

Visualización de Datos con Python: Técnicas y Ejemplos de Gráficos para Análisis de Datos

Elmer Achalma

Economía, Universidad Nacional de San Cristóbal de Huamanga

Este documento presenta una colección de ejemplos prácticos de visualización de datos utilizando Python y la biblioteca Matplotlib. Se incluyen gráficos de líneas, barras, histogramas, circulares, de caja y combinados, cada uno acompañado de código comentado y optimizado. El objetivo es proporcionar una guía educativa para estudiantes y profesionales interesados en representar datos de manera clara y efectiva, con énfasis en buenas prácticas de diseño y presentación.

Palabras Claves: Visualización de datos, Python, Matplotlib, Gráficos estadísticos, Análisis de datos

Tabla de contenidos

Introduction	1
Grafico de lineas	1
Gráfico de barras	2
horizontal	2
vertical	2
Histograma	2
Grafico circular	3
Grafico de Donut	3
Grafico de cajas	3
Grafico de barras combinadas	4
Graficos combinados	5

Visualización de Datos con Python

Grafico de lineas

```
import numpy as np
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para mejor visualización
plt.figure(figsize=(10, 8))

# Datos de consumo de carne bovina (kg por habitante)
consumo_bovino = [22.1, 22.1, 23.1, 23.9, 24.6, 21.7, 20.5, 21.2, 22.8, 23.5, 24.1, 23.8, 24.5, 25.2, 24.9, 25.6, 26.3, 27.1, 27.8, 28.5, 29.2, 30.1, 30.8, 31.5, 32.2, 33.1, 33.8, 34.5, 35.2, 36.1, 36.8, 37.5, 38.2, 39.1, 39.8, 40.5, 41.2, 42.1, 42.8, 43.5, 44.2, 45.1, 45.8, 46.5, 47.2, 48.1, 48.8, 49.5, 50.2, 51.1, 51.8, 52.5, 53.2, 54.1, 54.8, 55.5, 56.2, 57.1, 57.8, 58.5, 59.2, 60.1, 60.8, 61.5, 62.2, 63.1, 63.8, 64.5, 65.2, 66.1, 66.8, 67.5, 68.2, 69.1, 69.8, 70.5, 71.2, 72.1, 72.8, 73.5, 74.2, 75.1, 75.8, 76.5, 77.2, 78.1, 78.8, 79.5, 80.2, 81.1, 81.8, 82.5, 83.2, 84.1, 84.8, 85.5, 86.2, 87.1, 87.8, 88.5, 89.2, 90.1, 90.8, 91.5, 92.2, 93.1, 93.8, 94.5, 95.2, 96.1, 96.8, 97.5, 98.2, 99.1, 99.8, 100.5, 101.2, 102.1, 102.8, 103.5, 104.2, 105.1, 105.8, 106.5, 107.2, 108.1, 108.8, 109.5, 110.2, 111.1, 111.8, 112.5, 113.2, 114.1, 114.8, 115.5, 116.2, 117.1, 117.8, 118.5, 119.2, 120.1, 120.8, 121.5, 122.2, 123.1, 123.8, 124.5, 125.2, 126.1, 126.8, 127.5, 128.2, 129.1, 129.8, 130.5, 131.2, 132.1, 132.8, 133.5, 134.2, 135.1, 135.8, 136.5, 137.2, 138.1, 138.8, 139.5, 140.2, 141.1, 141.8, 142.5, 143.2, 144.1, 144.8, 145.5, 146.2, 147.1, 147.8, 148.5, 149.2, 150.1, 150.8, 151.5, 152.2, 153.1, 153.8, 154.5, 155.2, 156.1, 156.8, 157.5, 158.2, 159.1, 159.8, 160.5, 161.2, 162.1, 162.8, 163.5, 164.2, 165.1, 165.8, 166.5, 167.2, 168.1, 168.8, 169.5, 170.2, 171.1, 171.8, 172.5, 173.2, 174.1, 174.8, 175.5, 176.2, 177.1, 