LAB\_DA\_2 SLOT:L3+L4

20BDS0354 A SAI CHARAN

2023-09-04

# Importing Libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Read 50% of records from file “data\_cat1.csv” with the following columns only

TotalRows<-nrow(read.csv("C:/Users/saic3/CSE3046-F2-LAB\_SLOT\_L3+L4/Datasets/data\_cat1.csv"))  
  
  
df<-read.csv("C:/Users/saic3/CSE3046-F2-LAB\_SLOT\_L3+L4/Datasets/data\_cat1.csv",nrows = TotalRows/2)%>%select("dep\_time","sched\_dep\_time","arr\_time","sched\_arr\_time","carrier" ,"flight" ,  
"origin","dest")  
  
  
head(df)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier flight origin dest  
## 1 517 515 830 819 UA 1545 EWR IAH  
## 2 533 529 850 830 UA 1714 LGA IAH  
## 3 542 540 923 850 AA 1141 JFK MIA  
## 4 544 545 1004 1022 B6 725 JFK BQN  
## 5 554 600 812 837 DL 461 LGA ATL  
## 6 554 558 740 728 UA 1696 EWR ORD

cat("Taken Rows are",nrow(df))

## Taken Rows are 168388

cat(" \nTotal Rows are",TotalRows)

##   
## Total Rows are 336776

# 2. How many variables have missing values?

colnames(df)[colSums(is.na(df)) > 0]

## [1] "dep\_time" "arr\_time"

# Remove all missing records and save them in another csv file.

df<-drop\_na(df)  
library(readxl)  
write.csv(df,"ModifiedDataCAT-I")  
apply(df,2,anyNA)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier   
## FALSE FALSE FALSE FALSE FALSE   
## flight origin dest   
## FALSE FALSE FALSE

# 3.Create new attributes for departure delay, and arrival delay.

class(df$dep\_time)

## [1] "integer"

class(df$sched\_dep\_time)

## [1] "integer"

class(df$arr\_time)

## [1] "integer"

class(df$sched\_arr\_time)

## [1] "integer"

# 3.Create new attributes for departure delay, and arrival delay.

C<-ifelse(df$dep\_time<1000,paste0("0",df$dep\_time),df$dep\_time)  
 D<-ifelse(df$sched\_dep\_time<1000,paste0("0",df$sched\_dep\_time),df$sched\_dep\_time)  
 h1<-as.integer(substring(C,1,2))-as.integer(substring(D,1,2))  
 m1<-as.integer(substring(C,3,4))-as.integer(substring(D,3,4))  
 Out<-h1\*60+m1  
 df<-df%>%mutate(dep\_delay=Out)

C<-ifelse(df$arr\_time<1000,paste0("0",df$arr\_time),df$arr\_time)  
 D<-ifelse(df$sched\_arr\_time<1000,paste0("0",df$sched\_arr\_time),df$sched\_arr\_time)  
 h1<-as.integer(substring(C,1,2))-as.integer(substring(D,1,2))  
 m1<-as.integer(substring(C,3,4))-as.integer(substring(D,3,4))  
 Out<-h1\*60+m1  
 df<-df%>%mutate(arr\_delay=Out)  
 head(df)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier flight origin dest  
## 1 517 515 830 819 UA 1545 EWR IAH  
## 2 533 529 850 830 UA 1714 LGA IAH  
## 3 542 540 923 850 AA 1141 JFK MIA  
## 4 544 545 1004 1022 B6 725 JFK BQN  
## 5 554 600 812 837 DL 461 LGA ATL  
## 6 554 558 740 728 UA 1696 EWR ORD  
## dep\_delay arr\_delay  
## 1 2 11  
## 2 4 20  
## 3 2 33  
## 4 -1 -18  
## 5 -6 -25  
## 6 -4 12

