

Co-financing model code documentation

Itamar Megiddo

02/03/2020

Contents

Setup	2
Clean data	5
Population data	5
Gross domestic product deflator and exchange rate	5
Cost-effectiveness data and GDP per capita	6
Co-financing data	8
Combine CEA and funding data into a tibble	9
Figures	11
All Ghana figure	11
All cross country figure 1 x GDP	15
All cross country figure 0.5 x GDP	19
Ghana 1 x GDP 2012-2016 (most data for these years)	22
Country comparison 1 x GDP per capita	25
GAVI co-financing Ghana	28
Tables	30
Ghana GAVI co-financing table	30
GAVI CEA table	30

Setup

Loading packages.

```
library(tidyverse)
```

```
## -- Attaching packages -----  
  
## v ggplot2 3.2.0      v purrr  0.3.2  
## v tibble  2.1.3      v dplyr  0.8.3  
## v tidyr   0.8.3      v stringr 1.4.0  
## v readr   1.3.1      v forcats 0.4.0  
  
## -- Conflicts -----  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()     masks stats::lag()
```

Load data.

```
gdp_data <- read_csv("data/data_gdp_per_capita.csv")
```

```
## Parsed with column specification:  
## cols(  
##   Country = col_character(),  
##   `2001` = col_double(),  
##   `2002` = col_double(),  
##   `2003` = col_double(),  
##   `2004` = col_number(),  
##   `2005` = col_number(),  
##   `2006` = col_number(),  
##   `2007` = col_number(),  
##   `2008` = col_number(),  
##   `2009` = col_number(),  
##   `2010` = col_number(),  
##   `2011` = col_number(),  
##   `2012` = col_number(),  
##   `2013` = col_number(),  
##   `2014` = col_number(),  
##   `2015` = col_number(),  
##   `2016` = col_number(),  
##   `2017` = col_number(),  
##   `2018` = col_number()  
## )
```

```
cea_data <- read_csv("data/data_cea.csv")
```

```
## Warning: Missing column names filled in: 'X15' [15], 'X16' [16],  
## 'X17' [17], 'X18' [18], 'X19' [19], 'X20' [20], 'X21' [21], 'X22' [22],  
## 'X23' [23], 'X24' [24]
```

```
## Parsed with column specification:
## cols(
##   .default = col_logical(),
##   Country = col_character(),
##   Vaccine = col_character(),
##   `Indicator (e.g. deaths averted)` = col_character(),
##   `Total cost for the period` = col_number(),
##   `Costs per year` = col_number(),
##   `Total benefit for the period` = col_number(),
##   `Benefits per year` = col_number(),
##   `Cost-effectiveness` = col_number(),
##   `Start year` = col_double(),
##   `End year` = col_double(),
##   `Year of dollar rate used` = col_double(),
##   Perspective = col_character(),
##   Assumptions = col_character(),
##   `Author(s)` = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
fund_data <- read_csv("data/data_vaccine_funding.csv")
```

```
## Warning: Missing column names filled in: 'X9' [9], 'X10' [10], 'X11' [11],
## 'X12' [12], 'X13' [13], 'X14' [14], 'X15' [15]
```

```
## Parsed with column specification:
## cols(
##   Country = col_character(),
##   Vaccine = col_character(),
##   Year = col_double(),
##   Total = col_double(),
##   `Domestic contribution ($)` = col_double(),
##   `Gavi contribution ($)` = col_double(),
##   `Domestic (%)` = col_double(),
##   `Gavi (%)` = col_double(),
##   X9 = col_logical(),
##   X10 = col_logical(),
##   X11 = col_logical(),
##   X12 = col_logical(),
##   X13 = col_logical(),
##   X14 = col_logical(),
##   X15 = col_logical()
## )
```

```
gdp_deflator <- read_csv("data/gdp_deflator.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Country Name` = col_character(),
##   `Country Code` = col_character(),
```

```
## `Indicator Name` = col_character(),
## `Indicator Code` = col_character(),
## `2019` = col_logical()
## )
## See spec(...) for full column specifications.
```

```
ex_rate <- read_csv("data/ex_rate.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Country Name` = col_character(),
##   `Country Code` = col_character(),
##   `Indicator Name` = col_character(),
##   `Indicator Code` = col_character(),
##   `2019` = col_logical()
## )
## See spec(...) for full column specifications.
```

```
population_data <- read_csv("data/population_data.csv")
```

```
## Parsed with column specification:
## cols(
##   `Country Name` = col_character(),
##   `2012` = col_double(),
##   `2013` = col_double(),
##   `2014` = col_double(),
##   `2015` = col_double(),
##   `2016` = col_double(),
##   `2017` = col_double(),
##   `2018` = col_double()
## )
```

Set some global values.

```
countries_model <- c("Ghana", "India", "Kenya",
                     "Nigeria", "Angola", "Senegal")
```

Capitlise first letter function (to be used later).

```
firstup <- function(x) {
  substr(x, 1, 1) <- toupper(substr(x, 1, 1))
  x
}
```

Round up nicely for plotting (to be used later).

```
roundUp <- function(x, nice_val=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)) {
  if(length(x) != 1) stop("'x' can only take vectors of length 1")
  10^floor(log10(x)) * nice_val[[which(x <= 10^floor(log10(x)) * nice_val)[[1]]]]
}
```

Clean data

Population data

```
# Change to long format data
population_data <-
  population_data %>%
  rename(country = `Country Name`) %>%
  gather(key = year, value = population, `2012`:`2018`) %>%
  mutate(year = as.numeric(year)) %>%
  filter(country %in% countries_model)
```

Gross domestic product deflator and exchange rate

Clean and combine Gross domestic product (GDP) deflator and exchange rate data.

```
# Clean and remove unnecessary columns
gdp_deflator <- gdp_deflator %>% rename(country = `Country Name`) %>%
  select(country, `2000`:`2018`) %>%
  filter(country %in% countries_model) %>%
  gather(key = year, value = deflator, `2000`:`2018`)

# Add 2018 year to get multiplier of def_2018/def_year
gdp_deflator <- left_join(gdp_deflator,
  gdp_deflator %>% filter(year == 2018) %>% select(-year) %>%
    rename(deflator2018 = deflator),
  by = "country") %>%
  mutate(def_multiplier = deflator2018/deflator) %>%
  select(country, year, def_multiplier)

# Clean exchange rate tibble
ex_rate <- ex_rate %>% rename(country = `Country Name`) %>%
  select(country, `2000`:`2018`) %>%
  filter(country %in% countries_model) %>%
  gather(key = year, value = ex, `2000`:`2018`)

# Add exchange rate multiplier to get ex_2018/ex_year
# multiplying by both multipliers will give
# to local_currency = reported$$ * ex
# to 2018_local_currency = local_currency * def_2018/def
# to 2018_$ = 2018_local_currency / ex_2018
# i.e. (reported$$ * ex * def_2018/def) / ex_2018 =
# reported$$ * def_2018/def * ex/ex_2018
ex_rate <- left_join(ex_rate,
  ex_rate %>% filter(year == 2018) %>% select(-year) %>%
    rename(ex2018 = ex),
  by = "country") %>%
  mutate(ex_multiplier = ex/ex2018) %>%
  select(country, year, ex_multiplier)

# Create currency exchange deflating and exchanging
```

```

currency_conv <- left_join(gdp_deflator, ex_rate, by = c("country", "year")) %>%
  mutate(currency_mult = def_multiplier * ex_multiplier) %>%
  mutate(dollar_year = as.numeric(year)) %>%
  select(country, dollar_year, currency_mult)

