Analysis Report

2024-11-16

Import necessary libraries

```
library(ggplot2)
library(comtradr)
library(tidyverse)
library(lubridate)
library(knitr)
library(RColorBrewer)
library(wesanderson)
library(patchwork)
library(scales)
```

Read and pre-process the Data

Import the monthly gold prices data. Convert date to first day of the month & USD/troy Oz to USD/Kg.

```
gold_df <- read.csv('../data/gold_prices_monthly.csv')
gold_df$date <- as.Date(gold_df$date, format = '%m/%d/%Y')
gold_df$date <- format(gold_df$date,"%Y-%m-01")
gold_df$priceUSD_troyounce <-
  as.numeric(gsub(",", "", gold_df$priceUSD_troyounce))*32.1507
```

Combine the monthly files (2017-2023) into one Data Frame and select the relevant columns.

```
file_list <- list.files(path="../data/export_data/", full.names = TRUE)
ldf <- lapply(file_list, read.csv, row.names = NULL)
df <- do.call("rbind", ldf)

rm(ldf, file_list)

colnames(df) <- colnames(df)[2:ncol(df)]
df <- df[1:(ncol(df)-1)]

df <- df[c("refYear", "refMonth", "reporterDesc", "partnerDesc", "cmdCode",
           "qtyUnitAbbr", "qty", "altQtyUnitAbbr", "altQty", "primaryValue")]

df[df$partnerDesc == "T\xfcrkiye",]$partnerDesc <- 'Turkey'
```

Sum of the HS6 values matches with HS4 values

```
singleC <- df[df$reporterDesc == 'Kenya',]

sum(singleC[singleC$partnerDesc != 'World' & singleC$cmdCode %in%
              c(710811, 710812, 710813, 710820),]$primaryValue)
```

```
## [1] 117828803
```

```
sum(singleC[singleC$partnerDesc == 'World' & singleC$cmdCode == 7108,]$primaryValue)
```

```
## [1] 117828803
```

Trade reported in HS4 matches with HS6 values. No instance of missing trade for HS subdivisions as reported in this article.

> "Due to confidentiality, countries may not report some of its detailed trade. This trade will however be included at the higher commodity level and in the total trade value."

(Need to check for HS2 - 71)

Compare Yearly and Monthly aggregated data

```
tdf <- read.csv('../data/yearly_cleaned_data.csv')

r1 <- df[df$cmdCode == 7108 & df$partnerDesc != 'World',] %>%
  group_by(reporterDesc, partnerDesc) %>%
  summarise(total_pvalue = sum(primaryValue), total_qty = sum(altQty))

r2 <- tdf[tdf$cmdCode == 7108 & tdf$partnerDesc != 'World',] %>%
  group_by(reporterDesc, partnerDesc) %>%
  summarise(total_pvalue = sum(primaryValue), total_qty = sum(altQty))
```

Display the first 5 rows of the aggregated data.

```
kable(head(r1, 5), caption = 'Monthly data')
```

Table 1: Monthly data

reporterDesc	partnerDesc	total_pvalue	total_qty
Kenya	Cameroon	64206.76	1000
Kenya	Canada	37462.53	1000
Kenya	China	206073.69	0
Kenya	China, Hong Kong SAR	2523486.63	40299
Kenya	Cyprus	48551.75	1

```r
kable(head(r2, 5), caption = 'Yearly data')
```

Table 2: Yearly data

reporterDesc	partnerDesc	total_pvalue	total_qty
Kenya	Cameroon	63215.52	1000
Kenya	Canada	37828.03	1000
Kenya	China	225209.28	0
Kenya	China, Hong Kong SAR	2821389.58	50406
Kenya	Cyprus	50441.28	1

```r
# Primary Values in Yearly that are not same in Monthly
length(unique(r2$total_pvalue[! r2$total_pvalue %in% r1$total_pvalue]))
```

```
## [1] 31
```

```r
# Quantity Values in Yearly that are not same in Monthly
length(unique(r2$total_qty[! r2$total_qty %in% r1$total_qty]))
```

```
## [1] 12
```

For the monthly vs yearly aggregated data, all the gold export values (primaryValue) between the respective reporters and partners don't match, they are off by a few perc. Meanwhile 60% of the quantity values are the same.

```r
unique(r2$partnerDesc[! r2$partnerDesc %in% r1$partnerDesc])
```

```
## [1] "France" "Oman"
```

France and Oman appear in the Yearly data, but not in the Monthly data.

Add BRICS flag, merge the date columns into date format (Month/Year).

```r
brics <- c("China, Hong Kong SAR", "China", "South Africa", "India", "Russia", "Brazil")

west <- c("USA", "United Kingdom", "Spain", "France", "Germany", "Turkey", "Italy",
          "Japan", "Rep. of Korea", "Switzerland", "Canada", "Cyprus", "Slovenia")

df$bricsflag <- NA
df[df$partnerDesc %in% brics,]$bricsflag <- 'b'
df[df$partnerDesc %in% west,]$bricsflag <- 'w'

df$date <- as.Date(paste('01', df$refMonth, df$refYear, sep = '/'), format = "%d/%m/%Y")
gold_df <- gold_df[gold_df$date >= min(df$date) & gold_df$date <= max(df$date),]
```

```
df$refYear <- NULL
df$refMonth <- NULL

df$gold_price <- sapply(df$date, function(x)
  gold_df[gold_df$date == x,]$priceUSD_troyounce)

df <- df[df$partnerDesc != 'World',]
```

Make Unit Price column and show issues with the Quantity columns

This column shows the price (USD) of the commodity per kg. Ideally it should be close to the Gold Spot price value and follow the same trend.

Looking at the `qty` and `altQty` columns, we can see the values do not match with each other. Lets make a a new column showing the proportion difference between `altQty` and `qty` columns. We can see some match `1:1`, while others are off by an order of `1000` (the `altQtyAbbr` shows that they are in grams). Some also fall in the range `(900,1000)` and `(1000-1100)`, we can take the `qty` value as the final value.