177.8, 178.5, 179.2, 180.1, 180.8, 181.5, 182.2, 183.1, 183.8, 184.5, 185.2, 186.1, 186.8, 187.5, 188.2, 189.1, 189.8, 190.5, 191.2, 192.1, 192.8, 193.5, 194.2, 195.1, 195.8, 196.5, 197.2, 198.1, 198.8, 199.5, 200.2, 201.1, 201.8, 202.5, 203.2, 204.1, 204.8, 205.5, 206.2, 207.1, 207.8, 208.5, 209.2, 210.1, 210.8, 211.5, 212.2, 213.1, 213.8, 214.5, 215.2, 216.1, 216.8, 217.5, 218.2, 219.1, 219.8, 220.5, 221.2, 222.1, 222.8, 223.5, 224.2, 225.1, 225.8, 226.5, 227.2, 228.1, 228.8, 229.5, 230.2, 231.1, 231.8, 232.5, 233.2, 234.1, 234.8, 235.5, 236.2, 237.1, 237.8, 238.5, 239.2, 240.1, 240.8, 241.5, 242.2, 243.1, 243.8, 244.5, 245.2, 246.1, 246.8, 247.5, 248.2, 249.1, 249.8, 250.5, 251.2, 252.1, 252.8, 253.5, 254.2, 255.1, 255.8, 256.5, 257.2, 258.1, 258.8, 259.5, 260.2, 261.1, 261.8, 262.5, 263.2, 264.1, 264.8, 265.5, 266.2, 267.1, 267.8, 268.5, 269.2, 270.1, 270.8, 271.5, 272.2, 273.1, 273.8, 274.5, 275.2, 276.1, 276.8, 277.5, 278.2, 279.1, 279.8, 280.5, 281.2, 282.1, 282.8, 283.5, 284.2, 285.1, 285.8, 286.5, 287.2, 288.1, 288.8, 289.5, 290.2, 291.1, 291.8, 292.5, 293.2, 294.1, 294.8, 295.5, 296.2, 297.1, 297.8, 298.5, 299.2, 300.1, 300.8, 301.5, 302.2, 303.1, 303.8, 304.5, 305.2, 306.1, 306.8, 307.5, 308.2, 309.1, 309.8, 310.5, 311.2, 312.1, 312.8, 313.5, 314.2, 315.1, 315.8, 316.5, 317.2, 318.1, 318.8, 319.5, 320.2, 321.1, 321.8, 322.5, 323.2, 324.1, 324.8, 325.5, 326.2, 327.1, 327.8, 328.5, 329.2, 330.1, 330.8, 331.5, 332.2, 333.1, 333.8, 334.5, 335.2, 336.1, 336.8, 337.5, 338.2, 339.1, 339.8, 340.5, 341.2, 342.1, 342.8, 343.5, 344.2, 345.1, 345.8, 346.5, 347.2, 348.1, 348.8, 349.5, 350.2, 351.1, 351.8, 352.5, 353.2, 354.1, 354.8, 355.5, 356.2, 357.1, 357.8, 358.5, 359.2, 360.1, 360.8, 361.5, 362.2, 363.1, 363.8, 364.5, 365.2, 366.1, 366.8, 367.5, 368.2, 369.1, 369.8, 370.5, 371.2, 372.1, 372.8, 373.5, 374.2, 375.1, 375.8, 376.5, 377.2, 378.1, 378.8, 379.5, 380.2, 381.1, 381.8, 382.5, 383.2, 384.1, 384.8, 385.5, 386.2, 387.1, 387.8, 388.5, 389.2, 390.1, 390.8, 391.5, 392.2, 393.1, 393.8, 394.5, 395.2, 396.1, 396.8, 397.5, 398.2, 399.1, 399.8, 400.5, 401.2, 402.1, 402.8, 403.5, 404.2, 405.1, 405.8, 406.5, 407.2, 408.1, 408.8, 409.5, 410.2, 411.1, 411.8, 412.5, 413.2, 414.1, 414.8, 415.5, 416.2, 417.1, 417.8, 418.5, 419.2, 420.1, 420.8, 421.5, 422.2, 423.1, 423.8, 424.5, 425.2, 426.1, 426.8, 427.5, 428.2, 429.1, 429.8, 430.5, 431.2, 432.1, 432.8, 433.5, 434.2, 435.1, 435.8, 436.5, 437.2, 438.1, 438.8, 439.5, 440.2, 441.1, 441.8, 442.5, 443.2, 444.1, 444.8, 445.5, 446.2, 447.1, 447.8, 448.5, 449.2, 450.1, 450.8, 451.5, 452.2, 453.1, 453.8, 454.5, 455.2, 456.1, 456.8, 457.5, 458.2, 459.1, 459.8, 460.5, 