Figure of 'The Ukraine-Russia war, global wheat, and food security'

Frédéric Baudron, Alison Bentley & Janet M Lewis

May 28th, 2022

```
# LOADING REQUIRED PACKAGES---------------------------------------------------

library(tidyverse)
```

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

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.1.3

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(spData)
```

```
## Warning: package 'spData' was built under R version 4.1.1

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.1.3

## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(ggfx)
```

```
## Warning: package 'ggfx' was built under R version 4.1.1
```

```
library(viridis)
```

```
## Warning: package 'viridis' was built under R version 4.1.2

## Loading required package: viridisLite
```

```
library(viridisLite)
library(ggalluvial)
library(ggthemes)
library(dplyr)
library(egg)
```

```
## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

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

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggthemes':
##
##     theme_map
```

```
library(patchwork)
```

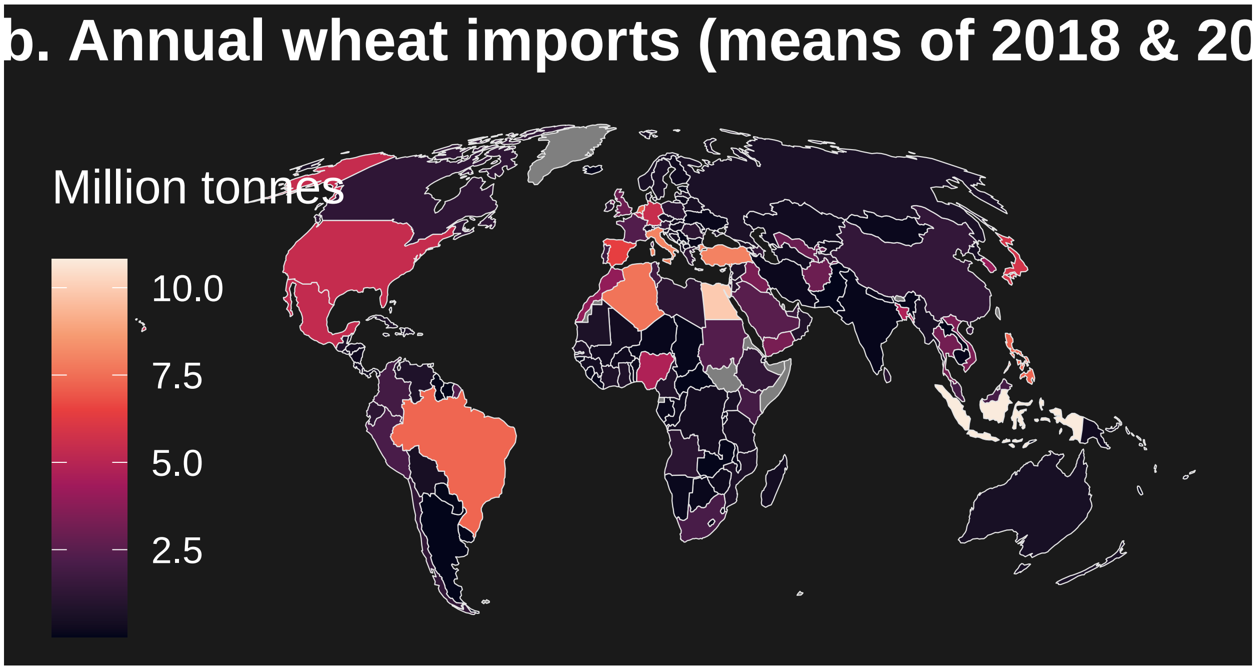
```
##
## Attaching package: 'patchwork'

## The following object is masked from 'package:cowplot':
##
##     align_plots
```

```
# SETTING DIRECTORY------------------------------------------------------------

setwd('C:\\Users\\FBaudron\\Documents\\CIMMYT\\0. Publi\\Bentley et al\\')


# MAPS-------------------------------------------------------------------------

data(world)
world <- subset(world, continent != "Antarctica")

W <-read.csv("wheat.csv")
cor <- read.csv("cor.csv")

W <- W[c(4,6,10,12)]
W[is.na(W)] <- 0

W$Area <- ifelse(W$Area == "China, Hong Kong SAR", "China", W$Area)
W$Area <- ifelse(W$Area == "China, Macao SAR", "China", W$Area)
W$Area <- ifelse(W$Area == "China, mainland", "China", W$Area)
W$Area <- ifelse(W$Area == "China, Taiwan Province of", "China", W$Area)

imp <- W[ which(W$Element == "Import Quantity"), ]
exp <- W[ which(W$Element == "Export Quantity"), ]

