

**INTRODUCCIÓN A LA CIENCIA DE DATOS (2019)**  
MÁSTER OFICIAL UNIVERSITARIO EN CIENCIA DE DATOS E INGENIERÍA DE  
COMPUTADORES  
UNIVERSIDAD DE GRANADA

---

## Trabajo Integrador: EDA, Clasificación y Regresión

---

Luis Balderas Ruiz  
[luisbalderas@correo.ugr.es](mailto:luisbalderas@correo.ugr.es)

22 de diciembre de 2019

# Índice

<b>1 Introducción</b>	<b>8</b>
<b>2 Análisis exploratorio de datos (EDA)</b>	<b>8</b>
2.1 Conjunto de datos Wankara . . . . .	8
2.1.1 Max-temperature . . . . .	9
2.1.2 Min-temperature . . . . .	11
2.1.3 Dewpoint . . . . .	13
2.1.4 Precipitation . . . . .	15
2.1.5 Sea-level-pressure . . . . .	16
2.1.6 Standard pressure . . . . .	18
2.1.7 Visibility . . . . .	20
2.1.8 Wind speed . . . . .	22
2.1.9 Max wind speed . . . . .	23
2.1.10 Mean temperatura . . . . .	25
2.1.11 Algunas hipótesis previas a estudiar la correlación . . . . .	26
2.1.12 División por meses . . . . .	28
2.1.13 Correlación de las variables . . . . .	30
2.1.14 Conclusiones . . . . .	31
2.2 Conjunto de datos Vowel . . . . .	32
2.2.1 F0 . . . . .	34
2.2.2 F1 . . . . .	36
2.2.3 F2 . . . . .	38
2.2.4 F3 . . . . .	40
2.2.5 F4 . . . . .	42
2.2.6 F5 . . . . .	45
2.2.7 F6 . . . . .	48
2.2.8 F7 . . . . .	52
2.2.9 F8 . . . . .	54
2.2.10 F9 . . . . .	58
2.2.11 Correlación entre variables . . . . .	60
2.2.12 Conclusiones . . . . .	65
<b>3 Problema de regresión: Wankara</b>	<b>73</b>
3.1 Regresión lineal simple sobre 5 regresores . . . . .	73
3.1.1 Modelo con Max-temperature . . . . .	73
3.1.2 Modelo lineal con Min-temperature . . . . .	74
3.1.3 Modelo lineal con Dewpoint . . . . .	76
3.1.4 Modelo lineal con Sea level pressure . . . . .	77
3.1.5 Modelo lineal con Visibility . . . . .	78
3.1.6 Conclusiones . . . . .	79
3.2 Modelos lineales múltiples. Interacciones y no linealidad . . . . .	79
3.2.1 Hacia un modelo competitivo y explicable . . . . .	79

3.2.2	Modelo lineal múltiple óptimo: interacciones y no linealidad . . . . .	84
3.3	kNN en regresión . . . . .	86
3.4	Comparación con algoritmos . . . . .	88
3.4.1	Comparativa sobre los distintos datasets . . . . .	88
3.4.2	Comparativa sobre los folds de Wankara . . . . .	92
<b>4</b>	<b>Problema de clasificación: Vowel</b>	<b>95</b>
4.1	KNN . . . . .	95
4.2	LDA . . . . .	98
4.3	QDA . . . . .	101
4.4	Comparación de algoritmos en test . . . . .	104
4.4.1	1NN vs LDA . . . . .	105
4.4.2	1NN vs QDA . . . . .	105
4.4.3	3NN vs LDA . . . . .	105
4.4.4	3NN vs QDA . . . . .	105
4.4.5	LDA vs QDA . . . . .	105
4.4.6	Comparativa general con el test de Friedman . . . . .	106
4.5	Comparación de algoritmos en entrenamiento . . . . .	106
4.5.1	1NN vs LDA . . . . .	107
4.5.2	1NN vs QDA . . . . .	107
4.5.3	3NN vs LDA . . . . .	107
4.5.4	3NN vs QDA . . . . .	107
4.5.5	LDA vs QDA . . . . .	107
4.5.6	Comparativa general con Friedman . . . . .	108
<b>5</b>	<b>Bibliografía</b>	<b>109</b>
<b>Apéndice 1: Código Vowel</b>		<b>109</b>
<b>Apéndice 2: Código Wankara</b>		<b>109</b>

## Índice de figuras

2.1.	Primer vistazo a las variables . . . . .	9
2.2.	Histograma de Max-temperature . . . . .	10
2.3.	Tests de normalidad sobre Max-temperature . . . . .	10
2.4.	QQPlot de Max-temperature . . . . .	11
2.5.	Histograma de Min-temperature . . . . .	12
2.6.	Tests de normalidad sobre Min-temperature . . . . .	12
2.7.	QQPlot de Min-temperature . . . . .	13
2.8.	Histograma de Dewpoint . . . . .	14
2.9.	Tests de normalidad sobre Dewpoint . . . . .	14
2.10.	QQPlot de Dewpoint . . . . .	15
2.11.	Histograma de Precipitation . . . . .	16

2.12. Tests de normalidad sobre Precipitation . . . . .	16
2.13. Histograma de Sea-level-pressure . . . . .	17
2.14. Tests de normalidad sobre Sea-level-pressure . . . . .	17
2.15. QQPlot de Sea level pressure . . . . .	18
2.16. Histograma de Standard pressure . . . . .	19
2.17. Tests de normalidad sobre Standard pressure . . . . .	19
2.18. QQPlot de Standard Pressure . . . . .	20
2.19. Histograma de Visibility . . . . .	21
2.20. Tests de normalidad sobre Visibility . . . . .	21
2.21. QQPlot de Visibility . . . . .	22
2.22. Histograma de Wind speed . . . . .	23
2.23. Tests de normalidad sobre Wind speed . . . . .	23
2.24. Histograma de Max Wind speed . . . . .	24
2.25. Tests de normalidad sobre Max Wind speed . . . . .	24
2.26. QQPlot de Max wind speed . . . . .	25
2.27. Histograma de Mean temperature . . . . .	26
2.28. Tests de normalidad sobre Mean temperature . . . . .	26
2.29. Temperatura máxima vs media . . . . .	27
2.30. Temperatura mínima vs media . . . . .	27
2.31. Temperatura máxima media de cada mes . . . . .	28
2.32. Temperatura mínima media de cada mes . . . . .	29
2.33. Resumen de la correlación . . . . .	30
2.34. Distribución por sexos de los interlocutores . . . . .	32
2.35. Histograma de la variable F0 . . . . .	34
2.36. Tests de normalidad sobre F0 . . . . .	34
2.37. Boxplot para F0 estudiando sexos e interlocutores . . . . .	35
2.38. Tests de normalidad sobre F0 (hombres) . . . . .	35
2.39. Tests de normalidad sobre F0 (mujeres) . . . . .	35
2.40. Histograma de la variable F1 . . . . .	36
2.41. Tests de normalidad sobre F1 . . . . .	36
2.42. Histograma con la densidad de F1 y normal para comparar . . . . .	37
2.43. QQPlot de la variable F1 . . . . .	37
2.44. Boxplot para F1 estudiando sexos e interlocutores . . . . .	38
2.45. Tests de normalidad sobre F1 (hombres) . . . . .	38
2.46. Tests de normalidad sobre F1 (mujeres) . . . . .	38
2.47. Histograma de la variable F2 . . . . .	39
2.48. Tests de normalidad sobre F2 . . . . .	39
2.49. Boxplot para F2 estudiando sexos e interlocutores . . . . .	40
2.50. Tests de normalidad sobre F2 (hombres) . . . . .	40
2.51. Tests de normalidad sobre F2 (mujeres) . . . . .	40
2.52. Histograma de la variable F3 . . . . .	41
2.53. Tests de normalidad sobre F3 . . . . .	41
2.54. Boxplot para F3 estudiando sexos e interlocutores . . . . .	42
2.55. Tests de normalidad sobre F3 (hombres) . . . . .	42

2.56. Tests de normalidad sobre F3 (mujeres)	42
2.57. Histograma de la variable F4	43
2.58. Tests de normalidad sobre F4	43
2.59. Gráfico QQPlot de F4	44
2.60. Boxplot para F4 estudiando sexos e interlocutores	44
2.61. Tests de normalidad sobre F4 (hombres)	44
2.62. Tests de normalidad sobre F4 (mujeres)	45
2.63. Gráfico QQPlot de F4 (mujeres)	45
2.64. Histograma de la variable F5	46
2.65. Tests de normalidad sobre F5	46
2.66. Boxplot para F5 estudiando sexos e interlocutores	47
2.67. Tests de normalidad sobre F5 (hombres)	47
2.68. Tests de normalidad sobre F5 (mujeres)	47
2.69. QQPlot de la variable F5 (mujeres)	48
2.70. Histograma de la variable F6	49
2.71. Tests de normalidad sobre F6	49
2.72. QQPlot de la variable F6	50
2.73. Boxplot para F6 estudiando sexos e interlocutores	50
2.74. Tests de normalidad sobre F6 (hombres)	51
2.75. QQPlot de la variable F6 (hombres)	51
2.76. Tests de normalidad sobre F6 (mujeres)	51
2.77. QQPlot de la variable F6 (mujeres)	52
2.78. Histograma de la variable F7	53
2.79. Tests de normalidad sobre F7	53
2.80. Boxplot para F7 estudiando sexos e interlocutores	54
2.81. Tests de normalidad sobre F7 (hombres)	54
2.82. Tests de normalidad sobre F7 (mujeres)	54
2.83. Histograma de la variable F8	55
2.84. Tests de normalidad sobre F8	55
2.85. QQPlot de la variable F8	56
2.86. Boxplot para F8 estudiando sexos e interlocutores	56
2.87. Tests de normalidad sobre F8 (hombres)	57
2.88. Tests de normalidad sobre F8 (mujeres)	57
2.89. QQPlot de la variable F8 (mujeres)	57
2.90. Histograma de la variable F9	58
2.91. Tests de normalidad sobre F9	59
2.92. Boxplot para F9 estudiando sexos e interlocutores	59
2.93. Tests de normalidad sobre F9 (hombres)	59
2.94. Tests de normalidad sobre F9 (mujeres)	60
2.95. Comparativa de las variables dos a dos	60
2.96. Comparativa de las variables dos a dos	61
2.97. Comparativa de las variables dos a dos (hombres)	62
2.98. Comparativa de las variables dos a dos (hombres)	63
2.99. Comparativa de las variables dos a dos (mujeres)	64

2.100Comparativa de las variables dos a dos (mujeres) . . . . .	65
2.101Boxplot para F0 y F1 . . . . .	66
2.102Boxplot para F2 y F3 . . . . .	67
2.103Boxplot para F4 y F5 . . . . .	67
2.104Boxplot para F6 y F7 . . . . .	68
2.105Boxplot para F8 y F9 . . . . .	69
2.106Boxplot para F0 y F1 . . . . .	69
2.107Boxplot para F2 y F3 . . . . .	70
2.108Boxplot para F4 y F5 . . . . .	71
2.109Boxplot para F6 y F7 . . . . .	71
2.110Boxplot para F8 y F9 . . . . .	72
3.1. Modelo lineal con Max-temperature . . . . .	73
3.2. Modelo lineal con Max-temperature. Scatter plot . . . . .	74
3.3. Modelo lineal con Min-temperature . . . . .	75
3.4. Modelo lineal con Min-temperature. Scatter plot . . . . .	75
3.5. Modelo lineal con Dewpoint . . . . .	76
3.6. Modelo lineal con Dewpoint. Scatter plot . . . . .	76
3.7. Modelo lineal con Sea level pressure . . . . .	77
3.8. Modelo lineal con Sea level pressure. Scatter plot . . . . .	77
3.9. Modelo lineal con Visibility . . . . .	78
3.10. Modelo lineal con Visibility. Scatter plot . . . . .	78
3.11. Modelo lineal múltiple con todas las variables . . . . .	80
3.12. Modelo lineal múltiple sin Precipitation . . . . .	80
3.13. Modelo lineal múltiple sin Precipitation y Sea level pressure . . . . .	81
3.14. Modelo lineal múltiple sin Precipitation, Sea level pressure, Max wind speed . . . . .	82
3.15. Modelo lineal múltiple sin Precipitation, Sea level pressure, Max wind speed, Visibility . . . . .	82
3.16. Modelo lineal múltiple sin Precipitation, Sea level pressure, Max wind speed, Visibility, Standard Pressure . . . . .	83
3.17. Modelo lineal múltiple más interpretable: Max temperature, Min temperature y Dewpoint . . . . .	83
3.18. Modelo lineal múltiple óptimo . . . . .	85
3.19. Plot del modelo óptimo encontrado sobre la gráfica Max-temperature-Mean-temperature . . . . .	85
3.20. Primera aproximación: knn con todas las variables . . . . .	86
3.21. Segunda aproximación: knn con la combinación óptima anterior. Modelo . . . . .	87
3.22. Primera aproximación: knn con la combinación óptima anterior . . . . .	87
3.23. Primera aproximación: knn con la combinación más interpretable . . . . .	88
3.24. Tabla de MSE para cada algoritmo sobre los conjuntos . . . . .	89
3.25. Test de Wilcoxon: LM vs KNN . . . . .	89
3.26. Test de Wilcoxon: LM vs M5P . . . . .	89
3.27. Test de Wilcoxon: kNN vs M5P . . . . .	90
3.28. Test de Friedman . . . . .	90
3.29. Post-hoc holm . . . . .	90

3.30. Test de Wilcoxon: LM vs KNN	91
3.31. Test de Wilcoxon: LM vs M5P	91
3.32. Test de Wilcoxon: kNN vs M5P	91
3.33. Test de Friedman	92
3.34. Post-hoc holm	92
3.35. Tabla resultados en test para los folds	92
3.36. Test de Wilcoxon: LM vs KNN	93
3.37. Test de Wilcoxon: LM vs RF	93
3.38. Test de Wilcoxon: kNN vs RF	93
3.39. Test de Friedman	94
3.40. Post-hoc holm	94
4.1. Accuracy para los distintos valores de k	95
4.2. Plot F1-F2 para mostrar las fronteras de decisión	96
4.3. Plot F2-F6 para mostrar las fronteras de decisión	96
4.4. Valores de k vs Accuracy para hombres y mujeres	97
4.5. Evolución de accuracy respecto de k en training	97
4.6. Evolución de accuracy respecto de k en test	98
4.7. Test de normalidad para cada variable	98
4.8. Varianza de las variables	99
4.9. Matriz de confusión para LDA	99
4.10. Estadísticas generales para LDA	99
4.11. Partimat sobre F0-F3 para el modelo de LDA en Vowel	100
4.12. Matriz de confusión para hombres LDA	100
4.13. Estadísticas generales para hombres LDA	101
4.14. Estadísticas generales para mujeres LDA	101
4.15. Varianza de los regresores entre las clases 0 y 4	102
4.16. Varianza de los regresores entre las clases 4 y 8	102
4.17. Varianza de los regresores entre las clases 9 y 10	102
4.18. Matriz de confusión para QDA	103
4.19. Estadísticas generales para QDA	103
4.20. Estadísticas generales para QDA por clase	103
4.21. Partimat sobre F0-F3 para el modelo de QDA en Vowel	104
4.22. Tabla de resultados para cada algoritmo y partición	104
4.23. Resultados del test de Friedman para clasificación	106
4.24. Post Hoc Holm en clasificación	106
4.25. Tabla de resultados para training	107
4.26. Resultados del test de Friedman en training	108
4.27. Post Hoc Holm en training clasificación	108

## Índice de cuadros

2.1. Resumen estadístico y desviación típica de las características reales	33
3.1. Resumen de resultados para modelos lineales simples	79

## 1. Introducción

El presente documento contiene los resultados obtenidos en el Trabajo Teórico/Práctico Integrador para la evaluación de la asignatura Introducción a la Ciencia de Datos. El trabajo está formado por tres apartados, a saber, análisis de datos (en adelante EDA), regresión y clasificación, centrado en dos conjuntos de datos: Wankara para regresión y Vowel para clasificación. Ambos dos forman parte del repositorio de Keel ([\[4\]](#)), el primero en [\[3\]](#) y el segundo en [\[2\]](#). A continuación, describo la estructura del documento. En la primera sección, se desarrolla el análisis exploratorio de ambos datasets. A continuación, tras sacar las conclusiones correspondientes, ataco el problema de regresión y, por último, el de clasificación.

## 2. Análisis exploratorio de datos (EDA)

### 2.1. Conjunto de datos Wankara

El presente conjunto de datos contiene información sobre el tiempo atmosférico entre los días 01/01/1994 y 28/05/1998 en la ciudad de Ankara, Turquía. En total, 1609 días (tantos como instancias) con información expresada en 10 variables numéricas que son las siguientes:

- Max-temperature: Temperatura máxima del día (°F).
- Min-temperature: Temperatura mínima del día (°F).
- Dewpoint: (Punto de rocío) Es la más alta temperatura a la que empieza a condensarse el vapor de agua contenido en el aire, produciendo rocío, neblina, cualquier tipo de nube o, en caso de que la temperatura sea lo suficientemente baja, escarcha (°F).
- Precipitation: Cantidad de precipitaciones en el día(mm).
- Sea-level-pressure: Presión a nivel del mar (suponemos que en P).
- Standard-pressure: Presión estándar (suponemos que en P).
- Visibility: Medida de la mayor distancia en la cual un objeto puede verse de forma nítida (suponemos millas).
- Wind-speed: Velocidad del viento (suponemos mph).
- Max-wind-speed: Velocidad máxima del viento (suponemos mph).
- Mean-temperature (variable a modelar): temperatura media del día (°F).

Tras entender el significado de cada variable y antes de empezar a estudiarlas un poco una a una, conviene tener una idea general de cuál es la tendencia entre las variables. Para ello, presento el siguiente gráfico:

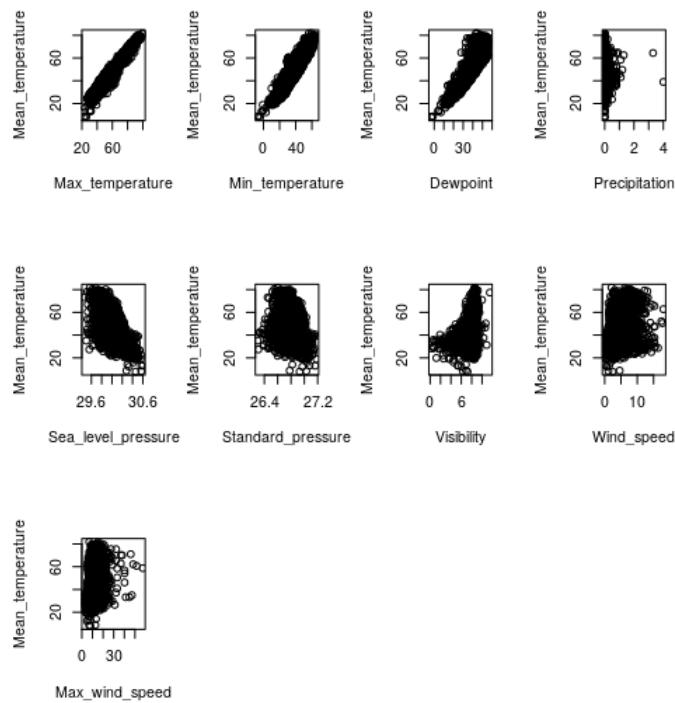


Figura 2.1: Primer vistazo a las variables

En cada plot enfrento la variable que queremos modelar, Mean-temperature, y las demás. Como se puede ver, hay tendencias muy claras entre Max-temperature, Min-temperature o Dewpoint y Mean-temperature. Las estudiaremos más a fondo a continuación para poder elegir correctamente los regresores. Hay que señalar que no existen valores perdidos en el conjunto de datos. Pasamos a estudiar cada variable de forma individual.

### 2.1.1. Max-temperature

La variable Max-temperature mide, en grados Fahrenheit, la temperatura máxima de cada día en Ankara. Presento primero un resumen estadístico de la misma:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
23.00	46.40	60.80	77.40	17.8727	77.40	100.00

Además, los valores de asimetría es 0,0197847 y curtosis, -1,138017, por lo que hay una muy ligera asimetría hacia la derecha y la distribución es platicúrtica. Veamos su histograma

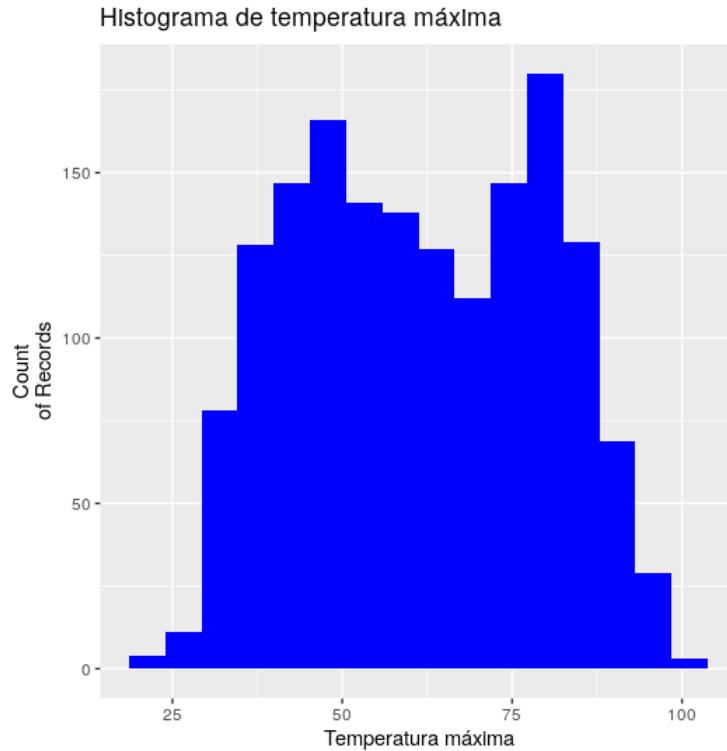


Figura 2.2: Histograma de Max-temperature

en el que vemos que los valores se concentran mayoritariamente en 40°F y 80°F aproximadamente. Este hecho se justifica con la estacionalidad, ya que los veranos de Ankara son calurosos y rondan los 30°C (80°F) (II). Respecto de outliers, la herramienta del paquete *outliers* nos dice que encuentra el valor 23.0 como outlier por la izquierda y 100.0 por la derecha, mínimo y máximo de la muestra respectivamente, y se puede ver en el histograma que apenas están representados. En efecto, 100°F es un valor muy alto para Ankara (dificilmente se pasa de los 35°C (II)) y 23°F roza sobrepuja los mínimos habituales.

Por último, comprobamos con los test de normalidad Shapiro-Wilk (8) y la corrección de Lilliefors del test de Kolmogorov-Smirnov (6), si la distribución de la variables es normal:

<pre> Max_temperature statistic 0.9658556 p.value 4.820105e-19 method "Shapiro-Wilk normality test" </pre>	<pre> Max_temperature statistic 0.07345801 p.value 1.170749e-22 method "Lilliefors (Kolmogorov-Smirnov) normality test" </pre>
(a) Shapiro-Wilk	(b) Lilliefors

Figura 2.3: Tests de normalidad sobre Max-temperature

cuyos p-valores son menores que 0.05 y, por tanto, rechazamos la hipótesis de normalidad.

Mostramos el gráfico QQ:

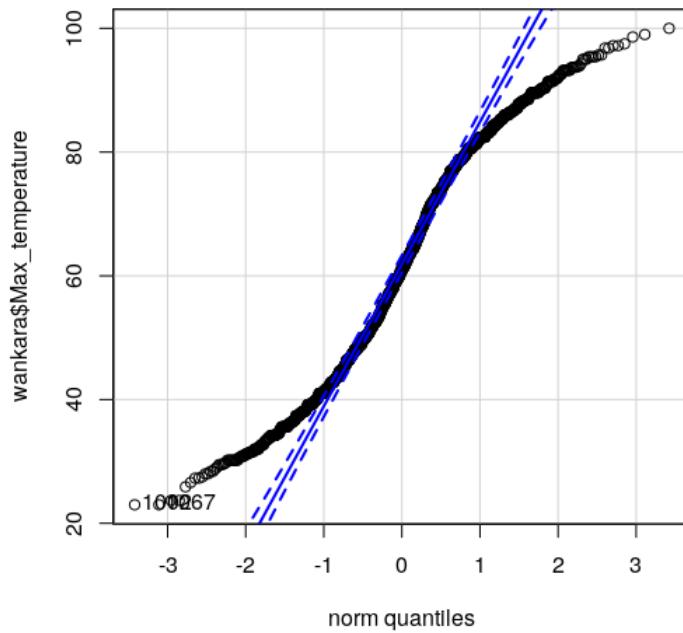


Figura 2.4: QQPlot de Max-temperature

donde claramente se ve que, efectivamente, la variable no se distribuye según una normal.

### 2.1.2. Min-temperature

La variable Min-temperature mide, en grados Fahrenheit, la temperatura mínima de cada día en Ankara. Presento primero un resumen estadístico de la misma:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
-7.1	26.60	36.00	37.08	13.34527	48.20	65.5

con valores de asimetría -0.055429 (ligeramente hacia la izquierda) y curtosis -0.7089539 (platicúrtica). Veamos su histograma.

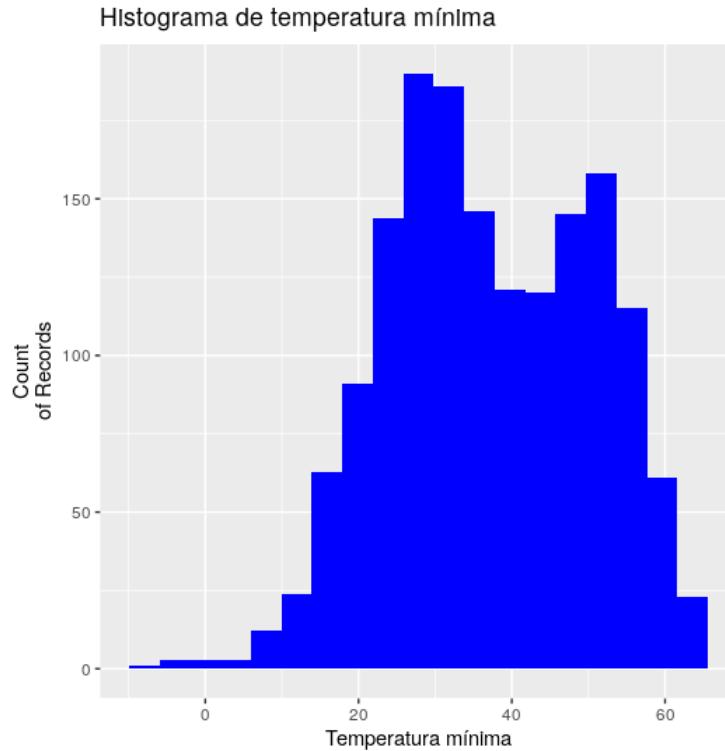


Figura 2.5: Histograma de Min-temperature

con una media de  $37.08^{\circ}\text{F}$  aunque también vemos un repunte cerca de los  $50^{\circ}\text{F}$ , lo que nos muestra temperaturas suaves en gran parte de los días. Respecto de outliers, encontramos valores de  $-7.1^{\circ}\text{F}$  ( $-22^{\circ}\text{C}$ ), claramente irregulares en la tendencia general de la temperatura mínima, por lo que pudo darse una ola de frío sin precedentes o errores en la medición.

Si aplicamos los test de normalidad,

```
Min_temperature
0.9804859
5.531039e-14
"Shapiro-Wilk normality test"
-----
```

(a) Shapiro-Wilk

```
Min_temperature
0.07262888
4.007458e-22
"Lilliefors (Kolmogorov-Smirnov) normality test"
-----
(b) Lilliefors
```

Figura 2.6: Tests de normalidad sobre Min-temperature

vemos que rechazamos la hipótesis de normalidad. En efecto, a través del QQPlot comprobamos que así es.

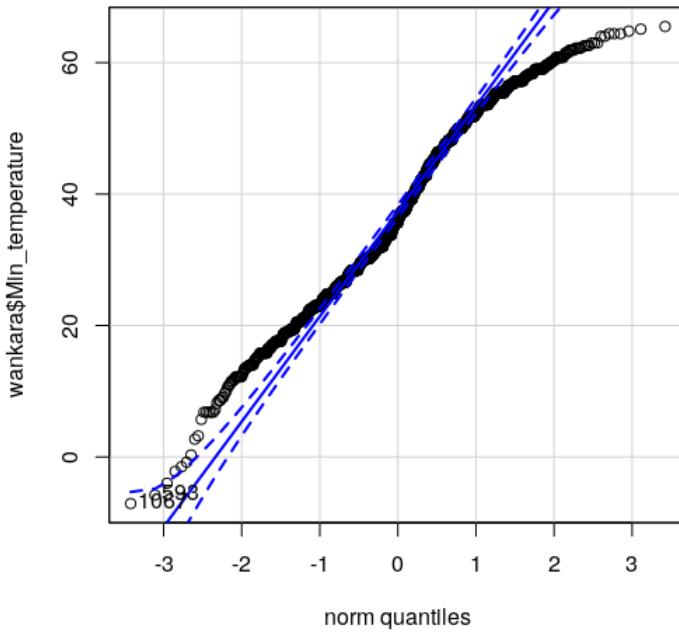


Figura 2.7: QQPlot de Min-temperature

### 2.1.3. Dewpoint

La variable Dewpoint se almacena la más alta temperatura a la que empieza a condensarse el vapor de agua contenido en el aire, produciendo rocío, neblina, cualquier tipo de nube o, en caso de que la temperatura sea lo suficientemente baja, escarcha ( $^{\circ}\text{F}$ ). Veamos un resumen estadístico:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
-3.1	28.50	36.80	36.29	10.81945	45.30	57.60

con valores de asimetría  $-0,3740575$  (por la izquierda) y curtosis  $-0,4824467$  (platicúrtica). Veamos su histograma.

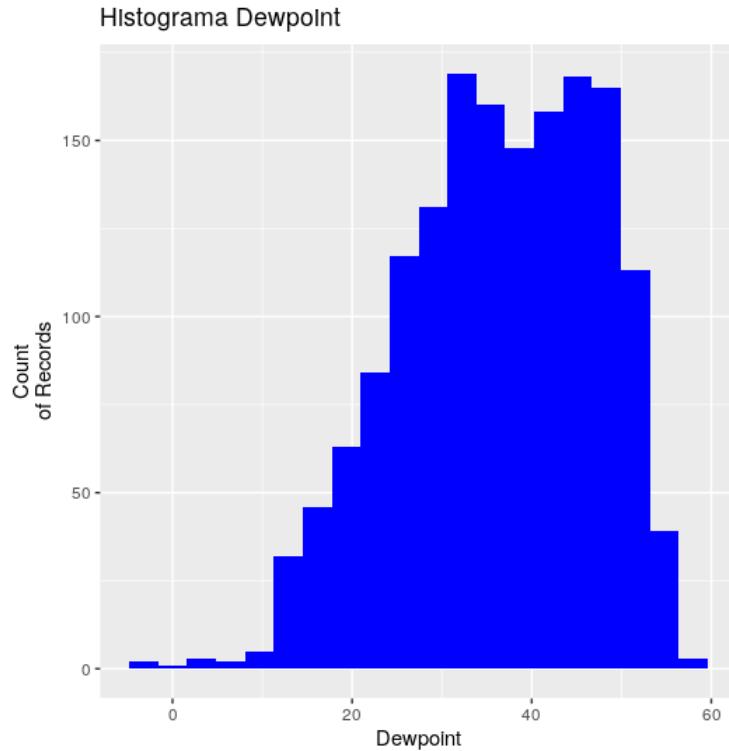


Figura 2.8: Histograma de Dewpoint

donde confirmamos la asimetría por la derecha y vemos como la media y la mediana están bastante cercanas. Respecto de outliers, la herramienta nos informa de los valores  $-3,1$  y  $57,60$ , como posibles candidatos, dada su escasez de apariciones. Podremos contrastarlo más detenidamente después. Si aplicamos los tests de normalidad

```
Dewpoint
0.9772369
2.723394e-15
"Shapiro-Wilk normality test"
```

(a) Shapiro-Wilk

```
Dewpoint
0.0576301
1.377454e-13
"Lilliefors (Kolmogorov-Smirnov) normality test"
```

(b) Lilliefors

Figura 2.9: Tests de normalidad sobre Dewpoint

vemos que rechazamos la hipótesis de normalidad. En efecto, a través del QQPlot comprobamos que así es.

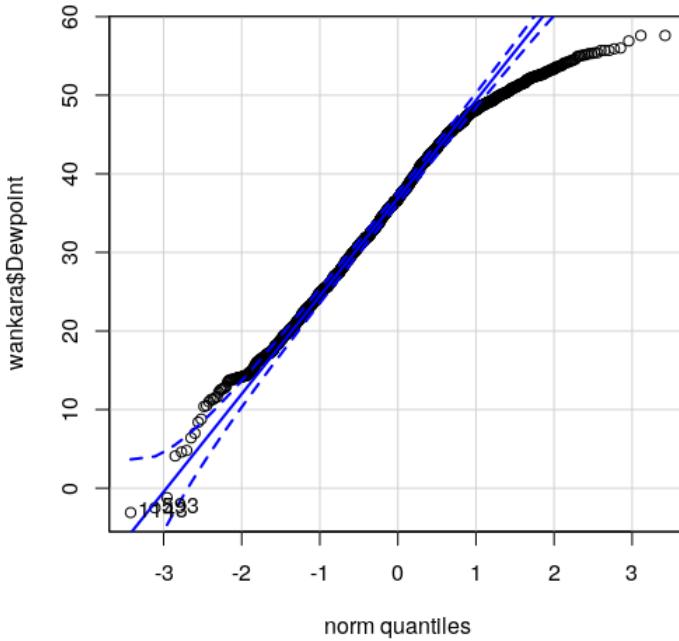


Figura 2.10: QQPlot de Dewpoint

#### 2.1.4. Precipitation

La variable Precipitation refleja la cantidad de precipitaciones (mm) en un día. Veamos un resumen estadístico de su contenido.

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
0	0	0	0.0541	0.1844009	0	4

Podemos ver el escaso nivel de precipitaciones. Respecto de la asimetría (11.07923), es asimétrica por la derecha y presenta una curtosis de 194.1193 (leptocúrtica). Veamos su histograma:

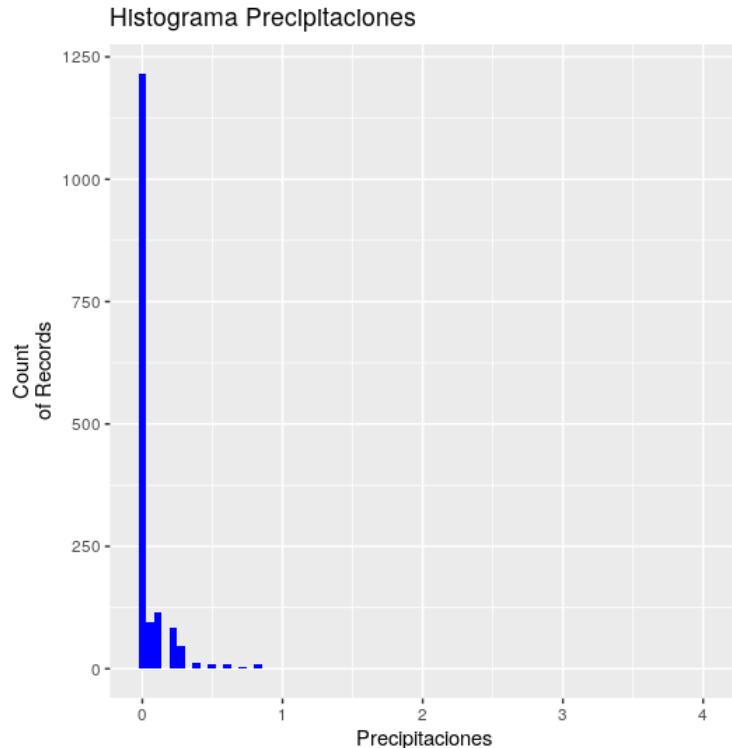


Figura 2.11: Histograma de Precipitation

en el que confirmamos el nivel de sequía (casi 1250 días sin llover). Encontramos un par de días con un nivel de lluvia de 4mm y 3.2mm, lo que pueden ser errores en la medición días extraordinariamente lluviosos para el histórico de la ciudad. Aplicando los tests de normalidad

```
Precipitation
0.2960197
2.780406e-61
"Shapiro-Wilk normality test"
```

(a) Shapiro-Wilk

```
Precipitation
0.3846109
0
"Lilliefors (Kolmogorov-Smirnov) normality test"
```

(b) Lilliefors

Figura 2.12: Tests de normalidad sobre Precipitation

vemos que por sus p-valores debemos rechazar la hipótesis de normalidad.

#### 2.1.5. Sea-level-pressure

En esta variable se aloja la presión a nivel del mar diaria en Ankara. A continuación, presento un resumen estadístico de la misma:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
29.46	29.83	29.96	29.98	0.2015043	30.11	30.60

Cabe destacar del resumen estadístico la cercanía de la media y la mediana y la muy reducida variabilidad. Además, presenta una asimetría por la derecha (0.4085841) y es platicúrtica (-0.1849383 de curtosis), aunque cerca de ser mesocúrtica. Veamos el histograma:

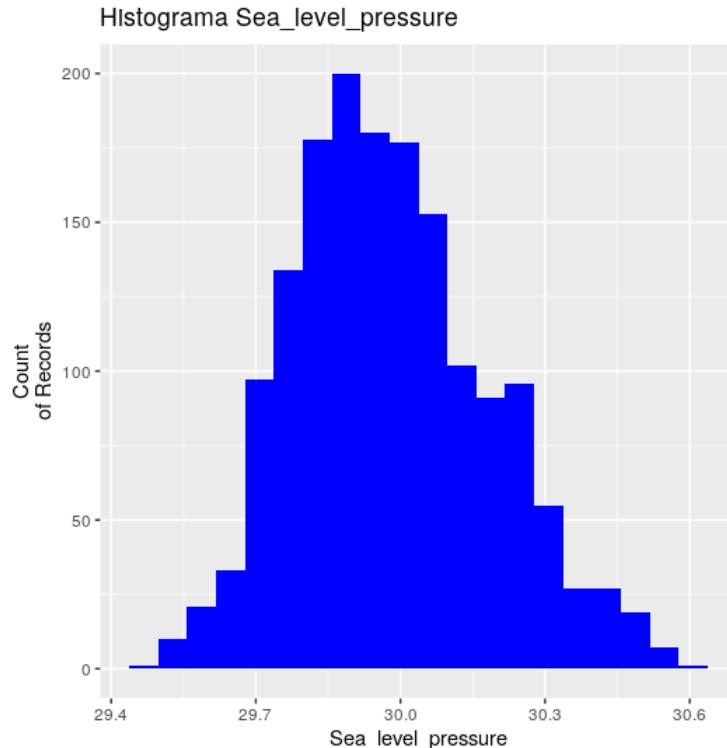


Figura 2.13: Histograma de Sea-level-pressure

Vemos que su forma se asemeja a la de una distribución normal. Sin embargo aplicando los tests de normalidad,

```
Sea_level_pressure
0.9853185
9.759597e-12
"Shapiro-Wilk normality test"
-----
```

(a) Shapiro-Wilk

```
Sea_level_pressure
0.0583415
6.062102e-14
"Lilliefors (Kolmogorov-Smirnov) normality test"
```

(b) Lilliefors

Figura 2.14: Tests de normalidad sobre Sea-level-pressure

vemos que se rechaza la hipótesis de normalidad en ambos tests. Podemos asegurarnos con el QQPlot de que, a pesar de que se parece a una normal, se sale de los rangos:

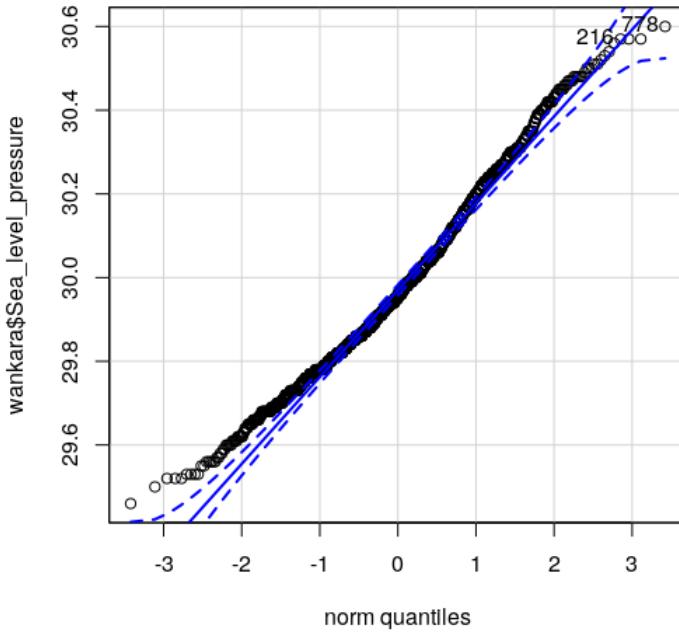


Figura 2.15: QQPlot de Sea level pressure

y por lo tanto, no sigue una distribución normal.

