Linear-Regresion-in-R-of-Asia-Population-2020.R

AJAS AJ7

2021-09-22

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.1.1

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.1.1

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(choroplethr)

## Warning: package 'choroplethr' was built under R version 4.1.1

## Loading required package: acs

## Warning: package 'acs' was built under R version 4.1.1

## Loading required package: stringr

## Warning: package 'stringr' was built under R version 4.1.1

## Loading required package: XML

## Warning: package 'XML' was built under R version 4.1.1

##   
## Attaching package: 'acs'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:base':  
##   
## apply

library(choroplethrMaps)

## Warning: package 'choroplethrMaps' was built under R version 4.1.1

library(openintro)

## Warning: package 'openintro' was built under R version 4.1.1

## Loading required package: airports

## Warning: package 'airports' was built under R version 4.1.1

## Loading required package: cherryblossom

## Warning: package 'cherryblossom' was built under R version 4.1.1

## Loading required package: usdata

## Warning: package 'usdata' was built under R version 4.1.1

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.1

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.4 v purrr 0.3.4  
## v tidyr 1.1.3 v forcats 0.5.1  
## v readr 2.0.1

## Warning: package 'tibble' was built under R version 4.1.1

## Warning: package 'tidyr' was built under R version 4.1.1

## Warning: package 'readr' was built under R version 4.1.1

## Warning: package 'purrr' was built under R version 4.1.1

## Warning: package 'forcats' was built under R version 4.1.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x acs::combine() masks dplyr::combine()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(scales)

## Warning: package 'scales' was built under R version 4.1.1

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(corrgram)

## Warning: package 'corrgram' was built under R version 4.1.1

print(getwd)

