Data Frames in R

Arie Twigt

4/7/2022

# Importing the Data Frame

We will import the **German Data** data, that we will use for this project:

# import data  
german\_data\_clean <- read.csv("data/german\_data\_clean.csv")

This is imported with the standard base functions

# Analyzing the Data Frame

## Check the columns

With the names() functions, we will get all the variables of the data frame.

names(german\_data\_clean)

## [1] "checking\_account" "duration\_months"   
## [3] "credit\_history" "purpose"   
## [5] "credit\_amount" "savings"   
## [7] "present\_employment\_since" "installment\_rate"   
## [9] "personal\_status\_sex" "other\_deptors"   
## [11] "present\_residence" "property"   
## [13] "age\_years" "other\_installment\_plans"   
## [15] "housing" "existing\_credits"   
## [17] "job" "people\_liable\_maintenance"  
## [19] "telephone" "foreign\_worker"   
## [21] "response"

With the $ operator, we can select a column in a Data Frame.

german\_data\_clean$credit\_amount

Apply the mean() function for a column in the Data Frame:

mean(german\_data\_clean$credit\_amount)

## [1] 3271.258

Wrap a function over another function:

round(mean(german\_data\_clean$credit\_amount), 2)

## [1] 3271.26

Provide a statistical summary of a column with the summary() function:

summary(german\_data\_clean$credit\_amount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 250 1366 2320 3271 3972 18424

Provide a summary over all columns in the Data Frame

summary(german\_data\_clean)

## checking\_account duration\_months credit\_history purpose   
## Length:1000 Min. : 4.0 Length:1000 Length:1000   
## Class :character 1st Qu.:12.0 Class :character Class :character   
## Mode :character Median :18.0 Mode :character Mode :character   
## Mean :20.9   
## 3rd Qu.:24.0   
## Max. :72.0   
## credit\_amount savings present\_employment\_since installment\_rate  
## Min. : 250 Length:1000 Length:1000 Min. :1.000   
## 1st Qu.: 1366 Class :character Class :character 1st Qu.:2.000   
## Median : 2320 Mode :character Mode :character Median :3.000   
## Mean : 3271 Mean :2.973   
## 3rd Qu.: 3972 3rd Qu.:4.000   
## Max. :18424 Max. :4.000   
## personal\_status\_sex other\_deptors present\_residence property   
## Length:1000 Length:1000 Min. :1.000 Length:1000   
## Class :character Class :character 1st Qu.:2.000 Class :character   
## Mode :character Mode :character Median :3.000 Mode :character   
## Mean :2.845   
## 3rd Qu.:4.000   
## Max. :4.000   
## age\_years other\_installment\_plans housing existing\_credits  
## Min. :19.00 Length:1000 Length:1000 Min. :1.000   
## 1st Qu.:27.00 Class :character Class :character 1st Qu.:1.000   
## Median :33.00 Mode :character Mode :character Median :1.000   
## Mean :35.55 Mean :1.407   
## 3rd Qu.:42.00 3rd Qu.:2.000   
## Max. :75.00 Max. :4.000   
## job people\_liable\_maintenance telephone   
## Length:1000 Min. :1.000 Length:1000   
## Class :character 1st Qu.:1.000 Class :character   
## Mode :character Median :1.000 Mode :character   
## Mean :1.155   
## 3rd Qu.:1.000   
## Max. :2.000   
## foreign\_worker response   
## Length:1000 Min. :0.0   
## Class :character 1st Qu.:0.0   
## Mode :character Median :0.0   
## Mean :0.3   
## 3rd Qu.:1.0   
## Max. :1.0

Show the different values and occurances of a character column

summary(factor(german\_data\_clean$purpose))

## business car (new) car (used) domestic appliances   
## 97 234 103 12   
## education furniture/equipment others radio/television   
## 50 181 12 280   
## repairs retraining   
## 22 9

table(german\_data\_clean$purpose)

##   
## business car (new) car (used) domestic appliances   
## 97 234 103 12   
## education furniture/equipment others radio/television   
## 50 181 12 280   
## repairs retraining   
## 22 9