461.2, 462.1, 462.8, 463.5, 464.2, 465.1, 465.8, 466.5, 467.2, 468.1, 468.8, 469.5, 470.2, 471.1, 471.8, 472.5, 473.2, 474.1, 474.8, 475.5, 476.2, 477.1, 477.8, 478.5, 479.2, 480.1, 480.8, 481.5, 482.2, 483.1, 483.8, 484.5, 485.2, 486.1, 486.8, 487.5, 488.2, 489.1, 489.8, 490.5, 491.2, 492.1, 492.8, 493.5, 494.2, 495.1, 495.8, 496.5, 497.2, 498.1, 498.8, 499.5, 500.2, 501.1, 501.8, 502.5, 503.2, 504.1, 504.8, 505.5, 506.2, 507.1, 507.8, 508.5, 509.2, 510.1, 510.8, 511.5, 512.2, 513.1, 513.8, 514.5, 515.2, 516.1, 516.8, 517.5, 518.2, 519.1, 519.8, 520.5, 521.2, 522.1, 522.8, 523.5, 524.2, 525.1, 525.8, 526.5, 527.2, 528.1, 528.8, 529.5, 530.2, 531.1, 531.8, 532.5, 533.2, 534.1, 534.8, 535.5, 536.2, 537.1, 537.8, 538.5, 539.2, 540.1, 540.8, 541.5, 542.2, 543.1, 543.8, 544.5, 545.2, 546.1, 546.8, 547.5, 548.2, 549.1, 549.8, 550.5, 551.2, 552.1, 552.8, 553.5, 554.2, 555.1, 555.8, 556.5, 557.2, 558.1, 558.8, 559.5, 560.2, 561.1, 561.8, 562.5, 563.2, 564.1, 564.8, 565.5, 566.2, 567.1, 567.8, 568.5, 569.2, 570.1, 570.8, 571.5, 572.2, 573.1, 573.8, 574.5, 575.2, 576.1, 576.8, 577.5, 578.2, 579.1, 579.8, 580.5, 581.2, 582.1, 582.8, 583.5, 584.2, 585.1, 585.8, 586.5, 587.2, 588.1, 588.8, 589.5, 590.2, 591.1, 591.8, 592.5, 593.2, 594.1, 594.8, 595.5, 596.2, 597.1, 597.8, 598.5, 599.2, 600.1, 600.8, 601.5, 602.2, 603.1, 603.8, 604.5, 605.2, 606.1, 606.8, 607.5, 608.2, 609.1, 609.8, 610.5, 611.2, 612.1, 612.8, 613.5, 614.2, 615.1, 615.8, 616.5, 617.2, 618.1, 618.8, 619.5, 620.2, 621.1, 621.8, 622.5, 623.2, 624.1, 624.8, 625.5, 626.2, 627.1, 627.8, 628.5, 629.2, 630.1, 630.8, 631.5, 632.2, 633.1, 633.8, 634.5, 635.2, 636.1, 636.8, 637.5, 638.2, 639.1, 639.8, 640.5, 641.2, 642.1, 642.8, 643.5, 644.2, 645.1, 645.8, 646.5, 647.2, 648.1, 648.8, 649.5, 650.2, 651.1, 651.8, 652.5, 653.2, 654.1, 654.8, 655.5, 656.2, 657.1, 657.8, 658.5, 659.2, 660.1, 660.8, 661.5, 662.2, 663.1, 663.8, 664.5, 665.2, 666.1, 666.8, 667.5, 668.2, 669.1, 669.8, 670.5, 671.2, 672.1, 672.8, 673.5, 674.2, 675.1, 675.8, 676.5, 677.2, 678.1, 678.8, 679.5, 680.2, 681.1, 681.8, 682.5, 683.2, 684.1, 684.8, 685.5, 686.2, 687.1, 687.8, 688.5, 689.2, 690.1, 690.8, 691.5, 692.2, 693.1, 693.8, 694.5, 695.2, 696.1, 696.8, 697.5, 698.2, 699.1, 699.8, 700.5, 701.2, 702.1, 702.8, 703.5, 704.2, 705.1, 705.8, 706.5, 707.2, 708.1, 708.8, 709.5, 710.2, 711.1, 711.8, 712.5, 713.2, 714.1, 714.8, 715.5, 716.2, 717.1, 717.8, 718.5, 719.2, 720.1, 720.8, 721.5, 722.2, 723.1, 723.8, 724.5, 725.2, 726.1, 726.8, 727.5, 728.2, 729.1, 729.8, 730.5, 731.2, 732.1, 732.8, 733.5, 734.2, 735.1, 735.8, 736.5, 737.2, 738.1, 738.8, 739.5, 740.2, 741.1, 741.8, 742.5, 743.2, 