```

Cost-effectiveness data and GDP per capita

Clean the cost-effectiveness analysis (CEA) data.

```

cea_data <- cea_data %>%
  # Rename to more data friendly
  rename(country = Country, vaccine = Vaccine,
    indicator = `Indicator (e.g. deaths averted)`,
    cost_for_horizon = `Total cost for the period`,
    cost = `Costs per year`,
    benefit_for_horizon = `Total benefit for the period`,
    benefit = `Benefits per year`, cea = `Cost-effectiveness`,
    s_year = `Start year`, e_year = `End year`,
    dollar_year = `Year of dollar rate used`,
    perspective = Perspective, assumptions = Assumptions,
    authors = `Author(s)` ) %>%
  select(country:authors) %>%
  # Consistent naming of indicators
  mutate(indicator = firstup(indicator)) %>%
  mutate(indicator = gsub("DALY ", "DALYs ", indicator)) %>%
  mutate(indicator = gsub("Death ", "Deaths ", indicator)) %>%
  mutate(indicator = gsub("Case ", "Cases ", indicator)) %>%
  mutate(indicator = gsub("YLL", "YLLs averted", indicator)) %>%
  # Some of the vaccine names differ between fund_data and cea_data
  mutate(vaccine = ifelse(grepl("DTP-Hep", vaccine, ignore.case = TRUE),
    "DTP-hep B",
    vaccine)) %>%
  mutate(vaccine = ifelse(grepl("Yellow Fever", vaccine),
    "Yellow Fever",
    vaccine)) %>%
  mutate(vaccine = ifelse(grepl("Measles", vaccine),
    "Measles",
    vaccine)) %>%
  mutate(vaccine = ifelse(grepl("Pneumococcal", vaccine),
    "PCV",
    vaccine)) %>%
  mutate(vaccine = ifelse(grepl("Rota", vaccine),
    "Rotavirus",
    vaccine)) %>%
  mutate(vaccine = ifelse(grepl("BCG", vaccine),
    "BCG",
    vaccine)) %>%
  # Select only countries in the analysis
  filter(country %in% countries_model)

```

Perform some checks on CEA data.

```
# Check diff between calculated and CEA reported in papers
# Small differences are likely due to rounding errors
```

```
cea_data %>% select(cost, benefit, cea) %>%
  mutate(cea2 = round(cost/benefit,2)) %>%
  mutate(diff = cea - cea2) %>%
  arrange(desc(abs(diff))) %>%
  filter(abs(diff) > 1e-09)
```

```
## # A tibble: 19 x 5
##       cost benefit    cea  cea2    diff
##   <dbl>   <dbl> <dbl> <dbl>   <dbl>
## 1 3689149     303 12196. 12175.  20.1
## 2 1409658     303  4660.  4652.   7.69
## 3  6409006     634 10102. 10109.  -6.37
## 4 11783120     888 13274. 13269.   4.48
## 5 18747118    1303 14392. 14388.   4.01
## 6  3721375     634  5866.  5870.  -3.7
## 7 1505569     311  4844.  4841.   3.11
## 8  7390364     888  8325.  8322.   2.81
## 9 1083447     311  3486.  3484.   2.24
## 10  379910     311  1222.  1222.   0.780
## 11 176198173  10601 16621. 16621.   0.470
## 12 176198173  10601 16621. 16621.   0.470
## 13 34223066   9193  3723.  3723.  -0.160
## 14  7766946   3053  2544.  2544.   0.160
## 15  7766946   3053  2544.  2544.   0.160
## 16 28770592   9193  3129.  3130.  -0.140
## 17 18747118   9228  2031.  2032.  -0.1000
## 18  3200000  12570   255.   255.   0.01
## 19  3721375  33380   111.   111. -0.01000
```

Transform GDP per capita data to long format, convert to \$2018, and add cost & benefit columns.

```
# Max cost +, to create a sequence for plotting
```

```
max_x <- max(cea_data$cost) + 50000000
seq_x <- seq(0, max_x, by = 100000)
```

```
# Convert GDP data to 2018 $
```

```
gdp_data <- gdp_data %>%
  gather(key = "year", value = "gdp_pc", -Country) %>%
  rename(country = Country) %>%
  mutate(year = as.numeric(year)) %>%
  # Convert currency
  left_join(currency_conv %>% rename(year = dollar_year),
    by = c("country", "year")) %>%
  mutate(gdp_pc = gdp_pc * currency_mult) %>%
  select(-currency_mult)
```

```
# Add cost and benefit columns to gdp_data to
```

```
# draw the CEA threshold
```

```
gdp_data <- gdp_data %>%
  # Create rows for each analysis
  slice(rep(row_number(), length(seq_x))) %>%
```

```

# Add a cost column, for plotting cost x-axis
mutate(cost = unlist(sapply(seq_x, function(x) rep(x, nrow(gdp_data)), simplify = FALSE))) %>%
# Add a number of benefit lines for plotting
mutate(benefit_line = cost/gdp_pc,
       benefit_half_line = cost/gdp_pc/0.5,
       benefit_3_line = cost/gdp_pc/3,
       benefit_4_5_line = cost/gdp_pc/4.5)

# Convert to long data tibble
gdp_data <- gdp_data %>%
gather(key = threshold, value = cea_line, benefit_line:benefit_4_5_line) %>%
# Write clear headings for different thresholds
mutate(threshold = ifelse(threshold == "benefit_line",
                        "1 x GDP\nper capita",
                        threshold)) %>%
mutate(threshold = ifelse(threshold == "benefit_half_line",
                        "0.5 x GDP\nper capita",
                        threshold)) %>%
mutate(threshold = ifelse(threshold == "benefit_3_line",
                        "3 x GDP\nper capita",
                        threshold)) %>%
mutate(threshold = ifelse(threshold == "benefit_4_5_line",
                        "4.5 x GDP\nper capita",
                        threshold))

```

Convert the CEA data into 2018 US Dollars using WB deflator.

```

# Combine cea data and currency conversion data
cea_data <- left_join(cea_data, currency_conv, by = c("country", "dollar_year")) %>%
# Convert all costs
mutate(cost_for_horizon = cost_for_horizon * currency_mult,
       cost = cost * currency_mult,
       cea = cea * currency_mult,
       benefit = ifelse(
         grepl("Total healthcare cost averted|Out-of-pocket expenditure",
               indicator),
         benefit * currency_mult,
         benefit))

```

Co-financing data

Clean co-financing data.

```

fund_data <-
fund_data %>%
# Rename columns so they are easier to work with.
rename(country = Country, vaccine = Vaccine, year = Year,
       total = Total,
       domestic = `Domestic contribution ($)` ,
       gavi = `Gavi contribution ($)` ,
       domestic_p = `Domestic (%)` , gavi_p = `Gavi (%)`) %>%
# Remove some columns that are unnecessary

```



```

select(country:gavi_p) %>%
# Remove rows with countries that are not part of the analysis (there are some
# unnecessary rows in the data that explain the data)
filter(country %in% countries_model) %>%
# Convert numbers to numeric
mutate(year = as.numeric(year),
       total = as.numeric(gsub("\\\\", "", total)),
       domestic = as.numeric(gsub("\\\\", "", domestic)),
       gavi = as.numeric(gsub("\\\\", "", gavi)),
       domestic_p = as.numeric(domestic_p)/100,
       gavi_p = as.numeric(gavi_p)/100) %>%
# Convert currency using currency conversion table
left_join(currency_conv %>% rename(year = dollar_year),
          by = c("country", "year")) %>%
mutate(total = total * currency_mult,
       domestic = domestic * currency_mult,
       gavi = gavi * currency_mult) %>%
select(-currency_mult) %>%
# Some of the vaccine names differ between fund_data and cea_data;
# fix.
mutate(vaccine = ifelse(grepl("DTP-Hep", vaccine),
                        "DTP-hep B",
                        vaccine)) %>%
mutate(vaccine = ifelse(grepl("Yellow Fever", vaccine),
                        "Yellow Fever",
                        vaccine)) %>%
mutate(vaccine = ifelse(grepl("Measles", vaccine),
                        "Measles",
                        vaccine)) %>%
mutate(vaccine = ifelse(grepl("Pneumococcal", vaccine),
                        "PCV",
                        vaccine)) %>%
mutate(vaccine = ifelse(grepl("Rota", vaccine),
                        "Rotavirus",
                        vaccine))

```

Combine CEA and funding data into a tibble

```

# Check what vaccines in CEA data are not in co-financing data
temp <- (cea_data %>% distinct(vaccine) %>% unlist) %in%
(fund_data %>% distinct(vaccine) %>% unlist)
(cea_data %>% distinct(vaccine) %>% unlist)[!temp]

```

```

## vaccine3
##      "BCG"

```

```

# Check what vaccines in co-financing data are not in CEA data
temp <- (fund_data %>% distinct(vaccine) %>% unlist) %in%
(cea_data %>% distinct(vaccine) %>% unlist)
(fund_data %>% distinct(vaccine) %>% unlist)[!temp]