#### 2.1.6. Standard pressure

Standard pressure representa la presión atmosférica diaria en la ciudad de Ankara. Veamos un resumen estadístico de la misma:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
26.30	26.69	26.77	26.78	0.1383007	26.87	27.18

Vemos de nuevo una distribución de valores muy concentrada alrededor del 26.78, con media y medianas muy cercanas y apenas variabilidad. Por tanto, no se darán outliers. Además, no presenta asimetría (un escaso  $-0,00832628$ ) y es leptocúrtica ( $0,1495295$ ). Veamos su histograma.

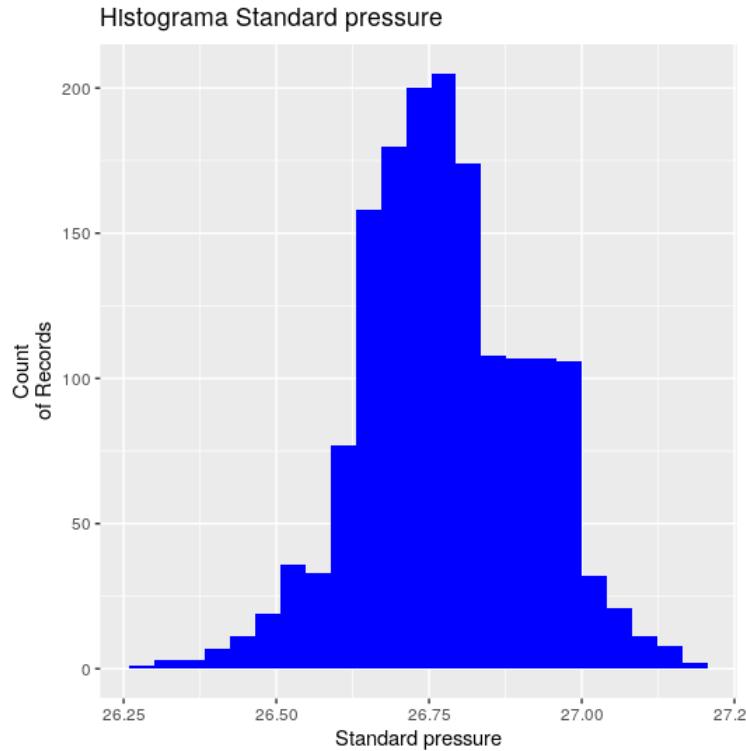


Figura 2.16: Histograma de Standard pressure

De nuevo vemos una distribución de valores que se parece a la normal. Comprobamos con los tests de normalidad

```
Standard_pressure
0.9951595
4.61698e-05
"Shapiro-Wilk normality test"
```

(a) Shapiro-Wilk

```
Standard_pressure
0.04636253
1.385896e-08
"Lilliefors (Kolmogorov-Smirnov) normality test"
```

(b) Lilliefors

Figura 2.17: Tests de normalidad sobre Standard pressure

que debemos rechazar la hipótesis de normalidad. El gráfico QQ nos lo pone algo difícil en la confirmación:

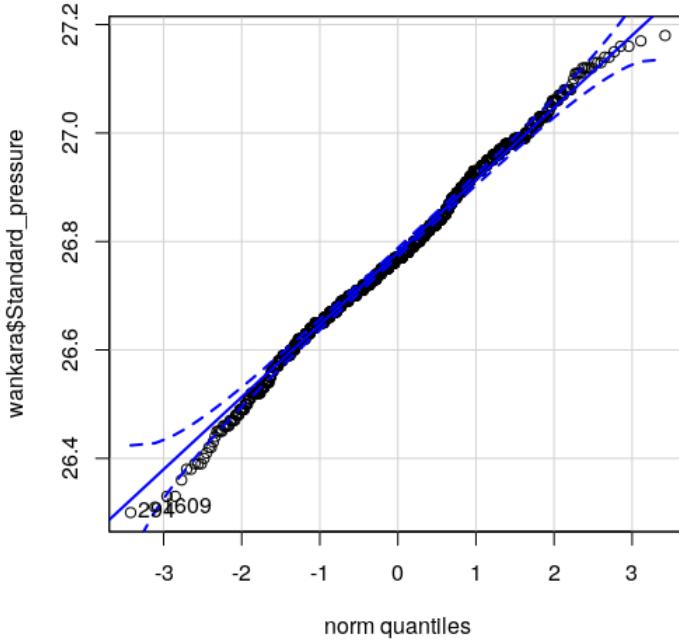


Figura 2.18: QQPlot de Standard Pressure

ya que los puntos siempre lindan con los rangos permitidos. No obstante, prevalece para nosotros el diagnóstico de los tests de normalidad.

### 2.1.7. Visibility

La variable **Visibility** contiene los datos acerca de la mayor distancia en la cual un objeto puede verse de forma nítida. Mostramos algunas estadísticas para conocerla mejor:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
0.2	7.4	8.3	7.718	1.47925	8.6	11.5

Encontramos gran distancia entre el valor mínimo y el primer cuartil, y teniendo en cuenta que la desviación estandar es 1.47925, podríamos estar ante un outlier. Además, la variable presenta asimetría hacia la izquierda (-1.947852) y es leptocúrtica (4.207581). Observamos el histograma de frecuencias

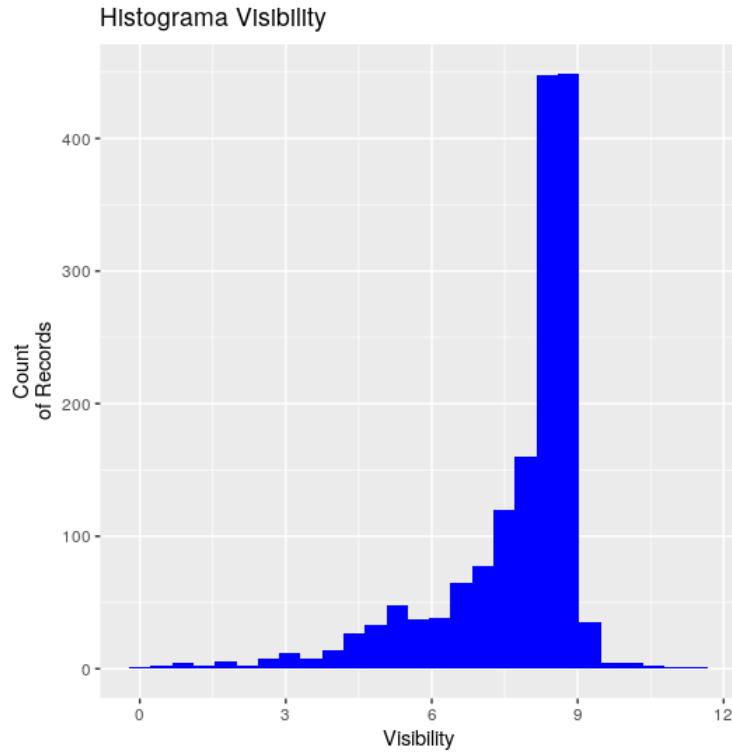


Figura 2.19: Histograma de Visibility

En él se observa bastante bien la asimetría y que los valores entorno a 8.3 son los más repetidos. Si estudiamos los tests de normalidad para Visibility

```
Visibility
0.7748121
2.033691e-42
"Shapiro-Wilk normality test"
(a) Shapiro-Wilk
```

```
Visibility
0.2202272
1.832235e-220
"Lilliefors (Kolmogorov-Smirnov) normality test"
(b) Lilliefors
```

Figura 2.20: Tests de normalidad sobre Visibility

vemos que la variable no se distribuye como una normal. Lo confirmamos con el QQPlot:

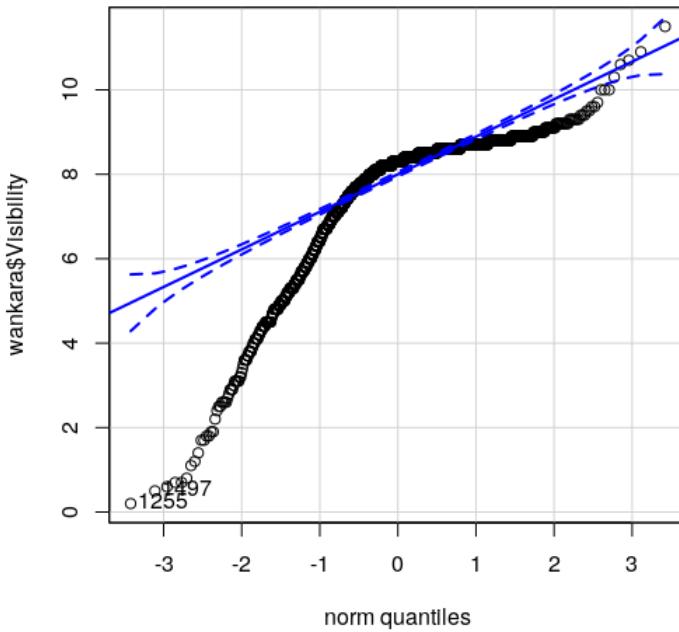


Figura 2.21: QQPlot de Visibility

donde claramente la distribución está fuera de la tendencia de una normal.

#### 2.1.8. Wind speed

Wind speed almacena los datos de la velocidad (entendemos que media) del viento en Ankara cada día. Estudiamos su resumen estadístico para comprenderla mejor:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
0	3.11	5.06	5.393	3.041806	7.25	18

Como podemos ver, se trata de una distribución con media y mediana parecidas y con una considerable desviación típica. Además, los valores máximos y mínimos distan bastante de la media, especialmente el máximo, por lo que se puede tratar de un outlier tanto por un error de medida o por una ventisca. Por otra parte, la variable presenta una asimetría por la derecha (0.7321927) y es leptocúrtica (0.2130076). Veamos su histograma.

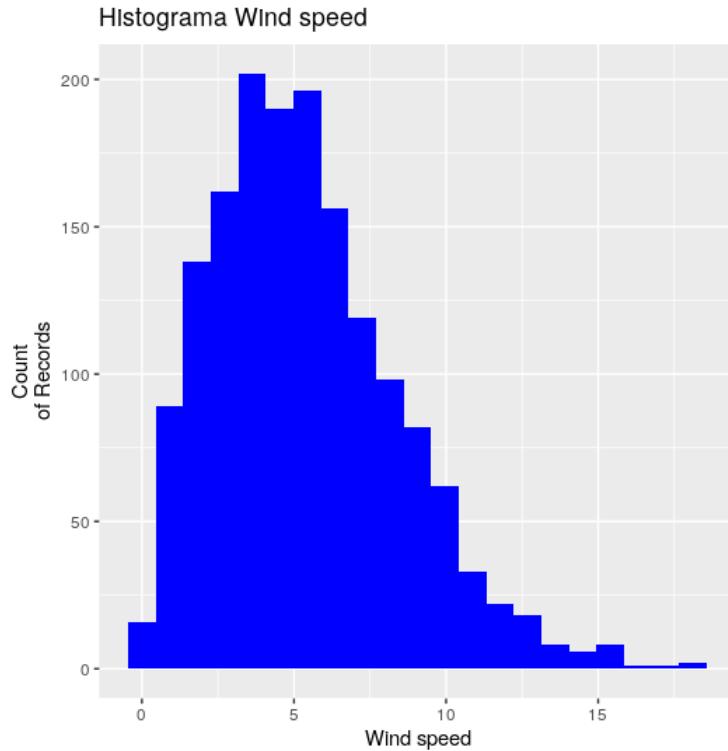


Figura 2.22: Histograma de Wind speed

Vía el histograma, vemos que en Ankara siempre hace viento. Pocos días la velocidad del viento es 0. A falta de conocer la unidad de medida, podríamos decir que los vientos no son excesivamente virulentos.

Estudiamos si sigue una distribución normal vía los tests de normalidad.

```
Wind_speed
0.96508
2.897973e-19
"Shapiro-Wilk normality test"
(a) Shapiro-Wilk
```

```
Wind_speed
0.06182729
9.254139e-16
"Lilliefors (Kolmogorov-Smirnov) normality test"
(b) Lilliefors
```

Figura 2.23: Tests de normalidad sobre Wind speed

Los p-valores nos dejan claro que debemos rechazar la hipótesis de normalidad.

### 2.1.9. Max wind speed

En este caso, estudiamos la variable Max wind speed, con la velocidad máxima del viento cada día en Ankara. Veamos el estudio estadístico de la misma:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
2.19	10.2	12.7	13.32	5.498989	16.1	57.4

Evidentemente, los valores de esta variable son, para cada día, mayores o iguales que los de Wind speed. Aquí encontramos el valor que generaba el outlier (o ventisca) en Wind speed, que es 57.4, prácticamente 5 veces mayor que la media. Presenta, al igual que Wind speed, asimetría por la derecha (1.855093) más pronunciada y es leptocúrtica, con curtosis aún mayor (8.624427). Veamos el histograma.

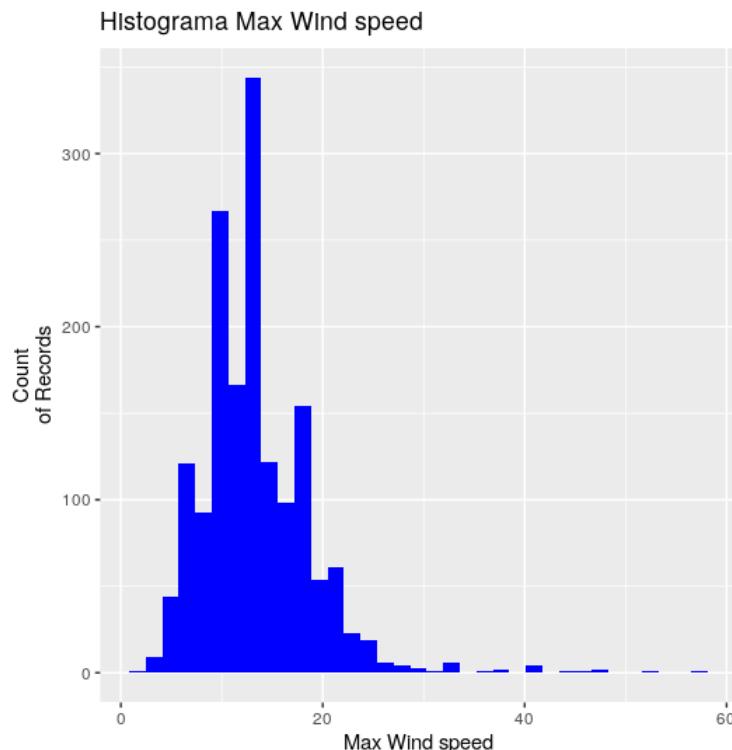


Figura 2.24: Histograma de Max Wind speed

En él podemos encontrar ciertos outliers cuando se supera el valor 30 de velocidad. Parece lógico pensar que no sigue una distribución normal dada la forma del histograma. Estudiamos los tests de normalidad para comprobarlo:

```
Max_wind_speed
0.8878523
1.214158e-32
"Shapiro-Wilk normality test"
-----
```

(a) Shapiro-Wilk

```
Max_wind_speed
0.1143473
3.862197e-57
"Lilliefors (Kolmogorov-Smirnov) normality test"
-----
```

(b) Lilliefors

Figura 2.25: Tests de normalidad sobre Max Wind speed

y su QQPlot

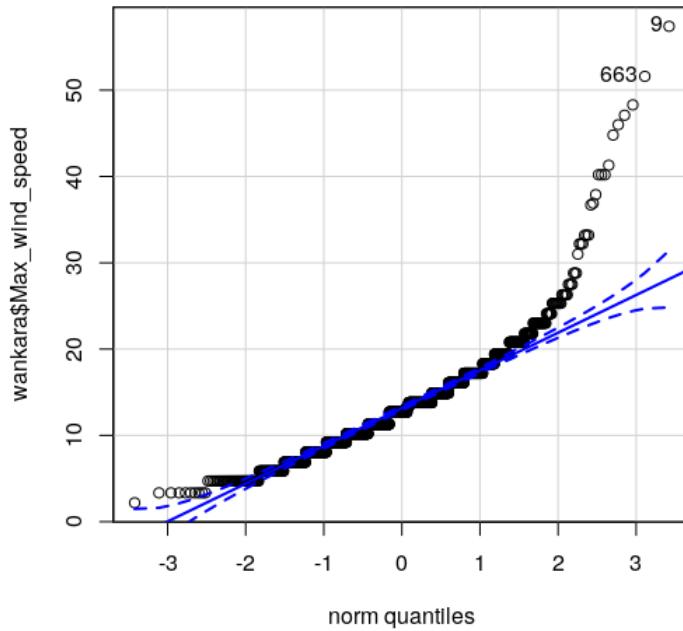


Figura 2.26: QQPlot de Max wind speed

En efecto, rechazamos la hipótesis de normalidad.

### 2.1.10. Mean temperatura

Mean temperature contiene la temperatura media del día en Ankara. Es la variable que queremos aproximar a través de regresores y distintos modelos. A continuación mostraré una serie de hipótesis sobre ella y la relación con las demás variables. Así podremos descubrir cómo aproximarla de forma efectiva y eficiente. Antes de eso, estudiamos su resumen estadístico:

Mín.	1st Qu.	Median	Mean	SD	3rd Qu.	Max
7.90	36.7	48.5	49.56	15.4564	63.3	81

Podemos ver que, con excepción de días muy fríos o algo calurosos, la temperatura media en Ankara se mantiene fría ( $48.5^{\circ}\text{F}$ , menos de  $10^{\circ}\text{C}$ ). Desde el punto de vista más estadístico, apenas tiene asimetría a la derecha (0.03417221) y es platicúrtica (-1.060112). Veamos el histograma.

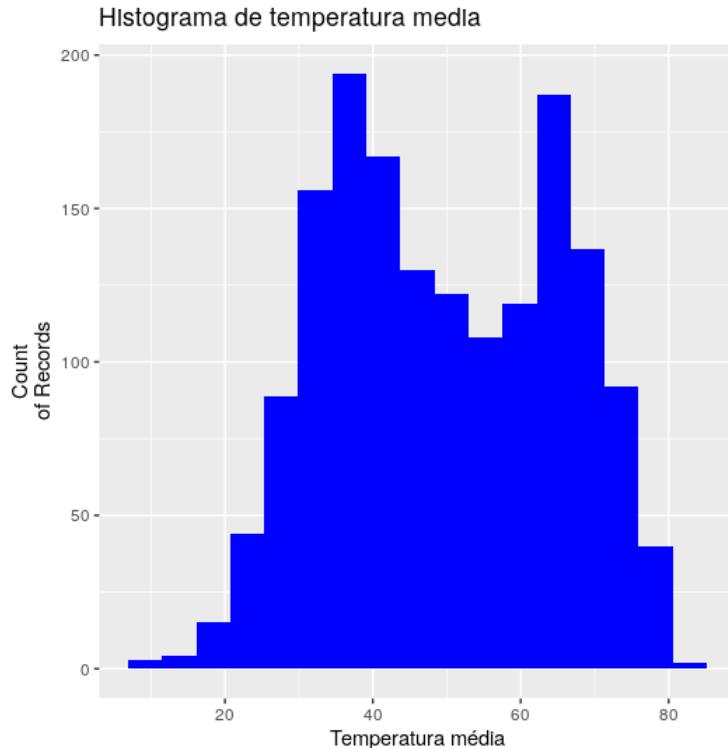


Figura 2.27: Histograma de Mean temperature

Parece como si fuera bimodal. En mi opinión, se corresponde con la estación fría y cálida, donde más se repiten las temperaturas. Veamos si sigue una distribución normal:

```
Mean_temperature
0.9682132
2.386456e-18
"Shapiro-Wilk normality test"
-----
```

(a) Shapiro-Wilk

```
Mean_temperature
0.07919267
1.559339e-26
"Lilliefors (Kolmogorov-Smirnov) normality test"
```

(b) Lilliefors

Figura 2.28: Tests de normalidad sobre Mean temperature

Por los p-valores, rechazamos la hipótesis de normalidad para la variable.

### 2.1.11. Algunas hipótesis previas a estudiar la correlación

Parece lógico pensar que si las temperaturas máximas de un día suben/bajan, las temperaturas medias del mismo día tendrán el mismo comportamiento. De igual manera para las temperaturas mínimas y las medias. Veamos si nuestros datos reflejan eso también.

- **Hipótesis 1: cuanto mayor (menor) es la temperatura máxima, mayor (menor) es la temperatura media**

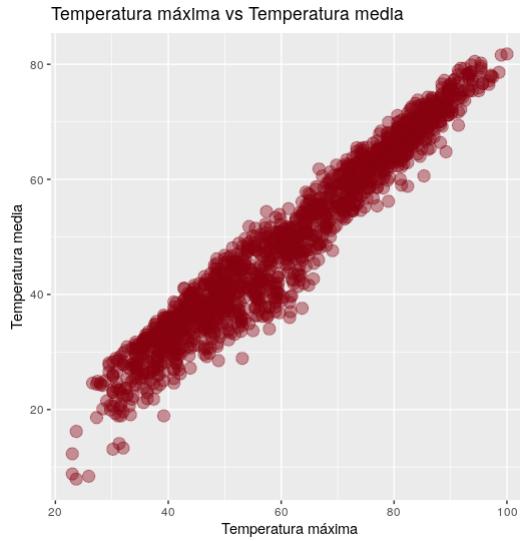


Figura 2.29: Temperatura máxima vs media

En efecto, así es. Hay una dependencia lineal muy clara entre las variables.

- **Hipótesis 2: cuanto mayor (menor) es la temperatura mínima, mayor (menor) es la temperatura media**

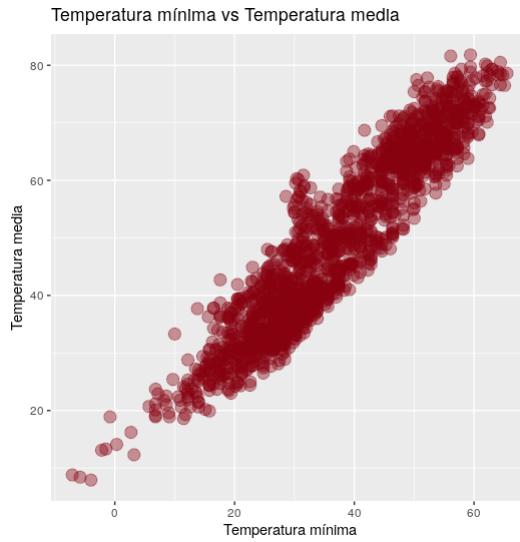


Figura 2.30: Temperatura mínima vs media

Aunque la tendencia no es tan clara como en la gráfica anterior, también vemos cierta dependencia lineal entre las variables, cumpliéndose nuestra hipótesis.

### 2.1.12. División por meses

Durante el análisis exploratorio de datos, he tratado de separar los días por meses. Como se ha comentado, cada instancia corresponde con un día de entre el 01/01/1994 y el 28/05/1998. Habiéndolo llevado a cabo, incluso teniendo en cuenta que 1996 fue bisiesto, he intentado estudiar cómo varía la temperatura máxima y mínima (en media) por meses. Sin embargo, los resultados han sido insatisfactorios:

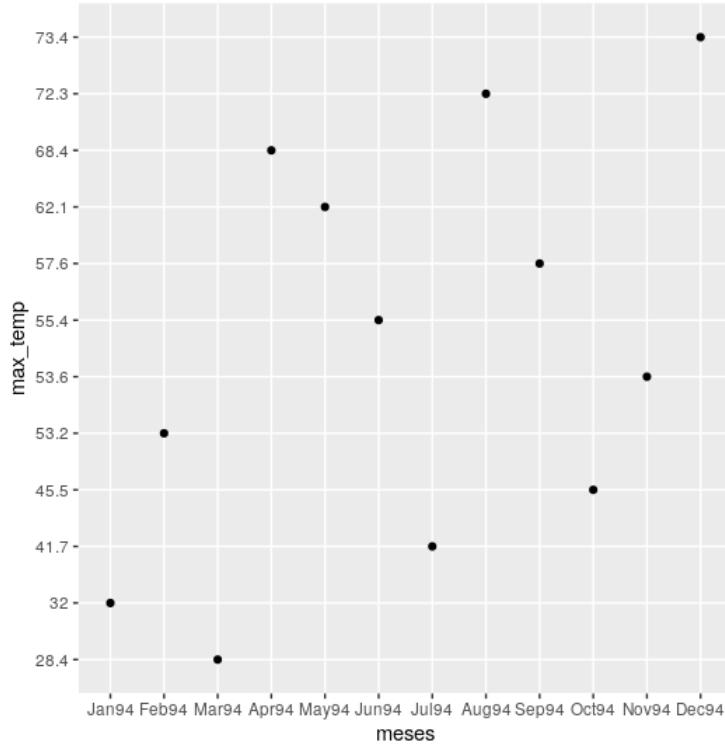


Figura 2.31: Temperatura máxima media de cada mes

Como se puede ver, la mayor temperatura máxima media de 1994 se dio en diciembre (invierno en Ankara) y la tercera más baja en julio (verano). Por tanto sospecho que las tuplas no están ordenadas por fecha. Continué estudiándolo para la temperatura mínima media por mes.

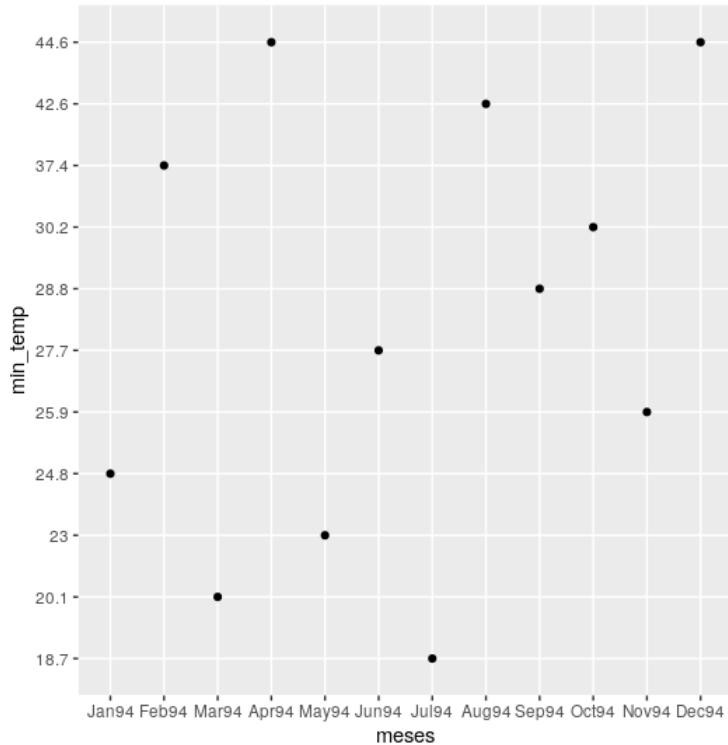


Figura 2.32: Temperatura mínima media de cada mes

De nuevo, los datos no tienen sentido porque la menor temperatura mínima se da en julio, con 18.7°F (-7°C). Por tanto, definitivamente, las instancias no están ordenadas por fechas y no puedo llevar a cabo un estudio mensual para encontrar patrones.

### 2.1.13. Correlación de las variables

Presentamos un gráfico resumen con la correlación de todas las parejas de variables y su nivel de significancia, tanto a través de scatter plots, histogramas y p-valores.

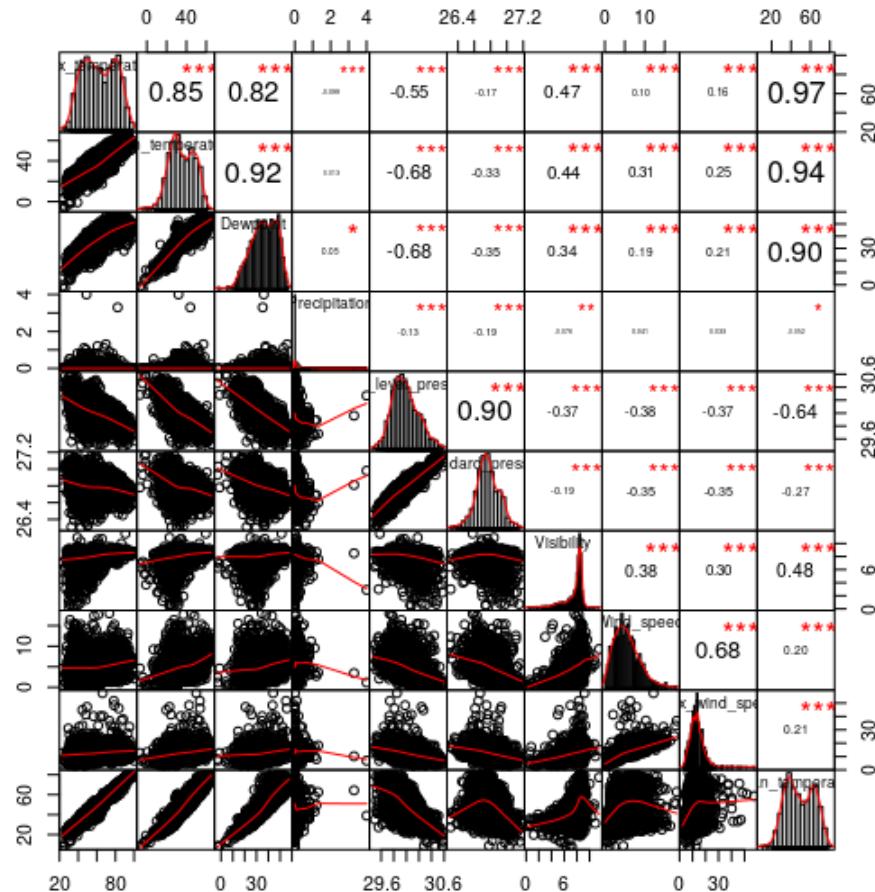


Figura 2.33: Resumen de la correlación

Podemos ver que las variables más correladas con Mean temperature son Max temperature (0.97), Min temperature (0.94) y Dewpoint (0.90). Por el contrario, Precipitation no tienen ninguna correlación con nuestro target. Además, descubrimos correlaciones entre los posibles regresores, que nos pueden dar motivo luego para eliminar uno u otro en los modelos predictivos. Es el caso de Min temperature y Dewpoint (0.92) O Sea level pressure y Standard pressure (0.90). Advertimos también ciertas correlaciones negativas, por ejemplo entre Sea level pressure y Mean temperature (-0.64).

#### **2.1.14. Conclusiones**

El análisis exploratorio de datos realizado sobre el conjunto de datos Wankara nos arroja un montón de información sobre el tiempo en Ankara. Esta ciudad, situada en el centro de la península de Anatolia, presenta un clima más bien continental, con temperaturas frescas que en verano no superan los 38°C (100°F), con una temperatura media de 10°C y con temperaturas mínimas que rondan los 0°C, con extremos de hasta -22°C (-7°F). Además, presenta precipitaciones escasas. La influencia del mar en la ciudad es escasa por la distancia y las montañas que los separa. Podríamos estimar los niveles de contaminación atmosférica entre 1994 y 1998 si supiéramos la unidad de medida de la variable visibilidad. Por ejemplo, si se tratara de km, como la media está alrededor de 7, podríamos asumir que la contaminación es reducida. Tampoco parece ser una ciudad muy ventosa ni con rachas de viento muy fuertes, aunque realmente no conocemos la unidad de medida y los números pueden ser engañosos. Además, hemos constatado que el dataset no está ordenado por fecha, por lo que no podemos encontrar patrones por meses.

## 2.2. Conjunto de datos Vowel

El presente conjunto de datos contiene información sobre el reconocimiento de las once vocales existentes en inglés por parte de 15 interlocutores independientes. Es un problema de clasificación de once clases (las once vocales inglesas) con trece características, diez de ellas reales y tres enteras. A pesar de ser numéricas, esas tres variables son realmente categóricas, dado que

- TT (0/1): Indica si la instancia es de entrenamiento (0) o test (1).
- Sex (0/1): Indica el género del hablante en dicha instancia.
- SpeakerNumber [0,14]: Indica el interlocutor de la instancia.

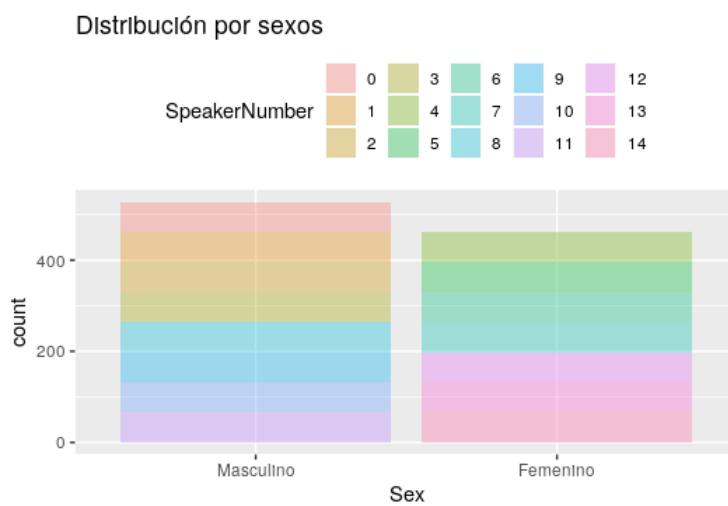


Figura 2.34: Distribución por sexos de los interlocutores

Por tanto, esas variables se pueden interpretar como factores más que numéricas. De hecho, TT será ignorada en el EDA porque es conveniente realizar el estudio sobre los datos al completo. Por tanto, tras estas modificaciones contamos con diez variables reales (F0-F9), dos factores (sexo y número de interlocutor) y once clases (0-10), con un total de 990 instancias.

Para explorar los datos, en primer lugar, calculo un resumen estadístico de cada variable y su dispersión. En las variables categóricas encontramos:

- Cada interlocutor tiene 66 apariciones en el conjunto de datos.
- 528 de ellos son hombres y 462 mujeres.

Para las variables numéricas, los resultados son los siguientes:

	<b>F0</b>	<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>	<b>F5</b>	<b>F6</b>	<b>F7</b>	<b>F8</b>	<b>F9</b>
<b>Min</b>	-5.211	-1.274	-2.487	-1.409	-2.127	-0.836	-1.537	-1.293	-1.631	-1.68
<b>1st Qua</b>	-3.888	1.052	-0.97575	-0.0655	-0.769	0.196	-0.307	-0.09575	-0.704	-0.548
<b>Mediana</b>	-3.146	1877	-0.5725	0.4335	-0.299	0.552	0.022	0.328	-0.3025	-0.1565
<b>Media</b>	-3.204	1.882	-0.50777	0.5155	-0.3057	0.6302	-0.004365	0.33655	-0.30298	-0.07134
<b>3th Qua</b>	-2.603	2.738	-0.06875	1.096	0.1695	1.0285	0.2965	0.77	0.09375	0.371
<b>Max</b>	-0.941	5.074	1.431	2.377	1.831	2.327	1.403	2.039	1.309	1.396
<b>SD</b>	0.8689872	1.1752720	0.7119483	0.7592613	0.6646023	0.6038711	0.4619268	0.5733020	0.5701616	0.6039855

Cuadro 2.1: Resumen estadístico y desviación típica de las características reales

Como se puede observar, el rango y el dominio de cada variable es distinto, lo que podría condicionar el rendimiento de los posteriores algoritmos que utilizaremos. Por tanto, será necesario un reescalado de las mismas para evitar esa discriminación positiva de unas variables respecto a otras sólo por ser "mayores".

A continuación, presento las 10 variables numéricas con más detenimiento. Una de las características principales de este conjunto de datos es que, si se estudia como un todo, el comportamiento de las variables a veces puede parecer errático. Sin embargo, no se debe soslayar el hecho de que tenemos una variable categórica, la del sexo, que nos hace prácticamente crear dos conjuntos de datos quasi-independientes: hombres y mujeres, donde sí que encontramos correlaciones y claves para entender el funcionamiento de las características. Como digo, presento cada una de las variables, primero estudiándola en conjunto y luego separando por sexo. Para comprobar si las variables siguen una distribución normal, se han utilizado el test de Shapiro-Wilk ([\[8\]](#)) y la corrección de Lilliefors del test de Kolmogorov-Smirnov ([\[6\]](#)). Además, para estudiar el comportamiento tanto por sexo como por interlocutor, represento vía boxplots cada variable.

### 2.2.1. F0

Presentamos la variable F0. En primer lugar, reflejo un resumen estadístico de la misma así como su histograma diferenciando entre hombres y mujeres.

- Media: -3.204
- Mediana: -3.146
- Desviación típica: 0.8689872
- Rango: [-5.211,-0.941]
- Primer y tercer cuartiles: (-3.888,-2.603)
- Asimetría: 0.0662973
- Curtosis: -0.4974651

Por el valor de la curtosis, la variable es platicúrtica y muy ligeramente asimétrica por la derecha.

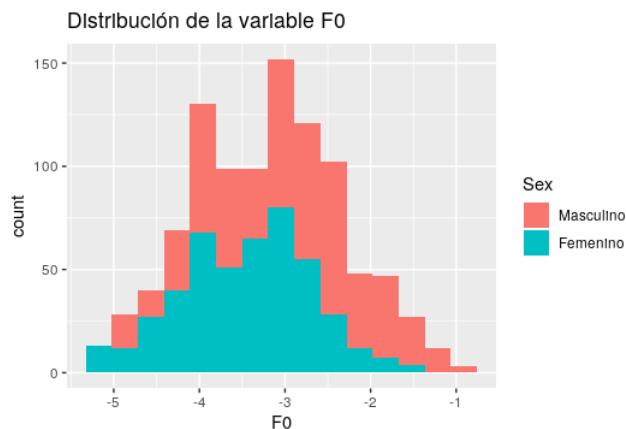


Figura 2.35: Histograma de la variable F0

Pasamos a comprobar si F0 se distribuye según una normal. Para ello, establecemos los tests de hipótesis de Shapiro-Wilk y Lilliefors (Kolmogorov-Smirnov) con los siguientes resultados:

```
> shapiro.test(vowel$F0)
Shapiro-Wilk normality test
data: vowel$F0
W = 0.9928, p-value = 9.807e-05
(a) Shapiro-Wilk
```

```
> lillie.test(vowel$F0)
Lilliefors (Kolmogorov-Smirnov) normality
test
data: vowel$F0
D = 0.040134, p-value = 0.000719
(b) Lilliefors
```

Figura 2.36: Tests de normalidad sobre F0

Como los p-valores son menores que 0.05, rechazamos la hipótesis nula (la variable sigue una distribución normal).

A continuación presento los boxplots:

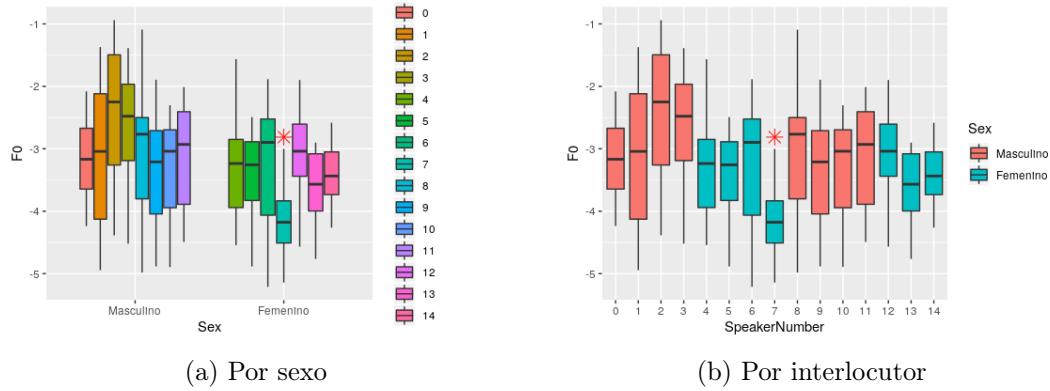


Figura 2.37: Boxplot para F0 estudiando sexos e interlocutores

Como se puede ver, existe un outlier en el interlocutor 7. Además, observamos que los valores para los hombres tienden a ser mayores que para las mujeres.