# 4.a Had an arrival delay of two or more hours

head(df%>%filter(arr\_delay>=120),n = 50)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier flight origin dest  
## 1 811 630 1047 830 MQ 4576 LGA CLT  
## 2 957 733 1056 853 UA 856 EWR BOS  
## 3 1114 900 1447 1222 UA 1086 LGA IAH  
## 4 1505 1310 1638 1431 EV 4497 EWR RIC  
## 5 1525 1340 1831 1626 B6 525 EWR MCO  
## 6 1528 1459 2002 1647 EV 3806 EWR STL  
## 7 1549 1445 1912 1656 EV 4181 EWR MCI  
## 8 1558 1359 1718 1515 EV 5712 JFK IAD  
## 9 1732 1630 2028 1825 EV 4092 EWR DAY  
## 10 1803 1620 2008 1750 MQ 4622 LGA BNA  
## 11 1815 1325 2120 1542 EV 4417 EWR OMA  
## 12 1842 1422 1958 1535 EV 4633 EWR BTV  
## 13 1856 1645 2212 2005 AA 181 JFK LAX  
## 14 1934 1725 2126 1855 MQ 4255 JFK BNA  
## 15 1938 1703 2109 1823 EV 4300 EWR RIC  
## 16 1942 1705 2124 1830 MQ 4410 JFK DCA  
## 17 2006 1630 2230 1848 EV 4644 EWR SAV  
## 18 2009 1808 2145 1942 EV 4440 EWR PIT  
## 19 2103 2110 2345 17 DL 1668 JFK LAS  
## 20 2115 1700 2330 1920 9E 3347 JFK CVG  
## 21 2119 1930 2358 2136 EV 4543 EWR DSM  
## 22 2209 2145 58 37 B6 35 JFK PBI  
## 23 2221 2000 2331 2124 EV 4462 EWR BUF  
## 24 817 630 1107 845 EV 4235 EWR IND  
## 25 833 558 1018 727 UA 651 EWR ORD  
## 26 905 822 1313 1045 EV 4140 EWR XNA  
## 27 923 624 1051 758 EV 4104 EWR PIT  
## 28 1125 925 1445 1146 9E 3658 LGA GRR  
## 29 1244 900 1431 1104 EV 4412 EWR MYR  
## 30 1332 904 1616 1128 EV 4364 EWR MCI  
## 31 1412 838 1710 1147 UA 468 EWR MCO  
## 32 1451 1232 1749 1533 UA 1121 EWR FLL  
## 33 1548 1340 1710 1500 EV 4617 EWR PIT  
## 34 1607 1030 2003 1355 AA 179 JFK SFO  
## 35 1710 1526 1857 1655 EV 4502 EWR BNA  
## 36 1728 1515 1843 1632 EV 4133 EWR IAD  
## 37 1751 1450 2041 1755 AA 1813 JFK MCO  
## 38 1755 1547 1929 1708 EV 3272 EWR BUF  
## 39 1806 1629 2008 1808 EV 4308 EWR RDU  
## 40 1845 1645 2137 1859 9E 3983 EWR CVG  
## 41 1848 1701 2112 1856 EV 4202 EWR STL  
## 42 1849 1724 2235 1938 EV 4321 EWR MCI  
## 43 1909 1621 2116 1816 EV 4092 EWR DAY  
## 44 2005 1705 2126 1830 MQ 4410 JFK DCA  
## 45 2043 1759 2251 1949 EV 4581 EWR CMH  
## 46 2113 2110 2346 17 DL 1627 JFK LAS  
## 47 2115 1837 2322 2045 EV 4125 EWR GRR  
## 48 2128 2130 43 18 B6 383 LGA FLL  
## 49 2131 1512 2340 1741 UA 488 LGA DEN  
## 50 235 2359 700 437 B6 727 JFK BQN  
## dep\_delay arr\_delay  
## 1 101 137  
## 2 144 123  
## 3 134 145  
## 4 115 127  
## 5 105 125  
## 6 29 195  
## 7 64 136  
## 8 119 123  
## 9 62 123  
## 10 103 138  
## 11 290 338  
## 12 260 263  
## 13 131 127  
## 14 129 151  
## 15 155 166  
## 16 157 174  
## 17 216 222  
## 18 121 123  
## 19 -7 1358  
## 20 255 250  
## 21 109 142  
## 22 24 121  
## 23 141 127  
## 24 107 142  
## 25 155 171  
## 26 43 148  
## 27 179 173  
## 28 120 179  
## 29 224 207  
## 30 268 288  
## 31 334 323  
## 32 139 136  
## 33 128 130  
## 34 337 368  
## 35 104 122  
## 36 133 131  
## 37 181 166  
## 38 128 141  
## 39 97 120  
## 40 120 158  
## 41 107 136  
## 42 85 177  
## 43 168 180  
## 44 180 176  
## 45 164 182  
## 46 3 1359  
## 47 158 157  
## 48 -2 175  
## 49 379 359  
## 50 -1284 143

cat("There are Totally ",nrow(df%>%filter(arr\_delay>=120)))