```

```
##          vaccine6          vaccine7          vaccine8
##          "MenA" "MR Follow up campaign" "MenA Routine"
##          vaccine9
##          "IPV"
```

```
# Combine CEA and co-financing data by country and vaccine.
# Include all of the vaccines we have in the CEA analysis data,
# but not ones we have only in the co-financing data.
data <- full_join(cea_data,
  fund_data %>% rename(year_gavi = year),
  by = c("country", "vaccine"))
```

```
# Add BCG vaccine, not supported by GAVI for plotting
BCG_data <- data %>%
  # Get BCG data; at this point only includes Levin et al 2007
  filter(vaccine == "BCG") %>%
  # Copy rows years to create years 2006-2018
  slice(rep(row_number(), length(2006:2018))) %>%
  arrange(indicator) %>%
  mutate(year_gavi = rep(2006:2018, 3)) %>%
  # Remove 2015 since do not have data
  filter(year_gavi != 2015)
```

```
# Add the BCG data
data <- data %>%
  # Remove the NA BCG data row
  filter(vaccine != "BCG") %>%
  # Add rest of BCG data
  rbind(BCG_data)
```

Clean combined data.

```
# Remove non base-case
data <- data %>%
  # Include only DALY or YLL
  filter(grepl("DALY|YLL", indicator)) %>%
  # Filter years do not have any GAVI data or only for
  # few countries and based on future projections
  # First add a year for BCG, so plot despite no GAVI
  # funding.
  mutate(year_gavi = ifelse(is.na(year_gavi) &
    authors == "Levin et al. 2007" &
    vaccine == "BCG",
    2007,
    year_gavi)) %>%
  filter(year_gavi < 2019) %>%
  # Only include base-case of Abbot et al. 2012
  # and use earliest year of GAVI funding (since intervention is 2003-08).
  filter(!(authors == "Abbott et al. 2012" &
    (assumptions != "$5 per dose")) %>%# | year_gavi != 2012))) %>%
  # Do not include the CEA with GAVI Subsidy accounted for
  # and focus on health system perspective.
  filter(!(authors == "Nonvignon et al. 2018" &
    (grepl("Ghana only", assumptions) |
```

```

!grepl("Health system", perspective)))) %>%
# Use relevant GAVI data year for Krishnamoorthy et al. 2019
# based on paper intervention years
# filter(!(authors == "Krishnamoorthy et al. 2019" &
# year_gavi != 2018)) %>%
# Use the analysis with the 90% intervention coverage and year 2018
# for GAVI funding based on paper intervention years
filter(!(authors == "Megiddo et al. 2018" &
(grepl("Expanded coverage (90%)", assumptions)))) %>%# /
# year_gavi != 2018))) %>%
# Use most recent year of GAVI funding for Ojal et al. 2019
# based on paper intervention years
#filter(!(authors == "Ojal et al. 2019" & year_gavi < 2016)) %>%
# Do not include the CEA with GAVI subsidy accounted for
filter(!(authors == "Debellut et al., 2019" &
(assumptions == "Vaccine programme costs with Gavi subsidy)))) %>%
# Remove rows with no co-financing data
filter(!(is.na(total) & (!is.na(gavi) | !is.na(domestic)))) %>%
# Add a column with vaccine and paper
mutate(vaccine_cea_paper = paste0(vaccine, "\n", authors)) %>%
# Remove columns that do not need for figures
select(-cost_for_horizon, -benefit_for_horizon, -currency_mult)

```

Figures

All Ghana figure

```

# Add plotting themes.
library(ggthemes)

# Extract Ghana data
data_ghana <- data %>%
  rename(year = year_gavi) %>%
  filter(country == "Ghana" &
    grepl("DALY|YLL", indicator)) %>%
  select(vaccine_cea_paper, indicator, cost, benefit, cea, year, total,
    domestic, gavi, domestic_p, gavi_p, s_year, e_year, vaccine, perspective) %>%
  # Data for 2018 only on BCG which is not supported
  filter(year < 2018)

# Max axes
max_ghana_x <- data_ghana %>%
  summarise(max(cost)) %>% unlist %>%
  roundUp + 10000000
max_ghana_y <- data_ghana %>%
  summarise(max(benefit)) %>% unlist %>%
  roundUp

# Select relevant GDP data (depending on which years have data
# for Ghana)

```

```

gdp_data_ghana <- left_join(data_ghana %>% select(year),
                             gdp_data %>% filter(country == "Ghana"),
                             by = "year") %>%
  # Data for 2018 only on BCG which is not supported
  filter(year < 2018)

# Plot legend labels
legend_labels <- data_ghana %>%
  distinct(vaccine_cea_paper) %>%
  unlist %>% sort

# Plot Ghana
p1 <- ggplot() +
  # Facet by year and threshold
  facet_grid(year~threshold, scales = "free") +
  # Threshold line
  geom_line(data = gdp_data_ghana %>%
             filter(country == "Ghana") %>%
             filter(cost < max_ghana_x),
            mapping = aes(x = cost/1000000, y = cea_line/1000), linetype = 2) +
  # Costs and effectiveness point
  geom_point(data = data_ghana,
             mapping = aes(x = cost/1000000, y = benefit/1000,
                           color = vaccine_cea_paper,
                           shape = vaccine_cea_paper),
             size = 2.5) +
  # Arrow based on proportion of GAVI support relative to domestic
  geom_segment(data = data_ghana,
               aes(x = cost/1000000, xend = (cost - gavi)/1000000, #(cost * domestic_p)/1000000,
                   y = benefit/1000, yend = benefit/1000),
               arrow = arrow(length = unit(0.1, "cm"))) +
  # x-axis name and $ formatting
  scale_x_continuous(name = "Cost (in millions USD 2018)",
                     labels = scales::dollar) +
  # y-axis name and formatting
  scale_y_continuous(name = "DALYs averted (in thousands)",
                     labels = scales::comma) +
  # Remove color/shape legend name
  # Display color legend in two rows and remove shape legend
  # Set colors and shapes
  guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
  scale_color_manual(name = "",
                     labels = legend_labels,
                     values = gdocs_pal()(7)) +

  scale_shape_manual(name = "",
                     labels = legend_labels,
                     values = c(15, 16, 18, 17, 17, 17, 7)) +
  # Change theme of plot
  theme_fivethirtyeight() +
  theme(legend.position = "bottom",

```

```

    axis.title = element_text(),
    plot.caption = element_text(hjust = 0))

# Save figure
ggsave("figures/ghana_all.png", p1, dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")

```

Warning: Removed 44 rows containing missing values (geom_segment).

```

# Create and save figure with appendix caption
p1 +
  # Add a caption
  labs(caption = paste(
    "Figure A1. GAVI contribution to vaccines in Ghana.",
    "Columns differ by cost-effectiveness thresholds, represented by the dashed black line;",
    "Rows differ by the year of GAVI funding (years missing are due to lack of data);",
    "Points represent costs and benefits are extracted from cost-effectiveness (CE) studies;",
    "Arrows represent financing by GAVI (CE studies typically include costs not covered by GAVI",
    "such as distribution costs).",
    "#Arrows represent the proportion of costs funded by GAVI (CE studies and total funding do not match)",
    sep = "\n"))

```

Warning: Removed 44 rows containing missing values (geom_segment).

