

Preprocessing Univ

2025-10-01

Contents

Simetría de las variables	1
Outliers variables simétricas	2
Outliers variables no simétricas	3
Verificación por proporciones respecto a la variable salida	6
Detección de valores imposibles	8
Conclusión del análisis de outliers post imputación	8

El objetivo de este análisis es compararlo con el previo a la imputación, y comprobar si ha habido cambios.

Detección de outliers univariante: variables numéricas

Variables a estudiar:

```
varNum<-c("Age", "CreditScore", "Tenure","EstimatedSalary", "Balance", "NumOfProducts",  
          "TransactionFrequency","AvgTransactionAmount","DigitalEngagementScore",  
          "ComplaintsCount","NetPromoterScore")
```

Simetría de las variables

Primero haremos una comparación entre los estadísticos de simetría para cada variable.

```
tabla_comparacion <- data.frame(  
  Skewness_Antes = round(sapply(varNum, function(v) skewness(data[[v]], na.rm = TRUE)), 3),  
  Skewness_Despues = round(sapply(varNum, function(v) skewness(data_imputado[[v]])), 3),  
  Diferencia = round(sapply(varNum, function(v) {  
    skewness(data_imputado[[v]]) - skewness(data[[v]], na.rm = TRUE)  
  }), 3)  
)  
  
print(tabla_comparacion)
```

##	Skewness_Antes	Skewness_Despues	Diferencia
## Age	1.030	1.007	-0.023
## CreditScore	-0.065	-0.065	0.000
## Tenure	0.013	0.018	0.005
## EstimatedSalary	0.009	0.011	0.002
## Balance	-0.134	-0.140	-0.006
## NumOfProducts	0.749	0.731	-0.018
## TransactionFrequency	0.224	0.219	-0.005

## AvgTransactionAmount	1.623	1.780	0.158
## DigitalEngagementScore	-0.074	-0.084	-0.010
## ComplaintsCount	2.737	2.781	0.044
## NetPromoterScore	-0.650	-0.643	0.008

Los resultados son prácticamente idénticos y las diferencias asignables a la aleatoriedad. Se observa un cambio mayor en Avg transaction amount, aunque a nivel visual ambas distribuciones son idénticas, y en cualquier caso continúa siendo muy asimétrica.

El análisis de las distribuciones será el mismo que en el análisis previo, puesto que en la visualización de las distribuciones previas y posteriores a la imputación se observa que son prácticamente idénticas.

Ahora procedemos a dividir nuestras variables numéricas en simétricas y no simétricas para continuar con el análisis adecuado en cada caso:

```
symmetric<-c("TransactionFrequency", "Tenure", "EstimatedSalary", "CreditScore")
non_symmetric<-c("Age", "AvgTransactionAmount", "DigitalEngagementScore", "NetPromoterScore")
varNum
```

## [1] "Age"	"CreditScore"	"Tenure"
## [4] "EstimatedSalary"	"Balance"	"NumOfProducts"
## [7] "TransactionFrequency"	"AvgTransactionAmount"	"DigitalEngagementScore"
## [10] "ComplaintsCount"	"NetPromoterScore"	