Show the unique values of a character column:

unique(german\_data\_clean$purpose)

## [1] "radio/television" "education" "furniture/equipment"  
## [4] "car (new)" "car (used)" "business"   
## [7] "domestic appliances" "repairs" "others"   
## [10] "retraining"

unique(german\_data\_clean$credit\_amount)

## [1] 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 1295 4308  
## [13] 1567 1199 1403 1282 2424 8072 12579 3430 2134 2647 2241 1804  
## [25] 2069 1374 426 409 2415 6836 1913 4020 5866 1264 1474 4746  
## [37] 6110 2100 1225 458 2333 1158 6204 6187 6143 1393 2299 1352  
## [49] 7228 2073 5965 1262 3378 2225 783 6468 9566 1961 6229 1391  
## [61] 1537 1953 14421 3181 5190 2171 1007 1819 2394 8133 730 1164  
## [73] 5954 1977 1526 3965 4771 9436 3832 5943 1213 1568 1755 2315  
## [85] 1412 12612 2249 1108 618 1409 797 3617 1318 15945 2012 2622  
## [97] 2337 7057 1469 2323 932 1919 2445 11938 6458 6078 7721 1410  
## [109] 1449 392 6260 7855 1680 3578 7174 2132 4281 2366 1835 3868  
## [121] 1768 781 1924 2121 701 639 1860 3499 8487 6887 2708 1984  
## [133] 10144 1240 8613 766 2728 1881 709 4795 3416 2462 2288 3566  
## [145] 860 682 5371 1582 1346 5848 7758 6967 1288 339 3512 1898  
## [157] 2872 1055 7308 909 2978 1131 1577 3972 1935 950 763 2064  
## [169] 1414 3414 7485 2577 338 1963 571 9572 4455 1647 3777 884  
## [181] 1360 5129 1175 674 3244 4591 3844 3915 2108 3031 1501 1382  
## [193] 951 2760 4297 936 1168 5117 902 1495 10623 1424 6568 1413  
## [205] 3074 3835 5293 1908 3342 3104 3913 3021 1364 625 1200 707  
## [217] 4657 2613 10961 7865 1478 3149 4210 2507 2141 866 1544 1823  
## [229] 14555 2767 1291 2522 915 1595 4605 1185 3447 1258 717 1204  
## [241] 1925 433 666 2251 2150 4151 2030 7418 2684 2149 3812 1154  
## [253] 1657 1603 5302 2748 1231 802 6304 1533 8978 999 2662 1402  
## [265] 12169 3060 11998 2697 2404 4611 1901 3368 1574 1445 1520 3878  
## [277] 10722 4788 7582 1092 1024 1076 9398 6419 4796 7629 9960 4675  
## [289] 1287 2515 2745 672 3804 1344 1038 10127 1543 4811 727 1237  
## [301] 276 5381 5511 3749 685 1494 2746 708 4351 3643 4249 1938  
## [313] 2910 2659 1028 3398 5801 1525 4473 1068 6615 1864 7408 11590  
## [325] 4110 3384 2101 1275 4169 1521 5743 3599 3213 4439 3949 1459  
## [337] 882 3758 1743 1136 1236 959 3229 6199 1246 2331 4463 776  
## [349] 2406 1239 3399 2247 1766 2473 1542 3850 3650 3446 3001 3079  
## [361] 6070 2146 13756 14782 7685 2320 846 14318 362 2212 12976 1283  
## [373] 1330 4272 2238 1126 7374 2326 1820 983 3249 1957 11760 2578  
## [385] 2348 1223 1516 1473 1887 8648 2899 2039 2197 1053 3235 939  
## [397] 1967 7253 2292 1597 1381 5842 2579 8471 2782 1042 3186 2028  
## [409] 958 1591 2762 2779 2743 1149 1313 1190 3448 11328 1872 2058  
## [421] 2136 1484 660 3394 609 1884 1620 2629 719 5096 1244 1842  
## [433] 2576 