imp <- imp[,c(1,4)] %>%
  group_by(Area) %>%
  summarise_each(funs(mean))
```

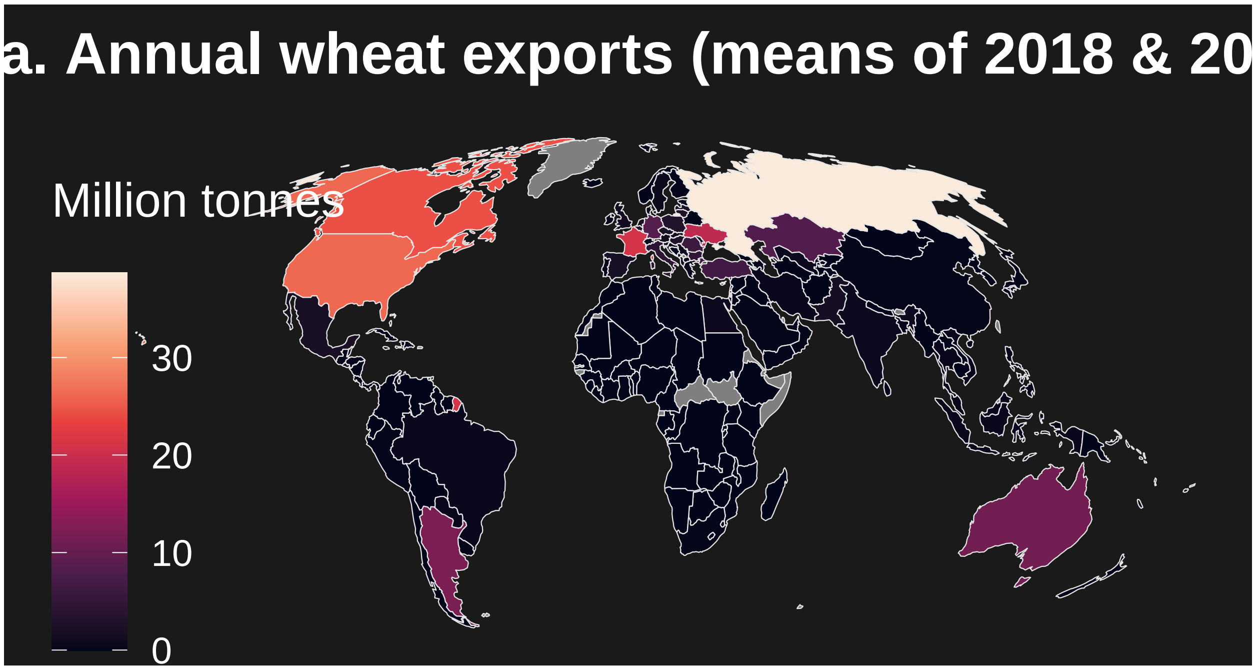
```
## Warning: `summarise_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
exp <- exp[,c(1,4)] %>%
  group_by(Area) %>%
  summarise_each(funs(mean))

imp <- merge(imp, cor, by = "Area", all.x = TRUE)
exp <- merge(exp, cor, by = "Area", all.x = TRUE)

imp <- merge(world, imp, by = "name_long", all.x = TRUE)
exp <- merge(world, exp, by = "name_long", all.x = TRUE)

imp <-
  st_cast(imp, 'MULTIPOLYGON') %>%
  st_transform(crs = "+proj=moll")

exp <-
  st_cast(exp, 'MULTIPOLYGON') %>%
  st_transform(crs = "+proj=moll")


pimp <- ggplot(imp) +
  geom_sf(aes(fill = Value/1000), size = 0.2, color = "grey90", na.rm = TRUE) +
  scale_fill_viridis_c(name="Million tonnes", option = "F") +
  theme_void() +
  labs(title = "b. Annual wheat imports (means of 2018 & 2019)")+
  theme(
    legend.position = c(0.05,0.025), legend.justification = c(0.05,0.025),
    plot.background = element_rect(fill = "grey10", color = NA),
    plot.title = element_text(color = "white", size = 22, face = "bold", hjust=0.05, margin = margin(0,
    legend.title = element_text(size = 18, color = "white", margin = margin(0,0,10,0)),
    legend.text = element_text(size = 14, color = "white", margin = margin(0,0,10,0)),
    legend.key.size = unit(1, "cm"),
    plot.margin = margin(5, 0, 10, 0))