Outliers variables simétricas

Procedemos a la detección de outliers para las variables simétricas. Para ello utilizaremos el rango intercuartílico (IQR), de modo que los valores que queden fuera de dicho intervalo serán reconocidos como outliers univariantes

```
data<-data_imputado
outlier_positions <- list()
IQROutlier <- function(variable, rmnas = TRUE) {
  IQ <- IQR(variable, na.rm = rmnas)
  intInf <- quantile(variable, probs = c(0.25, 0.75), na.rm = rmnas)[[1]] - 1.5*IQ
  intSup <- quantile(variable, probs = c(0.25, 0.75), na.rm = rmnas)[[2]] + 1.5*IQ
  posiciones <- which(variable >= intSup | variable <= intInf)
  if (length(posiciones) > 0) {
    cat("Existen outliers en las posiciones:", paste0(posiciones, collapse = ", "))
  } else {
    cat("No existen outliers")
  }
  return(posiciones)
}
for (var in symmetric) {
  cat("\nVariable:", var, "\n")
  pos <- IQROutlier(data[[var]])
  if (length(pos) > 0) {
    outlier_positions[[var]] <- pos
  }
}
```

```
##
```

```
## Variable: TransactionFrequency
## Existen outliers en las posiciones: 145, 236, 494, 921, 1497, 1581, 1769, 1917, 2110, 2117, 2181, 22
## Variable: Tenure
## No existen outliers
## Variable: EstimatedSalary
## No existen outliers
## Variable: CreditScore
## Existen outliers en las posiciones: 728, 1256, 1307, 1412, 1525, 1673, 3182, 3223, 4300, 4580, 5160,
```

```
outlier_positions
```

```
## $TransactionFrequency
## [1] 145 236 494 921 1497 1581 1769 1917 2110 2117 2181 2273 2617 2940 3104
## [16] 3148 3206 3596 3791 4454 4887 4943 5061 5193 5303 5698 5819 5987 6209 7282
## [31] 7320 7592 7960 8000 8416 8764 8969 9118 9776 9784 9953
##
## $CreditScore
## [1] 728 1256 1307 1412 1525 1673 3182 3223 4300 4580 5160 6612 7486 7634 9986
```

```
longitudes_s <- sapply(outlier_positions, length)
longitudes_s
```

```
## TransactionFrequency      CreditScore
##                41                15
```

Vemos como el resultado es exactamente el mismo que en el análisis previo, añadiendo aproximadamente un 30% a cada variable. Estos son los outliers que podíamos considerar como naturales. Es buena señal que el número aumente proporcionalmente al del número de datos al haber imputado el 30% de NA.