## There are Totally 3211

cat("Rows Actually But Truncated Print to 1000")

## Rows Actually But Truncated Print to 1000

# 4.b Flew to DEN or JFK

head(df%>%filter(dest=="DEN"|dest=="JFK"),n = 50)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier flight origin dest  
## 1 646 645 910 916 UA 883 LGA DEN  
## 2 727 730 959 952 UA 1162 EWR DEN  
## 3 752 750 1025 1029 UA 477 LGA DEN  
## 4 754 755 1103 1030 WN 733 LGA DEN  
## 5 813 815 1103 1056 DL 914 LGA DEN  
## 6 833 835 1134 1102 F9 835 LGA DEN  
## 7 906 843 1134 1125 UA 1643 EWR DEN  
## 8 929 925 1220 1150 WN 766 EWR DEN  
## 9 1005 1000 1239 1234 UA 1625 EWR DEN  
## 10 1123 1110 1410 1336 UA 405 LGA DEN  
## 11 1253 1212 1524 1436 UA 754 EWR DEN  
## 12 1255 1255 1540 1535 WN 1251 LGA DEN  
## 13 1506 1512 1723 1741 UA 407 LGA DEN  
## 14 1531 1520 1809 1750 UA 365 EWR DEN  
## 15 1603 1600 1839 1830 WN 591 EWR DEN  
## 16 1610 1555 1852 1834 DL 1939 LGA DEN  
## 17 1628 1630 1907 1923 DL 920 JFK DEN  
## 18 1634 1626 1913 1852 UA 69 EWR DEN  
## 19 1712 1719 1939 1958 UA 509 LGA DEN  
## 20 1716 1730 1947 1953 F9 511 LGA DEN  
## 21 1827 1829 2105 2056 UA 1139 EWR DEN  
## 22 2129 2120 2342 2351 B6 97 JFK DEN  
## 23 647 645 849 916 UA 320 LGA DEN  
## 24 728 730 1001 952 UA 311 EWR DEN  
## 25 740 740 1025 1015 WN 1581 LGA DEN  
## 26 741 740 1005 1019 UA 328 LGA DEN  
## 27 811 815 1100 1056 DL 914 LGA DEN  
## 28 827 835 1120 1102 F9 835 LGA DEN  
## 29 844 843 1114 1125 UA 1643 EWR DEN  
## 30 932 925 1157 1155 WN 1484 EWR DEN  
## 31 1004 1003 1235 1237 UA 1625 EWR DEN  
## 32 1123 1110 1434 1340 UA 1477 LGA DEN  
## 33 1209 1212 1451 1440 UA 1492 EWR DEN  
## 34 1258 1255 1537 1535 WN 1251 LGA DEN  
## 35 1521 1520 1745 1754 UA 1107 EWR DEN  
## 36 1524 1500 1803 1743 DL 1939 LGA DEN  
## 37 1616 1610 1839 1840 WN 2305 EWR DEN  
## 38 1629 1630 1905 1923 DL 920 JFK DEN  
## 39 1644 1625 1915 1851 UA 1556 EWR DEN  
## 40 1712 1700 1927 1939 UA 509 LGA DEN  
## 41 1728 1730 1952 1953 F9 511 LGA DEN  
## 42 1831 1829 2104 2056 UA 1462 EWR DEN  
## 43 1907 1815 2137 2052 DL 884 LGA DEN  
## 44 2116 2120 2342 2351 B6 97 JFK DEN  
## 45 2131 1512 2340 1741 UA 488 LGA DEN  
## 46 644 645 847 918 UA 338 LGA DEN  
## 47 702 658 906 915 UA 510 EWR DEN  
## 48 757 755 1008 1030 WN 733 LGA DEN  
## 49 801 759 1025 1025 UA 429 LGA DEN  
## 50 815 817 1048 1058 DL 914 LGA DEN  
## dep\_delay arr\_delay  
## 1 1 -6  
## 2 -3 7  
## 3 2 -4  
## 4 -1 33  
## 5 -2 7  
## 6 -2 32  
## 7 23 9  
## 8 4 30  
## 9 5 5  
## 10 13 34  
## 11 41 48  
## 12 0 5  
## 13 -6 -18  
## 14 11 19  
## 15 3 9  
## 16 15 18  
## 17 -2 -16  
## 18 8 21  
## 19 -7 -19  
## 20 -14 -6  
## 21 -2 9  
## 22 9 -9  
## 23 2 -27  
## 24 -2 9  
## 25 0 10  
## 26 1 -14  
## 27 -4 4  
## 28 -8 18  
## 29 1 -11  
## 30 7 2  
## 31 1 -2  
## 32 13 54  
## 33 -3 11  
## 34 3 2  
## 35 1 -9  
## 36 24 20  
## 37 6 -1  
## 38 -1 -18  
## 39 19 24  
## 40 12 -12  
## 41 -2 -1  
## 42 2 8  
## 43 52 45  
## 44 -4 -9  
## 45 379 359  
## 46 -1 -31  
## 47 4 -9  
## 48 2 -22  
## 49 2 0  
## 50 -2 -10

