**Portfolio Data Analytics with Global Fishing Watch**

Methodology Document: Exploring Fishing Activity in Member Countries of the Sub-Regional Fisheries Commission (SRFC) and the Fisheries Committee for the West Central Gulf of Guinea (FCWC)

This document describes the technical methods used in detail, step by step. This document may be useful for reproducibility.

Table of Contents

[Overview 2](#_Toc206091663)

[Specific Questions 3](#_Toc206091664)

[SQL 4](#_Toc206091665)

[Aggregation of Global Fishing Watch data from 2020 to 2024 4](#_Toc206091666)

[Extraction of Exclusive Economic Zone (EEZ) spatial data for CSRP/SRFC member countries 11](#_Toc206091667)

[Extraction of spatial data for the Cabo Verde Exclusive Economic Zone 11](#_Toc206091668)

[Extraction of spatial data for the Mauritania Exclusive Economic Zone 12](#_Toc206091669)

[Extraction of spatial data for the Senegal Exclusive Economic Zone 12](#_Toc206091670)

[Extraction of spatial data for the Gambia Exclusive Economic Zone 12](#_Toc206091671)

[Extraction of spatial data for the Guinea-Bissau Exclusive Economic Zone 13](#_Toc206091672)

[Extraction of spatial data for the Guinea Exclusive Economic Zone 14](#_Toc206091673)

[Extraction of spatial data for the Sierra Leone Exclusive Economic Zone 15](#_Toc206091674)

[Combining all extracted spatial data tables of CSRP/SRFC member countries into a single table 15](#_Toc206091675)

[Extraction of Exclusive Economic Zone (EEZ) spatial data for CPCO/FCWC member countries 15](#_Toc206091676)

[Extraction of spatial data for the Benin Exclusive Economic Zone 16](#_Toc206091677)

[Extraction of spatial data for the Côte d’Ivoire Exclusive Economic Zone 16](#_Toc206091678)

[Extraction of spatial data for the Ghana Exclusive Economic Zone 16](#_Toc206091679)

[Extraction of spatial data for the Liberia Exclusive Economic Zone 17](#_Toc206091680)

[Extraction of spatial data for the Nigeria Exclusive Economic Zone 17](#_Toc206091681)

[Extraction of spatial data for the Togo Exclusive Economic Zone 17](#_Toc206091682)

[Combining all extracted spatial data tables of CPCO/FCWC member countries into a single table 18](#_Toc206091683)

[QGIS 19](#_Toc206091684)

[Download of the World Marine Protected Areas (MPAs) dataset in Shapefile format 19](#_Toc206091685)

[Adding World MPAs shapefiles in QGIS (to form polygon layers) 19](#_Toc206091686)

[Merge of the three polygon layers in QGIS 19](#_Toc206091687)

[Filtering MPAs of Sierra Leone in QGIS 19](#_Toc206091688)

[Export of the MPA\_Sierra\_Leone layer in CSV format for processing in BigQuery 20](#_Toc206091689)

[Conversion of the WKT column in the MPA\_Sierra\_Leone table from STRING to GEOGRAPHY type in BigQuery 20](#_Toc206091690)

[Tagging Global Fishing Watch data for Sierra Leone with corresponding MPA names 20](#_Toc206091691)

[Filtering to keep only fishing activity inside Marine Protected Areas of Sierra Leone 20](#_Toc206091692)

[Filtering MPAs of Cabo Verde in QGIS 20](#_Toc206091693)

[Export of the MPA\_Cabo\_Verde layer in CSV format for processing in BigQuery 21](#_Toc206091694)

[Conversion of the WKT column in the MPA\_Cabo\_Verde table from STRING to GEOGRAPHY type in BigQuery 21](#_Toc206091695)

[Tagging Global Fishing Watch data for Cabo Verde with corresponding MPA names 21](#_Toc206091696)

[Filtering to keep only fishing activity inside Marine Protected Areas of Cabo Verde 22](#_Toc206091697)

[Filtering MPAs of Gambia in QGIS 22](#_Toc206091698)

[Export of the MPA\_Gambia layer in CSV format for processing in BigQuery 22](#_Toc206091699)

[Conversion of the WKT column in the MPA\_Gambia table from STRING to GEOGRAPHY type in BigQuery 23](#_Toc206091700)