Si ahora estudiamos los tests de normalidad por sexos, encontramos los siguientes resultados:

```
> shapiro.test(hombres$F0)
Shapiro-Wilk normality test
data: hombres$F0
W = 0.98625, p-value = 6.965e-05
(a) Shapiro-Wilk
```

```
> lillie.test(hombres$F0)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F0
D = 0.050372, p-value = 0.002766
(b) Lilliefors
```

Figura 2.38: Tests de normalidad sobre F0 (hombres)

En este caso, aunque también se rechaza la hipótesis de normalidad, el test de Lilliefors arroja un resultado menor pero cercano a 0.05, por lo que parece acercarse más la variable en los hombres a una distribución normal que con el conjunto completo.

Para las mujeres,

```
| > shapiro.test(mujeres$F0)
Shapiro-Wilk normality test
data: mujeres$F0
W = 0.99068, p-value = 0.005136
(a) Shapiro-Wilk
```

```
> lillie.test(mujeres$F0)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F0
D = 0.060494, p-value = 0.0003573
(b) Lilliefors
```

Figura 2.39: Tests de normalidad sobre F0 (mujeres)

se encuentran unos p-valores menores que para los hombres, por lo que también se rechaza la hipótesis de normalidad.

### 2.2.2. F1

Como en el apartado anterior, primero reflejo un resumen estadístico de la variable:

- Media: 1.882
- Mediana: 1.877
- Desviación típica: 1.175272
- Rango: [-1.274,5.074]
- Primer y tercer cuartil: (1.052,2.738)
- Asimetría: -0.04269788
- Curtosis: -0.39925

Por el valor de la curtosis, la variable es platicúrtica y muy ligeramente asimétrica por la izquierda.

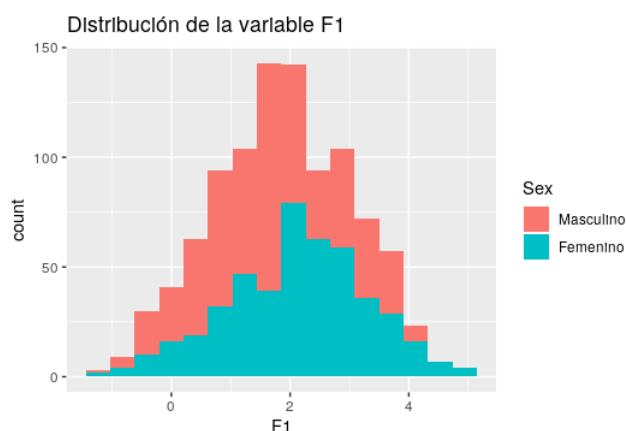


Figura 2.40: Histograma de la variable F1

Pasamos a estudiar la normalidad de la variable. Estos son los resultados de los tests:

```
> shapiro.test(vowel$F1)
Shapiro-Wilk normality test
data: vowel$F1
W = 0.9966, p-value = 0.0312
(a) Shapiro-Wilk
```

```
> lillie.test(vowel$F1)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F1
D = 0.022451, p-value = 0.2628
(b) Lilliefors
```

Figura 2.41: Tests de normalidad sobre F1

Encontramos una discrepancia en los tests. Según el p-valor de Shapiro-Wilk, debemos rechazar la hipótesis de normalidad. Sin embargo, Lilliefors nos indica (0.2628) que no podemos rechazarla. Para esclarecer un poco más el comportamiento de F1, realizo la gráfica de densidad con una normal superpuesta con la media y desviación típica de F1:

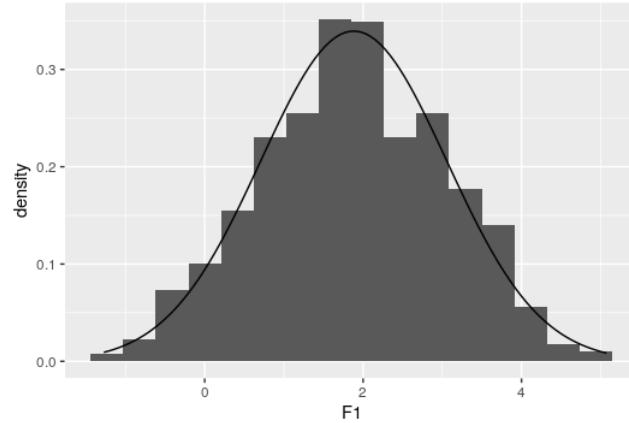


Figura 2.42: Histograma con la densidad de F1 y normal para comparar

así como un QQPlot ([9]):

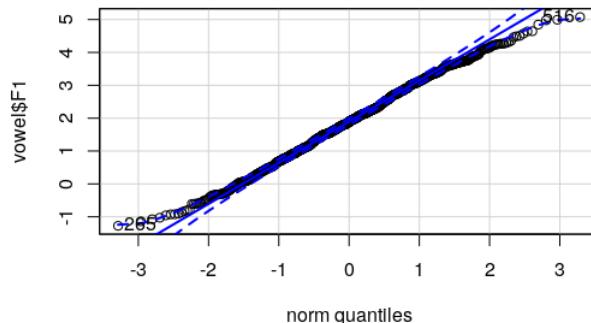


Figura 2.43: QQPlot de la variable F1

A la luz de los resultados, podemos interpretar que F1 tiene un comportamiento muy parecido a la distribución normal. A continuación, presento los boxplots:

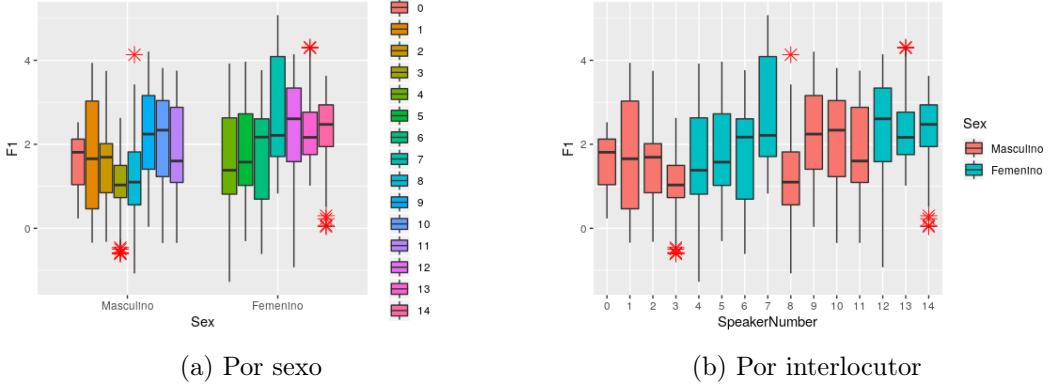


Figura 2.44: Boxplot para F1 estudiando sexos e interlocutores

Podemos apreciar outliers en los interlocutores 3,8,13 y 14.

Si ahora estudiamos los tests de normalidad por sexos, encontramos los siguientes resultados. Para los hombres

```
> shapiro.test(hombres$F1)
Shapiro-Wilk normality test
data: hombres$F1
W = 0.98797, p-value = 0.0002435
(a) Shapiro-Wilk
```

```
> lillie.test(hombres$F1)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F1
D = 0.047653, p-value = 0.006123
(b) Lilliefors
```

Figura 2.45: Tests de normalidad sobre F1 (hombres)

Ambos tests nos llevan a rechazar la hipótesis de normalidad. En el caso de las mujeres

```
> shapiro.test(mujeres$F1)
Shapiro-Wilk normality test
data: mujeres$F1
W = 0.99278, p-value = 0.02526
(a) Shapiro-Wilk
```

```
> lillie.test(mujeres$F1)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F1
D = 0.064796, p-value = 8.261e-05
(b) Lilliefors
```

Figura 2.46: Tests de normalidad sobre F1 (mujeres)

también rechazamos la hipótesis nula. Por tanto, en conjunto la variable se comporta según una normal pero por separado (hombres/mujeres) no.

### 2.2.3. F2

Presentamos la variable F2. En primer lugar, reflejo un resumen estadístico de la misma así como su histograma diferenciando entre hombres y mujeres.

- Media: -0.50777

- Mediana: -0.5725
- Desviación típica: 0.7119483
- Rango: [-2.487,-1.431]
- Primer y tercer cuartiles: (-0.97575,-0.06875)
- Asimetría: 0.2352169
- Curtosis: -0.1575597

Por el valor de la curtosis, la variable es platicúrtica y ligeramente asimétrica por la derecha.

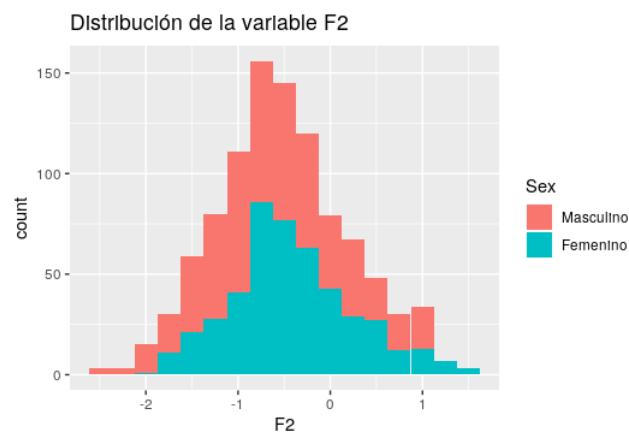


Figura 2.47: Histograma de la variable F2

Pasamos a estudiar la normalidad de la variable. Estos son los resultados de los tests:

```
> shapiro.test(vowel$F2)
Shapiro-Wilk normality test
data: vowel$F2
W = 0.99217, p-value = 4.245e-05
(a) Shapiro-Wilk
```

```
> lillie.test(vowel$F2)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F2
D = 0.038867, p-value = 0.001264
(b) Lilliefors
```

Figura 2.48: Tests de normalidad sobre F2

Ambos tests nos indican que debemos rechazar la hipótesis de normalidad. En cuanto a los boxplots:

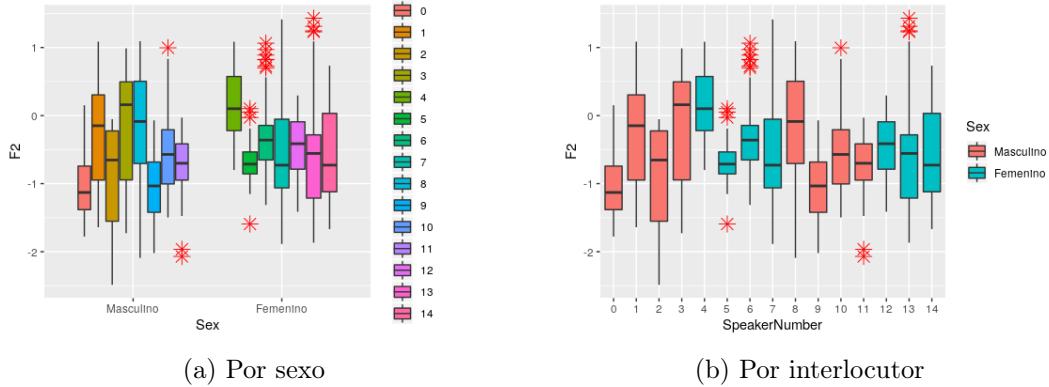


Figura 2.49: Boxplot para F2 estudiando sexos e interlocutores

Encontramos en F2 una gran concentración de outliers para los interlocutores 5,6,10,11 y 13. Es posible que las mediciones hayan sido erróneas. En cualquier caso, podría ser importante la aplicación de técnicas para la disminución de estos valores extraños.

Si ahora estudiamos los tests de normalidad por sexos, encontramos los siguientes resultados:

```
> shapiro.test(hombres$F2)
Shapiro-Wilk normality test
data: hombres$F2
W = 0.98848, p-value = 0.0003589
> lillie.test(hombres$F2)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F2
D = 0.047508, p-value = 0.006377
(a) Shapiro-Wilk
(b) Lilliefors
```

Figura 2.50: Tests de normalidad sobre F2 (hombres)

```
> shapiro.test(mujeres$F2)
Shapiro-Wilk normality test
data: mujeres$F2
W = 0.9815, p-value = 1.284e-05
> lillie.test(mujeres$F2)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F2
D = 0.069486, p-value = 1.461e-05
(a) Shapiro-Wilk
(b) Lilliefors
```

Figura 2.51: Tests de normalidad sobre F2 (mujeres)

Para ambos subconjuntos, los tests nos indican que debemos rechazar la hipótesis de normalidad.

#### 2.2.4. F3

Como en el apartado anterior, primero reflejo un resumen estadístico de la variable:

- Media: 0.5155

- Mediana: 0.4335
- Desviación típica: 0.7592613
- Rango: [-1.409,2.377]
- Primer y tercer cuartil: (-0.0655,1.096)
- Asimetría: 0.1287436
- Curtosis: -0.39925

Por el valor de la curtosis, la variable es platicúrtica y ligeramente asimétrica por la derecha.

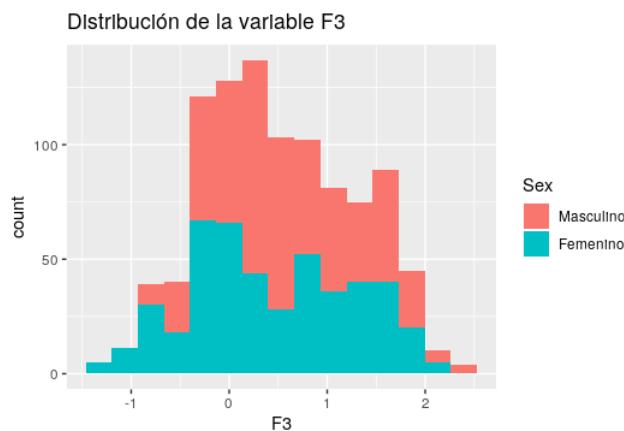


Figura 2.52: Histograma de la variable F3

Evalúo ahora los tests de normalidad:

```
> shapiro.test(vowel$F3)
Shapiro-Wilk normality test
data: vowel$F3
W = 0.98364, p-value = 4.324e-09

> lillie.test(vowel$F3)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F3
D = 0.053431, p-value = 5.169e-07
```

(a) Shapiro-Wilk (b) Lilliefors

Figura 2.53: Tests de normalidad sobre F3

Como se puede ver, rechazamos la hipótesis de normalidad por ser los p-valores menores de 0.05.

Estudiamos ahora los boxplots:

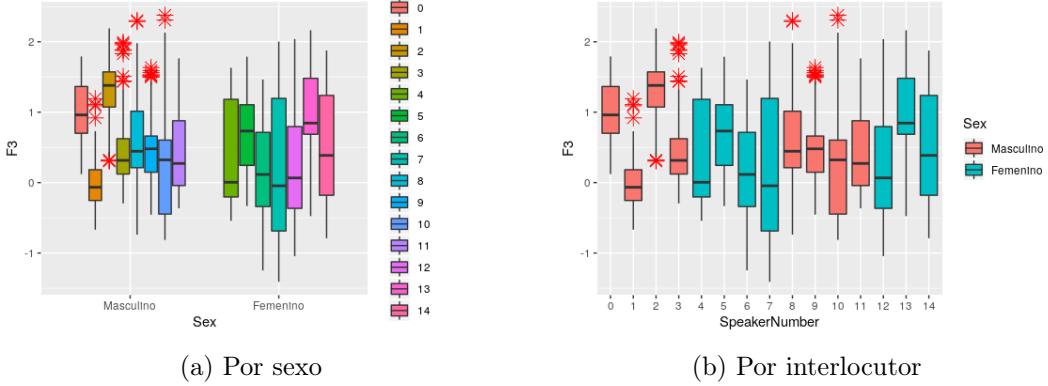


Figura 2.54: Boxplot para F3 estudiando sexos e interlocutores

Como se puede observar, los interlocutores 1,3 y 9 tienen bastantes outliers, mientras que los 2, 8 y 10 presentan outliers en menor medida. Con respecto a los tests de normalidad para hombres y mujeres, vemos que debemos rechazar la hipótesis de normalidad en ambos casos:

```
> shapiro.test(hombres$F3)
Shapiro-Wilk normality test
data: hombres$F3
W = 0.97673, p-value = 1.919e-07
(a) Shapiro-Wilk
```

```
> lillie.test(hombres$F3)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F3
D = 0.062758, p-value = 3.723e-05
(b) Lilliefors
```

Figura 2.55: Tests de normalidad sobre F3 (hombres)

```
> shapiro.test(mujeres$F3)
Shapiro-Wilk normality test
data: mujeres$F3
W = 0.97442, p-value = 3.099e-07
(a) Shapiro-Wilk
```

```
> lillie.test(mujeres$F3)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F3
D = 0.070814, p-value = 8.722e-06
(b) Lilliefors
```

Figura 2.56: Tests de normalidad sobre F3 (mujeres)

## 2.2.5. F4

En primer lugar, reflejo un resumen estadístico de la variable:

- Media: -0.3057
- Mediana: -0.299
- Desviación típica: 0.6646023
- Rango: [-2.127,1.831]

- Primer y tercer cuartil: (-0.769,0.1695)
- Asimetría: 0.01647417
- Curtosis: -0.2773318

Por el valor de la curtosis, la variable es platicúrtica y apenas asimétrica por la derecha. Presentamos el histograma para conocer más los datos:

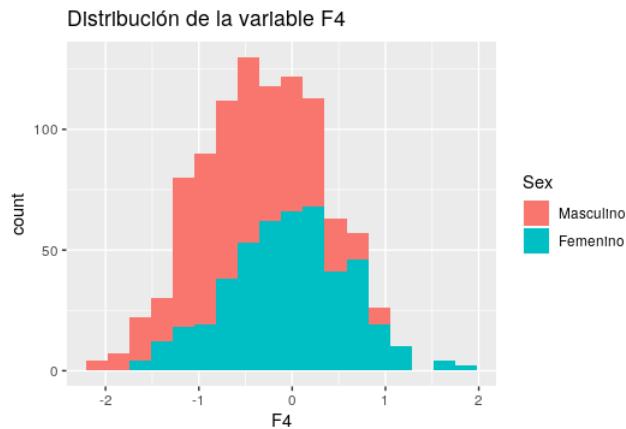


Figura 2.57: Histograma de la variable F4

Pasamos a comprobar si F4 se distribuye según una normal. Para ello, establecemos los tests de hipótesis de Shapiro-Wilk y Lilliefors:

```
> shapiro.test(vowel$F4)
Shapiro-Wilk normality test
data: vowel$F4
W = 0.9971, p-value = 0.07073
(a) Shapiro-Wilk

> lillie.test(vowel$F4)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F4
D = 0.027142, p-value = 0.08258
(b) Lilliefors
```

Figura 2.58: Tests de normalidad sobre F4

Ambos tests indican que no podemos rechazar la hipótesis de normalidad. Confirmamos el resultado con el gráfico QQPlot.

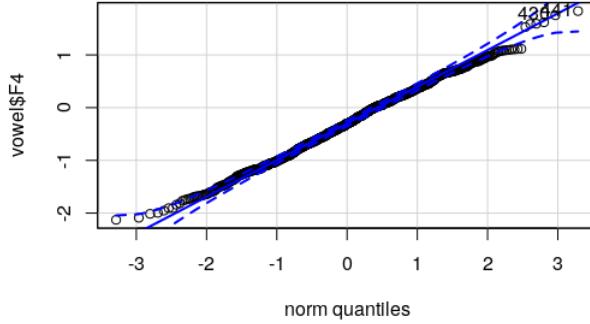


Figura 2.59: Gráfico QQPlot de F4

Por tanto, podemos asumir que F4 se distribuye como una normal. Estudiamos ahora los boxplots:

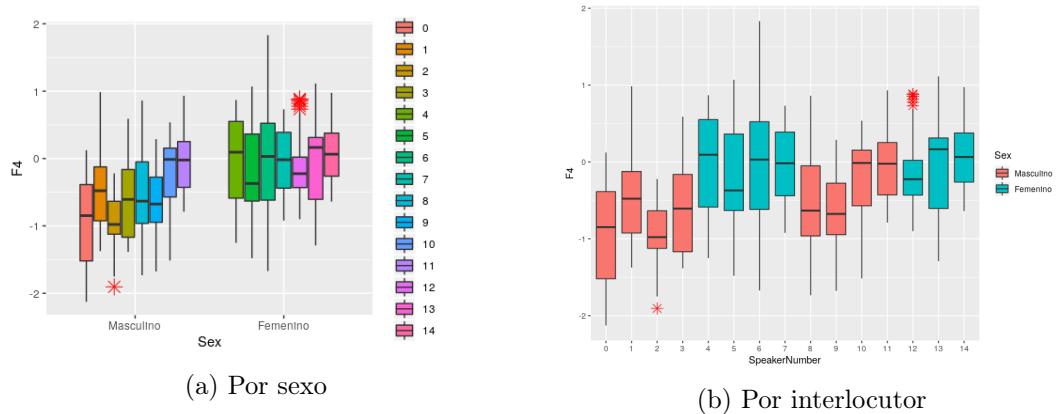


Figura 2.60: Boxplot para F4 estudiando sexos e interlocutores

donde apreciamos una cantidad considerable de outliers en el interlocutor 12. Además, en esta variable, las mujeres consiguen valores más altos que los hombres.

Por último, estudiamos los tests de normalidad por sexo. En el caso de los hombres,

```
> shapiro.test(hombres$F4)
Shapiro-Wilk normality test
data: hombres$F4
W = 0.9949, p-value = 0.07804
(a) Shapiro-Wilk

> lillie.test(hombres$F4)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F4
D = 0.039525, p-value = 0.04757
(b) Lilliefors
```

Figura 2.61: Tests de normalidad sobre F4 (hombres)

el test de Shapiro-Wilk indica que no podemos rechazar la hipótesis de normalidad (p valor 0.07) mientras que el de Lilliefors indica que la rechacemos. En la literatura, el test de Lilliefors resulta más fiable que el de Shapiro-Wilk por el número de instancias que tenemos, así que rechazamos la hipótesis de normalidad.

Para las mujeres,

```
> shapiro.test(mujeres$F4)

Shapiro-Wilk normality test
data: mujeres$F4
W = 0.99475, p-value = 0.1163
```

(a) Shapiro-Wilk

```
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F4
D = 0.028704, p-value = 0.4692
```

(b) Lilliefors

Figura 2.62: Tests de normalidad sobre F4 (mujeres)

los p-valores son bastante mayores que 0.05, por lo que no podemos rechazar la hipótesis de normalidad. Para confirmarlo, mostramos el QQPlot de F4 para mujeres.

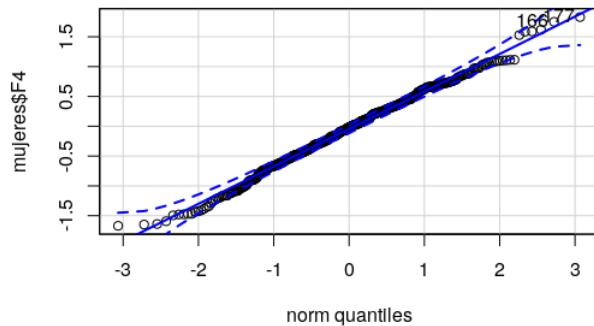


Figura 2.63: Gráfico QQPlot de F4 (mujeres)

Efectivamente, las instancias de F4 de sexo femenino siguen una distribución normal.

### 2.2.6. F5

A continuación presentamos la variable F5. Como en los casos anteriores, comenzamos por un resumen estadístico:

- Media: 0.6302
- Mediana: 0.552
- Desviación típica: 0.6038711
- Rango: [-0.836,2.327]

- Primer y tercer cuartil: (0.196,1.0285)
- Asimetría: 0.3559791
- Curtosis: -0.2964589

A la vista de los resultados, la variable es platicúrtica y ligeramente asimétrica hacia la derecha. Veamos ahora el histograma separado por sexos:

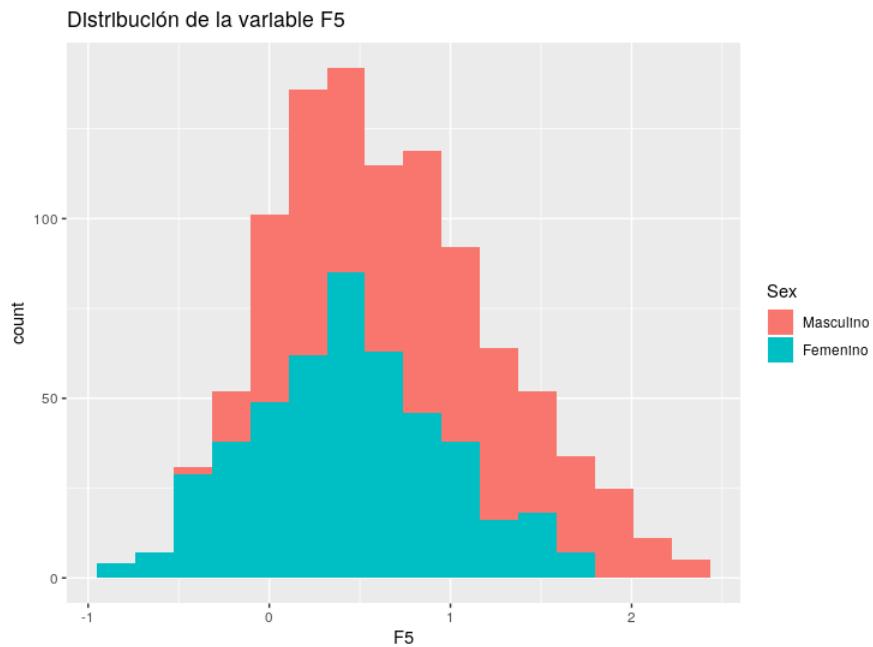


Figura 2.64: Histograma de la variable F5

Estudio si F5 se distribuye según una normal vía los tests estadísticos. Los resultados son los siguientes:

```
> shapiro.test(vowel$F5)
Shapiro-Wilk normality test
data: vowel$F5
W = 0.98632, p-value = 5.435e-08
(a) Shapiro-Wilk

> lillie.test(vowel$F5)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F5
D = 0.058356, p-value = 1.921e-08
(b) Lilliefors
```

Figura 2.65: Tests de normalidad sobre F5

Ambos tests nos indican que debemos rechazar la hipótesis de normalidad por el bajo p-valor. Si estudiamos los boxplots, encontramos que

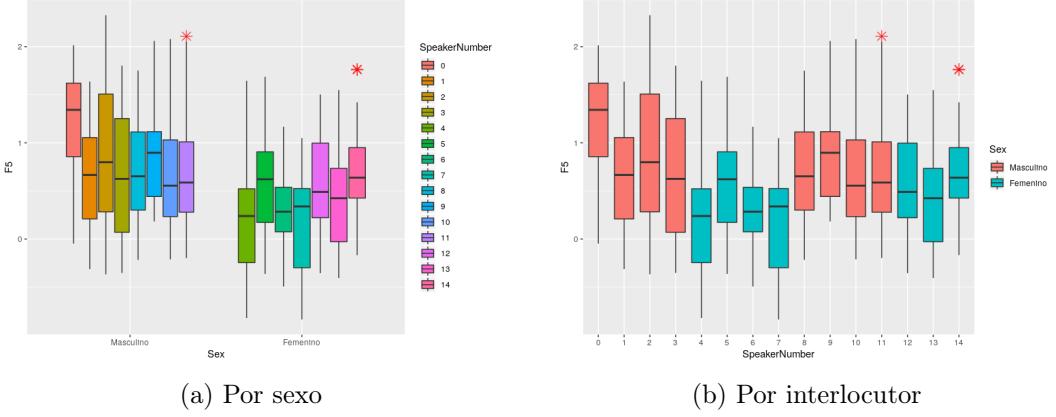


Figura 2.66: Boxplot para F5 estudiando sexos e interlocutores

con apenas outliers en los interlocutores 11 y 14. Paso a estudiar la distribución por sexos. En el caso de los hombres, comprobamos la normalidad

```
> shapiro.test(hombres$F5)
Shapiro-Wilk normality test
data: hombres$F5
W = 0.9704, p-value = 7.718e-09
(a) Shapiro-Wilk
```

```
> lilliefors.test(hombres$F5)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F5
D = 0.083122, p-value = 2.685e-09
(b) Lilliefors
```

Figura 2.67: Tests de normalidad sobre F5 (hombres)

rechazando la hipótesis de normalidad como consecuencia de los dos tests. Para las mujeres

```
> shapiro.test(mujeres$F5)
Shapiro-Wilk normality test
data: mujeres$F5
W = 0.99348, p-value = 0.04365
(a) Shapiro-Wilk
```

```
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F5
D = 0.039485, p-value = 0.08174
(b) Lilliefors
```

Figura 2.68: Tests de normalidad sobre F5 (mujeres)

encontramos que Shapiro-Wilk nos indica que rechacemos la hipótesis de normalidad ( $0.04365 < 0.05$ ) pero Lilliefors no ( $0.08174 > 0.05$ ). Como siempre, le doy mayor credibilidad al test de Lilliefors, por lo que no podemos rechazar la hipótesis de normalidad. Para confirmarlo, muestro el gráfico QQ.

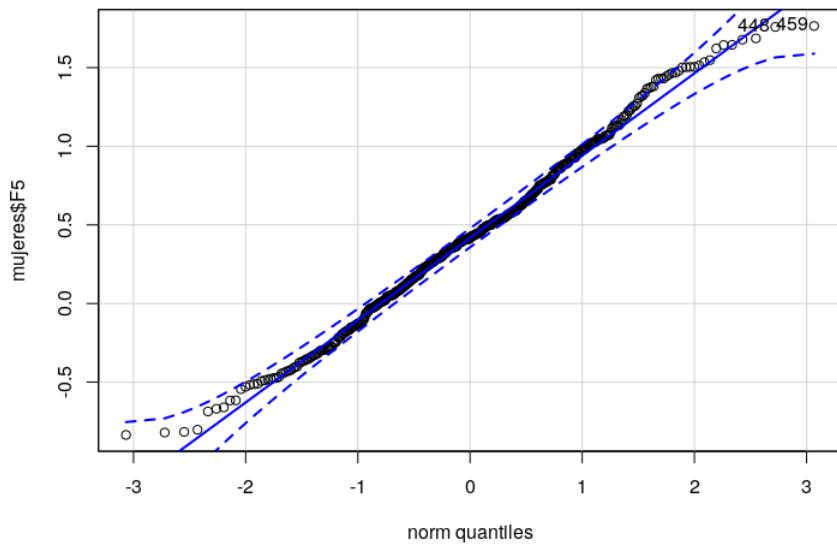


Figura 2.69: QQPlot de la variable F5 (mujeres)

En efecto, la gráfica nos muestra que la variable se distribuye al menos de forma muy pareja a la normal. Volvemos a encontrar un caso donde el subconjunto de interlocutores masculinos difiere en distribución respecto del femenino.

### 2.2.7. F6

El resumen estadístico de la variable F6 es el siguiente:

- Media: -0.004365
- Mediana: 0.022
- Desviación típica: 0.4619268
- Rango: [-1.537,1.403]
- Primer y tercer cuartiles: (-0.307,0.2965)
- Asimetría: -0.2055278
- Curtosis: 0.139262

Por lo que la variable F6 tiene una asimetría por la izquierda y es leptocúrtica. Veamos su histograma para confirmarlo.

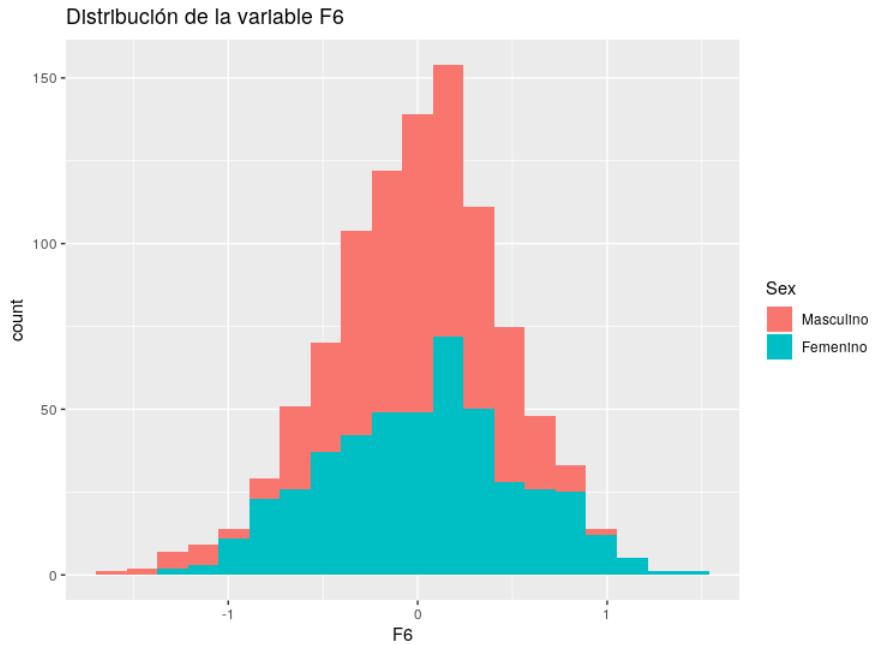


Figura 2.70: Histograma de la variable F6

Examinamos si se distribuye como una normal con los tests siguientes.

```
> shapiro.test(vowel$F6)
Shapiro-Wilk normality test
data: vowel$F6
W = 0.9964, p-value = 0.0225
(a) Shapiro-Wilk

> lillie.test(vowel$F6)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F6
D = 0.0268, p-value = 0.09071
(b) Lilliefors
```

Figura 2.71: Tests de normalidad sobre F6

De nuevo, Shapiro-Wilk nos indica que rechacemos la hipótesis de normalidad mientras Kolmogorov-Smirnov no nos da motivos suficientes para rechazar la hipótesis. Confirmamos que se distribuye como una normal gracias al gráfico QQ.

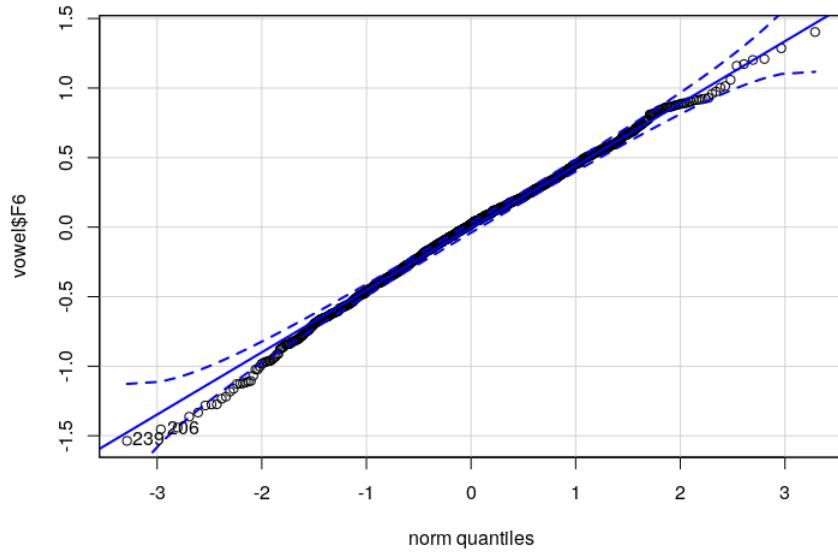


Figura 2.72: QQPlot de la variable F6

quedándose dentro de los márgenes de la normal, por lo que asumimos que F6 se distribuye como una normal. Examinemos más detenidamente su comportamiento con boxplots:

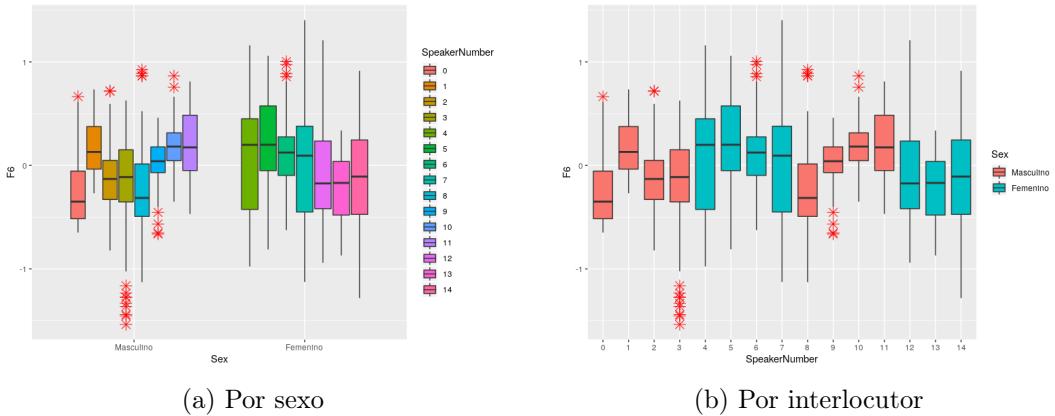


Figura 2.73: Boxplot para F6 estudiando sexos e interlocutores

Encontramos en ellos una gran cantidad de outliers, sobre todo concentrados en el sexo masculino (interlocutor 3 especialmente). Veamos ahora la distribución por sexos vía los tests estadísticos.

```

> shapiro.test(hombres$F6)
Shapiro-Wilk normality test
data: hombres$F6
W = 0.97772, p-value = 3.306e-07
(a) Shapiro-Wilk

```

```

Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F6
D = 0.039182, p-value = 0.05132
(b) Lilliefors

```

Figura 2.74: Tests de normalidad sobre F6 (hombres)

El test de Shapiro-Wilk nos indica que rechacemos la hipótesis de normalidad mientras que Lilliefors, con un p-valor de 0.05132, no puede rechazar la hipótesis nula, aunque con muy poca certeza estadística. El gráfico QQ nos confirma que las colas de los datos están fuera del comportamiento de la distribución normal, por lo que el p-valor se acerca mucho a 0.05.

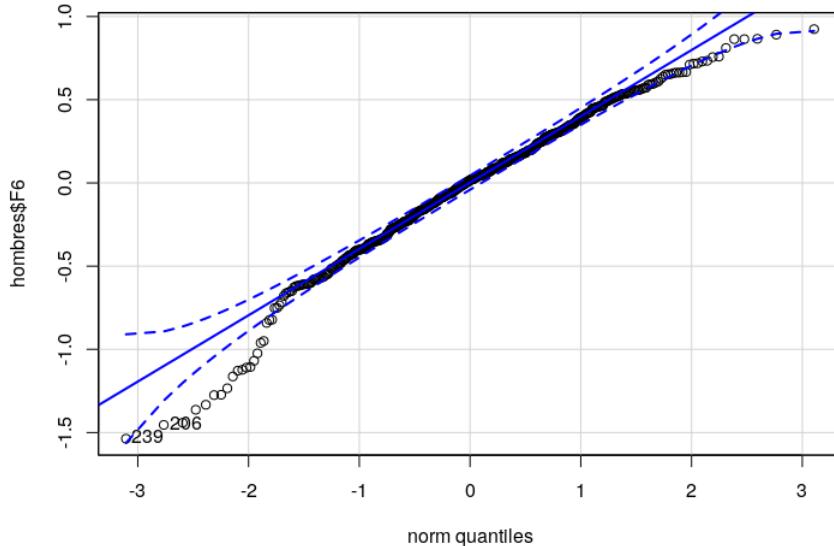


Figura 2.75: QQPlot de la variable F6 (hombres)

Este resultado puede venir influenciado por la gran cantidad de outliers. En el caso de las mujeres

```

> shapiro.test(mujeres$F6)
Shapiro-Wilk normality test
data: mujeres$F6
W = 0.99451, p-value = 0.09697
(a) Shapiro-Wilk

```

```

Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F6
D = 0.03287, p-value = 0.2615
(b) Lilliefors

```

Figura 2.76: Tests de normalidad sobre F6 (mujeres)

ambos tests nos indican que no podemos rechazar la hipótesis de normalidad. Además, el gráfico QQ nos lo confirma

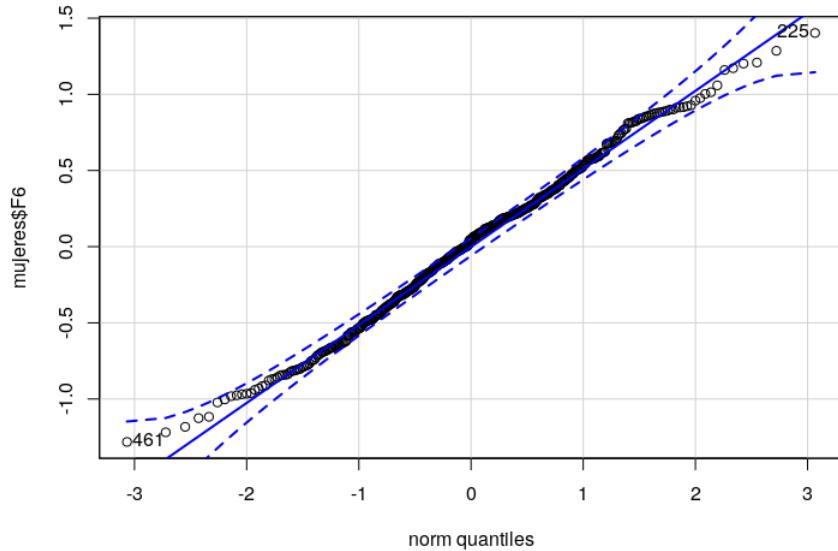


Figura 2.77: QQPlot de la variable F6 (mujeres)

por lo que asumimos que la variable F6 para las mujeres sigue una distribución normal.