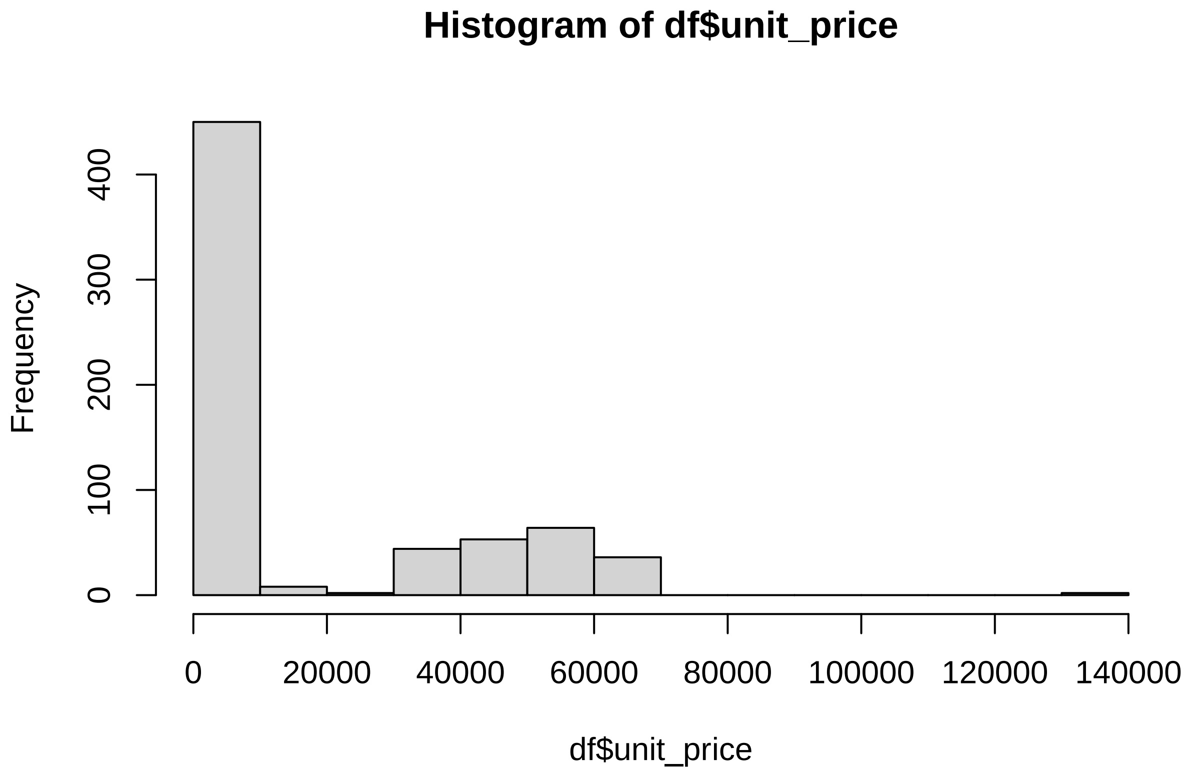
We will assume the `primaryValue` column has no errors, while the quantity columns have some unit conversion errors (displayed in kgs, grams and milligrams).

```
df$qtyTransform <-df$altQty/df$qty

unique(df$qtyTransform)
```

```
##  [1]    1.0000 1000.0000       NaN       Inf 3000.0000  246.0000 1166.6667
##  [8]  371.7967  833.3333 1000.0000  151.0000 1000.0000  137.0000  214.0000
## [15]  519.0000    0.0000 1005.0000 1016.3333 1000.0401  999.9978 1000.0079
## [22] 1000.0088  999.9967  999.7605 1000.0044  997.4359 1000.0058 1002.7778
## [29] 1001.8750 1000.0082 1000.0030  999.9921 1000.0043  999.9915  999.9943
## [36] 1000.0012  999.9973 1000.0105  999.1826  999.9978  999.9841
```

```
df$unit_price <- df$primaryValue/df$altQty
df[df$unit_price == Inf,]$unit_price <- NA
```

```
hist(df$unit_price)
```



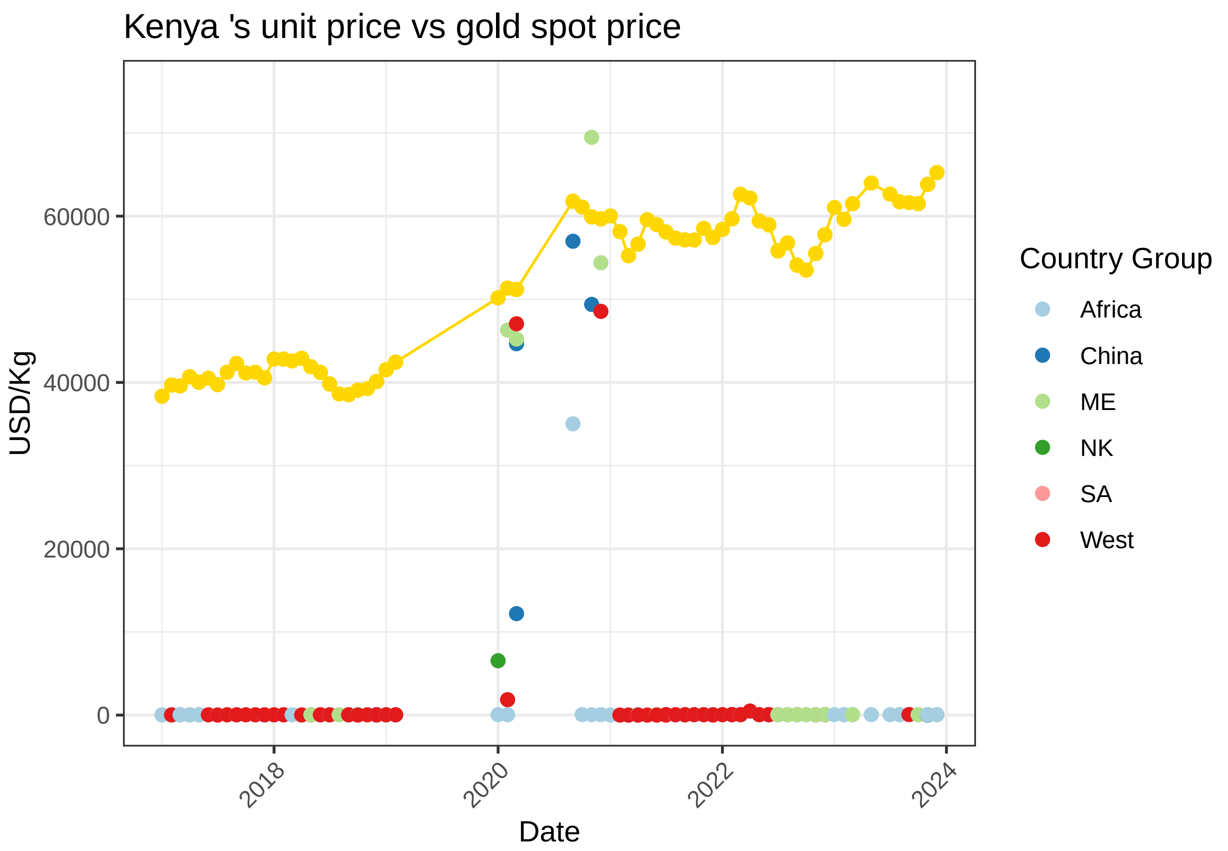
A lot of the unit price values fall between (0, 100000) USD/Kg.