[Tagging Global Fishing Watch data for Gambia with corresponding MPA names 23](#_Toc206091701)

[Filtering to keep only fishing activity inside Marine Protected Areas of Gambia 23](#_Toc206091702)

[Combining all fishing cells within Marine Protected Areas across all CSRP countries 29](#_Toc206091703)

[Classification of foreign and local fishing flags with SQL 35](#_Toc206091704)

[Case of CSRP RFMO 35](#_Toc206091705)

[Case of CPCO RFMO 35](#_Toc206091706)

[Filtering data related to trawlers with SQL 35](#_Toc206091707)

[Grouping fishing hours for trawlers by month and year using SQL 36](#_Toc206091708)

[Grouping fishing hours for trawlers by country and month for each year using SQL 36](#_Toc206091709)

[Tableau 37](#_Toc206091710)

[R 51](#_Toc206091711)

# Overview

This portfolio explores the variation of fishing activity in member countries of the Sub-Regional Fisheries Commission (CSRP/SRFC) and the Fisheries Committee for the West Central Gulf of Guinea (CPCO/FCWC) across key parameters, including year and month, flag state, gear type, and Marine Protected Areas (MPAs). The analysis covers the period from 2020 to 2024. The SRFC (CSRP in French) member states are Senegal, Mauritania, The Gambia, Guinea-Bissau, Guinea, Sierra Leone, and Cabo Verde, while the FCWC (CPCO in French) member states are Benin, Côte d’Ivoire, Ghana, Liberia, Nigeria, and Togo.

# Specific Questions

The exploratory data analysis consisted of exploring Global Fishing Watch data to answer key questions, such as :

* How vary the total fishing hours according to year and fleet type (local vs foreign) in the area covered by CSRP and the area covered by CPCO? How vary the total fishing hours according to year and fleet type (local vs foreign) per CSRP country and CPCO country?
* What are the top five flag states with the highest total fishing hours in the area covered by CSRP and the area covered by CPCO? What are the top five flag states with the highest total fishing hours in each CSRP country and CPCO country?
* How vary the total fishing hours according to geartype in the area covered by CSRP and the area covered by CPCO? What are the top five most used types of fishing gear in each CSRP country and CPCO country?
* What are the top five most used types of fishing gear by the top five flag states with the highest total fishing hours in the area covered by CSRP and the area covered by CPCO?
* How vary the total fishing hours according to month in the area covered by CSRP and the area covered by CPCO? How vary the total fishing hours according to month in each CSRP country and CPCO country?
* How vary the total fishing hours done with trawlers gears according to the month in each CSRP country and CPCO?
* What are the total fishing hours in CSRP and CPCO Marine Protected Areas?

# SQL

## Aggregation of Global Fishing Watch data from 2020 to 2024

**Aggregating data of global fishing watch of the year 2024**

* In BigQuery, I create a database named « GFW2 » in a project named « smiling-gasket-459215-b7 »
* I upload the data « fleet-monthly-csvs-10-v3-2024.zip » from Global Fishing Watch <https://globalfishingwatch.org/data-download/datasets/public-fishing-effort>
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-01 » ; The Table is named « GFW2\_2024\_1 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-02 » ; The Table is named « GFW2\_2024\_2 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-03 » ; The Table is named « GFW2\_2024\_3 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-04 » ; The Table is named « GFW2\_2024\_4 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-05 » ; The Table is named « GFW2\_2024\_5 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-06 » ; The Table is named « GFW2\_2024\_6 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-07 » ; The Table is named « GFW2\_2024\_7 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-08 » ; The Table is named « GFW2\_2024\_8 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-09 » ; The Table is named « GFW2\_2024\_9 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-10 » ; The Table is named « GFW2\_2024\_10 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-11 » ; The Table is named « GFW2\_2024\_11 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2024-01-12 » ; The Table is named « GFW2\_2024\_12 ».
* With SQL, I combine the tables « GFW2\_2024\_1 » to « GFW2\_2024\_12 », the following code was used

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_1`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_2`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_3`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_4`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_5`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_6`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_7`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_8`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_9`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_10`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_11`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_12`

The resulted table from the combination was saved in BigQuery and named as « GFW2\_2024 ». The table contains 13 838 619 rows.