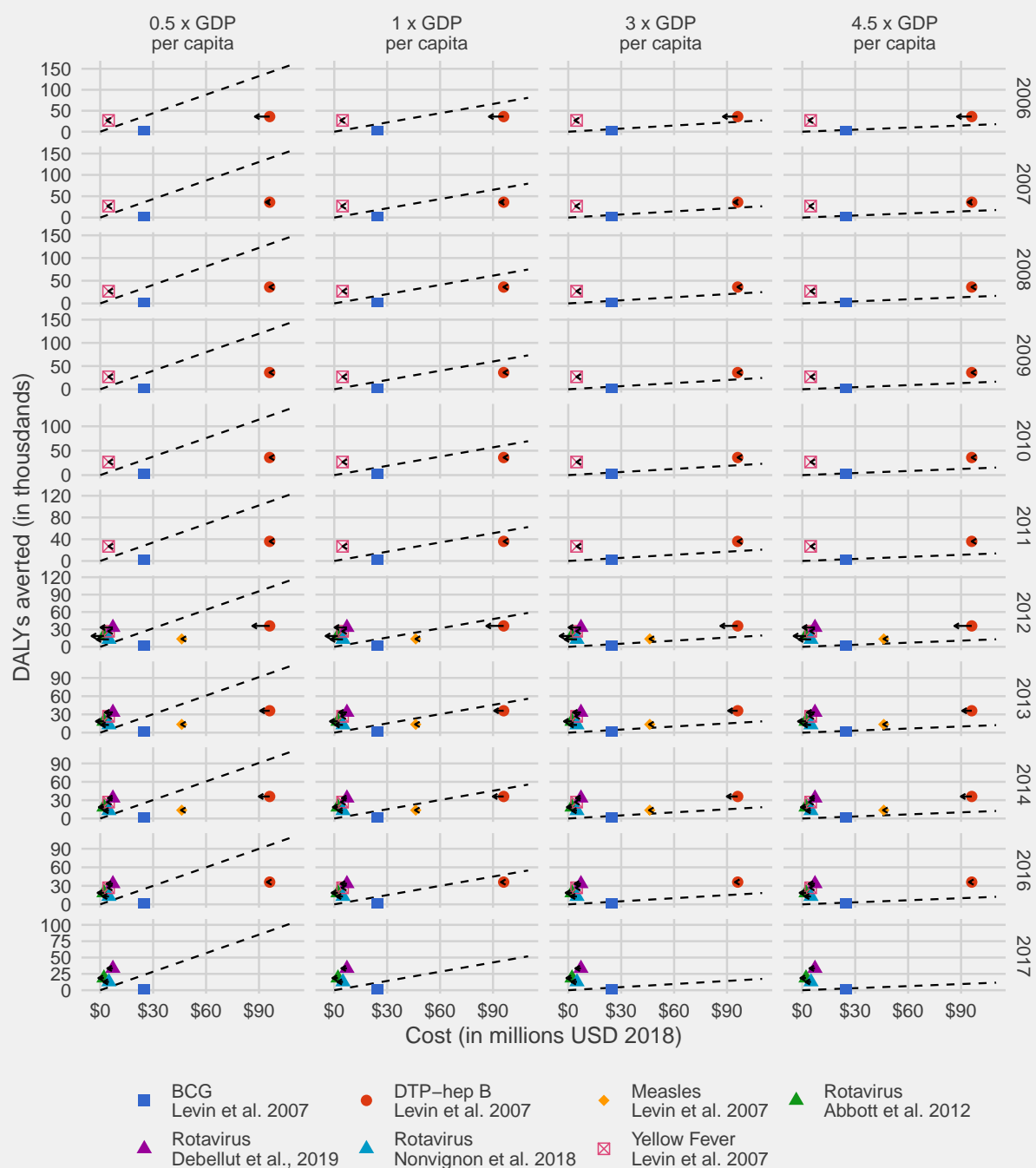


Figure A1. GAVI contribution to vaccines in Ghana.
Columns differ by cost-effectiveness thresholds, represented by the dashed black line;
Rows differ by the year of GAVI funding (years missing are due to lack of data);
Points represent costs and benefits are extracted from cost-effectiveness (CE) studies;
Arrows represent financing by GAVI (CE studies typically include costs not covered by GAVI such as distribution costs).

```
ggsave("figures/A1_ghana_all_appendix_caption.png", dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")
```

```
## Warning: Removed 44 rows containing missing values (geom_segment).
```

All cross country figure 1 x GDP

```
# Get years that include non Ghana countries
years_include <- data %>%
  filter(country != "Ghana") %>% distinct(year_gavi) %>% unlist

# All country data
data_all_countries <-
  data %>%
  rename(year = year_gavi) %>%
  filter(grepl("DALY|YLL", indicator)) %>%
  filter(year %in% years_include) %>%
  select(country, vaccine_cea_paper, indicator, cost, benefit, cea, year, total,
         domestic, gavi, domestic_p, gavi_p, s_year, e_year, vaccine, perspective) %>%
  # Data for 2018 only on BCG which is not supported
  # Also remove 2015 for which there is very limited data
  filter(year < 2018 & year != 2015) %>%
  # Add population data
  left_join(population_data, by = c("country", "year")) %>%
  # Convert values to per 100,000 population
  mutate(cost = cost/population*100000,
         benefit = benefit/population*100000,
         gavi = gavi/population*100000,
         domestic = domestic/population*100000)

# Max axes
max_all_countries_x <- data_all_countries %>%
  summarise(max(cost)) %>% unlist %>%
  roundUp + 10
max_all_countries_y <- data_all_countries %>%
  summarise(max(benefit)) %>% unlist %>%
  roundUp

# GDP data select
gdp_data_all_countries <-
  left_join(data_all_countries %>% select(year, country),
            gdp_data,
            by = c("year", "country")) %>%
  filter(year %in% years_include) %>%
  # Data for 2018 only on BCG which is not supported
  # Also remove 2015 for which there is very limited data
  filter(year < 2018 & year != 2015)

# Plot legend labels
legend_labels <- data_all_countries %>%
  distinct(vaccine_cea_paper) %>%
  unlist %>% sort

# Plot all countries using 1 X GDP per capita threshold
p2 <- ggplot() +
  # Facet by country and year
  facet_grid(year~country, scales = "free") +
  # Threshold line
```

```

geom_line(data = gdp_data_all_countries %>%
  filter(threshold == "1 x GDP\nper capita") %>%
  filter(cost < max_all_countries_x),
  mapping = aes(x = cost/1000, y = cea_line), linetype = 2) +
# Costs and effectiveness point
geom_point(data = data_all_countries,
  mapping = aes(x = cost/1000, y = benefit,
    color = vaccine_cea_paper,
    shape = vaccine_cea_paper),
  size = 2.5) +
# Arrow based on proportion of GAVI support relative to domestic
geom_segment(data = data_all_countries,
  aes(x = cost/1000, xend = (cost - gavi)/1000, #(cost/1000 * domestic_p),
    y = benefit, yend = benefit),
  arrow = arrow(length = unit(0.1, "cm")))) +
# x-axis name and $ formatting
scale_x_continuous(name = "Cost per 100,000 people (in thousands USD 2018)",
  labels = scales::dollar) +
# y-axis name and formatting
scale_y_continuous(name = "DALYs averted per 100,000 people",
  labels = scales::comma) +
# Remove color/shape legend name
# Display color legend in two rows and remove shape legend
# Set colors and shapes
guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
scale_color_manual(name = "",
  labels = legend_labels,
  values = gdocs_pal()(10)) +

scale_shape_manual(name = "",
  labels = legend_labels,
  values = c(15, 16, 18, 13, 13, 13, 17, 17, 17, 7)) +
# Display color legend in 4 columns
# Change theme of plot
theme_fivethirtyeight() +
theme(legend.position = "bottom",
  axis.title = element_text(),
  axis.text.x = element_text(angle = 45),
  plot.caption = element_text(hjust = 0))

# Save figure
ggsave("figures/all_country_compare_1GDP.png", p2, dpi = 300,
  width = 8*2.5, height = 10*2.5, units = "cm")