1512 11054 518 2759 2670 4817 2679 3905 3386 343 4594  
## [445] 3620 1721 3017 754 1950 2924 1659 7238 2764 4679 3092 448  
## [457] 654 1238 1245 3114 2569 5152 1037 3573 1201 3622 960 1163  
## [469] 1209 3077 3757 1418 3518 1934 8318 368 2122 2996 9034 1585  
## [481] 1301 1323 3123 5493 1216 1207 1309 2360 6850 8588 759 4686  
## [493] 2687 585 2255 1361 7127 1203 700 5507 3190 7119 3488 1113  
## [505] 7966 1532 1503 2302 662 2273 2631 1311 3105 2319 3612 7763  
## [517] 3049 1534 2032 6350 2864 1255 1333 2022 1552 626 8858 996  
## [529] 1750 6999 1995 1331 2278 5003 3552 1928 2964 1546 683 12389  
## [541] 4712 1553 1372 3979 6758 3234 5433 806 1082 2788 2930 1927  
## [553] 2820 937 1056 3124 1388 2384 2133 2799 1289 1217 2246 385  
## [565] 1965 1572 2718 1358 931 1442 4241 2775 3863 2329 918 1837  
## [577] 3349 2828 4526 2671 2051 1300 741 3357 3632 1808 12204 9157  
## [589] 3676 3441 640 3652 1530 3914 1858 2600 1979 2116 1437 4042  
## [601] 3660 1444 1980 1355 1376 15653 1493 4370 750 1308 4623 1851  
## [613] 1880 7980 4583 1386 947 684 7476 1922 2303 8086 2346 3973  
## [625] 888 10222 4221 6361 1297 900 1050 1047 6314 3496 3609 4843  
## [637] 4139 5742 10366 2080 2580 4530 5150 5595 1453 1538 2279 5103  
## [649] 9857 6527 1347 2862 2753 3651 975 2896 4716 2284 1103 926  
## [661] 1800 1905 1123 6331 1377 2503 2528 5324 6560 2969 1206 2118  
## [673] 629 1198 2476 1138 14027 7596 1505 3148 6148 1337 1228 790  
## [685] 2570 250 1316 1882 6416 6403 1987 760 2603 3380 3990 11560  
## [697] 4380 6761 4280 2325 1048 3160 2483 14179 1797 2511 1274 5248  
## [709] 3029 428 976 841 5771 1555 1285 1299 1271 691 5045 2124  
## [721] 2214 12680 2463 1155 3108 2901 1655 2812 8065 3275 2223 1480  
## [733] 1371 3535 3509 5711 3872 4933 1940 836 1941 2675 2751 6224  
## [745] 5998 1188 6313 1221 2892 3062 2301 7511 1549 1795 7472 9271  
## [757] 590 930 9283 1778 907 484 9629 3051 3931 7432 1338 1554  
## [769] 15857 1345 1101 3016 2712 731 3780 1602 3966 4165 8335 6681  
## [781] 2375 11816 5084 2327 886 601 2957 2611 5179 2993 1943 1559  
## [793] 3422 3976 1249 2235 1471 10875 894 3343 3959 3577 5804 2169  
## [805] 2439 2210 2221 2389 3331 7409 652 7678 1343 874 3590 1322  
## [817] 3595 1422 6742 7814 9277 2181 1098 4057 795 2825 15672 6614  
## [829] 7824 2442 1829 5800 8947 2606 1592 2186 4153 2625 3485 10477  
## [841] 1278 1107 3763 3711 3594 3195 4454 4736 2991 2142 3161 18424  
## [853] 2848 14896 2359 3345 1817 12749 1366 2002 6872 697 1049 10297  
## [865] 1867 1747 1670 1224 522 1498 745 2063 6288 6842 3527 929  
## [877] 1455 1845 8358 2859 3621 2145 4113 10974 1893 3656 4006 3069  
## [889] 1740 2353 3556 2397 454 1715 2520 3568 7166 3939 1514 7393  
## [901] 1193 7297 2831 753 2427 2538 8386 4844 2923 8229 1433 6289  
## [913] 6579 3565 1569 1936 2390 1736 3857 804 4576