744.1, 744.8, 745.5, 746.2, 747.1, 747.8, 748.5, 749.2, 750.1, 750.8, 751.5, 752.2, 753.1, 753.8, 754.5, 755.2, 756.1, 756.8, 757.5, 758.2, 759.1, 759.8, 760.5, 761.2, 762.1, 762.8, 763.5, 764.2, 765.1, 765.8, 766.5, 767.2, 768.1, 768.8, 769.5, 770.2, 771.1, 771.8, 772.5, 773.2, 774.1, 774.8, 775.5, 776.2, 777.1, 777.8, 778.5, 779.2, 780.1, 780.8, 781.5, 782.2, 783.1, 783.8, 784.5, 785.2, 786.1, 786.8, 787.5, 788.2, 789.1, 789.8, 790.5, 791.2, 792.1, 792.8, 793.5, 794.2, 795.1, 795.8, 796.5, 797.2, 798.1, 798.8, 799.5, 800.2, 801.1, 801.8, 802.5, 803.2, 804.1, 804.8, 805.5, 806.2, 807.1, 807.8, 808.5, 809.2, 810.1, 810.8, 811.5, 812.2, 813.1, 813.8, 814.5, 815.2, 816.1, 816.8, 817.5, 818.2, 819.1, 819.8, 820.5, 821.2, 822.1, 822.8, 823.5, 824.2, 825.1, 825.8, 826.5, 827.2, 828.1, 828.8, 829.5, 830.2, 831.1, 831.8, 832.5, 833.2, 834.1, 834.8, 835.5, 836.2, 837.1, 837.8, 838.5, 839.2, 840.1, 840.8, 841.5, 842.2, 843.1, 843.8, 844.5, 845.2, 846.1, 846.8, 847.5, 848.2, 849.1, 849.8, 850.5, 851.2, 852.1, 852.8, 853.5, 854.2, 855.1, 855.8, 856.5, 857.2, 858.1, 858.8, 859.5, 860.2, 861.1, 861.8, 862.5, 863.2, 864.1, 864.8, 865.5, 866.2, 867.1, 867.8, 868.5, 869.2, 870.1, 870.8, 871.5, 872.2, 873.1, 873.8, 874.5, 875.2, 876.1, 876.8, 877.5, 878.2, 879.1, 879.8, 880.5, 881.2, 882.1, 882.8, 883.5, 884.2, 885.1, 885.8, 886.5, 887.2, 888.1, 888.8, 889.5, 890.2, 891.1, 891.8, 892.5, 893.2, 894.1, 894.8, 895.5, 896.2, 897.1, 897.8, 898.5, 899.2, 900.1, 900.8, 901.5, 902.2, 903.1, 903.8, 904.5, 905.2, 906.1, 906.8, 907.5, 908.2, 909.1, 909.8, 910.5, 911.2, 912.1, 912.8, 913.5, 914.2, 915.1, 915.8, 916.5, 917.2, 918.1, 918.8, 919.5, 920.2, 921.1, 921.8, 922.5, 923.2, 924.1, 924.8, 925.5, 926.2, 927.1, 927.8, 928.5, 929.2, 930.1, 930.8, 931.5, 932.2, 933.1, 933.8, 934.5, 935.2, 936.1, 936.8, 937.5, 938.2, 939.1, 939.8, 940.5, 941.2, 942.1, 942.8, 943.5, 944.2, 945.1, 945.8, 946.5, 947.2, 948.1, 948.8, 949.5, 950.2, 951.1, 951.8, 952.5, 953.2, 954.1, 954.8, 955.5, 956.2, 957.1, 957.8, 958.5, 959.2, 960.1, 960.8, 961.5, 962.2, 963.1, 963.8, 964.5, 965.2, 966.1, 966.8, 967.5, 968.2, 969.1, 969.8, 970.5, 971.2, 972.1, 972.8, 973.5, 974.2, 975.1, 975.8, 976.5, 977.2, 978.1, 978.8, 979.5, 980.2, 981.1, 981.8, 982.5, 983.2, 984.1, 984.8, 985.5, 986.2, 987.1, 987.8, 988.5, 989.2, 990.1, 990.8, 991.5, 992.2, 993.1, 993.8, 994.5, 995.2, 996.1, 996.8, 997.5, 998.2, 999.1, 999.8, 1000.5, 1001.2, 1002.1, 1002.8, 1003.5, 1004.2, 1005.1, 1005.8, 1006.5, 1007.2, 1008.1, 1008.8, 1009.5, 1010.2, 1011.1, 1011.8, 1012.5, 1013.2, 1014.1, 1014.8, 1015.5, 1016.2, 1017.1, 1017.8, 1018.5, 1019.2, 1020.1, 1020.8, 1021.5, 1022.2, 