```

Warning: Removed 5 rows containing missing values (geom_segment).

```

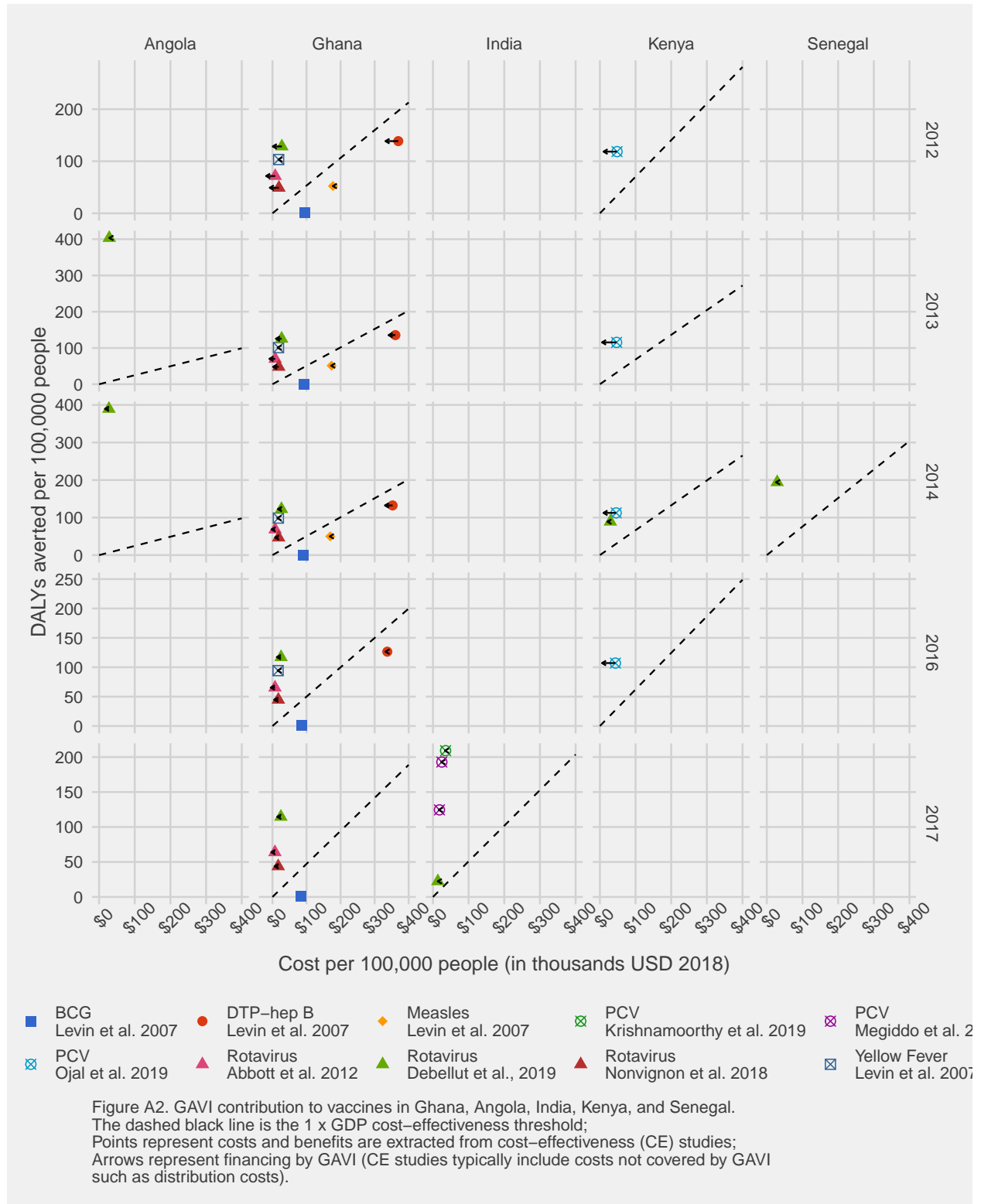
p2 + labs(
  caption = paste(
    "Figure A2. GAVI contribution to vaccines in Ghana, Angola, India, Kenya, and Senegal.",
    "The dashed black line is the 1 x GDP cost-effectiveness threshold;",
    "Points represent costs and benefits are extracted from cost-effectiveness (CE) studies;",
    "Arrows represent financing by GAVI (CE studies typically include costs not covered by GAVI",
    "such as distribution costs).",

```



```
#"Arrows represent the proportion of costs funded by GAVI (CE studies and total funding do not match)  
sep = "\n"))
```

```
## Warning: Removed 5 rows containing missing values (geom_segment).
```



```
# Create and save figure with appendix caption
ggsave("figures/A2_all_country_compare_1GDP.png", dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")
```

Warning: Removed 5 rows containing missing values (geom_segment).

All cross country figure 0.5 x GDP

```
# Plot all countries using 1 X GDP per capita threshold
p3 <- ggplot() +
  # Facet by country and year
  facet_grid(year~country, scales = "free") +
  # Threshold line
  geom_line(data = gdp_data_all_countries %>%
    filter(threshold == "0.5 x GDP\nper capita") %>%
    filter(cost < max_all_countries_x),
    mapping = aes(x = cost/1000, y = cea_line), linetype = 2) +
  # Costs and effectiveness point
  geom_point(data = data_all_countries,
    mapping = aes(x = cost/1000, y = benefit,
      color = vaccine_cea_paper,
      shape = vaccine_cea_paper),
    size = 2.5) +
  # Arrow based on proportion of GAVI support relative to domestic
  geom_segment(data = data_all_countries,
    aes(x = cost/1000, xend = (cost - gavi)/1000, #(cost/1000 * domestic_p),
      y = benefit, yend = benefit),
    arrow = arrow(length = unit(0.1, "cm")))) +
  # x-axis name and $ formatting
  scale_x_continuous(name = "Cost per 100,000 people (in thousands USD 2018)",
    labels = scales::dollar) +
  # y-axis name and formatting
  scale_y_continuous(name = "DALYs averted per 100,000 people",
    labels = scales::comma) +
  # Remove color/shape legend name
  # Display color legend in four rows and remove shape legend
  # Set colors and shapes
  guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
  scale_color_manual(name = "",
    labels = legend_labels,
    values = gdocs_pal()(10)) +

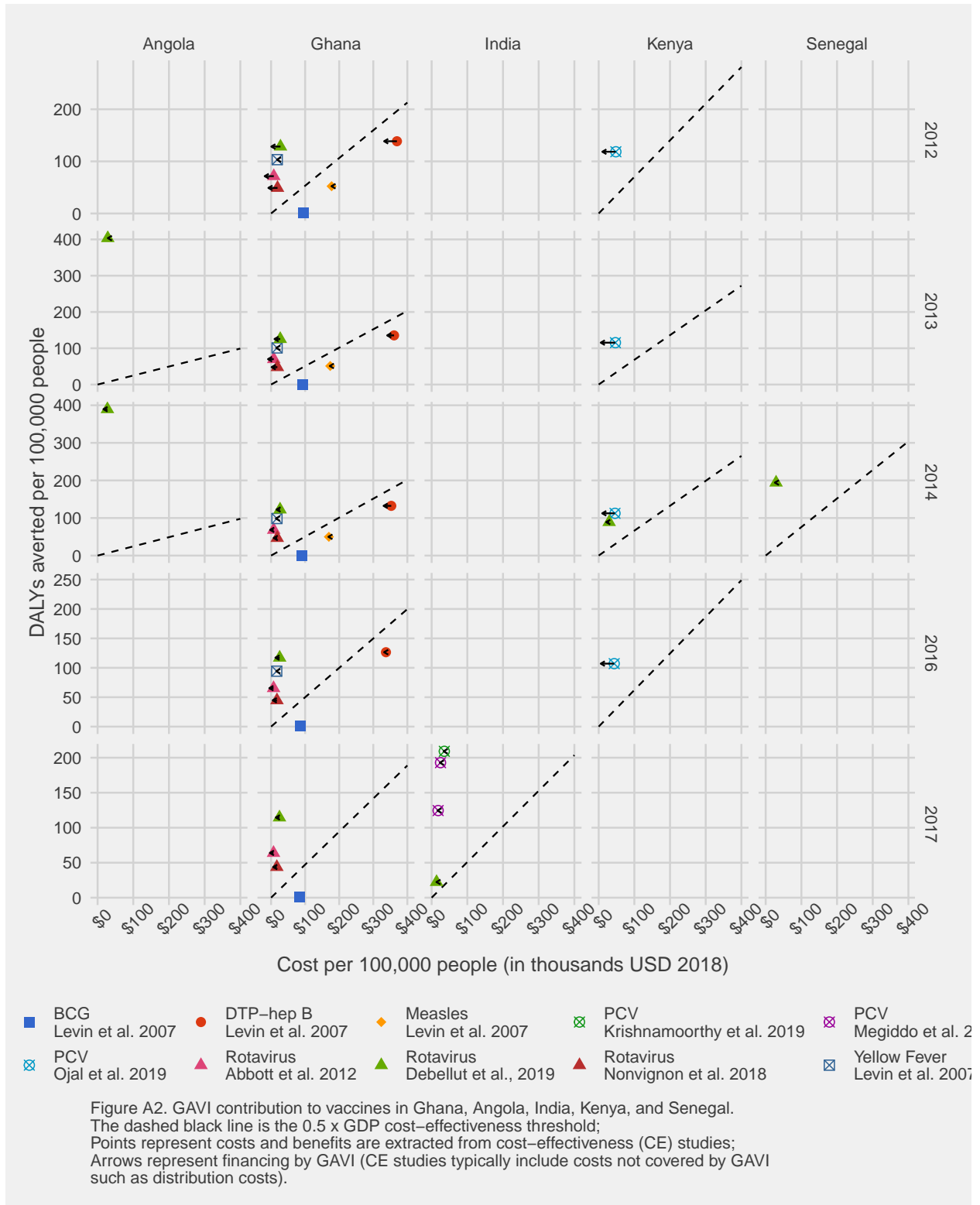
  scale_shape_manual(name = "",
    labels = legend_labels,
    values = c(15, 16, 18, 13, 13, 13, 17, 17, 17, 7)) +
  # Display color legend in 4 columns
  # Change theme of plot
  theme_fivethirtyeight() +
  theme(legend.position = "bottom",
    axis.title = element_text(),
    axis.text.x = element_text(angle = 45),
    plot.caption = element_text(hjust = 0))

# Save figure
ggsave("figures/all_country_compare_05GDP.png", p2, dpi = 300,
  width = 8*2.5, height = 10*2.5, units = "cm")
```

```
## Warning: Removed 5 rows containing missing values (geom_segment).
```

```
p2 + labs(  
  caption = paste(  
    "Figure A2. GAVI contribution to vaccines in Ghana, Angola, India, Kenya, and Senegal.",  
    "The dashed black line is the 0.5 x GDP cost-effectiveness threshold;",  
    "Points represent costs and benefits are extracted from cost-effectiveness (CE) studies;",  
    "Arrows represent financing by GAVI (CE studies typically include costs not covered by GAVI",  
    "such as distribution costs).",  
    "Arrows represent the proportion of costs funded by GAVI (CE studies and total funding do not match)",  
    sep = "\n"))
```

```
## Warning: Removed 5 rows containing missing values (geom_segment).
```



```
# Create and save figure with appendix caption
ggsave("figures/A2_all_country_compare_05GDP.png", dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")
```

```
## Warning: Removed 5 rows containing missing values (geom_segment).
```