With the head() function we can show the first 6 values of a vector

head(german\_data\_clean$credit\_amount, 10)

## [1] 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234

Show the last items with the tail() function

tail(german\_data\_clean$credit\_amount, 10)

## [1] 3565 1569 1936 3959 2390 1736 3857 804 1845 4576

With the sort() function, we can sort a vector:

sort(german\_data\_clean$credit\_amount)

## [1] 250 276 338 339 343 362 368 385 392 409 426 428  
## [13] 433 433 448 454 458 484 518 522 571 585 590 601  
## [25] 609 609 618 625 626 629 639 640 652 654 660 662  
## [37] 666 672 674 682 683 684 685 691 697 700 701 701  
## [49] 707 708 709 709 717 717 719 727 727 730 731 741  
## [61] 745 750 753 754 759 760 763 766 776 781 783 790  
## [73] 795 797 802 802 804 806 836 841 846 860 866 874  
## [85] 882 884 886 888 894 900 902 907 909 915 918 926  
## [97] 929 930 931 932 932 936 937 939 947 950 951 958  
## [109] 959 960 975 976 976 983 996 999 1007 1024 1028 1037  
## [121] 1038 1042 1047 1048 1049 1050 1053 1055 1056 1068 1076 1082  
## [133] 1082 1092 1098 1101 1103 1107 1108 1113 1123 1126 1126 1131  
## [145] 1136 1138 1149 1154 1154 1155 1158 1163 1164 1168 1169 1169  
## [157] 1175 1185 1188 1190 1193 1198 1199 1199 1200 1201 1203 1204  
## [169] 1206 1207 1209 1213 1216 1216 1217 1221 1223 1224 1225 1228  
## [181] 1231 1231 1236 1236 1237 1237 1238 1239 1240 1240 1244 1245  
## [193] 1246 1249 1255 1258 1258 1258 1262 1262 1262 1264 1264 1271  
## [205] 1274 1275 1275 1275 1278 1282 1282 1283 1285 1287 1287 1288  
## [217] 1289 1291 1295 1295 1297 1299 1300 1301 1308 1309 1311 1313  
## [229] 1316 1318 1322 1323 1330 1331 1333 1337 1338 1343 1344 1344  
## [241] 1345 1346 1347 1352 1355 1358 1360 1361 1364 1364 1366 1371  
## [253] 1372 1374 1374 1376 1377 1381 1382 1382 1386 1386 1388 1391  
## [265] 1393 1393 1393 1402 1403 1409 1409 1410 1410 1412 1413 1413  
## [277] 1414 1418 1422 1424 1424 1433 1437 1442 1442 1444 1445 1449  
## [289] 1449 1453 1455 1459 1469 1471 1473 1474 1474 1478 1478 1478  
## [301] 1480 1484 1493 1494 1495 1498 1501 1503 1503 1505 1512 1514  
## [313] 1516 1520 1521 1525 1526 1530 1532 1533 1533 1534 1537 1538  
## [325] 1542 1543 1544 1546 1546 1549 1552 1553 1553 1554 1555 1559  
## [337] 1567 1568 1569 1572 1574 1577 1582 1585 1591 1592 1595 1597  
## [349] 1597 1602 1603 1620 1647 1655 1657 1659 1670 1680 1715 1721  
## [361] 1736 1740 1743 1743 1747 1750 1755 1766 1768 1778 1795 1797  
## [373] 1800 1804 1808 1817 1819 1820 1823 1829 1835 1837 1842 1845  
## [385] 1845 1851 1858 1860 1864 1867 1872 1880 1881 1882 1884 1887  
## [397] 1893 1898 1901 1905 1908 1913 1919 1919 1922 1924 1924 1925  
## [409] 1927 1928 1934 1935 1935 1936 1938 1940 1940 1941 1943 1950  
## [421] 1953 1957 1961 1963 1965 1967 1977 1979 1980 1984 1987 1995  
## [433] 2002 2012 2022 2028 2028 2030 2032 2039 2039 2051 2058 2063  
## [445] 2064 2069 2073 2080 2096 2100 2101 2108 2116 2118 2121 2122  
## [457] 2124 2132 2133 2134 2136 2141 2142 2145 2146 2149 2150 2169  
## [469] 2171 2171 2181 2186 2197 2210 2212 2214 2221 2223 2225 