Homework 1: Data Wrangling Walkthrough

Team Syria

Environment Prep

# load packages  
library(tidyverse)  
library(readxl)  
library(pander)  
library(scales)

# set options so tables render nicely  
pander::panderOptions('table.split.table', Inf)

# Data

Data were retrieved from:

* [World Bank Data](https://data.worldbank.org)
  + Improved water source (% of population with access)
  + Life expectancy at birth, total (years)
  + Urban population growth (annual %)
* [UNDP Human Development Reports](http://hdr.undp.org/en/data)
  + Adult Literacy Rate (% Ages 15 and older)
  + Population with at least some secondary education (% ages 25 and older)
  + Mean years of schooling (years)
  + Inequality in education (%)
  + GNI per capita (2011 PPP$)

## Read and Inspect Raw Data

We read the excel file containing our raw data and store it in a data\_frame variable named combo.

combo <- read\_excel("~/Development-Economics/data/combo.xlsx")  
  
pander(combo)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Country | Improved water source (% of population with access) | Life expectancy at birth, total (years) | Urban population % | Adult Literacy Rate (% Ages 15 and older) | Population with at least some secondary education (% ages 25 and older) | Mean years of schooling (years) | GNI per capita (2011 PPP$) | Press Freedom Score (0 worst - 100 best) |
| Cuba | 94.6 | 79.39 | 77 | 99.7 | 84.8 | 11.8 | 7280 | 90 |
| Ghana | 87.6 | 62.11 | 55 | 76.6 | 59.8 | 6.9 | 3472 | 28 |
| India | 94.1 | 68.05 | 33 | 72.1 | 48.7 | 6.1 | 5329 | 39 |
| Mexico | 96.1 | 76.7 | 80 | 94.4 | 57.4 | 8.4 | 16154 | 61 |
| Russia | 96.9 | 70.74 | 74 | 99.7 | 94.6 | 12 | 24067 | 81 |
| South Africa | 92.8 | 60.95 | 65 | 94.3 | 74.9 | 10.3 | 12113 | 33 |
| Syria | 90.1 | 70.16 | 58 | 86.4 | 38.9 | 5.6 | 2905 | 89 |
| Thailand | 97.8 | 74.86 | 52 | 96.7 | 43.3 | 7.9 | 14169 | 64 |

## Normalize

### Normalize Formula

In order to normalize our values to a 0 to 1 scale, we use the following formula:

Divide (cardinal number – minimum) by (maximum – minimum)

We create a function called normalize to achieve this result.

normalize <- function(x){  
 (x - min(x)) / (max(x) - min(x))  
}

## Mutate

We take our normalize() function, apply it to our combo data, and store the results in a new data\_frame called combo\_norm. Since Country is a non-numeric column, mutate\_if allows us to only apply our functions to columns where is.numeric() returns TRUE. We also rename our variables to simplify our table and drop any scale references.

combo\_norm <- combo %>%  
 mutate\_if(is.numeric, funs(normalize)) %>%  
 rename(`Press Freedom` = `Press Freedom Score (0 worst - 100 best)`,  
 `Water Access` = `Improved water source (% of population with access)`,  
 `Life Expectancy` = `Life expectancy at birth, total (years)`,  
 `Urban Population` = `Urban population %`,  
 `Adult Literacy` = `Adult Literacy Rate (% Ages 15 and older)`,  
 `GNI per Capita` = `GNI per capita (2011 PPP$)`  
 )  
  
pander(combo\_norm)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Country | Water Access | Life Expectancy | Urban Population | Adult Literacy | Population with at least some secondary education (% ages 25 and older) | Mean years of schooling (years) | GNI per Capita | Press Freedom |
| Cuba | 0.6863 | 1 | 0.9362 | 1 | 0.8241 | 0.9688 | 0.2067 | 1 |
| Ghana | 0 | 0.06287 | 0.4681 | 0.163 | 0.3752 | 0.2031 | 0.02679 | 0 |
| India | 0.6373 | 0.3849 | 0 | 0 | 0.1759 | 0.07812 | 0.1145 | 0.1774 |
| Mexico | 0.8333 | 0.8541 | 1 | 0.808 | 0.3321 | 0.4375 | 0.6261 | 0.5323 |
| Russia | 0.9118 | 0.531 | 0.8723 | 1 | 1 | 1 | 1 | 0.8548 |
| South Africa | 0.5098 | 0 | 0.6809 | 0.8043 | 0.6463 | 0.7344 | 0.4351 | 0.08065 |
| Syria | 0.2451 | 0.4995 | 0.5319 | 0.5181 | 0 | 0 | 0 | 0.9839 |
| Thailand | 1 | 0.7545 | 0.4043 | 0.8913 | 0.07899 | 0.3594 | 0.5323 | 0.5806 |