### 2.2.8. F7

Como en todas las anteriores, presento el resumen estadístico de la variable F7 y su histograma.

- Media: 0.33655
- Mediana: 0.328
- Desviación típica: 0.5733020
- Rango: [-1.293,2.039]
- Primer y tercer cuartiles: (-0.09575,0.77)
- Asimetría: 0.005939385
- Curtosis: -0.4980897

La variable apenas presenta asimetría por la derecha y es platicúrtica.

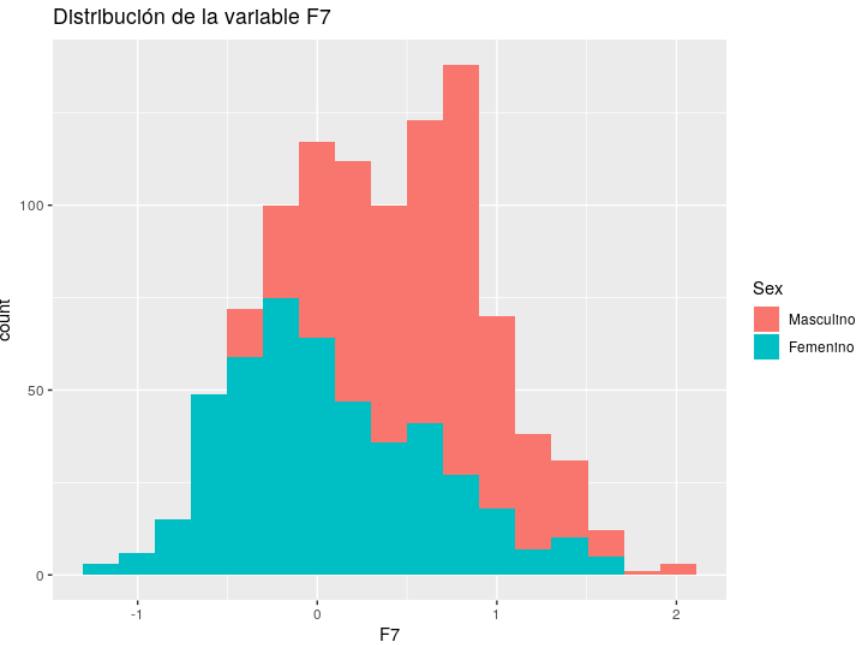


Figura 2.78: Histograma de la variable F7

Podemos ver, gracias al histograma, que la población de mujeres tiene gran asimetría por la derecha (0.4939301). Estudiando los tests estadísticos para la normalidad

```
> shapiro.test(vowel$F7)
Shapiro-Wilk normality test
data: vowel$F7
W = 0.99397, p-value = 0.0005086
(a) Shapiro-Wilk

> lillie.test(vowel$F7)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F7
D = 0.042723, p-value = 0.0002122
(b) Lilliefors
```

Figura 2.79: Tests de normalidad sobre F7

vemos que ambos tests nos hacen rechazar la hipótesis de normalidad. Si observamos los boxplots

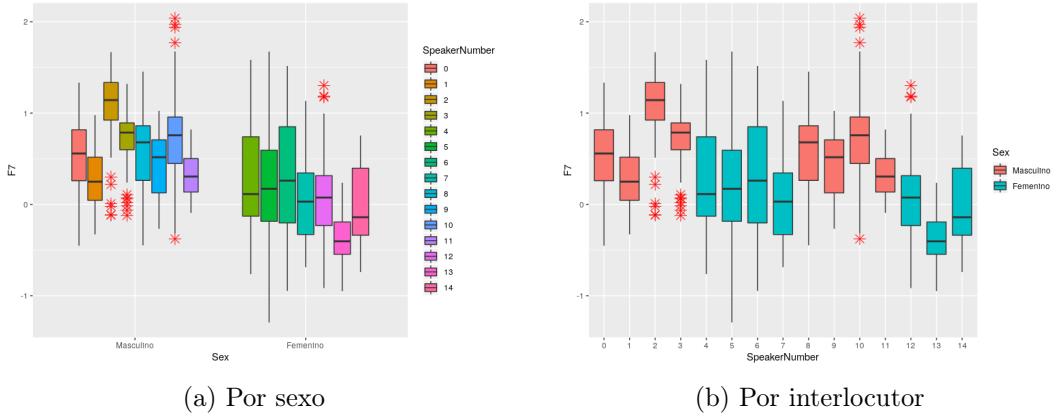


Figura 2.80: Boxplot para F7 estudiando sexos e interlocutores

vemos varios outliers de nuevo sobre todo en los hombres (interlocutores 2,3 y 11). Si estudiamos estos conjuntos por separado

```
> shapiro.test(hombres$F7)
Shapiro-Wilk normality test
data: hombres$F7
W = 0.99257, p-value = 0.009972
```

(a) Shapiro-Wilk

```
> lillie.test(hombres$F7)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F7
D = 0.045749, p-value = 0.01034
```

(b) Lilliefors

Figura 2.81: Tests de normalidad sobre F7 (hombres)

ambos estadísticos nos hacen rechazar la hipótesis de normalidad (con posibles modificaciones tras tratar los outliers), al igual que las mujeres

```
Shapiro-Wilk normality test
data: mujeres$F7
W = 0.97637, p-value = 8.115e-07
```

(a) Shapiro-Wilk

```
> lillie.test(mujeres$F7)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F7
D = 0.070585, p-value = 9.54e-06
```

(b) Lilliefors

Figura 2.82: Tests de normalidad sobre F7 (mujeres)

Debido a la asimetría, una transformación de tipo logaritmo o raíz cuadrada podría corregirse y dar lugar, además, a una distribución más parecida a la normal.

## 2.2.9. F8

He aquí el resumen estadístico e histograma de la variable F8:

- Media: -0.30298
- Mediana: -0.3025

- Desviación típica: 0.5701616
- Rango: [-1.631,1.309]
- Primer y tercer cuartiles: (-0.704,0.09375)
- Asimetría: 0.05370663
- Curtosis: -0.465887

La variable presenta asimetría por la derecha y es platicúrtica.

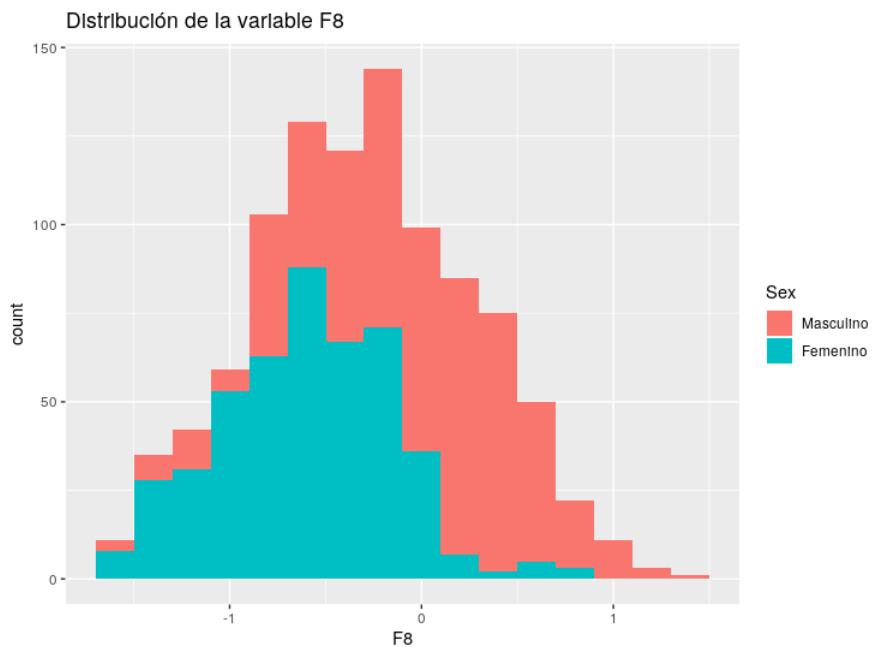


Figura 2.83: Histograma de la variable F8

Estudio la distribución de F8 vía los tests estadísticos con hipótesis nula que la variable sigue una distribución normal.

```
> shapiro.test(vowel$F8)
Shapiro-Wilk normality test
data: vowel$F8
W = 0.99467, p-value = 0.001437
(a) Shapiro-Wilk
```

```
> lillie.test(vowel$F8)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F8
D = 0.024719, p-value = 0.1505
(b) Lilliefors
```

Figura 2.84: Tests de normalidad sobre F8

El test de Lilliefors nos indica que no podemos rechazar la hipótesis de normalidad. Confirmamos este resultado con el gráfico QQ:

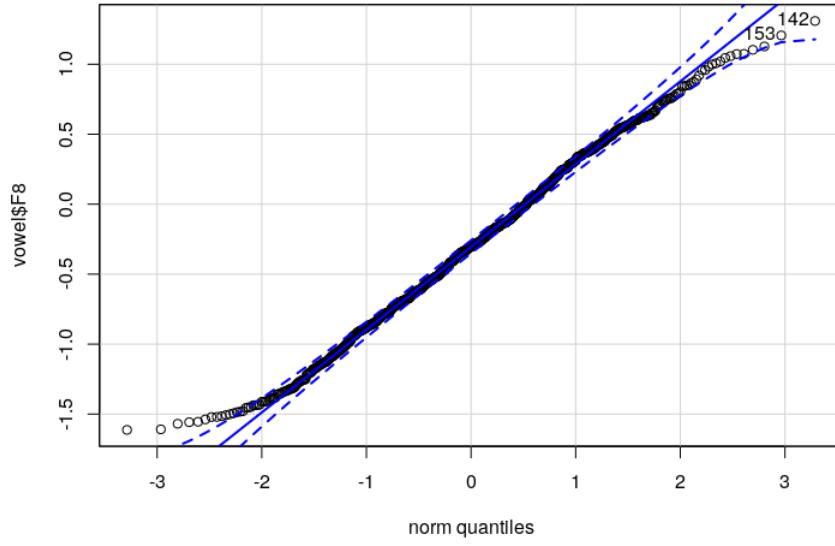


Figura 2.85: QQPlot de la variable F8

Si estudiamos los boxplots, vemos que la mayoría de los hombres tienen valores mayores que las mujeres. Además, encontramos ciertos outliers en los interlocutores 1, 7 y 12.

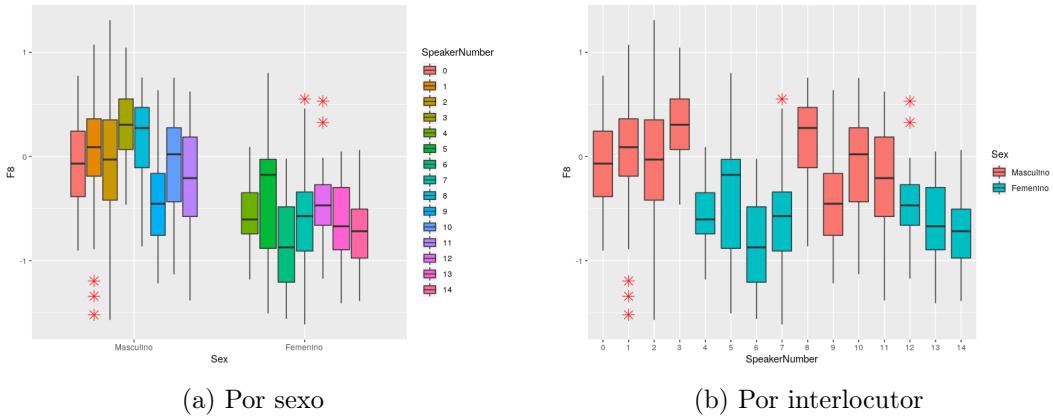


Figura 2.86: Boxplot para F8 estudiando sexos e interlocutores

Si estudiamos ambos性 por separado, en primer lugar los hombres,

```

> shapiro.test(hombres$F8)
Shapiro-Wilk normality test
data: hombres$F8
W = 0.99001, p-value = 0.00119
(a) Shapiro-Wilk

> lillie.test(hombres$F8)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F8
D = 0.045197, p-value = 0.01197
(b) Lilliefors

```

Figura 2.87: Tests de normalidad sobre F8 (hombres)

vemos que debemos rechazar la hipótesis de normalidad. Para las mujeres,

```

> shapiro.test(mujeres$F8)
Shapiro-Wilk normality test
data: mujeres$F8
W = 0.99008, p-value = 0.003301
(a) Shapiro-Wilk

> lillie.test(mujeres$F8)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F8
D = 0.037778, p-value = 0.1125
(b) Lilliefors

```

Figura 2.88: Tests de normalidad sobre F8 (mujeres)

el test de Shapiro-Wilk nos invita a rechazar la hipótesis de normalidad, mientras que el de Lilliefors no encuentra suficientes motivos para rechazar la hipótesis de normalidad ( $p$ -valor  $>0.05$ ). Confirmamos ese resultado con el gráfico QQ:

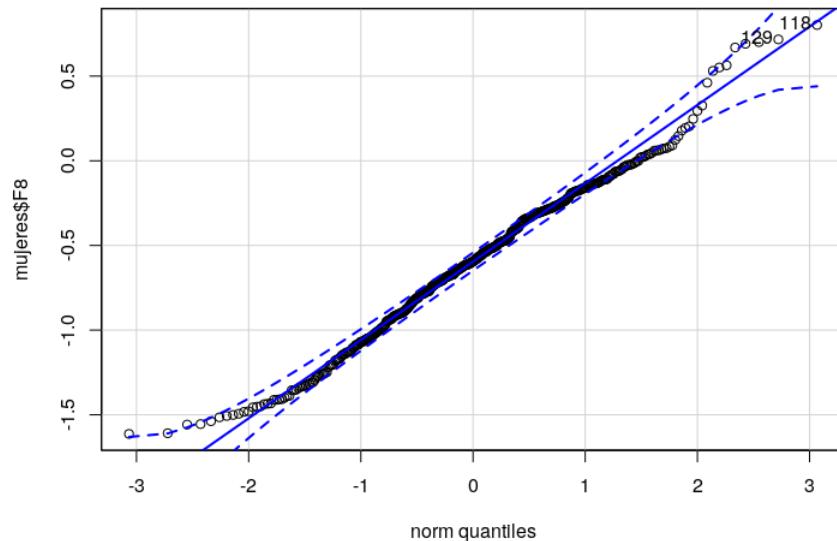


Figura 2.89: QQPlot de la variable F8 (mujeres)

donde vemos que la variable siempre se queda entre los márgenes establecidos para un comportamiento "normal". Vemos de nuevo una discrepancia entre el subconjunto de

interlocutores masculinos y femeninos.

### 2.2.10. F9

Presentamos el resumen estadístico de la última variable, F9, junto con su histograma:

- Media: -0.07134
- Mediana: -0.1565
- Desviación típica: 0.6039855
- Rango: [-1.68,1.396]
- Primer y tercer cuartiles: (-0.548,0.371)
- Asimetría: 0.294874
- Curtosis: -0.7644792

vemos que, a la luz de los resultados, existe una asimetría a la derecha de la distribución, que además es platicúrtica. Si observamos el histograma

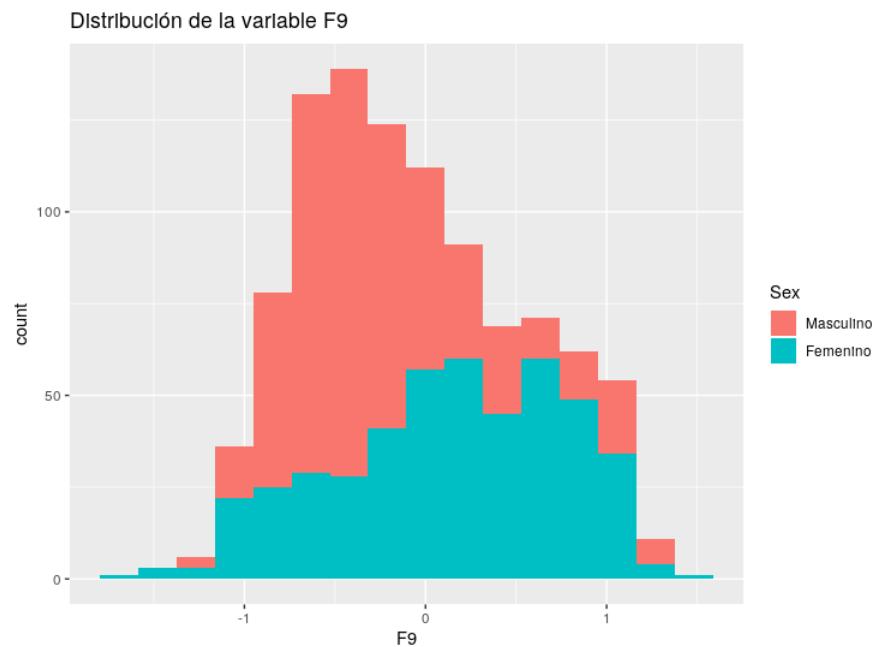


Figura 2.90: Histograma de la variable F9

como se puede ver, en la distribución de hombres existe una asimetría por la derecha (0.9905848) y para las mujeres, uno por la izquierda (-0.3907809), por lo que vemos la

gran disparidad entre ambas distribuciones.

Comprobamos ahora la normalidad de la distribución:

```
> shapiro.test(vowel$F9)
Shapiro-Wilk normality test
data: vowel$F9
W = 0.97377, p-value = 2.208e-12
(a) Shapiro-Wilk
```

```
> lillie.test(vowel$F9)
Lilliefors (Kolmogorov-Smirnov) normality test
data: vowel$F9
D = 0.073751, p-value = 7.729e-14
(b) Lilliefors
```

Figura 2.91: Tests de normalidad sobre F9

Ambos tests nos indican que debemos rechazar la hipótesis de normalidad. Estudiamos ahora los boxplots.

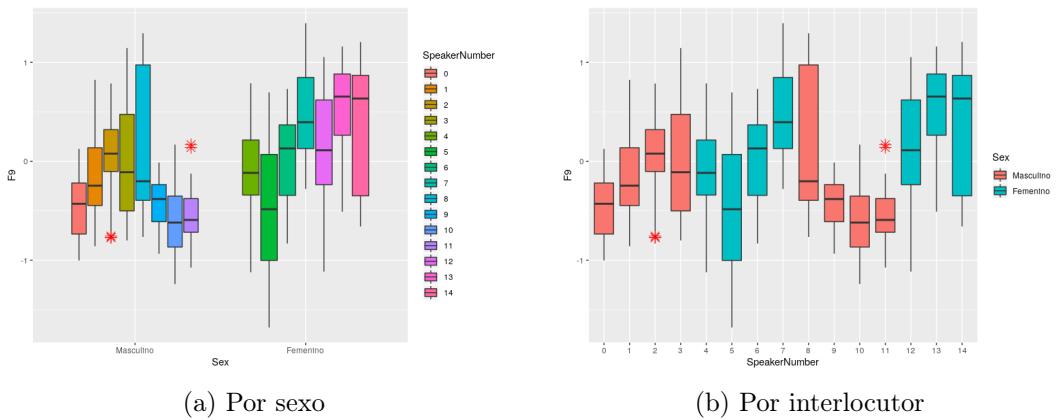


Figura 2.92: Boxplot para F9 estudiando sexos e interlocutores

vemos apenas presencia de outliers aunque sí resultados muy dispares entre hombres y mujeres, e incluso dentro de los mismos sexos.

Si estudiamos la distribución de los hombres

```
> shapiro.test(hombres$F9)
Shapiro-Wilk normality test
data: hombres$F9
W = 0.9251, p-value = 1.466e-15
(a) Shapiro-Wilk
```

```
> lillie.test(hombres$F9)
Lilliefors (Kolmogorov-Smirnov) normality test
data: hombres$F9
D = 0.12048, p-value < 2.2e-16
(b) Lilliefors
```

Figura 2.93: Tests de normalidad sobre F9 (hombres)

vemos que debemos rechazar la hipótesis de normalidad. Por parte de las mujeres

```
> shapiro.test(mujeres$F9)
Shapiro-Wilk normality test
data: mujeres$F9
W = 0.97182, p-value = 9.116e-08
```

(a) Shapiro-Wilk

```
> lilliefors.test(mujeres$F9)
Lilliefors (Kolmogorov-Smirnov) normality test
data: mujeres$F9
D = 0.073852, p-value = 2.564e-08
```

(b) Lilliefors

Figura 2.94: Tests de normalidad sobre F9 (mujeres)

también rechazamos la hipótesis de normalidad.

### 2.2.11. Correlación entre variables

Para estudiar la correlación entre las variables, en primer lugar, hago la representación de todas las variables, tomadas dos a dos, mostrando debajo de la diagonal, el scatter plot de las variables con un ajuste lineal. Por encima de la diagonal, vemos el valor de correlación de dichas variables con el nivel de significancia vía estrellas ( $p$ -valores(0, 0.001, 0.01, 0.05, 0.1, 1)  $\Leftrightarrow$  símbolos("\*\*\*\*", "\*\*\*", "\*\*", "\*", ".", "")).

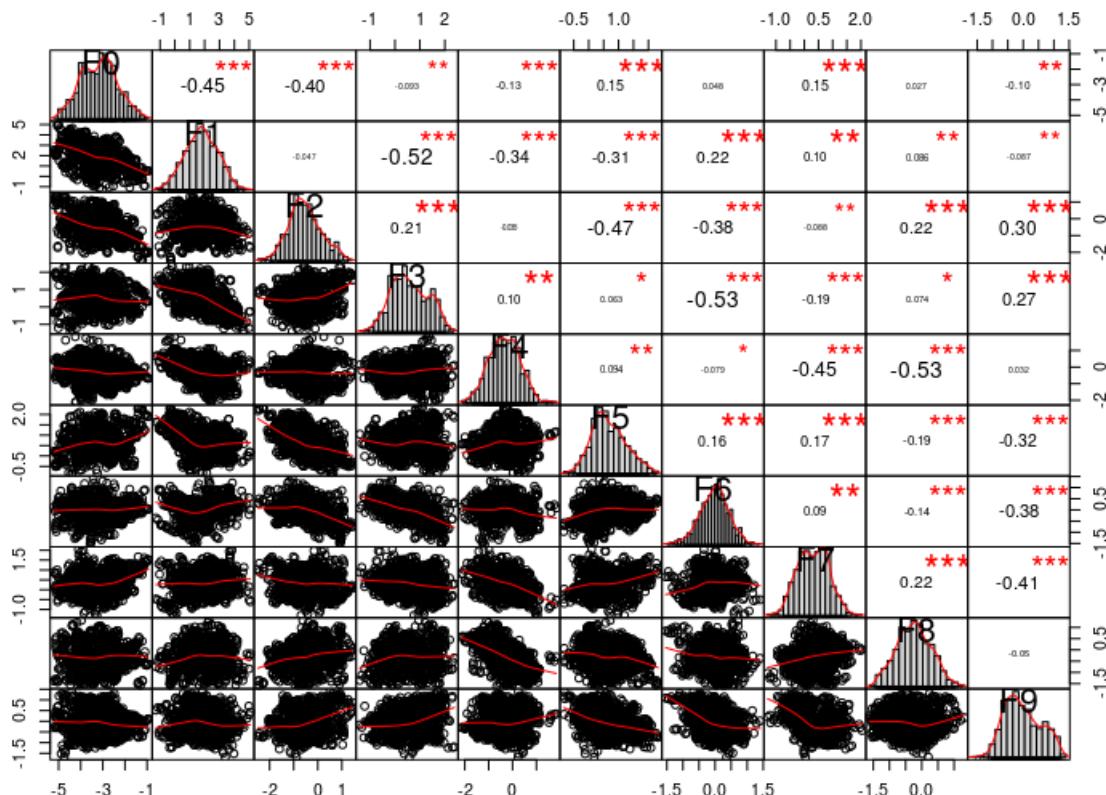


Figura 2.95: Comparativa de las variables dos a dos

Como podemos ver, la máxima correlación (negativa) es -0.53 entre las variables F3-F6 y F4-F8. Las demás son tendencias vagas.

Visto de otra manera, tenemos este gráfico

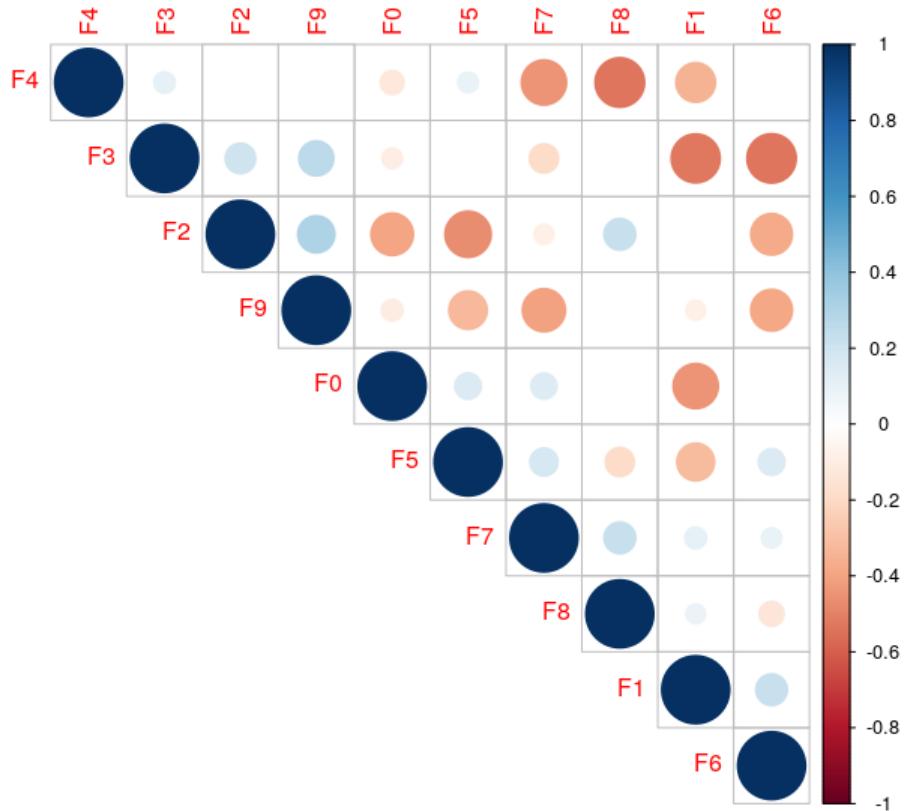


Figura 2.96: Comparativa de las variables dos a dos

Si ahora introducimos la distinción entre hombres y mujeres, obtenemos los siguientes resultados:

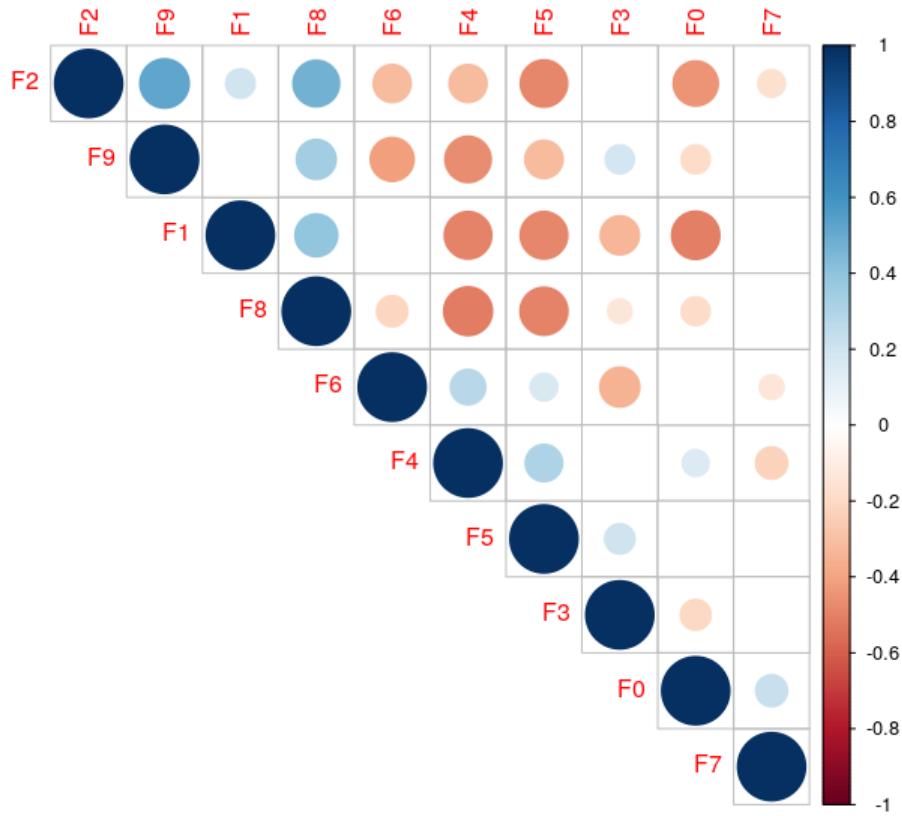


Figura 2.97: Comparativa de las variables dos a dos (hombres)

donde aparecen nuevas correlaciones (positivas) entre F2-F9, F2-F8, F1-F8 y nuevas negativas F2-F5, F2-F0, F1-F4, F1-F5, F1-F0, F8-F4 o F8-F5 con valores cercanos a  $\pm 0,48$ .

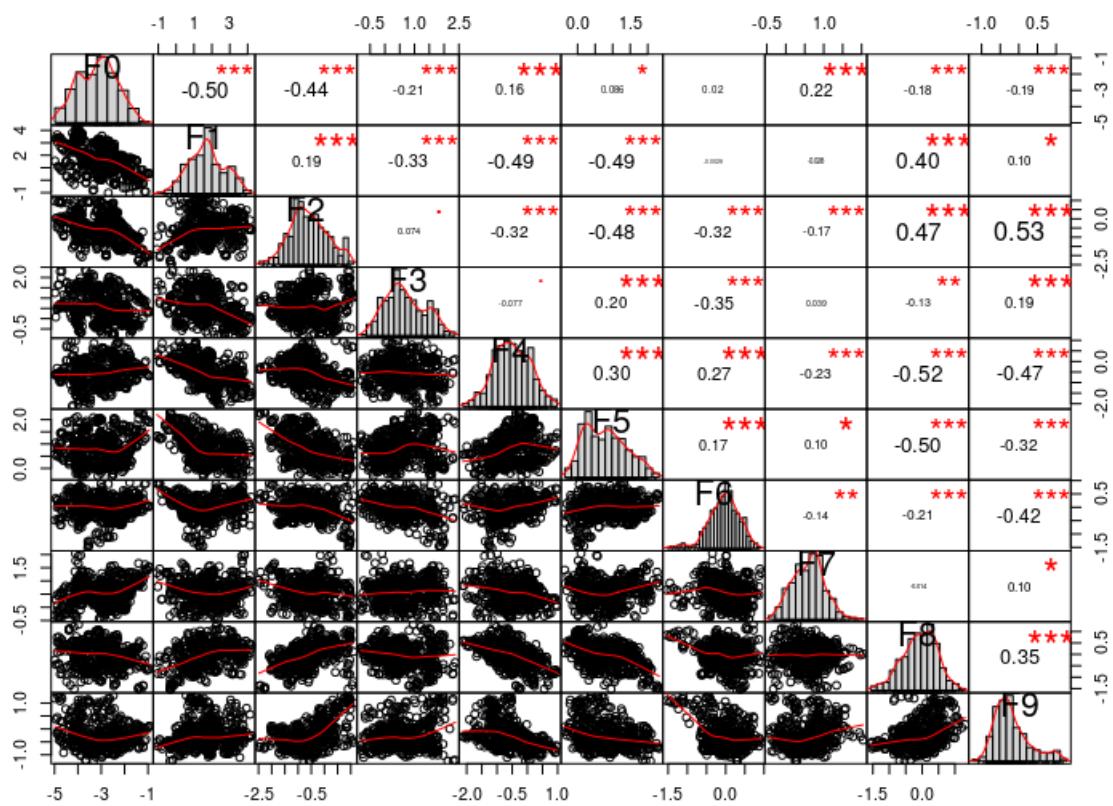


Figura 2.98: Comparativa de las variables dos a dos (hombres)

Para las mujeres

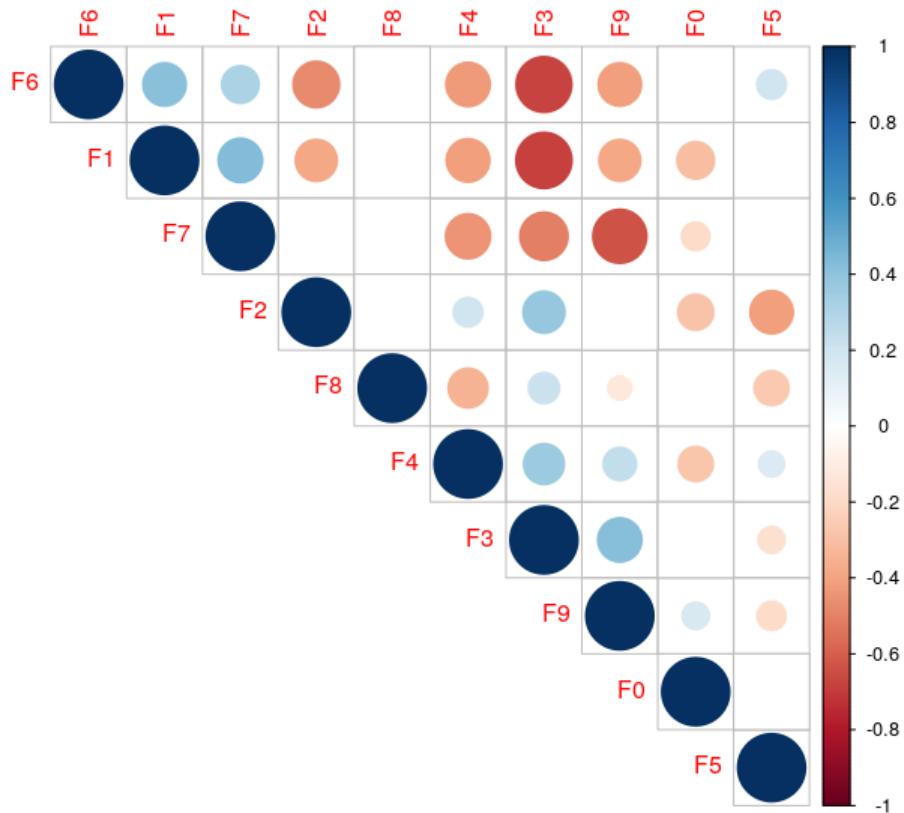


Figura 2.99: Comparativa de las variables dos a dos (mujeres)

Por tanto, vemos una vez más que es útil hacer diferenciación por sexos para encontrar correlaciones entre variables.

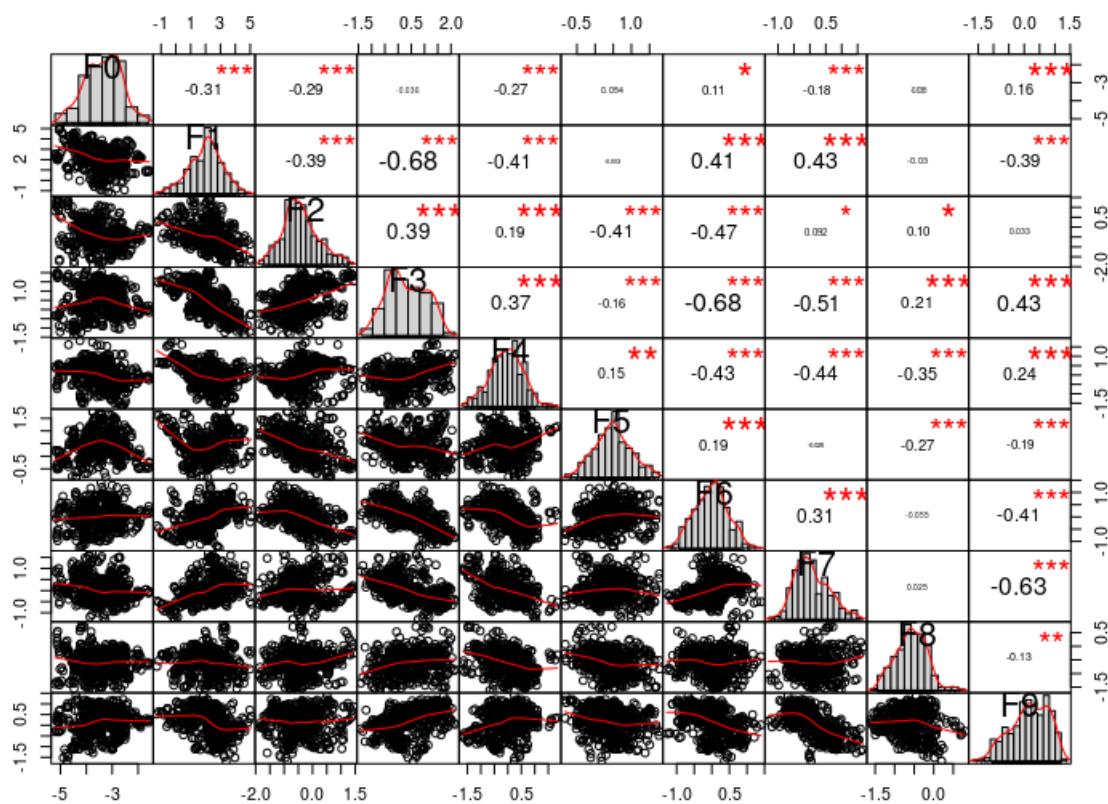


Figura 2.100: Comparativa de las variables dos a dos (mujeres)

encontramos grandes correlaciones negativas entre F6-F3 (-0.68), F1-F3 (-0.68) o F7-F9 (-0.63).

### 2.2.12. Conclusiones

El conjunto de datos Vowel, visto en su totalidad, parece tener poco sentido desde el punto de vista estadístico, dando lugar a variables sin mucha relación entre ellas. Sin embargo, la existencia de la variable Sexo genera dos nuevos conjuntos de datos independientes donde sí encontramos variables correladas y distribuidas según una normal. Por otra parte, cada variable tiene un rango y un dominio distinto, por lo que es necesario hacer un reescalado para que todas estén en el mismo rango de valores. A partir de ahí, debido a la asimetría y curtosis de cada característica, podría ser posible aplicar transformaciones de tipo raíz cuadrada o logaritmo para así conseguir más normalidad en las variables.

Por otra parte, encontramos patrones de outliers en ciertos interlocutores. Es el caso del interlocutor 2, que presenta en las variables F3, F4, F6, F7 y F9; o el 11, en la F2,F5,F7 y F9. Podríamos llegar a pensar que, si cada variable es una medida de parámetros o

características lingüísticas, esos interlocutores pueden tener una forma propia de hablar o algún problema de dicción.

Como se ha podido ver durante este desarrollo, las variables numéricas que tenemos no tienen ningún tipo de explicación ni justificación. Parecieran, a priori, valores descontextualizados, probablemente fruto de transformaciones matemáticas o salidas de instrumentos de medición como micrófonos o amplificadores. En consecuencia, no he podido establecer ninguna suerte de hipótesis, como sí hicimos en regresión. Para terminar el análisis exploratorio de datos, me gustaría enfatizar qué variables son interesantes para marcar fronteras de decisión entre las distintas clases de nuestro problema. Como viene siendo habitual, estudiar esto con el conjunto de datos completo no tiene unos resultados muy concluyentes. Sin embargo, si volvemos a separar por sexos, vemos que algunas variables pueden ser claramente determinantes en la distinción entre clases, dando que sus rangos de valores son disjuntos. Veámoslo con más detalle mediante boxplots:

■ **Hombres.**

- F0 y F1.