2235  
## [481] 2238 2241 2241 2246 2247 2249 2251 2255 2273 2278 2279 2284  
## [493] 2288 2292 2299 2301 2302 2303 2315 2319 2320 2323 2325 2326  
## [505] 2327 2329 2331 2333 2333 2337 2346 2348 2353 2359 2360 2366  
## [517] 2375 2384 2384 2389 2390 2394 2397 2404 2406 2406 2415 2424  
## [529] 2427 2439 2442 2445 2462 2463 2473 2476 2483 2503 2507 2511  
## [541] 2515 2520 2522 2528 2538 2569 2570 2576 2577 2578 2578 2579  
## [553] 2580 2600 2603 2606 2611 2613 2622 2625 2629 2631 2631 2647  
## [565] 2659 2662 2670 2671 2675 2679 2684 2687 2697 2708 2712 2718  
## [577] 2728 2743 2745 2746 2748 2751 2753 2759 2760 2762 2764 2767  
## [589] 2775 2779 2782 2788 2799 2812 2820 2825 2828 2831 2835 2848  
## [601] 2859 2862 2864 2872 2892 2896 2899 2901 2910 2923 2924 2930  
## [613] 2957 2964 2969 2978 2978 2991 2993 2996 3001 3016 3017 3017  
## [625] 3021 3029 3031 3049 3051 3059 3060 3062 3069 3074 3077 3077  
## [637] 3079 3092 3104 3105 3108 3114 3123 3124 3148 3149 3160 3161  
## [649] 3181 3186 3190 3195 3213 3229 3234 3235 3244 3249 3275 3331  
## [661] 3342 3343 3345 3349 3349 3357 3368 3378 3380 3384 3386 3394  
## [673] 3398 3399 3414 3416 3422 3430 3441 3446 3447 3448 3485 3488  
## [685] 3496 3499 3509 3512 3518 3527 3535 3552 3556 3565 3566 3568  
## [697] 3573 3577 3578 3590 3590 3594 3595 3599 3609 3612 3617 3617  
## [709] 3620 3621 3622 3632 3643 3650 3651 3652 3656 3660 3676 3711  
## [721] 3749 3757 3758 3763 3777 3780 3804 3812 3832 3832 3835 3844  
## [733] 3850 3857 3863 3868 3872 3878 3905 3913 3914 3915 3931 3939  
## [745] 3949 3959 3959 3965 3966 3972 3973 3976 3979 3990 4006 4020  
## [757] 4042 4042 4057 4110 4113 4139 4151 4153 4165 4169 4210 4221  
## [769] 4241 4249 4272 4272 4280 4281 4297 4308 4351 4370 4380 4439  
## [781] 4454 4455 4463 4473 4526 4526 4530 4576 4583 4591 4594 4605  
## [793] 4611 4623 4657 4675 4679 4686 4712 4716 4736 4746 4771 4788  
## [805] 4795 4796 4811 4817 4843 4844 4870 4933 5003 5045 5084 5096  
## [817] 5103 5117 5129 5150 5152 5179 5190 5234 5248 5293 5302 5324  
## [829] 5371 5381 5433 5493 5507 5511 5595 5711 5742 5743 5771 5800  
## [841] 5801 5804 5842 5848 5866 5943 5951 5954 5954 5965 5998 6070  
## [853] 6078 6110 6143 6148 6187 6199 6204 6224 6229 6260 6288 6289  
## [865] 6304 6313 6314 6331 6350 6361 6403 6416 6419 6458 6468 6468  
## [877] 6527 6560 6568 6579 6614 6615 6681 6742 6758 6761 6761 6836  
## [889] 6842 6850 6872 6887 6948 6967 6999 7057 7119 7127 7166 7174  
## [901] 7228 7238 7253 7297 7308 7374 7393 7408 7409 7418 7432 7472  
## [913] 7476 7485 7511 7582 7596 7629 7678 7685 7721 7758 7763 7814  
## [925] 7824 7855 7865 7882 7966 7980 8065 8072 8086 8133 8229 8318  
## [937] 8335 8358 8386 8471 8487 8588 8613 8648 8858 8947 8978 9034  
## [949] 9055 9157 9271 9277 9283 9398 9436 9566 9572 9629 9857 9960  
## [961] 10127 10144 10222 10297 10366 10477 10623 10722 10875 10961 10974 11054  
## [973] 11328 11560 11590 11760 11816 11938 11998 12169 12204 12389 12579 12612  
## [985] 12680 12749 12976 13756 14027 14179 14318 14421 14555 14782 14896 15653  
## [997] 15672 15857 15945 18424