1023.1, 1023.8, 1024.5, 1025.2, 1026.1, 1026.8, 1027.5, 1028.2, 1029.1, 1029.8, 1030.5, 1031.2, 1032.1, 1032.8, 1033.5, 1034.2, 1035.1, 1035.8, 1036.5, 1037.2, 1038.1, 1038.8, 1039.5, 1040.2, 1041.1, 1041.8, 1042.5, 1043.2, 1044.1, 1044.8, 1045.5, 1046.2, 1047.1, 1047.8, 1048.5, 1049.2, 1050.1, 1050.8, 1051.5, 1052.2, 1053.1, 1053.8, 1054.5, 1055.2, 1056.1, 1056.8, 1057.5, 1058.2, 1059.1, 1059.8, 1060.5, 1061.2, 1062.1, 1062.8, 1063.5, 1064.2, 1065.1, 1065.8, 1066.5, 1067.2, 1068.1, 1068.8, 1069.5, 1070.2, 1071.1, 1071.8, 1072.5, 1073.2, 1074.1, 1074.8, 1075.5, 1076.2, 1077.1, 1077.8, 1078.5, 1079.2, 1080.1, 1080.8, 1081.5, 1082.2, 1083.1, 1083.8, 1084.5, 1085.2, 1086.1, 1086.8, 1087.5, 1088.2, 1089.1, 1089.8, 1090.5, 1091.2, 1092.1, 1092.8, 1093.5, 1094.2, 1095.1, 1095.8, 1096.5, 1097.2, 1098.1, 1098.8, 1099.5, 1100.2, 1101.1, 1101.8, 1102.5, 1103.2, 1104.1, 1104.8, 1105.5, 1106.2, 1107.1, 1107.8, 1108.5, 1109.2, 1110.1, 1110.8, 1111.5, 1112.2, 1113.1, 1113.8, 1114.5, 1115.2, 1116.1, 1116.8, 1117.5, 1118.2, 1119.1, 1119.8, 1120.5, 1121.2, 1122.1, 1122.8, 1123.5, 1124.2, 1125.1, 1125.8, 1126.5, 1127.2, 1128.1, 1128.8, 1129.5, 1130.2, 1131.1, 1131.8, 1132.5, 1133.2, 1134.1, 1134.8, 1135.5, 1136.2, 1137.1, 1137.8, 1138.5, 1139.2, 1140.1, 1140.8, 1141.5, 1142.2, 1143.1, 1143.8, 1144.5, 1145.2, 1146.1, 1146.8, 1147.5, 1148.2, 1149.1, 1149.8, 1150.5, 1151.2, 1152.1, 1152.8, 1153.5, 1154.2, 1155.1, 1155.8, 1156.5, 1157.2, 1158.1, 1158.8, 1159.5, 1160.2, 1161.1, 1161.8, 1162.5, 1163.2, 1164.1, 1164.8, 1165.5, 1166.2, 1167.1, 1167.8, 1168.5, 1169.2, 1170.1, 1170.8, 1171.5, 1172.2, 1173.1, 1173.8, 1174.5, 1175.2, 1176.1, 1176.8, 1177.5, 1178.2, 1179.1, 1179.8, 1180.5, 1181.2, 1182.1, 1182.8, 1183.5, 1184.2, 1185.1, 1185.8, 1186.5, 1187.2, 1188.1, 1188.8, 1189.5, 1190.2, 1191.1, 1191.8, 1192.5, 1193.2, 1194.1, 1194.8, 1195.5, 1196.2, 1197.1, 1197.8, 1198.5, 1199.2, 1200.1, 1200.8, 1201.5, 1202.2, 1203.1, 1203.8, 1204.5, 1205.2, 1206.1, 1206.8, 1207.5, 1208.2, 1209.1, 1209.8, 1210.5, 1211.2, 1212.1, 1212.8, 1213.5, 1214.2, 1215.1, 1215.8, 1216.5, 1217.2, 1218.1, 1218.8, 1219.5, 1220.2, 1221.1, 1221.8, 1222.5, 1223.2, 1224.1, 1224.8, 1225.5, 1226.2, 1227.1, 1227.8, 1228.5,
```

```
# Añadir una cuadrícula para mejorar la legibilidad
plt.grid(True, linestyle='--', alpha=0.7)

# Ajustar el diseño para evitar recortes de etiquetas
plt.tight_layout()

# Guardar la figura en un archivo (opcional, se puede omitir)
# plt.savefig('consumo_carne_chile.png')

# Mostrar el gráfico
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2 * np.pi, 200)
y = np.sin(x)

fig, ax = plt.subplots()
ax.plot(x, y)
plt.show()
```