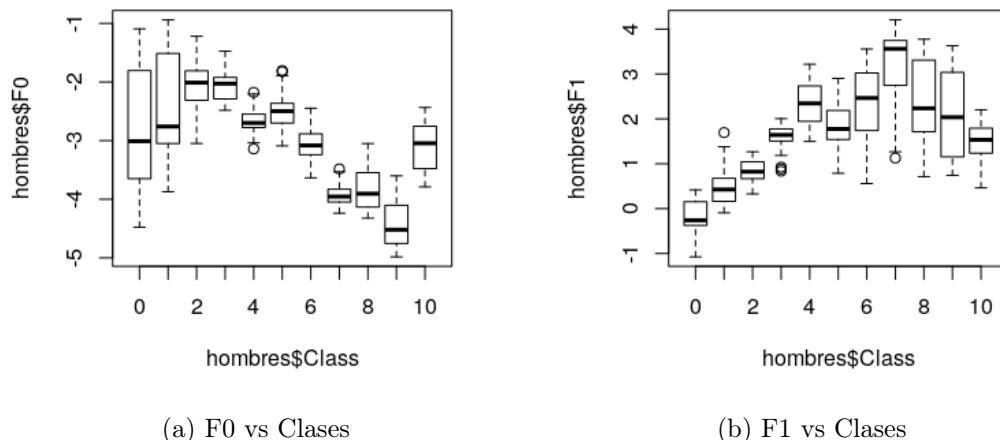


Figura 2.101: Boxplot para F0 y F1

Podemos ver que la variable F0 es útil para separar las clases 3,6,8 o 9 mientras que puede ser confusa para la 0 o la 1 porque sus valores se solapan en exceso. Además, apenas presenta outliers, centrados en las clases 4,5 y 7. En el caso de F1, se pueden separar fácilmente las clases 0, 2, 4 o 10, mientras que la 8 y 9 se solapan mucho y la 3 presenta bastantes outliers en el rango de la clase 2, por lo que puede ser confusa.

- F2 y F3.

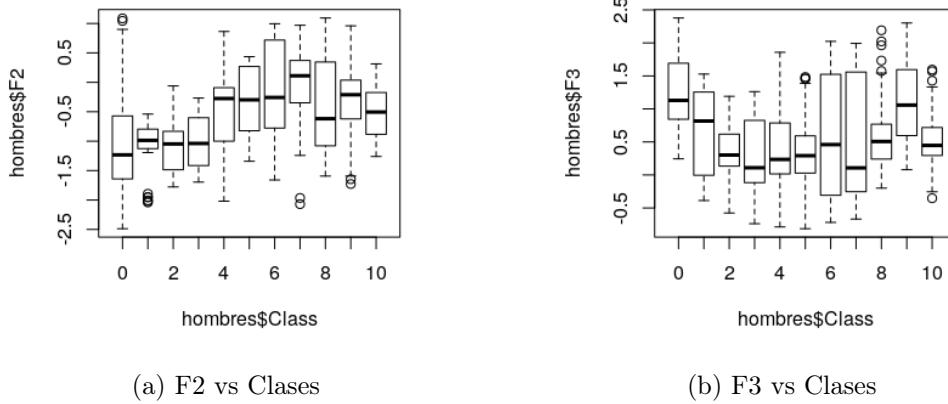


Figura 2.102: Boxplot para F2 y F3

En el caso de la variable F2, se podrían diferenciar las clases 2 o 3 de la 7 fácilmente (a pesar de los outliers para la variable 7). Las demás presentan algunas diferencias pero todas se centran más o menos en un rango de una unidad. Para F3, la clase 0 y 2 se separan muy bien, de la misma forma que la 0 y la 4. 0 y 8 no por la gran cantidad de outliers en la clase 8. Por el contrario, vemos como las clases 6 y 7 tienen valores prácticamente iguales en esta variable y, como el rango es amplio, engloban muchas otras clases menores en el mismo.

- F4 y F5.

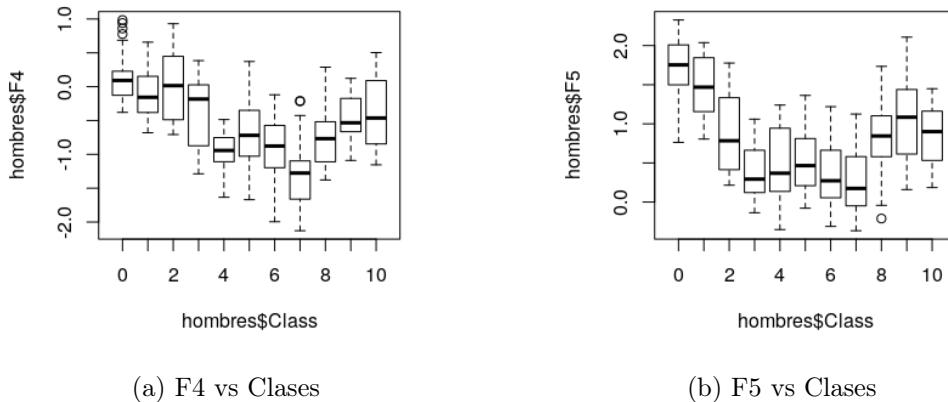


Figura 2.103: Boxplot para F4 y F5

Para la variable F4, las clases 1 y 4, 1 y 8 se diferencian bien, al igual que la 4 y la 9. La 7 es la más distante a todas a pesar del outlier que presenta, aunque todas las clases se encuentran en un rango bastante reducido. Por su parte, la variable F5 parece ser estupenda para separar. Como vemos, hay dos franjas de clases, las que toman valores aproximadamente por encima de 1.0 y las que toman valores menores. Por tanto, se presentan buenas oportunidades de encontrar diferencias, por ejemplo, entre las clases 1 y 7.

- F6 y F7.

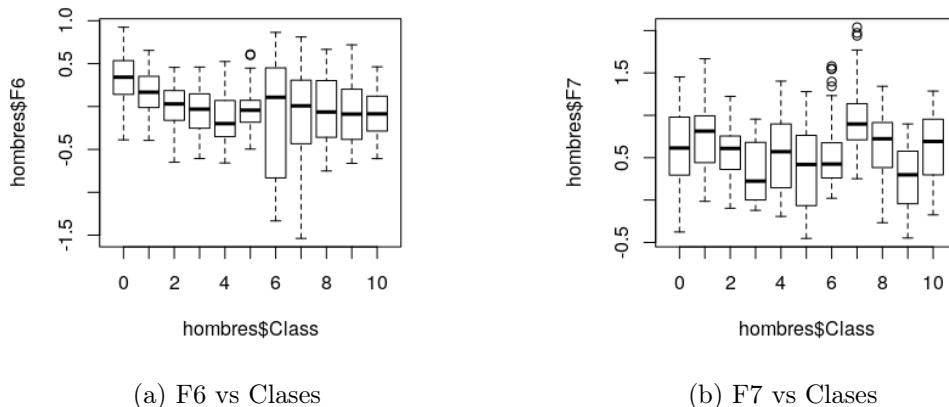
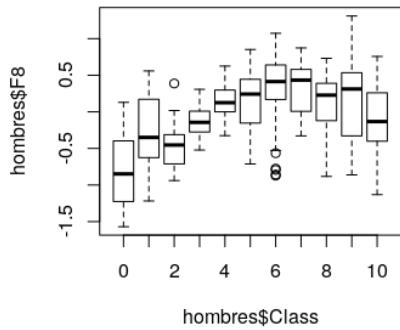


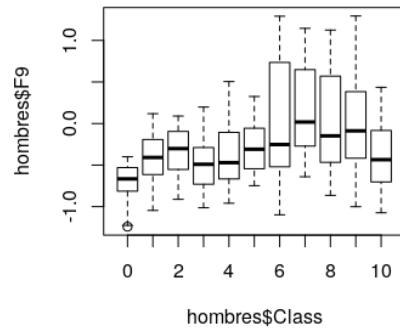
Figura 2.104: Boxplot para F6 y F7

Como se puede observar, la variable F6 tiene un rango de valores muy parejo para las variables. Tan sólo podemos saber que la clase 6 es la única que toma valores por debajo de -0.5, por lo que podría ser una posible distinción. F7, del mismo modo, apenas tiene diferencias entre las clases. Hay que subrayar la presencia de outliers para la clase 6 y la 7, lo que dificulta aún más la separación.

- F8 y F9.



(a) F8 vs Clases



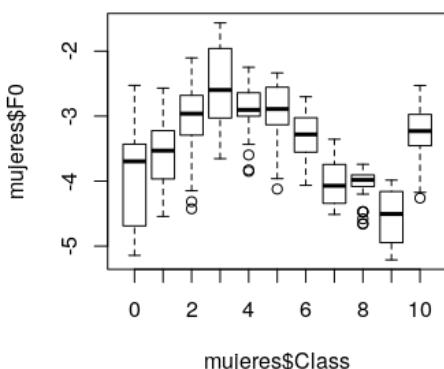
(b) F9 vs Clases

Figura 2.105: Boxplot para F8 y F9

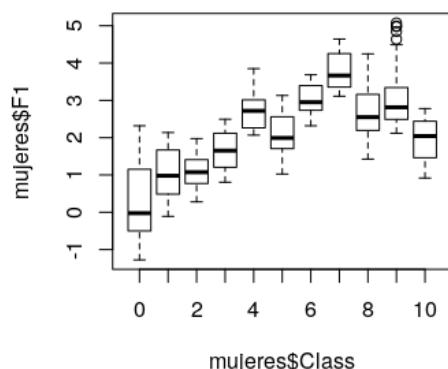
La variable F8, por el contrario, es estupenda para separar entre sí las clases 2,3 y 4; 1,3,4; 3 y 7; 3 y 8. Otras se solapan, como la 8, 9 y 10 o la 4 y 5. Por el contrario, la variable F9 nos lo pone más difícil, con solapamiento en prácticamente todas las clases a excepción de la 3 y 7 o 9, cuya separación podría ser de utilidad.

#### ■ Mujeres.

- F0 y F1



(a) F0 vs Clases



(b) F1 vs Clases

Figura 2.106: Boxplot para F0 y F1

Para la variable F0, las clases 0 y 3, 0 y 4, 0 y 5, 0 y 6, 1 y 2, 1 y 4, 1 y 5 u 9 y 10 son fácilmente separables. Por otro lado, las clases 2,3,4 y 5, 6 y 10, 7 y 8 (a pesar de outliers) se solapan. F1, por el contrario, es una estupenda variable para separar clases como 1 y 4,5,6,7,8 o 10.

- F2 y F3.

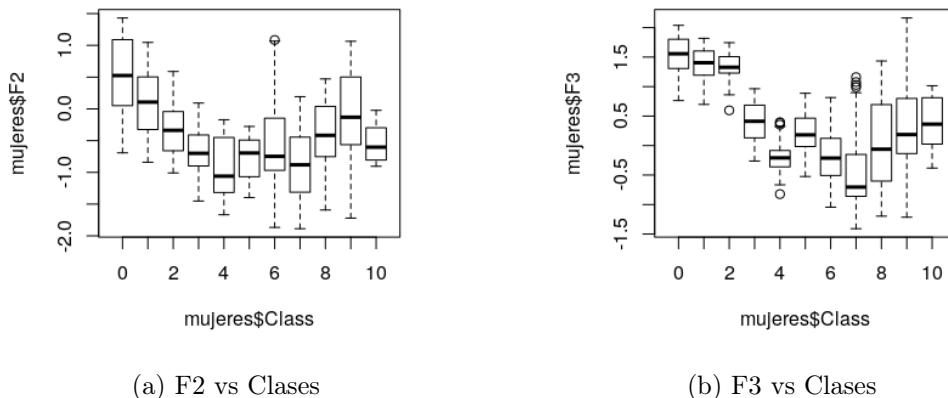


Figura 2.107: Boxplot para F2 y F3

En la variable F2, los intervalos para cada clase van disminuyendo muy progresivamente, por lo que separa las clases 0 con 2,3,4,5,6 (salvo el outlier), 7 y 10. Por ejemplo, las clases 3,4,5,6 y 7 se solapan, por lo que esta variable no es conveniente para separarlas entre sí. En la variable F3, las clases 0, 1 y 2 se solapan en valores, al igual que las 8, 9 y 10. Sin embargo, las tres primeras están en el entorno de 1.5 y las demás toman valores menores, por lo que son separables por pares salvo con la clase 7, cuya abundante cantidad de outliers hace que pudiera confundirse con las clases 0,1 o 2.

- F4 y F5.

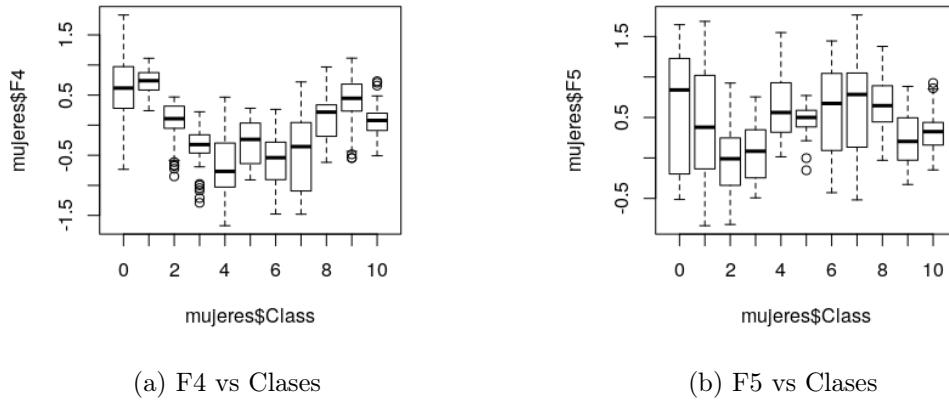


Figura 2.108: Boxplot para F4 y F5

La variable F4, a pesar de concentrar gran número de outliers en las clases 2, 3, 9 y 10, puede resultar muy útil para separar las clases 0 con 4 ,5, 6,7 y 8, al igual que 1 con 4 ,5, 6,7 y 8. F5, por su parte, podría servir para separar las clases 2 con 4 y 8, al igual que la 3. Sin embargo, la gran mayoría de las clases se solapan en los valores por ser una variable muy concentrada (2 unidades).

- F6 y F7.

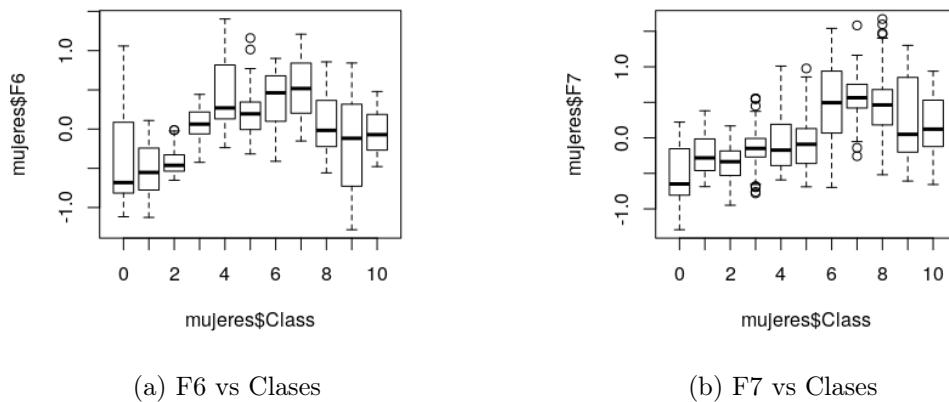


Figura 2.109: Boxplot para F6 y F7

El variable F6, la clase 1 se separa de todas excepto de la 0 y la 2. Sin embargo,

el resto se solapan. En la variable 7, encontramos un grupo de clases que se agolpan bajo el 0.0 (de la 0 a la 5) y otras que están por encima de 0.0 (6 a 10), por lo que son fácilmente separables entre grupos y se solapan entre grupos.

- F8 y F9.

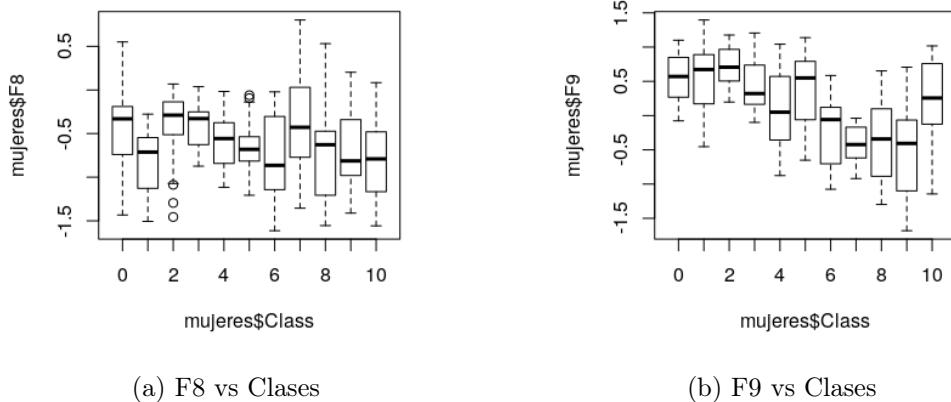


Figura 2.110: Boxplot para F8 y F9

La variable F8 no es buena para separar ninguna clase en especial, dado que se solapan prácticamente los valores entorno al 0.5. Por el contrario, la variable F9 separa bien las clases 0,1,2,3,4 y 6,7,8 y 9. 0,1,2,3,4,5 y 10 se solapan, al igual que 6,7,8 y 9.

### 3. Problema de regresión: Wankara

Tras realizar el análisis exploratorio de datos sobre el conjunto de datos Wankara, estamos en condiciones de abordar el problema de regresión, es decir, encontrar y ajustar un modelo que prediga bien la variable Mean temperature. Dicho trabajo estará dividido en cuatro secciones, que coinciden con los puntos exigidos en el trabajo. El primero tratará de ajustar y comparar modelos lineales simples con los 5 regresores más interesantes. El segundo, crear modelos lineales más complejos combinando distintos regresores, utilizando interacciones y combinaciones no lineales. En la tercera, ajustamos un modelo kNN y en el último hacemos una comparación de los algoritmos incorporando los resultados obtenidos en la fila de Wankara.

#### 3.1. Regresión lineal simple sobre 5 regresores

Como Wankara tiene más de 5 variables, elijo las 5 más representativas para la predicción de Mean temperature. En otras palabras, aquellas que tienen una correlación mayor (vista en el EDA): Max-temperature, Min-temperature, Dewpoint, Sea-level-pressure y Visibility. Antes de aplicar ajustar cualquier modelo, reescalo las variables de Wankara para que todas estén en el mismo rango (y así evitar sesgos hacia las que tienen un rango de valores mayor). Para todas ellas, el procedimiento será el mismo. Entreno un modelo lineal con todo el conjunto a través del regresor elegido, muestro la recta obtenida y el error cuadrático medio. Para validar estos resultados, llevo a cabo una validación cruzada con 5 particiones y calculo el error cuadrático medio en training y test.

##### 3.1.1. Modelo con Max-temperature

El modelo generado es el siguiente:

```
lm(formula = wankara_scale$Mean_temperature ~ wankara_scale$Max_temperature)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.186200 -0.028390  0.006092  0.034599  0.112920 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.124945   0.002893  43.18 <2e-16 ***
wankara_scale$Max_temperature 0.876230   0.005243 167.13 <2e-16 ***
...
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0488 on 1607 degrees of freedom
Multiple R-squared:  0.9456,    Adjusted R-squared:  0.9456 
F-statistic: 2.793e+04 on 1 and 1607 DF,  p-value: < 2.2e-16
```

Figura 3.1: Modelo lineal con Max-temperature

Con un  $R^2$  ajustado de 0.9456 y un error cuadrático medio de 0,04876785. El siguiente scatter plot muestra el resultado del ajuste:

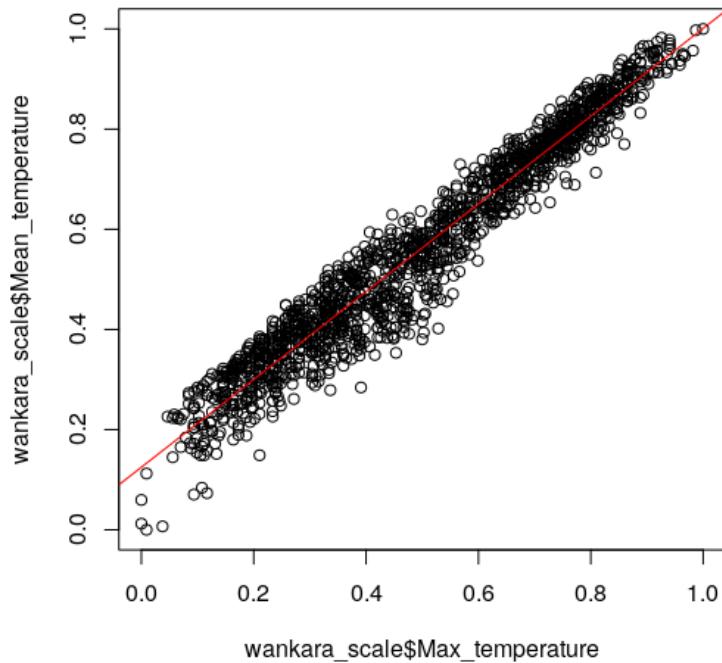


Figura 3.2: Modelo lineal con Max-temperature. Scatter plot

Tras realizar la validación cruzada, obtenemos un error cuadrático medio en training de 12.98672 y en test de 13.00239.

### 3.1.2. Modelo lineal con Min-temperature

Entrenamos un segundo modelo lineal con la variable Min-temperature. Este es el resultado:

```

lm(formula = wankara_scale$Mean_temperature ~ wankara_scale$Min_temperature)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.152177 -0.055621 -0.002217  0.054211  0.243358 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.084865   0.006312 -13.45 <2e-16 ***
wankara_scale$Min_temperature 1.065912   0.009930 107.34 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0732 on 1607 degrees of freedom
Multiple R-squared:  0.8776,    Adjusted R-squared:  0.8775 
F-statistic: 1.152e+04 on 1 and 1607 DF,  p-value: < 2.2e-16

```

Figura 3.3: Modelo lineal con Min-temperature

que nos deja un  $R^2$  ajustado de 0.8775 y un error cuadrático medio de 0,07315107. Podemos verlo gráficamente como sigue.

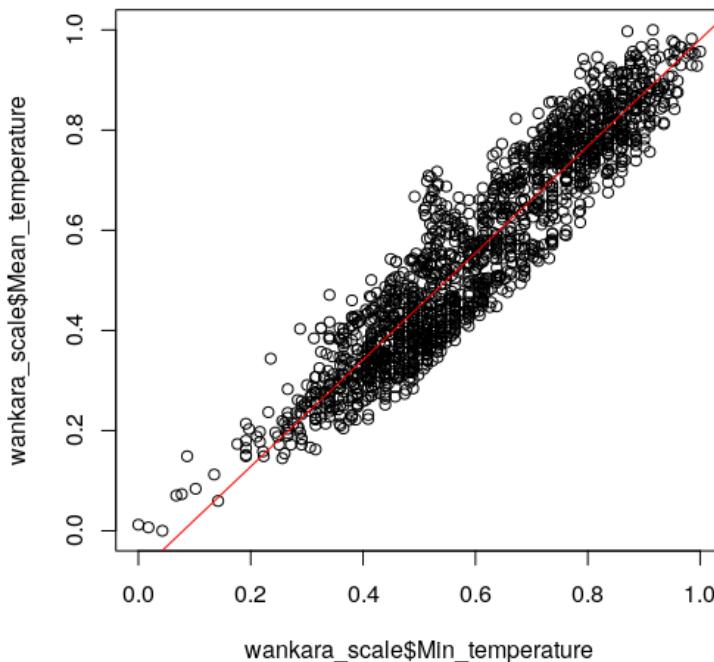


Figura 3.4: Modelo lineal con Min-temperature. Scatter plot

Podemos ver que el ajuste no es tan bueno como el anterior, dado que la nube de puntos es más ancha y se separa algo más de la recta obtenida. Tras la validación cruzada de 5 particiones, obtenemos un error cuadrático medio en training de 29,21725 y en test de 29,27856. La diferencia tan pequeña del error en entrenamiento y test indica que no existe sobreajuste y la generalización ha sido buena.

### 3.1.3. Modelo lineal con Dewpoint

Entrenamos un tercer modelo lineal con Dewpoint. Este es el resultado:

```
-----
lm(formula = wankara_scale$Mean_temperature ~ wankara_scale$Dewpoint)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.17516 -0.06869 -0.01150  0.05248  0.41776 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.118867   0.008729 -13.62   <2e-16 ***
wankara_scale$Dewpoint 1.051895   0.012971  81.09   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09271 on 1607 degrees of freedom
Multiple R-squared:  0.8036,    Adjusted R-squared:  0.8035 
F-statistic:  6576 on 1 and 1607 DF,  p-value: < 2.2e-16
```

Figura 3.5: Modelo lineal con Dewpoint

con un  $R^2$  ajustado de 0,8035 y un error cuadrático medio de 0,09265712. Gráficamente vemos cómo se ajusta esta recta a los puntos:

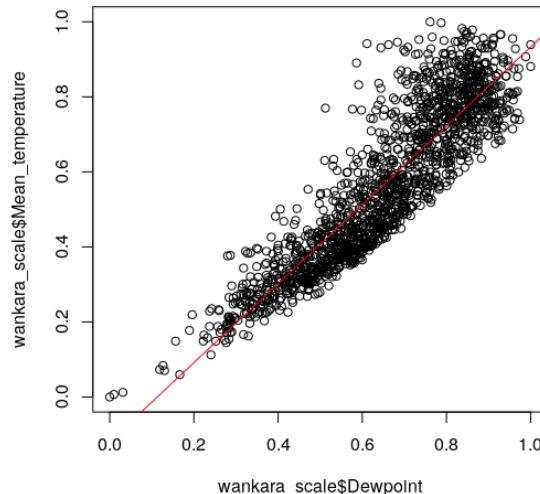


Figura 3.6: Modelo lineal con Dewpoint. Scatter plot

El ajuste va bajando en calidad conforme pasamos avanzamos en los modelos. Tras la validación cruzada de 5 particiones, obtenemos un error cuadrático medio en training de 46,88105 y en test de 46,93901, con pequeñas diferencias entre ellos también.

### 3.1.4. Modelo lineal con Sea level pressure

Cuarto modelo lineal, ahora con la variable Sea level pressure. El resultado es el siguiente:

```
lm(formula = wankara_scale$Mean_temperature ~ wankara_scale$Sea_level_pressure)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.57841 -0.11653  0.00861  0.12990  0.36359 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.90892   0.01103  82.38 <2e-16 ***
wankara_scale$Sea_level_pressure -0.75969   0.02263 -33.57 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1604 on 1607 degrees of freedom
Multiple R-squared:  0.4122, Adjusted R-squared:  0.4118 
F-statistic: 1127 on 1 and 1607 DF, p-value: < 2.2e-16
```

Figura 3.7: Modelo lineal con Sea level pressure

con un  $R^2$  ajustado de 0,4118 y un error cuadrático medio de 0,1603035. Gráficamente vemos cómo se ajusta esta recta a los puntos:

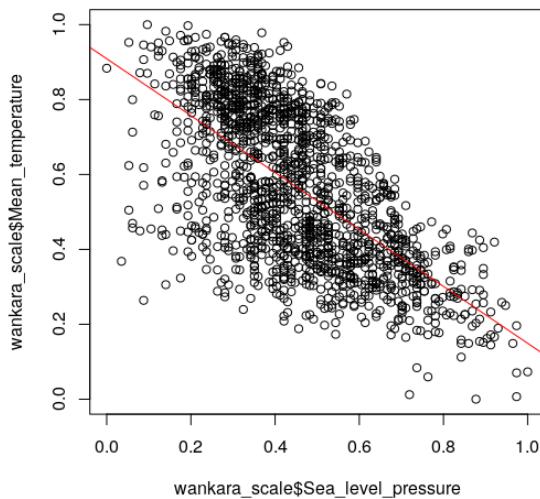


Figura 3.8: Modelo lineal con Sea level pressure. Scatter plot

El ajuste ha empeorado mucho con esta variable. Tras la validación cruzada de 5 particio-

nes, obtenemos un error cuadrático medio en training de 140,3178 y en test de 140,5083

### 3.1.5. Modelo lineal con Visibility

Último modelo lineal con una variable, en este caso Visibility. Este es el resultado:

```
lm(formula = wankara_scale$Mean_temperature ~ wankara_scale$Visibility)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.52820 -0.12295  0.00608  0.14651  0.37585 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.04842   0.02364   2.048   0.0407 *  
wankara_scale$Visibility 0.77450   0.03486  22.217 <2e-16 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 0.183 on 1607 degrees of freedom
Multiple R-squared:  0.235,   Adjusted R-squared:  0.2345 
F-statistic: 493.6 on 1 and 1607 DF,  p-value: < 2.2e-16
```

Figura 3.9: Modelo lineal con Visibility

con un  $R^2$  ajustado de 0,2345 y un error cuadrático medio de 0,1828792. Gráficamente vemos cómo se ajusta esta recta a los puntos:

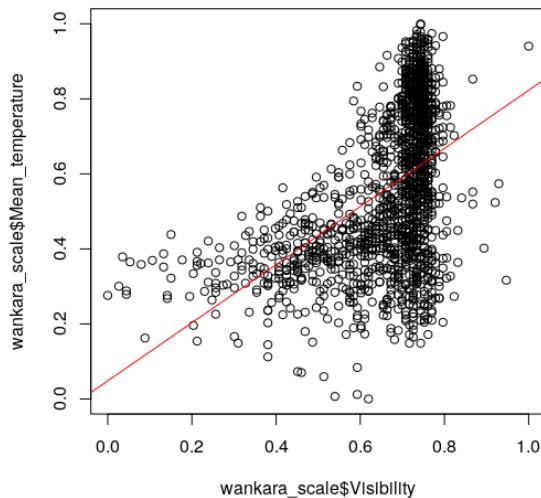


Figura 3.10: Modelo lineal con Visibility. Scatter plot

Tras la validación cruzada de 5 particiones, obtenemos un error cuadrático medio en training de 182,5684 y en test de 183,3558

### 3.1.6. Conclusiones

Partimos de una tabla resumen para sacar las conclusiones del estudio anterior.

	$R^2$ ajustado	MSE	CV 5fold MSE train(media)	CV 5fold MSE test(media)
Max-temperature	0.9456	0.04876785	12.98670	13.00239
Min-temperature	0.8775	0.07315107	29.21725	29.27856
Dewpoint	0.8035	0.09265712	46.88105	46.93901
Sea level pressure	0.4118	0.1603035	140.3178	140.5083
Visibility	0.2345	0.1828792	182.5684	183.3558

Cuadro 3.1: Resumen de resultados para modelos lineales simples

Como elegimos los regresores en función de la correlación, es evidente que aquellos que están más correlacionados con Mean temperature obtienen valores de  $R^2$  ajustado más alto y MSE más pequeño. Una gran diferencia entre el MSE sobre todo el conjunto de datos y la media en training y test de la validación cruzada, la cual se justifica debido a la cantidad de datos que tiene un modelo y otro: mientras que el primero se realiza con todos los datos, lo que le da un gran poder de aprendizaje, el segundo se hace sobre un quinto de los datos cada vez y luego se calcula la media de los resultados. Aún así, vemos que la diferencia entre entrenamiento y test es muy pequeña, lo que nos indica que nuestro modelo está generalizando correctamente y no hay sobreajuste. Por supuesto, en virtud del valor de  $R^2$  ajustado y los MSE, el modelo lineal basado en Max temperature es el más potente para predecir el valor de Mean temperature.

## 3.2. Modelos lineales múltiples. Interacciones y no linealidad

Una vez explorados los 5 modelos lineales con un solo regresor. Enfrentamos ahora el problema con regresión múltiple. Dado que tenemos 9 regresores, por eficiencia elijo hacerlo con el llamado Backward Model, es decir, partiendo de todos los regresores e ir eliminando los regresores menos significativos. Nuestro objetivo es doble: el primero es encontrar un modelo competitivo y a la vez explicable (); el segundo, encontrar un modelo con el mayor  $R^2$  ajustado posible. Para el primer objetivo, utilizamos backward model; para el segundo, interacciones y no linealidad.

### 3.2.1. Hacia un modelo competitivo y explicable

Partimos de un modelo lineal con los 9 regresores. Obtenemos este resultado:

```

lm(formula = wankara_scale$Mean_temperature ~ ., data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.086300 -0.012697  0.000052  0.011892  0.092681 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.043892  0.007118  6.166 8.85e-10 ***
Max_temperature 0.551458  0.006128  89.986 < 2e-16 ***
Min_temperature 0.299042  0.009402  31.806 < 2e-16 ***
Dewpoint       0.088256  0.008725  10.115 < 2e-16 ***
Precipitation   -0.001728 0.011962  -0.144   0.885  
Sea_level_pressure -0.182720 0.021065  -8.674 < 2e-16 ***
Standard_pressure 0.131505  0.018582  7.077 2.20e-12 ***
Visibility       0.020033  0.005118  3.914 9.46e-05 ***
Wind_speed        0.039851  0.005005  7.962 3.19e-15 ***
Max_wind_speed   -0.044904  0.007533  -5.961 3.08e-09 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02116 on 1599 degrees of freedom
Multiple R-squared:  0.9898,    Adjusted R-squared:  0.9898 
F-statistic: 1.728e+04 on 9 and 1599 DF,  p-value: < 2.2e-16

```

Figura 3.11: Modelo lineal múltiple con todas las variables

Obtenemos un  $R^2$  ajustado de 0.9898 y 1599 grados de libertad. Mirando los p-valores de los regresores, vemos uno, el de Precipitation, de 0.885, por lo que no es relevante en el ajuste. Siendo así, la eliminamos y generamos un nuevo modelo:

```

lm(formula = wankara_scale$Mean_temperature ~ . - Precipitation,
   data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.086311 -0.012663  0.000074  0.011910  0.092612 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.043827  0.007102  6.171 8.58e-10 ***
Max_temperature 0.551600  0.006047  91.214 < 2e-16 ***
Min_temperature 0.299007  0.009396  31.823 < 2e-16 ***
Dewpoint       0.088146  0.008689  10.144 < 2e-16 ***
Sea_level_pressure -0.182682 0.021057  -8.676 < 2e-16 ***
Standard_pressure 0.131553  0.018574  7.083 2.11e-12 ***
Visibility       0.020055  0.005114  3.921 9.18e-05 ***
Wind_speed        0.039870  0.005002  7.971 2.97e-15 ***
Max_wind_speed   -0.044907  0.007531  -5.963 3.04e-09 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02115 on 1600 degrees of freedom
Multiple R-squared:  0.9898,    Adjusted R-squared:  0.9898 
F-statistic: 1.946e+04 on 8 and 1600 DF,  p-value: < 2.2e-16

```

Figura 3.12: Modelo lineal múltiple sin Precipitation

Como vemos, el error residual ha disminuido, el  $R^2$  ajustado se mantiene y ganamos un grado de libertad, 1600, por lo que ha sido acertado eliminar esa variable. Todos los p-valores ahora son bastante pequeños, así que eliminamos la variable que tenga mayor error estándar, es decir, Sea level pressure:

```
lm(formula = wankara_scale$Mean_temperature ~ . - Precipitation -
  Sea_level_pressure, data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.086797 -0.013069  0.000066  0.012413  0.096602 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.001135  0.004967 -0.229   0.819    
Max_temperature 0.579991  0.005202 111.494 < 2e-16 ***
Min_temperature 0.316297  0.009393 33.674 < 2e-16 ***
Dewpoint        0.113648  0.008364 13.587 < 2e-16 ***
Standard_pressure -0.026047  0.003962 -6.575 6.57e-11 ***
Visibility       0.027976  0.005148  5.435 6.34e-08 ***
Wind_speed       0.045098  0.005079  8.879 < 2e-16 ***
Max_wind_speed   -0.048592  0.007691 -6.318 3.43e-10 ***  
...
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02163 on 1601 degrees of freedom
Multiple R-squared:  0.9893,    Adjusted R-squared:  0.9893 
F-statistic: 2.124e+04 on 7 and 1601 DF,  p-value: < 2.2e-16
```

Figura 3.13: Modelo lineal múltiple sin Precipitation y Sea level pressure

Este modelo disminuye muy ligeramente el  $R^2$  ajustado (0.9893) y aumenta el error residual (0.02163). Sin embargo, la disminución es tan pequeña que merece la pena eliminar una variable y ganar interpretabilidad. Seguimos en la misma dinámica y elimino el regresor con un error estándar elevado (aunque no el mayor, que es Dewpoint), Max wind speed. No elimino Dewpoint porque antes vimos que está muy correlacionada con Mean temperature, lo que sospecho nos puede hacer bajar demasiado el  $R^2$  ajustado.

```

lm(formula = wankara_scale$Mean_temperature ~ . - Precipitation -
  Sea_level_pressure - Max_wind_speed, data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.087902 -0.013478  0.000054  0.012890  0.102808 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.007139  0.004934 -1.447   0.148    
Max_temperature 0.574872  0.005201 110.540 < 2e-16 ***
Min_temperature 0.326919  0.009353  34.955 < 2e-16 ***
Dewpoint       0.108008  0.008417  12.832 < 2e-16 ***
Standard_pressure -0.022336  0.003965 -5.633 2.09e-08 ***
Visibility      0.027941  0.005210   5.363 9.37e-08 ***
Wind_speed      0.025021  0.004010   6.239 5.62e-10 ***
...
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0219 on 1602 degrees of freedom
Multiple R-squared:  0.9891,    Adjusted R-squared:  0.989 
F-statistic: 2.419e+04 on 6 and 1602 DF,  p-value: < 2.2e-16

```

Figura 3.14: Modelo lineal múltiple sin Precipitation, Sea level pressure, Max wind speed

Como resultado de la eliminación de Max wind speed, disminuimos 3 diezmilésimas el  $R^2$  ajustado (0.98) y el error residual aumenta apenas nada. Visibility es la siguiente variable a eliminar, que tiene el mayor p-valor en general y mayor error estándar de entre las poco correladas:

```

lm(formula = wankara_scale$Mean_temperature ~ . - Precipitation -
  Sea_level_pressure - Max_wind_speed - Visibility, data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.095443 -0.013364  0.000901  0.012962  0.102783 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.008482  0.004017  2.111   0.0349 *  
Max_temperature 0.584282  0.004938 118.327 < 2e-16 ***
Min_temperature 0.330823  0.009405  35.176 < 2e-16 ***
Dewpoint       0.099526  0.008338  11.936 < 2e-16 ***
Standard_pressure -0.023620  0.003992 -5.917 4.01e-09 ***
Wind_speed      0.031967  0.003828   8.351 < 2e-16 ***
...
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02208 on 1603 degrees of freedom
Multiple R-squared:  0.9889,    Adjusted R-squared:  0.9889 
F-statistic: 2.852e+04 on 5 and 1603 DF,  p-value: < 2.2e-16

```

Figura 3.15: Modelo lineal múltiple sin Precipitation, Sea level pressure, Max wind speed, Visibility

Volvemos a perder una diezmilésima en el  $R^2$  ajustado (0.9889), aún sigue siendo un

resultado muy bueno (apenas hemos perdido una centésima) quitando cuatro regresores. La siguiente en eliminarse es

```
lm(formula = wankara_scale$Mean_temperature ~ . - Standard_pressure -
  Precipitation - Sea_level_pressure - Max_wind_speed - Visibility,
  data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.09881 -0.01350  0.00060  0.01349  0.10272 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.011021  0.002321 -4.749 2.22e-06 ***  
Max_temperature 0.579343  0.004918 117.797 < 2e-16 ***  
Min_temperature 0.331780  0.009502 34.915 < 2e-16 ***  
Dewpoint       0.110054  0.008232 13.369 < 2e-16 ***  
Wind_speed      0.037836  0.003736 10.127 < 2e-16 ***  
...
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 0.02232 on 1604 degrees of freedom
Multiple R-squared:  0.9886,   Adjusted R-squared:  0.9886 
F-statistic: 3.491e+04 on 4 and 1604 DF,  p-value: < 2.2e-16
```

Figura 3.16: Modelo lineal múltiple sin Precipitation, Sea level pressure, Max wind speed, Visibility, Standard Pressure

El resultado vuelve a ser favorable, 0.9886. Por último, para encontrar el modelo más interpretable, con las variables que más sentido podemos darle respecto de la temperatura media, eliminamos Wind speed.

```
lm(formula = wankara_scale$Mean_temperature ~ . - Wind_speed -
  Standard_pressure - Precipitation - Sea_level_pressure -
  Max_wind_speed - Visibility, data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.102928 -0.013515  0.000624  0.013409  0.113277 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.006704  0.002352  -2.85  0.00443 **  
Max_temperature 0.564568  0.004843 116.57 < 2e-16 ***  
Min_temperature 0.373246  0.008842 42.21 < 2e-16 ***  
Dewpoint       0.093391  0.008317 11.23 < 2e-16 ***  
...
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 0.02301 on 1605 degrees of freedom
Multiple R-squared:  0.9879,   Adjusted R-squared:  0.9879 
F-statistic: 4.374e+04 on 3 and 1605 DF,  p-value: < 2.2e-16
```

Figura 3.17: Modelo lineal múltiple más interpretable: Max temperature, Min temperature y Dewpoint

Hemos perdido un par de centésimas desde un modelo con nueve regresores a uno con 3, con  $R^2$  ajustado 0.9879, 1605 grados de libertad y un error residual de 0.02301. Además, esos tres regresores son semánticamente muy relevantes con el problema, ya que están basados en las hipótesis que comenté en el EDA e incluyen Dewpoint, una variable que también tiene que ver con la temperatura, para explicar la temperatura media. Por tanto, hemos cumplido nuestro primer objetivo en regresión lineal múltiple.