Outliers variables no simétricas

Utilizaremos el mismo método que en el análisis previo.

```
hampelOutlier <- function(variable, mad_constant = 1, threshold = 3.5) {
  med <- median(variable, na.rm = TRUE)
  mad_val <- mad(variable, constant = mad_constant, na.rm = TRUE)

  lower_bound <- med - threshold * mad_val
  upper_bound <- med + threshold * mad_val

  which((variable < lower_bound) | (variable > upper_bound))
}

outlier_positions_non_symmetric <- list()

for (var in non_symmetric) {
  outliers <- hampelOutlier(data[[var]])
  if (length(outliers) > 0) {
    outlier_positions_non_symmetric[[var]] <- outliers
  }
}

outlier_positions_non_symmetric
```

```

## $Age
## [1] 39 41 72 107 109 149 153 182 205 242 270 272 275 297 338
## [16] 379 384 406 407 430 440 446 461 463 472 480 504 509 511 530
## [31] 552 557 582 618 641 657 671 685 694 758 760 765 779 786 807
## [46] 835 839 849 869 875 880 890 896 915 922 928 961 990 1013 1015
## [61] 1016 1029 1073 1093 1099 1125 1141 1146 1194 1206 1216 1217 1221 1230 1234
## [76] 1260 1262 1267 1274 1275 1297 1298 1302 1310 1316 1326 1354 1363 1371 1386
## [91] 1394 1411 1422 1429 1432 1440 1442 1452 1483 1548 1551 1553 1569 1606 1612
## [106] 1620 1627 1636 1646 1652 1654 1679 1705 1711 1739 1750 1763 1777 1786 1803
## [121] 1816 1831 1859 1875 1878 1907 1926 1931 1955 1968 1969 1976 1988 1990 2000
## [136] 2004 2007 2011 2026 2030 2032 2036 2055 2103 2112 2142 2145 2150 2156 2166
## [151] 2173 2187 2188 2192 2235 2242 2251 2270 2284 2289 2320 2330 2342 2355 2376
## [166] 2392 2405 2421 2422 2442 2447 2476 2484 2499 2506 2526 2562 2566 2573 2611
## [181] 2629 2647 2674 2688 2695 2704 2708 2712 2722 2726 2729 2745 2767 2786 2792
## [196] 2809 2825 2850 2873 2878 2889 2931 2970 2988 3007 3030 3066 3075 3084 3097
## [211] 3104 3120 3122 3132 3135 3159 3182 3226 3232 3233 3241 3266 3275 3292 3302
## [226] 3308 3315 3356 3368 3399 3434 3450 3456 3483 3493 3547 3551 3560 3579 3588
## [241] 3636 3655 3660 3667 3675 3694 3707 3710 3725 3764 3785 3792 3800 3823 3839
## [256] 3861 3865 3880 3893 3900 3926 3930 3936 3946 3954 3961 3989 3997 4008 4030
## [271] 4069 4095 4103 4111 4115 4140 4144 4160 4170 4217 4259 4315 4319 4324 4330
## [286] 4356 4387 4430 4506 4510 4511 4534 4544 4557 4592 4617 4618 4619 4624 4640
## [301] 4646 4650 4655 4666 4704 4706 4709 4712 4742 4750 4760 4768 4829 4861 4882
## [316] 4899 4932 