## function ()   
## .Internal(getwd())  
## <bytecode: 0x0000000012ce0b48>  
## <environment: namespace:base>

# read the dataset named vaccination data  
d<-read.csv("C:/Users/Sumi/AsiaPopulation2020.csv")  
print(d)

## Country Population YearlyChange NetChange Density LandArea  
## 1 Afghanistan 38928346 2.33 886592 60 652860  
## 2 Armenia 2963243 0.19 5512 104 28470  
## 3 Azerbaijan 10139177 0.91 91459 123 82658  
## 4 Bahrain 1701575 3.68 60403 2239 760  
## 5 Bangladesh 164689383 1.01 1643222 1265 130170  
## 6 Bhutan 771608 1.12 8516 20 38117  
## 7 Brunei 437479 0.97 4194 83 5270  
## 8 Cambodia 16718965 1.41 232423 95 176520  
## 9 China 1439323776 0.39 5540090 153 9388211  
## 10 Cyprus 1207359 0.73 8784 131 9240  
## 11 Georgia 3989167 -0.19 -7598 57 69490  
## 12 Hong Kong 7496981 0.82 60827 7140 1050  
## 13 India 1380004385 0.99 13586631 464 2973190  
## 14 Indonesia 273523615 1.07 2898047 151 1811570  
## 15 Iran 83992949 1.30 1079043 52 1628550  
## 16 Iraq 40222493 2.32 912710 93 434320  
## 17 Israel 8655535 1.60 136158 400 21640  
## 18 Japan 126476461 -0.30 -383840 347 364555  
## 19 Jordan 10203134 1.00 101440 115 88780  
## 20 Kazakhstan 18776707 1.21 225280 7 2699700  
## 21 Kuwait 4270571 1.51 63488 240 17820  
## 22 Kyrgyzstan 6524195 1.69 108345 34 191800  
## 23 Laos 7275560 1.48 106105 32 230800  
## 24 Lebanon 6825445 -0.44 -30268 667 10230  
## 25 Macao 649335 1.39 8890 21645 30  
## 26 Malaysia 32365999 1.30 416222 99 328550  
## 27 Maldives 540544 1.81 9591 1802 300  
## 28 Mongolia 3278290 1.65 53123 2 1553560  
## 29 Myanmar 54409800 0.67 364380 83 653290  
## 30 Nepal 29136808 1.85 528098 203 143350  
## 31 North Korea 25778816 0.44 112655 214 120410  
## 32 Oman 5106626 2.65 131640 16 309500  
## 33 Pakistan 220892340 2.00 4327022 287 770880  
## 34 Philippines 109581078 1.35 1464463 368 298170  
## 35 Qatar 2881053 1.73 48986 248 11610  
## 36 Saudi Arabia 34813871 1.59 545343 16 2149690  
## 37 Singapore 5850342 0.79 46005 8358 700  
## 38 South Korea 51269185 0.09 43877 527 97230  
## 39 Sri Lanka 21413249 0.42 89516 341 62710  
## 40 State of Palestine 5101414 2.41 119994 847 6020  
## 41 Syria 17500658 2.52 430523 95 183630  
## 42 Taiwan 23816775 0.18 42899 673 35410  
## 43 Tajikistan 9537645 2.32 216627 68 139960  
## 44 Thailand 69799978 0.25 174396 137 510890  
## 45 Timor-Leste 1318445 1.96 25326 89 14870  
## 46 Turkey 84339067 1.09 909452 110 769630  
## 47 Turkmenistan 6031200 1.50 89111 13 469930  
## 48 United Arab Emirates 9890402 1.23 119873 118 83600  
## 49 Uzbekistan 33469203 1.48 487487 79 425400  
## 50 Vietnam 97338579 0.91 876473 314 310070  
## 51 Yemen 29825964 2.28 664042 56 527970  
## Migrants FertRate MedAge UrbanPop WorldShare  
## 1 -62920 4.6 18 25 0.50  
## 2 -4998 1.8 35 63 0.04  
## 3 1200 2.1 32 56 0.13  
## 4 47800 2.0 32 89 0.02  
## 5 -369501 2.1 28 39 2.11  
## 6 320 2.0 28 46 0.01  
## 7 0 1.8 32 80 0.01  
## 8 -30000 2.5 26 24 0.21  
## 9 -348399 1.7 38 61 18.47  
## 10 5000 1.3 37 67 0.02  
## 11 -10000 2.1 38 58 0.05  
## 12 29308 1.3 45 NA 0.10  
## 13 -532687 2.2 28 35 17.70  
## 14 -98955 2.3 30 56 3.51  
## 15 -55000 2.2 32 76 1.08  
## 16 7834 3.7 21 73 0.52  
## 17 10000 3.0 30 93 0.11  
## 18 71560 1.4 48 92 1.62  
## 19 10220 2.8 24 91 0.13  
## 20 -18000 2.8 31 58 0.24  
## 21 39520 2.1 37 NA 0.05  
## 22 -4000 3.0 26 36 0.08  
## 23 -14704 2.7 24 36 0.09  
## 24 -30012 2.1 30 78 0.09  
## 25 5000 1.2 39 NA 0.01  
## 26 50000 2.0 30 78 0.42  
## 27 11370 1.9 30 35 0.01  
## 28 -852 2.9 28 67 0.04  
## 29 -163313 2.2 29 31 0.70  
## 30 41710 1.9 25 21 0.37  
## 31 -5403 1.9 35 63 0.33  
## 32 87400 2.9 31 87 0.07  
## 33 -233379 3.6 23 35 2.83  
## 34 -67152 2.6 26 47 1.41  
## 35 40000 1.9 32 96 0.04  
## 36 134979 2.3 32 84 0.45  
## 37 27028 1.2 42 NA 0.08  
## 38 11731 1.1 44 82 0.66  
## 39 -97986 2.2 34 18 0.27  
## 40 -10563 3.7 21 80 0.07  
## 41 -427391 2.8 26 60 0.22  
## 42 30001 1.2 42 79 0.31  
## 43 -20000 3.6 22 27 0.12  
## 44 19444 1.5 40 51 0.90  
## 45 -5385 4.1 21 33 0.02  
## 46 283922 2.1 32 76 1.08  
## 47 -5000 2.8 27 53 0.08  
## 48 40000 1.4 33 86 0.13  
## 49 -8863 2.4 28 50 0.43  
## 50 -80000 2.1 32 38 1.25  
## 51 -30000 3.8 20 38 0.38