With the decreased=TRUE parameter, we can sort on a *descending* order

With the head() and sort() function combined, we can create a **top 10**

head(sort(german\_data\_clean$credit\_amount, decreasing=TRUE), 10)

## [1] 18424 15945 15857 15672 15653 14896 14782 14555 14421 14318

Combined head, sort and table functions to create a top list with names and values

head(sort(table(german\_data\_clean$purpose), decreasing=TRUE), 3)

##   
## radio/television car (new) furniture/equipment   
## 280 234 181

### Exploring the dimensions of a Table:

Show the number of rows in a Data Frame with the nrow function

nrow(german\_data\_clean)

## [1] 1000

Show the number of columns in a Data Frame with the ncol function

ncol(german\_data\_clean)

## [1] 21

Show the *dimensions* of the Data Frame (both rows and columns) with the dim function

dim(german\_data\_clean)

## [1] 1000 21

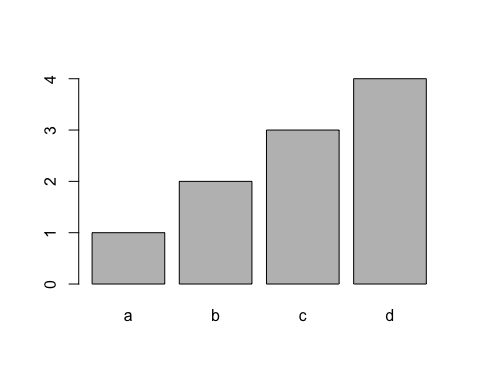
str(german\_data\_clean)

## 'data.frame': 1000 obs. of 21 variables:  
## $ checking\_account : chr "... < 0 euro" "0 <= ... < 102 euro" "No checking account" "... < 0 euro" ...  
## $ duration\_months : int 6 48 12 42 24 36 24 36 12 30 ...  
## $ credit\_history : chr "critical account/other credits existing (not at this bank)" "existing credits paid back duly till now" "critical account/other credits existing (not at this bank)" "existing credits paid back duly till now" ...  
## $ purpose : chr "radio/television" "radio/television" "education" "furniture/equipment" ...  
## $ credit\_amount : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...  
## $ savings : chr "unknown/ no savings account" "... < 51 euro" "... < 51 euro" "... < 51 euro" ...  
## $ present\_employment\_since : chr ".. >= 7 years" "1 <= ... < 4 years" "4 <= ... < 7 years" "4 <= ... < 7 years" ...  
## $ installment\_rate : int 4 2 2 2 3 2 3 2 2 4 ...  
## $ personal\_status\_sex : chr "male : single" "female : divorced/separated/married" "male : single" "male : single" ...  
## $ other\_deptors : chr "none" "none" "none" "guarantor" ...  
## $ present\_residence : int 4 2 3 4 4 4 4 2 4 2 ...  
## $ property : chr "real estate" "real estate" "real estate" "(if not real estate) building society savings agreement/ life insurance" ...  
## $ age\_years : int 67 22 49 45 53 35 53 35 61 28 ...  
## $ other\_installment\_plans : chr "none" "none" "none" "none" ...  
## $ housing : chr "own" "own" "own" "for free" ...  
## $ existing\_credits : int 2 1 1 1 2 1 1 1 1 2 ...  
## $ job : chr "skilled employee / official" "skilled employee / official" "unskilled - resident" "skilled employee / official" ...  
## $ people\_liable\_maintenance: int 1 1 2 2 2 2 1 1 1 1 ...  
## $ telephone : chr "yes, registered under the customers name" "none" "none" "none" ...  
## $ foreign\_worker : chr "yes" "yes" "yes" "yes" ...  
## $ response : int 0 1 0 0 1 0 0 0 0 1 ...

