Appending provider datasets to one master dataset

August 2024

1. To start, install the r packages *tidyverse*,*writexl*, *readxl* and *lubridate*. Then, run the library. Delete the # to run the installation.

```
#install.packages("tidyverse")
#install.packages("writexl")
#install.packages("readxl")
#install.packages("lubridate")
#install.packages("knitr")

library(tidyverse)
library(readxl)
library(writexl)
library(lubridate)
library(knitr)
```

2. Read each provider's fee tables. The `list.files` is a helpful function to draw names of files with a specific naming pattern.

```
file.list <- list.files(pattern='2024-01-25_.xlsx') #Scraped fee tables reviewed on 12/11/2022
file.list2 <- list.files(pattern='2024-01-20_.xlsx') #Scraped fee tables reviewed on 12/22/2022
file.list3 <- list.files(pattern='2024-01-14_.xlsx') #Scraped fee tables reviewed on 1/03/2023

df.list <- lapply(file.list, read_excel)
df.list2 <- lapply(file.list2, read_excel)
df.list3 <- lapply(file.list3, read_excel)
```

3. Append datasets.

```
change_date_format <- function(tib) {
  tib %>%
    mutate(date_collection = ifelse(!is.na(date_collection), format(ymd(date_collection), "%Y-%m-%d"), I
}

df.list3 <- map(df.list3, change_date_format)
df.list2 <- map(df.list2, change_date_format)
df.list <- map(df.list, change_date_format)

df <- bind_rows(df.list, df.list2, df.list3, .id = "id")

## Order rows & drop ID
df |> arrange(country, provider, transaction_type) |>
  select(-id) -> df
```

4. Conduct checks.

```r
#1. Check that transaction_type naming is consistent
unique(df$transaction_type)
```

```
##  [1] "cash-in"
##  [2] "cash-out"
##  [3] "international remittance"
##  [4] "on-network p2p transfer"
##  [5] "account-to-cash otc transfer"
##  [6] "cash-to-account otc transfer"
##  [7] "cash-to-cash otc transfer"
##  [8] "cash-out (from off-network transfers)"
##  [9] "off-network p2p transfer"
## [10] "wallet-to-bank transfer"
```

```r
#2. check that all observations have either fee or fee-pct
df %>% filter(is.na(fee) & is.na(fee_pct)) %>% view()

#3. check channel naming is consistent
unique(df$channel)
```

```
## [1] "agent"        "self-service" "agent or ATM"
```

```r
#3. check customer_type naming is consistent
unique(df$customer_type)
```

```
## [1] "registered"                 "unregistered"
## [3] "registered or unregistered"
```

```r
df$customer_type[df$customer_type == "Registered"] <- "registered"
df$customer_type[df$customer_type == "registered user"] <- "registered"
df$customer_type[df$customer_type == "unregistered user"] <- "unregistered"
df$customer_type[is.na(df$customer_type)] <- "registered"

unique(df$customer_type)
```

```
## [1] "registered"                 "unregistered"
## [3] "registered or unregistered"
```

```r
#4. check that all fee ranges have a value_min and value_max
df %>% filter(is.na(value_min))
```

```
## # A tibble: 0 x 20
## # i 20 variables: country <chr>, mobile_money <chr>, provider <chr>,
## #   transaction_type <chr>, channel <chr>, customer_type <chr>,
## #   value_min <dbl>, value_max <dbl>, fee <dbl>, tax_pct <dbl>, currency <chr>,
## #   exchange_rate <dbl>, value_min_USD <dbl>, value_max_USD <dbl>,
## #   fee_USD <dbl>, date_collection <chr>, web_address <chr>, tax <dbl>,
## #   tax_USD <dbl>, fee_pct <dbl>
```

```r
df %>% filter(is.na(value_max))
```

```
## # A tibble: 0 x 20
## # i 20 variables: country <chr>, mobile_money <chr>, provider <chr>,
## #   transaction_type <chr>, channel <chr>, customer_type <chr>,
## #   value_min <dbl>, value_max <dbl>, fee <dbl>, tax_pct <dbl>, currency <chr>,
## #   exchange_rate <dbl>, value_min_USD <dbl>, value_max_USD <dbl>,
## #   fee_USD <dbl>, date_collection <chr>, web_address <chr>, tax <dbl>,
## #   tax_USD <dbl>, fee_pct <dbl>
```

```r
#check distinct providers
df %>%
  distinct(country, provider) %>% view() -> provider_list

#reorder columns
col_order <- c("country", "mobile_money", "provider", "transaction_type", "channel", "customer_type",
               "currency", "exchange_rate", "value_min", "value_max", "fee", "fee_pct",
               "tax", "tax_pct", "value_min_USD", "value_max_USD","fee_USD", "tax_USD",
               "date_collection","web_address")
df<- df[,col_order]

#filter for transactions we want to focus on for our analysis
df %>%
  filter(transaction_type %in% c("cash-in",
                                 "cash-out",
                                 "on-network p2p transfer",
                                 "off-network p2p transfer")) ->df
```

5. View and export data into excel.

```r
view(df)
head(df[,1:5],)  #prints the first 5 rows
```

```
## # A tibble: 6 x 5
##   country mobile_money provider    transaction_type        channel
##   <chr>   <chr>        <chr>       <chr>                   <chr>
## 1 Mali    Moov Money   Moov Africa cash-in                 agent
## 2 Mali    Moov Money   Moov Africa cash-out                agent
## 3 Mali    Moov Money   Moov Africa on-network p2p transfer self-service
## 4 Mali    Orange Money Orange      cash-in                 agent
## 5 Mali    Orange Money Orange      cash-out                agent
## 6 Mali    Orange Money Orange      cash-out                agent
```

```r
write_xlsx(df, "Listed Prices Dataset_2023Q1.xlsx")
```