**Aggregating data of global fishing watch of the year 2023**

* I upload the data « fleet-monthly-csvs-10-v3-2023.zip » from Global Fishing Watch <https://globalfishingwatch.org/data-download/datasets/public-fishing-effort>
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-01 » ; The Table is named « GFW2\_2023\_1 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-02 » ; The Table is named « GFW2\_2023\_2 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-03 » ; The Table is named « GFW2\_2023\_3 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-04 » ; The Table is named « GFW2\_2023\_4 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-05 » ; The Table is named « GFW2\_2023\_5 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-06 » ; The Table is named « GFW2\_2023\_6 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-07 » ; The Table is named « GFW2\_2023\_7 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-08 » ; The Table is named « GFW2\_2023\_8 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-09 » ; The Table is named « GFW2\_2023\_9 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-10 » ; The Table is named « GFW2\_2023\_10 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-11 » ; The Table is named « GFW2\_2023\_11 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2023-01-12 » ; The Table is named « GFW2\_2023\_12 ».
* With SQL, I combine the tables « GFW2\_2023\_1 » to « GFW2\_2023\_12 », the following code was used

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_1`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_2`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_3`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_4`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_5`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_6`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_7`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_8`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_9`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_10`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_11`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023\_12`

The resulted table from the combination was saved in BigQuery and named as « GFW2\_2023 ». The table contains 14 560 171 rows.

**Aggregating data of global fishing watch of the year 2022**

* I upload the data « fleet-monthly-csvs-10-v3-2022.zip » from Global Fishing Watch <https://globalfishingwatch.org/data-download/datasets/public-fishing-effort>
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-01 » ; The Table is named « GFW2\_2022\_1 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-02 » ; The Table is named « GFW2\_2022\_2 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-03 » ; The Table is named « GFW2\_2022\_3 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-04 » ; The Table is named « GFW2\_2022\_4 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-05 » ; The Table is named « GFW2\_2022\_5 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-06 » ; The Table is named « GFW2\_2022\_6 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-07 » ; The Table is named « GFW2\_2022\_7 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-08 » ; The Table is named « GFW2\_2022\_8 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-09 » ; The Table is named « GFW2\_2022\_9 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-10 » ; The Table is named « GFW2\_2022\_10 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-11 » ; The Table is named « GFW2\_2022\_11 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2022-01-12 » ; The Table is named « GFW2\_2022\_12 ».
* With SQL, I combine the tables « GFW2\_2022\_1 » to « GFW2\_2022\_12 », the following code was used

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_1`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_2`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_3`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_4`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_5`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_6`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_7`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_8`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_9`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_10`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_11`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022\_12`

The resulted table from the combination was saved in BigQuery and named as « GFW2\_2022 ». The table contains 13 463 690 rows.

**Aggregating data of global fishing watch of the year 2021**

* I upload the data « fleet-monthly-csvs-10-v3-2021.zip » from Global Fishing Watch <https://globalfishingwatch.org/data-download/datasets/public-fishing-effort>
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-01 » ; The Table is named « GFW2\_2021\_1 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-02 » ; The Table is named « GFW2\_2021\_2 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-03 » ; The Table is named « GFW2\_2021\_3 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-04 » ; The Table is named « GFW2\_2021\_4 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-05 » ; The Table is named « GFW2\_2021\_5 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-06 » ; The Table is named « GFW2\_2021\_6 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-07 » ; The Table is named « GFW2\_2021\_7 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-08 » ; The Table is named « GFW2\_2021\_8 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-09 » ; The Table is named « GFW2\_2021\_9 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-10 » ; The Table is named « GFW2\_2021\_10 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-11 » ; The Table is named « GFW2\_2021\_11 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2021-01-12 » ; The Table is named « GFW2\_2021\_12 ».
* With SQL, I combine the tables « GFW2\_2021\_1 » to « GFW2\_2021\_12 », the following code was used

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_1`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_2`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_3`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_4`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_5`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_6`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_7`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_8`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_9`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_10`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_11`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_12`

The resulted table from the combination was not saved in BigQuery due to limitation, failure to export results. The table contains 12 954 173 rows.