### Combine Education Variables

We decided to combine our education variables by taking their mean and assigning it to a new variable called Combined Education.

combo\_norm\_comb\_ed <- combo\_norm %>%  
 group\_by(Country) %>%  
 mutate(`Combined Education` = mean(c(`Population with at least some secondary education (% ages 25 and older)`, `Mean years of schooling (years)`, `Adult Literacy`))) %>%  
 select(-`Population with at least some secondary education (% ages 25 and older)`,  
 -`Mean years of schooling (years)`,  
 -`Adult Literacy`)  
  
pander(combo\_norm\_comb\_ed)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Country | Water Access | Life Expectancy | Urban Population | GNI per Capita | Press Freedom | Combined Education |
| Cuba | 0.6863 | 1 | 0.9362 | 0.2067 | 1 | 0.9309 |
| Ghana | 0 | 0.06287 | 0.4681 | 0.02679 | 0 | 0.2471 |
| India | 0.6373 | 0.3849 | 0 | 0.1145 | 0.1774 | 0.08469 |
| Mexico | 0.8333 | 0.8541 | 1 | 0.6261 | 0.5323 | 0.5259 |
| Russia | 0.9118 | 0.531 | 0.8723 | 1 | 0.8548 | 1 |
| South Africa | 0.5098 | 0 | 0.6809 | 0.4351 | 0.08065 | 0.7283 |
| Syria | 0.2451 | 0.4995 | 0.5319 | 0 | 0.9839 | 0.1727 |
| Thailand | 1 | 0.7545 | 0.4043 | 0.5323 | 0.5806 | 0.4432 |

## Weight

We decided on the following weighting scheme for our data: \* Water: 0.30 \* Education: 0.25 \* GNI per Capita: 0.18 \* Urban Population: 0.12 \* Press Freedom: 0.09 \* Life Expectancy: 0.06

We assign our weights to individual variables before using them. To ensure that we're weighting correctly, we also check that our weight sum to 1.

water\_wt <- 0.30  
education\_wt <- 0.25  
GNI\_wt <- 0.18  
urban\_wt <- 0.12  
press\_freedom\_wt <- 0.09  
life\_expect\_wt <- 0.06  
  
weights <- c(water\_wt, education\_wt, GNI\_wt, urban\_wt, press\_freedom\_wt, life\_expect\_wt)  
  
sum(weights) == 1

## [1] TRUE

In order to weight our data, we multiply each column by the appropriate weight variable. At this point, we also round our data to 2 digits following the decimal point.

combo\_weighted <- combo\_norm\_comb\_ed %>%  
 mutate(`Water Access` = `Water Access` \* water\_wt,  
 `Life Expectancy` = `Life Expectancy` \* life\_expect\_wt,  
 `Urban Population` = `Urban Population` \* urban\_wt,  
 `GNI per Capita` = `GNI per Capita` \* GNI\_wt,  
 `Press Freedom` = `Press Freedom` \* press\_freedom\_wt,  
 `Combined Education` = `Combined Education` \* education\_wt) %>%  
 mutate\_if(is.numeric, funs(round(., digits = 2)))  
  
pander(combo\_weighted)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Country | Water Access | Life Expectancy | Urban Population | GNI per Capita | Press Freedom | Combined Education |
| Cuba | 0.21 | 0.06 | 0.11 | 0.04 | 0.09 | 0.23 |
| Ghana | 0 | 0 | 0.06 | 0 | 0 | 0.06 |
| India | 0.19 | 0.02 | 0 | 0.02 | 0.02 | 0.02 |
| Mexico | 0.25 | 0.05 | 0.12 | 0.11 | 0.05 | 0.13 |
| Russia | 0.27 | 0.03 | 0.1 | 0.18 | 0.08 | 0.25 |
| South Africa | 0.15 | 0 | 0.08 | 0.08 | 0.01 | 0.18 |
| Syria | 0.07 | 0.03 | 0.06 | 0 | 0.09 | 0.04 |
| Thailand | 0.3 | 0.05 | 0.05 | 0.1 | 0.05 | 0.11 |

## Score

By summing our normalized and weighted variables, we create our Score data. We also round these data to 2 digits following the decimal point.

