartar una de ellas.

El script 15.2 muestra el funcionamiento de `dummy.data.frame()` para una matriz de datos con dos variables categóricas. La tabla 15.1 muestra, en las columnas de la izquierda, el conjunto de datos original y en las de la derecha, el conjunto de datos con las nuevas variables artificiales.

sujeto	sexo	tipo	valor	sujeto	sexoM	tipoB	tipoC	tipoD	valor
1	F	B	1.68	1	0	1	0	0	1.68
2	F	D	2.79	2	0	0	0	1	2.79
3	M	A	1.92	3	1	0	0	0	1.92
4	M	B	2.26	4	1	1	0	0	2.26
5	M	A	2.10	5	1	0	0	0	2.10
6	M	C	2.63	6	1	0	1	0	2.63
7	F	D	2.19	7	0	0	0	1	2.19
8	M	D	3.62	8	1	0	0	1	3.62
9	F	D	2.76	9	0	0	0	1	2.76
10	F	A	1.26	10	0	0	0	0	1.26

Tabla 15.1: creación de variables artificiales para una matriz de datos con variables categóricas.

Script 15.2: creación de variables artificiales para variables categóricas.

```

1 library(dummies)
2
3 # Crear una matriz de datos.
4 sujeto <- 1:10
5 sexo <- c("F", "F", "M", "M", "M", "F", "M", "F", "F")
6 tipo <- c("B", "D", "A", "B", "A", "C", "D", "D", "A")
7 valor <- c(1.68, 2.79, 1.92, 2.26, 2.1, 2.63, 2.19, 3.62, 2.76, 1.26)
8 datos <- data.frame(sujeto , sexo , tipo , valor)
9
10 # Crear variables artificiales.
11 datos.dummy <- dummy.data.frame(datos , drop = TRUE)
12 datos.dummy[["sexoF"]] <- NULL
13 datos.dummy[["tipoA"]] <- NULL
14
15 # Crear modelos lineales.
16 m1 <- lm(valor ~ sexo + tipo , datos)
17 print(m1)
18
19 m2 <- lm(valor ~ sexom + tipoB + tipoC + tipoD , datos.dummy)
20 print(m2)
```

Finalmente, para usar la variable categórica como predictor, agregamos al modelo todas las variables artificiales creadas a partir de ella. Cabe señalar que la función `lm()` realiza internamente este proceso cuando recibe una variable categórica entre los predictores (las variables indicadoras descartadas en el script 15.2 replican el resultado que entrega `lm()`). No obstante, al usar el modelo, debemos fijarnos en que la cantidad de predictores categóricos sea la misma, como también en que la cantidad y el orden de sus niveles coincida con los del conjunto de entrenamiento.

El script 15.2 ajusta además dos modelos, el primero usando el conjunto de datos original y el segundo, el conjunto de datos transformado para que tenga variables indicadoras en reemplazo de las variables categóricas. Si nos fijamos en la figura 15.3, podemos ver que en ambos casos el modelo tiene los mismos predictores.

```

Call:
lm(formula = valor ~ sexo + tipo, data = datos)

Coefficients:
(Intercept)      sexoM       tipoB       tipoC       tipoD
1.2139         0.8191      0.3465      0.5970      1.4213

Call:
lm(formula = valor ~ sexoM + tipoB + tipoC + tipoD, data = datos.dummy)

Coefficients:
(Intercept)      sexoM       tipoB       tipoC       tipoD
1.2139         0.8191      0.3465      0.5970      1.4213

```

Figura 15.3: resultado del script 15.2.

15.2 CONDICIONES PARA USAR RLM

Llegado este punto, necesitamos examinar con más detalle las condiciones que debemos cumplir para que un modelo de regresión lineal sea generalizable:

1. Las variables predictoras deben ser cuantitativas o dicotómicas (de ahí la necesidad de variables indicadoras para manejar más de dos niveles).
2. La variable de respuesta debe ser cuantitativa y continua, sin restricciones para su variabilidad.
3. Los predictores deben tener algún grado de variabilidad (su varianza no debe ser igual a cero). En otras palabras, no pueden ser constantes.
4. No debe existir **multicolinealidad**. Esto significa que no deben existir relaciones lineales *fuertes* entre dos o más predictores (coeficientes de correlación altos).
5. Los residuos deben ser homocedásticos (con varianzas similares) para cada nivel de los predictores.
6. Los residuos deben seguir una distribución cercana a la normal centrada en cero.
7. Los valores de la variable de respuesta son independientes entre sí.
8. Cada predictor se relaciona linealmente con la variable de respuesta.

15.3 EVALUACIÓN DEL AJUSTE DE UNA RLM

En el capítulo anterior introdujimos el coeficiente de determinación (R^2) como instrumento para evaluar la bondad de ajuste de una regresión lineal. Sin embargo, cuando el modelo es multivariado, la función ya conocida para estimar R^2 genera una mala estimación del porcentaje de la varianza explicada por el modelo, pues los grados de libertad asociados a la variabilidad de los residuos es ahora diferente, como muestra la ecuación 15.3, donde n es el tamaño de la muestra y k , la cantidad de variables predictoras.

$$\nu = n - k - 1 \quad (15.3)$$

Así, para evaluar una RLM tenemos que usar un coeficiente de determinación ajustado. Algunos autores han propuesto distintas maneras de efectuar este ajuste, una de las cuales presentamos en la ecuación 15.4. También podemos usar este ajuste cuando tenemos un único predictor, aunque la diferencia en este caso suele ser muy pequeña como para ser relevante.

$$R^2 = 1 - \frac{s_e^2}{s_y^2} \cdot \frac{n - 1}{n - k - 1} \quad (15.4)$$

Existen otras alternativas para evaluar la bondad de ajuste de un modelo que se basan en el **principio de parsimonia**, también llamado navaja de Occam, el cual indica que un modelo debe mantenerse tan simple como sea posible. Dos de ellas son el **criterio de información de Akaike**, abreviado AIC, y el **criterio bayesiano de Schwarz** (BIC o SBC), que penalizan el modelo por contener variables adicionales, por lo que mientras menor sea su valor, mejor será el modelo. Si bien el cálculo de estas medidas no se detalla aquí por ser un tópico más avanzado, podemos obtenerlas en R mediante las funciones `AIC(object)` y `BIC(object)`, donde `object` corresponde a un modelo lineal ajustado.

Para el modelo que habíamos ajustado usando únicamente el peso como predictor, obtenemos $AIC = 166,03$ y $BIC = 170,43$. Del mismo modo, para el modelo que usa como predictores el peso y el cuarto de milla, en cambio, tenemos que $AIC = 156,72$ y $BIC = 162,58$. En consecuencia, el segundo modelo parece ser “mejor” bajo estos criterios.

Otra opción, adecuada cuando necesitamos saber cuáles predictores son estadísticamente significativos, es observar los valores p asociados a cada predictor. Habitualmente consideraremos significativos aquellos predictores para los cuales $p < 0,05$.

15.4 COMPARACIÓN DE MODELOS

En la sección anterior vimos que métricas como el AIC o el BIC nos pueden resultar útiles para comparar dos modelos de regresión lineal, considerando la noción general que un modelo es mejor mientras menor sea su valor de AIC (o BIC). Si calculamos el AIC para cada uno de los modelos ajustados hasta ahora (script 15.3), veremos que el AIC del modelo con dos predictores es menor. Sin embargo, al ser una medida relativa, hasta ahora no contamos con una prueba estadística que nos permita determinar si la diferencia es significativa.

Cuando los modelos son jerárquicos, es decir, el segundo incorpora nuevos predictores además de mantener los del primer modelo, podemos hacer una prueba de hipótesis usando los coeficientes de determinación para ver si la diferencia es significativa. El estadístico de prueba se calcula mediante la ecuación 15.5, donde:

- n : cantidad de observaciones.
- k_2 : cantidad de predictores en el modelo con más variables (segundo modelo).
- R_{cambio}^2 : diferencia entre los valores de R^2 del segundo y el primer modelo.
- k_{cambio} : diferencia entre la cantidad de predictores del segundo y el primer modelo.
- R_2^2 : coeficiente de determinación del segundo modelo.

$$F_{cambio} = \frac{(n - k_2 - 1)R_{cambio}^2}{k_{cambio}(1 - R_2^2)} \quad (15.5)$$

Así, para los dos modelos ajustados hasta ahora tenemos:

$$F_{cambio} = \frac{(32 - 2 - 1)(0,8264 - 0,7528)}{1(1 - 0,8264)} = 12,295$$

Notemos que el estadístico de prueba sigue una distribución F con k_{cambio} y $n - k_2 - 1$ grados de libertad, a partir de lo cual podemos determinar el valor p correspondiente mediante la llamada `pf(12.295, 1, 29, lower.tail = FALSE)`, obteniendo como resultado $p = 0,0015$. En consecuencia, podemos concluir con 95 % de confianza que la diferencia es significativa, por lo que el segundo modelo es mejor.

Como ya es habitual, en R podemos hacer esta tarea de forma simple gracias a la función `anova(object, ...)`, que recibe como argumentos los diferentes modelos a comparar. La interpretación del resultado de esta prueba es sencilla: si el valor p obtenido es significativo, entonces el modelo más complejo (con más predictores) es mejor. Al ejecutar el script 15.3, podemos ver que obtenemos el mismo resultado que con la prueba F.

Script 15.3: comparación de dos modelos lineales.

```

1 # Cargar datos.
2 datos <- mtcars
3
4 # Ajustar modelo con el peso como predictor.
5 modelo_1 <- lm(mpg ~ wt, data = datos)
6 print(summary(modelo_1))
7 aic_1 <- AIC(modelo_1)
8 cat("Modelo 1: AIC =", AIC(modelo_1), "\n")
9
10 # Ajustar modelo con el peso y el cuarto de milla como predictores.
11 modelo_2 <- lm(mpg ~ wt + qsec, data = datos)
12 print(summary(modelo_2))
13 aic_2 <- AIC(modelo_2)
14 cat("Modelo 2: AIC =", AIC(modelo_2), "\n")
15
16 # Comparar ambos modelos.
17 comparacion <- anova(modelo_1, modelo_2)
18 print(comparacion)

```

15.5 SELECCIÓN DE PREDICTORES

La cuarta condición para emplear RLM indica que debemos evitar la multicolinealidad. Esto es importante porque el ajuste de un modelo RLM asume que podemos cambiar una variable predictora, *manteniendo las otras constantes*. Cuando las variables predictoras están correlacionadas, se hace imposible cambiar el valor de una sin alterar también a las demás, desestabilizando la estimación de los coeficientes del modelo que indican cómo influye cada variable predictora en la variable de salida de forma *independiente*.

Cuando existe colinealidad, los valores de los coeficientes varían enormemente si se agregan o quitan unos pocos datos de entrenamiento y se reduce el poder estadístico del modelo. Por esta razón, es importante que escogamos con cuidado las variables predictoras a considerar en un modelo RML. Es más, existe un riesgo adicional cuando los predictores están correlacionados: el orden en que se agreguen puede influenciar la calidad del modelo. En consecuencia, necesitamos contar con alguna estrategia que nos permita determinar cuáles predictores incluir. Existen diversas alternativas para esta tarea.

El método más adecuado, aunque también el más complejo, es la **regresión jerárquica**. Es el que debemos considerar al momento de intentar probar una teoría y consiste en comenzar por incorporar en primer lugar aquellos predictores ya conocidos, en orden de importancia, en base a investigaciones previas. Una vez incorporados todos los predictores ya conocidos, podemos incorporar otros nuevos si creemos que existen **buenas y justificadas razones** para ello. Antes de la masificación de los computadores y de entornos como R, ¡esta era la única alternativa viable!

En R, podemos realizar este método con ayuda de la función `update(object, formula)`, que nos permite incorporar o quitar variables del modelo, donde:

- **object**: modelo previamente ajustado, en este caso con `lm()`.
- **formula**: actualización de la fórmula para el nuevo modelo.

En este caso no contamos con investigaciones previas que nos permitan formular una teoría, por lo que nos limitaremos a proporcionar un ejemplo de cómo usar esta función (script 15.4), cuyos resultados presentamos en la figura 15.4.

Script 15.4: incorporación y eliminación de variables en un modelo de RLM.

```

1 # Cargar datos.
2 datos <- mtcars
3
4 # Ajustar modelo inicial con la variable wt como predictor.
5 modelo <- lm(mpg ~ wt, data = datos)
6 cat("==> Modelo inicial ==\n")
7 print(modelo)
8
9 # Incorporar el predictor cyl.
10 modelo <- update(modelo, . ~ . + cyl)
11 cat("==> Modelo con predictores wt y cyl ==\n")
12 print(modelo)
13
14 # Quitar el predictor wt.
15 modelo <- update(modelo, . ~ . - wt)
16 cat("==> Modelo con predictor cyl ==\n")
17 print(modelo)
18
19 # Agregar predictores wt y drat, y quitar predictor cyl.
20 modelo <- update(modelo, . ~ . + wt + drat - cyl)
21 cat("==> Modelo con predictores wt y drat ==\n")
22 print(modelo)
```

Si, en lugar de probar una teoría, lo que queremos es **explorar los datos**, podemos usar otras estrategias.

Selección hacia adelante Se crea un modelo inicial nulo, es decir, sin predictores, para el cual únicamente se estima la intercepción. A continuación, se escoge como primer predictor aquel que tenga la correlación más alta con la variable de respuesta. Si dicho predictor incrementa la capacidad predictiva del modelo, se retiene en el modelo y se procede a seleccionar un segundo predictor. El modelo con una única variable ajustado al inicio de este capítulo tiene como predictor el peso de los automóviles, variable que en efecto tiene la más alta correlación con el rendimiento (variable de respuesta). El coeficiente de determinación para este modelo es $R^2 = 0,7528$, lo cual significa que explica aproximadamente el 75 % de la variabilidad de la respuesta. En consecuencia, existe aún un 25 % de variabilidad de la respuesta que aún no ha sido explicado.

