

Joining data from multiple sources

Prof. Maria Tackett

Click for PDF of slides

Working with multiple data frames

Fisheries of the world

Fisheries and Aquaculture Department of the Food and Agriculture Organization of the United Nations collects data on fisheries production of countries.

Country	Capture	Aquaculture	Total
 China	17,800,000	63,700,000	81,500,000
 Indonesia	6,584,419	16,600,000	23,200,000
 India	5,082,332	5,703,002	10,800,000
 Vietnam	2,785,940	3,634,531	6,420,471
 United States	4,931,017	444,369	5,375,386
 Russia	4,773,413	173,840	4,947,253
 Japan	3,275,263	1,067,994	4,343,257
 Philippines	2,027,992	2,200,914	4,228,906
 Peru	3,811,802	100,187	3,911,989
 Bangladesh	1,674,770	2,203,554	3,878,324

Source: https://en.wikipedia.org/wiki/Fishing_industry_by_country

Load data

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

First look at the data

```
glimpse(fisheries)
```

```
## Rows: 216
```

```
## Columns: 4
```

```
## $ country      <chr> "Afghanistan", "Albania", "Algeria", "American Samoa",
```

```
## $ capture      <dbl> 1000, 7886, 95000, 3047, 0, 486490, 3000, 755226, 3758
```

```
## $ aquaculture  <dbl> 1200, 950, 1361, 20, 0, 655, 10, 3673, 16381, 0, 96847
```

```
## $ total        <dbl> 2200, 8836, 96361, 3067, 0, 487145, 3010, 758899, 20139
```

Quick summaries of the data

```
skim(fisheries) #skimr package
```

```
## — Data Summary —————
```

```
##                               Values
## Name                          fisheries
## Number of rows                 216
## Number of columns              4
```

```
## -----
```

```
## Column type frequency:
```

```
##   character                    1
```

```
##   numeric                      3
```

```
## -----
```

```
## Group variables                None
```

```
##
```

```
## — Variable type: character —————
```

```
##   skim_variable n_missing complete_rate  min  max empty n_unique whitespace
```

```
## 1 country           0             1    4   32    0    215           0
```

```
##
```

```
## — Variable type: numeric —————
```

```
##   skim_variable n_missing complete_rate  mean      sd    p0    p25    p50    p75    p100 hist
```

```
## 1 capture           0             1 421916. 1478638.    0 3280. 33797 221884. 17800000 █
```

```
## 2 aquaculture        0             1 508368. 4496073.    0  25.2 1574. 25998 63700000 █
```

```
## 3 total              0             1 930284. 5846301.    0 7270. 44648. 271901. 81500000 █
```

Some summary stats

```
fisheries %>%  
  summarise(  
    mean_cap = mean(capture),  
    mean_aqc = mean(aquaculture),  
    mean_tot = mean(total)  
  )
```

```
## # A tibble: 1 x 3  
##   mean_cap mean_aqc mean_tot  
##   <dbl>    <dbl>    <dbl>  
## 1  421916.  508368.  930284.
```

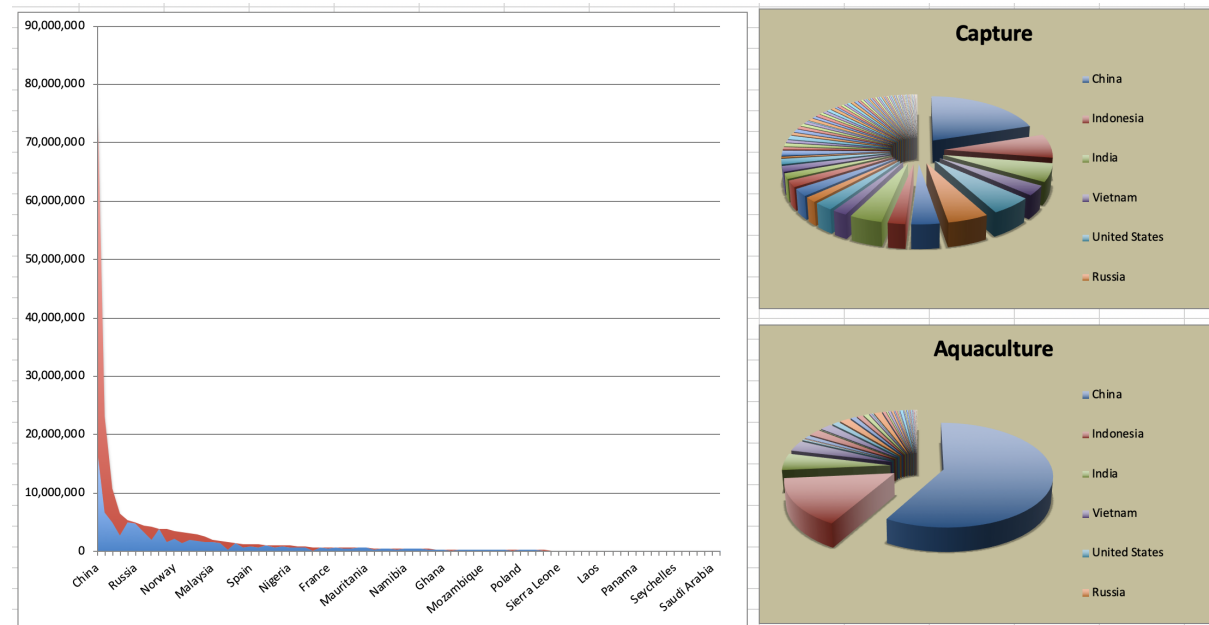
well, that was boring...