hol

Figura 1

```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

Gráfico de barras

horizontal

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una mejor visualización
plt.figure(figsize=(10, 8))

# Tipos de legumbres y sus respectivos consumos en kg por habitante en 2001
tipos_legumbres = ["Poroto", "Lenteja", "Garbanzo", "Arveja"]
consumo_legumbres = [2.1, 1.0, 0.3, 0.5]
```

```
# Crear un gráfico de barras con color personalizado y bordes
plt.bar(tipos_legumbres, consumo_legumbres, color='green', edgecolor='black')

# Etiquetas de los ejes y título con formato adecuado
plt.xlabel('Tipos de Legumbres')
plt.ylabel('Consumo (kg por habitante)')
plt.title('Consumo de Legumbres en Chile (2001)')

# Añadir una cuadrícula en el eje y para facilitar la lectura
plt.grid(True, axis='y', linestyle='--', alpha=0.7)

# Ajustar el diseño para evitar recortes de etiquetas
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descomentar)
# plt.savefig('consumo_legumbres_2001.png')

# Mostrar el gráfico
plt.show()
```

vertical

```
plt.figure(figsize=[10,8])
#Consumo de legumbres en el 2001
legumbres=["Poroto","Lenteja","Garbanzo","Arveja"]
consumo=[2.1, 1.0, 0.3, 0.5]

plt.barh(legumbres,consumo)

plt.ylabel("Tipos de legumbre")
plt.xlabel("Consumo (kg/hab)")
plt.title("Consumo de Legumbres en el 2001")
plt.show()
```

Histograma

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una mejor visualización
plt.figure(figsize=(10, 8))

# Datos de niveles de glucosa (mg/dl)
niveles_glucosa = [52, 54, 55, 57, 56, 57, 54, 59, 60, 76, 79, 81, 85, 88, 84, 89, 92, 85, 90, 91, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150]

# Bordes de los intervalos para el histograma (bins)
intervalos = [50, 60, 70, 80, 90, 100, 110, 120, 130]

# Crear el histograma con color personalizado y bordes
plt.hist(niveles_glucosa, bins=intervalos, color='skyblue', edgecolor='black')
```

```
# Etiquetas de los ejes y título con formato adecuado
plt.xlabel('Nivel de Glucosa (mg/dl)')
plt.ylabel('Número de Pacientes')
plt.title('Distribución de Niveles de Glucosa en Pacientes')
```

Grafico de Donut

```
# Añadir una cuadrícula en el eje y para facilitar la lectura
plt.grid(True, axis='y', linestyle='--', alpha=0.5)

# Ajustar el diseño para evitar recortes de etiquetas
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descomentar)
# plt.savefig('distribucion_glucosa.png')

# Mostrar el gráfico
plt.show()
```

Grafico circular

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una mejor visualización
plt.figure(figsize=(10, 8))

# Datos de marcas de autos y sus ventas (en alguna unidad)
marcas_autos = ["Kia", "Toyota", "Nissan", "Suzuki", "Honda"]
ventas = [10.5, 15.3, 14.2, 16.1, 9.8]
# Resaltar la primera marca (Kia) ligeramente
resaltar = [0.1, 0, 0, 0, 0]

# Aplicar un estilo visual predefinido (ggplot)
plt.style.use("ggplot")

# Crear el gráfico de pastel
plt.pie(x=ventas, explode=resaltar, labels=marcas_autos, autopct='%1.1f%%')

# Asegurar que el gráfico sea circular
plt.axis("equal")

# Añadir un título descriptivo
plt.title("Distribución de Ventas de Autos en EE.UU.")

# Añadir una leyenda en la esquina superior izquierda
plt.legend(marcas_autos, loc="upper left")

# Añadir un círculo central para crear el efecto de donut
circulo_central = plt.Circle(xy=(0, 0), radius=0.75, facecolor='white')
plt.gca().add_artist(circulo_central)

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descomentar)
# plt.savefig('ventas_autos_eeuu_donut.png')

# Mostrar el gráfico
plt.show()
```