Para la selección de predictores adicionales, en cada repetición se escoge aquel predictor (que no haya sido previamente agregado al modelo) que tenga la **máxima correlación semi-parcial con la respuesta**, es decir, que explique la máxima porción de la varianza no cubierta por el modelo ya existente. Si la inclusión de este nuevo predictor mejora el poder predictivo del modelo, se incorpora de manera definitiva y se evalúa el siguiente predictor.

Adicionalmente, se evalúa si la inclusión de cada nuevo predictor mejora (es decir, reduce) el AIC. Si ninguno de los posibles predictores restantes logra reducir este indicador, se detiene la inclusión de nuevos predictores.

Eliminación hacia atrás Es el proceso inverso a la selección hacia adelante, puesto que se comienza desde un modelo con todas las variables para luego eliminar predictores uno a uno y evaluar el AIC. Si este último se reduce, se elimina dicho predictor y se reevalúa la contribución de los predictores que aún se encuentran en el modelo. Una vez más, el proceso se repite hasta que no es posible reducir el AIC.

Regresión escalonada En términos generales, opera de manera análoga a la selección hacia adelante. Sin embargo, para reducir el potencial efecto debido al orden de incorporación de los predictores, cada vez que se incorpora uno nuevo se evalúa qué ocurre al descartar el menos importante. En consecuencia, el modelo es permanentemente reevaluado para descartar posibles redundancias entre predictores.

```

==== Modelo inicial ===

Call:
lm(formula = mpg ~ wt, data = datos)

Coefficients:
(Intercept)                  wt
            37.285             -5.344

==== Modelo con predictores wt y cyl ===

Call:
lm(formula = mpg ~ wt + cyl, data = datos)

Coefficients:
(Intercept)                  wt                  cyl
            39.686             -3.191             -1.508

==== Modelo con predictor cyl ===

Call:
lm(formula = mpg ~ cyl, data = datos)

Coefficients:
(Intercept)                  cyl
            37.885             -2.876

==== Modelo con predictores wt y drat ===

Call:
lm(formula = mpg ~ wt + drat, data = datos)

Coefficients:
(Intercept)                  wt                  drat
            30.290             -4.783              1.442

```

Figura 15.4: resultado del script 15.4.

Todos los subconjuntos Una alternativa más exhaustiva, altamente costosa en tiempo de ejecución, es usar un algoritmo de fuerza bruta en el que se exploran todos los subconjuntos de predictores.

Es importante reiterar que solo debemos usar estos métodos si estamos **explorando datos**, pues de lo contrario incurriríamos en faltas a la ética. Veamos por ejemplo el trabajo de Montero Muñoz et al. (2012), donde estudian algunas implicaciones clínicas en el uso de antibióticos en ancianos mayores de 80 años. Podemos ver que, en la recolección de datos, recopilaron variables que la medicina ha probado a lo largo de su historia que son importantes al evaluar la salud de un paciente y que pueden ser afectadas por el uso de medicamentos. Un equipo poco ético podría haber empleado otras variables registradas en las bases de datos de origen, por ejemplo, el monto de la pensión del paciente. ¿Qué consecuencias podrían existir si tuviéramos una correlación espuria de esta variable con la variable de respuesta? ¡Podríamos afectar las vidas de millones de personas si tal estudio concluyera que los antibióticos son más nocivos para ancianos con bajas pensiones!

Todas las estrategias mencionadas están disponibles en R con ayuda de la ya conocida función `update()` y las funciones `add1(object, scope)` y `drop1(object, scope)`, donde:

- **object**: un modelo ajustado.
- **scope**: fórmula que proporciona los términos a agregar o quitar.

La función `add1()` evalúa la incorporación de cada nuevo predictor potencial (separadamente) a un modelo base y entrega algunas métricas para el efecto que tiene su incorporación, entre ellas el AIC. El mejor nuevo predictor corresponde, entonces, a aquella variable con el menor AIC. Las líneas 15 y 22 del script 15.5 ilustran su uso, obteniéndose los resultados que se muestran en la figura 15.5.

Single term additions

Model:				
	Df	Sum of Sq	RSS	AIC
<none>			1126.05	115.943
cyl	1	817.71	308.33	76.494
disp	1	808.89	317.16	77.397
hp	1	678.37	447.67	88.427
drat	1	522.48	603.57	97.988
wt	1	847.73	278.32	73.217
qsec	1	197.39	928.66	111.776
vs	1	496.53	629.52	99.335
am	1	405.15	720.90	103.672
gear	1	259.75	866.30	109.552
carb	1	341.78	784.27	106.369

Single term additions

Model:				
	Df	Sum of Sq	RSS	AIC
<none>			278.32	73.217
cyl	1	87.150	191.17	63.198
disp	1	31.639	246.68	71.356
hp	1	83.274	195.05	63.840
drat	1	9.081	269.24	74.156
qsec	1	82.858	195.46	63.908
vs	1	54.228	224.09	68.283
am	1	0.002	278.32	75.217
gear	1	1.137	277.19	75.086
carb	1	44.602	233.72	69.628

Figura 15.5: resultado de las llamadas a `add1()` en el script 15.5

De manera similar, la función `drop1()` evalúa (separadamente) la eliminación potencial de cada predictor presente en un modelo base y entrega las mismas métricas que `add1()` para el efecto que tiene su eliminación. El mejor predictor a descartar es, una vez más, aquel que lleva a la mayor reducción en AIC. La línea 29 del script 15.5 ilustra su uso, obteniéndose los resultados que se muestran en la figura 15.6.

Script 15.5: Evaluación de variables a incorporar y eliminar en un modelo de RLM.

```

1 # Cargar datos.
2 datos <- mtcars
3
4 # Ajustar modelo nulo.
5 nulo <- lm(mpg ~ 1, data = datos)

```

Single term deletions

```

Model:
mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb
      Df Sum of Sq    RSS   AIC
<none>           147.49 70.898
cyl     1     0.0799 147.57 68.915
disp    1     3.9167 151.41 69.736
hp      1     6.8399 154.33 70.348
drat    1     1.6270 149.12 69.249
wt      1     27.0144 174.51 74.280
qsec    1     8.8641 156.36 70.765
vs      1     0.1601 147.66 68.932
am      1     10.5467 158.04 71.108
gear    1     1.3531 148.85 69.190
carb    1     0.4067 147.90 68.986

```

Figura 15.6: resultado de la llamada a `drop1()` en el script 15.5

```

6 # cat("==> Modelo nulo ==>\n")
7 # print(summary(nulo))
8
9 # Ajustar modelo completo.
10 completo <- lm(mpg ~ ., data = datos)
11 # cat("==> Modelo completo ==>\n")
12 # print(summary(completo))
13
14 # Evaluar variables para incorporar.
15 print(add1(nulo, scope = completo))
16 cat("\n\n")
17
18 # Agregar la variable con menor AIC.
19 modelo <- update(nulo, . ~ . + wt)
20
21 # Evaluar variables para incorporar.
22 print(add1(modelo, scope = completo))
23 cat("\n\n")
24
25 # Agregar la variable con menor AIC.
26 modelo <- update(modelo, . ~ . + cyl)
27
28 # Evaluar variables para eliminar.
29 print(drop1(completo, scope = completo))
30 cat("\n\n")
31
32 # Eliminar la variable con menor AIC.
33 modelo <- update(modelo, . ~ . - cyl)

```

Por supuesto, R en la práctica ya cuenta con funciones que implementan los métodos para seleccionar predictores antes descritos (excepto la regresión jerárquica, por supuesto). Los tres primeros pueden efectuarse mediante la función `step(object, scope, direction, trace)`, que usa `add1()` y `drop1()` de manera iterativa, donde:

- `object`: es un modelo ya ajustado que es usado como punto de partida.
- `scope`: es una lista de fórmulas que define el rango de modelos a explorar.
- `direction`: indica el tipo de selección a realizar, donde “`forward`” corresponde a selección hacia ade-

lante; “backward”, a eliminación hacia atrás, y “both”, a regresión escalonada.

- **trace**: argumento opcional que indica si se quiere ver por consola el proceso realizado.

El script 15.6 muestra el funcionamiento de la función **step()** para seleccionar los predictores a incorporar en un modelo donde la respuesta es, una vez más, el rendimiento de un automóvil.

Script 15.6: selección de predictores a incluir en una RLM.

```
1 library(leaps)
2
3 # Cargar datos.
4 datos <- mtcars
5
6 # Ajustar modelo nulo.
7 nulo <- lm(mpg ~ 1, data = datos)
8 cat("==> Modelo nulo ==>\n")
9 print(summary(nulo))
10
11 # Ajustar modelo completo.
12 completo <- lm(mpg ~ ., data = datos)
13 cat("==> Modelo completo ==>\n")
14 print(summary(completo))
15
16 # Ajustar modelo con selección hacia adelante.
17 adelante <- step(nulo, scope = list(upper = completo), direction = "forward",
18                     trace = 0)
19
20 cat("==> Modelo con selección hacia adelante ==>\n")
21 print(summary(adelante))
22 cat("AIC =", AIC(adelante), "\n\n")
23
24 # Ajustar modelo con eliminación hacia atrás.
25 atras <- step(completo, scope = list(lower = nulo), direction = "backward",
26                  trace = 0)
27
28 cat("==> Modelo con eliminación hacia atrás ==>\n")
29 print(summary(atras))
30 cat("AIC =", AIC(atras), "\n\n")
31
32 # Ajustar modelo con regresión escalonada.
33 escalonado <- step(nulo, scope = list(lower = nulo, upper = completo),
34                      direction = "both", trace = 0)
35
36 cat("==> Modelo con regresión escalonada ==>\n")
37 print(summary(escalonado))
38 cat("AIC =", AIC(escalonado), "\n\n")
39
40 # Ajustar modelo con todos los subconjuntos.
41 modelos <- regsubsets(mpg ~ ., data = datos, method = "exhaustive",
42                         nbest = 1, nvmax = 10)
43
44 print(plot(modelos))
```

La línea 7 del script 15.6 ajusta el modelo nulo, obteniéndose el resultado que se muestra en la figura 15.7. De manera similar, la línea 12 ajusta el modelo completo (figura 15.8).

Las líneas 17–18 usan el método de selección hacia adelante, comenzando desde el modelo nulo. Notemos que en el argumento **scope** entregamos, en este caso, el modelo completo con todos los posibles predictores. No obstante, no es necesario siempre comenzar desde el modelo nulo o evaluar hasta el modelo completo.

```

Call:
lm(formula = mpg ~ 1, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.6906 -4.6656 -0.8906  2.7094 13.8094 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 20.091     1.065   18.86 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.027 on 31 degrees of freedom

```

Figura 15.7: modelo nulo.

En la llamada a `step()` de las líneas 17–18, el argumento `trace` tiene por defecto el valor 1, por lo que la función muestra información de los diferentes modelos evaluados (las figuras 15.9 y 15.10 muestran esta información para todas las iteraciones). Notemos que se comienza por calcular el AIC para el modelo nulo, y luego se determina el AIC (y otras métricas) que se obtendría para diferentes modelos, cada uno de los cuales contendría solo una de las variables restantes. El predictor que se escoge es aquel que genera el modelo con menor AIC (el peso del automóvil en la iteración 1).

En la siguiente iteración se conserva el peso como predictor y se evalúa el AIC de los modelos que incorporan un predictor adicional. Una vez más, se incorpora aquel con menor AIC, correspondiente en esta ocasión a la cantidad de cilindros.

El proceso se detiene una vez que ninguno de los predictores logra un AIC menor que el del modelo ya obtenido en la iteración previa. El modelo final obtenido mediante selección hacia adelante se presenta en la figura 15.11, que se consigue en tres iteraciones y considera, además del peso del automóvil y el número de cilindros, los caballos de fuerza bruta (`hp`).

De manera similar, las figuras 15.12 y 15.13 muestran los modelos resultantes al usar eliminación hacia atrás y regresión escalonada, respectivamente (líneas 25–34). Notemos que, en estos casos, la llamada a `step()` se realiza con el argumento `trace = 0`, por lo que no nos muestra información acerca de los diferentes modelos evaluados.

Las líneas 41–42 del script 15.6, por otra parte, muestran la aplicación del método de todos los subconjuntos para determinar el mejor modelo, proceso que hacemos mediante la función `regsubsets(formula, data, nbest, nvmax, force.in, force.out, method = “exhaustive”)` del paquete `leaps`, donde:

- `formula`: indica la variable de respuesta y los posibles predictores.
- `data`: matriz de datos.
- `nbest`: cantidad de modelos a reportar por cada tamaño de subconjunto. Si, por ejemplo, `nbest = 3`, entonces se reportan los tres mejores modelos con un único predictor, los tres mejores modelos con dos predictores, etc.
- `nvmax`: fija una cantidad máxima de predictores a considerar.
- `force.in`: argumento opcional que toma la forma de un vector con los índices de las columnas que deben ser forzosamente consideradas en los modelos evaluados.
- `force.out`: argumento opcional que toma la forma de un vector con los índices de las columnas que deben ser forzosamente excluidas en los modelos evaluados.

La figura 15.14 representa gráficamente el resultado de la aplicación del método de todos los subconjuntos. En ella, el eje *x* lista todas las variables predictoras y el eje *y* corresponde al BIC de cada modelo, que es el criterio utilizado por defecto por la función `regsubsets()`. Fijémonos en que el eje *y* no es una escala graduada, sino

```

Call:
lm(formula = mpg ~ ., data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4506 -1.6044 -0.1196  1.2193  4.6271 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 12.30337  18.71788  0.657   0.5181    
cyl        -0.11144  1.04502  -0.107   0.9161    
disp        0.01334  0.01786  0.747   0.4635    
hp         -0.02148  0.02177  -0.987   0.3350    
drat        0.78711  1.63537  0.481   0.6353    
wt         -3.71530  1.89441  -1.961   0.0633 .  
qsec        0.82104  0.73084  1.123   0.2739    
vs          0.31776  2.10451  0.151   0.8814    
am          2.52023  2.05665  1.225   0.2340    
gear        0.65541  1.49326  0.439   0.6652    
carb       -0.19942  0.82875  -0.241   0.8122    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.65 on 21 degrees of freedom
Multiple R-squared:  0.869, Adjusted R-squared:  0.8066 
F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07

```

Figura 15.8: modelo completo.

una lista ordenada del BIC para cada uno de los modelos evaluados. Así, la figura corresponde a una matriz en que cada fila está asignada a un modelo, donde se colorea el recuadro para las variables presentes en dicho modelo. Así, el mejor modelo (es decir, el que tiene el menor BIC) es aquel que contiene como predictores el peso (**wt**), el cuarto de milla (**qsec**) y el tipo de transmisión (**am**), además de la intercepción.