A new approach!

```
fisheries %>%  
  summarise(across(capture:total, mean))
```

```
## # A tibble: 1 x 3  
##   capture aquaculture total  
##   <dbl>      <dbl> <dbl>  
## 1 421916.    508368. 930284.
```

The (not-so-great) visualization below shows the distribution of fishery harvest of countries for 2016, by capture and aquaculture. What are some ways you would improve this visualization? Note that countries whose total harvest was less than 100,000 tons are not included in the visualization.



Goal: calculate summary statistics at the continent level and visualize them

Data prep

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

Filter out countries whose total harvest was less than 100,000 tons since they are not included in the visualization:

```
fisheries <- fisheries %>%  
  filter(total > 100000)  
  
fisheries
```

```
## # A tibble: 82 x 4  
##   country      capture aquaculture  total  
##   <chr>         <dbl>         <dbl>  <dbl>  
## 1 Angola        486490          655  487145  
## 2 Argentina    755226          3673  758899  
## 3 Australia    174629          96847  271476  
## 4 Bangladesh  1674770         2202554  3878324
```

Data joins

```
fisheries %>% select(country)
```

```
## # A tibble: 82 x 1
##   country
##   <chr>
## 1 Angola
## 2 Argentina
## 3 Australia
## 4 Bangladesh
## 5 Brazil
## 6 Cambodia
## 7 Cameroon
## 8 Canada
## 9 Chad
## 10 Chile
## # ... with 72 more rows
```

```
continents
```

```
## # A tibble: 245 x 2
##   country      continent
##   <chr>      <chr>
## 1 Afghanistan Asia
## 2 Åland Islands Europe
## 3 Albania     Europe
## 4 Algeria     Africa
## 5 American Samoa Oceania
## 6 Andorra     Europe
## 7 Angola      Africa
## 8 Anguilla     Americas
## 9 Antigua & Barbuda Americas
## 10 Argentina   Americas
## # ... with 235 more rows
```

Joining data frames

`something_join(x, y)`

- **inner_join()**: all rows from x where there are matching values in y, return all combination of multiple matches in the case of multiple matches
- **left_join()**: all rows from x
- **right_join()**: all rows from y
- **full_join()**: all rows from both x and y
- **semi_join()**: all rows from x where there are matching values in y, keeping just columns from x.
- **anti_join()**: return all rows from x where there are not matching values in y, never duplicate rows of x
- ...

Setup

For the next few slides...

x

```
## # A tibble: 3 x 1
##   value
##   <dbl>
## 1     1
## 2     2
## 3     3
```

y

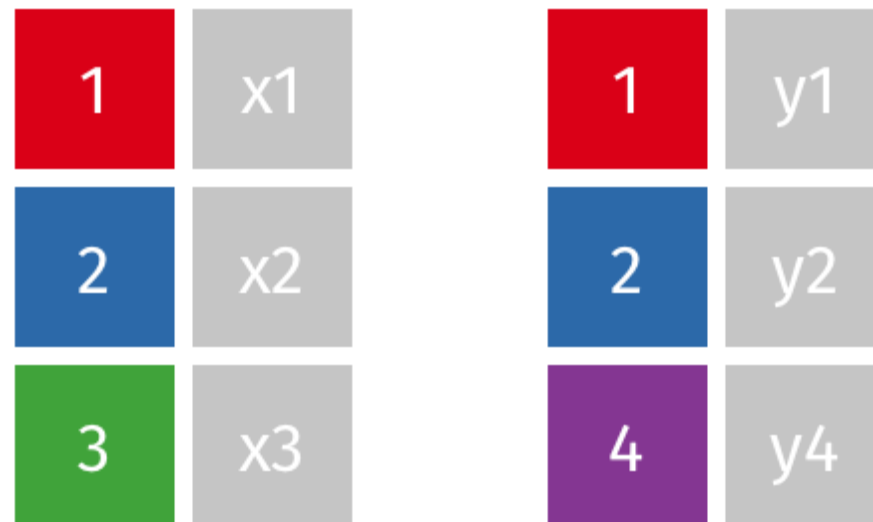
```
## # A tibble: 3 x 1
##   value
##   <dbl>
## 1     1
## 2     2
## 3     4
```