Score <- combo\_weighted %>%  
 group\_by(Country) %>%  
 summarise(Score = sum(`Water Access`,  
 `Life Expectancy`,  
 `Urban Population`,  
 `GNI per Capita`,  
 `Press Freedom`,  
 `Combined Education`))  
  
pander(Score)

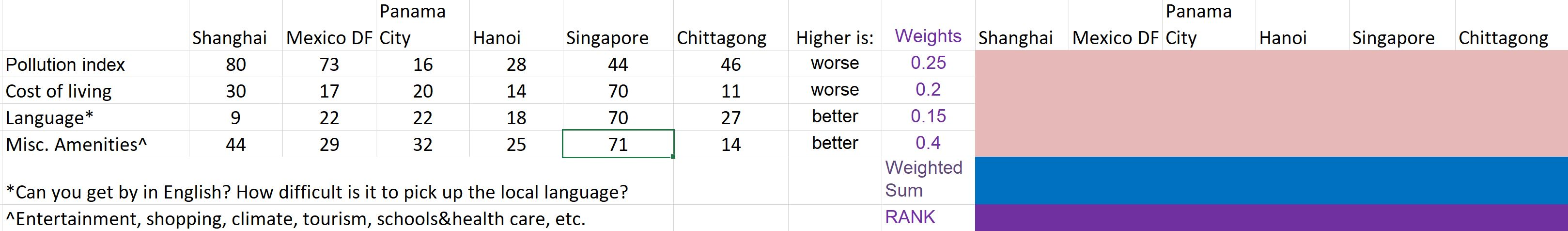
|  |  |
| --- | --- |
| Country | Score |
| Cuba | 0.74 |
| Ghana | 0.12 |
| India | 0.27 |
| Mexico | 0.71 |
| Russia | 0.91 |
| South Africa | 0.5 |
| Syria | 0.29 |
| Thailand | 0.66 |

With our Score data, we can then create our Rank data.

Score\_Rank <- Score %>%  
 mutate(Rank = dense\_rank(desc(Score)))  
  
pander(Score\_Rank)

|  |  |  |
| --- | --- | --- |
| Country | Score | Rank |
| Cuba | 0.74 | 2 |
| Ghana | 0.12 | 8 |
| India | 0.27 | 7 |
| Mexico | 0.71 | 3 |
| Russia | 0.91 | 1 |
| South Africa | 0.5 | 5 |
| Syria | 0.29 | 6 |
| Thailand | 0.66 | 4 |

The example format that we have seen looks like so:



To match this we need to convert our data\_frame to a matrix, transpose the data, and store it in a variable that we'll call Score.

Score\_Rank <- Score\_Rank %>%  
 as.matrix() %>%  
 t()  
  
pander(Score\_Rank)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Country** | Cuba | Ghana | India | Mexico | Russia | South Africa | Syria | Thailand |
| **Score** | 0.74 | 0.12 | 0.27 | 0.71 | 0.91 | 0.50 | 0.29 | 0.66 |
| **Rank** | 2 | 8 | 7 | 3 | 1 | 5 | 6 | 4 |

We're then going to detach the Country row so that we can bind the Score and Rank rows to our summary table.