# print head and tail rows  
print(head(d))

## Country Population YearlyChange NetChange Density LandArea Migrants  
## 1 Afghanistan 38928346 2.33 886592 60 652860 -62920  
## 2 Armenia 2963243 0.19 5512 104 28470 -4998  
## 3 Azerbaijan 10139177 0.91 91459 123 82658 1200  
## 4 Bahrain 1701575 3.68 60403 2239 760 47800  
## 5 Bangladesh 164689383 1.01 1643222 1265 130170 -369501  
## 6 Bhutan 771608 1.12 8516 20 38117 320  
## FertRate MedAge UrbanPop WorldShare  
## 1 4.6 18 25 0.50  
## 2 1.8 35 63 0.04  
## 3 2.1 32 56 0.13  
## 4 2.0 32 89 0.02  
## 5 2.1 28 39 2.11  
## 6 2.0 28 46 0.01

print(tail(d))

## Country Population YearlyChange NetChange Density LandArea  
## 46 Turkey 84339067 1.09 909452 110 769630  
## 47 Turkmenistan 6031200 1.50 89111 13 469930  
## 48 United Arab Emirates 9890402 1.23 119873 118 83600  
## 49 Uzbekistan 33469203 1.48 487487 79 425400  
## 50 Vietnam 97338579 0.91 876473 314 310070  
## 51 Yemen 29825964 2.28 664042 56 527970  
## Migrants FertRate MedAge UrbanPop WorldShare  
## 46 283922 2.1 32 76 1.08  
## 47 -5000 2.8 27 53 0.08  
## 48 40000 1.4 33 86 0.13  
## 49 -8863 2.4 28 50 0.43  
## 50 -80000 2.1 32 38 1.25  
## 51 -30000 3.8 20 38 0.38

print(is.na(d))

## Country Population YearlyChange NetChange Density LandArea Migrants  
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [7,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [12,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [14,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [15,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [16,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [17,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [18,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [19,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [20,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [21,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [22,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [23,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [24,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [26,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [27,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [28,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [29,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [30,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [31,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [32,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [33,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [34,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [35,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [36,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [38,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [39,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [40,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [41,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [42,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [43,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [44,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [45,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [46,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [47,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [48,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [50,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [51,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## FertRate MedAge UrbanPop WorldShare  
## [1,] FALSE FALSE FALSE FALSE  
## [2,] FALSE FALSE FALSE FALSE  
## [3,] FALSE FALSE FALSE FALSE  
## [4,] FALSE FALSE FALSE FALSE  
## [5,] FALSE FALSE FALSE FALSE  
## [6,] FALSE FALSE FALSE FALSE  
## [7,] FALSE FALSE FALSE FALSE  
## [8,] FALSE FALSE FALSE FALSE  
## [9,] FALSE FALSE FALSE FALSE  
## [10,] FALSE FALSE FALSE FALSE  
## [11,] FALSE FALSE FALSE FALSE  
## [12,] FALSE FALSE TRUE FALSE  
## [13,] FALSE FALSE FALSE FALSE  
## [14,] FALSE FALSE FALSE FALSE  
## [15,] FALSE FALSE FALSE FALSE  
## [16,] FALSE FALSE FALSE FALSE  
## [17,] FALSE FALSE FALSE FALSE  
## [18,] FALSE FALSE FALSE FALSE  
## [19,] FALSE FALSE FALSE FALSE  
## [20,] FALSE FALSE FALSE FALSE  
## [21,] FALSE FALSE TRUE FALSE  
## [22,] FALSE FALSE FALSE FALSE  
## [23,] FALSE FALSE FALSE FALSE  
## [24,] FALSE FALSE FALSE FALSE  
## [25,] FALSE FALSE TRUE FALSE  
## [26,] FALSE FALSE FALSE FALSE  
## [27,] FALSE FALSE FALSE FALSE  
## [28,] FALSE FALSE FALSE FALSE  
## [29,] FALSE FALSE FALSE FALSE  
## [30,] FALSE FALSE FALSE FALSE  
## [31,] FALSE FALSE FALSE FALSE  
## [32,] FALSE FALSE FALSE FALSE  
## [33,] FALSE FALSE FALSE FALSE  
## [34,] FALSE FALSE FALSE FALSE  
## [35,] FALSE FALSE FALSE FALSE  
## [36,] FALSE FALSE FALSE FALSE  
## [37,] FALSE FALSE TRUE FALSE  
## [38,] FALSE FALSE FALSE FALSE  
## [39,] FALSE FALSE FALSE FALSE  
## [40,] FALSE FALSE FALSE FALSE  
## [41,] FALSE FALSE FALSE FALSE  
## [42,] FALSE FALSE FALSE FALSE  
## [43,] FALSE FALSE FALSE FALSE  
## [44,] FALSE FALSE FALSE FALSE  
## [45,] FALSE FALSE FALSE FALSE  
## [46,] FALSE FALSE FALSE FALSE  
## [47,] FALSE FALSE FALSE FALSE  
## [48,] FALSE FALSE FALSE FALSE  
## [49,] FALSE FALSE FALSE FALSE  
## [50,] FALSE FALSE FALSE FALSE  
## [51,] FALSE FALSE FALSE FALSE