También es importante fijarnos en que, en este caso particular, obtenemos el mismo modelo al usar selección hacia adelante y regresión escalonada. Del mismo modo, la eliminación hacia atrás y el método de todos los subconjuntos generan un mismo modelo. Si usamos el AIC como indicador, el primero de estos dos modelos es el mejor. No obstante, el segundo es mejor si consideramos el BIC.

15.6 EVALUACIÓN DE UN MODELO DE RLM

Para ilustrar el proceso de evaluación de un modelo de RLM, consideraremos el modelo obtenido mediante eliminación hacia atrás en la sección precedente, es decir, el modelo que tiene como predictores el peso, el cuarto de milla y el tipo de marcha. Al graficar este modelo (script 15.7, línea 6) obtenemos los gráficos de la figura 15.15, donde podemos ver que la distribución de los residuos se desvía un poco de la normal y que parece haber algunas observaciones atípicas influenciando el ajuste del modelo.

```

Start: AIC=115.94
mpg ~ 1

          Df Sum of Sq    RSS    AIC
+ wt     1   847.73  278.32  73.217
+ cyl    1   817.71  308.33  76.494
+ disp   1   808.89  317.16  77.397
+ hp     1   678.37  447.67  88.427
+ drat   1   522.48  603.57  97.988
+ vs     1   496.53  629.52  99.335
+ am     1   405.15  720.90 103.672
+ carb   1   341.78  784.27 106.369
+ gear   1   259.75  866.30 109.552
+ qsec   1   197.39  928.66 111.776
<none>
                    1126.05 115.943

Step: AIC=73.22
mpg ~ wt

          Df Sum of Sq    RSS    AIC
+ cyl    1   87.150 191.17 63.198
+ hp     1   83.274 195.05 63.840
+ qsec   1   82.858 195.46 63.908
+ vs     1   54.228 224.09 68.283
+ carb   1   44.602 233.72 69.628
+ disp   1   31.639 246.68 71.356
<none>
                    278.32 73.217
+ drat   1   9.081 269.24 74.156
+ gear   1   1.137 277.19 75.086
+ am     1   0.002 278.32 75.217

```

Figura 15.9: modelos evaluados por la función `step()` durante el proceso de selección hacia adelante (parte 1).

15.6.1 Identificación de valores con sobreinfluencia

Existen diversos estadísticos que nos permiten evaluar la influencia de una observación en el ajuste de un modelo de regresión lineal:

- Residuo estandarizado:** los residuos deben seguir una distribución normal estándar, por lo que se esperaría que el 95 % de ellos se encuentre entre -1,96 y 1,96, y el 99 % entre -2,58 y 2,58.
- Valor predicho ajustado:** corresponde al valor predicho si se excluyera dicho punto en el ajuste del modelo. Si el punto no ejerce gran influencia en el ajuste del modelo, se esperaría que este valor fuera muy cercano al predicho cuando dicho punto sí es considerado para el ajuste.
- Residuo estudiantizado:** está dado por el valor predicho ajustado dividido por el error estándar. Una característica importante de esta medida es que es estandarizada y sigue una distribución t, por lo que puede emplearse para hacer comparaciones entre distintos modelos de regresión. Sin embargo, esta medida solo indica cuánto influye la presencia de un punto en el conjunto de entrenamiento en su valor predicho, pero no proporciona información alguna en cuanto a la influencia de la observación en el modelo como un todo.
- Diferencia en ajuste:** más conocido como DFFit, es la diferencia entre el valor predicho para la observación evaluada cuando esta es considerada en el ajuste del modelo y cuando no lo es.
- Diferencia en betas:** más conocido como DFBeta, corresponde a la diferencia entre los valores de un

```

Step: AIC=63.2
mpg ~ wt + cyl

          Df Sum of Sq    RSS    AIC
+ hp     1   14.5514 176.62 62.665
+ carb   1   13.7724 177.40 62.805
<none>           191.17 63.198
+ qsec   1   10.5674 180.60 63.378
+ gear   1    3.0281 188.14 64.687
+ disp   1   2.6796 188.49 64.746
+ vs     1    0.7059 190.47 65.080
+ am     1    0.1249 191.05 65.177
+ drat   1    0.0010 191.17 65.198

Step: AIC=62.66
mpg ~ wt + cyl + hp

          Df Sum of Sq    RSS    AIC
<none>           176.62 62.665
+ am     1   6.6228 170.00 63.442
+ disp   1   6.1762 170.44 63.526
+ carb   1   2.5187 174.10 64.205
+ drat   1   2.2453 174.38 64.255
+ qsec   1   1.4010 175.22 64.410
+ gear   1   0.8558 175.76 64.509
+ vs     1   0.0599 176.56 64.654

```

Figura 15.10: modelos evaluados por la función `step()` durante el proceso de selección hacia adelante (parte 2).

parámetro cuando es estimado usando todas las observaciones y cuando es estimado sin considerar la observación evaluada. Se calcula para cada parámetro del modelo. Se consideran preocupantes aquellas observaciones en que este estimador es mayor a 1.

6. **Distancia de Cook:** es una medida del efecto que tiene una observación en particular combinadamente en todos los parámetros de un modelo. Aquellos valores para los cuales la distancia de Cook sea mayor a 1 pueden ser considerados como potencialmente problemáticos.
7. **Apalancamiento:** estima la influencia del valor observado en los valores predichos. Toma valores entre 0 y 1. Un apalancamiento igual a 0 señala que un punto no ejerce influencia alguna, mientras que un valor de 1 indica que la influencia ejercida por esa observación es total. Se consideran preocupantes aquellas observaciones para las cuales esta medida supere en dos o tres veces el apalancamiento promedio, dado por la ecuación 15.6, donde k es la cantidad de predictores en el modelo y n la cantidad de observaciones empleadas para el ajuste.

$$\bar{x}_{Apalancamiento} = \frac{k+1}{n} \quad (15.6)$$

8. **Razón de covarianza:** corresponde a la razón entre los determinantes de la matriz de covarianzas cuando se consideran todas las observaciones y cuando se omite la observación en estudio. Aquellas observaciones para las cuales el valor de esta medida estén fuera del intervalo definido por la ecuación 15.7 se consideran preocupantes.

$$|Covratio - 1| < \frac{3(k+1)}{n} \quad (15.7)$$

```

Call:
lm(formula = mpg ~ wt + cyl + hp, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.9290 -1.5598 -0.5311  1.1850  5.8986 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 38.75179   1.78686  21.687 < 2e-16 ***
wt          -3.16697   0.74058  -4.276 0.000199 ***
cyl         -0.94162   0.55092  -1.709 0.098480 .  
hp          -0.01804   0.01188  -1.519 0.140015 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.512 on 28 degrees of freedom
Multiple R-squared:  0.8431, Adjusted R-squared:  0.8263 
F-statistic: 50.17 on 3 and 28 DF,  p-value: 2.184e-11

AIC = 155.4766

```

Figura 15.11: modelo obtenido mediante selección hacia adelante.

El script 15.7 muestra la detección de posibles valores atípicos que puedan estar influenciando el modelo de la figura 15.12, obteniéndose los resultados presentados en la figura 15.16. Al examinar estos resultados, en conjunto con los gráficos de la figura 15.15, las observaciones correspondientes al Chrysler Imperial, el Fiat 128 y el Toyota Corolla parecen ser candidatos a eliminación para la generación del modelo. Sin embargo, la distancia de Cook estimada para todas las observaciones potencialmente influyentes están lejos de sobrepasar el valor recomendado, por lo que en este caso no parece ser necesario quitarlos, aun cuando algunas muestren valores un tanto alto de apalancamiento y covarianza. Eliminar datos para construir un modelo debe tener una **sólida justificación**, y por ningún motivo debe hacerse por conveniencia, lo que sería **profundamente antiético**. Consideremos, por ejemplo, que la *pobreza extrema* o los *super-ricos* son casos extremos que, usualmente, no deben eliminarse de un modelo que intente estudiar la distribución de los ingresos de un país.

Script 15.7: identificación de valores atípicos.

```

1 # Cargar datos.
2 datos <- mtcars
3
4 # Ajustar modelo.
5 modelo <- lm(mpg ~ wt + qsec + am, data = datos)
6 plot(modelo)
7
8 # Reducir matriz de datos para que solo contenga los predictores
9 # empleados y la respuesta.
10 predictores <- names(coef(modelo))[-1]
11 datos <- datos[, c(predictores, "mpg")]
12
13 # Construir una matriz de datos con la respuesta predicha, los
14 # residuos y algunas estadísticas para evaluar la influencia de
15 # cada observación.
16 resultados <- data.frame(respuesta_predicha = fitted(modelo))

```

```

Call:
lm(formula = mpg ~ wt + qsec + am, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4811 -1.5555 -0.7257  1.4110  4.6610 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.6178    6.9596   1.382 0.177915    
wt          -3.9165    0.7112  -5.507 6.95e-06 ***  
qsec         1.2259    0.2887   4.247 0.000216 ***  
am           2.9358    1.4109   2.081 0.046716 *   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.459 on 28 degrees of freedom
Multiple R-squared:  0.8497, Adjusted R-squared:  0.8336 
F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11

AIC = 154.1194

```

Figura 15.12: modelo obtenido mediante eliminación hacia atrás.

```

Call:
lm(formula = mpg ~ wt + cyl + hp, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.9290 -1.5598 -0.5311  1.1850  5.8986 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 38.75179   1.78686  21.687 < 2e-16 ***  
wt          -3.16697   0.74058  -4.276 0.000199 ***  
cyl         -0.94162   0.55092  -1.709 0.098480 .    
hp          -0.01804   0.01188  -1.519 0.140015    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.512 on 28 degrees of freedom
Multiple R-squared:  0.8431, Adjusted R-squared:  0.8263 
F-statistic: 50.17 on 3 and 28 DF,  p-value: 2.184e-11

AIC = 155.4766

```

Figura 15.13: modelo obtenido mediante regresión escalonada.

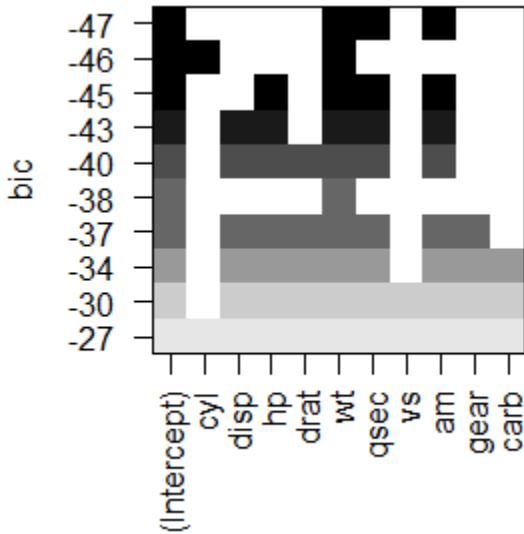


Figura 15.14: representación gráfica de los mejores modelos encontrados mediante el método de todos los subconjuntos.

```

17 resultados[["residuos_estandarizados"]] <- rstandard(modelo)
18 resultados[["residuos_estudiantizados"]] <- rstudent(modelo)
19 resultados[["distancia_Cook"]] <- cooks.distance(modelo)
20 resultados[["dfbeta"]] <- dfbeta(modelo)
21 resultados[["dffit"]] <- dffits(modelo)
22 resultados[["apalancamiento"]] <- hatvalues(modelo)
23 resultados[["covratio"]] <- covratio(modelo)
24
25 cat("Identificación de valores atípicos:\n")
26 # Observaciones con residuos estandarizados fuera del 95% esperado.
27 sospechosos1 <- which(abs(
28   resultados[["residuos_estandarizados"]]) > 1.96)
29
30 cat("- Residuos estandarizados fuera del 95% esperado:",
31   sospechosos1, "\n")
32
33 # Observaciones con distancia de Cook mayor a uno.
34 sospechosos2 <- which(resultados[["cooks.distance"]] > 1)
35
36 cat("- Residuos con una distancia de Cook alta:",
37   sospechosos2, "\n")
38
39 # Observaciones con apalancamiento mayor igual al doble del
40 # apalancamiento promedio.
41
42 apal_medio <- (ncol(datos) + 1) / nrow(datos)
43 sospechosos3 <- which(resultados[["apalancamiento"]] > 2 * apal_medio)
44
45 cat("- Residuos con apalancamiento fuera de rango:",
46   sospechosos3, "\n")
47
48 # Observaciones con DFBeta mayor o igual a 1.
49 sospechosos4 <- which(apply(resultados[["dfbeta"]]) >= 1, 1, any))
50 names(sospechosos4) <- NULL
51
52 cat("- Residuos con DFBeta >= 1:",
```