4940 4956 4985 4987 4997 5008 5013 5020 5022 5057 5081 5113 5125
## [331] 5127 5128 5166 5205 5221 5252 5272 5277 5279 5291 5293 5334 5347 5348 5355
## [346] 5389 5394 5407 5465 5491 5501 5515 5543 5624 5632 5636 5654 5701 5708 5723
## [361] 5749 5759 5789 5790 5807 5813 5876 5894 5903 5905 5908 5918 5933 5938 5950
## [376] 5972 5980 5995 6010 6025 6055 6057 6063 6064 6067 6081 6111 6119 6166 6167
## [391] 6168 6174 6191 6205 6207 6242 6250 6273 6308 6348 6356 6369 6370 6375 6388
## [406] 6400 6412 6461 6476 6477 6480 6485 6497 6513 6527 6532 6543 6544 6546 6549
## [421] 6563 6573 6577 6581 6608 6631 6660 6662 6666 6681 6688 6702 6717 6719 6725
## [436] 6747 6760 6763 6767 6777 6822 6832 6841 6878 6893 6903 6912 6923 6924 6957
## [451] 6964 6975 6980 6996 6999 7007 7060 7064 7096 7100 7109 7117 7132 7157 7164
## [466] 7174 7190 7191 7193 7223 7242 7250 7258 7260 7263 7282 7285 7321 7352 7393
## [481] 7400 7409 7438 7461 7502 7524 7534 7544 7581 7590 7594 7596 7598 7610 7616
## [496] 7624 7644 7650 7658 7665 7668 7681 7693 7696 7703 7718 7728 7736 7752 7782
## [511] 7792 7832 7849 7856 7885 7917 7923 7933 7940 7943 7974 7992 8003 8013 8016
## [526] 8051 8096 8100 8112 8129 8130 8150 8153 8178 8185 8195 8203 8214 8221 8241
## [541] 8248 8274 8294 8302 8310 8326 8335 8338 8350 8367 8416 8417 8420 8423 8465
## [556] 8479 8485 8490 8491 8505 8516 8548 8549 8576 8593 8614 8629 8631 8661 8662
## [571] 8671 8705 8774 8778 8781 8782 8808 8823 8849 8888 8914 8937 8938 8943 8961
## [586] 8962 8968 8973 8979 9058 9078 9079 9086 9091 9106 9107 9113 9141 9214 9228
## [601] 9241 9248 9250 9252 9285 9300 9315 9332 9353 9370 9409 9439 9518 9534 9540
## [616] 9547 9553 9613 9643 9651 9691 9697 9701 9705 9782 9785 9803 9815 9818 9820
## [631] 9846 9857 9869 9873 9893 9962
##
## $AvgTransactionAmount
## [1] 3 11 24 35 82 84 86 97 113 127 139 169 174 176 208
## [16] 213 251 264 265 269 276 309 311 336 351 373 413 443 467 469
## [31] 479 480 502 513 559 564 603 613 626 640 649 661 671 683 687
## [46] 691 710 713 736 738 771 776 800 821 822 825 827 880 895 899
## [61] 906 930 939 955 961 977 978 998 1048 1051 1056 1084 1117 1120 1126
## [76] 1132 1134 1148 1169 1174 1189 1214 1218 1222 1247 1253 1267 1269 1271 1288
## [91] 1291 1302 1314 1328 1347 1376 1380 1390 1402 1418 1457 1474 1490 1530 1559
## [106] 1582 1594 1606 1624 1626 1633 1654 1672 1685 1711 1748 1791 1799 1822 1836