# print column names  
print(colnames(d))

## [1] "Country" "Population" "YearlyChange" "NetChange" "Density"   
## [6] "LandArea" "Migrants" "FertRate" "MedAge" "UrbanPop"   
## [11] "WorldShare"

print((names(d)))

## [1] "Country" "Population" "YearlyChange" "NetChange" "Density"   
## [6] "LandArea" "Migrants" "FertRate" "MedAge" "UrbanPop"   
## [11] "WorldShare"

# statistical values  
print(nrow(d))

## [1] 51

print(ncol(d))

## [1] 11

print(summary(d))

## Country Population YearlyChange NetChange   
## Length:51 Min. :4.375e+05 Min. :-0.440 Min. : -383840   
## Class :character 1st Qu.:5.104e+06 1st Qu.: 0.805 1st Qu.: 47496   
## Mode :character Median :1.672e+07 Median : 1.300 Median : 119873   
## Mean :9.100e+07 Mean : 1.268 Mean : 778109   
## 3rd Qu.:4.575e+07 3rd Qu.: 1.710 3rd Qu.: 536720   
## Max. :1.439e+09 Max. : 3.680 Max. :13586631   
##   
## Density LandArea Migrants FertRate   
## Min. : 2.0 Min. : 30 Min. :-532687 Min. :1.100   
## 1st Qu.: 73.5 1st Qu.: 25055 1st Qu.: -30006 1st Qu.:1.900   
## Median : 123.0 Median : 143350 Median : -4000 Median :2.100   
## Mean : 997.6 Mean : 608493 Mean : -33904 Mean :2.331   
## 3rd Qu.: 357.5 3rd Qu.: 490410 3rd Qu.: 23236 3rd Qu.:2.800   
## Max. :21645.0 Max. :9388211 Max. : 283922 Max. :4.600   
##   
## MedAge UrbanPop WorldShare   
## Min. :18.00 Min. :18.00 Min. : 0.010   
## 1st Qu.:26.00 1st Qu.:37.00 1st Qu.: 0.070   
## Median :30.00 Median :58.00 Median : 0.210   
## Mean :30.86 Mean :58.45 Mean : 1.168   
## 3rd Qu.:34.50 3rd Qu.:78.50 3rd Qu.: 0.590   
## Max. :48.00 Max. :96.00 Max. :18.470   
## NA's :4

print(str(d))