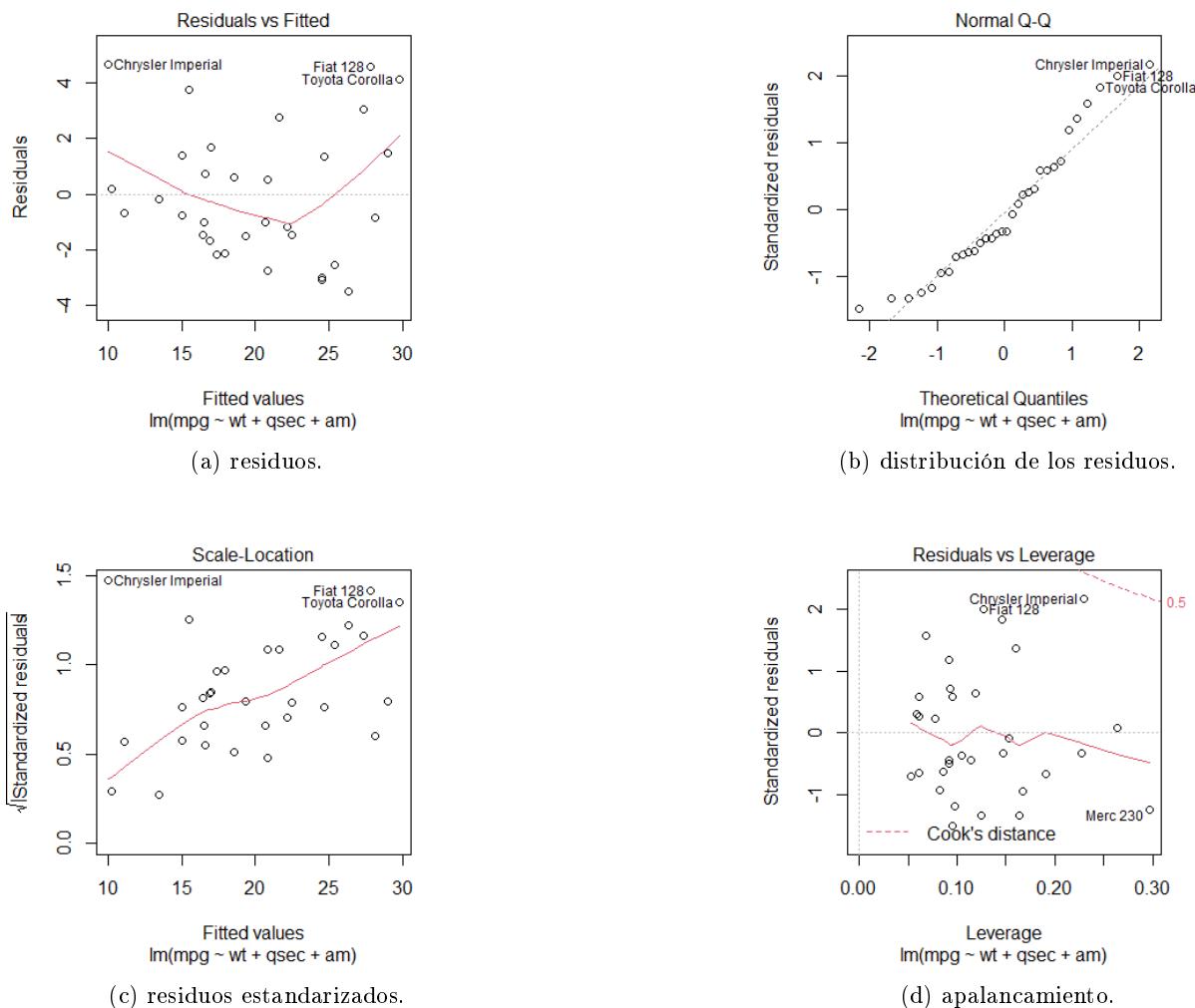


Figura 15.15: gráficos disponibles en R (base) para evaluar un modelo lineal.

```

53     sospechosos4, "\n")
54
55 # Observaciones con razón de covarianza fuera de rango.
56 inferior <- 1 - 3 * apal_medio
57 superior <- 1 + 3 * apal_medio
58 sospechosos5 <- which(resultados[["covratio"]] < inferior |
59                           resultados[["covratio"]] > superior)
60
61 cat("- Residuos con razón de covarianza fuera de rango:",
62     sospechosos5, "\n")
63
64 # Resumen de valores sospechosos.
65 sospechosos <- c(sospechosos1, sospechosos2, sospechosos3,
66                   sospechosos4, sospechosos5)
67
68 sospechosos <- sort(unique(sospechosos))
69
70 cat("\nResumen de valores sospechosos:\n")
71 cat("Apalancamiento promedio:", apal_medio, "\n")

```

```

72
73 cat("Intervalo razón de covarianza: [", inferior, "; ",
74   superior, "]\n\n", sep = "")
75
76 print(round(resultados[sospechosos, c("distancia_Cook", "apalancamiento",
77                           "covratio")], 3))

```

Identificación de valores atípicos:

- Residuos estandarizados fuera del 95% esperado: 17 18
- Residuos con una distancia de Cook alta:
- Residuos con apalancamiento fuera de rango: 9 16
- Residuos con DFBeta ≥ 1 : 3 5 6 9 25 28 32
- Residuos con razón de covarianza fuera de rango: 15 16

Resumen de valores sospechosos:

Apalancamiento promedio: 0.125

Intervalo razón de covarianza: [0.625; 1.375]

	distancia_Cook	apalancamiento	covratio
Datsun 710	0.058	0.095	0.919
Hornet Sportabout	0.013	0.093	1.182
Valiant	0.038	0.097	1.045
Merc 230	0.162	0.297	1.313
Cadillac Fleetwood	0.008	0.227	1.474
Lincoln Continental	0.001	0.264	1.570
Chrysler Imperial	0.348	0.230	0.724
Fiat 128	0.146	0.128	0.715
Pontiac Firebird	0.045	0.068	0.856
Lotus Europa	0.088	0.161	1.050
Volvo 142E	0.063	0.124	1.016

Figura 15.16: identificación de valores atípicos.

15.6.2 Verificación de las condiciones

A fin de que el modelo sea generalizable, tenemos que verificar el cumplimiento de las condiciones descritas en las primeras páginas de este capítulo. Es sencillo comprobar que las variables predictoras son dicotómicas o numéricas a nivel de intervalo y que ninguna de ellas corresponde a una constante. Adicionalmente, las observaciones son independientes entre sí por tratarse de modelos diferentes de automóviles que no parecen seguir un criterio de selección (más que los años de fabricación). A su vez, podemos comprobar que la variable dependiente es numérica a nivel de intervalo sin restricciones.

Al examinar la matriz de correlación (figura 14.4), sin embargo, podemos apreciar una relación positiva moderada entre el tiempo mínimo para recorrer un cuarto de milla y el rendimiento ($R = 0,419$).

La verificación de otras condiciones resulta algo más compleja, por lo que requiere de herramientas matemáticas adicionales.

15.6.2.1 Independencia de los residuos

Esta condición significa que no debe existir autocorrelación en los residuos. Podemos probarlo con una prueba estadística específica conocida con el nombre de sus autores: la prueba de Durbin-Watson, que verifica si dos residuos adyacentes (un retardo), o incluso más alejados, están correlacionados (Durbin & Watson, 1950, 1951). A diferencia de la mayoría de las pruebas de hipótesis, esta prueba define tres regiones: rechazo de H_0 , no rechazo de H_0 y una región no concluyente, por lo que existen dos valores críticos, los cuales se buscan en tablas publicadas.

La función `durbinWatsonTest(model)`, del paquete `car`, nos permite aplicar la prueba de Durbin-Watson a los residuos. Sin embargo, debemos tener en cuenta que los resultados de esta prueba dependen del orden de los datos, por lo que al reordenar los datos se podrían obtener resultados diferentes. Al aplicar esta prueba para el ejemplo (script 15.8, línea 12) obtenemos un valor $p = 0,236$, por lo que podemos concluir que los residuos son, en efecto, independientes.

15.6.2.2 Distribución normal de los residuos

Tal como mencionamos previamente, la figura 15.15b muestra que los residuos podrían alejarse un poco de la distribución normal. Al aplicar la prueba de Shapiro-Wilk (script 15.8, línea 16), obtenemos como resultado $p = 0,080$, por lo que podemos asumir que el supuesto se cumple, aunque manteniendo cautela por la cercanía con el nivel de significación.

15.6.2.3 Homocedasticidad de los residuos

El gráfico de la figura 15.15a muestra algunos residuos que se alejan levemente del rango general, pero que no parecen ser muy problemáticos.

Una prueba adecuada para verificar esta condición es la de Breusch-Pagan-Godfrey (Glen, 2016a), cuya hipótesis nula es que las varianzas de los residuos son iguales. En R, esta prueba está implementada en la función `ncvTest(model)` del paquete `car`. Al usarla para el ejemplo (script 15.8, línea 20), obtenemos como resultado $p = 0,212$, por lo que podemos concluir que el supuesto de homocedasticidad se cumple.

15.6.2.4 Multicolinealidad

Esta condición establece que no debe existir una relación lineal entre dos o más predictores. En otras palabras, la correlación entre variables no debe ser muy alta (o muy baja, si la relación es inversa).

Como hemos dejado entrever a lo largo de este capítulo, el incumplimiento de esta condición puede causar diversos problemas:

- Estimaciones poco confiables de los parámetros. Cuando dos (o más) predictores están perfectamente correlacionados, existen en realidad infinitos valores para sus parámetros que resuelven el problema

de optimización de minimizar la suma de los residuos cuadrados, por lo que la variabilidad de dichos parámetros puede ser muy elevada.

- Limita la magnitud de R , puesto que si los predictores están correlacionados en realidad explican la misma porción de la variabilidad de la respuesta.
- Dificulta evaluar la importancia de cada predictor, por el mismo motivo anterior.

Podemos revisar esta condición mediante el factor de inflación de varianza (*VIF*) y el estadístico tolerancia ($1/VIF$).

El *VIF* para cada predictor i se calcula mediante la ecuación 15.8, donde R_i^2 se obtiene usando todos los predictores restantes para ajustar una RLM con el predictor i como respuesta (Frost, 2021).

$$VIF_i = \frac{1}{1 - R_i^2} \quad (15.8)$$

Aunque no hay un acuerdo general, muchos autores usan el valor $VIF \geq 10$ como umbral para preocuparse, aunque hay autores que consideran críticos valores más conservadores, de 5 o incluso de 2,5 (Frost, 2021). También se ha encontrado que si el *VIF* promedio es mayor a 1, podría haber sesgo en el modelo.

En el caso de la tolerancia, la literatura sugiere que valores bajo 0,2 podrían ser problemáticos, aunque algunos académicos creen que valores cercanos a 0,4 deberían ser revisados.

El paquete `car` de R incluye la función `vif(model)` para calcular el factor de inflación de la varianza. Al usarla para el ejemplo (script 15.8, líneas 23–29) obtenemos los resultados de la figura 15.17.

```
Verificar la multicolinealidad:
- VIFs:
      wt      qsec       am
2.482952 1.364339 2.541437
- Tolerancias:
      wt      qsec       am
0.4027465 0.7329556 0.3934781
- VIF medio: 2.129576
```

Figura 15.17: verificación de condición de multicolinealidad.

Si miramos los factores de inflación de la varianza, en general no parecen ser preocupantes. Sin embargo, los estadísticos de tolerancia son preocupantes. Adicionalmente, el VIF promedio también indica que el modelo podría estar sesgado. Es necesario, entonces, buscar un modelo más confiable. Podríamos, por ejemplo, eliminar la variable menos significativa (`am`, que tiene el menor valor p), obteniendo el modelo con dos predictores de la figura 15.1, el que tendríamos que evaluar y comparar con este modelo de tres predictores (ver los últimos ejercicios propuestos).

Script 15.8: verificación de condiciones para el modelo.

```
1 library(car)
2
3 # Cargar datos.
4 datos <- mtcars
5
6 # Ajustar modelo.
7 modelo <- lm(mpg ~ wt + qsec + am, data = datos)
8
9 # Comprobar independencia de los residuos.
10 cat("Prueba de Durbin-Watson para autocorrelaciones ")
11 cat("entre errores:\n")
12 print(durbinWatsonTest(modelo))
13
```

```

14 # Comprobar normalidad de los residuos.
15 cat("\nPrueba de normalidad para los residuos:\n")
16 print(shapiro.test(modelo$residuals))
17
18 # Comprobar homocedasticidad de los residuos.
19 cat("Prueba de homocedasticidad para los residuos:\n")
20 print(ncvTest(modelo))
21
22 # Comprobar la multicolinealidad.
23 vifs <- vif(modelo)
24 cat("\nVerificar la multicolinealidad:\n")
25 cat("- VIFs:\n")
26 print(vifs)
27 cat("- Tolerancias:\n")
28 print(1 / vifs)
29 cat("- VIF medio:", mean(vifs), "\n")

```

15.6.3 Validación cruzada

Al igual que en el caso de regresión lineal simple, también podemos usar validación cruzada como herramienta para mejorar la estimación del error cuadrado medio. No hay diferencia en la realización de este proceso con respecto a lo estudiado en el capítulo anterior, por lo que no se aborda aquí con mayor detalle.

15.6.4 Tamaño de la muestra

Otro aspecto importante a tener en cuenta al momento de determinar si un modelo de RLM es confiable es el tamaño de la muestra. Existen distintas reglas que simplifican la tarea de determinar el tamaño de la muestra, pero lo cierto es que mientras más observaciones tengamos, mejor. Una de las reglas más simplistas en verificar que se tengan al menos 10 o 15 observaciones por cada predictor. De acuerdo a este criterio, la muestra del ejemplo cuenta con 32 observaciones, con 3 variables predictoras en el modelo, por lo que estaríamos en el borde inferior recomendado.

Considerando, además, que ya hemos encontrado indicios de que el modelo de tres variables podría no ser confiable, parece ser más pertinente usar un modelo más sencillo, como el de la figura 15.1.

15.7 EJERCICIOS PROPUESTOS

1. ¿En qué se diferencia la regresión lineal simple (RLS) de la regresión lineal multivariada (RLM)?
2. ¿Cómo luce la ecuación de un modelo RLM?
3. Si un modelo RLS y un modelo RLM usan una misma variable, ¿tendrán los mismos coeficientes?
4. ¿Qué son los predictores colineales y por qué hay que identificarlos?
5. Explica qué es el coeficiente de determinación R^2 ajustado y para qué se usa.

6. Explica qué son los modelos nulo y completo en el contexto RML.
7. Explica en qué consiste la estrategia de selección de variables hacia adelante.
8. Explica en qué consiste la estrategia de eliminación de variables hacia atrás.
9. Enumera las condiciones necesarias para aplicar RLM.
10. Explica qué se puede ver en los gráficos que entrega la función `plot(modelo)` cuando `modelo` es una RLM.
11. Verifica si el modelo presentado en la figura 15.1 cumple las condiciones para poder usar RLM.
12. ¿Son significativamente distintos los modelos de las figuras 15.1 y 15.12?