Lets plot the unit price column monthly and compare it with the Gold Spot price value (before transforming Quantity Column)

```
# Graph 2
cmdcode_unitprice_plot = function(df, Reporter){

  plot_df <- df[df$reporterDesc == Reporter & df$cmdCode != '7108',]


  plot_df$ctyGroup <- NA
  africa <- c('Cameroon', 'Dem. Rep. of the Congo', 'Nigeria', 'South Africa', 'Ethiopia', 'Eswatini')
  ME <- c('United Arab Emirates', 'Qatar')
  SA <- c('Paraguay')
  legTitle <- 'Country'

  if(Reporter == 'Kenya'){
    plot_df[plot_df$partnerDesc %in% africa,]$ctyGroup <- 'Africa'
    plot_df[plot_df$partnerDesc %in% west,]$ctyGroup <- 'West'
    plot_df[plot_df$partnerDesc %in% c('China', 'China, Hong Kong SAR'),]$ctyGroup <- 'China'
    plot_df[plot_df$partnerDesc == 'Dem. People\'s Rep. of Korea',]$ctyGroup <- 'NK'
    plot_df[plot_df$partnerDesc %in% ME,]$ctyGroup <- 'ME'
```

```r
    plot_df[plot_df$partnerDesc %in% SA,]$ctyGroup <- 'SA'
    legTitle <- 'Country Group'
  }

  cl <- length(unique(plot_df$partnerDesc))

  g <- ggplot(data = plot_df, aes(x = date)) +
    geom_point(aes(y = gold_price), color = 'gold', size=2) +
    geom_line(aes(y = gold_price), color = 'gold') +
    geom_point(aes(y = unit_price, color =
                     !!ifelse(Reporter == 'Kenya', sym("ctyGroup"), sym("partnerDesc"))), size=2) +
    scale_y_continuous(limits = c(0, 75000)) +
    theme_bw() +
    theme(axis.text.x = element_text(angle=45, hjust = 1)) +
    ylab("USD/Kg") + xlab("Date") +  guides(color=guide_legend(title=legTitle)) +
    labs(title = paste(Reporter, '\'s unit price vs gold spot price'))

  if(cl <= 5){
    g = g + scale_color_manual(values =
                                 wes_palette(n=length(unique(plot_df$partnerDesc)), name="Darjeeling1")
  }else{
    g = g + scale_colour_brewer(palette = "Paired")
  }

  return(list(g, plot_df))
}
```

```r
Reporter <- 'Kenya'

plot_stats = cmdcode_unitprice_plot(df, Reporter)

plot_stats[[1]]
```

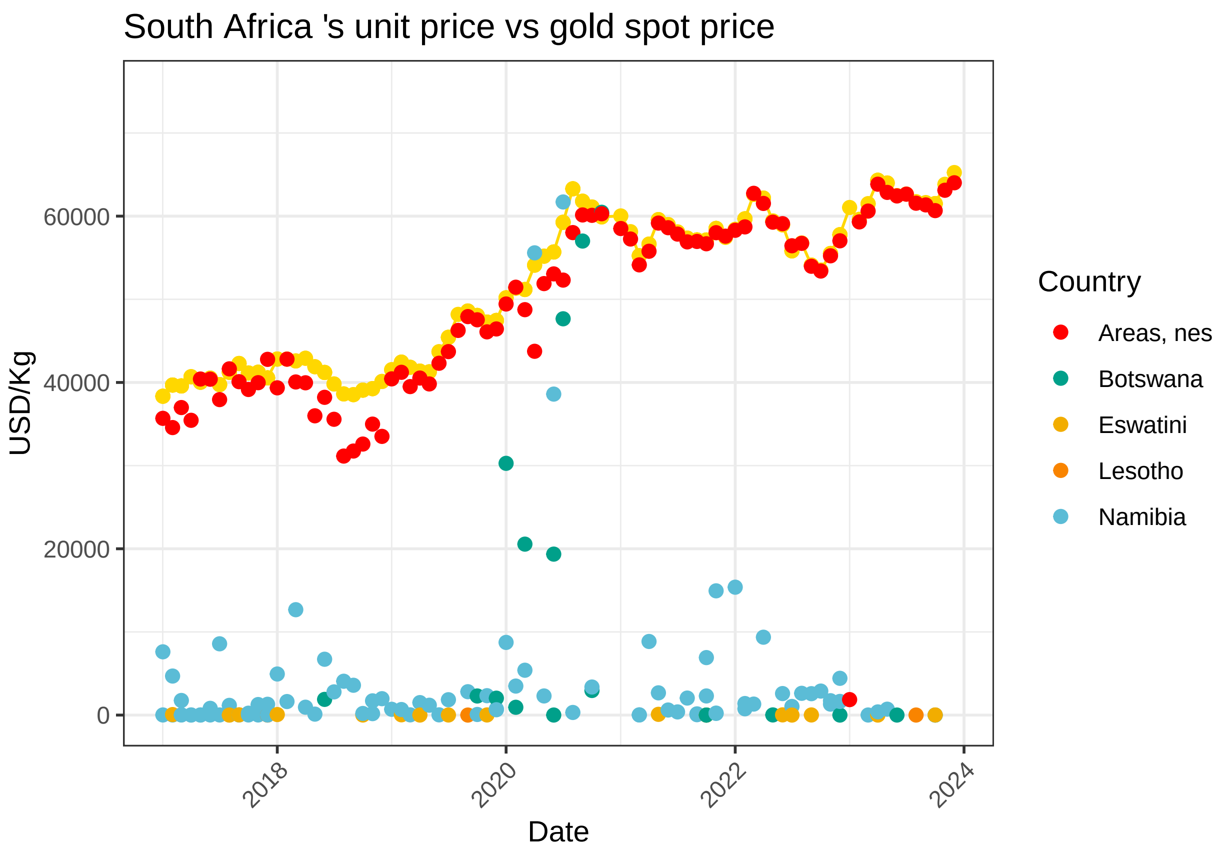


NOTE: 710811, 710812, 710813, 710820 are shown individually, so for the same month, there will be multiple points, they are not aggregated

Could later check if there is pattern for the different commodities (unit price vs commodity).