abc<-nrow(df%>%filter(dest=="DEN"|dest=="JFK"))  
cat("There are Totally ",abc)

## There are Totally 3584

cat(" Rows Actually But Truncated Result")

## Rows Actually But Truncated Result

# 4.c Arrived more than two hours late, but didn’t leave late

head(df%>%filter(arr\_delay>120 & dep\_delay<=0),n = 50)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier flight origin dest  
## 1 2103 2110 2345 17 DL 1668 JFK LAS  
## 2 2128 2130 43 18 B6 383 LGA FLL  
## 3 235 2359 700 437 B6 727 JFK BQN  
## 4 1025 1032 1521 1240 EV 4255 EWR CHS  
## 5 2129 2130 2326 18 B6 199 JFK LAS  
## 6 2155 2159 52 100 B6 11 JFK FLL  
## 7 2059 2100 2349 31 DL 2363 JFK LAX  
## 8 2156 2159 46 100 B6 11 JFK FLL  
## 9 37 2230 341 131 B6 11 JFK FLL  
## 10 2116 2130 2400 18 B6 383 LGA FLL  
## 11 2058 2100 2350 31 DL 2363 JFK LAX  
## 12 1433 1436 1844 1543 EV 4372 EWR DCA  
## 13 2131 2135 2358 24 B6 43 JFK MCO  
## 14 2121 2130 2354 25 B6 383 LGA FLL  
## 15 1718 1725 2339 2050 UA 512 JFK SFO  
## 16 2100 2100 2356 31 DL 2363 JFK LAX  
## 17 1504 1506 1937 1641 UA 1702 EWR ORD  
## 18 842 845 1630 1350 AA 1357 JFK SJU  
## 19 616 625 1203 937 UA 304 LGA IAH  
## 20 2054 2100 2359 31 DL 2363 JFK LAX  
## 21 644 655 1624 1030 DL 1415 JFK SLC  
## 22 829 830 1633 1146 UA 1601 EWR FLL  
## 23 1100 1100 1815 1420 DL 1275 JFK SLC  
## 24 1724 1725 2257 2050 UA 512 JFK SFO  
## 25 1419 1420 1754 1550 MQ 3728 EWR ORD  
## 26 921 929 1410 1044 EV 4636 EWR DCA  
## 27 1159 1205 1813 1520 AA 743 LGA DFW  
## 28 1234 1240 1855 1540 AA 1853 EWR DFW  
## 29 1242 1245 1907 1550 AA 745 LGA DFW  
## 30 1424 1430 2058 1735 AA 883 EWR DFW  
## 31 1451 1455 1908 1645 AA 337 LGA ORD  
## 32 2056 2100 2348 18 DL 427 JFK LAX  
## 33 2126 2137 2356 27 B6 1371 LGA FLL  
## 34 2105 2113 2358 30 UA 617 EWR SFO  
## 35 800 805 1323 1022 EV 4364 EWR MCI  
## 36 2128 2137 2356 27 B6 1371 LGA FLL  
## 37 1507 1515 1947 1700 AA 341 LGA ORD  
## 38 2119 2137 2353 27 B6 1371 LGA FLL  
## 39 2245 2255 2400 12 B6 2002 JFK BUF  
## 40 2023 2025 2359 26 UA 1071 EWR BQN  
## 41 2237 2255 2346 12 B6 2002 JFK BUF  
## 42 1350 1350 1736 1526 EV 5181 LGA MSN  
## 43 1357 1359 1858 1654 AA 1151 LGA DFW  
## 44 2055 2100 2329 20 DL 427 JFK LAX  
## 45 2105 2113 2349 30 UA 242 EWR SFO  
## 46 2056 2100 2347 20 DL 427 JFK LAX  
## 47 2108 2113 2342 30 UA 617 EWR SFO  
## 48 2248 2255 2359 12 B6 2002 JFK BUF  
## 49 2057 2100 2340 20 DL 427 JFK LAX  
## 50 758 800 1240 1003 9E 3353 JFK DTW  
## dep\_delay arr\_delay  
## 1 -7 1358  
## 2 -2 175  
## 3 -1284 143  
## 4 -7 161  
## 5 -1 1338  
## 6 -4 242  
## 7 -1 1248  
## 8 -3 186  
## 9 -1163 130  
## 10 -14 1372  
## 11 -2 1249  
## 12 -3 181  
## 13 -4 1314  
## 14 -9 1309  
## 15 -7 169  
## 16 0 1255  
## 17 -2 176  
## 18 -3 160  
## 19 -9 146  
## 20 -6 1258  
## 21 -11 354  
## 22 -1 287  
## 23 0 235  
## 24 -1 127  
## 25 -1 124  
## 26 -8 206  
## 27 -6 173  
## 28 -6 195  
## 29 -3 197  
## 30 -6 203  
## 31 -4 143  
## 32 -4 1360  
## 33 -11 1309  
## 34 -8 1258  
## 35 -5 181  
## 36 -9 1309  
## 37 -8 167  
## 38 -18 1306  
## 39 -10 1378  
## 40 -2 1313  
## 41 -18 1364  
## 42 0 130  
## 43 -2 124  
## 44 -5 1289  
## 45 -8 1249  
## 46 -4 1307  
## 47 -5 1242  
## 48 -7 1377  
## 49 -3 1300  
## 50 -2 157