**Aggregating data of global fishing watch of the year 2020**

* I upload the data « fleet-monthly-csvs-10-v3-2020.zip » from Global Fishing Watch <https://globalfishingwatch.org/data-download/datasets/public-fishing-effort>
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-01 » ; The Table is named « GFW2\_2020\_1 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-02 » ; The Table is named « GFW2\_2020\_2 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-03 » ; The Table is named « GFW2\_2020\_3 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-04 » ; The Table is named « GFW2\_2020\_4 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-05 » ; The Table is named « GFW2\_2020\_5 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-06 » ; The Table is named « GFW2\_2020\_6 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-07 » ; The Table is named « GFW2\_2020\_7 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-08 » ; The Table is named « GFW2\_2020\_8 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-09 » ; The Table is named « GFW2\_2020\_9 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-10 » ; The Table is named « GFW2\_2020\_10 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-11 » ; The Table is named « GFW2\_2020\_11 ».
* In BigQuery, I create a table in the database « GFW » by uploading the file « fleet-monthly-csvs-10-v3-2020-01-12 » ; The Table is named « GFW2\_2020\_12 ».
* With SQL, I combine the tables « GFW2\_2020\_1 » to « GFW2\_2020\_12 », the following code was used

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_1`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_2`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_3`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_4`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_5`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_6`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_7`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_8`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_9`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_10`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_11`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020\_12`

The resulted table from the combination was saved in BigQuery and named as « GFW2\_2020 ». The table contains 12 603 878 rows.

**Aggregating Global Fishing Watch data from 2020 to 2024**

* With SQL, I combine the tables « GFW2\_2024» to « GFW2\_2020», the following code was used

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2023`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2022`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2020`

Due to size limitations, Firstly, in one table the data of years 2020, 2023, and 2024 were grouped and in another table the data of years 2021 and 2022 were grouped. The resulted tables from the combinations was saved in BigQuery and named as « GFW2\_2024\_2023\_2020 » and « GFW2\_2021\_2022 ». The first table contains 41 002 668 rows and the second 26 417 863 rows. These two last tables were joined. The following code was used :

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2024\_2023\_2020`

UNION ALL

SELECT \*

FROM `smiling-gasket-459215-b7.GFW2.GFW2\_2021\_2022`

The results were saved in a table named GFW2\_1 in an another BigQuery project. GFW2\_1 contains 67 420 531 rows.

## Extraction of Exclusive Economic Zone (EEZ) spatial data for CSRP/SRFC member countries

* + - * Instead of using bounding boxes, we use polygons shapefiles to extract spatial data. The polygon shapefiles were sourced from ; <https://www.marineregions.org/>. We click on the JSON format download to get coordinates of the EEZ zones.

### Extraction of spatial data for the Cabo Verde Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS) ».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Cabo Verde' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": [[[[-20.07059607,18.81626882],[-20,18.73666667],[-19.93333333,18.58166667],[-19.89666667,18.48],[-19.85833333,18.4],[-19.81666667,18.31333333],[-19.78333333,18.22333333],[-19.73666667,18.13],[-19.70166667,18.04166667],[-19.63333333,17.88833333],[-19.63333333,17.735]…………………………………………………………………………………………………………….. 24.64981794,14.97268271],[-24.65010655,14.97251666],[-24.65014803,14.96935487],[-24.64945245,14.9691627]]]]}'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Cabo\_Verde. It contains 228 369 rows.

### Extraction of spatial data for the Mauritania Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS) ».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Mauritania' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Mauritania. It contains 176 986 rows.

### Extraction of spatial data for the Senegal Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Senegal' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Senegal. It contains 88 336 rows.

### Extraction of spatial data for the Gambia Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Gambia' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Gambia. It contains 17 204 rows.

### Extraction of spatial data for the Guinea-Bissau Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Guinea Bissau' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

After running the running the code in the commentary section, BigQuery sent an error message « The query is too large. The maximum standard SQL query length is 1024.00K characters, including comments and white space characters. »

To fix this issue, we use the tool Mapshaper <https://mapshaper.org/> to simplify the Guinea Bissau EEZ polygons. We upload in Mapshaper the original json file (sourced from marineregions site). We simplify using a setting of 10% retained points, for the method we check the option « Visvalingam / weighted area ». We check the option « Prevent shape removal ». In the console section we write the code « clean ». This code allow to clean the simplified polygon, which make it valid for BigQuery. After we export the simplified version in geojson format.

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Guinea Bissau' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Guinea\_Bissau. It contains 67 032 rows.