As we can see in the graph, lot of the values are close to 0. This indicates an issue with the quantity column.

```r
Reporter <- 'South Africa'

plot_stats = cmdcode_unitprice_plot(df, Reporter)

plot_stats[[1]]
```



`Area, nes` unit price has its values close to the gold spot price indicating that the quantity columns are in proper units.

Transform / Pre-process the Quantity column

Find the quantity values that are likely in grams, milligrams and convert to Kg.

For the quantity that are in grams, the corresponding unit price will have two whole part digits.

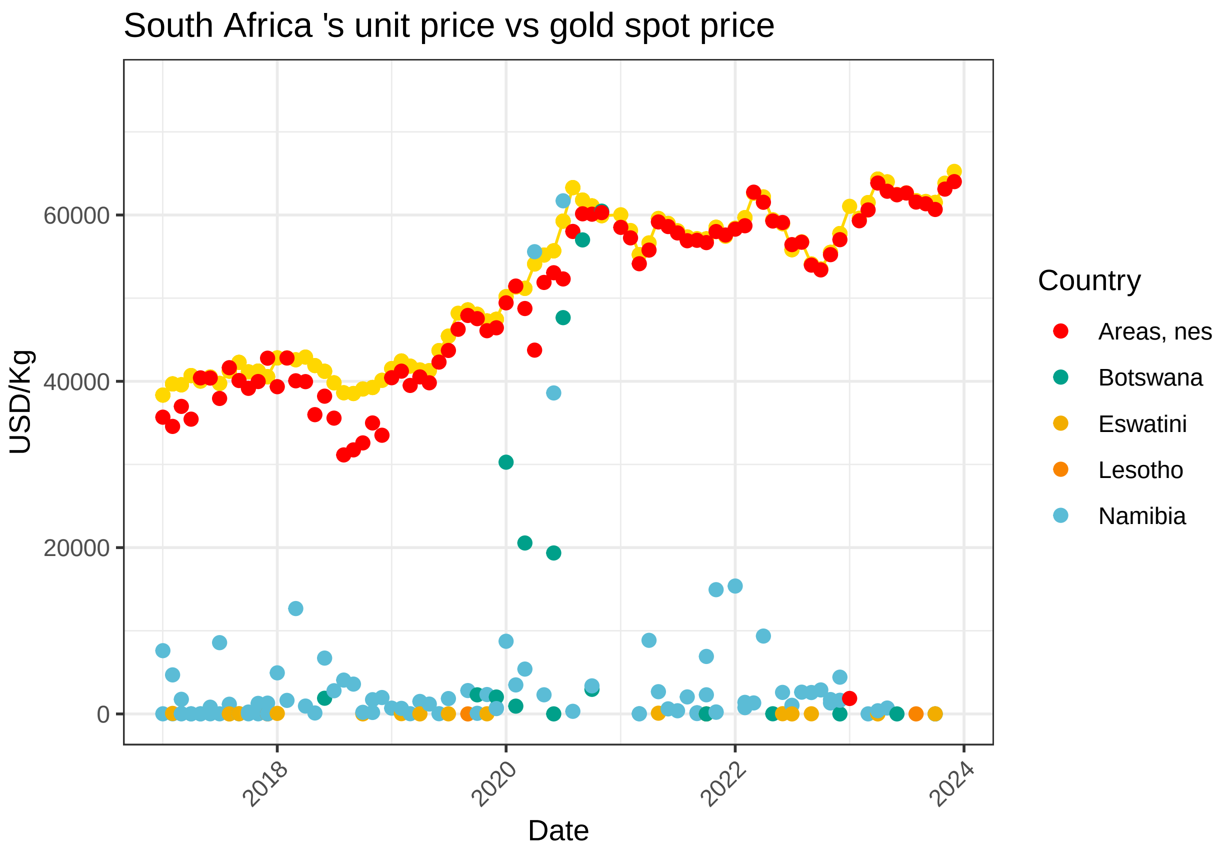
For the quantity that are in milligrams, the corresponding unit price's first decimal digit & only whole part will be 0.

```
temp <- data.frame(do.call(rbind, strsplit(as.character(df$unit_price), '\\.')))
temp$altQtyUnitAbbr <- df$altQtyUnitAbbr

# handle grams
#df[which(nchar(temp$X1) == 2),]
df[which(nchar(temp$X1) == 2 & temp$altQtyUnitAbbr == 'g'),]$altQty <-
  df[which(nchar(temp$X1) == 2 & temp$altQtyUnitAbbr == 'g'),]$altQty/1000
df[which(nchar(temp$X1) == 2 & temp$altQtyUnitAbbr == 'g'),]$unit_price <-
  df[which(nchar(temp$X1) == 2 & temp$altQtyUnitAbbr == 'g'),]$unit_price*1000

# handle milligrams
df[which(temp$X1 == '0' & substr(temp$X2, start = 1, stop = 1) == '0'),]$altQty <-
  df[which(temp$X1 == '0' & substr(temp$X2, start = 1, stop = 1) == '0'),]$altQty/1000000
```

```
df[which(temp$X1 == '0' & substr(temp$X2, start = 1, stop = 1) == '0'),]$unit_price <-
  df[which(temp$X1 == '0' & substr(temp$X2, start = 1, stop = 1) == '0'),]$unit_price*1000000

# convert grams to kg
#df[df$altQtyUnitAbbr == 'g',]$altQty <- df[df$altQtyUnitAbbr == 'g',]$altQty/1000
#df$unit_price <- df$primaryValue/df$altQty

# find the difference between gold spot price and unit price
df$gdiff <- df$gold_price - df$unit_price
```