cat("There are Totally ",nrow(df%>%filter(arr\_delay>120 & dep\_delay<=0)))

## There are Totally 228

# 5. Which carrier has the worst delays?

#Lets Check If Null Exists Due to Variable Creation

apply(df,2,anyNA)

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier   
## FALSE FALSE FALSE FALSE FALSE   
## flight origin dest dep\_delay arr\_delay   
## FALSE FALSE FALSE TRUE TRUE

# So We Remove Or We Doesnt Consider Them While Calculating MaxDelays So na.Rm=TRUE Below

WorseDelays<-df%>%group\_by(carrier)%>%summarise(maxarrdel=max(arr\_delay,na.rm = T),maxdepdelay=max(dep\_delay,na.rm = T),totalDelay=maxarrdel+maxdepdelay)%>%arrange(desc(totalDelay))  
WorseDelays

## # A tibble: 16 × 4  
## carrier maxarrdel maxdepdelay totalDelay  
## <chr> <dbl> <dbl> <dbl>  
## 1 DL 1378 911 2289  
## 2 B6 1378 502 1880  
## 3 UA 1378 408 1786  
## 4 AA 1138 636 1774  
## 5 F9 242 853 1095  
## 6 EV 538 548 1086  
## 7 FL 436 470 906  
## 8 9E 396 408 804  
## 9 YV 381 387 768  
## 10 WN 385 378 763  
## 11 MQ 366 361 727  
## 12 VX 344 367 711  
## 13 US 336 374 710  
## 14 AS 198 222 420  
## 15 HA 154 206 360  
## 16 OO 107 67 174

CarrierNa<-WorseDelays%>%filter(totalDelay==max(totalDelay))%>%select(carrier)  
cat("So The Carrier With Maximum/Worse Delay is",as.character(CarrierNa))

## So The Carrier With Maximum/Worse Delay is DL

# 6. Count all flights operating between JFK and MIA airports.

head(df%>%filter((origin=="JFK"&dest=="MIA")|(origin=="MIA"&dest=="JFK")))

## dep\_time sched\_dep\_time arr\_time sched\_arr\_time carrier flight origin dest  
## 1 542 540 923 850 AA 1141 JFK MIA  
## 2 759 800 1057 1127 DL 1843 JFK MIA  
## 3 826 715 1136 1045 AA 443 JFK MIA  
## 4 912 900 1241 1220 AA 647 JFK MIA  
## 5 1306 1240 1622 1555 AA 2041 JFK MIA  
## 6 1446 1455 1803 1825 AA 1769 JFK MIA  
## dep\_delay arr\_delay  
## 1 2 33  
## 2 -1 -30  
## 3 71 51  
## 4 12 21  
## 5 26 27  
## 6 -9 -22

count(df%>%filter((origin=="JFK"&dest=="MIA")|(origin=="MIA"&dest=="JFK")))

## n  
## 1 1677

# 7. Group by column “career” and “origin” and extract group class (‘UA’, ‘JFK’)

grp<-df%>%group\_by(carrier,origin)%>%summarise()

## `summarise()` has grouped output by 'carrier'. You can override using the  
## `.groups` argument.

class(grp%>%filter(carrier=="UA"&origin=="JFK"))

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

# From Question-8 We Use Data.csv

#Importing DataSet

df2<-read.csv("C:/Users/saic3/CSE3046-F2-LAB\_SLOT\_L3+L4/In-Lab-Exercise-Dataset/Data.csv")

## Changing Dataset From Character Forms to Primitive Forms

df2$Tv.Size..Inches.=as.numeric(df2$Tv.Size..Inches.)