pimp
```

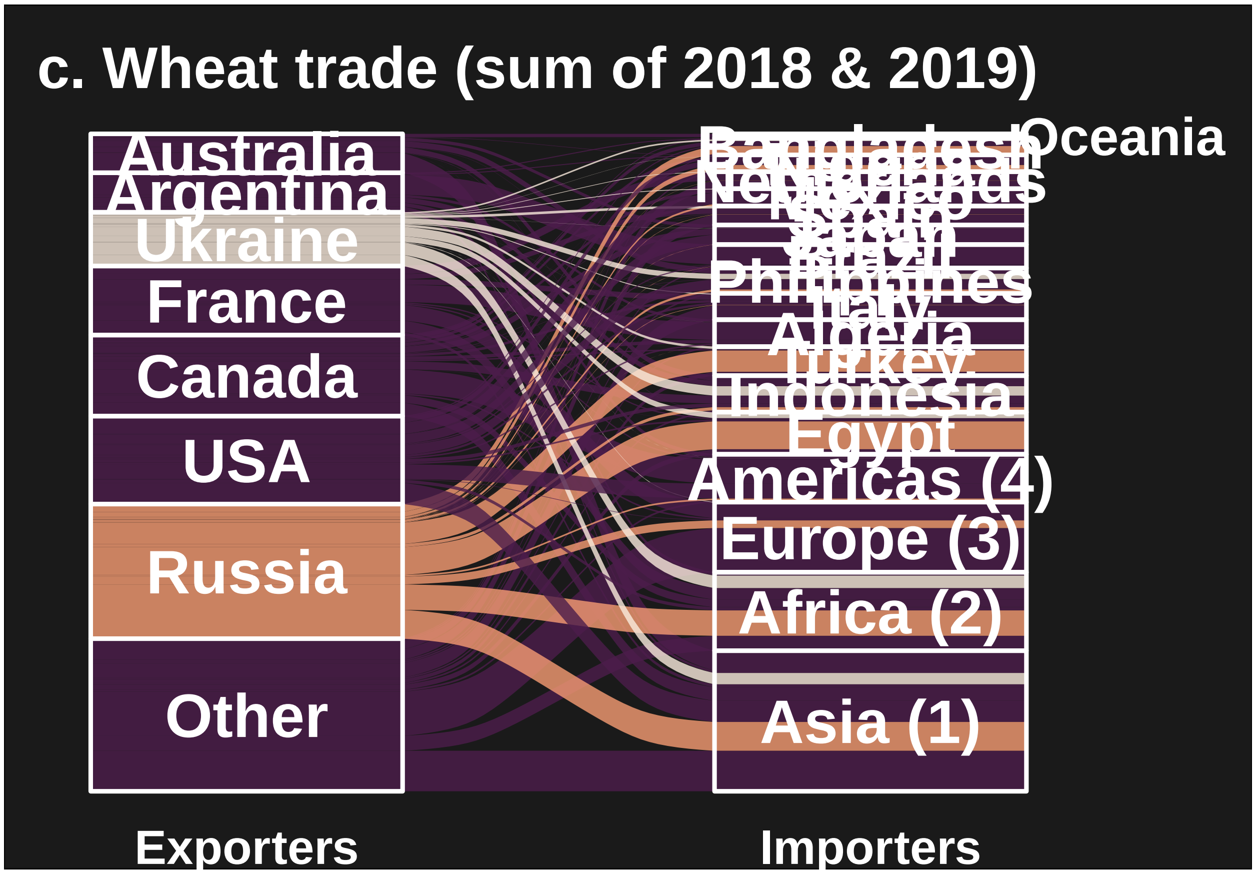


```r
pexp <- ggplot(exp) +
  geom_sf(aes(fill = Value/1000), size = 0.2, color = "grey90", na.rm = TRUE) +
  scale_fill_viridis_c(name="Million tonnes", option = "F") +
  theme_void() +
  labs(title = "a. Annual wheat exports (means of 2018 & 2019)")+
  theme(
    legend.position = c(0.05,0.025), legend.justification = c(0.05,0.025),
    plot.background = element_rect(fill = "grey10", color = NA),
    plot.title = element_text(color = "white", size = 22, face = "bold", hjust=0.05, margin = margin(0,
    legend.title = element_text(size = 18, color = "white", margin = margin(0,0,10,0)),
    legend.text = element_text(size = 14, color = "white", margin = margin(0,0,10,0)),
    legend.key.size = unit(1, "cm"),
    plot.margin = margin(10, 0, 5, 0))

pexp
```



```
# ALLUVIAL DIAGRAMS-----------------------------------------------------------

data <-read.csv("wheat trade matrix.csv")

exp <- read.csv("exporters.csv")
imp <- read.csv("importers.csv")

data <- data[data$Year > 2017,]
data <- data[data$Year < 2020,]
data <- data[ which(data$Element == "Export Quantity"), ]

data <- data[,c(4,6,12,14)]
names(data) <- c("Exporters", "Importers", "Year", "Value")

data <- data[,c(1:2,4)] %>%
  group_by(Exporters, Importers) %>%
  summarise_each(funs(sum))

data <- merge(data, exp, by = "Exporters")
data <- merge(data, imp, by = "Importers")

datimp <- data[,c(1,3)]
datimp <- datimp %>%
  group_by(Importers) %>%
  summarise_each(funs(sum))
```