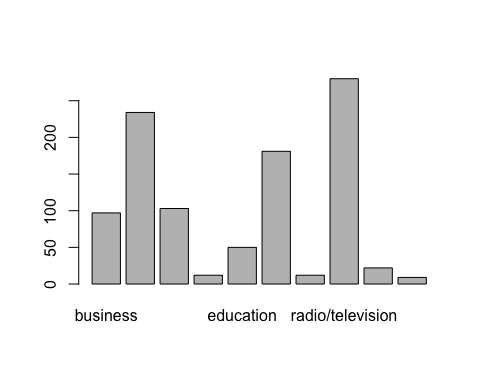
## Explorering the data frame with visualisations

### Barplot

barplot(c(1,2,3,4), names.arg = c("a", "b", "c", "d"))



barplot(table(german\_data\_clean$purpose))

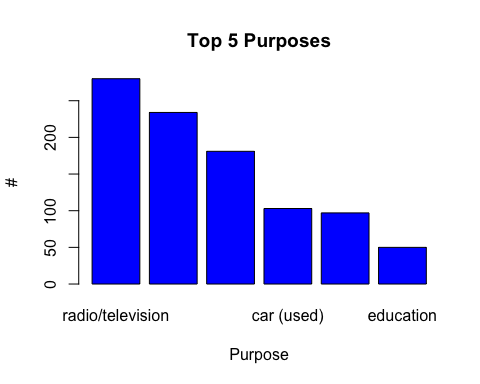
 Example named vectors:

ages <- c(20, 30, 40)  
names(ages) <- c("arie", "jaap", "dirk")  
sum(ages )

## [1] 90

Creating a sorted barplot

barplot(  
 head(  
 sort(  
 table(german\_data\_clean$purpose), decreasing=TRUE)),  
 col = "blue", main="Top 5 Purposes", xlab="Purpose", ylab = "#")

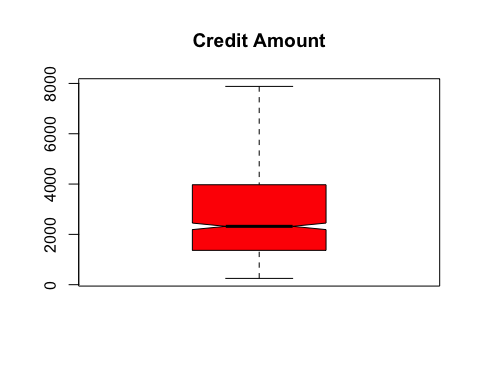


sort(table(german\_data\_clean$purpose), decreasing=TRUE)

##   
## radio/television car (new) furniture/equipment car (used)   
## 280 234 181 103   
## business education repairs domestic appliances   
## 97 50 22 12   
## others retraining   
## 12 9

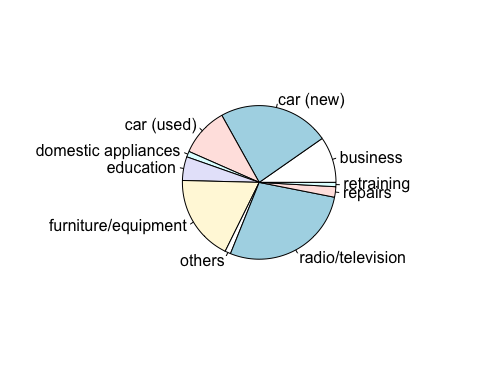
### Boxplot

standard\_color <- "red"  
  
boxplot(german\_data\_clean$credit\_amount, col=standard\_color, outline = FALSE, notch = TRUE, main="Credit Amount")