CAPÍTULO 16. REGRESIÓN LOGÍSTICA

En los capítulos 14 y 15 estudiamos la regresión lineal, útil para predecir una respuesta numérica a partir de una o más variables, aunque con una serie de condiciones. Como explican Field et al. (2012, pp. 312-345), la **regresión logística** es un **modelo lineal generalizado**, que admite una variable de respuesta cuyos residuos sigan una distribución diferente a la normal.

La regresión logística relaciona la distribución de la variable de respuesta con un modelo lineal usando como función de enlace la **función logística estándar**, también conocida como `logit()`, que presentamos en la ecuación 16.1 y mostramos gráficamente en la figura 16.1. Esta función describe una **transición de cero a uno**, por lo que resulta especialmente útil para representar la **probabilidad** de que ocurra algún evento: un valor cercano a cero indica que es muy poco probable, mientras un valor cercano a 1 corresponde a una alta probabilidad (lógicamente, un valor de 0,5 indica que es igualmente probable que el evento ocurra o no). Así, la regresión logística resulta adecuada para predecir una respuesta dicotómica, pues puede ser asociada a una **distribución binomial**.

$$f(z) = \frac{1}{1 + e^{-z}} \quad (16.1)$$

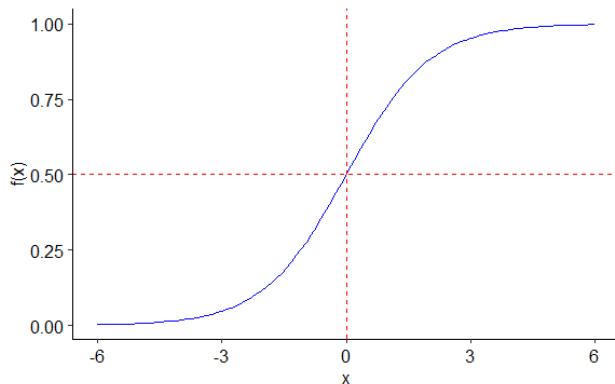


Figura 16.1: función logística.

Para entender mejor esta idea, necesitamos introducir el concepto de **odds** (Cerda et al., 2013), el cual no tiene una traducción directa al castellano, pero que puede entenderse como “oportunidad” o “chance”, aunque a veces se traduce incorrectamente como “probabilidad”. Matemáticamente, el **odds ratio** está dado por la ecuación 16.2, por lo que se define como la razón entre la probabilidad de que ocurra un evento y la probabilidad de que este no ocurra.

$$odds = \frac{p}{1 - p} \quad (16.2)$$

Tomemos el ejemplo que usan (Cerda et al., 2013): supongamos que los registros históricos dicen que en junio llueve 12 días. Así, la probabilidad de que un día de junio sea lluvioso es:

$$p = \frac{12}{30} = 0,4$$

Pero la oportunidad de que el día sea lluvioso es:

$$odds = \frac{12}{18} = 0,67$$

Ambas medidas presentan la misma información, pero de manera diferente. Cuando un evento e tiene las mismas posibilidades de ocurrir o no, $p(e) = 0,5$ y $odds(e) = 1$.

Suponiendo que el logaritmo de los *odds* sigue una distribución normal, podemos relacionar los *odds* y las probabilidades como muestran las ecuaciones 16.3 y 16.4.

$$z = \log\left(\frac{p}{1-p}\right) \quad (16.3)$$

$$p = \text{logit}(z) = \frac{1}{1 + e^{-z}} \quad (16.4)$$

A su vez, también podemos asociar z a otras variables, como muestra la ecuación 16.5.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (16.5)$$

Así, la regresión logística nos permite **asociar** la probabilidad de ocurrencia de un evento e a una combinación lineal de variables predictoras x_1, x_2, \dots, x_n , de acuerdo a la ecuación 16.6.

$$p(e) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}} \quad (16.6)$$

De esta forma podemos seleccionar una división o **umbral** que permita **predecir** la ocurrencia de un evento e o, de forma equivalente, **clasificar** un objeto en dos categorías posibles:

- Para valores menores que el umbral se predice que el evento e “no ocurre”, otorgándose una clasificación cero ($\hat{y} = 0$) o negativa ($\hat{y} = -$).
- Mientras que para valores mayores o iguales que el umbral se predice que el evento “sí ocurre”, a lo que corresponde una clasificación uno ($\hat{y} = 1$) o positiva ($\hat{y} = +$).

Si bien es usual utilizar el valor $p(e) = 0,5$ como umbral, esto no es obligatorio ni siempre conveniente, como veremos más adelante.

En capítulos precedentes explicamos que el ajuste de un modelo de regresión lineal se realiza mediante la resolución de un problema de optimización que busca minimizar la suma de las desviaciones cuadradas entre las respuestas predichas y las observadas. En el caso de la regresión logística, también se ajusta mediante la resolución de un problema de optimización, donde buscamos minimizar la diferencia entre las respuestas observadas y las respuestas predichas. Como las respuestas corresponden a una **variable dicotómica**, esta optimización se realiza usando la función de verosimilitud $\mathcal{L}(p)$, aunque, por conveniencia, se suele optimizar el logaritmo natural de la verosimilitud, tal y como se muestra en la ecuación 16.7.

$$\begin{aligned} \mathcal{L}(p) &= P(y_1, y_2, \dots, y_n | p) \text{ with } y_i \in \{0, 1\} \\ &= \prod_{i=1}^n p^{y_i} (1-p)^{1-y_i} \\ &\quad \text{Luego,} \\ \ln \mathcal{L}(p) &= \sum_{i=1}^n [y_i \ln p_{(y_i=1)} + (1-y_i) \ln (1-p_{(y_i=1)})] \end{aligned} \quad (16.7)$$

16.1 EVALUACIÓN DE UN CLASIFICADOR

Una forma de evaluar modelos de clasificación, entre ellos los de regresión logística, es de acuerdo a la cantidad de errores cometidos (Zelada, 2017). Para ello, el primer paso consiste en construir una tabla de contingencia (también llamada matriz de confusión) para las respuestas predichas y observadas, como muestra la tabla 16.1, bastante similar a la que ya conocimos para explicar los errores de decisión en la prueba de hipótesis (tabla 4.1). Las cuatro celdas de la matriz de confusión contienen:

- **Verdaderos positivos (VP):** cantidad de instancias correctamente clasificadas como pertenecientes a la clase positiva.
- **Falsos positivos (FP):** cantidad de instancias erróneamente clasificadas como pertenecientes a la clase positiva.
- **Falsos negativos (FN):** cantidad de instancias erróneamente clasificadas como pertenecientes a la clase negativa.
- **Verdaderos negativos (VN):** cantidad de instancias correctamente clasificadas como pertenecientes a la clase negativa.

		Real		Total
		1 (+)		
Clasificación	1 (+)	VP	FP	$VP + FP$
	0 (-)	FN	VN	$FN + VN$
Total		$VP + FN$	$FP + VN$	n

Tabla 16.1: tabla de contingencia para evaluar un clasificador.

La **exactitud** (*accuracy*) del modelo corresponde a la proporción de observaciones correctamente clasificadas, dada por la ecuación 16.8.

$$\text{exactitud} = \frac{VP + VN}{n} \quad (16.8)$$

A su vez, el **error** del modelo corresponde a la proporción de observaciones clasificadas de manera equivocada (ecuación 16.9).

$$\text{error} = \frac{FP + FN}{n} = 1 - \text{exactitud} \quad (16.9)$$

La **sensibilidad** (*sensitivity* o *recall*, ecuación 16.10) indica cuán apto es el modelo para detectar aquellas observaciones pertenecientes a la clase positiva.

$$\text{sensibilidad} = \frac{VP}{VP + FN} \quad (16.10)$$

De manera análoga, la **especificidad** (*specificity*, ecuación 16.11) permite determinar cuán exacta es la asignación de elementos a la clase positiva. También puede entenderse como la aptitud del modelo para correctamente asignar observaciones a la clase negativa.

$$\text{especificidad} = \frac{VN}{FP + VN} \quad (16.11)$$

La **precisión** (*precision*) o valor predictivo positivo (VPP , ecuación 16.12) indica la proporción de instancias clasificadas como positivas que realmente lo son.

$$VPP = \frac{VP}{VP + FP} \quad (16.12)$$

Asimismo, el **valor predictivo negativo** (*VPN*, ecuación 16.13) señala la proporción de instancias correctamente clasificadas como pertenecientes a la clase negativa.

$$VPN = \frac{VN}{FN + VN} \quad (16.13)$$

Otra herramienta útil es la **curva de calibración**, también llamada curva ROC por las siglas inglesas para *receiver-operating characteristic*, que muestra la relación entre la sensibilidad y la especificidad del modelo (Glen, 2017). Este gráfico también permite evaluar la precisión del modelo, puesto que mientras más se aleje la curva de la diagonal, mayor es la precisión. Para ilustrar mejor la utilidad de este gráfico, la figura 16.2 muestra las curvas ROC para dos modelos diferentes, además de la diagonal. Esta figura indica que el clasificador representado por la curva morada es mejor que el representado por la curva azul, pues se aleja más de la diagonal.

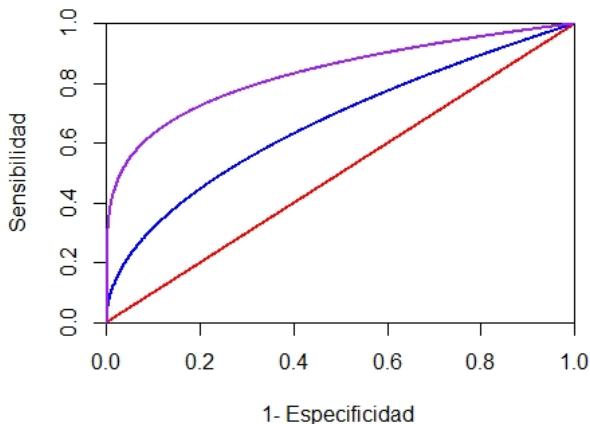


Figura 16.2: dos curvas ROC. Fuente: Ayala (2020).

16.2 BONDAD DE AJUSTE DEL MODELO

Al igual que en el caso de la regresión lineal, existen diversos mecanismos para evaluar el ajuste de un modelo de regresión logística. El estadístico de **log-verosimilitud** ($\ln \mathcal{L}$), dado por la ecuación 16.7, nos permite cuantificar la diferencia entre las probabilidades predichas y las observadas. Este estadístico se asemeja a la suma de los residuos cuadrados de la regresión lineal en el sentido de que cuantifica la cantidad de información que carece de explicación tras el ajuste del modelo. Así, mientras menor sea su valor, mejor es el ajuste del modelo.

La **desviación** (en inglés *deviance*), a menudo denotada por $-2LL$ y en pocas ocasiones llamada *devianza*, suele usarse en lugar de la log-verosimilitud porque sigue una distribución χ^2 , lo que facilita calcular el nivel de significación del valor. Está dada por la ecuación 16.14.

$$-2LL = -2 \cdot \ln \mathcal{L} \quad (16.14)$$

Los criterios de evaluación de modelos basados, en el principio de parsimonia que estudiamos en el capítulo 15, también están definidos para la regresión logística. El más sencillo es el criterio de información de Akaike (*AIC*), dado por la ecuación 16.15, donde k corresponde a la cantidad de predictores en el modelo.

$$AIC = -2LL + 2k \quad (16.15)$$

Similar al *AIC*, el criterio bayesiano de Schwarz (*BIC*) ajusta la penalización a la complejidad del modelo según el tamaño de la muestra, como muestra la ecuación 16.16.

$$BIC = -2LL + 2k \cdot \ln n \quad (16.16)$$

16.3 REGRESIÓN LOGÍSTICA EN R

En R, podemos ajustar un modelo de regresión logística mediante la función `glm(formula, family = binomial(link = "logit"), data)`, donde:

- `formula` tiene la forma <variable de respuesta> \sim <variable predictora>.
- `data`: matriz de datos.

Puesto que existen otros modelos generalizados de regresión lineal, el argumento `family = binomial(link = "logit")` indica que asumiremos una distribución binomial para la variable de respuesta y que usaremos la función logística.

Las líneas 11–19 del script 16.1 ilustran el uso de la función `glm()` para ajustar un modelo de regresión logística que prediga el tipo de transmisión de un automóvil (0 = automática, 1 = manual) a partir de su peso, usando para ello el ya conocido conjunto de datos `mtcars` disponible en R (recordemos que podemos consultar la descripción de las variables en la tabla 14.1). Para ello, consideramos un conjunto de entrenamiento con 80 % de las instancias. El modelo resultante se muestra en la figura 16.3, donde podemos apreciar que el *AIC* es bastante bajo (*AIC* = 16,23) y que la desviación del modelo con una variable (23 grados de libertad) es de 12,23.

Las líneas 22–33 evalúan el modelo ajustado usando las herramientas descritas en la sección anterior y el conjunto de entrenamiento. La función `roc(response, predictor)` del paquete `pROC`, donde los argumentos corresponden, respectivamente, a las respuestas observadas y las respuestas predichas, nos permite obtener la curva ROC de la figura 16.4. La curva se aleja bastante de la diagonal, por lo que al parecer se trata de un buen modelo. A su vez, la función `confusionMatrix(data, reference)` del paquete `caret`, donde `data` corresponde a la respuesta predicha y `reference` a la observada, genera la matriz de confusión y obtiene las medidas de evaluación descritas anteriormente, como muestra la figura 16.5. Podemos ver que el modelo tiene una exactitud de 92,0 %. La sensibilidad de 100 % y la especificidad de 83,33 % muestran que el modelo se desempeña un poco mejor identificando elementos de la clase positiva, correspondiente en este caso a los vehículos de transmisión automática.