## Warning: NAs introduced by coercion

df2$OpSys=as.factor(df2$OpSys)  
df2$OpSys<-as.numeric(df2$OpSys)  
df2["OpSys"][df2["OpSys"]==1]<-NA  
  
  
df2$DOB<-parse\_date\_time(df2$DOB,"dmy")

## Warning: 5 failed to parse.

df2$DOB<-as.Date(df2$DOB)

df2<-df2 %>%  
 fill(OpSys, .direction = 'up')  
df2<-df2 %>%  
 fill(Tv.Size..Inches., .direction = 'up')  
df2<-df2 %>%  
 fill(DOB, .direction = 'up')

# 8. Label TV size in small (10,20), medium (21,30), and large (31,40).

# Without Conversion

df2%>%pull(Tv.Size..Inches.)%>%cut(breaks = c(10,21,31,40),labels = c("Small","Medium","Large"))

## [1] Small Small Small Small Small Small Small Small Small Small   
## [11] Small Small Small Small Small Small Small Small Small Small   
## [21] Small Small Small Small Small Small Small Small Small Small   
## [31] Small Small Small Small Small Small Small Small Small Small   
## [41] Small Small Small Small Small Small Small Small Small Small   
## [51] Small Small Small Small Small Small Small Small Small Small   
## [61] Small Small Small Small Small Small Small Small Small Small   
## [71] Small Small Small Small Small Small Small Small Small Small   
## [81] Small Small Small Small Small Small Small Small Small Small   
## [91] Small Small Small Small Small Small Small Small Small Small   
## [101] Small Small Small Small Small Small Small Small Small Small   
## [111] Small Small Small Small Small Small Small Small Small Small   
## [121] Small Small Small Small Small Small Small Small Small Small   
## [131] Small Small Small Small Small Small Small Small Small Small   
## [141] Small Small Small Small Small Small Small Small Small Small   
## [151] Small Small Small Small Small Small Small Small Small Small   
## [161] Small Small Small Small Small Small Small Small Small Small   
## [171] Small Small Small Small Small Small Small Small Small Small   
## [181] Small Small Small Small Small Small Small Small Small Small   
## [191] Small Small Small Small Small Small Small Small Small Small   
## [201] Small Small Small Small Small Small Small Small Small Small   
## [211] Small Small Small Small Small Small Small Small Small Small   
## [221] Small Small Small Small Small Small Small Small Small Small   
## [231] Small Small Small Small Small Small Small Small Small Small   
## [241] Small Small Small Small Small Small Small Small Small Small   
## [251] Small Small Small Small Small Small Small Small Small Small   
## [261] Small Small Small Small Small Small Small Small Small Small   
## [271] Small Small Small Small Small Small Small Small Small Small   
## [281] Small Small Small Small Small Small Small Small Small Small   
## [291] Small Small Small Small Small Small Small Small Small Small   
## [301] Small Small Small Small Small Small Small Small Small Small   
## [311] Small Small Small Small Small Small Small Small Small Small   
## [321] Small Small Small Small Small Small Small Small Small Small   
## [331] Small Small Small Small Small Small Small Small Small Small   
## [341] Small Small Small Small Small Small Small Small Small Small   
## [351] Small Small Small Small Small Small Small Small Small Small   
## [361] Small Small Small Small Small Small Small Small Small Small   
## [371] Small Small Small Small Small Small Small Small Small Small   
## [381] Small Small Small Small Small Small Small Small Small Small   
## [391] Small Small Small Small Small Small Small Small Small Small   
## [401] Small Small Medium Small Small Small Small Small Large Small   
## [411] Small Small Small Small Small Small Small Small Small Small   
## [421] Small Small Small Large Medium Small Small Small Small Small   
## [431] Small Small Small Small Small Small Small Small Medium Small   
## [441] Small Small Small Small Small Small Small Small Small Small   
## [451] Small Small Small Small Small Small Small Small Large Small   
## [461] Small Small Small Small Small Small Small Small Small Small   
## [471] Small Small Small Small Small Small Small Small Small Small   
## [481] Small Small Small Small Small Small Small Small Small Small   
## [491] Small Small Small Small Small Small Small Small Small Small   
## [501] Small Small Small Small Small Small Small Small Small Small   
## [511] Small Small Small Small Small Small Small Small Small Small   
## [521] Small Small Small Small Small Medium Small Small Small Small   
## [531] Small Small Small Small Small Small Small Small Small Small   
## [541] Small Small Small Small Small Small Small Small Small Small   
## [551] Small Small Small Small Small Small Small Small Small Small   
## [561] Small Small Small Small Small Small Small Small Small Small   
## [571] Small Small Small Small Small Small Small Small Small Small   
## [581] Small Small Small Small Small Small Small Small Small Small   
## [591] Small Small Large Small Small Small Small Small Small Small   
## [601] Small Small Small Small Small Small Small Small Small Small   
## [611] Small Small Small Small Small Small Small Small Small Small   
## [621] Small Small Small Small Small Small Small Small Small Small   
## [631] Small Small Small Small Small Small Small Small Small Small   
## [641] Small Small Small Small Small Small Small Small Small Small   
## [651] Small Small Small Small Small Small Small Small Small Small   
## [661] Small Small Small Small Small Small Small Small Small Small   
## [671] Small Small Small Large Small Small Small Small Small Small   
## [681] Small Small Small Small Small Small Small Small Small Small   
## [691] Small Small Small Small Small Small Small Small Small Small   
## [701] Small Small Small Small Small Small Small Small Small Small   
## [711] Small Small Small Small Small Small Small Small Small Small   
## [721] Small Small Small Small Small Small Small Small Small Small   
## [731] Small Small Small Small Small Small Small Small Small Small   
## [741] Small Small Small Small Small Small Small Small Small Small   
## [751] Small Small Small Small Small Small Small Small Small Small   
## [761] Small Small Small Small Small Small Small Small Small Small   
## [771] Small Small Small Small Small Small Small Small Small Small   
## [781] Small Small Small Small Small Small Small Small Small Small   
## [791] Small Small Small Small Small Small Small Small Small Small   
## [801] Small Small Small Small Small Small Small Small Small Small   
## [811] Small Small Small Small Small Small Small Small Small Small   
## [821] Small Small Small Small Small Small Small Small Small Small   
## [831] Small Small Small Small Small Small Small Small Small Small   
## [841] Small Small Small Small Small Small Small Small Small Small   
## [851] Small Small Small Small Small Small Small Small Small Small   
## [861] Small Small Small Small Small Small Small Small Small Small   
## [871] Small Small Small Small Small Small Small Small Small Small   
## [881] Small Small Small Small Small Small Small Small Small Small   
## [891] Small Small Small Small Small Small Small Small Small Small   
## [901] Small Small Small Small Small Small Small Small Small Small   
## [911] Small Small Small Small Small Small Small Small Small Small   
## [921] Small Small Small Small Small Small Small Small Small Small   
## [931] Small Small Small Small Small Small Small Small Small Small   
## [941] Small Small Small Small Small Small Small Small Small Small   
## [951] Small Small Small Small Small Small Small Small Small Small   
## [961] Small Small Small Small Small Small Small Small Small Small   
## [971] Small Small Small Small Small Small Small Small Small Small   
## [981] Small Small Small Small Small Small Small Small Small Small   
## [991] Small Small Small Small Small Small Small Small Small Small   
## [1001] Small Small Small Small Small Small Small Small Small Small   
## [1011] Small Small Small Small Small Small Small Small Small Small   
## [1021] Small Small Small Small Small Small Small Small Small Small   
## [1031] Small Small Small Small Small Small Small Small Small Small   
## [1041] Small Small Small Small Small Small Small Small Small Small   
## [1051] Small Small Small Small Small Small Small Small Small Small   
## [1061] Small Small Small Small Small Small Small Small Small Small   
## [1071] Small Small Small Small Small Small Small Small Small Small   
## [1081] Small Small Small Small Small Small Small Small Small Small   
## [1091] Small Small Small Small Small Small Small Small Small Small   
## [1101] Small Small Small Small Small Small Small Small Small Small   
## [1111] Small Small Small Small Small Small Small Small Small Small   
## [1121] Small Small Small Small Small Small Small Small Small Small   
## [1131] Small Small Small Small Small Small Small Small Small Small   
## [1141] Small Small Small Small Small Small Small Small Small Small   
## [1151] Small Small Small Small Small Small Small Small Small Small   
## [1161] Small Small Small Small Small Small Small Small Small Small   
## [1171] Small Small Small Small Small Small Small Small Small Small   
## [1181] Small Small Small Small Small Small Small Small Small Small   
## [1191] Small Small Small Small Small Small Small Small Small Small   
## [1201] Small Small Small Small Small Small Small Small Small Small   
## [1211] Small Small Small Small Small Small Small Small Small Small   
## [1221] Small Small Small Small Small Small Small Small Small Small   
## [1231] Small Small Small Small Small Small Small Small Small Small   
## [1241] Small Small Small Small Small Small Small Small Small Small   
## [1251] Small Small Small Small Small Small Small Small Small Small   
## [1261] Small Small Small Small Small Small Small Small Small Small   
## [1271] Small Small Small Small Small Small Small Small Small Small   
## [1281] Small Small Small Small Small Small Small Small Small Small   
## [1291] Small Small Small Small Small Small Small Small Small Small   
## [1301] Small Small Small Small Small Small Small Small Small Small   
## [1311] Small Small Small   
## Levels: Small Medium Large