### 3.2.2. Modelo lineal múltiple óptimo: interacciones y no linealidad

El proceso de búsqueda del modelo lineal múltiple óptimo tiene mucho de eso, de proceso de búsqueda. Muchas veces la heurística y la intuición van de la mano hasta encontrar algo satisfactorio. Parece lógico pensar que hay que enfatizar aquellas variables que sean más importantes (las del modelo más interpretable anterior), eliminar las que apenas aportan nada y combinar las que pueden tener una relación desde el punto de vista semántico. He pasado por siete modelos (pueden verse en el código del apéndice II), líneas 480-506 con distintas aproximaciones. En el primero, busco la interacción entre Sea level pressure y Standard pressure y elimino Precipitation con resultado 0.9899. En el segundo, elimino Precipitation e interacciono Min-temperature y Dewpoint, con resultado 0.9899 (sin mejora). En el tercero, elimino Precipitation y Dewpoint (tienen un p-valor de 0.99 si lo mantengo), interacciono Dewpoint y Min-temperature y elevo al cuadrado Dewpoint con resultado 0.99. Observo que en el resultado del tercero, Visibility es poco representativo, por lo que en el cuarto mantengo lo anterior y elimino Visibility, con resultado 0.9899, empeorando. En el quinto, añado el término de Max-temperature al cuadrado y mantengo Visibility, con resultado 0.9918 (mejor hasta el momento). En el sexto, elevo al cuadrado también Min-temperature, con resultado 0.9923 (mejorando el anterior). Para evitar la redundancia por la interacción entre Max wind speed y Wind speed, elimino Max wind speed obteniendo el mismo 0.9923, como definitivo modelo óptimo. He aquí el resultado:

```

```
lm(formula = wankara_scale$Mean_temperature ~ . - Precipitation +
   Max_wind_speed * Wind_speed + I(Min_temperature^2) + I(Max_temperature^2) +
   Min_temperature * Dewpoint + I(Dewpoint^2) - Dewpoint - Max_wind_speed,
   data = wankara_scale)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.087131 -0.011086 -0.000113  0.011684  0.067102 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.006869  0.007243  0.948   0.343    
Max_temperature 0.348258  0.013096 26.593 < 2e-16 ***
Min_temperature 0.591677  0.018973 31.185 < 2e-16 ***
Sea_level_pressure -0.163728  0.018318 -8.938 < 2e-16 ***
Standard_pressure 0.126126  0.016138  7.815 9.87e-15 ***
Visibility       0.028716  0.004504  6.376 2.37e-10 ***
Wind_speed        0.048270  0.006801  7.098 1.90e-12 ***
I(Min_temperature^2) 0.517412  0.050180 10.311 < 2e-16 ***
I(Max_temperature^2) 0.194803  0.012303 15.833 < 2e-16 ***
I(Dewpoint^2)      0.759713  0.042869 17.722 < 2e-16 ***
Wind_speed:Max_wind_speed -0.086448  0.015202 -5.687 1.54e-08 ***
Min_temperature:Dewpoint -1.438498  0.090299 -15.930 < 2e-16 ***
```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01837 on 1597 degrees of freedom
Multiple R-squared: 0.9923, Adjusted R-squared: 0.9923
F-statistic: 1.88e+04 on 11 and 1597 DF, p-value: < 2.2e-16

```

Figura 3.18: Modelo lineal múltiple óptimo

$R^2$  ajustado de 0.9923, 1597 grado de libertad y error residual de 0.01837. El error cuadrático medio cometido es 0.01830154. Podemos ver gráficamente el resultado obtenido:

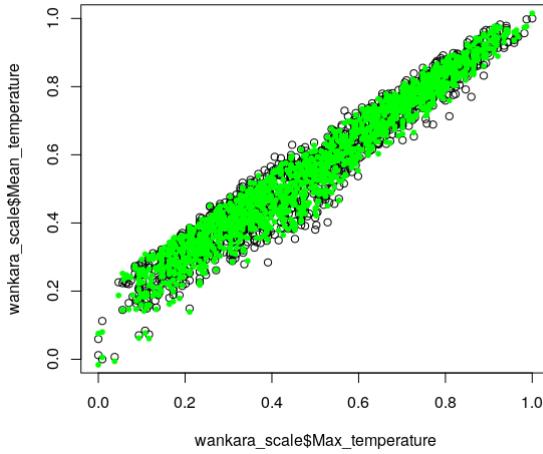


Figura 3.19: Plot del modelo óptimo encontrado sobre la gráfica Max-temperature-Mean-temperature

Como se ve, se cubre gran parte de los puntos. Llevamos a cabo una validación cruzada de siete particiones para garantizar los resultados. Se comete un error cuadrático medio

en training de 1.848325 y en test de 1.784818, por lo que vemos que el modelo no sobreajusta y funciona de manera satisfactoria.

No cabe duda de que los modelos lineales múltiples, introduciendo interacciones y no linealidades mejoran los resultados del apartado anterior, si bien se pierde interpretabilidad. Mi opción es siempre dar las dos facetas: el modelo interpretable más útil en términos de ajuste y una opción óptima, menos interpretable pero que gana, en este caso, una centésima en rendimiento.

### 3.3. kNN en regresión

Ajustamos ahora un modelo kNN. Lo hago de tres formas distintas: la primera, con todos los regresores; la segunda, con el modelo óptimo obtenido en el apartado anterior; la tercera, con el modelo más interpretable.

- **kNN con todos las variables.**

Aplico kNN con todas las variables y este es el resultado:

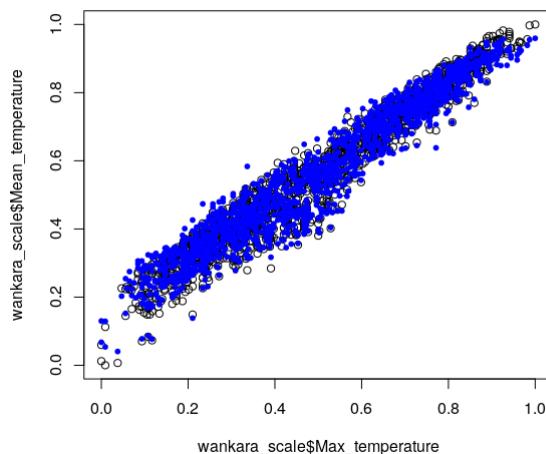


Figura 3.20: Primera aproximación: knn con todas las variables

El error cuadrático medio cometido es 0.02173683

- **kNN con la combinación óptima (para el modelo lineal) de las variables**

Rescato la combinación para que esté clara:

```

fitknn2 = kknn(wankara_scale$Mean_temperature~.-Precipitation+Max_wind_speed*Wind_speed+I(Min_temperature^2)+I(Max_temperature^2)
               +Min_temperature*Dewpoint+I(Dewpoint^2)|Dewpoint-Visibility-Max_wind_speed,wankara_scale,wankara_scale)

```

Figura 3.21: Segunda aproximación: knn con la combinación óptima anterior. Modelo

El resultado es el siguiente:

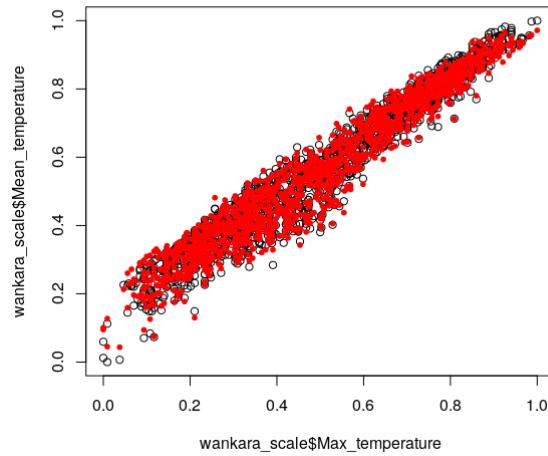


Figura 3.22: Primera aproximación: knn con la combinación óptima anterior

El error cuadrático medio cometido es 0.01609035, por lo que este modelo es mejor que kNN con todas las variables.

- **kNN con el modelo lineal más interpretable.**

El modelo lineal más interpretable era el que sólo incluía Max temperature, Min temperature y Dewpoint como regresores. El resultado es el siguiente:

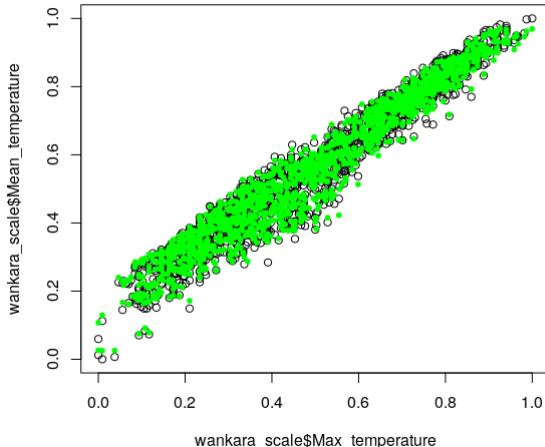


Figura 3.23: Primera aproximación: knn con la combinación más interpretable

El error cuadrático medio cometido es 0.01461469, aún menor que con la combinación óptima del modelo lineal, por lo que vemos que el modelo lineal y kNN varían en sus resultados.

En resumen, confirmamos que las combinaciones de regresores para kNN y el modelo lineal obtienen diferentes salidas y que, por ahora, para kNN, la interpretabilidad pesa más que la optimalidad con interacciones y términos no lineales.

### 3.4. Comparación con algoritmos

Llegamos a la última sección de regresión: comparación de algoritmos. En este caso, la llevo a cabo de dos formas. En la primera, realizo una validación cruzada de 5 particiones con el modelo lineal y kNN, ambos dos formados por todos los regresores. Calculo la media del error cuadrático medio para entrenamiento y test y almaceno esos resultados en los ficheros csv cedidos en PRADO. A continuación, realizo los test de Wilcoxon por pares y el de Friedman con Post-Hoc Holm para ver si hay diferencias significativas entre los algoritmos evaluados en los distintos datasets.

Por otro lado, supongo las 5 particiones cedidas para la validación cruzada como distintos conjuntos de datos y hago una comparación sobre los resultados de los modelos lineal, kNN y Random Forest para ver si hay diferencias significativas exclusivamente en el dataset Wankara.

#### 3.4.1. Comparativa sobre los distintos datasets

- **Test** Tras realizar la validación cruzada para el modelo lineal y kNN e incorporar los errores al csv, los resultados de los algoritmos en test son los siguientes:

```

> tablatst
      out_test_lm out_test_kknn out_test_m5p
abalone    4.950000e+00  5.400000e+00  4.680000e+00
ANACALT   1.700000e-01  1.200000e-02  7.000000e-03
autoMPG6   1.162000e+01  7.740000e+00  8.240000e+00
autoMPG8   1.140000e+01  8.110000e+00  8.350000e+00
baseball   5.366760e+05  5.661130e+05  5.464640e+05
california 4.844366e+09  3.845914e+09  3.158145e+09
concrete   1.090000e+02  6.835600e+01  3.800000e+01
dee        1.705200e-01  1.732600e-01  1.699600e-01
delta_ail  2.960000e-08  3.140000e-08  2.720000e-08
delta_elv  2.100000e-06  2.410000e-06  2.050000e-06
forestFires 4.060940e+03  5.841000e+03  4.071040e+03
friedman   7.298700e+00  3.196100e+00  5.349100e+00
house      2.072908e+09  1.425915e+09  1.305419e+09
mortgage   1.484100e-02  3.003600e-02  1.448300e-02
stock       5.510000e+00  4.500000e-01  1.000000e+00
treasury   6.082100e-02  4.743900e-02  8.124800e-02
wankara    2.480880e+00  6.773059e+00  1.650000e+00
wizmir     1.605000e+00  6.060000e+00  1.449000e+00

```

Figura 3.24: Tabla de MSE para cada algoritmo sobre los conjuntos

A continuación, aplico el test de Wilcoxon por parejas.

- LM (other) vs kNN (reference) Los resultados son los siguientes:

```

> LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1], alternative = "two.sided", paired=TRUE)
> Rmenos <- LMvsKNNtst$statistic
> Rmas
V
78
> Rmenos
V
93
> pvalue
[1] 0.7660294

```

Figura 3.25: Test de Wilcoxon: LM vs KNN

No hay diferencias significativas (p-valor 0.7660294), ya que solo hay un 23.39706 % de confianza de que sean distintos.

- LM (other) vs M5P (reference) Los resultados son los siguientes:

```

> LMvsM5Ptst <- wilcox.test(wilc_1_3[,2], wilc_1_3[,1], alternative = "two.sided", paired=TRUE)
> Rmenos <- LMvsM5Ptst$statistic
> Rmas
V
17
> Rmenos
V
154
> pvalue
[1] 0.001579285

```

Figura 3.26: Test de Wilcoxon: LM vs M5P

Hay diferencias significativas con un 99.8420715 % de confianza.

- kNN (other) vs M5P (reference) Los resultados son los siguientes:

```
> KKNNvsM5Ptst <- wilcox.test(wilc_2_3[,2], wilc_2_3[,1], alternative = "two.sided", paired=TRUE)
> Rmenos <- KKNNvsM5Ptst$statistic
> Rmas
V
53
> Rmenos
V
118
> pvalue
[1] 0.1673508
```

Figura 3.27: Test de Wilcoxon: kNN vs M5P

No se pueden asumir diferencias significativas, ya que sólo tenemos un 83.26492 % de confianza de que sean distintos.

Si aplicamos el test de Friedman

```
Friedman rank sum test

data: as.matrix(tablatst)
Friedman chi-squared = 8.4444, df = 2, p-value = 0.01467
```

Figura 3.28: Test de Friedman

y el Post Hoc Holm

```
Pairwise comparisons using Wilcoxon signed rank test

data: as.matrix(tablatst) and groups

 1     2
2 0.580 -
3 0.081 0.108

P value adjustment method: holm
```

Figura 3.29: Post-hoc holm

vemos que hay diferencias significativas entre M5P y LM favorables a M5P (0.081) y entre M5P y kNN (0.108) con aproximadamente un 90 % de confianza, mientras que LM y kNN pueden suponerse iguales.

## ■ Train

Evalúo ahora los resultados de training, para encontrar posibles sobreajustes. Trabajo de la misma manera: primero Wilcoxon por parejas y después Friedman:

- LM (other) vs kNN (reference) Los resultados son los siguientes:

```
> LMvsKNNtr <- wilcox.test(wilc_1_2_tr[,2], wilc_1_2_tr[,1], alternative = "two.sided", paired=TRUE)
> Rmenos_tr <- LMvsKNNtr$statistic
> Rmas_tr
V
10
> Rmenos_tr
V
161
> pvalue_tr
[1] 0.000328064
```

Figura 3.30: Test de Wilcoxon: LM vs KNN

Hay diferencias significativas (p-valor 0.000328064), con una confianza del 99 %.

- LM (other) vs M5P (reference) Los resultados son los siguientes:

```
> LMvsM5Ptr <- wilcox.test(wilc_1_3_tr[,2], wilc_1_3_tr[,1], alternative = "two.sided", paired=TRUE)
> Rmenos_tr <- LMvsM5Ptr$statistic
> Rmas_tr
V
3
> Rmenos_tr
V
168
> pvalue_tr
[1] 3.814697e-05
```

Figura 3.31: Test de Wilcoxon: LM vs M5P

Hay diferencias significativas con más de un 99 % de confianza.

- kNN (other) vs M5P (reference) Los resultados son los siguientes:

```
> KKNNvsM5Ptr <- wilcox.test(wilc_2_3_tr[,2], wilc_2_3_tr[,1], alternative = "two.sided", paired=TRUE)
> Rmenos_tr <- KKNNvsM5Ptr$statistic
> Rmas_tr
V
160
> Rmenos_tr
V
11
> pvalue_tr
[1] 0.0004196167
```

Figura 3.32: Test de Wilcoxon: kNN vs M5P

Hay diferencias significativas con un 99 % de confianza.

Si aplicamos el test de Friedman

```

> test_friedman_tr
Friedman rank sum test

data: as.matrix(tablatr)
Friedman chi-squared = 20.333, df = 2, p-value = 3.843e-05

```

Figura 3.33: Test de Friedman

y el Post Hoc Holm

```

> pairwise.wilcox.test(as.matrix(tablatr), groups, p.adjust = "holm", paired = TRUE)

Pairwise comparisons using Wilcoxon signed rank test

data: as.matrix(tablatr) and groups

  1      2
2 0.0031 -
3 0.0032 0.0032

P value adjustment method: holm

```

Figura 3.34: Post-hoc holm

Se puede considerar que todos los algoritmos son estadísticamente diferentes con el mismo intervalo de confianza (99.7 % aprox.). Como se puede ver, el comportamiento en training y test puede sugerir sobreajuste o que los datos son erróneos.

### 3.4.2. Comparativa sobre los folds de Wankara

Tras ejecutar el modelo lineal, kNN y Random Forest con una validación cruzada de 5 particiones del dataset Wankara, obtenemos en test los siguientes errores cuadráticos medios:

	lmMSEtest	kknnMSEtest	rfMSEtest
1	2.598022	6.985077	2.179853
2	2.448518	7.265437	1.863810
3	2.426096	7.068859	1.838477
4	2.250193	6.850206	2.146991
5	2.681572	5.695715	2.014394

Figura 3.35: Tabla resultados en test para los folds

Aplico ahora los tests de Wilcoxon y Friedman para terminar.

- LM (other) vs kNN (reference) Los resultados son los siguientes:

```

> LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1], alternative = "two.sided", paired=TRUE)
> Rmenos <- LMvsKNNtst$statistic
> Rmas
V
15
> Rmenos
V
0
> pvalue
[1] 0.0625

```

Figura 3.36: Test de Wilcoxon: LM vs KNN

Hay diferencias significativas con un 93.75 % de confianza

- LM (other) vs M5P (reference) Los resultados son los siguientes:

```

> LMvsRFtst <- wilcox.test(wilc_1_3[,2], wilc_1_3[,1], alternative = "two.sided", paired=TRUE)
> Rmenos <- LMvsRFtst$statistic
> Rmas
V
0
> Rmenos
V
15
> pvalue
[1] 0.0625

```

Figura 3.37: Test de Wilcoxon: LM vs RF

Hay diferencias significativas con un 93.75 % de confianza

- kNN (other) vs RF (reference) Los resultados son los siguientes:

```

> KKNNvsRFtst <- wilcox.test(wilc_2_3[,2], wilc_2_3[,1], alternative = "two.sided", paired=TRUE)
> Rmenos <- KKNNvsRFtst$statistic
> Rmas
V
0
> Rmenos
V
15
> pvalue
[1] 0.0625

```

Figura 3.38: Test de Wilcoxon: kNN vs RF

Hay diferencias significativas con un 93.75 % de confianza.

Si aplicamos el test de Friedman

```
Friedman rank sum test  
data: as.matrix(tablatst)  
Friedman chi-squared = 10, df = 2, p-value = 0.006738
```

Figura 3.39: Test de Friedman

y el Post Hoc Holm

```
Pairwise comparisons using Wilcoxon signed rank test  
data: as.matrix(tablatst) and groups  
 1   2  
2 0.19 -  
3 0.19 0.19  
P value adjustment method: holm
```

Figura 3.40: Post-hoc holm

Vemos que de forma conjunta, no podemos asegurar que sean distintos porque sólo contamos con un 81 % de confianza en ello.

## 4. Problema de clasificación: Vowel

Comenzamos con el estudio de clasificación para el conjunto de datos Vowel. Esta sección constará de cuatro partes: la primera dedicada al algoritmo KNN; la segunda, a LDA; la tercera, a QDA; por último, realizo una comparación entre los algoritmos nombrados. Dicha comparación se hace gracias a la ejecución de los mismos sobre 5 particiones previamente fijadas. Se almacena el accuracy de cada algoritmo y partición y se llevan a cabo los tests estadísticos de Wilcoxon (para el estudio por parejas de algoritmos) y el de Friedman, con Post-Hoc Holm para la comparación conjunta. Previa ejecución de cualquier algoritmo, las variables han sido escaladas para que la diferencia de rangos no afecte a los resultados. Además, sólo utilizo como variables predictivas las numéricas, obviando por el momento el sexo (para hacer distinciones) o el interlocutor.

### 4.1. KNN

Dentro del ejercicio 1 propuesto, se pide que se trabaje con distintos valores de  $k$  para el algoritmo kNN. En esta primera parte, sobre el conjunto de datos al completo, realizo un particionamiento training-test del 70%-30% para entrenar el algoritmo con distintos valores de  $k$ , desde 1 hasta 15. En la siguiente imagen podemos ver el rendimiento extraído:

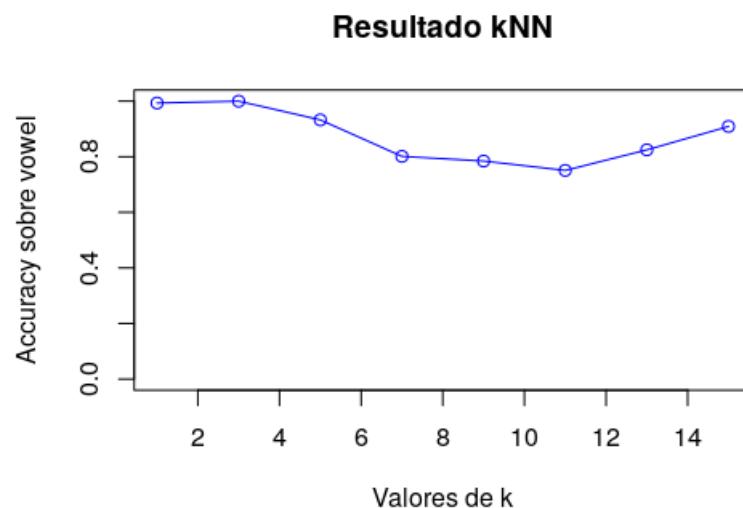


Figura 4.1: Accuracy para los distintos valores de  $k$

A la luz de los resultados, elijo el valor de  $k = 3$  para continuar y ahora intento mostrar las fronteras de decisión entre clases, haciendo un plot con las variables, por ejemplo,, F0 y F1.

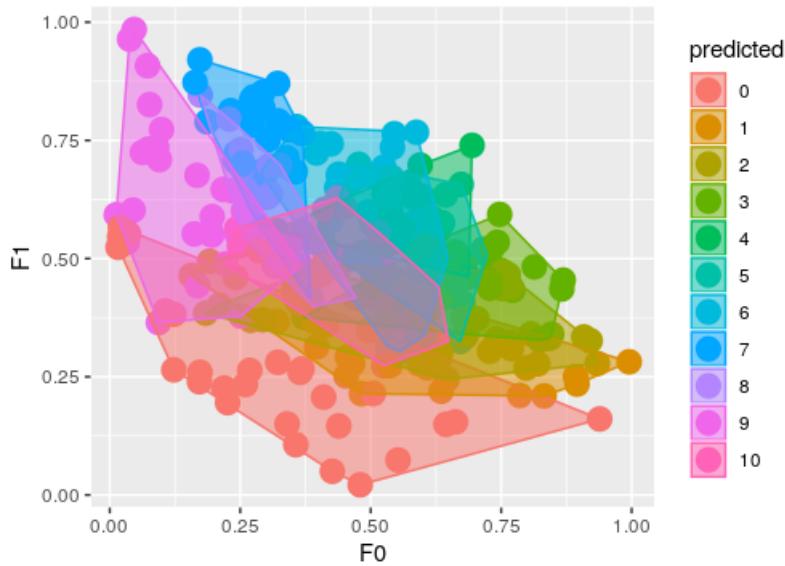


Figura 4.2: Plot F1-F2 para mostrar las fronteras de decisión

o para F2 y F6

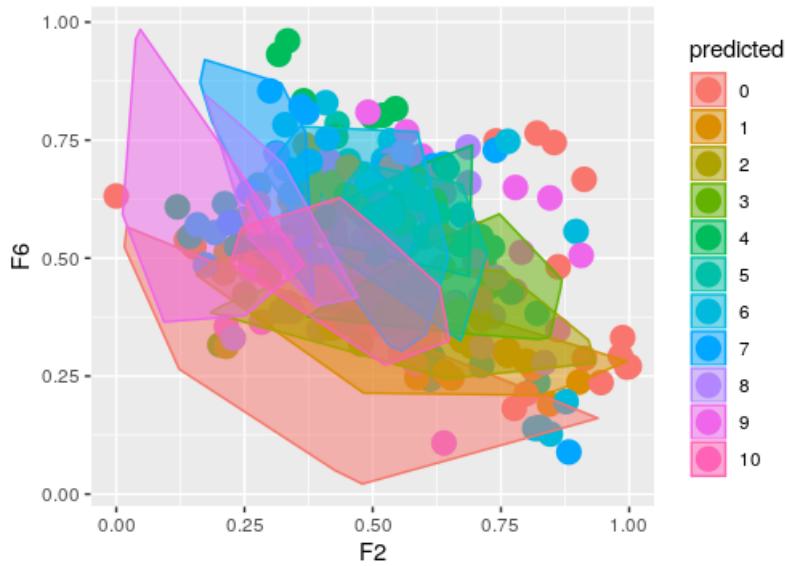


Figura 4.3: Plot F2-F6 para mostrar las fronteras de decisión

Si realizamos la clasificación por sexos y el rendimiento respecto del valor de  $k$ , vemos que los resultados son algo distintos:

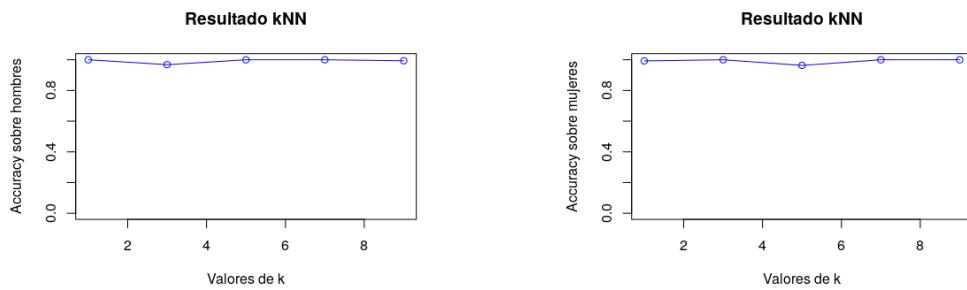


Figura 4.4: Valores de k vs Accuracy para hombres y mujeres

lo que nos muestra que, por separado, el resultado de la clasificación sería aún mejor. Por último, intento validar estos resultados con una validación cruzada de 10 particiones para ver los mejores valores de k tanto en training como en test. Primero, para training,

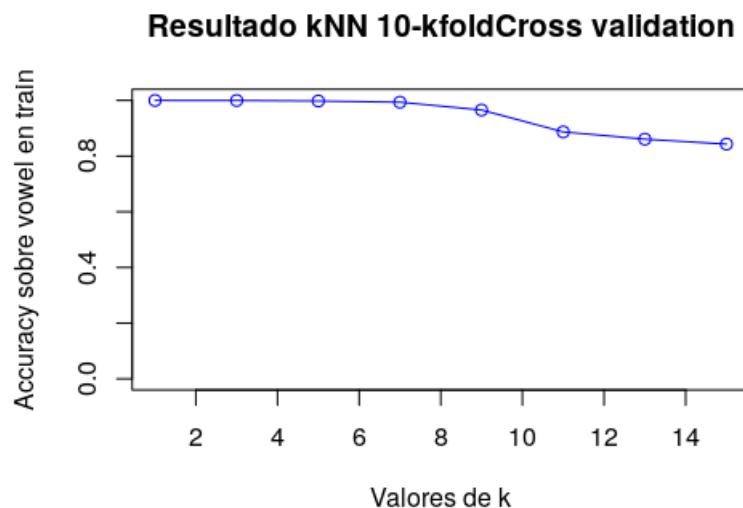


Figura 4.5: Evolución de accuracy respecto de k en training

donde los mejores valores son  $k = 1, k = 3$ . Para test,

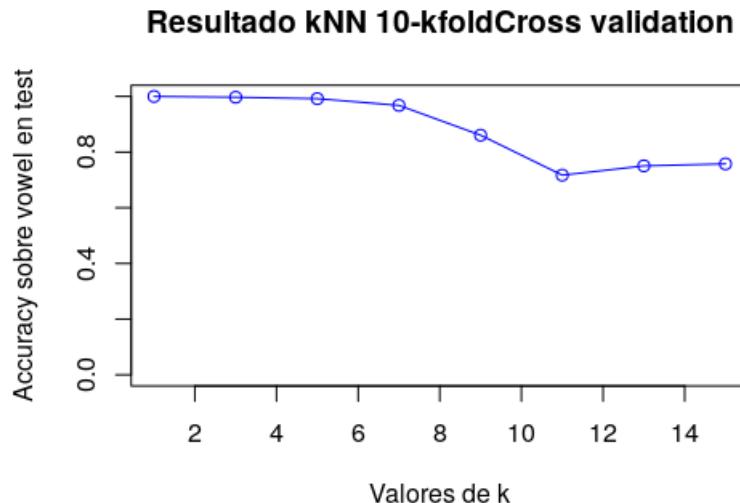


Figura 4.6: Evolución de accuracy respecto de  $k$  en test

donde vemos que los mejores valores se mantienen pero que caen a partir de  $k = 11$ . Para la comparación de algoritmos, utilizo los valores de  $k = 1, k = 3$  y almaceno el accuracy para cada partición. Además, a modo de resumen, calculo la media de los resultados en test para cada partición, obteniendo un 99,69697 %.

## 4.2. LDA

Pasamos ahora a LDA. Previa a su implementación sobre el conjunto de datos, tenemos que comprobar que las variables siguen una distribución normal y todas tienen la misma varianza (5).

```
> sapply(vowel[,3:13],shapiro.test)
      Sex          F0          F1          F2
statistic 0.6349725 0.9927972 0.9966029 0.9921715
p.value   1.735485e-41 9.807241e-05 0.03119502 4.245163e-05
method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
data.name "X[[i]]"      "X[[i]]"      "X[[i]]"      "X[[i]]"
      F4          F5          F6          F7
statistic 0.9971018 0.9863215 0.9964036 0.9939658
p.value   0.07072585 5.434785e-08 0.02250036 0.0005086354
method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
data.name "X[[i]]"      "X[[i]]"      "X[[i]]"      "X[[i]]"
      F9
statistic 0.9737706
p.value   2.208267e-12
method    "Shapiro-Wilk normality test"
```

Figura 4.7: Test de normalidad para cada variable

Como vemos, las variables no siguen una distribución normal (ya lo vimos también en el

EDA).

```
> sapply(vowel[,4:13],var)
   F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.04141627 0.03427701 0.03301931 0.04021808 0.02819500 0.03644937 0.02468606 0.02960445 0.03807462 0.03855497
```

Figura 4.8: Varianza de las variables

Podemos ver que las varianzas difieren en  $\pm 0,02$  aproximadamente, por lo que tampoco se cumple esta condición. En consecuencia, los resultados de LDA no serán especialmente rigurosos. Primero, entrenamos un modelo generando una partición, con las mismas proporciones que en el caso de kNN, sobre nuestro conjunto de datos. Veamos la matriz de confusión generada:

Confusion Matrix and Statistics											
Reference											
Prediction	0	1	2	3	4	5	6	7	8	9	10
0	12	3	2	0	0	0	0	0	0	4	0
1	7	12	5	0	0	0	0	0	0	0	5
2	0	6	17	5	0	1	0	0	0	0	2
3	0	0	2	21	1	4	0	0	0	0	1
4	0	0	0	0	11	2	6	0	0	0	2
5	0	0	0	8	9	8	1	0	0	0	5
6	0	0	1	0	14	1	11	0	1	0	1
7	0	0	0	0	0	0	1	20	1	0	1
8	0	0	0	0	0	0	3	7	17	1	1
9	1	0	0	0	0	0	0	4	2	18	0
10	0	0	0	2	0	7	2	0	2	0	16

Figura 4.9: Matriz de confusión para LDA

y cuyas estadísticas generales son

Overall Statistics
Accuracy : 0.5488
95% CI : (0.4903, 0.6064)
No Information Rate : 0.1212
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.5039
Mcnemar's Test P-Value : NA

Figura 4.10: Estadísticas generales para LDA

El resultado, sobre la partición estratificada, es bastante pobre. Intentamos una visualización con la herramienta *Partimat* sobre las variables F0, F1, F2 y F3:

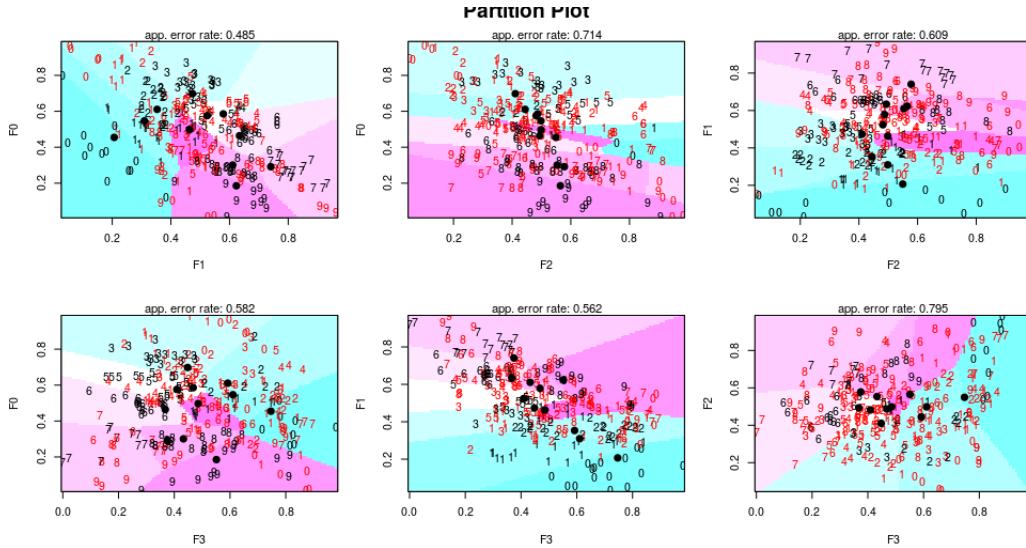


Figura 4.11: Partimat sobre F0-F3 para el modelo de LDA en Vowel

Si dividimos el dataset por sexos, a parte de que se consigue con mayor facilidad la normalidad (como ya vimos en EDA), los resultados mejoran mucho:

Confusion Matrix and Statistics

		Reference										
		0	1	2	3	4	5	6	7	8	9	10
Prediction		17	1	0	0	0	0	0	0	0	0	0
0		17	1	0	0	0	0	0	0	0	0	0
1		0	7	0	0	0	0	0	0	0	0	3
2		0	0	15	0	0	0	0	0	0	0	0
3		0	0	0	15	0	1	0	0	0	0	0
4		0	0	0	0	9	0	6	0	0	0	0
5		0	0	0	1	4	4	2	0	0	0	1
6		0	0	0	0	3	0	11	0	0	0	0
7		0	0	0	0	0	0	1	12	1	0	0
8		0	0	0	0	0	0	0	2	12	0	1
9		0	0	0	0	0	0	0	0	4	11	0
10		0	0	0	0	0	0	0	0	0	0	13

Figura 4.12: Matriz de confusión para hombres LDA

Así también se refleja en las estadísticas

```

Overall Statistics

    Accuracy : 0.8025
    95% CI : (0.7316, 0.8617)
    No Information Rate : 0.1274
    P-Value [Acc > NIR] : < 2.2e-16

    Kappa : 0.7823

    Mcnemar's Test P-Value : NA

```

Figura 4.13: Estadísticas generales para hombres LDA

con un aumento del 20 % en accuracy. También mejora para el subconjunto de mujeres en un 15 %:

```

Overall Statistics

    Accuracy : 0.75
    95% CI : (0.6686, 0.8202)
    No Information Rate : 0.125
    P-Value [Acc > NIR] : < 2.2e-16

    Kappa : 0.7244

    Mcnemar's Test P-Value : NA

```

Figura 4.14: Estadísticas generales para mujeres LDA

Para validar, los resultados, llevo a cabo una validación cruzada de 10 particiones con predicciones sobre el conjunto de training como el de test, almacenándose el accuracy para cada partición y obteniéndose una media de 64,82604 % en training y 60,30303 % en test.