Score\_Rank <- Score\_Rank[-1, ]  
  
pander(Score\_Rank)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Score** | 0.74 | 0.12 | 0.27 | 0.71 | 0.91 | 0.50 | 0.29 | 0.66 |
| **Rank** | 2 | 8 | 7 | 3 | 1 | 5 | 6 | 4 |

## Data Restructuring

To produce the desired table, we convert our combo\_norm variable to a matrix, which we then transpose and store in a variable called combo\_norm\_matr\_trans.

combo\_norm\_matr\_trans <- combo\_weighted %>%  
 as.matrix() %>%  
 t()  
  
pander(combo\_norm\_matr\_trans)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Country** | Cuba | Ghana | India | Mexico | Russia | South Africa | Syria | Thailand |
| **Water Access** | 0.21 | 0.00 | 0.19 | 0.25 | 0.27 | 0.15 | 0.07 | 0.30 |
| **Life Expectancy** | 0.06 | 0.00 | 0.02 | 0.05 | 0.03 | 0.00 | 0.03 | 0.05 |
| **Urban Population** | 0.11 | 0.06 | 0.00 | 0.12 | 0.10 | 0.08 | 0.06 | 0.05 |
| **GNI per Capita** | 0.04 | 0.00 | 0.02 | 0.11 | 0.18 | 0.08 | 0.00 | 0.10 |
| **Press Freedom** | 0.09 | 0.00 | 0.02 | 0.05 | 0.08 | 0.01 | 0.09 | 0.05 |
| **Combined Education** | 0.23 | 0.06 | 0.02 | 0.13 | 0.25 | 0.18 | 0.04 | 0.11 |

Next, we bind Score\_Rank to combo\_norm\_matr\_trans by row and store the result in a variable called combo\_matrix.

combo\_matrix <- rbind(combo\_norm\_matr\_trans, Score\_Rank)

Before viewing the result, we bold our Score and Rank rows and our Syria column.

emphasize.strong.rows(c(1, nrow(combo\_matrix), nrow(combo\_matrix) - 1))  
  
emphasize.strong.cols(7)

# Summary Table

pander(combo\_matrix)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Country** | **Cuba** | **Ghana** | **India** | **Mexico** | **Russia** | **South Africa** | **Syria** | **Thailand** |
| **Water Access** | 0.21 | 0.00 | 0.19 | 0.25 | 0.27 | 0.15 | **0.07** | 0.30 |
| **Life Expectancy** | 0.06 | 0.00 | 0.02 | 0.05 | 0.03 | 0.00 | **0.03** | 0.05 |
| **Urban Population** | 0.11 | 0.06 | 0.00 | 0.12 | 0.10 | 0.08 | **0.06** | 0.05 |
| **GNI per Capita** | 0.04 | 0.00 | 0.02 | 0.11 | 0.18 | 0.08 | **0.00** | 0.10 |
| **Press Freedom** | 0.09 | 0.00 | 0.02 | 0.05 | 0.08 | 0.01 | **0.09** | 0.05 |
| **Combined Education** | 0.23 | 0.06 | 0.02 | 0.13 | 0.25 | 0.18 | **0.04** | 0.11 |
| **Score** | **0.74** | **0.12** | **0.27** | **0.71** | **0.91** | **0.50** | **0.29** | **0.66** |
| **Rank** | **2** | **8** | **7** | **3** | **1** | **5** | **6** | **4** |

# Map

To faciliate understanding the data, we can visualize it with a map.

map\_world <- map\_data(map = "world")  
  
index\_countries <- Score %>%  
 inner\_join(map\_world, by = c("Country" = "region"))  
  
nonindex\_countries <- map\_world %>%  
 anti\_join(combo\_norm, by = c("region" = "Country"))

index\_map <- ggplot() +  
 geom\_polygon(data = index\_countries,  
 aes(x = long, y = lat,  
 group = group,  
 fill = Score), size = 0.01) +  
 geom\_polygon(data = nonindex\_countries,  
 aes(x = long, y = lat, group = group),  
 fill = "white",  
 size = 0.1) +  
 coord\_map() +  
 scale\_fill\_distiller(type = "seq",  
 palette = "RdBu",  
 breaks = pretty\_breaks(n=10),  
 direction = 1) +  
 coord\_fixed(ratio = 1.5) +  
 guides(fill = guide\_legend(keywidth = 1, keyheight = 3, reverse = TRUE)) +  
 ggtitle("Development Index") +  
 theme(axis.title.x = element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 axis.title.y = element\_blank(),  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank()  
 )

index\_map