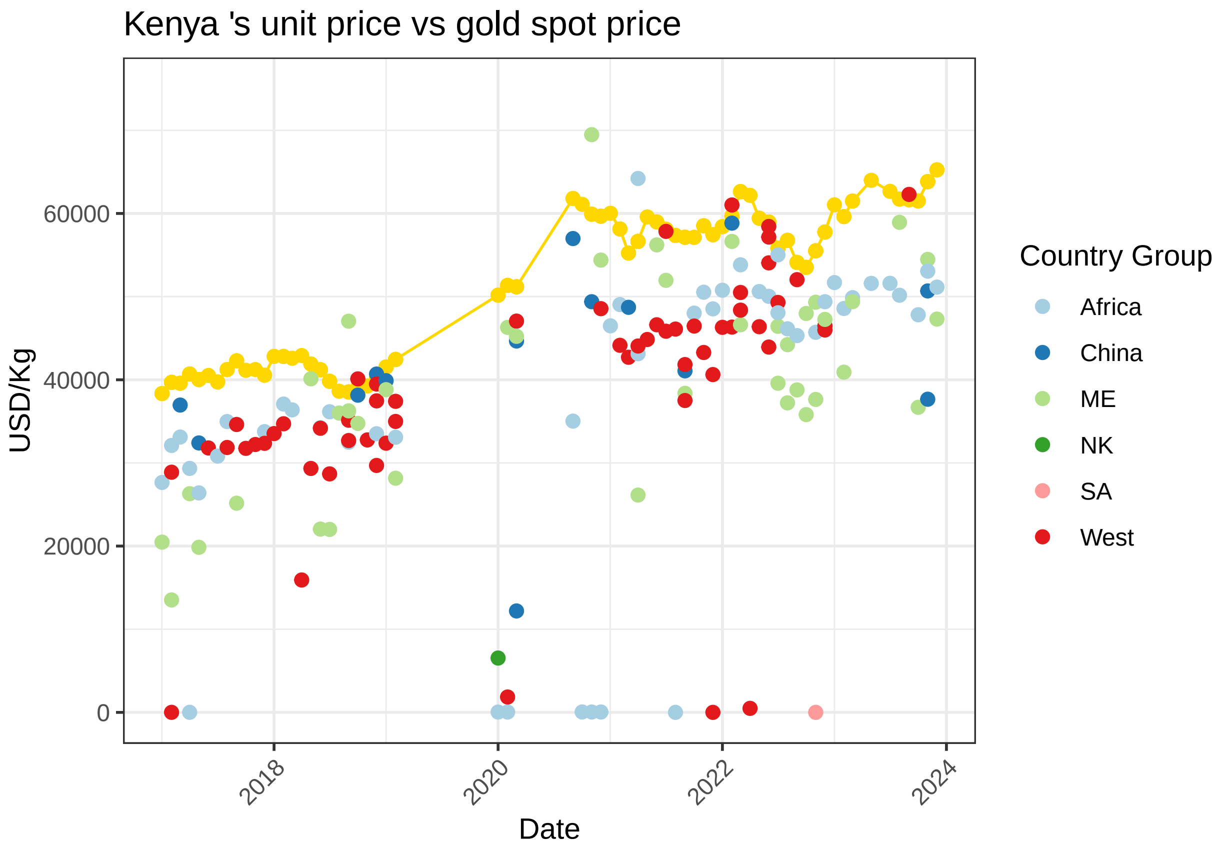
Compare unit price vs gold spot price after transformation

```
Reporter <- 'South Africa'

plot_stats = cmdcode_unitprice_plot(df, Reporter)

plot_stats[[1]]
```



No change for South Africa subset, this should indicate no unit conversion issues. Eg; Looking at Namibia's quantity columns, there doesn't seem to be any unit conversion errors. From what we learned in the class lecture there could be some sort of resource trade happening (ie; Gold is exchanged for Uranium ore). But lot of the trade value (primaryValue) and quantity are still too low to make sense.

```r
Reporter <- 'Kenya'

plot_stats = cmdcode_unitprice_plot(df, Reporter)

plot_stats[[1]]
```



We can see that there were unit conversion issues with the Kenya subset.

It seems like the country group has an effect on the unit price, eg; Middle-east buys gold for cheaper/kg than the West or Africa.

Lets check if the transformed quantity -> corrected unit price trend is similar to the gold price trend

Again given our assumption that the reported trade unit price (USD/Kg) should follow the trend of Gold Spot price …

If it follows the same trend, then our correction for the quantity values should ideally be correct.

We will do an anova test to see if the slopes of the linear model fit for the two variables (unit price, gold spot price) are similar. We have to remove some outliers before we run the model.

```r
plot_df <- plot_stats[[2]]
ttestDF <- plot_df[plot_df$ctyGroup == 'West',] %>% select(unit_price, gold_price, date)
ttestDF <- ttestDF[ttestDF$unit_price > 10000 & ttestDF$unit_price < 70000,]
ttestDF <- ttestDF %>% group_by(date) %>%
  summarise(unit_price = mean(unit_price), gold_price = mean(gold_price))

expected <- ttestDF$gold_price
observed <- ttestDF$unit_price
time <- ttestDF$date

wide <- data.frame(expected, observed)
long <- cbind(time, stack(wide))

fm2 <- lm(values ~ ind/(time + 1) + 0, long)
fm1 <- lm(values ~ ind + time + 0, long)
anova(fm1, fm2)
```

```
## Analysis of Variance Table
##
## Model 1: values ~ ind + time + 0
## Model 2: values ~ ind/(time + 1) + 0
##   Res.Df        RSS Df Sum of Sq      F Pr(>F)
## 1     79 1158347962
## 2     78 1157685128  1    662833 0.0447 0.8332
```

```
P-value > 0.05
```

We ran a linear model with two slopes and and another with one slope. Both have two intercepts. Then compare them using anova. The slopes are not significantly different so we cannot reject the NULL hypothesis that the slopes are same.