Pero, como ya hemos estudiado en capítulos anteriores, debemos evaluar el modelo con un conjunto de datos diferente al que usamos para su construcción. Así, las líneas 36–46 obtienen la curva ROC (figura 16.6) y la matriz de confusión (figura 16.7) para el conjunto de prueba, donde observamos un resultado con menor exactitud que con el conjunto de entrenamiento. Esto es una indicación de que el modelo podría estar un poco sobreajustado para el conjunto de entrenamiento, pero también de que el conjunto de prueba puede ser muy pequeño para obtener una evaluación confiable.

```

Call:
glm(formula = am ~ wt, family = binomial(link = "logit"), data = entrenamiento)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-1.17498 -0.40172 -0.00176  0.12321  2.26151 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) 18.525     8.504    2.178   0.0294 *  
wt          -5.883     2.645   -2.224   0.0261 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 34.617  on 24  degrees of freedom
Residual deviance: 12.230  on 23  degrees of freedom
AIC: 16.23

Number of Fisher Scoring iterations: 7

```

Figura 16.3: ajuste de un modelo de regresión logística.

Script 16.1: ajuste de un modelo de regresión logística en R.

```

1 library(pROC)
2 library(caret)
3
4 set.seed(1313)
5
6 # Cargar los datos.
7 datos <- mtcars
8 datos$am <- factor(datos$am)
9
10 # Separar conjuntos de entrenamiento y prueba.
11 n <- nrow(datos)
12 n_entrenamiento <- floor(0.8 * n)
13 muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
14 entrenamiento <- datos[muestra, ]
15 prueba <- datos[-muestra, ]
16
17 # Ajustar modelo.
18 modelo <- glm(am ~ wt, family = binomial(link = "logit"), data = entrenamiento)
19 print(summary(modelo))
20
21 # Evaluar el modelo con el conjunto de entrenamiento.
22 cat("Evaluación del modelo a partir del conjunto de entrenamiento:\n")
23 probs_e <- predict(modelo, entrenamiento, type = "response")
24
25 umbral <- 0.5
26 preds_e <- sapply(probs_e, function(p) ifelse(p >= umbral, "1", "0"))
27 preds_e <- factor(preds_e, levels = levels(datos[["am"]]))
28
29 ROC_e <- roc(entrenamiento[["am"]], probs_e)
30 plot(ROC_e)

```

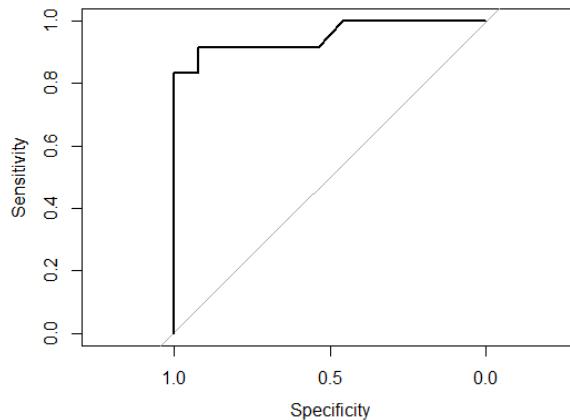


Figura 16.4: curva ROC obtenida al evaluar el modelo con el conjunto de entrenamiento.

```

31
32 matriz_e <- confusionMatrix(preds_e, entrenamiento[["am"]])
33 print(matriz_e)
34
35 # Evaluar el modelo con el conjunto de prueba.
36 cat("Evaluación del modelo a partir del conjunto de prueba:\n")
37 probs_p <- predict(modelo, prueba, type = "response")
38
39 preds_p <- sapply(probs_p, function(p) ifelse(p >= umbral, "1", "0"))
40 preds_p <- factor(preds_p, levels = levels(datos[["am"]]))
41
42 ROC_p <- roc(prueba[["am"]], probs_p)
43 plot(ROC_p)
44
45 matriz_p <- confusionMatrix(preds_p, prueba[["am"]])
46 print(matriz_p)

```

16.4 CONDICIONES PARA USAR REGRESIÓN LOGÍSTICA

Desde luego, no basta con evaluar el desempeño del clasificador, sino que también necesitamos verificar el cumplimiento de ciertas condiciones para que un modelo de regresión logística sea válido:

1. Debe existir una relación lineal entre los predictores y la respuesta transformada.
2. Los residuos deben ser independientes entre sí.

Además de las condiciones anteriores, existen otras situaciones en que puede ocurrir que el método de optimización no converja:

1. Multicolinealidad entre los predictores, que en este caso se aborda del mismo modo que para RLM (por ejemplo, mediante el factor de inflación de la varianza o la tolerancia).
2. Información incompleta, que se produce cuando no contamos con observaciones suficientes para todas las posibles combinaciones de predictores.
3. Separación perfecta, que ocurre cuando no hay superposición entre las clases, es decir, ¡cuando los predictores separan ambas clases completamente!

```

Confusion Matrix and Statistics

                    Reference
Prediction  0   1
          0 13  2
          1   0 10

Accuracy : 0.92
95% CI  : (0.7397, 0.9902)
No Information Rate : 0.52
P-Value [Acc > NIR] : 2.222e-05

Kappa : 0.8387

McNemar's Test P-Value : 0.4795

Sensitivity : 1.0000
Specificity : 0.8333
Pos Pred Value : 0.8667
Neg Pred Value : 1.0000
Prevalence : 0.5200
Detection Rate : 0.5200
Detection Prevalence : 0.6000
Balanced Accuracy : 0.9167

'Positive' Class : 0

```

Figura 16.5: matriz de confusión y medidas de evaluación con el conjunto de entrenamiento para el modelo ajustado.

16.5 GENERALIZACIÓN DEL MODELO

En capítulos anteriores conocimos la validación cruzada como herramienta para mejorar la estimación del error, la cual podemos usar de manera análoga para regresión logística. El script 16.2 mejora el ejercicio realizado en el script 16.1, incorporando el uso de validación cruzada de 5 pliegues. Notemos que la llamada a la función `train()` también solicita que “se guarden” los valores predichos, lo que nos permite estimar el rendimiento promedio del modelo como si se repitiera el script 16.2, seleccionando aleatoriamente un conjunto de entrenamiento y otro de prueba, cinco veces.

Debemos fijarnos en que el modelo obtenido es idéntico al anterior (por lo que no se muestra aquí), ya que la función `train()` reentrena el modelo del pliegue que obtuvo mejor rendimiento con todos los datos disponibles. En el caso de la regresión logística (como con la regresión lineal), los pliegues solo se diferencian en los datos que utilizan, por lo que siempre se llega al mismo modelo. Esto no sería así si la validación cruzada se usara, por ejemplo, para seleccionar las variables predictoras a incluir en el modelo.

Script 16.2: ajuste de un modelo de regresión logística usando validación cruzada.

```

1 library(caret)
2
3 set.seed(1313)
4
5 # Cargar los datos.
6 datos <- mtcars
7 datos$am <- factor(datos$am)

```

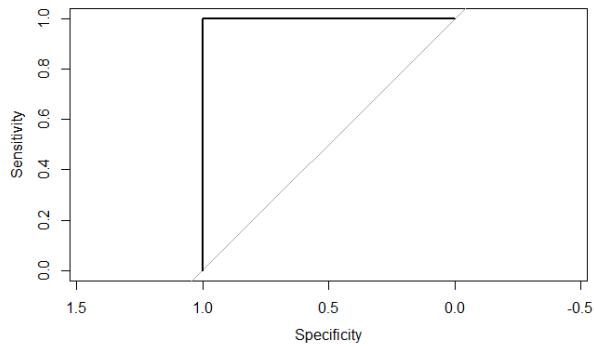


Figura 16.6: curva ROC obtenida al evaluar el modelo con el conjunto de prueba.

```

8
9 # Ajustar modelo usando validación cruzada de 5 pliegues.
10 modelo <- train(am ~ wt, data = entrenamiento, method = "glm",
11   family = binomial(link = "logit"),
12   trControl = trainControl(method = "cv", number = 5,
13   savePredictions = TRUE))
14
15 print(summary(modelo))
16
17 # Evaluar el modelo
18 cat("Evaluación del modelo basada en validación cruzada:\n")
19 matriz <- confusionMatrix(modelo$pred$pred, modelo$pred$obs)
20 print(matriz)

```

16.6 SELECCIÓN DE PREDICTORES

Cuando tenemos múltiples predictores potenciales, debemos decidir cuáles de ellos incorporar en el modelo. Una vez más, y tal como detallamos en el capítulo 15, el ideal es usar la regresión jerárquica para escoger los predictores de acuerdo a evidencia disponible en la literatura. Sin embargo, al explorar los datos, podemos emplear los demás métodos ya descritos: selección hacia adelante, eliminación hacia atrás, regresión escalonada o todos los subconjuntos. Se usan para ello las mismas funciones de R descritas en el capítulo 15.

16.7 COMPARACIÓN DE MODELOS

Al igual que con los modelos de regresión lineal, podemos comparar modelos de regresión logística mediante la función `anova()`, aunque ahora la prueba F resulta inapropiada. En cambio, una prueba muy utilizada en este caso es el *Likelihood Ratio Test* (LRT), el cual compara qué tanto más “probables” son los datos con un modelo que con el otro. Podemos ver un ejemplo de esta comparación más adelante en el script 16.3.

Confusion Matrix and Statistics

```

Reference
Prediction 0 1
          0 5 0
          1 1 1

Accuracy : 0.8571
95% CI  : (0.4213, 0.9964)
No Information Rate : 0.8571
P-Value [Acc > NIR] : 0.7365

Kappa : 0.5882

McNemar's Test P-Value : 1.0000

Sensitivity : 0.8333
Specificity : 1.0000
Pos Pred Value : 1.0000
Neg Pred Value : 0.5000
Prevalence : 0.8571
Detection Rate : 0.7143
Detection Prevalence : 0.7143
Balanced Accuracy : 0.9167

'Positive' Class : 0

```

Figura 16.7: matriz de confusión y medidas de evaluación con el conjunto de prueba para el modelo ajustado.

16.8 REGRESIÓN LOGÍSTICA EN R CON SELECCIÓN DE PREDICTORES

En páginas previas ajustamos un modelo de regresión logística para determinar el tipo de transmisión de un automóvil a partir de su peso. Sin embargo, el predictor fue seleccionado de manera aleatoria, simplemente para ilustrar el proceso, por lo que podríamos encontrar un mejor modelo usando algún método de selección de predictores. Las líneas 19–32 del script 16.3 llevan a cabo esta tarea usando regresión escalonada, obteniéndose como resultado el modelo presentado en la figura 16.8.

Sin embargo, al ajustar este modelo R emite algunas advertencias, como muestra la figura 16.9. Estas ocurren cuando los predictores separan completamente las clases, o bien cuando existen problemas de colinealidad. Al verificar los factores de inflación de la varianza de los predictores (script 16.3, líneas 35–41) podemos apreciar que, si bien ninguna de las variables presenta un *VIF* superior a 10, el promedio es bastante superior a 1 (figura 16.10), lo que confirma que el modelo puede tener problemas. En consecuencia, no es recomendable usar este modelo.

Por regla general, se recomienda eliminar la variable con mayor *VIF*, pero en este caso ambos son iguales. En consecuencia, en las líneas 44–57 del script 16.3 se ajustan los dos modelos posibles (figuras 16.11 y 16.12) y luego se comparan (figura 16.13). Pero en este caso la prueba ANOVA no sirve, pues ambos modelos tienen igual cantidad de predictores y no entrega un valor p. Sin embargo, podemos ver que el *VIF* del modelo con la potencia como predictor (*VIF* = 37,444) es más alto que para el modelo con el peso como predictor (*VIF* = 16,23). En consecuencia, este último parece ser mejor.

A modo de ejercicio, a pesar de que lo descartamos por tener problemas de colinealidad, comparamos el

```

Call:
glm(formula = am ~ wt + hp, family = binomial(link = "logit"),
     data = entrenamiento)

Deviance Residuals:
    Min          1Q      Median          3Q      Max
-3.378e-05 -2.100e-08 -2.100e-08  2.100e-08  2.597e-05

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 4.165e+02 3.478e+05  0.001   0.999
wt         -1.555e+02 1.287e+05 -0.001   0.999
hp          4.788e-01 4.620e+02  0.001   0.999

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 3.4617e+01 on 24 degrees of freedom
Residual deviance: 2.2982e-09 on 22 degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 25

```

Figura 16.8: modelo de regresión logística obtenido mediante regresión escalonada.

```

Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred

```

Figura 16.9: modelo de regresión logística obtenido mediante regresión escalonada.

modelo obtenido mediante regresión escalonada con el que tiene al peso como variable predictora (script 16.3, línea 68), obteniendo como resultado un valor $p < 0,001$ (figura 16.14). Puesto que el valor p obtenido es significativo, la prueba arroja que el modelo más complejo (es decir, el que tiene dos predictores) reduce la varianza de los residuos de forma significativa (¡llegando a cero!).

Dado que el mejor modelo es el mismo que habíamos usado en secciones previas, no repetiremos la evaluación con el conjunto de prueba, pues ya presentamos el resultado en la figura 16.7.

Sin embargo, aún resta verificar el cumplimiento de la condición de independencia de los residuos, para lo cual, al igual que con modelos de regresión lineal, empleamos la prueba de Durbin-Watson (script 16.3, línea 75), cuyo resultado mostramos en la figura 16.15, donde podemos notar que, aunque cerca del borde para $\alpha = 0,05$, los residuos son independientes.

```

Verificación de colinealidad
-----
VIF:
      wt      hp
4.14191 4.14191

Promedio VIF: [1] 4.14191

```

Figura 16.10: factores de inflación de la varianza para los modelos.

```

Call:
glm(formula = am ~ wt, family = binomial(link = "logit"), data = entrenamiento)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-1.17498 -0.40172 -0.00176  0.12321  2.26151 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) 18.525     8.504    2.178   0.0294 *  
wt          -5.883     2.645   -2.224   0.0261 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 34.617  on 24  degrees of freedom
Residual deviance: 12.230  on 23  degrees of freedom
AIC: 16.23

Number of Fisher Scoring iterations: 7

```

Figura 16.11: modelo de regresión logística con el peso como predictor.