## 'data.frame': 51 obs. of 11 variables:  
## $ Country : chr "Afghanistan" "Armenia" "Azerbaijan" "Bahrain" ...  
## $ Population : int 38928346 2963243 10139177 1701575 164689383 771608 437479 16718965 1439323776 1207359 ...  
## $ YearlyChange: num 2.33 0.19 0.91 3.68 1.01 1.12 0.97 1.41 0.39 0.73 ...  
## $ NetChange : int 886592 5512 91459 60403 1643222 8516 4194 232423 5540090 8784 ...  
## $ Density : int 60 104 123 2239 1265 20 83 95 153 131 ...  
## $ LandArea : int 652860 28470 82658 760 130170 38117 5270 176520 9388211 9240 ...  
## $ Migrants : int -62920 -4998 1200 47800 -369501 320 0 -30000 -348399 5000 ...  
## $ FertRate : num 4.6 1.8 2.1 2 2.1 2 1.8 2.5 1.7 1.3 ...  
## $ MedAge : int 18 35 32 32 28 28 32 26 38 37 ...  
## $ UrbanPop : int 25 63 56 89 39 46 80 24 61 67 ...  
## $ WorldShare : num 0.5 0.04 0.13 0.02 2.11 ...  
## NULL

print(mean(d$income))

## Warning in mean.default(d$income): argument is not numeric or logical: returning  
## NA

## [1] NA

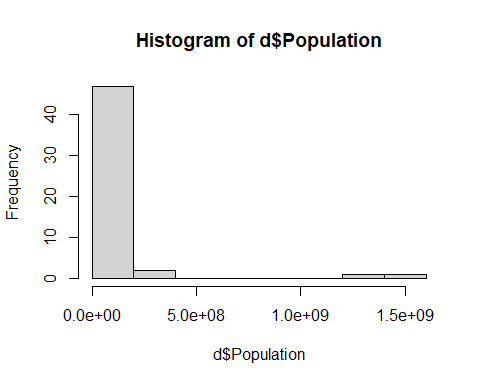
print(mean(is.na(d)))

## [1] 0.007130125

print(unique(d))