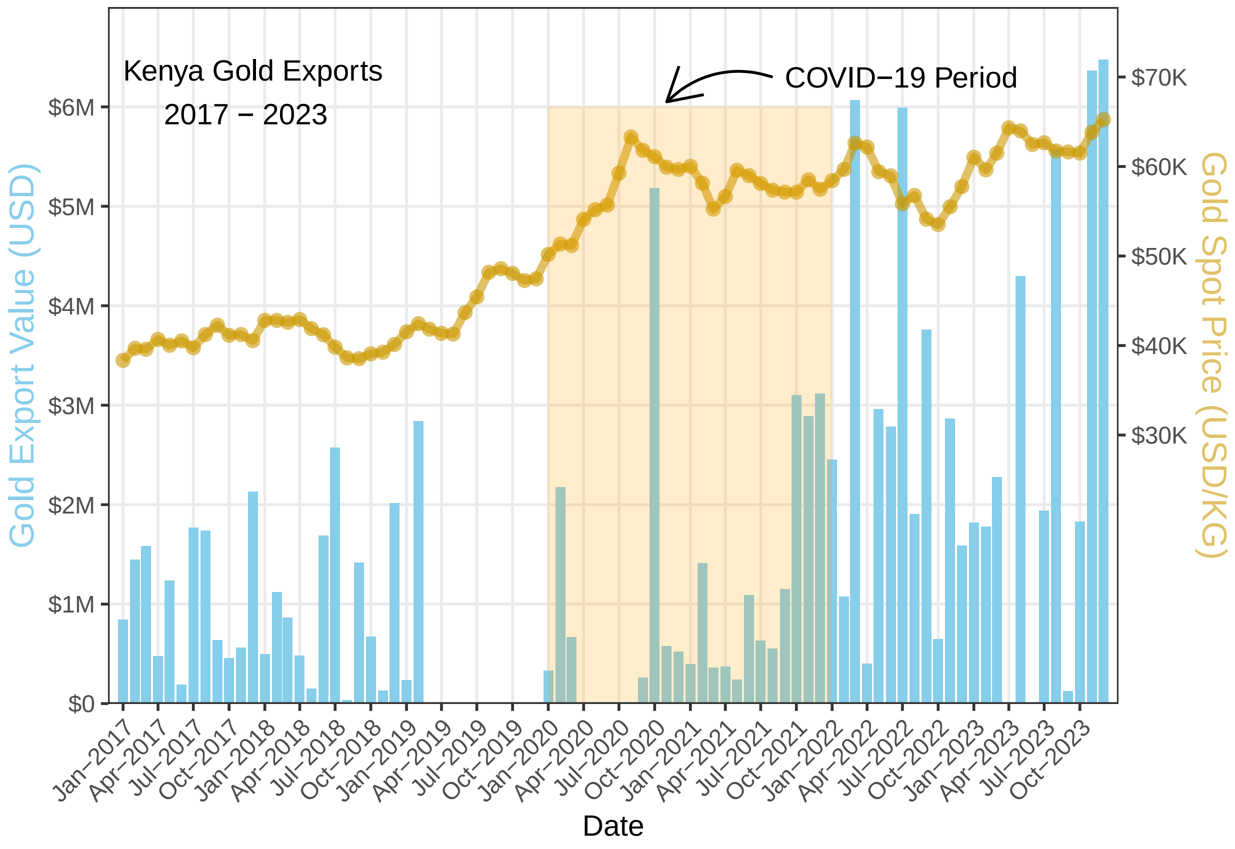
We can also use the granger test to see if a timeseries model fit for gold spot price can be used for the unit price. This should indicate if the two trends are same.

```r
# https://www.r-bloggers.com/2021/11/granger-causality-test-in-r-with-example/
#library(lmtest)

#tsData <- data.frame(observed, expected)
#grangertest(observed ~ expected, order = 3, data = tsData)

#grangertest(expected ~ observed, order = 3, data = tsData)

#We may reject the null hypothesis of the test because the p-value is smaller than 0.05,
#and infer that knowing the values of Gold Spot Price is valuable for forecasting
#the future values of Unit Price.
```

Plot gold export trade over 2017-2023 with COVID-19 period shown

```r
# Graph 1

# df$partnerDesc %in% brics &
plot_df <- df[df$reporterDesc == "Kenya" & df$cmdCode == 7108,]

plot_df <- plot_df %>% group_by(date) %>% summarise(sum = sum(primaryValue))
```

```r
plot_df$gold_price <- NA

plot_df <- plot_df %>%
  complete(date = seq(date[1], date[nrow(plot_df)], by = "1 month"),
           fill = list(sum = 0, gold_price = 0))

plot_df$gold_price <- sapply(plot_df$date, function(x)
  gold_df[gold_df$date == x,]$priceUSD_troyounce)

yrng <- range(plot_df$sum)
xrng <- range(plot_df$date)
caption <- paste("Kenya Gold Exports\n", "    2017 - 2023")

from <- as.Date('2020-01-01')
to <- as.Date('2022-01-01')

shade <- data.frame(from, to)

breaks.vec <- seq(min(plot_df$date), max(plot_df$date), by = "3 month")
scaleVal <- 1/90
gpriceColor <- rgb(0.8, 0.6, 0, 0.6)

ggplot(data = plot_df, aes(x = date, y = sum)) +
  geom_bar(stat = "identity", fill="skyblue") +
  geom_point(aes(y = gold_price/scaleVal), size=2, color=gpriceColor) +
  geom_line(aes(y = gold_price/scaleVal), linewidth=1.5, color=gpriceColor) +
  #scale_x_date(expand = c(0.01, 0.01), date_labels = "%b-%Y", date_breaks = "3 month") +
  scale_x_date(expand = c(0.01,0.01), breaks = breaks.vec, date_labels = "%b-%Y") +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 7000000), breaks = seq(0, 6000000, by = 1000000),
                     labels = scales::label_dollar(scale_cut = scales::cut_short_scale()),
                     # https://finchstudio.io/blog/ggplot-dual-y-axes/
                     sec.axis = sec_axis(~.*scaleVal, name="Gold Spot Price (USD/KG)",
                                         labels = scales::label_dollar(scale_cut =
                                                        scales::cut_short_scale()),
                                         breaks = c(30000,40000,50000,60000,70000))
                     ) +
  annotate(geom = "text", x = xrng[1], y = yrng[2], label = caption,
           hjust = 0, vjust = 1, size = 4) +
  theme_bw() +
  theme(axis.text.x = element_text(angle=45, hjust = 1),
        axis.title.y = element_text(color = 'skyblue', size = 13),
        axis.title.y.right = element_text(color = gpriceColor, size = 13)) +
  theme(panel.grid.minor.x = element_blank()) +
  theme(panel.grid.minor.y = element_blank()) +
  geom_rect(data = shade, inherit.aes=FALSE,
            aes(xmin = from, xmax = to, ymin = -Inf, ymax = 6000000),
            color="transparent", fill="orange", alpha=0.2) +
  annotate(geom = "curve", x = as.Date('2021-08-01'), y = 6300000,
           xend = as.Date('2020-11-01'), yend = 6050000,
           curvature = .3, arrow = arrow(length = unit(5, "mm"))) +
  annotate(geom = "text", x = as.Date('2021-09-01'), y = 6300000,
           label = "COVID-19 Period", hjust = "left") +
  xlab("Date") + ylab("Gold Export Value (USD)")
```