inner_join()

```
inner_join(x, y)
```

```
## # A tibble: 2 x 1
##   value
##   <dbl>
## 1     1
## 2     2
```

inner_join(x, y)



left_join()

```
left_join(x, y)
```

```
## # A tibble: 3 x 1
##   value
##   <dbl>
## 1     1
## 2     2
## 3     3
```

left_join(x, y)

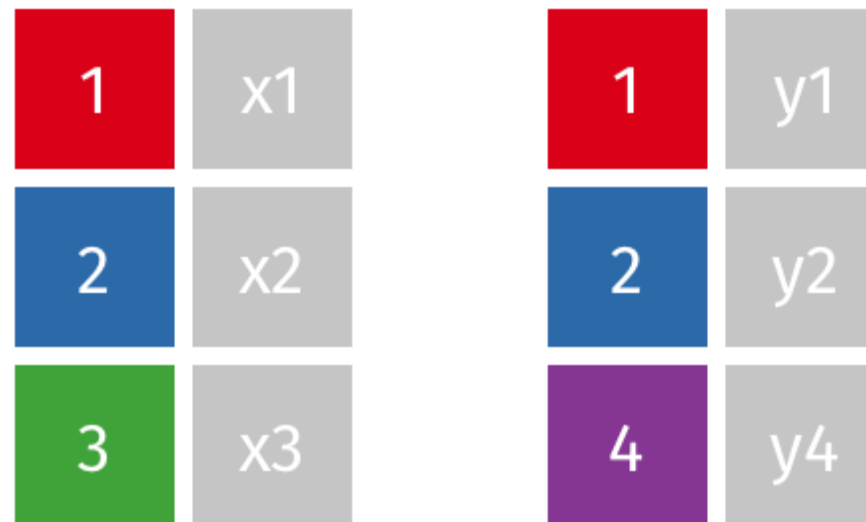
1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

right_join()

```
right_join(x, y)
```

```
## # A tibble: 3 x 1
##   value
##   <dbl>
## 1     1
## 2     2
## 3     4
```

right_join(x, y)



full_join()

```
full_join(x, y)
```

```
## # A tibble: 4 x 1
##   value
##   <dbl>
## 1     1
## 2     2
## 3     3
## 4     4
```

```
full_join(x, y)
```

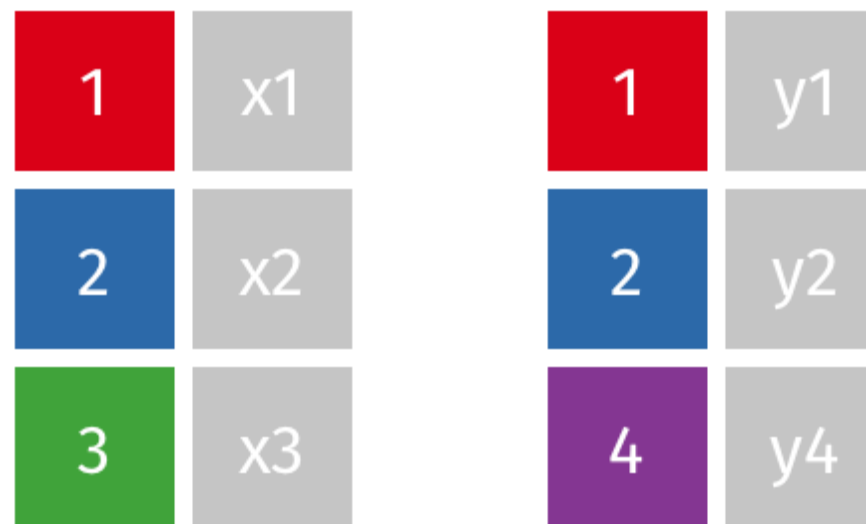
1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

semi_join()

```
semi_join(x, y)
```

```
## # A tibble: 2 x 1
##   value
##   <dbl>
## 1     1
## 2     2
```

semi_join(x, y)



anti_join()

```
anti_join(x, y)
```

```
## # A tibble: 1 x 1
##   value
##   <dbl>
## 1     3
```

anti_join(x, y)