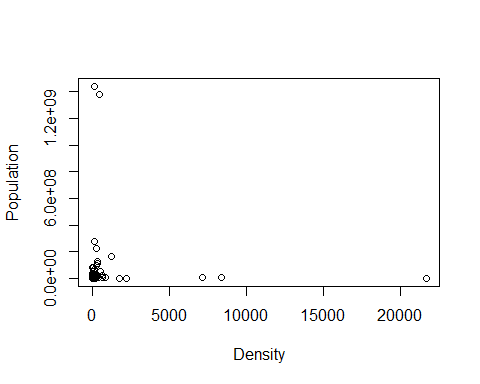
## Country Population YearlyChange NetChange Density LandArea  
## 1 Afghanistan 38928346 2.33 886592 60 652860  
## 2 Armenia 2963243 0.19 5512 104 28470  
## 3 Azerbaijan 10139177 0.91 91459 123 82658  
## 4 Bahrain 1701575 3.68 60403 2239 760  
## 5 Bangladesh 164689383 1.01 1643222 1265 130170  
## 6 Bhutan 771608 1.12 8516 20 38117  
## 7 Brunei 437479 0.97 4194 83 5270  
## 8 Cambodia 16718965 1.41 232423 95 176520  
## 9 China 1439323776 0.39 5540090 153 9388211  
## 10 Cyprus 1207359 0.73 8784 131 9240  
## 11 Georgia 3989167 -0.19 -7598 57 69490  
## 12 Hong Kong 7496981 0.82 60827 7140 1050  
## 13 India 1380004385 0.99 13586631 464 2973190  
## 14 Indonesia 273523615 1.07 2898047 151 1811570  
## 15 Iran 83992949 1.30 1079043 52 1628550  
## 16 Iraq 40222493 2.32 912710 93 434320  
## 17 Israel 8655535 1.60 136158 400 21640  
## 18 Japan 126476461 -0.30 -383840 347 364555  
## 19 Jordan 10203134 1.00 101440 115 88780  
## 20 Kazakhstan 18776707 1.21 225280 7 2699700  
## 21 Kuwait 4270571 1.51 63488 240 17820  
## 22 Kyrgyzstan 6524195 1.69 108345 34 191800  
## 23 Laos 7275560 1.48 106105 32 230800  
## 24 Lebanon 6825445 -0.44 -30268 667 10230  
## 25 Macao 649335 1.39 8890 21645 30  
## 26 Malaysia 32365999 1.30 416222 99 328550  
## 27 Maldives 540544 1.81 9591 1802 300  
## 28 Mongolia 3278290 1.65 53123 2 1553560  
## 29 Myanmar 54409800 0.67 364380 83 653290  
## 30 Nepal 29136808 1.85 528098 203 143350  
## 31 North Korea 25778816 0.44 112655 214 120410  
## 32 Oman 5106626 2.65 131640 16 309500  
## 33 Pakistan 220892340 2.00 4327022 287 770880  
## 34 Philippines 109581078 1.35 1464463 368 298170  
## 35 Qatar 2881053 1.73 48986 248 11610  
## 36 Saudi Arabia 34813871 1.59 545343 16 2149690  
## 37 Singapore 5850342 0.79 46005 8358 700  
## 38 South Korea 51269185 0.09 43877 527 97230  
## 39 Sri Lanka 21413249 0.42 89516 341 62710  
## 40 State of Palestine 5101414 2.41 119994 847 6020  
## 41 Syria 17500658 2.52 430523 95 183630  
## 42 Taiwan 23816775 0.18 42899 673 35410  
## 43 Tajikistan 9537645 2.32 216627 68 139960  
## 44 Thailand 69799978 0.25 174396 137 510890  
## 45 Timor-Leste 1318445 1.96 25326 89 14870  
## 46 Turkey 84339067 1.09 909452 110 769630  
## 47 Turkmenistan 6031200 1.50 89111 13 469930  
## 48 United Arab Emirates 9890402 1.23 119873 118 83600  
## 49 Uzbekistan 33469203 1.48 487487 79 425400  
## 50 Vietnam 97338579 0.91 876473 314 310070  
## 51 Yemen 29825964 2.28 664042 56 527970  
## Migrants FertRate MedAge UrbanPop WorldShare  
## 1 -62920 4.6 18 25 0.50  
## 2 -4998 1.8 35 63 0.04  
## 3 1200 2.1 32 56 0.13  
## 4 47800 2.0 32 89 0.02  
## 5 -369501 2.1 28 39 2.11  
## 6 320 2.0 28 46 0.01  
## 7 0 1.8 32 80 0.01  
## 8 -30000 2.5 26 24 0.21  
## 9 -348399 1.7 38 61 18.47  
## 10 5000 1.3 37 67 0.02  
## 11 -10000 2.1 38 58 0.05  
## 12 29308 1.3 45 NA 0.10  
## 13 -532687 2.2 28 35 17.70  
## 14 -98955 2.3 30 56 3.51  
## 15 -55000 2.2 32 76 1.08  
## 16 7834 3.7 21 73 0.52  
## 17 10000 3.0 30 93 0.11  
## 18 71560 1.4 48 92 1.62  
## 19 10220 2.8 24 91 0.13  
## 20 -18000 2.8 31 58 0.24  
## 21 39520 2.1 37 NA 0.05  
## 22 -4000 3.0 26 36 0.08  
## 23 -14704 2.7 24 36 0.09  
## 24 -30012 2.1 30 78 0.09  
## 25 5000 1.2 39 NA 0.01  
## 26 50000 2.0 30 78 0.42  
## 27 11370 1.9 30 35 0.01  
## 28 -852 2.9 28 67 0.04  
## 29 -163313 2.2 29 31 0.70  
## 30 41710 1.9 25 21 0.37  
## 31 -5403 1.9 35 63 0.33  
## 32 87400 2.9 31 87 0.07  
## 33 -233379 3.6 23 35 2.83  
## 34 -67152 2.6 26 47 1.41  
## 35 40000 1.9 32 96 0.04  
## 36 134979 2.3 32 84 0.45  
## 37 27028 1.2 42 NA 0.08  
## 38 11731 1.1 44 82 0.66  
## 39 -97986 2.2 34 18 0.27  
## 40 -10563 3.7 21 80 0.07  
## 41 -427391 2.8 26 60 0.22  
## 42 30001 1.2 42 79 0.31  
## 43 -20000 3.6 22 27 0.12  
## 44 19444 1.5 40 51 0.90  
## 45 -5385 4.1 21 33 0.02  
## 46 283922 2.1 32 76 1.08  
## 47 -5000 2.8 27 53 0.08  
## 48 40000 1.4 33 86 0.13  
## 49 -8863 2.4 28 50 0.43  
## 50 -80000 2.1 32 38 1.25  
## 51 -30000 3.8 20 38 0.38