### Extraction of spatial data for the Guinea Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Guinea' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

After running the running the code in the commentary section, BigQuery sent an error message « The query is too large. The maximum standard SQL query length is 1024.00K characters, including comments and white space characters. »

To fix this issue, we use the tool Mapshaper <https://mapshaper.org/> to simplify the Guinea Bissau EEZ polygons. We upload in Mapshaper the original json file (sourced from marineregions site). We simplify using a setting of 10% retained points, for the method we check the option « Visvalingam / weighted area ». We check the option « Prevent shape removal ». In the console section we write the code « clean ». This code allow to clean the simplified polygon, which make it valid for BigQuery. After we export the simplified version in geojson format.

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Guinea' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Guinea. It contains 54 479 rows.

### Extraction of spatial data for the Sierra Leone Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Sierra Leone' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Sierra\_Leone. It contains 29 078 rows.

## Combining all extracted spatial data tables of CSRP/SRFC member countries into a single table

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Cabo\_Verde`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Mauritania`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Senegal`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Gambia`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Guinea\_Bissau`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Guinea`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Sierra\_Leone`

The results were saved in a table named GFW\_CSRP\_Countries\_2020\_2024. It contains 661 484 rows.

## Extraction of Exclusive Economic Zone (EEZ) spatial data for CPCO/FCWC member countries

* + - * Instead of using bounding boxes, we use polygons shapefiles to extract spatial data. The polygon shapefiles were sourced from ; <https://www.marineregions.org/>. We click on the JSON format download to get coordinates of the EEZ zones.

### Extraction of spatial data for the Benin Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS) ».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Benin' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Benin. It contains 2175 rows.

### Extraction of spatial data for the Côte d’Ivoire Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS) ».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

"Côte d'Ivoire" AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Côte\_Ivoire. It contains 69 460 rows.

### Extraction of spatial data for the Ghana Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Ghana' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Ghana. It contains 110 890 rows.

### Extraction of spatial data for the Liberia Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Liberia' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Liberia. It contains 51 041 rows.

### Extraction of spatial data for the Nigeria Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Nigeria' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Nigeria. It contains 36 899 rows.

### Extraction of spatial data for the Togo Exclusive Economic Zone

Citation of the polygon data « Flanders Marine Institute (2023). Maritime Boundaries Geodatabase: Maritime Boundaries and Exclusive Economic Zones (200NM), version 12. Available online at http://www.marineregions.org/. https://doi.org/10.14284/632 (look up in IMIS)».

The SQL code is shown below. The above SQL code was troncked, all the spatial latitudes and longitudes were not displayed above (the entire code is in the commentary section).

SELECT

\*,

'Togo' AS country

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1`

WHERE ST\_CONTAINS(

ST\_GEOGFROMGEOJSON('{"type":"MultiPolygon","coordinates": }'),

ST\_GEOGPOINT(cell\_ll\_lon, cell\_ll\_lat)

)

The results were saved in a table named GFW2\_1\_Togo. It contains 1567 rows.

## Combining all extracted spatial data tables of CPCO/FCWC member countries into a single table

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Benin`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Côte\_Ivoire`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Ghana`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Liberia`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Nigeria`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Togo`

The results were saved in a table named GFW\_CPCO\_Countries\_2020\_2024. It contains 272 032 rows.

# QGIS

## Download of the World Marine Protected Areas (MPAs) dataset in Shapefile format

World Marine Protected Areas dataset was uploaded from protected planet site in a shapefile SHP format. <https://www.protectedplanet.net/en/thematic-areas/marine-protected-areas>

Citation : UNEP-WCMC and IUCN (2025), Protected Planet: The World Database on Protected Areas (WDPA) and World Database on Other Effective Area-based Conservation Measures (WD-OECM) [Online], July 2025, Cambridge, UK: UNEP-WCMC and IUCN. Available at: www.protectedplanet.net.

## Adding World MPAs shapefiles in QGIS (to form polygon layers)

The file « WDPA\_WDOECM\_Jul2025\_Public\_marine\_shp-polygons.shp » from the folder « WDPA\_WDOECM\_Jul2025\_Public\_marine\_shp\_0 » was added in QGIS.

The file « WDPA\_WDOECM\_Jul2025\_Public\_marine\_shp-polygons.shp » from the folder « WDPA\_WDOECM\_Jul2025\_Public\_marine\_shp\_1 » was added in QGIS.