Una vez verificadas las condiciones, podemos concluir que el modelo es adecuado y puede ser generalizado. No obstante, aún falta determinar si su ajuste se ve afectado por la presencia de valores atípicos (script 16.3, líneas 78–137). La figura 16.16 muestra los gráficos asociados al modelo. El gráfico de la figura 16.16a muestra una única instancia cuyo residuo se aleja muchísimo de los demás, correspondiente al Maserati Bora, el cual también se aleja bastante de la recta esperada en el gráfico Q-Q de los residuos (figura 16.16b). A su vez, la figura 16.16d muestra claramente que esa misma instancia ejerce un importante apalancamiento.

Usando herramientas más precisas para replicar el análisis, parte de cuyos resultados presentamos en la figura 16.17¹, podemos determinar que la única observación cuyo residuo estandarizado escapa a la normalidad es el Maserati Bora. Asimismo, esta instancia presenta la mayor distancia de Cook y los mayores DFBeta, por lo que es la única que resulta preocupante y podría ser útil eliminarla para el ajuste del modelo. También queda como ejercicio reentrenar y evaluar el modelo sin considerar esta observación.

Script 16.3: ajuste y evaluación del mejor modelo para predecir el tipo de transmisión de un automóvil.

```

1 library(car)
2
3 set.seed(1313)
4
5 # Cargar los datos.
6 datos <- mtcars
7 am <- factor(datos$am)
8 datos$am <- NULL
9 datos <- cbind(am, datos)
10
11 # Separar conjuntos de entrenamiento y prueba.
12 n <- nrow(datos)
13 n_entrenamiento <- floor(0.8 * n)
14 muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
15 entrenamiento <- datos[muestra, ]

```

¹ Por una cuestión de espacio, queda como ejercicio para el lector ejecutar el script 16.3 y ver el detalle de los valores obtenidos para las distintas métricas evaluadas para las observaciones sospechosas.

```

Call:
glm(formula = am ~ hp, family = binomial(link = "logit"), data = entrenamiento)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-1.3663 -1.0648 -0.8953  1.1182  1.7415 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  0.820490  0.941997  0.871   0.384    
hp          -0.006236  0.005965 -1.045   0.296    
                                                        
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 34.617 on 24 degrees of freedom
Residual deviance: 33.444 on 23 degrees of freedom
AIC: 37.444

Number of Fisher Scoring iterations: 4

```

Figura 16.12: modelo de regresión logística con la potencia como predictor.

Analysis of Deviance Table

Model 1: am ~ wt					
Model 2: am ~ hp					
Resid.	Df	Resid.	Df	Dev	Pr(>Chi)
1	23			12.230	
2	23		0	33.444	-21.214

Figura 16.13: comparación de los modelos con un único predictor.

```

16 prueba <- datos[-muestra, ]
17
18 # Ajustar modelo nulo.
19 nulo <- glm(am ~ 1, family = binomial(link = "logit"), data = entrenamiento)
20
21 # Ajustar modelo completo.
22 cat("\n\n")
23 completo <- glm(am ~ ., family = binomial(link = "logit"),
24                   data = entrenamiento)
25
26 # Ajustar modelo con regresión escalonada.
27 cat("Modelo con regresión escalonada\n")
28 cat("-----\n")
29 mejor <- step(nulo, scope = list(lower = nulo, upper = completo),
30                  direction = "both", trace = 0)
31
32 print(summary(mejor))
33
34 # Verificación de multicolinealidad.
35 cat("Verificación de colinealidad\n")
36 cat("-----\n")
37 cat("\nVIF:\n")
38 vifs <- vif(mejor)

```

Analysis of Deviance Table

```

Model 1: am ~ wt
Model 2: am ~ wt + hp
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1          23      12.23
2          22      0.00  1     12.23 0.0004704 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Figura 16.14: comparación del modelo con dos predictores y el que solo tiene el peso como predictor.

```

lag Autocorrelation D-W Statistic p-value
 1      -0.30715746    2.583329   0.084
Alternative hypothesis: rho[lag] != 0

```

Figura 16.15: resultado de la prueba de Durbin-Watson para verificar la independencia de los residuos del modelo que solo tiene el peso como predictor.

```

39 print(vifs)
40 cat("\nPromedio VIF: ")
41 print(mean(vifs))
42
43 # Ajustar modelo con el peso como predictor.
44 cat("Modelo con el peso como predictor\n")
45 cat("-----\n")
46 modelo_peso <- glm(am ~ wt, family = binomial(link = "logit"),
47                      data = entrenamiento)
48
49 print(summary(modelo_peso))
50
51 # Ajustar modelo con la potencia como predictor.
52 cat("Modelo con la potencia como predictor\n")
53 cat("-----\n")
54 modelo_potencia <- glm(am ~ hp, family = binomial(link = "logit"),
55                           data = entrenamiento)
56
57 print(summary(modelo_potencia))
58
59 # Comparar los modelos con el peso y la potencia como predictores.
60 cat("\n\n")
61 cat("Likelihood Ratio Test para los modelos\n")
62 cat("-----\n")
63 print(anova(modelo_peso, modelo_potencia, test = "LRT"))
64
65 # A modo de ejercicio, comparar el modelo obtenido mediante
66 # regresión escalonada con el que solo tiene el peso como predictor.
67 cat("\n\n")
68 cat("Likelihood Ratio Test para los modelos\n")
69 cat("-----\n")
70 print(anova(modelo_peso, mejor, test = "LRT"))
71
72 # Independencia de los residuos.
73 cat("Verificación de independencia de los residuos\n")
74 cat("-----\n")
75 print(durbinWatsonTest(modelo_peso, max.lag = 5))

```

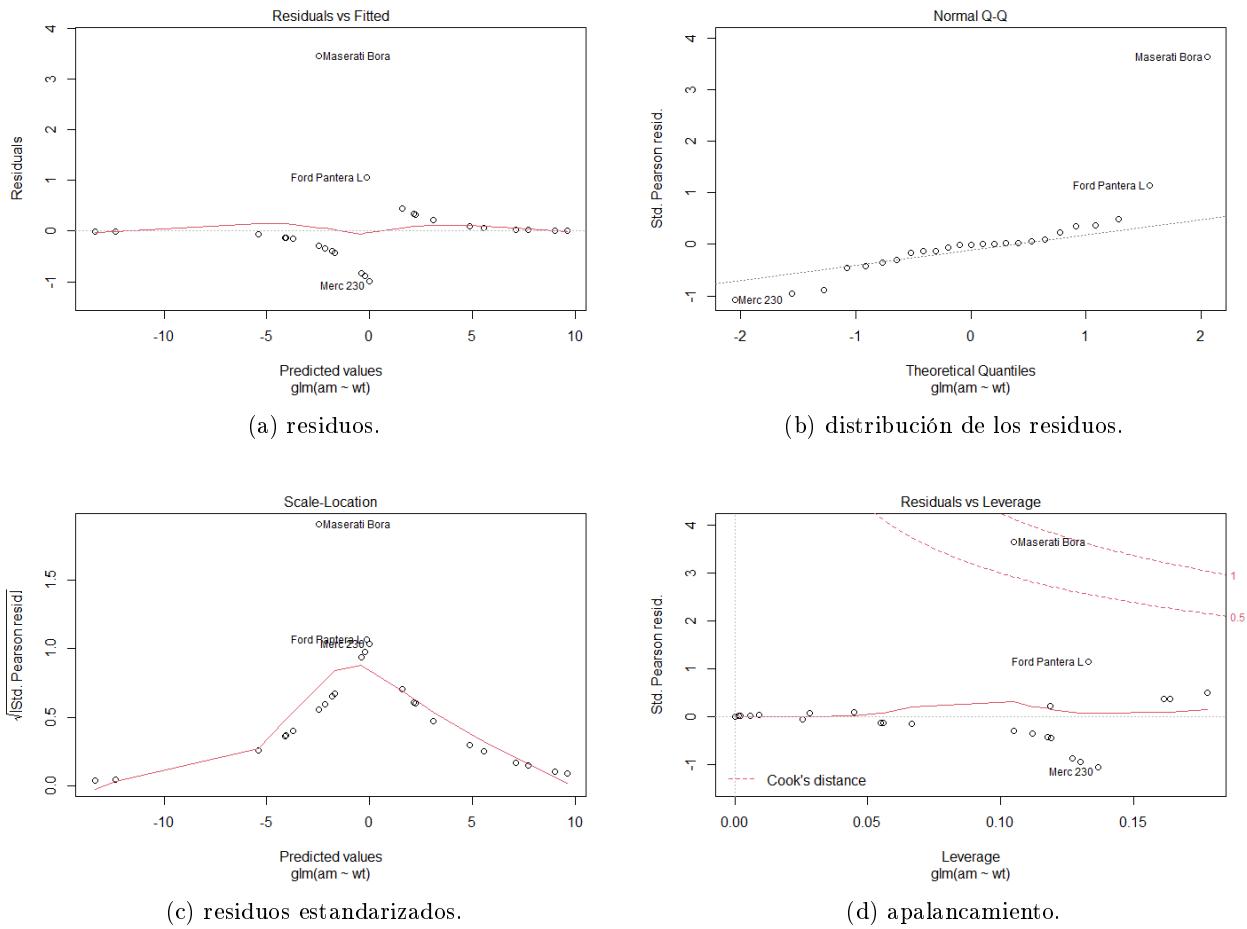


Figura 16.16: gráficos para evaluar el modelo de regresión logística.

```

76
77 # Detectar posibles valores atípicos.
78 cat("Identificación de posibles valores atípicos\n")
79 cat("-----\n")
80 plot(mejor)
81
82 # Obtener los residuos y las estadísticas.
83 output <- data.frame(predicted.probabilities = fitted(modelo_peso))
84 output[["standardized.residuals"]] <- rstandard(modelo_peso)
85 output[["studentized.residuals"]] <- rstudent(modelo_peso)
86 output[["cooks.distance"]] <- cooks.distance(modelo_peso)
87 output[["dfbeta"]] <- dfbeta(modelo_peso)
88 output[["dffit"]] <- dffits(modelo_peso)
89 output[["leverage"]] <- hatvalues(modelo_peso)
90
91 # Evaluar residuos estandarizados que escapan a la normalidad.
92 # 95 % de los residuos estandarizados deberían estar entre
93 # -1.96 y 1.96, y 99 % entre -2.58 y 2.58.
94 sospechosos1 <- which(abs(output[["standardized.residuals"]]) > 1.96)
95 sospechosos1 <- sort(sospechosos1)
96 cat("\n\n")
97 cat("Residuos estandarizados fuera del 95 % esperado\n")

```

```

Residuales estandarizados fuera del 95% esperado
-----
[1] "Maserati Bora"

Residuales con una distancia de Cook alta
-----
character(0)

Residuales con leverage fuera de rango (> 0.344)
-----
character(0)

Residuales con DFBeta sobre 1
-----
[1] "Dodge Challenger" "Maserati Bora"      "Merc 280"
[4] "Valiant"          "Ferrari Dino"       "Volvo 142E"
[7] "Mazda RX4 Wag"

Casos sospechosos
-----
      am mpg cyl disp hp drat wt qsec vs gear carb
Dodge Challenger 0 15.5   8 318.0 150 2.76 3.520 16.87 0   3   2
Maserati Bora    1 15.0   8 301.0 335 3.54 3.570 14.60 0   5   8
Merc 280          0 19.2   6 167.6 123 3.92 3.440 18.30 1   4   4
Valiant           0 18.1   6 225.0 105 2.76 3.460 20.22 1   3   1
Ferrari Dino     1 19.7   6 145.0 175 3.62 2.770 15.50 0   5   6
Volvo 142E        1 21.4   4 121.0 109 4.11 2.780 18.60 1   4   2
Mazda RX4 Wag    1 21.0   6 160.0 110 3.90 2.875 17.02 0   4   4

```

Figura 16.17: identificación de posibles valores atípicos.

```

98 cat("-----\n")
99 print(rownames(entrenamiento[sospechosos1, ]))

100
101 # Revisar casos con distancia de Cook mayor a uno.
102 sospechosos2 <- which(output[["cooks.distance"]] > 1)
103 sospechosos2 <- sort(sospechosos2)
104 cat("\n\n")
105 cat("Residuales con una distancia de Cook alta\n")
106 cat("-----\n")
107 print(rownames(entrenamiento[sospechosos2, ]))

108
109 # Revisar casos cuyo apalancamiento sea más del doble
110 # o triple del apalancamiento promedio.
111 leverage.promedio <- ncol(entrenamiento) / nrow(datos)
112 sospechosos3 <- which(output[["leverage"]] > leverage.promedio)
113 sospechosos3 <- sort(sospechosos3)
114 cat("\n\n")
115 cat("Residuales con leverage fuera de rango (> ")
116 cat(round(leverage.promedio, 3), ") ", "\n", sep = "")
117 cat("-----\n")
118 print(rownames(entrenamiento[sospechosos3, ]))

119
120 # Revisar casos con DFBeta >= 1.
121 sospechosos4 <- which(apply(output[["dfbeta"]], >= 1, 1, any))

```