Ghana 1 x GDP 2012-2016 (most data for these years)

```
# Plot legend labels
legend_labels <- data_ghana %>%
  distinct(vaccine_cea_paper) %>%
  unlist %>% sort

# Plot Ghana
p4 <- ggplot() +
  # Facet by year and threshold
  facet_wrap(year~., scales = "free", ncol = 2) +
  # Threshold line
  geom_line(data = gdp_data_ghana %>%
    filter(country == "Ghana") %>%
    filter(cost < max_ghana_x) %>%
    filter(threshold == "1 x GDP\ntper capita") %>%
    filter(year %in% 2012:2016),
    mapping = aes(x = cost/1000000, y = cea_line/1000), linetype = 2) +
  # Costs and effectiveness point
  geom_point(data = data_ghana %>%
    filter(year %in% 2012:2016),
    mapping = aes(x = cost/1000000, y = benefit/1000,
                  color = vaccine_cea_paper,
                  shape = vaccine_cea_paper),
    size = 2.5) +
  # Arrow based on proportion of GAVI support relative to domestic
  geom_segment(data = data_ghana %>%
    filter(year %in% 2012:2016),
    aes(x = cost/1000000, xend = (cost - gavi)/1000000, #(cost * domestic_p)/1000000,
        y = benefit/1000, yend = benefit/1000),
    arrow = arrow(length = unit(0.1, "cm"))) +
  # x-axis name and $ formatting
  scale_x_continuous(name = "Cost (in millions USD 2018)",
    labels = scales::dollar) +
  # y-axis name and formatting
  scale_y_continuous(name = "DALYs averted (in thousands)",
    labels = scales::comma) +
  # Remove color/shape legend name
  # Display color legend in two rows and remove shape legend
  # Set colors and shapes
  guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
  scale_color_manual(name = "",
    labels = legend_labels,
    values = gdocs_pal()(7)) +
  scale_shape_manual(name = "",
    labels = legend_labels,
    values = c(15, 16, 18, 17, 17, 17, 7)) +
  # Change theme of plot
  theme_fivethirtyeight() +
  theme(legend.position = "bottom",
```

```

axis.title = element_text(),
plot.caption = element_text(hjust = 0))

# Save figure
ggsave("figures/ghana_x1gdp.png", p4, dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")

```

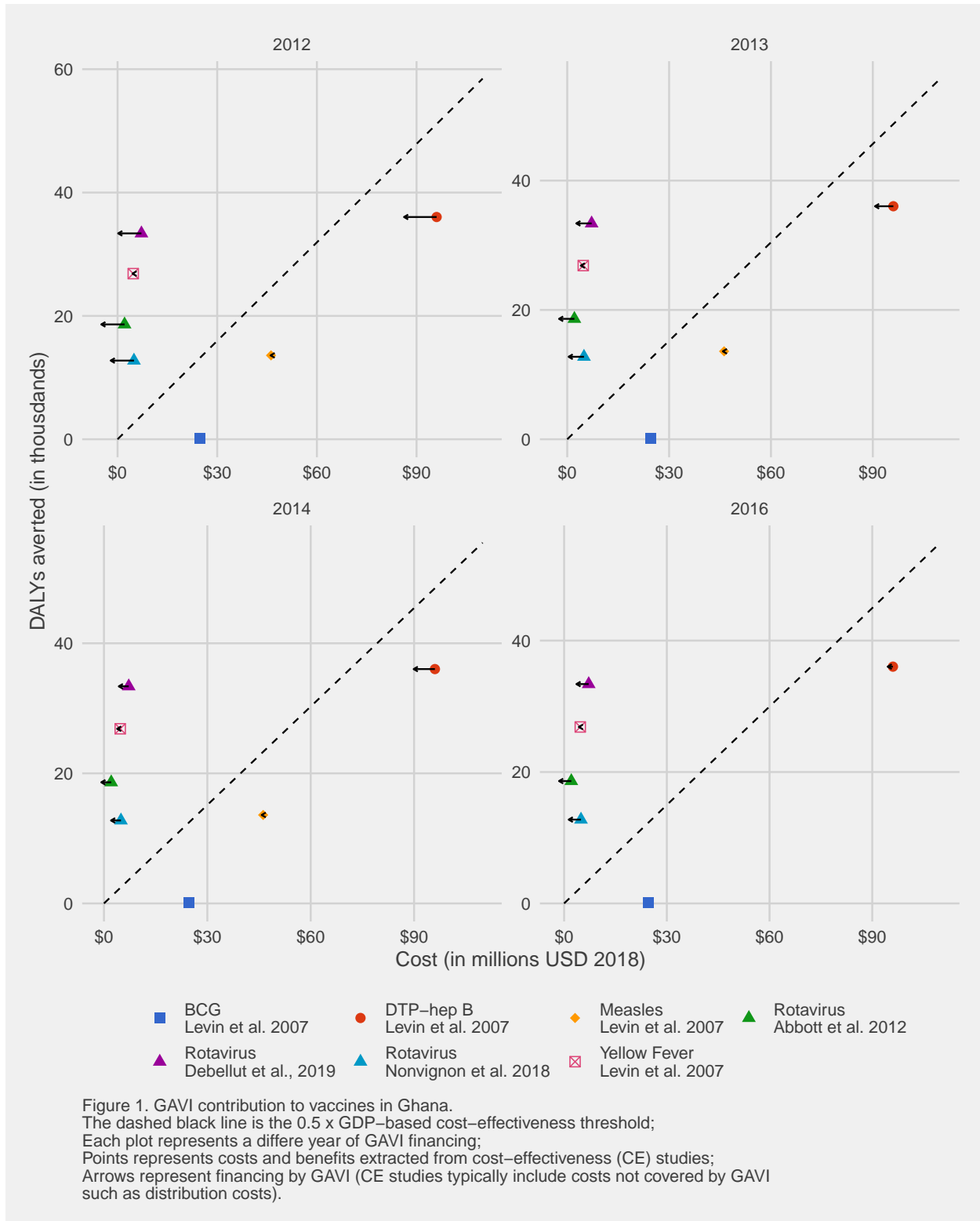
Warning: Removed 4 rows containing missing values (geom_segment).

```

# Create and save figure with appendix caption
p4 +
  # Add a caption
  labs(caption = paste(
    "Figure 1. GAVI contribution to vaccines in Ghana.",
    "The dashed black line is the 0.5 x GDP-based cost-effectiveness threshold;",
    "Each plot represents a differe year of GAVI financing;",
    "Points represents costs and benefits extracted from cost-effectiveness (CE) studies;",
    "Arrows represent financing by GAVI (CE studies typically include costs not covered by GAVI",
    "such as distribution costs).",
    #"Arrows represent the proportion of costs funded by GAVI (CE studies and total funding do not match)",
    sep = "\n"))