```

```

## [121] 1839 1847 1859 1864 1877 1891 1923 1959 1968 2038 2060 2062 2081 2086 2104
## [136] 2111 2116 2141 2168 2196 2210 2233 2254 2289 2337 2366 2371 2372 2399 2402
## [151] 2403 2407 2411 2412 2417 2434 2499 2530 2537 2547 2557 2561 2584 2591 2601
## [166] 2615 2660 2680 2681 2686 2730 2751 2758 2765 2790 2791 2794 2812 2835 2850
## [181] 2902 2921 2942 2956 2958 2971 2996 3012 3017 3068 3070 3081 3088 3106 3108
## [196] 3111 3117 3145 3153 3156 3157 3168 3171 3179 3197 3216 3218 3222 3233 3287
## [211] 3307 3334 3336 3340 3367 3370 3390 3398 3426 3453 3494 3516 3527 3534 3583
## [226] 3588 3645 3663 3678 3702 3743 3757 3759 3819 3823 3870 3891 3893 3912 3913
## [241] 3916 3922 3933 3937 3970 3982 4004 4018 4028 4029 4034 4047 4060 4076 4100
## [256] 4125 4131 4136 4138 4142 4150 4175 4176 4198 4225 4241 4243 4269 4283 4293
## [271] 4317 4322 4327 4331 4334 4346 4347 4354 4376 4388 4418 4467 4498 4504 4513
## [286] 4543 4566 4609 4632 4644 4648 4654 4661 4671 4685 4701 4707 4725 4730 4735
## [301] 4757 4783 4800 4806 4842 4848 4853 4854 4858 4882 4892 4936 4940 4991 4992
## [316] 5007 5026 5027 5052 5075 5076 5146 5150 5165 5180 5181 5186 5189 5196 5219
## [331] 5222 5250 5251 5254 5259 5268 5275 5280 5293 5295 5300 5323 5338 5341 5357
## [346] 5358 5361 5412 5420 5441 5449 5479 5499 5519 5526 5567 5571 5575 5577 5583
## [361] 5596 5610 5611 5623 5635 5649 5693 5796 5801 5804 5810 5832 5859 5872 5881
## [376] 5883 5902 5906 5912 5922 5924 5927 5935 5941 5959 5974 5986 5994 6013 6027
## [391] 6029 6033 6035 6073 6074 6101 6116 6118 6139 6150 6198 6218 6237 6241 6259
## [406] 6399 6460 6466 6484 6493 6513 6523 6542 6546 6561 6577 6622 6654 6695 6710
## [421] 6713 6726 6729 6732 6761 6794 6798 6816 6842 6859 6860 6866 6889 6898 6905
## [436] 6976 6992 6993 7004 7008 7010 7015 7044 7048 7067 7073 7096 7104 7110 7149
## [451] 7164 7186 7187 7204 7217 7224 7258 7266 7278 7283 7284 7287 7299 7301 7311
## [466] 7316 7341 7349 7352 7361 7370 7378 7401 7407 7415 7433 7472 7473 7502 7508
## [481] 7515 7558 7559 7562 7576 7588 7597 7603 7618 7622 7643 7647 7675 7678 7691
## [496] 7698 7710 7719 7721 7724 7798 7853 7866 7892 7893 7918 7922 7944 7946 7948
## [511] 7990 8021 8022 8028 8033 8046 8056 8058 8071 8118 8138 8141 8144 8151 8160
## [526] 8205 8213 8284 8287 8288 8307 8315 8326 8476 8481 8493 8501 8508 8521 8574
## [541] 8576 8594 8606 8630 8672 8682 8687 8725 8740 8746 8774 8790 8816 8824 8836
## [556] 8839 8840 8851 8864 8876 8890 8956 8971 8985 8991 9002 9023 9071 9090 9132
## [571] 9134 9154 9171 9174 9199 9219 9221 9270 9273 9283 9299 9301 9303 9328 9330
## [586] 9341 9346 9352 9363 9365 9377 9379 9382 9413 9422 9425 9430 9445 9450 9480
## [601] 9489 9496 9503 9520 9523 9529 9535 9589 9597 9599 9632 9648 9649 9666 9669
## [616] 9672 9685 9688 9690 9706 9708 9730 9742 9745 9751 9753 9755 9756 9760 9765
## [631] 9766 9771 9781 9794 9820 9876 9879 9900 9932 9961 9973 9974
##
## $DigitalEngagementScore
## [1] 11 97 113 129 130 187 487 567 622 724 763 869 949 1098 1117
## [16] 1124 1194 1344 1346 1514 1683 1723 1930 1976 2038 2106 2405 2459 2470 2535
## [31] 2559 2739 2749 2848 2871 2989 3011 3098 3113 3142 3425 3426 3519 3544 3639
## [46] 3706 3712 3890 4018 4048 4093 4338 4376 4412 4687 4693 4712 4776 4926 4956
## [61] 5018 5057 5109 5114 5167 5301 5323 5337 5445 5479 5550 5593 5642 5801 5815
## [76] 5825 5870 5879 6067 6137 6182 6549 6575 6742 6880 6928 6942 7011 7053 7227
## [91] 7229 7290 7332 7432 7450 7495 7549 7552 7579 7596 7698 7750 7776 7792 7887
## [106] 8007 8269 8323 8547 8571 8575 8637 8658 8903 9161 9410 9564 9788 9811 9865
## [121] 9970 9978
##
## $NetPromoterScore
## [1] 20 35 85 86 94 95 105 134 147 177 189 196 215 233 252
## [16] 266 270 373 421 422 424 455 459 475 484 556 562 578 609 650
## [31] 657 659 713 714 754 780 813 820 842 846 853 854 857 858 881
## [46] 901 902 947 951 966 976 993 1025 1030 1033 1060 1103 1104 1123 1151
## [61] 1152 1156 1163 1196 1229 1245 1254 1255 1272 1293 1309 1403 1413 1461 1464
## [76] 1469 1493 1507 1513 1583 1654 1672 1686 1705 1709 1722 1728 1739 1743 1754