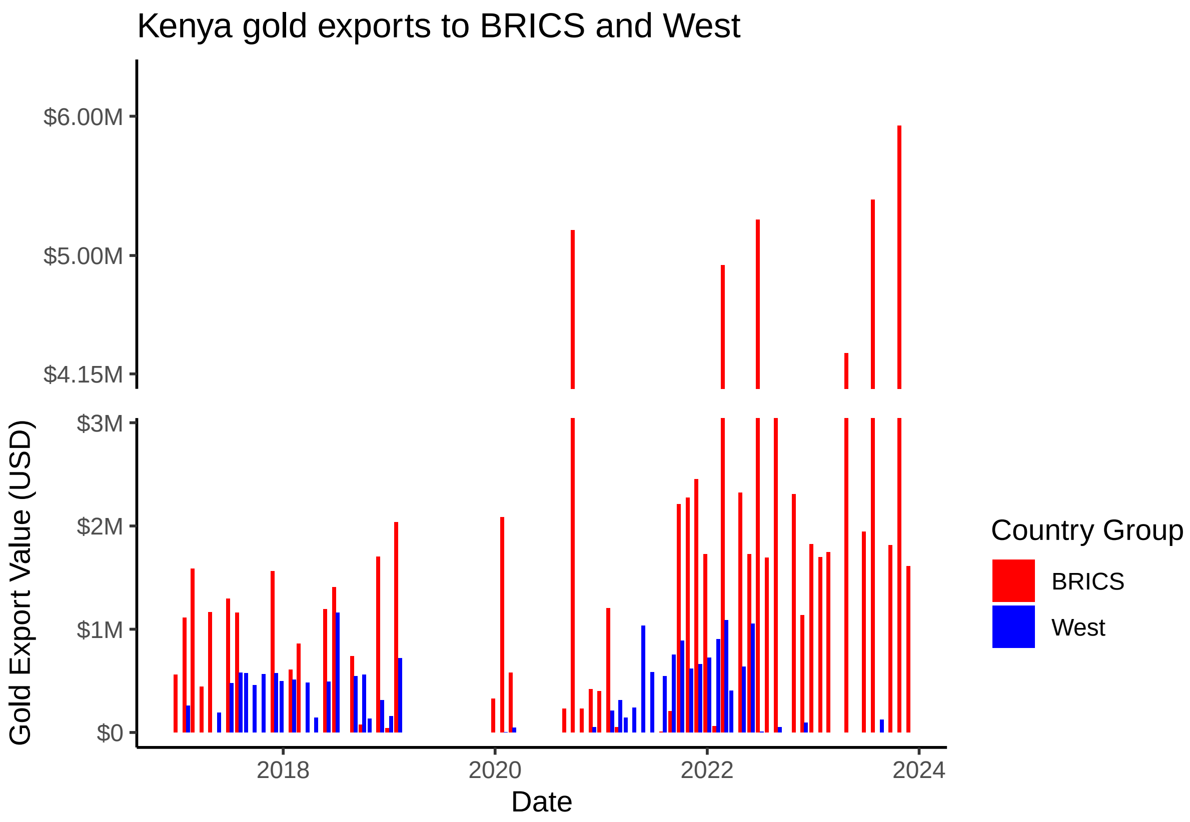


Compare BRICS and West trade with the corresponding reporters

```
plot_df <- df[df$bricsflag %in% c('b', 'w') & df$cmdCode == 7108,]

plot_df <- plot_df %>% group_by(bricsflag, date) %>% summarise(sum = sum(primaryValue))
```

```
## ‘summarise()‘ has grouped output by ’bricsflag’. You can override using the
## ‘.groups‘ argument.
```

```
breaks.vec <- seq(min(plot_df$date), max(plot_df$date), by = "3 month")

# Graph 3.1 - with Y break for better readability

# bottom part
p1=ggplot(data=plot_df,aes(date,sum,fill=bricsflag))+
  geom_bar(stat = "identity", position=position_dodge(preserve = "single"))+
  coord_cartesian(ylim = c(0,2900000))+ #setting the down part
  scale_y_continuous(labels = scales::label_dollar(scale_cut = scales::cut_short_scale()))+
  ylab("Gold Export Value (USD)") + xlab("Date") +
  guides(fill=guide_legend(title='Country Group')) +
  scale_fill_manual(labels = c("BRICS", "West"), values = c("red", "blue"))+
  theme_classic()

# top part
p2=ggplot(data=plot_df,aes(date,sum,fill=bricsflag))+
  geom_bar(stat = "identity", position=position_dodge(preserve = "single"))+
  coord_cartesian(ylim = c(4150000,6300000))+ #setting the upper part
  scale_y_continuous(breaks = c(4150000, 5000000, 6000000),
                     labels = scales::label_dollar(scale_cut = scales::cut_short_scale()))+
  scale_fill_manual(values = c("red", "blue"))+
  guides(x = "none")+ # remove x line
  labs( title = "Kenya gold exports to BRICS and West")+
  theme_classic()+
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title = element_blank(),
        legend.position = "none", # remove legend
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"))

#patchwork
p2/p1
```



Export - Import descrepancies

```
selectList <- c('Mali', 'United Rep. of Tanzania', 'Ghana', 'Togo', 'Cameroon', 'Benin', 'Ethiopia',
                'Niger', 'Madagascar', 'Rwanda', 'Mozambique', 'Senegal', 'Burundi')

export <- read.csv('../data/uae_export.csv', row.names = NULL)
colnames(export) <- colnames(export)[2:ncol(export)]
export <- export[1:(ncol(export)-1)]

import <- read.csv('../data/uae_import.csv', row.names = NULL)
colnames(import) <- colnames(import)[2:ncol(import)]
import <- import[1:(ncol(import)-1)]


export <- export[export$reporterDesc %in% selectList,]
export <- export %>% select(reporterDesc, primaryValue)

import <- import[import$partnerDesc %in% selectList,]
import <- import %>% select(partnerDesc, primaryValue)
colnames(import) <- c('reporterDesc', 'primaryValue')

export <- export[export$reporterDesc %in% unique(import$reporterDesc),]
import <- import[import$reporterDesc %in% unique(export$reporterDesc),]

wide <- data.frame(export$primaryValue, import$primaryValue)
long <- cbind(export$reporterDesc, stack(wide))
colnames(long) <- c('Country', 'Value', 'Flag')
long$Value <- long$Value/1000000

right_label <- long %>%
  group_by(Country) %>%
  arrange(desc(Value)) %>%
  top_n(1)
```

```
## Selecting by Flag
```