Grafico de cajas

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una mejor visualización
plt.figure(figsize=(10, 8))

# Datos de las edades de los alumnos
```

Grafico de barras combinadas

```

edades_alumnos = [12, 13, 12, 17, 16, 15, 14, 15, 15, 16, 14, 12, 15, 16, 14, 12, 16, 17]

# Crear el diagrama de caja con un estilo persona
plt.boxplot(edades_alumnos, vert=True, patch_artist=True,
             boxprops=dict(facecolor='lightblue', # Configurar el tamaño de la figura para una mejor visualización
                           medianprops=dict(color='red'), whiskerprops=dict(color='black'), flierprops=dict(color='black', markersize=8))

# Etiquetas de los ejes y título con formato adecuado
plt.ylabel('Edad (años)')
plt.title('Distribución de Edades de Alumnos de Secundaria')

# Añadir una cuadrícula en el eje y para facilitar la lectura
plt.grid(True, axis='y', linestyle='--', alpha=0.5)

# Configurar las etiquetas del eje x (opcional, y para centrar las barras)
plt.xticks([1], ['Alumnos'])

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descripción)
# plt.savefig('edades_alumnos_secundaria.png')

# Mostrar el gráfico
plt.show()

import matplotlib.pyplot as plt

# Datos de ventas de autos por día (en unidades)
ventas_toyota = [10, 15, 19, 14, 9]
ventas_audi = [15, 25, 27, 24, 28]
dias = [0, 1, 2, 3, 4] # Posiciones para los días
ancho_barras = 0.5 # Grosor de las barras

# Crear el gráfico de barras apiladas
barras_toyota = plt.bar(dias, ventas_toyota, ancho_barras)
barras_audi = plt.bar(dias, ventas_audi, ancho_barras)

# Configurar las etiquetas del eje x con los días
plt.xticks(dias, ['Día 1', 'Día 2', 'Día 3', 'Día 4', 'Día 5'])

# Etiquetas de los ejes y título con formato adecuado
plt.xlabel('Días')
plt.ylabel('Unidades Vendidas')
plt.title('Ventas de Autos por Día (Toyota vs. Audi)')

```

```
# Configurar las marcas del eje y para mayor clar # plt.savefig('vacunacion_real_vs_proyectada.png')
plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45,
            # Mostrar el gráfico
            # Añadir una cuadrícula en el eje y para facilita plt.show()
            plt.grid(True, axis='y', linestyle='--', alpha=0.7)

# Añadir una leyenda para identificar las marcas
plt.legend()

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descomentar para usar)
# plt.savefig('ventas_autos_toyota_audi.png')

# Mostrar el gráfico
plt.show()
```

Graficos combinados

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una mejor visualización
plt.figure(figsize=(10, 8))

# Datos de vacunación proyectada (en número de pacientes)
vacunacion_proyectada = [250, 120, 270, 560, 450, 280, 550]
# Datos de vacunación real (en número de pacientes)
vacunacion_real = [150, 300, 120, 550, 500, 240, 600]
meses = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio"]

# Graficar la vacunación proyectada como una línea con marcadores
plt.plot(meses, vacunacion_proyectada, marker='d', linestyle='--', color='red', label='Vacunación Proyectada')

# Graficar la vacunación real como barras
plt.bar(meses, vacunacion_real, color='skyblue', edgecolor='black', alpha=0.7, label='Vacunación Real')

# Etiquetas de los ejes y título con formato adecuado
plt.xlabel('Meses')
plt.ylabel('Número de Pacientes Vacunados')
plt.title('Vacunación Real vs. Proyectada (Enero - Julio)')

# Añadir una leyenda
plt.legend()

# Añadir una cuadrícula en el eje y para facilitar la lectura
plt.grid(True, axis='y', linestyle='--', alpha=0.7)

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descomentar para usar)
```