```

```
## [91] 1763 1772 1793 1800 1815 1828 1888 1892 1896 1903 1917 1934 1982 1996 2002
## [106] 2046 2066 2084 2090 2093 2144 2165 2169 2177 2178 2192 2201 2257 2261 2285
## [121] 2286 2300 2323 2325 2331 2370 2378 2387 2415 2427 2438 2458 2469 2521 2535
## [136] 2540 2579 2581 2588 2617 2629 2650 2653 2664 2669 2675 2676 2687 2705 2710
## [151] 2740 2763 2845 2846 2882 2927 2932 2938 2948 2956 2960 2988 2997 3001 3038
## [166] 3049 3059 3066 3068 3071 3091 3125 3129 3142 3162 3169 3171 3198 3199 3245
## [181] 3259 3269 3278 3294 3307 3345 3348 3396 3415 3436 3441 3464 3481 3512 3517
## [196] 3523 3533 3557 3566 3597 3612 3639 3658 3660 3693 3739 3758 3777 3802 3806
## [211] 3816 3823 3827 3836 3841 3862 3874 3902 3916 3931 3992 4034 4049 4055 4063
## [226] 4090 4119 4162 4173 4193 4202 4212 4236 4275 4281 4283 4292 4294 4334 4379
## [241] 4402 4417 4418 4429 4508 4510 4527 4540 4549 4557 4567 4601 4616 4637 4647
## [256] 4650 4666 4674 4680 4692 4797 4810 4839 4841 4845 4847 4871 4884 4903 4909
## [271] 4923 5020 5051 5056 5096 5111 5115 5122 5158 5200 5247 5261 5287 5298 5344
## [286] 5382 5417 5435 5438 5440 5469 5479 5489 5506 5510 5511 5535 5536 5555 5596
## [301] 5625 5629 5636 5646 5654 5691 5695 5701 5737 5764 5779 5788 5793 5820 5830
## [316] 5833 5839 5854 5875 5904 5924 6002 6035 6052 6055 6067 6076 6111 6115 6121
## [331] 6147 6255 6277 6283 6285 6314 6371 6394 6395 6405 6416 6424 6438 6444 6477
## [346] 6491 6561 6563 6626 6647 6657 6663 6680 6702 6741 6743 6748 6758 6766 6788
## [361] 6790 6811 6833 6842 6849 6866 6867 6882 6886 6889 6903 6926 6933 6961 6970
## [376] 6988 6990 6993 6996 7002 7007 7008 7028 7035 7048 7067 7069 7080 7114 7120
## [391] 7121 7123 7125 7144 7146 7163 7166 7167 7168 7177 7183 7187 7201 7227 7244
## [406] 7246 7253 7314 7328 7337 7371 7418 7427 7428 7431 7438 7440 7470 7523 7526
## [421] 7531 7541 7545 7546 7555 7578 7615 7621 7623 7631 7656 7660 7675 7681 7722
## [436] 7725 7727 7754 7756 7776 7791 7794 7811 7846 7901 7912 7917 7924 8018 8022
## [451] 8027 8050 8059 8062 8074 8118 8123 8159 8181 8187 8197 8207 8210 8273 8277
## [466] 8292 8300 8328 8339 8344 8366 8370 8405 8426 8442 8556 8575 8582 8592 8598
## [481] 8603 8629 8634 8661 8693 8725 8765 8774 8807 8815 8818 8824 8845 8874 8875
## [496] 8883 8884 8886 8923 8938 9003 9011 9034 9038 9048 9058 9059 9066 9117 9163
## [511] 9178 9185 9189 9191 9217 9264 9287 9289 9296 9309 9314 9333 9346 9350 9357
## [526] 9389 9397 9400 9421 9423 9425 9431 9499 9514 9566 9570 9587 9595 9640 9667
## [541] 9695 9700 9712 9713 9720 9727 9738 9747 9758 9765 9774 9781 9793 9801 9814
## [556] 9834 9840 9844 9847 9879 9907 9931 9968 9975 9985
```

```
longitudes_ns <- sapply(outlier_positions_non_symmetric, length)
longitudes_ns
```

```
##           Age      AvgTransactionAmount DigitalEngagementScore
##           636           642           122
##      NetPromoterScore
##           565
```