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

We want to keep all rows and columns from **fisheries** and add a column for corresponding continents. Which join function should we use?

```
fisheries %>% select(country)
```

```
## # A tibble: 82 x 1
##   country
##   <chr>
## 1 Angola
## 2 Argentina
## 3 Australia
## 4 Bangladesh
## 5 Brazil
## 6 Cambodia
## 7 Cameroon
## 8 Canada
```

```
continents
```

```
## # A tibble: 245 x 2
##   country      continent
##   <chr>      <chr>
## 1 Afghanistan Asia
## 2 Åland Islands Europe
## 3 Albania     Europe
## 4 Algeria     Africa
## 5 American Samoa Oceania
## 6 Andorra     Europe
## 7 Angola      Africa
## 8 Anguilla    Americas
```

Join fisheries and continents

```
fisheries <- left_join(fisheries, continents)
```

How does **left_join()** know to join the two data frames by **country**?

Hint:

- Variables in the original fisheries dataset:

```
## [1] "country"      "capture"      "aquaculture" "total"
```

- Variables in the continents dataset:

```
## [1] "country"      "continent"
```


Check the data

```
fisheries %>%  
  filter(is.na(continent))
```

```
## # A tibble: 3 x 5
```

##	country	capture	aquaculture	total	continent
##	<chr>	<dbl>	<dbl>	<dbl>	<chr>
## 1	Democratic Republic of the Congo	237372	3161	240533	<NA>
## 2	Hong Kong	142775	4258	147033	<NA>
## 3	Myanmar	2072390	1017644	3090034	<NA>

Implement fixes

```
fisheries <- fisheries %>%  
  mutate(continent = case_when(  
    country == "Democratic Republic of the Congo" ~ "Africa",  
    country == "Hong Kong" ~ "Asia",  
    country == "Myanmar" ~ "Asia",  
    TRUE ~ continent  
  )  
)
```

...and check again

```
fisheries %>%  
  filter(is.na(continent))
```

```
## # A tibble: 0 x 5  
## # ... with 5 variables: country <chr>, capture <dbl>, aquaculture <dbl>, total  
## # continent <chr>
```

What does the following code do?

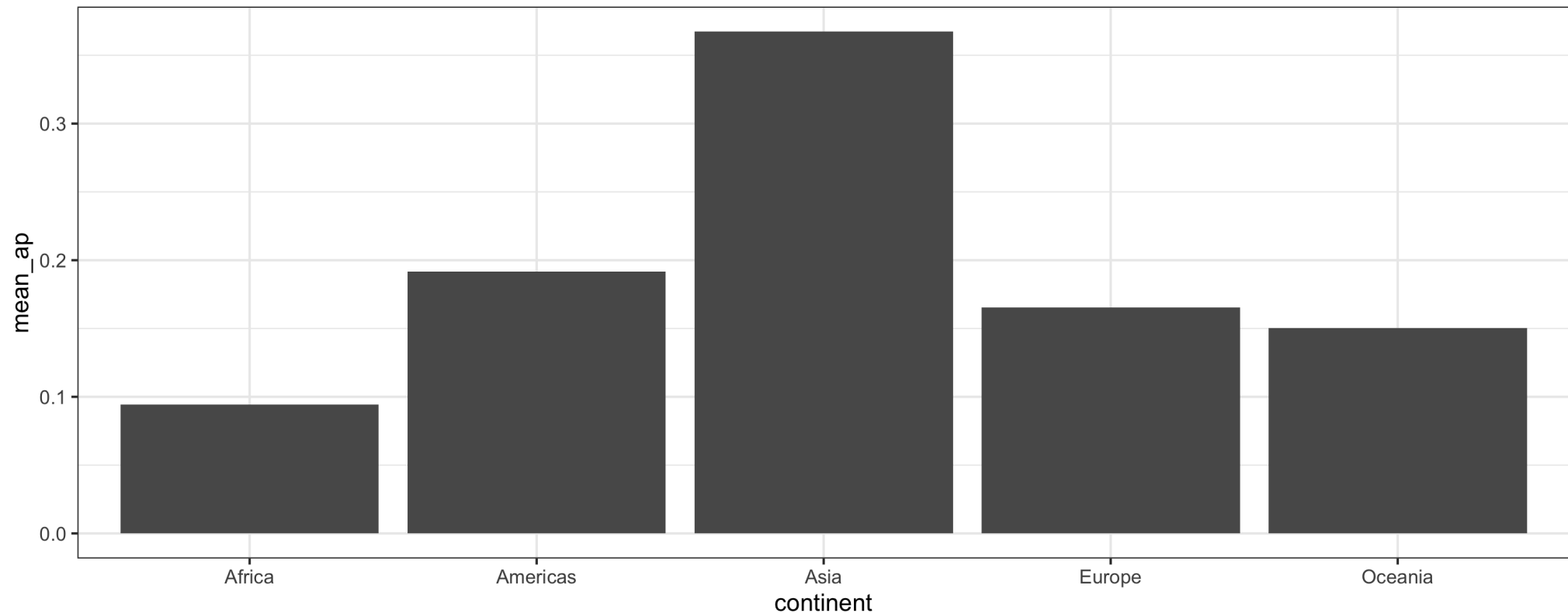
```
fisheries %>%  
  mutate(aquaculture_perc = aquaculture / total)
```