```

122 sospechosos4 <- sort(sospechosos4)
123 names(sospechosos4) <- NULL
124 cat("\n\n")
125 cat("Residuales con DFBeta sobre 1\n")
126 cat("-----\n")
127 print(rownames(entrenamiento[sospechosos4, ]))
128
129 # Detalle de las observaciones posiblemente atípicas.
130 sospechosos <- c(sospechosos1, sospechosos2, sospechosos3, sospechosos4)
131 sospechosos <- sort(unique(sospechosos))
132 cat("\n\n")
133 cat("Casos sospechosos\n")
134 cat("-----\n")
135 print(entrenamiento[sospechosos, ])
136 cat("\n\n")
137 print(output[sospechosos, ])

```

16.9 EJERCICIOS PROPUESTOS

1. ¿Qué es un modelo lineal generalizado?
2. ¿Qué modela una regresión logística?
3. Explica por qué este modelo lleva el apellido “logística”.
4. ¿Por qué se considera un modelo lineal?
5. Menciona las condiciones necesarias para aplicar regresión logística.
6. Explica las evaluaciones que deben aplicarse a un modelo de regresión logística.
7. Investiga cuáles son las hipótesis que se contrastan al hacer inferencia con la regresión logística.
8. ¿Se podría buscar un buen modelo de regresión logística usando el paquete **caret**? Investigue.
9. Reconstruye el último modelo de regresión logística que conseguimos, pero ahora sin considerar el “Maserati Bora” y evalúa si cumple las condiciones para ser usado.

REFERENCIAS

- Agresti, A. (2019). *An introduction to categorical data analysis* (3.^a ed.). John Wiley & Sons, Inc.
- Agresti, A., & Coull, B. A. (1998). Approximate is better than “exact” for interval estimation of binomial proportions. *The American Statistician*, 52(2), 119-126.
- Amat Rodrigo, J. (2016a). *Resampling: test de permutación, simulación de Monte Carlo y Bootstrapping*. Consultado el 31 de mayo de 2021, desde https://www.cienciadedatos.net/documentos/23_resampling_test_permutacion_simulacion_de_monte_carlo_bootstrapping
- Amat Rodrigo, J. (2016b). *Test de Friedman*. Consultado el 29 de mayo de 2021, desde https://www.cienciadedatos.net/documentos/21_friedman_test
- Amat Rodrigo, J. (2016c). *Test Kruskal-Wallis*. Consultado el 29 de mayo de 2021, desde https://www.cienciadedatos.net/documentos/20_kruskal-wallis_test
- Amat Rodrigo, J. (2016d). *Validación de modelos predictivos: Cross-validation, OneLeaveOut, Bootstrapping*. Consultado el 23 de diciembre de 2021, desde https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap#K-Fold_Cross-Validation
- Ayala, J. (2020). *Minería de datos*. Consultado el 23 de junio de 2021, desde <https://rpubs.com/JairoAyala/592802>
- Bache, S. (2014). *Introducing magrittr*. Consultado el 7 de abril de 2021, desde <https://cran.r-project.org/web/packages/magrittr/vignettes/magrittr.html>
- Baguley, T. (2012). *Beware the Friedman test!* Consultado el 13 de diciembre de 2021, desde <https://seriousstats.wordpress.com/2012/02/14/friedman/>
- Berman, H. (2000). *Scheffé’s Test for Multiple Comparisons*. Consultado el 7 de mayo de 2021, desde <https://stattrek.com/anova/follow-up-tests/scheffe.aspx>
- Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical science*, 16(2), 101-117.
- Carchedi, N., De Mesmaeker, D., & Vannoorenberghe, L. (s.f.). RDocumentation. Consultado el 2 de abril de 2021, desde <https://www.rdocumentation.org/>
- Cerda, J., Vera, C., & Rada, G. (2013). Odds ratio: aspectos teóricos y prácticos. *Revista médica de Chile*, 141, 1329-1335.
- Champely, S., Ekstrom, C., Dalgaard, P., Gill, J., Weibelzahl, S., Anandkumar, A., Ford, C., Volcic, R., & de Rosario, H. (2020). *pwr: Basic Functions for Power Analysis*. Consultado el 1 de octubre de 2021, desde <https://cran.r-project.org/web/packages/pwr/vignettes/pwr-vignette.html>
- Cross, J. (2017). *Discrete Random Variables*. Consultado el 9 de abril de 2021, desde <https://rpubs.com/jcross/discreteRV>
- Dagnino, J. (2014). Tipos de datos y escalas de medida. *Revista Chilena de Anestesia*, 42(2), 109-111.
- Devore, J. L. (2008). *Probabilidad y Estadística para Ingeniería y Ciencias* (7.^a ed.). CENGAGE Learning.
- Diez, D., Barr, C. D., & Çetinkaya-Rundel, M. (2017). *OpenIntro Statistics* (3.^a ed.). <https://www.openintro.org/book/os/>.
- Durbin, J., & Watson, G. S. (1950). Testing for serial correlation in least squares regression: I. *Biometrika*, 37(3/4), 409-428.
- Durbin, J., & Watson, G. S. (1951). Testing for serial correlation in least squares regression: II. *Biometrika*, 38(1/2), 159-179.
- Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R*. SAGE Publications Ltd.
- Freedman, D. A. (2009). *Modelización*. Cambridge University Press.
- Freund, R. J., & Wilson, W. J. (2003). *Statistical Methods* (2.^a ed.). Academic Press.
- Frost, J. (2021). *Variance Inflation Factors (VIFs)*. Consultado el 17 de junio de 2021, desde <https://statisticsbyjim.com/regression/variance-inflation-factors/>

- Glen, S. (2016a). *Breusch-Pagan-Godfrey Test: Definition*. Consultado el 16 de junio de 2021, desde <https://www.statisticshowto.com/breusch-pagan-godfrey-test/>
- Glen, S. (2016b). *Cochran's Q Test*. Consultado el 9 de octubre de 2021, desde <https://www.statisticshowto.com/cochrans-q-test/>
- Glen, S. (2016c). *Holm-Bonferroni Method: Step by Step*. Consultado el 7 de mayo de 2021, desde <https://www.statisticshowto.com/holm-bonferroni-method/>
- Glen, S. (2017). *Receiver Operating Characteristic (ROC) Curve: Definition, Example*. Consultado el 23 de junio de 2021, desde <https://www.statisticshowto.com/receiver-operating-characteristic-roc-curve/>
- Glen, S. (2021a). *Coefficient of Determination (R Squared)*. Consultado el 10 de junio de 2021, desde <https://www.statisticshowto.com/probability-and-statistics/coefficient-of-determination-r-squared/>
- Glen, S. (2021b). *Geometric Mean Definition and Formula*. Consultado el 27 de mayo de 2021, desde <https://www.statisticshowto.com/geometric-mean-2/>
- Glen, S. (2021c). *Kruskal Wallis H Test: Definition, Examples & Assumptions*. Consultado el 5 de junio de 2021, desde <https://www.statisticshowto.com/kruskal-wallis/>
- Glen, S. (2021d). *Post-Hoc Definition and Types of Post Hoc Tests*. Consultado el 7 de mayo de 2021, desde <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/post-hoc/#PHscheffes>
- Goeman, J. J., & Solari, A. (2014). Multiple hypothesis testing in genomics. *Statistics in Medicine*, 33(11), 1946-1978.
- Hesterberg, T., Monaghan, S., Moore, D. S., Clipson, A., & Epstein, R. (2003). *Bootstrap Methods and Permutation Tests*. Consultado el 3 de junio de 2021, desde <https://statweb.stanford.edu/~tibs/stat315a/Supplements/bootstrap.pdf>
- Horn, R. A. (2008). *Sphericity in repeated measures analysis*. Consultado el 11 de mayo de 2021, desde <http://oak.ucc.nau.edu/rh232/courses/EPS625/Handouts/RM-ANOVA/Sphericity.pdf>
- IBM. (1989). *ANOVA de un factor: Contrastes post hoc*. Consultado el 30 de abril de 2021, desde <https://www.ibm.com/docs/es/spss-statistics/25.0.0?topic=anova-one-way-post-hoc-tests>
- Irizarry, R. A. (2019). *Introduction to Data Science*. <https://rafalab.github.io/dsbook/>.
- Joly, F. (1988). *La Cartografía*. Oikos-Tau, S.A. Ediciones.
- Kabacoff, R. I. (2017). *Power Analysis*. Consultado el 1 de octubre de 2021, desde <https://www.statmethods.net/stats/power.html>
- Kaplan, D. (2009). *Statistical Modeling: A Fresh Approach*. Consultado el 8 de marzo de 2019, desde http://works.bepress.com/daniel_kaplan/38
- Karadimitriou, S. M., & Marshall, E. (2016). *Repeated measures ANOVA in R* [statstutor community project]. Consultado el 12 de mayo de 2021, desde https://www.sheffield.ac.uk/polopoly_fs/1.885219!/file/105_RepeatedANOVA.pdf
- Kassambara, A. (2019a). *Practical Statistics in R II - Comparing Groups: Numerical Variables*. Datanovia.
- Kassambara, A. (2019b). *T-test Effect Size using Cohen's d Measure*. Consultado el 27 de abril de 2021, desde <https://www.datanovia.com/en/lessons/t-test-effect-size-using-cohens-d-measure/#cohens-d-for-paired-samples-t-test>
- Lærd Statistics. (2020a). *Friedman Test in SPSS Statistics* [Lund Research Ltd.]. Consultado el 5 de junio de 2021, desde <https://statistics.laerd.com/spss-tutorials/friedman-test-using-spss-statistics.php>
- Lærd Statistics. (2020b). *Sphericity* [Lund Research Ltd.]. Consultado el 11 de mayo de 2021, desde <https://statistics.laerd.com/statistical-guides/sphericity-statistical-guide.php>
- Lane, D. (s.f.). *Online Statistics Education: A Multimedia Course of Study*. Consultado el 4 de mayo de 2021, desde <https://onlinestatbook.com/>
- Lowry, R. (1999). *Concepts & Applications of Inferential Statistics*. Consultado el 3 de mayo de 2021, desde <http://vassarstats.net/textbook/>
- Mair, P., & Wilcox, R. (2020). Robust statistical methods in R using the WRS2 package. *Behavior Research Methods*, 52(2), 464-488.
- Mangiafico, S. S. (2016). *Cochran's Q Test for Paired Nominal Data*. Consultado el 9 de octubre de 2021, desde https://rcompanion.org/handbook/H_07.html
- McCullagh, P. (2002). What Is a Statistical Model? *The Annals of Statistics*, 30(5), 1225-1267.
- Meena, S. (2020). *Statistics for Analytics and Data Science: Hypothesis Testing and Z-Test vs. T-Test*. Consultado el 22 de septiembre de 2021, desde

- https://www.analyticsvidhya.com/blog/2020/06/statistics-analytics-hypothesis-testing-z-test-t-test/#h2_1
- Meier, L. (2021). *ANOVA: A Short Intro Using R*. Consultado el 7 de mayo de 2021, desde <https://stat.ethz.ch/~meier/teaching/anova/>
- Mendez Ramírez, I. (1998). Empirismo, método científico y estadística. *Revista de Geografía Agrícola (Mexico)*.
- Mendez Ramírez, I. (2012). Método Científico: aspectos epistemológicos y metodológicos para el uso de la Estadística. *SaberEs*, 4.
- Montero Muñoz, J., Solla Suárez, P. E., & Gutiérrez Rodríguez, J. (2012). Estimación del filtrado glomerular en el paciente anciano. Implicaciones clínicas en el uso de antibióticos. *Revista Española de Geriatría y Gerontología*, 56(5), 268-271.
- Müller, K. (2021). *dplyr*. Consultado el 10 de septiembre de 2021, desde <https://dplyr.tidyverse.org/>
- NIST/SEMATECH. (2013). *e-Handbook of Statistical Methods*. Consultado el 29 de abril de 2021, desde <http://www.itl.nist.gov/div898/handbook/>
- Parada, L. F. (2019). *Prueba de normalidad de Shapiro-Wilk*. Consultado el 22 de septiembre de 2021, desde <https://rpubs.com/F3rnando/507482>
- Pardoe, I., Simon, L., & Young, D. (2018). *Residuals vs. Fits Plot*. Consultado el 21 de diciembre de 2021, desde <https://online.stat.psu.edu/stat462/node/117/>
- Pértiga, S., & Pita, S. (2004). *Asociación de variables cualitativas: El test exacto de Fisher y el test de Mcnemar*. Consultado el 29 de abril de 2021, desde <https://www.fisterra.com/mbe/investiga/fisher/fisher.asp#Mcnemar>
- Real Academia Española. (2014). *Diccionario de la lengua española* (23.^a ed.). Consultado el 30 de marzo de 2021, desde <https://dle.rae.es>
- Real Statistics Using Excel. (s.f.). *Mann-Whitney Table*. Consultado el 28 de mayo de 2021, desde <https://www.real-statistics.com/statistics-tables/mann-whitney-table/>
- Ríos, S. (1995). *Modelización*. Alianza Ediciones.
- Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. John wiley & sons.
- RStudio. (2021). RStudio. Consultado el 2 de abril de 2021, desde <https://rstudio.com/products/rstudio/>
- SAS Institute Inc. (2008). SAS/STAT® 9.2 User's Guide.
- STDHA. (s.f.). Descriptive Statistics and Graphics - Easy Guides - Wiki - STHDA. Consultado el 31 de marzo de 2021, desde <http://www.sthda.com/english/wiki/descriptive-statistics-and-graphics>
- The R Foundation. (s.f.). Documentation. Consultado el 2 de abril de 2021, desde <https://www.r-project.org/other-docs.html>
- United States Census Bureau. (2004). *CT1970p2-13: Colonial and Pre-Federal Statistics*. Consultado el 26 de mayo de 2021, desde <https://www2.census.gov/prod2/statcomp/documents/CT1970p2-13.pdf>
- United States Census Bureau. (2021). Decennial Census of Population and Housing. Consultado el 26 de mayo de 2021, desde <https://www.census.gov/programs-surveys/decennial-census/decade.html>
- Wallis, S. (2013). Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods. *Journal of Quantitative Linguistics*, 20(3), 178-208.
- Wickham, H. (2021). *tidyR*. Consultado el 10 de septiembre de 2021, desde <https://r4ds.had.co.nz/index.html>
- Wickham, H., & Grolemund, G. (2017). *R for Data Science*. <https://r4ds.had.co.nz/index.html>
- Willem, K. (2017). *Formulas in R Tutorial*. Consultado el 11 de septiembre de 2021, desde <https://dplyr.tidyverse.org/>
- Winner, L. (2021). *Simple Linear Regression I — Least Squares Estimation*. Consultado el 8 de junio de 2021, desde <http://users.stat.ufl.edu/~winner/qmb3250/notespart2.pdf>
- Zelada, C. (2017). *Evaluación de modelos de clasificación*. Consultado el 23 de junio de 2021, desde <https://rpubs.com/chzelada/275494>
- Zimmerman, D. W., & Zumbo, B. D. (1993). Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. *The Journal of Experimental Education*, 62(1), 75-86.