Misma conclusión que para las variables simétricas: el aumento de outliers es proporcional al aumento de datos y sigue estando dentro del recuento aceptable.

Verificación por proporciones respecto a la variable salida

Ahora haremos una comparación de las proporciones segun el grupo (train o test) para evaluar si hay alguna anomalía.

```
prop_test <- 30
prop_train <- 70
todas_outliers <- c(outlier_positions, outlier_positions_non_symmetric)
```

```

proporcion_outliers_completa <- data.frame(
  Variable = character(),
  Grupo = character(),
  N_Outliers = integer(),
  Total_Outliers = integer(),
  Proporcion_Outliers = numeric(),
  Proporcion_Esperada = numeric(),
  Diferencia = numeric(),
  stringsAsFactors = FALSE
)

for (var in names(todas_outliers)) {
  outliers <- todas_outliers[[var]]
  total_outliers_var <- length(outliers)

  if (total_outliers_var > 0) {
    outliers_test <- sum(data$group[outliers] == "test")
    outliers_train <- sum(data$group[outliers] == "train")

    proporcion_outliers_completa <- rbind(proporcion_outliers_completa,
      data.frame(
        Variable = var,
        Grupo = "test",
        N_Outliers = outliers_test,
        Total_Outliers = total_outliers_var,
        Proporcion_Outliers = round(outliers_test / total_outliers_var * 100, 1),
        Proporcion_Esperada = round(prop_test, 1),
        Diferencia = round((outliers_test / total_outliers_var * 100) - prop_test, 1)
      ),
      data.frame(
        Variable = var,
        Grupo = "train",
        N_Outliers = outliers_train,
        Total_Outliers = total_outliers_var,
        Proporcion_Outliers = round(outliers_train / total_outliers_var * 100, 1),
        Proporcion_Esperada = round(prop_train, 1),
        Diferencia = round((outliers_train / total_outliers_var * 100) - prop_train, 1)
      )
    )
  }
}

# Mostrar resultados
proporcion_outliers_completa

```

##		Variable	Grupo	N_Outliers	Total_Outliers	Proporcion_Outliers
## 1	TransactionFrequency	test	12	41	29.3	
## 2	TransactionFrequency	train	29	41	70.7	
## 3	CreditScore	test	3	15	20.0	
## 4	CreditScore	train	12	15	80.0	
## 5	Age	test	181	636	28.5	
## 6	Age	train	455	636	71.5	
## 7	AvgTransactionAmount	test	204	642	31.8	

```
## 8   AvgTransactionAmount train      438      642      68.2
## 9   DigitalEngagementScore test      35      122      28.7
## 10  DigitalEngagementScore train      87      122      71.3
## 11   NetPromoterScore test      186      565      32.9
## 12   NetPromoterScore train      379      565      67.1
##   Proportion_Esperada Diferencia
## 1           30      -0.7
## 2           70       0.7
## 3           30     -10.0
## 4           70      10.0
## 5           30     -1.5
## 6           70       1.5
## 7           30       1.8
## 8           70     -1.8
## 9           30     -1.3
## 10          70       1.3
## 11          30       2.9
## 12          70     -2.9
```

Las proporciones se mantienen

Detección de valores imposibles

Ahora haremos un summary de las variables donde hemos detectado outliers para determinar si hay valores imposibles (como una persona con edad 180).

```
variables_con_outliers <- names(todas_outliers)
summary(data[variables_con_outliers])
```

```
## TransactionFrequency CreditScore      Age      AvgTransactionAmount
## Min.   :13.00      Min.   :350.0  Min.   :18.00  Min.   : 19.60
## 1st Qu.:26.00      1st Qu.:584.0  1st Qu.:32.00  1st Qu.: 70.14
## Median :30.00      Median :652.0  Median :37.00  Median : 99.01
## Mean   :30.08      Mean   :650.9  Mean   :39.08  Mean   :112.09
## 3rd Qu.:34.00      3rd Qu.:717.0  3rd Qu.:44.00  3rd Qu.:138.41
## Max.   :58.00      Max.   :850.0  Max.   :92.00  Max.   :611.35
## DigitalEngagementScore NetPromoterScore
## Min.   : 5.00      Min.   : 0.000
## 1st Qu.: 50.00      1st Qu.: 4.000
## Median : 60.00      Median : 8.000
## Mean   : 59.66      Mean   : 6.444
## 3rd Qu.: 70.00      3rd Qu.: 9.000
## Max.   :100.00      Max.   :10.000
```

Sigue sin haber valores imposibles

Conclusión del análisis de outliers post imputación

- Se respetan las proporciones previas.
- Sigue sin haber valores imposibles.
- Son outliers que se pueden aceptar como normales.

Estas conclusiones sumadas al análisis del ajuste de la distribución imputada nos permiten **validar** la imputación multivariante mediante el método MICE.