# Conversion to CM From Inches  
  
df2$Tv.Size..Inches.<-df2$Tv.Size..Inches.\*2.54  
df2%>%pull(Tv.Size..Inches.)%>%cut(breaks = c(10,21,31,40),labels = c("Small","Medium","Large"))

## [1] Large Large Large Large Large Large Large Large Large Large   
## [11] Large Large Large Large Medium Large Large Large Large Large   
## [21] Large Large Large Large Large Large Large Large Large <NA>   
## [31] Large Large <NA> Large Large Large Large <NA> Large Large   
## [41] Large Large Large Large <NA> Large <NA> <NA> Large <NA>   
## [51] Large Large Large Large Large Large Large Large <NA> Large   
## [61] <NA> Large Large Large Large Large Large Large Large <NA>   
## [71] Large Large Large <NA> Large Large Large Large Large <NA>   
## [81] Large Medium Large Large Large Large Large Large Large Large   
## [91] Large Large Large Large Large Large Large Large Large Large   
## [101] Large Large Large Large Large Large Large Large Large Large   
## [111] Large Large Large Large Large Large Large Large <NA> Large   
## [121] Large Large Large <NA> Large Large Large Large <NA> <NA>   
## [131] Large Large Large Large Large Large Large <NA> Large Large   
## [141] <NA> Large Large Large Large Large Large Large Large Large   
## [151] Large Large Large <NA> Large Large Large Large Large Large   
## [161] Large <NA> Large Large Large Large Large Large <NA> Large   
## [171] Large <NA> Large Large <NA> Large Large <NA> Large Large   
## [181] Large Large Large Large Large <NA> Large Large Large <NA>   
## [191] Large Large Large <NA> Large <NA> <NA> Large <NA> <NA>   
## [201] Large Large Large Large Large Large Large Large Large Large   
## [211] Large <NA> Large Large Large Large <NA> Large Large Large   
## [221] Large Large Large Large <NA> Large Large Large <NA> Large   
## [231] Large Large Large Large Large Large Large Large <NA> Large   
## [241] Large <NA> <NA> Large Large Large Large <NA> Large Large   
## [251] Large <NA> Large Large <NA> Large Large <NA> <NA> Large   
## [261] <NA> Large Large Large Large Large Large <NA> <NA> Large   
## [271] Large <NA> Large Large Large Large <NA> Large <NA> <NA>   
## [281] <NA> Large Large Large Large Large Large Large Large Large   
## [291] Large <NA> <NA> Large Large Large Large <NA> Large Large   
## [301] Large <NA> <NA> Large Large Large Large Large Large Large   
## [311] Large Large Large Large Medium Large Large Large Large Medium  
## [321] Large <NA> Large Large Large Large Large Large Large Large   
## [331] <NA> Large Large Large Large Large Large Large Large Large   
## [341] Large Large Large Large Large Large Large Large Medium Large   
## [351] Large Large <NA> Large Large Large Large Large Large Large   
## [361] Large Large <NA> Large Large Large <NA> Large Large Large   
## [371] Large Large <NA> <NA> Large Large Medium Large Large Large   
## [381] <NA> Large Large Large Large Large Large Large Large Large   
## [391] <NA> <NA> Large <NA> Large Large Large Large Large Large   
## [401] Large <NA> <NA> Large <NA> Large Large Large <NA> Medium  
## [411] Large Large Large Large Large Large Large Large Large Large   
## [421] Large Large Large <NA> <NA> Large Large Large <NA> Medium  
## [431] Medium Large Large Large Large <NA> Large Large <NA> Large   
## [441] <NA> Large Large Large Large Large Large Large Large Large   
## [451] Large Large Large Large Large Medium <NA> Large <NA> <NA>   
## [461] Large Medium Large Large Large Large Large Large <NA> Large   
## [471] Large Large Large Large Large Large Large Large Large Large   
## [481] Large Large Large Large Large Large Large Large Large Large   
## [491] Large Large Large Large <NA> Large Large Large Large Large   
## [501] Medium Large Large Large Medium Large Large Large Large Large   
## [511] Large Large Large Large Large Medium Large Large Large <NA>   
## [521] Large Large Large Large <NA> <NA> Large Large Large Large   
## [531] <NA> Large Large Large Large Large Large Large <NA> Large   
## [541] Large Large Large Large Large Large Large Large Large Large   
## [551] Large Large <NA> <NA> <NA> Large Large <NA> Large Large   
## [561] Medium Large Large <NA> Large Large Large Large Large Large   
## [571] <NA> Large Large Large Large Large Large <NA> <NA> Large   
## [581] Large Large Large Large Medium <NA> <NA> Large Large <NA>   
## [591] Large Large <NA> Large <NA> Large Large Large Large Large   
## [601] Large Large Large <NA> Large Large Large Large Large Large   
## [611] Large Large Large Large Large <NA> Large Large Large Medium  
## [621] Large Large Large Large Large Large Large Large Large Large   
## [631] Large Large Large Large Large Large Large Large Large Large   
## [641] Large <NA> Large Large Large Large Large Large Large Large   
## [651] <NA> Large Large Large Large Large Large Large Large <NA>   
## [661] Large Large Large Large <NA> Large Large Large Large Large   
## [671] Large Large Large <NA> Large Large Large Medium Large Large   
## [681] Large Large Large <NA> Large Large Large Large <NA> Large   
## [691] Large Large <NA> Large Large Medium Large Large Large Large   
## [701] Large Large Large Large Large Large Large Large Large <NA>   
## [711] Large Large Large Large Large Large Large Large Large <NA>   
## [721] Large Large Large <NA> <NA> Large Large Large Large Large   
## [731] <NA> Large Large Large Large Large Large Large Large Large   
## [741] Large Large Large Large Large Large Large Large Large <NA>   
## [751] Medium Large Large Large Large Large Large Large Large Large   
## [761] Large Large Large Large Large Large <NA> Large Large Large   
## [771] Large Large Large Large Large Large Large Large Large Large   
## [781] <NA> <NA> Large Large Large Large Large Large <NA> Large   
## [791] Large Large Large Large Medium Large Large Large Large Large   
## [801] Large Large Large Large Medium Large Large Large Large Large   
## [811] <NA> <NA> Large Large Large Large Large Medium <NA> Large   
## [821] Large Large Large Large Large Large Large Large Medium Large   
## [831] <NA> Large Large Large Large Large <NA> Large Large Large   
## [841] <NA> <NA> <NA> Large Large Large Medium Large Large Large   
## [851] Large Large <NA> Large Large Large Large Large Large Large   
## [861] Large Large Large Large Large Large Large Large Large Large   
## [871] Large Large Large Large Large Large Large Large Large Large   
## [881] Large Large Large Large Large Large Large <NA> Medium <NA>   
## [891] Large Large Large Large <NA> Large Large Large Large Large   
## [901] Large <NA> Large Large Large Large Large Large Large Large   
## [911] Large Large Large Large Large Large Large <NA> Large Large   
## [921] Large Large Large Large Large Large Large Large Large Large   
## [931] Large Large Large Large Large Large Large Large <NA> <NA>   
## [941] <NA> <NA> Large Large Medium Large Large Large <NA> Large   
## [951] Large Large Large Large Large <NA> Large Medium Large Medium  
## [961] Large Large Large Large Large Large Large Large <NA> Large   
## [971] Large Large <NA> Large <NA> Large Large Large <NA> <NA>   
## [981] Large Large Large Large Large Large Large Large Large Large   
## [991] Large Large Large Large Large Large Large Large Large Large   
## [1001] Large Large Large Large Large Large Large Large Large Large   
## [1011] Large Large Large Large Large Large Large <NA> Large Large   
## [1021] Large Large Large Large Large Large Large Large Large <NA>   
## [1031] Large Large <NA> Large Large Large Large Large Large Large   
## [1041] Large Large Large Large Large Large Large <NA> <NA> Medium  
## [1051] Large Large Large Large Large Large Large Large Large Large   
## [1061] <NA> <NA> Large Large Large Large <NA> Large Large Medium  
## [1071] Large Large Large Large Large <NA> Large <NA> Large Large   
## [1081] Large <NA> Medium Large Large Large Large Large Large Large   
## [1091] Large Large <NA> Large Large Large Large <NA> Large <NA>   
## [1101] Large Large Large <NA> Large Large <NA> Large Large Large   
## [1111] Large Large Large Large Medium Large <NA> Large Large <NA>   
## [1121] Large Large Large Large Large Large Medium Large Large <NA>   
## [1131] Large Large Large Large Large Large <NA> Large <NA> Large   
## [1141] Large Large Large Large Large Large Large Large Large Large   
## [1151] Large Large Large Large Large Large <NA> Large Large Large   
## [1161] Large Large Large Large Large Large Large Large Large Large   
## [1171] Large Large Large Large Large Large Large Large Large Large   
## [1181] Large Large Large Large Large Large Large Large Large <NA>   
## [1191] Large Large Large Medium Large Large Large <NA> Large <NA>   
## [1201] Large Large Large Large Large Large Large Large <NA> Large   
## [1211] Medium Large <NA> Large <NA> Medium <NA> <NA> Large Large   
## [1221] Large Large Large Large Large Large Large Large Large Large   
## [1231] Large Large Large <NA> Medium Large Large Large Large Large   
## [1241] Large Large Large <NA> Large Large Large Large Large Large   
## [1251] Large Large Large Large Large Large <NA> Large Large Large   
## [1261] Large Large Large Large Large Large Large Large Medium Large   
## [1271] Large Large Large Large Large Large Large Large Large Large   
## [1281] Large Large Medium Large Large Large Large Large Large Large   
## [1291] Large Large Large Large Large Large Medium Large Large Large   
## [1301] Large Large Large Large Large Large Large Large Large Large   
## [1311] <NA> Large <NA>   
## Levels: Small Medium Large