Demo

Demo

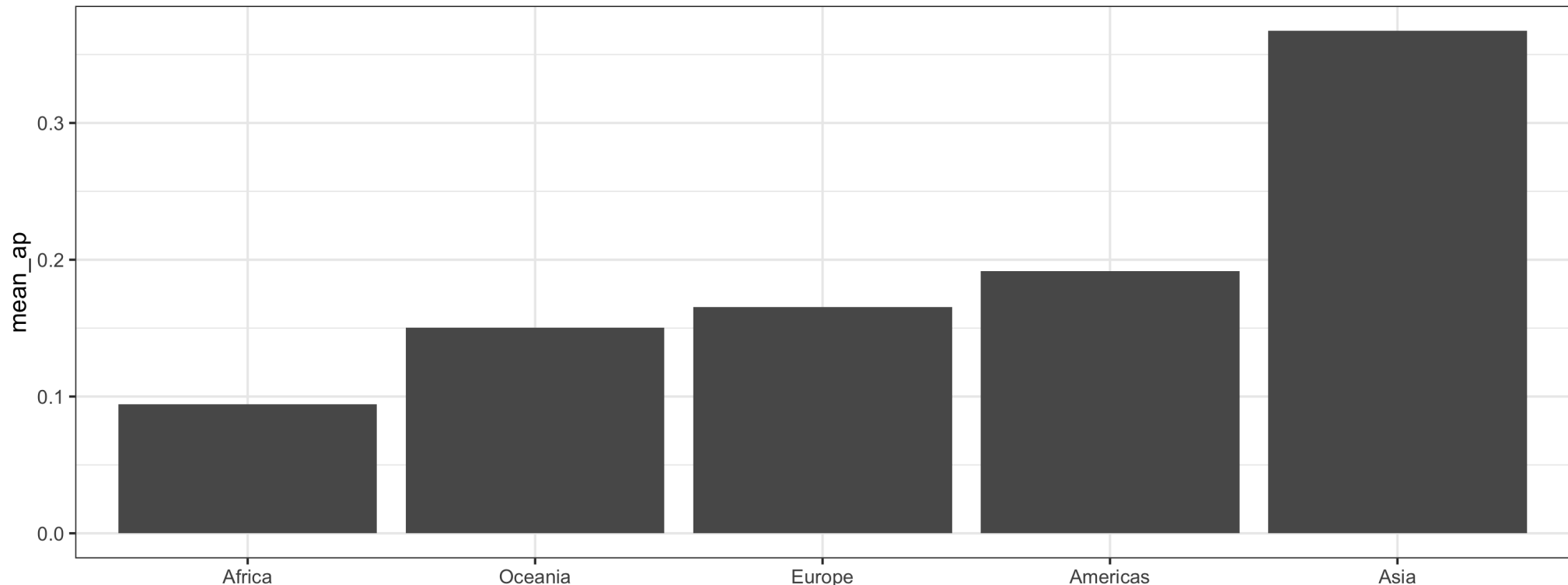
Visualize continent summary stats

```
ggplot(fisheries_summary, aes(x = continent, y = mean_ap)) +  
  geom_col()
```



Improve visualization

```
ggplot(fisheries_summary,  
       aes(x = fct_reorder(continent, mean_ap), y = mean_ap)) +  
       geom_col()
```



Improve visualization further

```
ggplot(fisheries_summary,  
      aes(y = fct_reorder(continent, mean_ap), x = mean_ap)) +  
  geom_col() +  
  scale_x_continuous(labels = label_percent(accuracy = 1)) +  
  labs(  
    x = "",  
    y = "",  
    title = "Average share of aquaculture by continent",  
    subtitle = "out of total fisheries harvest, 2016",  
    caption = "Source: bit.ly/2VrawTt"  
  ) +  
  theme_minimal()
```

➡ See next slide...

Average share of aquaculture by continent out of total fisheries harvest, 2016