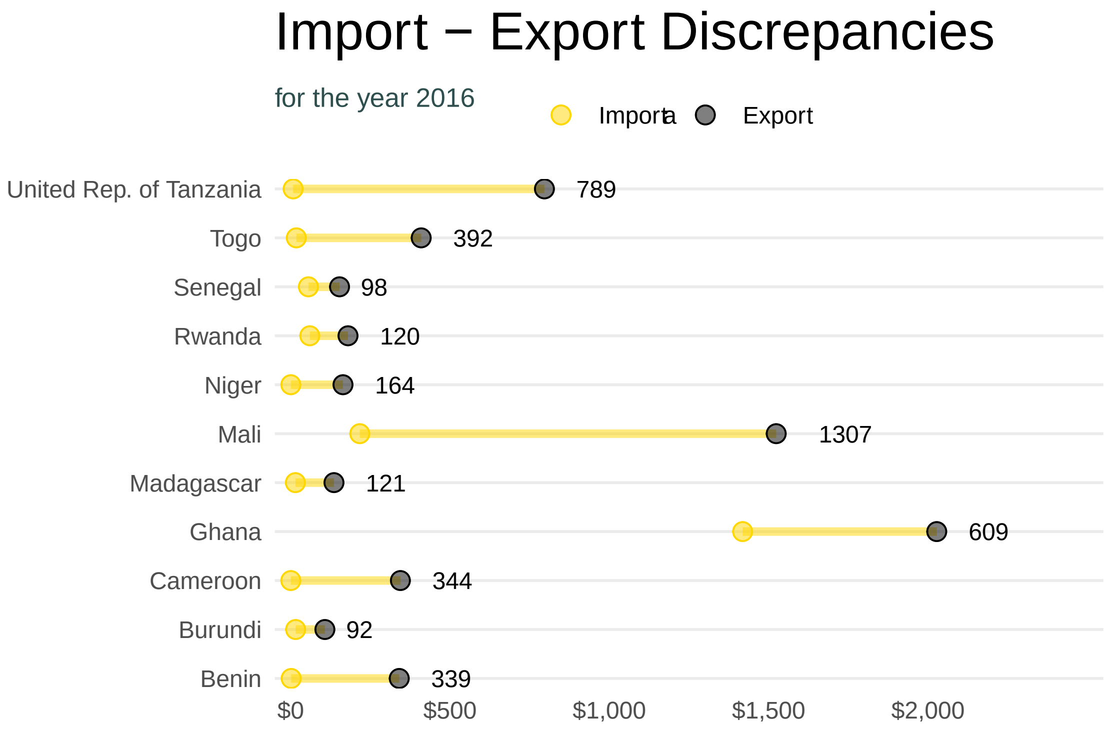
```
left_label <- long %>%
  group_by(Country) %>%
  arrange(desc(Value)) %>%
  slice(2)


# create data frame that identifies revenue differences over 20%
big_diff <- long %>%
  spread(Flag, Value) %>%
  group_by(Country) %>%
  mutate(Max = max(import.primaryValue, export.primaryValue),
         Min = min(import.primaryValue, export.primaryValue),
         Diff = Max - Min) %>%
  arrange(desc(Diff)) %>%
  filter(Diff > .2)
```

```r
# filter the label data frames to only include those cities where the
# difference exceeds 20%
right_label <- filter(right_label, Country %in% big_diff$Country)
left_label <- filter(left_label, Country %in% big_diff$Country)

# filter the main data frame to only include those cities where the
# difference exceeds 20%.
highlight <- filter(long, Country %in% big_diff$Country)

plot_label <- big_diff %>%
  select(Country, Value = Max, Diff) %>%
  right_join(right_label)
```

```
## Joining with ‘by = join_by(Country, Value)‘
```

```r
p <- ggplot(long, aes(Value, Country)) +
  geom_line(aes(group = Country), alpha = .5, color = 'gold', linewidth = 1.5) +
  geom_point(aes(color = Flag), size = 3, alpha = .5) +
  #geom_line(data = highlight, aes(group = Country)) +
  geom_point(data = highlight, aes(color = Flag), size = 3, pch=21) +
  geom_text(data = plot_label, aes(color = Flag,
                                    label = round(Diff, 0)),
            size = 3, hjust = -.8)

p + scale_color_manual(labels = c("Import", "Export"),values = c('gold', 'black')) +
  scale_x_continuous(labels = scales::dollar, expand = c(0.02, 0),
                     limits = c(0, 2500),
                     breaks = seq(0, 2000, by = 500)) +
  scale_y_discrete(expand = c(.02, 0)) +
  labs(title = "Import - Export Discrepancies", subtitle = "for the year 2016") +
  theme_minimal() +
  theme(axis.title = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank(),
        legend.title = element_blank(),
        legend.justification = c(0, 1),
        legend.position = c(.3, 1.2),
        legend.background = element_blank(),
        legend.direction="horizontal",
        plot.title = element_text(size = 20, margin = margin(b = 10)),
        plot.subtitle = element_text(size = 10, color = "darkslategrey", margin = margin(b = 25)),
        plot.caption = element_text(size = 8, margin = margin(t = 10), color = "grey70", hjust = 0),
        plot.margin = margin(t = 1, r = 1, b = 1, l = 1, unit = "cm"))
```

```
## Warning: A numeric ‘legend.position‘ argument in ‘theme()‘ was deprecated in ggplot2
## 3.5.0.
## i Please use the ‘legend.position.inside‘ argument of ‘theme()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```


Import − Export Discrepancies
for the year 2016
Importa
Export
United Rep. of Tanzania
789
Togo
392
Senegal
98
Rwanda
120
Niger
164
Mali
1307
Madagascar
121
Ghana
609
Cameroon
344
Burundi
92
Benin
339
$0
$500
$1,000
$1,500
$2,000