# 9. Create Age column from DOB

df2<-df2%>%mutate(Age=Sys.Date()-DOB)  
df2$Age<-as.integer(df2$Age)  
df2$Age<-df2$Age/365  
df2$Age<-round(df2$Age,0)

head(df2)

## Tv.Size..Inches. OpSys DOB X Age  
## 1 33.782 6 2008-02-22 5 16  
## 2 33.782 6 2020-02-27 8 4  
## 3 39.624 7 2016-12-18 1 7  
## 4 39.116 6 1983-03-03 23 41  
## 5 33.782 6 2012-04-20 30 11  
## 6 39.624 10 2019-03-04 27 5

# 10. Create Age Category:

## Infants (1 month to 1 year)

## Children (1 year through 12 years)

## Teenagers (13 years through 17 years)

## Adults (18 years to 65 years)

## Older adults (65 and older)

df2$Age<-df2%>%pull(Age)%>%cut(breaks =c(-Inf,1,13,18,65,Inf),labels = c("Infants","Children","Teenagers","Adults","Older Adults") )  
head(df2)

## Tv.Size..Inches. OpSys DOB X Age  
## 1 33.782 6 2008-02-22 5 Teenagers  
## 2 33.782 6 2020-02-27 8 Children  
## 3 39.624 7 2016-12-18 1 Children  
## 4 39.116 6 1983-03-03 23 Adults  
## 5 33.782 6 2012-04-20 30 Children  
## 6 39.624 10 2019-03-04 27 Children

df2%>%group\_by(Age)%>%summarise(n=n())

## # A tibble: 5 × 2  
## Age n  
## <fct> <int>  
## 1 Infants 23  
## 2 Children 254  
## 3 Teenagers 95  
## 4 Adults 939  
## 5 Older Adults 2

# 11.Filter records outside this DOB range 1960 to 2023.

subset(df2, DOB < as.Date("1960-01-01") & DOB > as.Date("2023-12-31"))

## [1] Tv.Size..Inches. OpSys DOB X   
## [5] Age   
## <0 rows> (or 0-length row.names)

nrow(df2)

## [1] 1313

cat("So Only 2 Records are Outside the DOB Range")

## So Only 2 Records are Outside the DOB Range

# 12.Remove Duplicate Records

head(distinct(df2))

## Tv.Size..Inches. OpSys DOB X Age  
## 1 33.782 6 2008-02-22 5 Teenagers  
## 2 33.782 6 2020-02-27 8 Children  
## 3 39.624 7 2016-12-18 1 Children  
## 4 39.116 6 1983-03-03 23 Adults  
## 5 33.782 6 2012-04-20 30 Children  
## 6 39.624 10 2019-03-04 27 Children

count(distinct(df2))

## n  
## 1 1311

# 13.Create a data frame by taking 20 samples from each age category

sampled\_data <- df2 %>%  
 group\_by(Age) %>%  
 sample\_n(size = 20, replace = TRUE) %>%  
 ungroup()  
sampled\_data

## # A tibble: 100 × 5  
## Tv.Size..Inches. OpSys DOB X Age   
## <dbl> <dbl> <date> <int> <fct>   
## 1 39.6 10 2022-03-12 11 Infants  
## 2 43.9 10 2022-10-30 16 Infants  
## 3 39.6 10 2022-10-30 26 Infants  
## 4 61.0 13 2022-07-19 30 Infants  
## 5 43.9 10 2022-10-30 16 Infants  
## 6 35.6 10 2022-06-13 28 Infants  
## 7 39.6 10 2022-03-12 11 Infants  
## 8 39.6 10 2022-12-13 5 Infants  
## 9 39.6 10 2022-12-14 5 Infants  
## 10 33.8 10 2022-05-23 20 Infants  
## # ℹ 90 more rows