The file « WDPA\_WDOECM\_Jul2025\_Public\_marine\_shp-polygons.shp » from the folder « WDPA\_WDOECM\_Jul2025\_Public\_marine\_shp\_2 » was added in QGIS.

To add file at the top menu bar of QGIS, we click Layer > Add Layer > Add Vector Layer

## Merge of the three polygon layers in QGIS

In QGIS, we create a single merged layer. At the QGIS top menu we click Vector → Data Management Tools → Merge Vector Layers. The three polygon layers are added as input layers". Under Destination CRS, we choose EPSG:4326 (WGS 84). The output file was saves in Geopackage format as « Merged\_MPA\_World.gpkg ». After running, the layer appear in the layers panel.

## Filtering MPAs of Sierra Leone in QGIS

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Cabo Verde | CPV |
| Gambia | GMB |
| Guinea | GIN |
| Guinea-Bissau | GNB |
| Mauritania | MRT |
| Senegal | SEN |
| **Sierra Leone** | **SLE** |

We paste this expression in order to filter MPAs of Sierra Leone after the click on Select Features by Expression.

Code :

"ISO3" IN ('SLE')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Sierra\_Leone.geojson → CRS: EPSG:4326 → OK

## Export of the MPA\_Sierra\_Leone layer in CSV format for processing in BigQuery

We right click on the geojson layer « MPA\_Sierra\_Leone » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Sierra\_Leone CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Sierra\_Leone file in BigQuery. For the Schema we check auto detect.

## Conversion of the WKT column in the MPA\_Sierra\_Leone table from STRING to GEOGRAPHY type in BigQuery

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Sierra\_Leone\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Sierra\_Leone`;

## Tagging Global Fishing Watch data for Sierra Leone with corresponding MPA names

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Sierra\_Leone` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Sierra\_Leone\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Sierra\_Leone\_MPA\_And\_Not\_MPA ».

## Filtering to keep only fishing activity inside Marine Protected Areas of Sierra Leone

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Sierra\_Leone\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

The results are saved in table named « GFW\_Sierra\_Leone\_MPA\_Only ».

## Filtering MPAs of Cabo Verde in QGIS

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| **Cabo Verde** | **CPV** |
| Gambia | GMB |
| Guinea | GIN |
| Guinea-Bissau | GNB |
| Mauritania | MRT |
| Senegal | SEN |
| Sierra Leone | SLE |

We paste this expression in order to filter MPAs of Cabo Verde after the click on Select Features by Expression.

Code :

"ISO3" IN ('CPV')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Cabo\_Verde.geojson → CRS: EPSG:4326 → OK

## Export of the MPA\_Cabo\_Verde layer in CSV format for processing in BigQuery

We right click on the geojson layer « MPA\_Cabo\_Verde » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Cabo\_Verde CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Cabo\_Verde file in BigQuery. For the Schema we check auto detect. To remove errors of uploading, some MPAs multipolygons (corresponding to the fids 4762, 4757, 4765, 4767, 4772, and 4776) were simplified using Mapshaper. Additionnaly, tools like <https://geojson-to-wkt-converter.onrender.com/>, <https://cloudconvert.com/xlsx-to-csv>, and <https://www.freeconvert.com/xls-to-csv/download> were used to convert files after some data cleaning aiming to upload properly the csv file in BigQuery.

## Conversion of the WKT column in the MPA\_Cabo\_Verde table from STRING to GEOGRAPHY type in BigQuery

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Cabo\_Verde\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Cabo\_Verde`;

## Tagging Global Fishing Watch data for Cabo Verde with corresponding MPA names

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Cabo\_Verde` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Cabo\_Verde\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Cabo\_Verde\_MPA\_And\_Not\_MPA ».

## Filtering to keep only fishing activity inside Marine Protected Areas of Cabo Verde

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Cabo\_Verde\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

The results are saved in table named « GFW\_Cabo\_Verde\_MPA\_Only ».

## Filtering MPAs of Gambia in QGIS

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Cabo Verde | CPV |
| Gambia | GMB |
| Guinea | GIN |
| Guinea-Bissau | GNB |
| Mauritania | MRT |
| Senegal | SEN |
| Sierra Leone | SLE |

We paste this expression in order to filter MPAs of Gambia after the click on Select Features by Expression.