```

Warning: Removed 4 rows containing missing values (geom_segment).



```
ggsave("figures/A3_ghana_xlgdp_appendix_caption.png", dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")
```

```
## Warning: Removed 4 rows containing missing values (geom_segment).
```


Country comparison 1 x GDP per capita

```
# Plot legend labels
legend_labels <- data_all_countries %>%
  distinct(vaccine_cea_paper) %>%
  unlist %>% sort

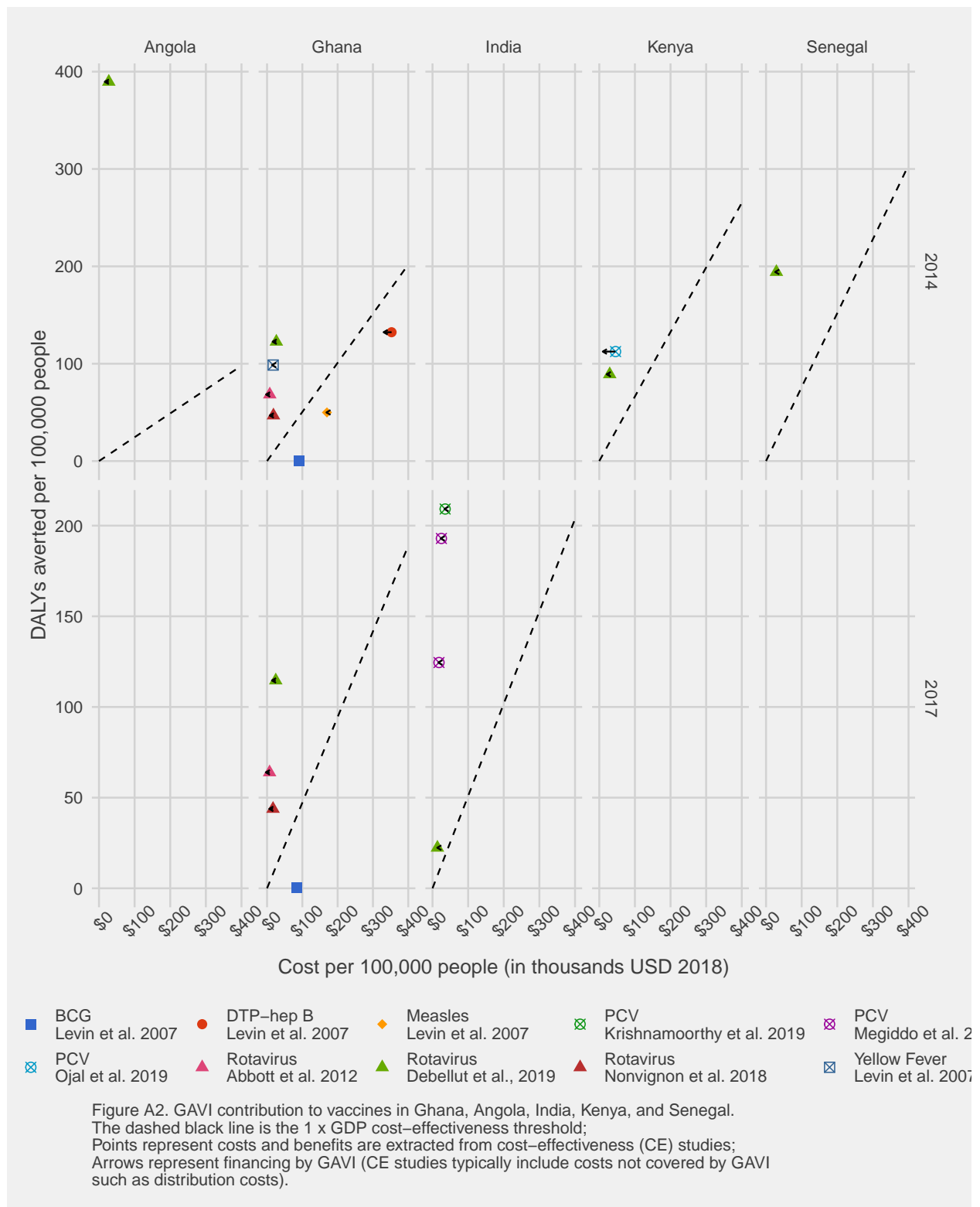
# Plot all countries using 1 X GDP per capita threshold
p5 <- ggplot() +
  # Facet by country and year
  facet_grid(year~country, scales = "free") +
  # Threshold line
  geom_line(data = gdp_data_all_countries %>%
    filter(threshold == "1 x GDP\nper capita") %>%
    filter(cost < max_all_countries_x) %>%
    filter(year %in% c(2014, 2017)),
    mapping = aes(x = cost/1000, y = cea_line), linetype = 2) +
  # Costs and effectiveness point
  geom_point(data = data_all_countries %>%
    filter(year %in% c(2014, 2017)),
    mapping = aes(x = cost/1000, y = benefit,
      color = vaccine_cea_paper,
      shape = vaccine_cea_paper),
    size = 2.5) +
  # Arrow based on proportion of GAVI support relative to domestic
  geom_segment(data = data_all_countries %>%
    filter(year %in% c(2014, 2017)),
    aes(x = cost/1000, xend = (cost - gavi)/1000, #(cost/1000 * domestic_p),
      y = benefit, yend = benefit),
    arrow = arrow(length = unit(0.1, "cm")))) +
  # x-axis name and $ formatting
  scale_x_continuous(name = "Cost per 100,000 people (in thousands USD 2018)",
    labels = scales::dollar) +
  # y-axis name and formatting
  scale_y_continuous(name = "DALYs averted per 100,000 people",
    labels = scales::comma) +
  # Remove color/shape legend name
  # Display color legend in four rows and remove shape legend
  # Set colors and shapes
  guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
  scale_color_manual(name = "",
    labels = legend_labels,
    values = gdocs_pal()(10)) +
  scale_shape_manual(name = "",
    labels = legend_labels,
    values = c(15, 16, 18, 13, 13, 13, 17, 17, 17, 7)) +
  # Display color legend in 4 columns
  # Change theme of plot
  theme_fivethirtyeight() +
  theme(legend.position = "bottom",
    axis.title = element_text(),
    axis.text.x = element_text(angle = 45),
    plot.caption = element_text(hjust = 0))
```

```
# Save figure
ggsave("figures/country_compare_1GDP.png", p5, dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")
```

```
## Warning: Removed 2 rows containing missing values (geom_segment).
```

```
p5 + labs(
  caption = paste(
    "Figure A2. GAVI contribution to vaccines in Ghana, Angola, India, Kenya, and Senegal.",
    "The dashed black line is the 1 x GDP cost-effectiveness threshold;",
    "Points represent costs and benefits are extracted from cost-effectiveness (CE) studies;",
    "Arrows represent financing by GAVI (CE studies typically include costs not covered by GAVI",
    "such as distribution costs).",
    #"Arrows represent the proportion of costs funded by GAVI (CE studies and total funding do not match)",
    sep = "\n"))
```

```
## Warning: Removed 2 rows containing missing values (geom_segment).
```