### 4.3. QDA

A juzgar por los resultados de LDA, parece que las 11 clases de nuestro dataset son difícilmente separables con una función lineal. Sin embargo, una cuadrática pueda tener un mejor rendimiento. Es por eso que utilizamos ahora QDA. Como en LDA, necesitamos comprobar unas hipótesis. En este caso, QDA requiere que la varianza de los regresores sea igual en cada clase, aunque pueda ser distinta entre clases. Como se verá a continuación, esta hipótesis tampoco se va a cumplir (como en LDA), por lo que la capacidad de predicción de QDA se verá mermada.

```

[1] "Clase 0"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.06299981 0.01545379 0.07917503 0.01525657 0.01884297 0.06085343 0.04205933 0.05017306 0.03064355 0.04923129
[1] "Clase 1"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.04736185 0.00995720 0.03472767 0.02791121 0.01719496 0.05988390 0.02293772 0.03567552 0.02846704 0.04070711
[1] "Clase 2"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.026368485 0.003157154 0.021862165 0.024776874 0.012763482 0.040408233 0.012105326 0.027862989 0.012003987 0.038932230
[1] "Clase 3"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.016363641 0.003766025 0.013105245 0.016332689 0.012903704 0.015264781 0.007645430 0.015640887 0.007422938 0.033553140

```

Figura 4.15: Varianza de los regresores entre las clases 0 y 4

```

[1] "Clase 4"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.005100315 0.006690323 0.027935709 0.023229549 0.011168113 0.020388444 0.024769589 0.027230264 0.024112178 0.027941582
[1] "Clase 5"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.010067152 0.006800740 0.017149907 0.017357498 0.013992326 0.009474645 0.010141955 0.021671803 0.036901323 0.030145924
[1] "Clase 6"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.006956865 0.014290704 0.044424324 0.050209003 0.015372938 0.023859686 0.043097622 0.024797353 0.069056532 0.038466015
[1] "Clase 7"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.004473334 0.014409588 0.037551202 0.063932880 0.029344288 0.027552124 0.043312827 0.016814771 0.037212804 0.029133023
[1] "Clase 8"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.00585881 0.01797293 0.02689837 0.03244559 0.02414673 0.01680614 0.01673443 0.02053186 0.04244090 0.03854256

```

Figura 4.16: Varianza de los regresores entre las clases 4 y 8

```

[1] "Clase 9"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.009078417 0.025251471 0.030536197 0.042335647 0.021034359 0.037966543 0.025704738 0.022053887 0.055629845 0.048724162
[1] "Clase 10"
      F0      F1      F2      F3      F4      F5      F6      F7      F8      F9
0.010793528 0.007592757 0.009062564 0.013765028 0.014630257 0.017387246 0.009058449 0.020685227 0.038193876 0.040036448

```

Figura 4.17: Varianza de los regresores entre las clases 9 y 10

Ha quedado patente que no se cumple la hipótesis de QDA. Aún así, seguimos adelante en el estudio. Como en los algoritmos anteriores, realizamos una partición estratificada 70-30 para training/test y genero un modelo con QDA. Vemos que el resultado es notoriamente mejor. Primero, con la matriz de confusión

#### Confusion Matrix and Statistics

		Reference											
		Prediction	0	1	2	3	4	5	6	7	8	9	10
0	25	1	0	0	0	0	0	0	0	0	0	0	
1	0	25	0	0	0	0	0	0	0	0	0	0	
2	0	2	24	0	0	1	1	0	0	2	0	0	
3	0	0	2	20	0	0	0	0	0	0	0	0	
4	0	0	0	1	21	4	1	0	0	0	1	0	
5	0	0	0	2	2	22	1	0	0	0	4	0	
6	0	0	0	0	2	0	23	1	0	0	0	0	
7	0	0	0	0	0	0	0	1	26	3	0	0	
8	0	0	0	0	0	0	0	0	1	22	1	1	
9	0	0	0	0	0	0	0	0	0	0	25	0	
10	0	0	0	0	0	2	0	0	0	0	0	27	

Figura 4.18: Matriz de confusión para QDA

y cuyas estadísticas generales son

Overall Statistics	
Accuracy :	0.8754
95% CI :	(0.8324, 0.9107)
No Information Rate :	0.1111
P-Value [Acc > NIR] :	< 2.2e-16
Kappa :	0.8629
McNemar's Test P-Value : NA	

Figura 4.19: Estadísticas generales para QDA

con un 87,54 % de accuracy. Sin embargo, la matriz de confusión da la sensación, al menos en la mayoría de las clases, que el rendimiento debería ser mayor. Veamos las estadísticas por clases

	Class: 0	Class: 1	Class: 2	Class: 3	Class: 4	Class: 5	Class: 6	Class: 7	Class: 8	Class: 9	Class: 10
Sensitivity	1.00000	0.89286	0.92308	0.86957	0.84000	0.75862	0.85185	0.92857	0.88000	0.89286	0.81818
Specificity	0.99632	1.00000	0.97786	0.99270	0.97426	0.96642	0.98889	0.98513	0.98897	1.00000	0.99242
Pos Pred Value	0.96154	1.00000	0.80000	0.90909	0.75000	0.70968	0.88462	0.86667	0.88000	1.00000	0.93103
Neg Pred Value	1.00000	0.98897	0.99251	0.98909	0.98513	0.97368	0.98524	0.99251	0.98897	0.98897	0.97761
Prevalence	0.08418	0.09428	0.08754	0.07744	0.08418	0.09764	0.09091	0.09428	0.08418	0.09428	0.11111
Detection Rate	0.08418	0.08418	0.08081	0.06734	0.07071	0.07407	0.07744	0.08754	0.07407	0.08418	0.09091
Detection Prevalence	0.08754	0.08418	0.10101	0.07407	0.09428	0.10438	0.08754	0.10101	0.08418	0.08418	0.09764
Balanced Accuracy	0.99816	0.94643	0.95047	0.93113	0.90713	0.86252	0.92037	0.95685	0.93449	0.94643	0.90530

Figura 4.20: Estadísticas generales para QDA por clase

donde vemos que todas las clases tienen un accuracy de más del 90 % excepto la 5, que tiene un 86,252 %. Este hecho podría explicarse con que, como hemos visto antes, para el subconjunto de mujeres, la clase 5 presenta outliers en la mayoría de las variables. Vemos, como antes, una separación del espacio representado en las variables F0-F3:

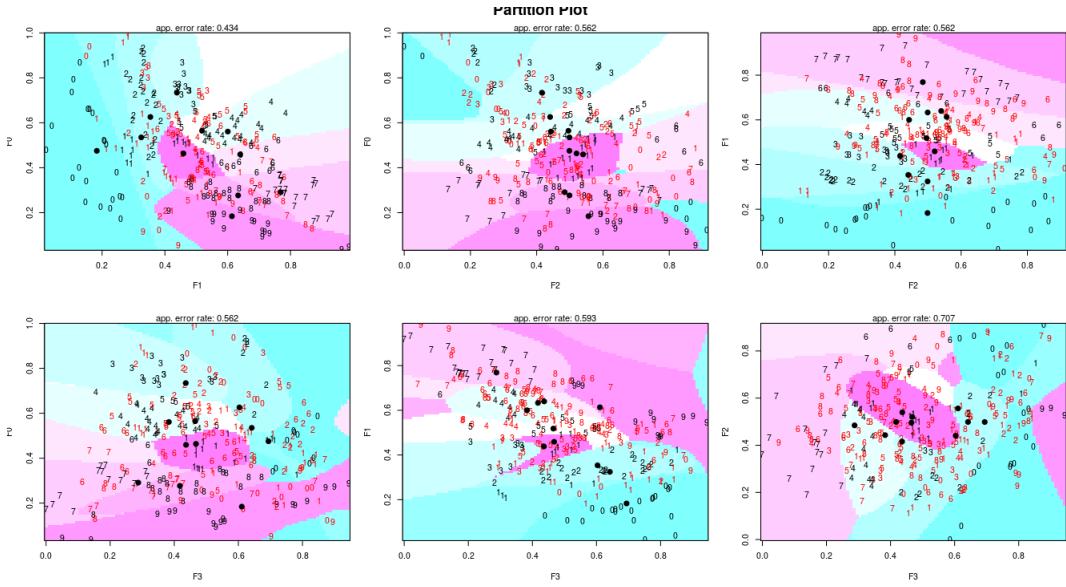


Figura 4.21: Partimat sobre F0-F3 para el modelo de QDA en Vowel

lo que nos muestra la dificultad en la partición por clases. Por último, para validar los resultados, llevo a cabo una validación cruzada con las 10 particiones ofrecidas, de nuevo almacenando el accuracy de cada partición tanto en training como en test. El accuracy medio en train resulta 94,92705 % y en test, 91,0101 %. Como estos resultados son satisfactorios, no hago la división por sexos como LDA. Sin embargo, casi con toda probabilidad obtendríamos un rendimiento mayor que con el conjunto total.

#### 4.4. Comparación de algoritmos en test

Para terminar con el apartado de clasificación, procedemos a la comparación de algoritmos. Como se ha comentado, para cada una de las etapas anteriores se han almacenado los distintos accuracy en las particiones. Tras ejecutar todos los modelos, la tabla de resultados es la siguiente:

	> tabla_resultados	resultados_knn1_test	resultados_knn3_test	resultados_lda_test	resultados_qda_test
1		1	0.989899	0.5656566	0.8787879
2		1	0.989899	0.6464646	0.9090909
3		1	1.000000	0.6363636	0.9090909
4		1	1.000000	0.5454545	0.9393939
5		1	1.000000	0.5959596	0.9595960
6		1	0.989899	0.5656566	0.8787879
7		1	1.000000	0.6060606	0.9595960
8		1	1.000000	0.6262626	0.8787879
9		1	1.000000	0.6060606	0.8686869
10		1	1.000000	0.6363636	0.9191919

Figura 4.22: Tabla de resultados para cada algoritmo y partición

Como se puede ver, en cada columna colocamos los modelos tratados: 1NN, 3NN, LDA y QDA. Por cada fila se encuentra el accuracy obtenido en test para la partición correspondiente. En esta sección utilizaremos el test de Wilcoxon para hacer comparaciones por pares y el de Friedman (con Post-Hoc Holm) para hacer una comparativa conjunta. Debido a que los valores de accuracy tienen valores muy parecidos en las columnas (de hecho, hay bastantes repetidos). Dichos empates van en detrimento de la capacidad de rechazar la hipótesis nula (esto es, que los algoritmos no tienen diferencias significativas entre sí). Dado que los tests no paramétricos se basan en rankings, los empates aporta información nula a dicha ordenación y el p-valor resultante tiende a ser peor. En la vida real, necesitaríamos buscar nuevas muestras que nos proporcionaran valores distintos para así dar lugar a variabilidad. Aquí, nos ceñiremos a los resultados obtenidos dado que las particiones están fijas.

#### **4.4.1. 1NN vs LDA**

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,005825024, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### **4.4.2. 1NN vs QDA**

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,005729376, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### **4.4.3. 3NN vs LDA**

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,005825024, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### **4.4.4. 3NN vs QDA**

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,005857099, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### **4.4.5. LDA vs QDA**

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 0 y  $R^-$  de 55, alcanzando un p-valor

de 0,00588927, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### 4.4.6. Comparativa general con el test de Friedman

Utilizamos el test de Friedman para comparar globalmente los algoritmos. En él, establecemos como hipótesis nula que no hay diferencias significativas entre los algoritmos. Estos son los resultados:

```
> test_friedman
Friedman rank sum test
data: as.matrix(tabla_resultados)
Friedman chi-squared = 29.323, df = 3, p-value = 1.916e-06
```

Figura 4.23: Resultados del test de Friedman para clasificación

Como podemos ver, el p-valor es bastante menor que 0.05, por lo que debemos rechazar la hipótesis nula. Por tanto, consideramos que hay diferencias significativas al menos entre un par de algoritmos. Vemos las diferencias a través del Post-Hoc

```
Pairwise comparisons using Wilcoxon signed rank test
data: as.matrix(tabla_resultados) and groups
  1     2     3
2 0.149 -    -
3 0.034 0.034 -
4 0.034 0.034 0.034
```

Figura 4.24: Post Hoc Holm en clasificación

Existen diferencias significativas entre QDA y 1NN (a favor de 1NN), QDA y 3NN (a favor de 3NN), QDA y LDA (a favor de QDA) con un 96,6 % de confianza, al igual que LDA y 1NN (a favor de 1NN) y LDA y 3NN (a favor de 3NN). 1NN y 3NN pueden considerarse equivalentes.

### 4.5. Comparación de algoritmos en entrenamiento

Por último, esta sección se encarga de comparar la actuación de los algoritmos en training, para poder diagnosticar posibles sobreaprendizaje. En primer lugar, muestro la tabla de resultados para training.

	resultados_knn1_tr	resultados_knn3_tr	resultados_lda_tr	resultados_qda_tr
1	1	1.0000000	0.6408530	0.9494949
2	1	1.0000000	0.6632997	0.9450056
3	1	1.0000000	0.6374860	0.9494949
4	1	1.0000000	0.6453423	0.9472503
5	1	0.9988777	0.6576880	0.9506173
6	1	1.0000000	0.6531987	0.9450056
7	1	0.9988777	0.6329966	0.9427609
8	1	1.0000000	0.6610550	0.9528620
9	1	1.0000000	0.6487093	0.9539843
10	1	1.0000000	0.6419753	0.9562290

Figura 4.25: Tabla de resultados para training

#### 4.5.1. 1NN vs LDA

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,001953125, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### 4.5.2. 1NN vs QDA

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,005857099, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### 4.5.3. 3NN vs LDA

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,001953125, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### 4.5.4. 3NN vs QDA

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 55 y  $R^-$  de 0, alcanzando un p-valor de 0,005857099, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### 4.5.5. LDA vs QDA

Realizamos el test de Wilcoxon para tratar de encontrar diferencias significativas en los algoritmos. Tras realizarlo, encontramos un  $R^+$  de 0 y  $R^-$  de 55, alcanzando un p-valor de 0,00588927, por lo que tenemos que rechazar la hipótesis de que los algoritmos no tengan diferencias significativas.

#### 4.5.6. Comparativa general con Friedman

Utilizamos el test de Friedman para comparar globalmente los algoritmos. En él, establecemos como hipótesis nula que no hay diferencias significativas entre los algoritmos. Estos son los resultados:

```
> test_friedman_tr

Friedman rank sum test

data: as.matrix(tabla_resultados_tr)
Friedman chi-squared = 29.478, df = 3, p-value = 1.777e-06
```

Figura 4.26: Resultados del test de Friedman en training

Como podemos ver, el p-valor es bastante menor que 0.05, por lo que debemos rechazar la hipótesis nula. Por tanto, consideramos que hay diferencias significativas al menos entre un par de algoritmos. Vemos las diferencias a través del Post-Hoc

```
Pairwise comparisons using Wilcoxon signed rank test

data: as.matrix(tabla_resultados_tr) and groups

  1     2     3
2 0.346 -   -
3 0.012 0.012 -
4 0.023 0.023 0.023
```

Figura 4.27: Post Hoc Holm en training clasificación

Existen diferencias significativas entre QDA y 1NN (a favor de 1NN), QDA y 3NN (a favor de 3NN), QDA y LDA (a favor de QDA) con un 96,6 % de confianza, al igual que LDA y 1NN (a favor de 1NN) y LDA y 3NN (a favor de 3NN). 1NN y 3NN pueden considerarse equivalentes. Como podemos ver, descartamos sobreaprendizaje porque los resultados coinciden.

## 5. Bibliografía

### Referencias

- [1] <https://es.weatherspark.com/y/97345/clima-promedio-en-ankara-turqu>
- [2] [https://sci2s.ugr.es/keel/dataset.php?cod=113.](https://sci2s.ugr.es/keel/dataset.php?cod=113)
- [3] [https://sci2s.ugr.es/keel/dataset.php?cod=41.](https://sci2s.ugr.es/keel/dataset.php?cod=41)
- [4] Keel: Knowledge extraction based on evolutionary learning. *www.keel.es*.
- [5] Trevor Hastie Gareth James, Daniela Witten and Robert Tibshirani. An Introduction to Statistical Learning with Applications in R. *Springer*, 2013.
- [6] Hubert W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967.
- [7] S Robbins. A Misdirected Principle with a Catch: Explicability for AI. *Minds and Machines*. *Springer*, 2019.
- [8] S. S. SHAPIRO and M. B. WILK. An analysis of variance test for normality (complete samples)†. *Biometrika*, 52(3-4):591–611, 12 1965.
- [9] M. B. WILK and R. GNANADESIKAN. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17, 03 1968.

# Código Vowel

Luis Balderas Ruiz

```
#####
# INTRODUCCIÓN A LA CIENCIA DE DATOS
# Autor: Luis Balderas Ruiz
# EDA+Clasificación
# Dataset: vowel
#####

# Cálculo de binwidth óptimo para un histograma
binwd = function(data){
  size = length(data)
  dt = sd(data)
  cr = size^(1/3)
  return(1/(cr)*dt*3.49)
}

# Función para crear la matriz de correlación más bonita
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor = cormat[ut],
    p = pmat[ut]
  )
}

# Lectura del fichero de datos para proceder al EDA
vowel = read.csv("./data/vowel/vowel.dat", header=FALSE, comment.char="@")
colnames(vowel) = c("TT", "SpeakerNumber", "Sex",
                   "F0", "F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "Class")

# Estudiemos la estructura del conjunto
str(vowel)

# Borramos la variable TT porque no considero necesario dividir entre training y test para el EDA
vowel$TT = NULL

# Convierto SpeakerNumber y Sex a factores
vowel$Sex = factor(vowel$Sex, levels = c(0,1),
                     labels = c("Masculino", "Femenino"))
vowel$SpeakerNumber = factor(vowel$SpeakerNumber)

# Resumen estadístico de cada variable
summary(vowel)
sapply(vowel[,3:12], sd)

# Búsqueda visual de correlaciones entre las variables en el dataset completo m
plot(vowel[,3:12])
```

```

# Separación por sexos
library(tidyverse)
hombres = vowel %>% filter(vowel$Sex == "Masculino")
mujeres = vowel %>% filter(vowel$Sex == "Femenino")

# Búsqueda visual de correlaciones por sexos
plot(hombres[,3:12])
plot(mujeres[,3:12])

#####
# Distribución de las variables numéricas. Skewness
library(ggplot2)
library(e1071)
dist_sexo = ggplot(vowel,aes(x=Sex, fill=SpeakerNumber))
+ geom_bar(alpha=1/3) + theme(legend.position = "top") + labs(title="Distribución por sexos")
dist_sexo

distf0 = ggplot(vowel,aes(x=F0, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F0)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F0")
distf0
# positiva
skewness(vowel$F0)
kurtosis(vowel$F0)

distf1 = ggplot(vowel,aes(x=F1, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F1)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F1")
distf1
#negativa
skewness(vowel$F1)
kurtosis(vowel$F1)

distf2 = ggplot(vowel,aes(x=F2, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F2)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F2")
distf2
# positiva
skewness(vowel$F2)
kurtosis(vowel$F2)

distf3 = ggplot(vowel,aes(x=F3, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F3)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F3")
distf3
# positiva
skewness(vowel$F3)
kurtosis(vowel$F3)

distf4 = ggplot(vowel,aes(x=F4, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F4)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F4")
distf4

```

```

# positiva
skewness(vowel$F4)
kurtosis(vowel$F4)

distf5 = ggplot(vowel,aes(x=F5, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F5)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F5")
distf5

# positiva
skewness(vowel$F5)
kurtosis(vowel$F5)

distf6 = ggplot(vowel,aes(x=F6, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F6)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F6")
distf6

# negativa
skewness(vowel$F6)
kurtosis(vowel$F6)

distf7 = ggplot(vowel,aes(x=F7, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F7)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F7")
distf7

# positiva
skewness(vowel$F7)
kurtosis(vowel$F7)

distf8 = ggplot(vowel,aes(x=F8, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F8)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F8")
distf8

# positiva
skewness(vowel$F8)
kurtosis(vowel$F8)

distf9 = ggplot(vowel,aes(x=F9, fill=Sex))
+ geom_histogram(binwidth = binwd(vowel$F9)) + theme(legend.position = "right")
+ labs(title="Distribución de la variable F9")
distf9

# positiva
skewness(vowel$F9)
kurtosis(vowel$F9)
#####
#####

##### # BOXPLOTS
# bps_i --> Boxplot separando por Sexo (y con colores cada interlocutor)
# bpsn_i --> Boxplot separando por interlocutores (y con colores el sexo)
bps0 = ggplot(vowel, aes(x=Sex, y=F0, fill = SpeakerNumber))
    + geom_boxplot(outlier.colour="red",
                    outlier.shape=8,outlier.size=4)

bps0

```

```

bpsn0 = ggplot(vowel, aes(x = SpeakerNumber, y = F0, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn0

bps1 = ggplot(vowel, aes(x=Sex, y=F1, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps1
bpsn1 = ggplot(vowel, aes(x = SpeakerNumber, y = F1, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn1

bps2 = ggplot(vowel, aes(x=Sex, y=F2, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps2
bpsn2 = ggplot(vowel, aes(x = SpeakerNumber, y = F2, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn2

bps3 = ggplot(vowel, aes(x=Sex, y=F3, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps3
bpsn3 = ggplot(vowel, aes(x = SpeakerNumber, y = F3, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn3

bps4 = ggplot(vowel, aes(x=Sex, y=F4, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps4
bpsn4 = ggplot(vowel, aes(x = SpeakerNumber, y = F4, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn4

bps5 = ggplot(vowel, aes(x=Sex, y=F5, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps5
bpsn5 = ggplot(vowel, aes(x = SpeakerNumber, y = F5, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn5

bps6 = ggplot(vowel, aes(x=Sex, y=F6, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps6
bpsn6 = ggplot(vowel, aes(x = SpeakerNumber, y = F6, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn6

bps7 = ggplot(vowel, aes(x=Sex, y=F7, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps7
bpsn7 = ggplot(vowel, aes(x = SpeakerNumber, y = F7, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn7

```

```

bps8 = ggplot(vowel, aes(x=Sex, y=F8, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps8
bpsn8= ggplot(vowel, aes(x = SpeakerNumber, y = F8, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn8

bps9 = ggplot(vowel, aes(x=Sex, y=F9, fill = SpeakerNumber))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bps9
bpsn9 = ggplot(vowel, aes(x = SpeakerNumber, y = F9, fill = Sex))
+ geom_boxplot(outlier.colour="red",outlier.shape=8,outlier.size=4)
bpsn9

#####
#####

# Test de Shapiro-Wilks vs Kolmogorov-Smirnov (Corrección de Lillie)
library(nortest)
library(car)
# Variable F0.
# Shapiro: El bajo p-valor (9.807e-5) nos hace rechazar la hipótesis de normalidad
shapiro.test(vowel$F0)
# Lillie p-valor 0.000719 Rechazamos la normalidad
lillie.test(vowel$F0)
qqPlot(vowel$F0)

# Variable F1.
# Shapiro:pvalue (0.0312) rechaza normalidad
shapiro.test(vowel$F1)
# Lillie: pvalue 0.2628. No podemos rechazar la hipótesis de normalidad.
lillie.test(vowel$F1)
# Comprobamos que es bastante próxima a la normal con un qqPlot
qqPlot(vowel$F1)
ggplot(vowel, aes(x=F1)) + geom_histogram(aes(y=..density..),binwidth = binwd(vowel$F1))
+ stat_function(fun=dnorm, args=list(mean=mean(vowel$F1),sd=sd(vowel$F1)))

# Variable F2.
# Shapiro:El bajo p-valor (4.245e-5) nos hace rechazar la hipótesis de normalidad
shapiro.test(vowel$F2)
#Lillie: p-value = 0.001264 rechazamos la normalidad
lillie.test(vowel$F2)

# Variable F3.
# Shapiro:El bajo p-valor (4.324e-9) nos hace rechazar la hipótesis de normalidad
shapiro.test(vowel$F3)
qqPlot(vowel$F3)
# Lillie: p-value = 5.169e-07. Rechazamos la normalidad
lillie.test(vowel$F3)

# Variable F4.
# Shapiro: p-valor (0.07073) mayor que 0.05, por lo que no podemos rechazar la hipótesis de normalidad

```

```

shapiro.test(vowel$F4)
qqPlot(vowel$F4)
# Lillie: p-value = 0.08258. No podemos rechazar la hipótesis de normalidad
lillie.test(vowel$F4)

# Variable F5.
#Shapiro: El bajo p-valor (5.435e-08) nos hace rechazar la hipótesis de normalidad
shapiro.test(vowel$F5)
# Lillie:p-value = 1.921e-08. Rechazamos la hipótesis de normalidad
lillie.test(vowel$F5)

# Variable F6.
# Shapiro:pvalue (0.0225)<0.05, rechazamos hipótesis de normalidad
shapiro.test(vowel$F6)
# Lillie: pvalue 0.09071, no podemos rechazar la hipótesis de normalidad
lillie.test(vowel$F6)
qqPlot(vowel$F6)

# Variable F7.
# Shapiro: pvalue (0.0005086)<0.05, rechazamos hipótesis de normalidad
shapiro.test(vowel$F7)
# Lillie:p-value = 0.0002122. Rechazamos la hipótesis de normalidad
lillie.test(vowel$F7)

# Variable F8.
# Shapiro:pvalue (0.001437)<0.05, rechazamos hipótesis de normalidad
shapiro.test(vowel$F8)
# Lillie: p-value = 0.1505. No podemos rechazar la hipótesis de normalidad
lillie.test(vowel$F8)
qqPlot(vowel$F8)

# Variable F9.
# Shapiro: pvalue (2.208e-12)<0.05, rechazamos hipótesis de normalidad
shapiro.test(vowel$F9)
# Lillie: p-value = 7.729e-14. Rechazamos la hipótesis de normalidad
lillie.test(vowel$F9)

## Por sexos

# Hombres

# pvalue (6.965e-05) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F0)
lillie.test(hombres$F0)
# pvalue (0.0002435) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F1)
lillie.test(hombres$F1)
# pvalue (0.0003589) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F2)
lillie.test(hombres$F2)
# pvalue (1.919e-07) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F3)

```

```

lillie.test(hombres$F3)
# pvalue (0.07804) > 0.05. Acepto hipótesis de normalidad
shapiro.test(hombres$F4)
lillie.test(hombres$F4)
qqPlot(hombres$F4)
# pvalue (7.718e-09) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F5)
lillie.test(hombres$F5)
# pvalue (3.306e-07) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F6)
lillie.test(hombres$F6) # p value 0.05132. No puedo rechazar la hipótesis según el test de Lillie
qqPlot(hombres$F6)
# pvalue (0.009972) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F7)
lillie.test(hombres$F7)
# pvalue (0.00119) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F8)
lillie.test(hombres$F8)
# pvalue (1.466e-15) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(hombres$F9)
lillie.test(hombres$F9)

# Mujeres
# pvalue (0.005136) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F0)
lillie.test(mujeres$F0)
# pvalue (0.02526) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F1)
lillie.test(mujeres$F1)
# pvalue (1.284e-05) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F2)
lillie.test(mujeres$F2)
# pvalue (3.099e-07) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F3)
lillie.test(mujeres$F3)
# pvalue (0.1163) > 0.05. No puedo rechazar la hipótesis de normalidad
shapiro.test(mujeres$F4)
lillie.test(mujeres$F4) #pvalue 0.4692. No puedo rechazar la hipótesis de normalidad
qqPlot(mujeres$F4)
# pvalue (0.04365) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F5)
lillie.test(mujeres$F5) # pvalue 0.08174 > 0.05. No podemos rechazar la hipótesis de normalidad
qqPlot(mujeres$F5)
# pvalue (0.09697) > 0.05. No podemos rechazar la hipótesis de normalidad
shapiro.test(mujeres$F6)
lillie.test(mujeres$F6) # pvalue 0.26. No puedo rechazar la hipótesis de normalidad
qqPlot(mujeres$F6)
# pvalue (8.115e-07) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F7)
lillie.test(mujeres$F7)
# pvalue (0.003301) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F8)
lillie.test(mujeres$F8) # pvalue 0.1125. No puedo rechazar la hipótesis de normalidad

```

```

qqPlot(mujeres$F8)
# pvalue (9.116e-08) < 0.05. Rechazo hipótesis de normalidad
shapiro.test(mujeres$F9)
lillie.test(mujeres$F9)
#####
##### Correlaciones entre variables
plot(vowel[,3:12])
plot(hombres[,3:12])
plot(mujeres[,3:12])
install.packages("corrplot")
library(corrplot)
corrplot(vowel[,3:12], type = "upper", order = "hclust",
          tl.col = "black", tl.srt = 45)

install.packages("PerformanceAnalytics")
library("PerformanceAnalytics")
chart.Correlation(vowel[,3:12], histogram=TRUE, pch=19)
chart.Correlation(hombres[,3:12],histrogram=TRUE, pch=19)
chart.Correlation(mujeres[,3:12],histrogram=TRUE, pch=19)

library(Hmisc)

res2<-rcorr(as.matrix(vowel[,3:12]))
corrmat = flattenCorrMatrix(res2$r, res2$P)
# Insignificant correlation are crossed
corrplot(res2$r, type="upper", order="hclust",
          p.mat = res2$P, sig.level = 0.01, insig = "blank")

res2h = rcorr(as.matrix(hombres[,3:12]))
corrmat = flattenCorrMatrix(res2h$r, res2h$P)
# Insignificant correlation are crossed
corrplot(res2h$r, type="upper", order="hclust",
          p.mat = res2h$P, sig.level = 0.01, insig = "blank")

res2m = rcorr(as.matrix(mujeres[,3:12]))
corrmat = flattenCorrMatrix(res2m$r, res2m$P)
# Insignificant correlation are crossed
corrplot(res2m$r, type="upper", order="hclust",
          p.mat = res2m$P, sig.level = 0.01, insig = "blank")

#####
# BOXPLOT POR CLASES
# vowel
boxplot(vowel$F0~vowel$Class,data=vowel)
boxplot(vowel$F1~vowel$Class,data=vowel)
boxplot(vowel$F2~vowel$Class,data=vowel)
boxplot(vowel$F3~vowel$Class,data=vowel)
boxplot(vowel$F4~vowel$Class,data=vowel)
boxplot(vowel$F5~vowel$Class,data=vowel)

```

```

boxplot(vowel$F6~vowel$Class,data=vowel)
boxplot(vowel$F7~vowel$Class,data=vowel)
boxplot(vowel$F8~vowel$Class,data=vowel)
boxplot(vowel$F9~vowel$Class,data=vowel)
# hombres
boxplot(hombres$F0~hombres$Class,data=hombres)
boxplot(hombres$F1~hombres$Class,data=hombres)
boxplot(hombres$F2~hombres$Class,data=hombres)
boxplot(hombres$F3~hombres$Class,data=hombres)
boxplot(hombres$F4~hombres$Class,data=hombres)
boxplot(hombres$F5~hombres$Class,data=hombres)
boxplot(hombres$F6~hombres$Class,data=hombres)
boxplot(hombres$F7~hombres$Class,data=hombres)
boxplot(hombres$F8~hombres$Class,data=hombres)
boxplot(hombres$F9~hombres$Class,data=hombres)
# mujeres
boxplot(mujeres$F0~mujeres$Class,data=mujeres)
boxplot(mujeres$F1~mujeres$Class,data=mujeres)
boxplot(mujeres$F2~mujeres$Class,data=mujeres)
boxplot(mujeres$F3~mujeres$Class,data=mujeres)
boxplot(mujeres$F4~mujeres$Class,data=mujeres)
boxplot(mujeres$F5~mujeres$Class,data=mujeres)
boxplot(mujeres$F6~mujeres$Class,data=mujeres)
boxplot(mujeres$F7~mujeres$Class,data=mujeres)
boxplot(mujeres$F8~mujeres$Class,data=mujeres)
boxplot(mujeres$F9~mujeres$Class,data=mujeres)

#####
#####

# TRANSFORMACIONES
install.packages("scales")
library("scales")

vowel$logF2 = log1p(rescale(vowel$F2))
# pvalue 0.4792. No rechazamos la hipótesis de normalidad
lillie.test(vowel$logF2)
qqPlot(vowel$logF2)
ggplot(vowel, aes(x=logF2)) + geom_histogram(aes(y=..density..),
binwidth = binwd(vowel$logF2))
+ stat_function(fun=dnorm, args=list(mean=mean(vowel$logF2),sd=sd(vowel$logF2)))

#####
#####

# CLASIFICACIÓN

# C.1 kNN
library(tidyverse)
library(caret)
library(scales)

```

```

library(class)
library(plyr)
library(ggplot2)
vowel = read.csv("./data/vowel/vowel.dat", header=FALSE, comment.char="@")
colnames(vowel) = c("TT", "SpeakerNumber", "Sex",
                    "F0", "F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "Class")
for (i in 4:13){
  vowel[,i] = rescale(vowel[,i])
}
accuracy_vowel = c()
for (i in seq(1,15,2)){
  train.index <- createDataPartition(vowel$Class, p = .7, list = FALSE)
  v.train <- vowel[ train.index,]
  v.test  <- vowel[-train.index,]
  pr <- knn(train=v.train,test=v.test,cl=v.train$Class,k=i)
  acc1 = sum(pr==v.test$Class)/nrow(v.test)
  accuracy_vowel = append(accuracy_vowel,acc1)
}

plot(x=seq(1,15,2),y=accuracy_vowel,xlab="Valores de k",
      ylab="Accuracy sobre vowel", main="Resultado kNN",ylim=c(0,1),type='o',col='blue')

## REGIONES CON KNN
train.index <- createDataPartition(vowel$Class, p = .7, list = FALSE)
v.train <- vowel[ train.index,]
v.test  <- vowel[-train.index,]
pr <- knn(train=v.train,test=v.test,cl=v.train$Class,k=3)
acc1 = sum(pr==v.test$Class)/nrow(v.test)
accuracy_vowel = append(accuracy_vowel,acc1)
plot.df = data.frame(v.test, predicted = pr)
plot.df1 = data.frame(x = plot.df$F0,
                      y = plot.df$F1,
                      predicted = plot.df$predicted)

find_hull = function(df) df[chull(df$x, df$y), ]
boundary = ddply(plot.df1, .variables = "predicted", .fun = find_hull)

ggplot(plot.df, aes(F0, F1, color = predicted, fill = predicted)) +
  geom_point(size = 5) +
  geom_polygon(data = boundary, aes(x,y), alpha = 0.5)

ggplot(plot.df, aes(F2, F6, color = predicted, fill = predicted)) +
  geom_point(size = 5) +
  geom_polygon(data = boundary, aes(x,y), alpha = 0.5)
#####
# Prueba por sexos

hombres = vowel %>% filter(Sex==0)
mujeres = vowel %>% filter(Sex==1)

acc_hombres = c()
for (i in c(1,3,5,7,9)){
  hombres.index = createDataPartition(hombres$Class, p = .7, list = FALSE)

```

```

h.train <- hombres[hombres.index,]
h.test = hombres[-hombres.index,]
modelo_hombres = knn(train=h.train,test = h.test, cl=h.train$Class, k=3)
acc_h = sum(modelo_hombres==h.test$Class)/nrow(h.test)
acc_hombres = append(acc_hombres,acc_h)
}

plot(x=c(1,3,5,7,9),y=acc_hombres,xlab="Valores de k",
      ylab="Accuracy sobre hombres",ylim=c(0,1), main="Resultado kNN",type='o',col='blue')

acc_mujeres = c()
for(i in c(1,3,5,7,9)){
  mujeres.index = createDataPartition(mujeres$Class, p = .7, list = FALSE)
  m.train <- mujeres[mujeres.index,]
  m.test = mujeres[-mujeres.index,]
  modelo_mujeres = knn(train=m.train,test = m.test, cl=m.train$Class, k=3)
  acc_m = sum(modelo_mujeres==m.test$Class)/nrow(m.test)
  acc_mujeres = append(acc_mujeres,acc_m)
}

plot(x=c(1,3,5,7,9),y=acc_mujeres,xlab="Valores de k",
      ylab="Accuracy sobre mujeres",ylim=c(0,1), main="Resultado kNN",type='o',col='blue')

# Cross-validation

nombre <- "./data/vowel/vowel"
run_knn_fold <- function(i, x, tt = "test",k_par) {
  file <- paste(x, "-10-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-10-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  pr <- knn(train=x_tra,test=test,cl=x_tra$Y,k=k_par)
  return(sum(pr==test$Y)/nrow(test))
}
kfolds_list_train = c()
kfolds_list_test = c()
for(i in seq(1,15,2)){
  acc_mean_train = mean(sapply(1:10,run_knn_fold,nombre,"train",i))
  acc_mean_test = mean(sapply(1:10,run_knn_fold,nombre,"test",i))
  kfolds_list_train = append(kfolds_list_train,acc_mean_train)
  kfolds_list_test = append(kfolds_list_test,acc_mean_test)
}

```

```

plot(x=seq(1,15,2),y=kfolds_list_train,xlab="Valores de k",
      ylab="Accuracy sobre vowel en train",ylim=c(0,1), main="Resultado kNN 10-kfoldCross validation",
      type='o',col='blue')
plot(x=seq(1,15,2),y=kfolds_list_test,xlab="Valores de k",
      ylab="Accuracy sobre vowel en test",ylim=c(0,1), main="Resultado kNN 10-kfoldCross validation",
      type='o',col='blue')

resultados_knn3_tr = sapply(1:10,run_knn_fold,nombre,"train",3)
resultados_knn1_tr = sapply(1:10,run_knn_fold,nombre,"train",1)
resultados_knn3_test = sapply(1:10,run_knn_fold,nombre,"test",3)
resultados_knn1_test = sapply(1:10,run_knn_fold,nombre,"test",1)
acc_mean_test_knn = mean(resultados_knn3_test)

#####
## C.2
# LDA
library(MASS)
# checks (ya hecho en EDA)
sapply(vowel[,3:13],shapiro.test)
sapply(vowel[,4:13],var)

train.index <- createDataPartition(vowel$Class, p = .7, list = FALSE)
v.train <- vowel[ train.index,]
v.test <- vowel[-train.index,]

v.train$TT= NULL
v.train$SpeakerNumber = NULL
v.test$TT = NULL
v.test$SpeakerNumber = NULL
v.train$Class = as.factor(v.train$Class)
v.test$Class = as.factor(v.test$Class)
vowel.lda.predict <- train(Class ~ ., method = "lda", data = v.train)
confusionMatrix(v.test$Class, predict(vowel.lda.predict, v.test))

#####
# Prueba por sexos

hombres = vowel %>% filter(Sex==0)
mujeres = vowel %>% filter(Sex==1)

# para hombres
train.index <- createDataPartition(hombres$Class, p = .7, list = FALSE)
v.train <- hombres[ train.index,]
v.test <- hombres[-train.index,]

v.train$TT= NULL
v.train$SpeakerNumber = NULL
v.train$Sex = NULL
v.test$TT = NULL
v.test$SpeakerNumber = NULL
v.test$Sex = NULL

```

```

v.train$Class = as.factor(v.train$Class)
v.test$Class = as.factor(v.test$Class)
hombres.lda.predict <- train(Class ~ ., method = "lda", data = v.train)
confusionMatrix(v.test$Class, predict(hombres.lda.predict, v.test))

# para mujeres
train.index <- createDataPartition(mujeres$Class, p = .7, list = FALSE)
v.train <- mujeres[ train.index,]
v.test <- mujeres[-train.index,]

v.train$TT= NULL
v.train$SpeakerNumber = NULL
v.train$Sex = NULL
v.test$TT = NULL
v.test$SpeakerNumber = NULL
v.test$Sex = NULL
v.train$Class = as.factor(v.train$Class)
v.test$Class = as.factor(v.test$Class)
mujeres.lda.predict <- train(Class ~ ., method = "lda", data = v.train)
confusionMatrix(v.test$Class, predict(mujeres.lda.predict, v.test))

install.packages("klaR")
library(klaR)
X11(width=15, height=15)
partimat(Class ~F0+F1+F2+F3, data = v.test, method = "lda")

# Cross-validation

nombre <- "./data/vowel/vowel"
run_lda_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-10-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-10-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  x_tra$X1 = NULL
  x_tra$X2 = NULL
  test$X1 = NULL
  test$X2 = NULL
  x_tra$Y = as.factor(x_tra$Y)
  test$Y = as.factor(test$Y)
  modelo <- train(Y ~ ., method = "lda", data = x_tra)
  pr = predict(modelo,test)
}

```

```

    return(sum(pr==test$Y)/nrow(test))
}
acc_mean_train_lda = mean(sapply(1:10,run_lda_fold,nombre,"train"))
resultados_lda_tr = sapply(1:10,run_lda_fold,nombre,"train")
resultados_lda_test = sapply(1:10,run_lda_fold,nombre,"test")
acc_mean_test_lda = mean(resultados_lda_test)

#####
# C.3
# QDA

# CHECKS: Misma varianza entre elementos de la misma clase

for(i in 0:10){
  aux = vowel %>% filter(Class == i)
  print(paste("Clase ",i))
  print(sapply(aux[,4:13],var))
}

train.index <- createDataPartition(vowel$Class, p = .7, list = FALSE)
v.train <- vowel[ train.index,]
v.test <- vowel[-train.index,]

v.train$TT= NULL
v.train$SpeakerNumber = NULL
v.test$TT = NULL
v.test$SpeakerNumber = NULL
v.train$Class = as.factor(v.train$Class)
v.test$Class = as.factor(v.test$Class)
vowel.lda.predict <- train(Class ~ ., method = "qda", data = v.train)
confusionMatrix(v.test$Class, predict(vowel.lda.predict, v.test))

library(klaR)
X11(width=15, height=15)
partimat(Class ~F0+F1+F2+F3, data = v.test, method = "qda")

# Cross-validation

nombre <- "./data/vowel/vowel"
run_qda_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-10-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-10-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
}
```

```

    }
  else {
    test <- x_tst
  }
x_tra$X1 = NULL
x_tra$X2 = NULL
test$X1 = NULL
test$X2 = NULL
x_tra$Y = as.factor(x_tra$Y)
test$Y = as.factor(test$Y)
modelo <- train(Y ~ ., method = "qda", data = x_tra)
pr = predict(modelo,test)
return(sum(pr==test$Y)/nrow(test))
}

acc_mean_train_qda = mean(sapply(1:10,run_qda_fold,nombre,"train"))
resultados_qda_tr = sapply(1:10,run_qda_fold,nombre,"train")
resultados_qda_test = sapply(1:10,run_qda_fold,nombre,"test")
acc_mean_test_qda = mean(resultados_qda_test)

#####
# C.4
# Comparación de algoritmos
# TEST
tabla = cbind(resultados_knn1_test,resultados_knn3_test
              ,resultados_lda_test,resultados_qda_test)
tabla_resultados = as.data.frame(tabla,col.names=c("1NN", "3NN","LDA","QDA"))

# COMPARACIONES CON PARES: TEST DE WILCOXON
## 1) 1NN - LDA

wilc_1_3 = cbind(tabla_resultados[,1],tabla_resultados[,3])
colnames(wilc_1_3) <- c(colnames(tabla_resultados)[1], colnames(tabla_resultados)[3])
head(wilc_1_3)

K1NNvsLDAtst = wilcox.test(wilc_1_3[,1],wilc_1_3[,2]
                            ,alternative = "two.sided",paired=TRUE)
Rmas = K1NNvsLDAtst$statistic
pvalue = K1NNvsLDAtst$p.value
K1NNvsLDAtst = wilcox.test(wilc_1_3[,2],wilc_1_3[,1]
                            ,alternative = "two.sided",paired=TRUE)
Rmenos = K1NNvsLDAtst$statistic
Rmas
Rmenos
pvalue

## 2) 1NN - QDA

wilc_1_4 = cbind(tabla_resultados[,1],tabla_resultados[,4])
colnames(wilc_1_4) <- c(colnames(tabla_resultados)[1], colnames(tabla_resultados)[4])
head(wilc_1_4)