### Pie chart

pie(table(german\_data\_clean$purpose))



unique(german\_data\_clean$purpose)

## [1] "radio/television" "education" "furniture/equipment"  
## [4] "car (new)" "car (used)" "business"   
## [7] "domestic appliances" "repairs" "others"   
## [10] "retraining"

custom\_colours <- c("darkred", "red", "darkblue", "blue", "darkgreen", "green", "yellow", "orange", "gray", "darkgray")

print(unique(german\_data\_clean$purpose))

## [1] "radio/television" "education" "furniture/equipment"  
## [4] "car (new)" "car (used)" "business"   
## [7] "domestic appliances" "repairs" "others"   
## [10] "retraining"

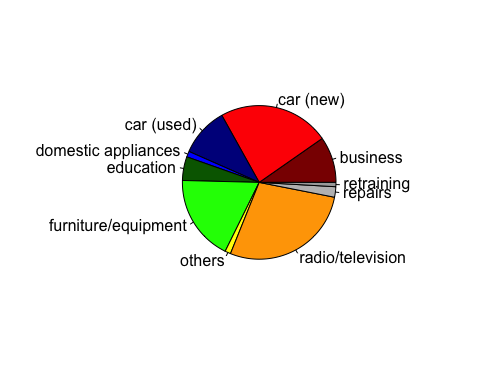
print("---")

## [1] "---"

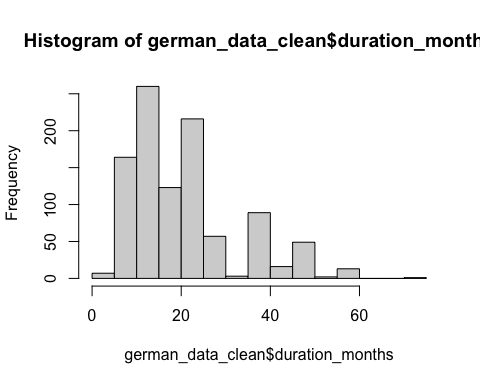
print(custom\_colours)

## [1] "darkred" "red" "darkblue" "blue" "darkgreen" "green"   
## [7] "yellow" "orange" "gray" "darkgray"

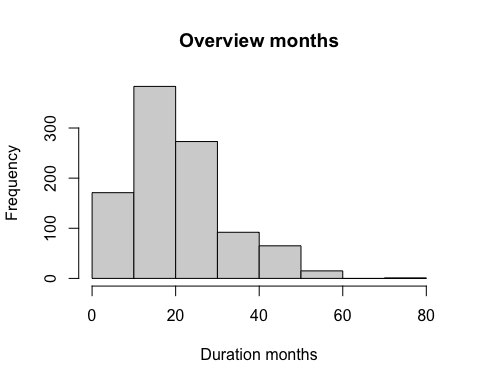
pie(table(german\_data\_clean$purpose), col = custom\_colours)

 ### Histogram

hist(german\_data\_clean$duration\_months)

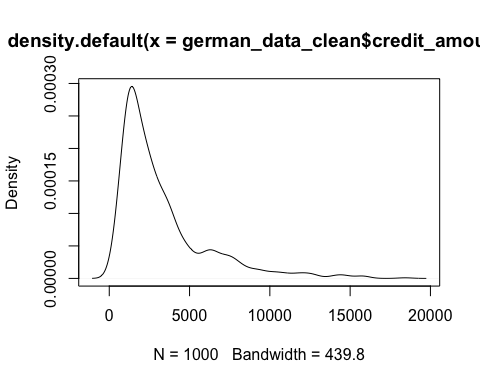


hist(german\_data\_clean$duration\_months, main = "Overview months", xlab = "Duration months", breaks = 5)



### Using a density plot in R

plot(density(german\_data\_clean$credit\_amount))



plot(german\_data\_clean$credit\_amount)