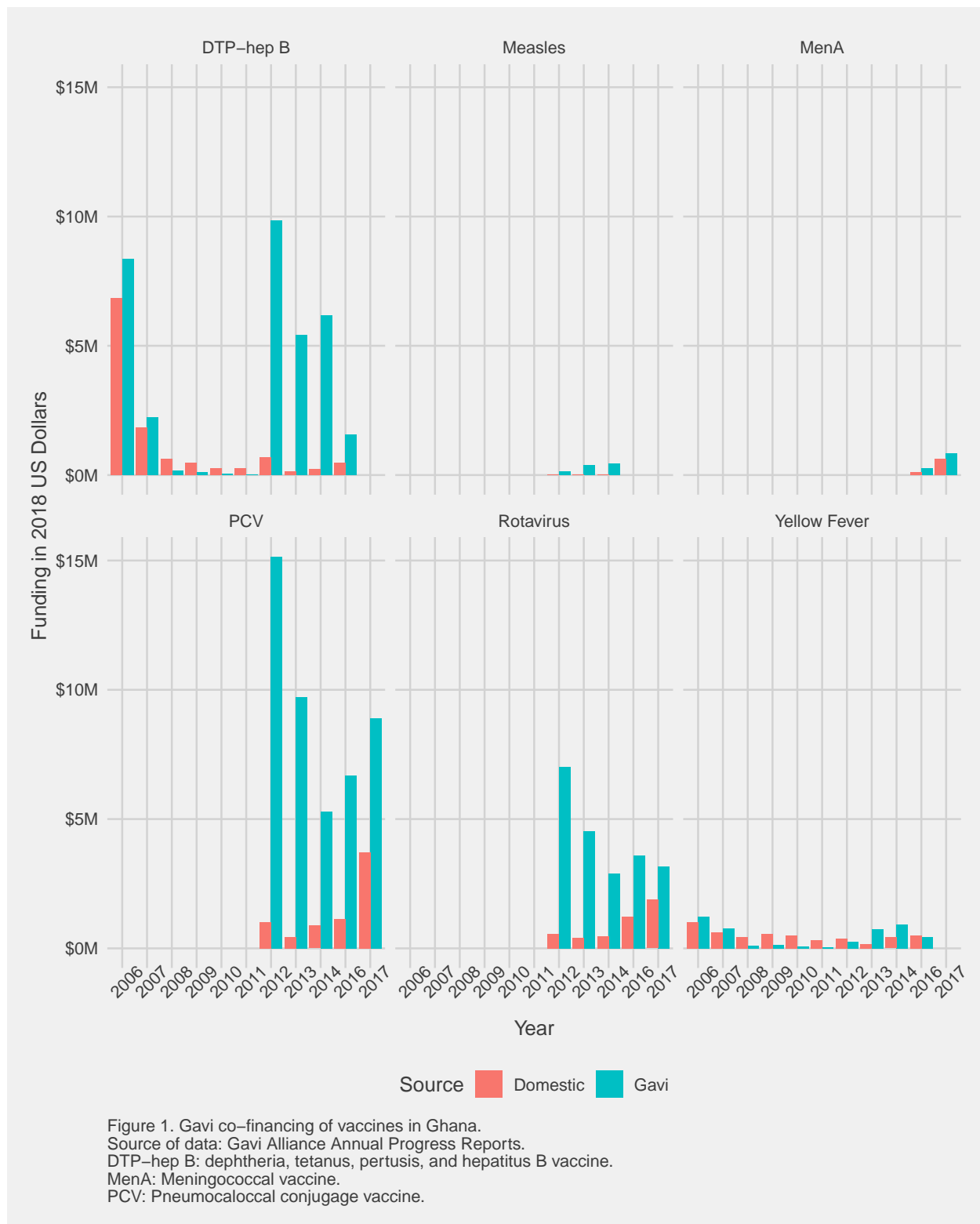


```
# Create and save figure with appendix caption
ggsave("figures/A5_country_compare_1GDP.png", dpi = 300,
       width = 8*2.5, height = 10*2.5, units = "cm")
```

```
## Warning: Removed 2 rows containing missing values (geom_segment).
```

GAVI co-financing Ghana

```
financing_ghana_plot_data <-  
  fund_data %>%  
  filter(country == "Ghana" & !is.na(gavi)) %>%  
  select(vaccine, year, domestic, gavi) %>%  
  gather(key = source, value = amount, domestic:gavi) %>%  
  mutate(source = gsub("gavi", "Gavi", source)) %>%  
  mutate(source = gsub("domestic", "Domestic", source))  
  
p6 <- ggplot(data = financing_ghana_plot_data,  
  mapping = aes(x = factor(year), y = amount/1000000, fill = source)) +  
  facet_wrap(vaccine~.) +  
  geom_bar(stat = "identity", position = "dodge") +  
  # x-axis name and $ formatting  
  scale_x_discrete(name = "Year") +  
  # y-axis name and formatting  
  scale_y_continuous(name = "Funding in 2018 US Dollars",  
    labels = scales::dollar_format(prefix = "$", suffix = "M")) +  
  # Remove fill legend title  
  # Display color legend in four rows and remove shape legend  
  # Set colors and shapes  
  scale_fill_discrete(name = "Source") +  
  # Change theme of plot  
  theme_fivethirtyeight() +  
  theme(legend.position = "bottom",  
    axis.title = element_text(),  
    plot.caption = element_text(hjust = 0),  
    axis.text.x = element_text(angle = 45))  
  
# Save figure  
ggsave("figures/ghana_gavi_cofinancing.png", p6, dpi = 300,  
  width = 6*2.5, height = 6*2.5, units = "cm")  
  
p6 + labs(  
  caption = paste(  
    "Figure 1. Gavi co-financing of vaccines in Ghana.",  
    "Source of data: Gavi Alliance Annual Progress Reports.",  
    "DTP-hep B: diphtheria, tetanus, pertussis, and hepatitis B vaccine.",  
    "MenA: Meningococcal vaccine.",  
    "PCV: Pneumococcal conjugate vaccine.",  
    sep = "\n"))
```



```
# Create and save figure with appendix caption
ggsave("figures/A6_ghana_gavi_cofinancing.png.png", dpi = 300,
       width = 9*2.5, height = 6*2.5, units = "cm")
```

Tables

Ghana GAVI co-financing table

```
table_out <- fund_data %>%
  filter(country == "Ghana") %>%
  filter(!is.na(gavi)) %>%
  mutate(domestic = paste(scales::dollar(signif(domestic, 3)),
    " (", scales::percent(domestic_p, accuracy = 1), "%)",
    sep = ""),
    gavi = paste(scales::dollar(signif(gavi, 3)),
    " (", scales::percent(gavi_p, accuracy = 1), "%)",
    sep = ""),
    total = scales::dollar(signif(total, 3))) %>%
  arrange(vaccine, year) %>%
  select(Vaccine = vaccine, Year = year,
    `Domestic support (%)` = domestic,
    `Gavi support (%)` = gavi, Total = total)
write_csv(table_out, "tables/csv_format/ghana_gavi_cofinancing.csv")
```

GAVI CEA table

```
table_out <-
  cea_data %>%
  filter(grepl("DALY|YLL", indicator)) %>%
  # Only include base-case of Abbot et al. 2012
  # and use earliest year of GAVI funding (since intervention is 2003-08).
  filter(!(authors == "Abbott et al. 2012" &
    (assumptions != "$5 per dose"))) %>% # | year_gavi != 2012))) %>%
  # Do not include the CEA with GAVI Subsidy accounted for
  # and focus on health system perspective.
  filter(!(authors == "Nonvignon et al. 2018" &
    (grepl("Ghana only", assumptions) |
    !grepl("Health system", perspective)))) %>%
  # Use relevant GAVI data year for Krishnamoorthy et al. 2019
  # based on paper intervention years
  # filter(!(authors == "Krishnamoorthy et al. 2019" &
  #   year_gavi != 2018)) %>%
  # Use the analysis with the 90% intervention coverage and year 2018
  # for GAVI funding based on paper intervention years
  filter(!(authors == "Megidido et al. 2018" &
    (grepl("Expanded coverage (90%)", assumptions)))) %>% # |
  #   year_gavi != 2018))) %>%
  # Use most recent year of GAVI funding for Ojal et al. 2019
  # based on paper intervention years
  #filter(!(authors == "Ojal et al. 2019" & year_gavi < 2016)) %>%
  # Do not include the CEA with GAVI subsidy accounted for
  filter(!(authors == "Debellut et al., 2019" &
    (assumptions == "Vaccine programme costs with Gavi subsidy"))) %>%
  # Remove comma from authors column
```

```

mutate(authors = gsub(",", "", authors)) %>%
# Convert to dollars
mutate(cost = scales::dollar(signif(cost, 3)),
       benefit = scales::comma(signif(benefit, 3)),
       cea = scales::dollar(signif(cea, 3)),
       year = as.numeric(gsub(".*\\.", "", authors))) %>%
arrange(country, vaccine, year) %>%
select(Country = country, Vaccine = vaccine, Indicator = indicator,
       Costs = cost, Benefits = benefit, CER = cea, Perspective = perspective,
       Source = authors)
write_csv(table_out, "tables/csv_format/cea_table.csv")

```