```r
datexp <- data[,c(2,3)]
datexp <- datexp %>%
  group_by(Exporters) %>%
  summarise_each(funs(sum))

data$Importers2 <- data$Regionimp

data$Importers2 <- ifelse(data$Importers == "Egypt", "Egypt", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Indonesia", "Indonesia", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Turkey", "Turkey", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Algeria", "Algeria", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Italy", "Italy", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Philippines", "Philippines", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Brazil", "Brazil", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Japan", "Japan", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Spain", "Spain", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Mexico", "Mexico", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Netherlands", "Netherlands", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Nigeria", "Nigeria", data$Importers2)
data$Importers2 <- ifelse(data$Importers == "Bangladesh", "Bangladesh", data$Importers2)

data$Importers2 <- ifelse(data$Importers2 == "Asia", "Asia (1)", data$Importers2)
data$Importers2 <- ifelse(data$Importers2 == "Africa", "Africa (2)", data$Importers2)
data$Importers2 <- ifelse(data$Importers2 == "Americas", "Americas (4)", data$Importers2)
data$Importers2 <- ifelse(data$Importers2 == "Europe", "Europe (3)", data$Importers2)
data$Importers2 <- ifelse(data$Importers2 == "Oceania", "", data$Importers2)

data$Exporters2 <- rep("Other", nrow(data))

data$Exporters2 <- ifelse(data$Exporters == "Russian Federation", "Russia", data$Exporters2)
data$Exporters2 <- ifelse(data$Exporters == "United States of America", "USA", data$Exporters2)
data$Exporters2 <- ifelse(data$Exporters == "Canada", "Canada", data$Exporters2)
data$Exporters2 <- ifelse(data$Exporters == "France", "France", data$Exporters2)
data$Exporters2 <- ifelse(data$Exporters == "Ukraine", "Ukraine", data$Exporters2)
data$Exporters2 <- ifelse(data$Exporters == "Argentina", "Argentina", data$Exporters2)
data$Exporters2 <- ifelse(data$Exporters == "Australia", "Australia", data$Exporters2)

dat <- data[,c(3,6:7)] %>%
  group_by(Exporters2, Importers2) %>%
  summarise_each(funs(sum))

theme_set(theme_bw(base_size = 18))

ptrad <- ggplot(dat, aes(y = Value, axis1 = Exporters2, axis2 = Importers2)) +
  theme_void() + ggtitle("c. Wheat trade (sum of 2018 & 2019)") +
  geom_alluvium(aes(fill = Exporters2, weight = Value), decreasing = TRUE, alpha = 0.8, width = 0.5) +
  geom_stratum(decreasing = TRUE, fill = "white", alpha = 0, size = 0.8, colour = "white", width = 0.5)
  ggrepel::geom_text_repel(stat = "stratum", direction = "y", decreasing = TRUE,
                           label = c("","","","","","","","","","","","","","","","","","","","","","",
                           size = 7, fontface = "bold", segment.color = 'white', colour = "white",
                           nudge_x = rep(c(0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1)),
                           hjust = rep(c(0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1))) +
  geom_text(stat = "stratum", fontface = "bold", aes(label = after_stat(stratum)), decreasing = TRUE,
            size = 8, color = "white") +
  scale_x_continuous(breaks = 1:2, labels = c('Exporters', 'Importers'), limits = c(0.70,2.48)) +
  scale_y_continuous(breaks = NULL, labels = NULL) +
  theme(plot.title = element_text(color = "white", hjust = 0.15, size=22, face="bold"),
        axis.text = element_text(color = "white", size=18, face="bold"),
        plot.background = element_rect(fill = "grey10"),
        legend.position = "none",
        plot.margin = margin(10, 10, 0, 0)) +
  scale_fill_manual(values = c("#4C1D4BFF","#4C1D4BFF","#4C1D4BFF","#4C1D4BFF","#4C1D4BFF","#F69C73FF",


ptrad
```



```
# PANEL-------------------------------------------------------------------------

layout <- "
AAAAAAABBBBB
AAAAAAABBBBB
AAAAAAABBBBB
AAAAAAABBBBB
CCCCCCCBBBBB
CCCCCCCBBBBB
CCCCCCCBBBBB
CCCCCCCBBBBB
"
```

```
# FIGURE <- pexp + ptrad + pimp +
#   plot_layout(design = layout) +
#   theme(plot.background = element_rect(fill = "grey10"), plot.margin = margin(20, 20, 20, 20))

# ggdraw(FIGURE) +
#   theme(plot.background = element_rect(fill = "grey10"))

# ggsave("PANEL FINAL - 28 May 2022 1.jpeg", units="cm", width = 60, height = 40, dpi = 320)
```