#Histogram of Population  
print(hist(d$Population))



## $breaks  
## [1] 0.0e+00 2.0e+08 4.0e+08 6.0e+08 8.0e+08 1.0e+09 1.2e+09 1.4e+09 1.6e+09  
##   
## $counts  
## [1] 47 2 0 0 0 0 1 1  
##   
## $density  
## [1] 4.607843e-09 1.960784e-10 0.000000e+00 0.000000e+00 0.000000e+00  
## [6] 0.000000e+00 9.803922e-11 9.803922e-11  
##   
## $mids  
## [1] 1.0e+08 3.0e+08 5.0e+08 7.0e+08 9.0e+08 1.1e+09 1.3e+09 1.5e+09  
##   
## $xname  
## [1] "d$Population"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

print(plot(Population~Density,data=d))



## NULL

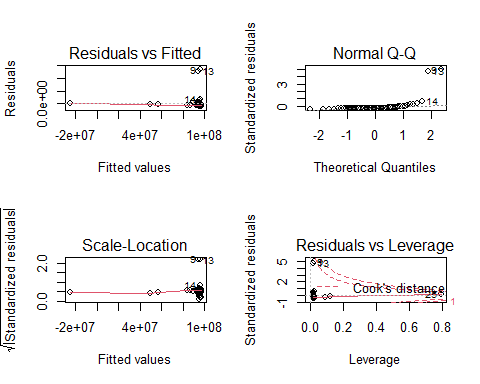
lr=lm(Population~Density,data=d)  
print(lr)

##   
## Call:  
## lm(formula = Population ~ Density, data = d)  
##   
## Coefficients:  
## (Intercept) Density   
## 96595419 -5608

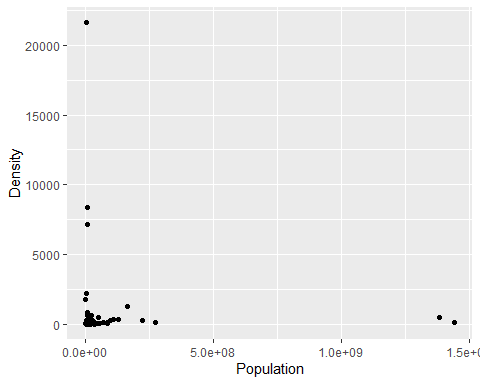
summary(lr)

##   
## Call:  
## lm(formula = Population ~ Density, data = d)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -95711660 -87942418 -73269999 -42045627 1343586311   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 96595419 40539528 2.383 0.0211 \*  
## Density -5608 11795 -0.475 0.6366   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.77e+08 on 49 degrees of freedom  
## Multiple R-squared: 0.004591, Adjusted R-squared: -0.01572   
## F-statistic: 0.226 on 1 and 49 DF, p-value: 0.6366

par(mfrow=c(2,2))  
  
plot(lr)



i<-ggplot(d,aes(x=Population,y=Density))+geom\_point()  
print(i)



i<-i+geom\_smooth(method="lm",col="blue")  
print(i)

## `geom\_smooth()` using formula 'y ~ x'