```

```

K1NNvsQDAst = wilcox.test(wilc_1_4[,1],wilc_1_4[,2],
                           alternative = "two.sided",paired=TRUE)
Rmas = K1NNvsQDAst$statistic
pvalue = K1NNvsQDAst$p.value
K1NNvsQDAst = wilcox.test(wilc_1_4[,2],wilc_1_4[,1],
                           alternative = "two.sided",paired=TRUE)
Rmenos = K1NNvsQDAst$statistic
Rmas
Rmenos
pvalue

## 3) 3NN- LDA

wilc_2_3 = cbind(tabla_resultados[,2],tabla_resultados[,3])
colnames(wilc_2_3) <- c(colnames(tabla_resultados)[2], colnames(tabla_resultados)[3])
head(wilc_2_3)

K3NNvsLDAst = wilcox.test(wilc_2_3[,1],wilc_2_3[,2]
                           ,alternative = "two.sided",paired=TRUE)
Rmas = K3NNvsLDAst$statistic
pvalue = K3NNvsLDAst$p.value
K3NNvsLDAst = wilcox.test(wilc_2_3[,2],wilc_2_3[,1]
                           ,alternative = "two.sided",paired=TRUE)
Rmenos = K3NNvsLDAst$statistic
Rmas
Rmenos
pvalue

## 4) 3NN - QDA

wilc_2_4 = cbind(tabla_resultados[,2],tabla_resultados[,4])
colnames(wilc_2_4) <- c(colnames(tabla_resultados)[2], colnames(tabla_resultados)[4])
head(wilc_2_4)

K3NNvsQDAst = wilcox.test(wilc_2_4[,1],wilc_2_4[,2]
                           ,alternative = "two.sided",paired=TRUE)
Rmas = K3NNvsQDAst$statistic
pvalue = K3NNvsQDAst$p.value
K3NNvsQDAst = wilcox.test(wilc_2_4[,2],wilc_2_4[,1]
                           ,alternative = "two.sided",paired=TRUE)
Rmenos = K3NNvsQDAst$statistic
Rmas
Rmenos
pvalue

## 5) LDA-QDA

wilc_3_4 = cbind(tabla_resultados[,3],tabla_resultados[,4])
colnames(wilc_3_4) <- c(colnames(tabla_resultados)[3]
                        , colnames(tabla_resultados)[4])
head(wilc_3_4)

```

```

LDAvsQDAst = wilcox.test(wilc_3_4[,1],wilc_3_4[,2]
                         ,alternative = "two.sided",paired=TRUE)
Rmas = LDAvsQDAst$statistic
pvalue = LDAvsQDAst$p.value
LDAvsQDAst = wilcox.test(wilc_3_4[,2],wilc_3_4[,1]
                         ,alternative = "two.sided",paired=TRUE)
Rmenos = LDAvsQDAst$statistic
Rmas
Rmenos
pvalue

# Comparativa general (Friedman)
test_friedman <- friedman.test(as.matrix(tabla_resultados))
test_friedman

# Post-hoc Holm
tam <- dim(tabla_resultados)
groups <- rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tabla_resultados),
                      groups, p.adjust = "holm", paired = TRUE)

# TRAIN
tabla = cbind(resultados_knn1_tr,resultados_knn3_tr,resultados_lda_tr,resultados_qda_tr)
tabla_resultados_tr = as.data.frame(tabla,col.names=c("1NN", "3NN","LDA","QDA"))

# COMPARACIONES CON PARES: TEST DE WILCOXON
## 1) 1NN - LDA

wilc_1_3 = cbind(tabla_resultados_tr[,1],tabla_resultados_tr[,3])
colnames(wilc_1_3) <- c(colnames(tabla_resultados_tr)[1]
                        , colnames(tabla_resultados_tr)[3])
head(wilc_1_3)

K1NNvsLDAst = wilcox.test(wilc_1_3[,1],wilc_1_3[,2],
                           alternative = "two.sided",paired=TRUE)
Rmas = K1NNvsLDAst$statistic
pvalue = K1NNvsLDAst$p.value
K1NNvsLDAst = wilcox.test(wilc_1_3[,2],wilc_1_3[,1],
                           alternative = "two.sided",paired=TRUE)
Rmenos = K1NNvsLDAst$statistic
Rmas
Rmenos
pvalue

## 2) 1NN - QDA

wilc_1_4 = cbind(tabla_resultados_tr[,1],tabla_resultados_tr[,4])
colnames(wilc_1_4) <- c(colnames(tabla_resultados_tr)[1],
                        colnames(tabla_resultados_tr)[4])
head(wilc_1_4)

```

```

K1NNvsQDAst = wilcox.test(wilc_1_4[,1],wilc_1_4[,2],
                           alternative = "two.sided",paired=TRUE)
Rmas = K1NNvsQDAst$statistic
pvalue = K1NNvsQDAst$p.value
K1NNvsQDAst = wilcox.test(wilc_1_4[,2],wilc_1_4[,1],
                           alternative = "two.sided",paired=TRUE)
Rmenos = K1NNvsQDAst$statistic
Rmas
Rmenos
pvalue

## 3) 3NN- LDA

wilc_2_3 = cbind(tabla_resultados_tr[,2],tabla_resultados_tr[,3])
colnames(wilc_2_3) <- c(colnames(tabla_resultados_tr)[2],
                        colnames(tabla_resultados_tr)[3])
head(wilc_2_3)

K3NNvsLDAst = wilcox.test(wilc_2_3[,1],wilc_2_3[,2],
                           alternative = "two.sided",paired=TRUE)
Rmas = K3NNvsLDAst$statistic
pvalue = K3NNvsLDAst$p.value
K3NNvsLDAst = wilcox.test(wilc_2_3[,2],wilc_2_3[,1],
                           alternative = "two.sided",paired=TRUE)
Rmenos = K3NNvsLDAst$statistic
Rmas
Rmenos
pvalue

## 4) 3NN - QDA

wilc_2_4 = cbind(tabla_resultados_tr[,2],tabla_resultados_tr[,4])
colnames(wilc_2_4) <- c(colnames(tabla_resultados_tr)[2],
                        colnames(tabla_resultados_tr)[4])
head(wilc_2_4)

K3NNvsQDAst = wilcox.test(wilc_2_4[,1],wilc_2_4[,2],
                           alternative = "two.sided",paired=TRUE)
Rmas = K3NNvsQDAst$statistic
pvalue = K3NNvsQDAst$p.value
K3NNvsQDAst = wilcox.test(wilc_2_4[,2],wilc_2_4[,1],
                           alternative = "two.sided",paired=TRUE)
Rmenos = K3NNvsQDAst$statistic
Rmas
Rmenos
pvalue

## 5) LDA-QDA

wilc_3_4 = cbind(tabla_resultados_tr[,3],tabla_resultados_tr[,4])
colnames(wilc_3_4) <- c(colnames(tabla_resultados_tr)[3], colnames(tabla_resultados_tr)[4])

```

```

head(wilc_3_4)

LDAvsQDATst = wilcox.test(wilc_3_4[,1],wilc_3_4[,2],
                           alternative = "two.sided",paired=TRUE)
Rmas = LDAvsQDATst$statistic
pvalue = LDAvsQDATst$p.value
LDAvsQDATst = wilcox.test(wilc_3_4[,2],wilc_3_4[,1],
                           alternative = "two.sided",paired=TRUE)
Rmenos = LDAvsQDATst$statistic
Rmas
Rmenos
pvalue

# Comparativa general (Friedman)
test_friedman_tr <- friedman.test(as.matrix(tabla_resultados_tr))
test_friedman_tr

# Post-hoc Holm
tam <- dim(tabla_resultados_tr)
groups <- rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tabla_resultados_tr),
                      groups, p.adjust = "holm", paired = TRUE)

```

# Código Wankara

Luis Balderas Ruiz

```
#####
# INTRODUCCIÓN A LA CIENCIA DE DATOS
# Autor: Luis Balderas Ruiz
# EDA+Regresion
# Dataset: wankara (01/01/1994 to 28/05/1998)
#####

library(ggplot2)
library(tidyverse)

binwd = function(data){
  size = length(data)
  dt = sd(data)
  cr = size^(1/3)
  return(1/(cr)*dt*3.49)
}
wankara = read.csv("./data/wankara/wankara.dat",header=FALSE, comment.char = "@")
colnames(wankara) = c("Max_temperature", "Min_temperature",
  "Dewpoint", "Precipitation", "Sea_level_pressure", "Standard_pressure",
  "Visibility", "Wind_speed", "Max_wind_speed", "Mean_temperature")

# Resumen estadístico
summary(wankara)

# Visualización de las variables respecto de mean_temperature
temp <- wankara
plotY <- function (x,y) {
  plot(temp[,y]-temp[,x], xlab=paste(names(temp)[x]),
    ylab=names(temp)[y])
}
par(mfrow=c(3,4)) #Si margin too large => (2,3)
x <- sapply(1:(dim(temp)[2]-1), plotY, dim(temp)[2])
par(mfrow=c(1,1))

#####
# HISTOGRAMAS

library(e1071)
# Max-temperature
skewness(wankara$Max_temperature)
kurtosis(wankara$Max_temperature)
ggplot(data=wankara, aes(x=Max_temperature)) +
  geom_histogram(binwidth = binwd(wankara$Max_temperature),fill="blue") +
  ggtitle("Histograma de temperatura máxima") +
  labs(x="Temperatura máxima", y="Count\nof Records")
```

```

# Min-temperature
skewness(wankara$Min_temperature)
kurtosis(wankara$Min_temperature)
ggplot(data=wankara, aes(x=Min_temperature)) +
  geom_histogram(binwidth = binwd(wankara$Min_temperature),fill="blue") +
  ggtitle("Histograma de temperatura mínima") +
  labs(x="Temperatura mínima", y="Count\nof Records")

# Dewpoint
skewness(wankara$Dewpoint)
kurtosis(wankara$Dewpoint)
ggplot(data=wankara, aes(x=Dewpoint)) +
  geom_histogram(binwidth = binwd(wankara$Dewpoint),fill="blue") +
  ggtitle("Histograma Dewpoint") +
  labs(x="Dewpoint", y="Count\nof Records")

# Precipitation
skewness(wankara$Precipitation)
kurtosis(wankara$Precipitation)
ggplot(data=wankara, aes(x=Precipitation)) +
  geom_histogram(binwidth = binwd(wankara$Precipitation),fill="blue") +
  ggtitle("Histograma Precipitaciones") +
  labs(x="Precipitaciones", y="Count\nof Records")

# Sea level pressure
skewness(wankara$Sea_level_pressure)
kurtosis(wankara$Sea_level_pressure)
ggplot(data=wankara, aes(x=Sea_level_pressure)) +
  geom_histogram(binwidth = binwd(wankara$Sea_level_pressure),fill="blue") +
  ggtitle("Histograma Sea_level_pressure") +
  labs(x="Sea_level_pressure", y="Count\nof Records")

# Standard pressure
skewness(wankara$Standard_pressure)
kurtosis(wankara$Standard_pressure)
ggplot(data=wankara, aes(x=Standard_pressure)) +
  geom_histogram(binwidth = binwd(wankara$Standard_pressure),fill="blue") +
  ggtitle("Histograma Standard pressure") +
  labs(x="Standard pressure", y="Count\nof Records")

# Visibility
skewness(wankara$Visibility)
kurtosis(wankara$Visibility)
ggplot(data=wankara, aes(x=Visibility)) +
  geom_histogram(binwidth = binwd(wankara$Visibility),fill="blue") +
  ggtitle("Histograma Visibility") +
  labs(x="Visibility", y="Count\nof Records")

# Wind speed
skewness(wankara$Wind_speed)
kurtosis(wankara$Wind_speed)
ggplot(data=wankara, aes(x=Wind_speed)) +
  geom_histogram(binwidth = binwd(wankara$Wind_speed),fill="blue") +

```

```

ggtitle("Histograma Wind speed") +
  labs(x="Wind speed", y="Count\nof Records")

# Max Wind speed
skewness(wankara$Max_wind_speed)
kurtosis(wankara$Max_wind_speed)
ggplot(data=wankara, aes(x=Max_wind_speed)) +
  geom_histogram(binwidth = binwd(wankara$Max_wind_speed),fill="blue") +
  ggtitle("Histograma Max Wind speed") +
  labs(x="Max Wind speed", y="Count\nof Records")

# Histograma temperatura media
skewness(wankara$Mean_temperature)
kurtosis(wankara$Mean_temperature)
ggplot(data=wankara, aes(x=Mean_temperature)) +
  geom_histogram(binwidth=binwd(wankara$Mean_temperature),fill="blue") +
  ggtitle("Histograma de temperatura media") +
  labs(x="Temperatura média", y="Count\nof Records")

#####
# VALORES PERDIDOS

wankara[is.na(wankara)]

#####
# OUTLIERS
install.packages("outliers")
library(outliers)

sapply(wankara,outlier)
sapply(wankara,outlier,opposite=TRUE)

#####
# DIVISIÓN POR MESES

# Días en cada mes
dias_mes = c(31,28,31,30,31,30,31,31,30,31,30,31)
# Días del dataset
dias = rep(dias_mes,5)[1:53]
# 1996 fue bisiesto
dias[12*3+2] = 29

max_temp = c()
min_temp = c()
dewp = c()
precip = c()
slp = c()
sp = c()
visib = c()
Ws = c()
Msp = c()

```

```

Mean_temp = c()
actual = 1
for(i in 1:53){
  if(i == 53){
    max_temp = append(max_temp, mean(wankara$Max_temperature[actual:1609]))
    min_temp = append(min_temp,mean(wankara$Min_temperature[actual:1609]))
    dewp = append(dewp, mean(wankara$Dewpoint[actual:1609]))
    precip = append(precip, mean(wankara$Precipitation[actual:1609]))
    slp = append(slp,mean(wankara$Sea_level_pressure[actual:1609]))
    sp = append(sp, mean(wankara$Standard_pressure[actual:1609]))
    visib = append(visib, mean(wankara$Visibility[actual:1609]))
    Ws = append(Ws, mean(wankara$Wind_speed[actual:1609]))
    Msp = append(Msp, mean(wankara$Max_wind_speed[actual:1609]))
    Mean_temp = append(Mean_temp, mean(wankara$Mean_temperature[actual:1609]))
  }
  else{
    max_temp = append(max_temp, mean(wankara$Max_temperature[actual:actual+dias[i]-1]))
    min_temp = append(min_temp,mean(wankara$Min_temperature[actual:actual+dias[i]-1]))
    dewp = append(dewp, mean(wankara$Dewpoint[actual:actual+dias[i]-1]))
    precip = append(precip, mean(wankara$Precipitation[actual:actual+dias[i]-1]))
    slp = append(slp,mean(wankara$Sea_level_pressure[actual:actual+dias[i]-1]))
    sp = append(sp, mean(wankara$Standard_pressure[actual:actual+dias[i]-1]))
    visib = append(visib, mean(wankara$Visibility[actual:actual+dias[i]-1]))
    Ws = append(Ws, mean(wankara$Wind_speed[actual:actual+dias[i]-1]))
    Msp = append(Msp, mean(wankara$Max_wind_speed[actual:actual+dias[i]-1]))
    Mean_temp = append(Mean_temp, mean(wankara$Mean_temperature[actual:actual+dias[i]-1]))
    actual = actual+dias[i]
    print(actual)
  }
}
meses = rep(month.abb,5)[1:53]
j=94
for(i in 1:53){
  if(i%>1 == 1 && i > 12){
    j = j+1
  }
  meses[i] = paste(meses[i],j,sep="")
}
df = as.data.frame(cbind(meses, max_temp,min_temp))
df$meses = factor(df$meses,levels=df$meses)
ggplot(df[1:12,],aes(x=meses, y=max_temp))+geom_point()
ggplot(df[1:12,],aes(x=meses,y=min_temp)) + geom_point()
# Los datos no parecen tener un orden cronológico

#####
# Normalidad
library(nortest)
library(car)

# Ninguna variable es normal
sapply(wankara,shapiro.test)
sapply(wankara,lillie.test)
sapply(wankara,qqPlot)

```

```

#####
# HIPÓTESIS

# Hipótesis: mayor temperatura máxima, mayor temperatura media
ggplot(data=wankara, aes(x=wankara$Max_temperature, y=wankara$Mean_temperature)) +
  geom_point(alpha=.4, size=4, color="#880011") +
  ggtitle("Temperatura máxima vs Temperatura media") +
  labs(x="Temperatura máxima", y="Temperatura media")

# Hipótesis: mayor temperatura mínima, mayor temperatura media
ggplot(data=wankara, aes(x=wankara$Min_temperature, y=wankara$Mean_temperature)) +
  geom_point(alpha=.4, size=4, color="#880011") +
  ggtitle("Temperatura mínima vs Temperatura media") +
  labs(x="Temperatura mínima", y="Temperatura media")

#####
# CORRELACIÓN

install.packages("PerformanceAnalytics")
library("PerformanceAnalytics")
chart.Correlation(wankara, histogram=TRUE, pch=19)

#####
# REESCALADO

library("scales")
wankara_scale = sapply(wankara,rescale)
wankara_scale = as.data.frame(wankara_scale)
summary(wankara_scale)

#####
# R.1
# Modelo lineal simple con la variable con más correlación: Max_temperature
fit_mls1 = lm(wankara_scale$Mean_temperature~wankara_scale$Max_temperature)
summary(fit_mls1)

par(mfrow=c(1,1))
plot(wankara_scale$Mean_temperature~wankara_scale$Max_temperature)
abline(fit_mls1,col="red")
confint(fit_mls1)

# Error cuadrático medio
yprime=predict(fit_mls1,data.frame(Max_temp=wankara_scale$Max_temperature))
sqrt(sum(abs(wankara_scale$Mean_temperature-yprime)^2)/length(yprime))

# Cross-validation

nombre <- "./data/wankara/wankara"
run_lm1_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
}

```

```

x_tst <- read.csv(file, comment.char="@", header=FALSE)
In <- length(names(x_tra)) - 1
names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
names(x_tra)[In+1] <- "Y"
names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
names(x_tst)[In+1] <- "Y"
if (tt == "train") {
  test <- x_tra
}
else {
  test <- x_tst
}
fitMulti=lm(Y~X1,x_tra)
yprime=predict(fitMulti,test)
sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
resultados_mls1_train = sapply(1:5,run_lm1_fold,nombre,"train")
resultados_mls1_test = sapply(1:5,run_lm1_fold,nombre,"test")
lmMSEtrain1<-mean(resultados_mls1_train)
lmMSEtest1<-mean(resultados_mls1_test)

# Modelo lineal simple con la variable con más correlación: Min_temperature
fit_mls2 = lm(wankara_scale$Mean_temperature~wankara_scale$Min_temperature)
summary(fit_mls2)

par(mfrow=c(1,1))
plot(wankara_scale$Mean_temperature~wankara_scale$Min_temperature)
abline(fit_mls2,col="red")
confint(fit_mls2)

# Error cuadrático medio
yprime=predict(fit_mls2,data.frame(Max_temp=wankara_scale$Min_temperature))
sqrt(sum(abs(wankara_scale$Mean_temperature-yprime)^2)/length(yprime))

# Cross-validation

nombre <- "./data/wankara/wankara"
run_lm2_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
}

```

```

}

fitMulti=lm(Y~X2,x_tra)
yprime=predict(fitMulti,test)
sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}

resultados_mls2_train = sapply(1:5,run_lm2_fold,nombre,"train")
resultados_mls2_test = sapply(1:5,run_lm2_fold,nombre,"test")
lmMSEtrain2<-mean(resultados_mls2_train)
lmMSEtest2<-mean(resultados_mls2_test)

# Dewpoint
fit_mls3 = lm(wankara_scale$Mean_temperature~wankara_scale$Dewpoint)
summary(fit_mls3)

par(mfrow=c(1,1))
plot(wankara_scale$Mean_temperature~wankara_scale$Dewpoint)
abline(fit_mls3,col="red")
confint(fit_mls3)

# Error cuadrático medio
yprime=predict(fit_mls3,data.frame(Max_temp=wankara_scale$Dewpoint))
sqrt(sum(abs(wankara_scale$Mean_temperature-yprime)^2)/length(yprime))

# Cross-validation

nombre <- "./data/wankara/wankara"
run_lm3_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  fitMulti=lm(Y~X3,x_tra)
  yprime=predict(fitMulti,test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
resultados_mls3_train = sapply(1:5,run_lm3_fold,nombre,"train")
resultados_mls3_test = sapply(1:5,run_lm3_fold,nombre,"test")
lmMSEtrain3<-mean(resultados_mls3_train)
lmMSEtest3<-mean(resultados_mls3_test)

# Sea_level_pressure

```

```

fit_mls4 = lm(wankara_scale$Mean_temperature~wankara_scale$Sea_level_pressure)
summary(fit_mls4)

par(mfrow=c(1,1))
plot(wankara_scale$Mean_temperature~wankara_scale$Sea_level_pressure)
abline(fit_mls4,col="red")
confint(fit_mls4)

# Error cuadrático medio
yprime=predict(fit_mls4,data.frame(SLP=wankara_scale$Sea_level_pressure))
sqrt(sum(abs(wankara_scale$Mean_temperature-yprime)^2)/length(yprime))

# Cross-validation

nombre <- "./data/wankara/wankara"
run_lm4_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  fitMulti=lm(Y~X5,x_tra)
  yprime=predict(fitMulti,test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
resultados_mls4_train = sapply(1:5,run_lm4_fold,nombre,"train")
resultados_mls4_test = sapply(1:5,run_lm4_fold,nombre,"test")
lmMSEtrain4<-mean(resultados_mls4_train)
lmMSEtest4<-mean(resultados_mls4_test)

# Visibility
fit_mls5 = lm(wankara_scale$Mean_temperature~wankara_scale$Visibility)
summary(fit_mls5)

par(mfrow=c(1,1))
plot(wankara_scale$Mean_temperature~wankara_scale$Visibility)
abline(fit_mls5,col="red")
confint(fit_mls5)

# Error cuadrático medio
yprime=predict(fit_mls5,data.frame(Vis=wankara_scale$Visibility))
sqrt(sum(abs(wankara_scale$Mean_temperature-yprime)^2)/length(yprime))

```

```

# Cross-validation

nombre <- "./data/wankara/wankara"
run_lm5_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  fitMulti=lm(Y~X7,x_tra)
  yprime=predict(fitMulti,test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
resultados_mls5_train = sapply(1:5,run_lm5_fold,nombre,"train")
resultados_mls5_test = sapply(1:5,run_lm5_fold,nombre,"test")
lmMSEtrain5<-mean(resultados_mls5_train)
lmMSEtest5<-mean(resultados_mls5_test)

#####
# R.2
# MODELO LINEAL MÚLTIPLE
# BACKWARD MODEL

# Elimino Precipitation por tener un p-valor de 0.885
fit_mlm2=lm(wankara_scale$Mean_temperature~.
             -Precipitation,data=wankara_scale)
summary(fit_mlm2)

# Elimino Sea_level_pressure por tener el mayor error standard
fit_mlm3=lm(wankara_scale$Mean_temperature~.-Precipitation
            -Sea_level_pressure,data=wankara_scale)
summary(fit_mlm3)

# Elimino Max wind speed por tener el mayor error standard
fit_mlm4 = lm(wankara_scale$Mean_temperature~.-Precipitation
              -Sea_level_pressure-Max_wind_speed,data=wankara_scale)
summary(fit_mlm4)

# Elimino visibility
fit_mlm5 = lm(wankara_scale$Mean_temperature~.-Precipitation
              -Sea_level_pressure-Max_wind_speed-Visibility,data=wankara_scale)
summary(fit_mlm5)

```

```

# Elimino standard pressure
fit_mlm6 = lm(wankara_scale$Mean_temperature~.-Standard_pressure-Precipitation
               -Sea_level_pressure-Max_wind_speed-Visibility,data=wankara_scale)
summary(fit_mlm6)

# Elimino Wind speed --> Modelo más interpretable
fit_mlm7 = lm(wankara_scale$Mean_temperature~.-Wind_speed-Standard_pressure
               -Precipitation-Sea_level_pressure-Max_wind_speed-Visibility,data=wankara_scale)
summary(fit_mlm7)

# INTERACCIONES

# Interacción entre la presión a nivel del mar y la estándar. Mejor resultado hasta ahora: 0.9899
fit_i1=lm(wankara_scale$Mean_temperature~.-Precipitation
           +Sea_level_pressure*Standard_pressure,data=wankara_scale)
summary(fit_i1)

# Interacción entre la temperatura mínima y dewpoint
fit_i2=lm(wankara_scale$Mean_temperature~.-Precipitation
           +Min_temperature*Dewpoint,data=wankara_scale)
summary(fit_i2)

# Mejor resultado hasta el momento 0.99
fit_i3 = lm(wankara_scale$Mean_temperature~.-Precipitation
             +Min_temperature*Dewpoint+I(Dewpoint^2)
             -Dewpoint,data=wankara_scale)
summary(fit_i3)

# Eliminamos Visibility por su alto p-value
fit_i4 = lm(wankara_scale$Mean_temperature~.-Precipitation
             +Min_temperature*Dewpoint+I(Dewpoint^2)-Dewpoint
             -Visibility,data=wankara_scale)
summary(fit_i4)

# Mejor resultado hasta el momento, 0.9916
fit_i5 = lm(wankara_scale$Mean_temperature~.-Precipitation
             +I(Max_temperature^2)+Min_temperature*Dewpoint
             +I(Dewpoint^2)-Dewpoint,data=wankara_scale)
summary(fit_i5)

# Mejor resultado --> 0.9923
fit_i6 = lm(wankara_scale$Mean_temperature~.-Precipitation
             +I(Min_temperature^2)+I(Max_temperature^2)
             +Min_temperature*Dewpoint+I(Dewpoint^2)
             -Dewpoint,data=wankara_scale)
summary(fit_i6)

# 0.9923
fit_i7 = lm(wankara_scale$Mean_temperature~.-Precipitation
             +Max_wind_speed*Wind_speed+I(Min_temperature^2)
             +I(Max_temperature^2)+Min_temperature*Dewpoint
             +I(Dewpoint^2)-Dewpoint-Max_wind_speed,data=wankara_scale)
summary(fit_i7)

```

```

# Cálculo de MSE
yprime_i7 = predict(fit_i7,wankara_scale)
sqrt(sum(abs(wankara_scale$Mean_temperature-yprime_i7)^2)/length(yprime_i7))

# CV del model más satisfactorio
nombre <- "./data/wankara/wankara"
run_i7_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@"
                    , header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@"
                    , header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  fit_i7 = lm(Y~.-X4+X9*X8+I(X2^2)+I(X1^2)+X2*X3+I(X3^2)-X3-X7-X9,data=test)
  yprime=predict(fit_i7,test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
resultados_i7_train = sapply(1:5,run_i7_fold,nombre,"train")
resultados_i7_test = sapply(1:5,run_i7_fold,nombre,"test")
i7MSEtrain<-mean(resultados_i7_train)
i7MSEtest<-mean(resultados_i7_test)

# Resultados de las interacción 7 (mejor resultado)
plot(wankara_scale$Mean_temperature~wankara_scale$Max_temperature)
points(wankara_scale$Max_temperature,fitted(fit_i7),col="green",pch=20)

#####
# R.3
# KNN
install.packages("knn")
library("knn")
fitknn1 <- knn(wankara_scale$Mean_temperature ~ ., wankara_scale, wankara_scale)
names(fitknn1)

# Visualización
plot(wankara_scale$Mean_temperature~wankara_scale$Max_temperature)
points(wankara_scale$Max_temperature,fitknn1$fitted.values,col="blue",pch=20)

# ECM
yprime = fitknn1$fitted.values
sqrt(sum((wankara_scale$Mean_temperature-yprime)^2)/length(yprime)) #RMSE

```

```

# Uso el mejor resultado anterior
fitknn2 = kknn(wankara_scale$Mean_temperature~.
                -Precipitation+Max_wind_speed*Wind_speed+I(Min_temperature^2)
                +I(Max_temperature^2)+Min_temperature*Dewpoint+I(Dewpoint^2)
                -Dewpoint-Visibility-Max_wind_speed,wankara_scale,wankara_scale)
yprime = fitknn2$fitted.values
sqrt(sum((wankara_scale$Mean_temperature-yprime)^2)/length(yprime))

plot(wankara_scale$Mean_temperature~wankara_scale$Max_temperature)
points(wankara_scale$Max_temperature,fitknn2$fitted.values,col="red",pch=20)

# Modelo más interpretable anterior --> Mejor resultado aún
fitknn3 = kknn(wankara_scale$Mean_temperature~.-Wind_speed
                -Standard_pressure-Precipitation-Sea_level_pressure
                -Max_wind_speed-Visibility,wankara_scale,wankara_scale)
yprime = fitknn3$fitted.values
sqrt(sum((wankara_scale$Mean_temperature-yprime)^2)/length(yprime))
plot(wankara_scale$Mean_temperature~wankara_scale$Max_temperature)
points(wankara_scale$Max_temperature,fitknn3$fitted.values,col="green",pch=20)

#####
# R.4
# Comparación de algoritmos

nombre <- "./data/wankara/wankara"
run_lm_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE )
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE )
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="");
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="");
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") { test <- x_tra }
  else { test <- x_tst }
  fitMulti=lm(Y~,x_tra)
  yprime=predict(fitMulti,test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
lmMSEtrain<-sapply(1:5,run_lm_fold,nombre,"train")
medialmMSEtrain = mean(lmMSEtrain)
lmMSEtest<-sapply(1:5,run_lm_fold,nombre,"test")
medialmMSEtest = mean(lmMSEtest)

run_kknn_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE )
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE )
  In <- length(names(x_tra)) - 1

```

```

names(x_tra)[1:In] <- paste ("X", 1:In, sep="");
names(x_tra)[In+1] <- "Y"
names(x_tst)[1:In] <- paste ("X", 1:In, sep="");
names(x_tst)[In+1] <- "Y"
if (tt == "train") { test <- x_tra }
else { test <- x_tst }
fitKNN=kknn(Y~,x_tra,test)
yprime=fitKNN$fitted.values
sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
kknnMSEtrain<-sapply(1:5,run_kknn_fold,nombre,"train")
mediakknnMSEtrain = mean(kknnMSEtrain)
kknnMSEtest<-sapply(1:5,run_kknn_fold,nombre,"test")
mediakknnMSEtest= mean(kknnMSEtest)

# Random Forest
install.packages("randomForest")
library(randomForest)
run_rf_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE )
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE )
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="");
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="");
  names(x_tst)[In+1] <- "Y"
  if (tt == "train") { test <- x_tra }
  else { test <- x_tst }
  fitrf=randomForest(Y~,data=x_tra)
  yprime = predict(fitrf, newdata=test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
rfMSEtrain<-sapply(1:5,run_rf_fold,nombre,"train")
rfMSEtest<-sapply(1:5,run_rf_fold,nombre,"test")

# COMPARATIVA EN TEST

resultados <- read.csv("./data/regr_test_alumnos.csv")
tablatst <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatst) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatst) <- resultados[,1]
#leemos la tabla con los errores medios de entrenamiento
resultados <- read.csv("./data/regr_train_alumnos.csv")
tablatr <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatr) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatr) <- resultados[,1]

# Añadiendo a las tablas mis resultados
tablatst[17,1] = medialmMSEtest
tablatst[17,2] = mediakknnMSEtest

```

```

tablatr[17,1] = medialmMSEtrain
tablatr[17,2] = mediakknnMSEtrain

##lm (other) vs knn (ref)
# + 0.1 porque wilcox R falla para valores == 0 en la tabla
difs <- (tablatst[,1] - tablatst[,2]) / tablatst[,1]
wilc_1_2 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1),
                   ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_1_2)

LMvsKNNtst <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2],
                            alternative = "two.sided", paired=TRUE)
Rmas <- LMvsKNNtst$statistic
pvalue <- LMvsKNNtst$p.value
LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1],
                            alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsKNNtst$statistic
Rmas
Rmenos
pvalue

## lm (other) vs m5p (ref)
difs <- (tablatst[,1] - tablatst[,3]) / tablatst[,1]
wilc_1_3 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1),
                   ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_3) <- c(colnames(tablatst)[1], colnames(tablatst)[3])
head(wilc_1_3)

LMvsM5Ptst <- wilcox.test(wilc_1_3[,1], wilc_1_3[,2],
                            alternative = "two.sided", paired=TRUE)
Rmas <- LMvsM5Ptst$statistic
pvalue <- LMvsM5Ptst$p.value
LMvsM5Ptst <- wilcox.test(wilc_1_3[,2], wilc_1_3[,1],
                            alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsM5Ptst$statistic
Rmas
Rmenos
pvalue

## kknn (other) vs m5p (ref)
difs <- (tablatst[,2] - tablatst[,3]) / tablatst[,2]
wilc_2_3 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1),
                   ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_2_3) <- c(colnames(tablatst)[2], colnames(tablatst)[3])
head(wilc_2_3)

KKNNvsM5Ptst <- wilcox.test(wilc_2_3[,1], wilc_2_3[,2],
                             alternative = "two.sided", paired=TRUE)
Rmas <- KKNNvsM5Ptst$statistic
pvalue <- KKNNvsM5Ptst$p.value
KKNNvsM5Ptst <- wilcox.test(wilc_2_3[,2], wilc_2_3[,1],
                             alternative = "two.sided", paired=TRUE)

```

```

Rmenos <- KKNNvsM5Ptst$statistic
Rmas
Rmenos
pvalue

# Comparativa general con Friedman
test_friedman <- friedman.test(as.matrix(tablatst))
test_friedman

tam <- dim(tablatst)
groups <- rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatst), groups,
                      p.adjust = "holm", paired = TRUE)

# COMPARATIVAS EN TRAINING
# lm (other) vs kknn (reference)
difs_tr <- (tablatr[,1] - tablatr[,2]) / tablatr[,1]
wilc_1_2_tr <- cbind(ifelse (difs_tr<0, abs(difs_tr)+0.1, 0+0.1),
                      ifelse (difs_tr>0, abs(difs_tr)+0.1, 0+0.1))
colnames(wilc_1_2_tr) <- c(colnames(tablatr)[1], colnames(tablatr)[2])
head(wilc_1_2_tr)

LMvsKNNtr <- wilcox.test(wilc_1_2_tr[,1], wilc_1_2_tr[,2],
                           alternative = "two.sided", paired=TRUE)
Rmas_tr <- LMvsKNNtr$statistic
pvalue_tr <- LMvsKNNtr$p.value
LMvsKNNtr <- wilcox.test(wilc_1_2_tr[,2], wilc_1_2_tr[,1],
                           alternative = "two.sided", paired=TRUE)
Rmenos_tr <- LMvsKNNtr$statistic
Rmas_tr
Rmenos_tr
pvalue_tr

# lm (other) vs m5p (reference)
difs_tr <- (tablatr[,1] - tablatr[,3]) / tablatr[,1]
wilc_1_3_tr <- cbind(ifelse (difs_tr<0, abs(difs_tr)+0.1, 0+0.1),
                      ifelse (difs_tr>0, abs(difs_tr)+0.1, 0+0.1))
colnames(wilc_1_3_tr) <- c(colnames(tablatra)[1], colnames(tablatra)[2])
head(wilc_1_3_tr)

LMvsM5Ptr <- wilcox.test(wilc_1_3_tr[,1], wilc_1_3_tr[,2],
                           alternative = "two.sided", paired=TRUE)
Rmas_tr <- LMvsM5Ptr$statistic
pvalue_tr <- LMvsM5Ptr$p.value
LMvsM5Ptr <- wilcox.test(wilc_1_3_tr[,2], wilc_1_3_tr[,1],
                           alternative = "two.sided", paired=TRUE)
Rmenos_tr <- LMvsM5Ptr$statistic
Rmas_tr
Rmenos_tr
pvalue_tr

# kknn(other) vs m5p (reference)
difs_tr <- (tablatr[,2] - tablatr[,3]) / tablatr[,2]

```

```

wilc_2_3_tr <- cbind(ifelse (difs_tr<0, abs(difs_tr)+0.1, 0+0.1),
                      ifelse (difs_tr>0, abs(difs_tr)+0.1, 0+0.1))
colnames(wilc_2_3_tr) <- c(colnames(tablatra)[1], colnames(tablatra)[2])
head(wilc_2_3_tr)

KKNNvsM5Ptr <- wilcox.test(wilc_2_3_tr[,1], wilc_2_3_tr[,2],
                            alternative = "two.sided", paired=TRUE)
Rmas_tr <- KKNNvsM5Ptr$statistic
pvalue_tr <- KKNNvsM5Ptr$p.value
KKNNvsM5Ptr <- wilcox.test(wilc_2_3_tr[,2], wilc_2_3_tr[,1],
                            alternative = "two.sided", paired=TRUE)
Rmenos_tr <- KKNNvsM5Ptr$statistic
Rmas_tr
Rmenos_tr
pvalue_tr

# Comparativa conjunta
test_friedman_tr <- friedman.test(as.matrix(tablatr))
test_friedman_tr

tam <- dim(tablatst)
groups_ <- rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatr), groups,
                      p.adjust = "holm", paired = TRUE)

#####
# EXTRA: Comparación con algoritmos sobre los resultados del 5-fold añadiendo Random Forest

resultados_train = cbind(lmMSEtrain,kknnMSEtrain,rfMSEtrain)
tablatra = as.data.frame(resultados_train,
                         col.names=c("lm_MSE_train","kknn_MSE_train","rf_MSE_train"))
resultados_test = cbind(lmMSEtest,kknnMSEtest,rfMSEtest)
tablatst = as.data.frame(resultados_test,
                         col.names=c("lm_MSE_test","kknn_MSE_test","rf_MSE_test"))

## lm (other) vs knn (ref)
# + 0.1 porque wilcox R falla para valores == 0 en la tabla
difs <- (tablatst[,1] - tablatst[,2]) / tablatst[,1]
wilc_1_2 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1),
                   ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_1_2)

LMvsKNNtst <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2],
                            alternative = "two.sided", paired=TRUE)
Rmas <- LMvsKNNtst$statistic
pvalue <- LMvsKNNtst$p.value
LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1],
                            alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsKNNtst$statistic
Rmas
Rmenos
pvalue

```

```

##lm (other) vs rf (ref)
# + 0.1 porque wilcox R falla para valores == 0 en la tabla
difst <- (tablatst[,1] - tablatst[,3]) / tablatst[,1]
wilc_1_3 <- cbind(ifelse (difst<0, abs(difst)+0.1, 0+0.1),
                   ifelse (difst>0, abs(difst)+0.1, 0+0.1))
colnames(wilc_1_3) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_1_3)

LMvsRftst <- wilcox.test(wilc_1_3[,1], wilc_1_3[,2],
                           alternative = "two.sided", paired=TRUE)
Rmas <- LMvsRftst$statistic
pvalue <- LMvsRftst$p.value
LMvsRftst <- wilcox.test(wilc_1_3[,2], wilc_1_3[,1],
                           alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsRftst$statistic
Rmas
Rmenos
pvalue

##kknn (other) vs rf (ref)
# + 0.1 porque wilcox R falla para valores == 0 en la tabla
difst <- (tablatst[,2] - tablatst[,3]) / tablatst[,2]
wilc_2_3 <- cbind(ifelse (difst<0, abs(difst)+0.1, 0+0.1),
                   ifelse (difst>0, abs(difst)+0.1, 0+0.1))
colnames(wilc_2_3) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_2_3)

KKNNvsRftst <- wilcox.test(wilc_2_3[,1], wilc_2_3[,2],
                             alternative = "two.sided", paired=TRUE)
Rmas <- KKNNvsRftst$statistic
pvalue <- KKNNvsRftst$p.value
KKNNvsRftst <- wilcox.test(wilc_2_3[,2], wilc_2_3[,1],
                             alternative = "two.sided", paired=TRUE)
Rmenos <- KKNNvsRftst$statistic
Rmas
Rmenos
pvalue

# COMPARATIVA GENERAL TEST

test_friedman <- friedman.test(as.matrix(tablatst))
test_friedman

tam <- dim(tablatst)
groups <- rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatst), groups,
                      p.adjust = "holm", paired = TRUE)

# COMPARATIVA EN TRAINING

# EXAMINAMOS TRAINING PARA COMPROBAR SI HAY SOBREAPRENDIZAJE
difst_tr <- (tablatra[,1] - tablatra[,2]) / tablatra[,1]
wilc_1_2_tr <- cbind(ifelse (difst_tr<0, abs(difst_tr)+0.1, 0+0.1),

```

```

      ifelse (difs_tr>0, abs(difs_tr)+0.1, 0+0.1))
colnames(wilc_1_2_tr) <- c(colnames(tablatra)[1], colnames(tablatra)[2])
head(wilc_1_2_tr)

LMvsKNNtr <- wilcox.test(wilc_1_2_tr[,1], wilc_1_2_tr[,2],
                           alternative = "two.sided", paired=TRUE)
Rmas_tr <- LMvsKNNtr$statistic
pvalue_tr <- LMvsKNNtr$p.value
LMvsKNNtr <- wilcox.test(wilc_1_2_tr[,2], wilc_1_2_tr[,1],
                           alternative = "two.sided", paired=TRUE)
Rmenos_tr <- LMvsKNNtr$statistic
Rmas_tr
Rmenos_tr
pvalue_tr

# Comparativa general
test_friedman_tr <- friedman.test(as.matrix(tablatra))
test_friedman_tr

tam_tr <- dim(tablatra)
groups_tr <- rep(1:tam_tr[2], each=tam_tr[1])
pairwise.wilcox.test(as.matrix(tablatr), groups,
                      p.adjust = "holm", paired = TRUE)

```