Code :

"ISO3" IN ('GMB')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Gambia.geojson → CRS: EPSG:4326 → OK

## Export of the MPA\_Gambia layer in CSV format for processing in BigQuery

We right click on the geojson layer « MPA\_Gambia » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Gambia CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Gambia file in BigQuery. For the Schema we check auto detect.

## Conversion of the WKT column in the MPA\_Gambia table from STRING to GEOGRAPHY type in BigQuery

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Gambia\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Gambia`;

## Tagging Global Fishing Watch data for Gambia with corresponding MPA names

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Gambia` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Gambia\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Gambia\_MPA\_And\_Not\_MPA ».

## Filtering to keep only fishing activity inside Marine Protected Areas of Gambia

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Gambia\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

**No result was found.**

**Filtering MPAs of Guinea in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Cabo Verde | CPV |
| Gambia | GMB |
| **Guinea** | **GIN** |
| Guinea-Bissau | GNB |
| Mauritania | MRT |
| Senegal | SEN |
| Sierra Leone | SLE |

We paste this expression in order to filter MPAs of Guinea after the click on Select Features by Expression.

Code :

"ISO3" IN ('GIN')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Guinea.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Guinea layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Guinea » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Guinea CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Guinea file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Guinea table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Guinea\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Guinea`;

**Tagging Global Fishing Watch data of Guinea with MPA name of Guinea**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Guinea` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Guinea\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Guinea\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Cabo Verde**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Guinea\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

**No result was found.**

**Filtering MPAs of Guinea Bissau in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Cabo Verde | CPV |
| Gambia | GMB |
| Guinea | GIN |
| **Guinea-Bissau** | **GNB** |
| Mauritania | MRT |
| Senegal | SEN |
| Sierra Leone | SLE |

We paste this expression in order to filter MPAs of Guinea Bissau after the click on Select Features by Expression.

Code :

"ISO3" IN ('GNB')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Guinea\_Bissau.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Guinea\_Bissau layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Cabo\_Verde » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Guinea\_Bissau CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Guinea\_Bissau file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Guinea\_Bissau table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Guinea\_Bissau\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Guinea\_Bissau`;

**Tagging Global Fishing Watch data of Guinea\_Bissau with MPA name of Guinea\_Bissau**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Guinea\_Bissau` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Guinea\_Bissau\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Guinea\_Bissau\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Guinea Bissau**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Guinea\_Bissau\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

The results are saved in table named « GFW\_Guinea\_Bissau\_MPA\_Only ».

**Filtering MPAs of Mauritania in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Cabo Verde | CPV |
| Gambia | GMB |
| Guinea | GIN |
| Guinea-Bissau | GNB |
| **Mauritania** | **MRT** |
| Senegal | SEN |
| Sierra Leone | SLE |

We paste this expression in order to filter MPAs of Mauritania after the click on Select Features by Expression.

Code :

"ISO3" IN ('MRT')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Mauritania.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Mauritania layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Mauritania » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Mauritania CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Mauritania file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Mauritania table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Mauritania\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Mauritania`;

**Tagging Global Fishing Watch data of Mauritania with MPA name of Mauritania**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Mauritania` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Mauritania\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Mauritania\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Mauritania**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Mauritania\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

**No result was found.**

**Filtering MPAs of Senegal in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Cabo Verde | CPV |
| Gambia | GMB |
| Guinea | GIN |
| Guinea-Bissau | GNB |
| Mauritania | MRT |
| **Senegal** | **SEN** |
| Sierra Leone | SLE |

We paste this expression in order to filter MPAs of Senegal after the click on Select Features by Expression.

Code :

"ISO3" IN ('SEN')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Senegal.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Senegal layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Senegal » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Senegal CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Senegal file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Senegal table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Senegal\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Senegal`;

**Tagging Global Fishing Watch data of Senegal with MPA name of Senegal**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Senegal` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Senegal\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Senegal\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Cabo Verde**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Senegal\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

The results are saved in table named « GFW\_Senegal\_MPA\_Only ».

## Combining all fishing cells within Marine Protected Areas across all CSRP countries

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Sierra\_Leone\_MPA\_Only`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Cabo\_Verde\_MPA\_Only`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Guinea\_Bissau\_MPA\_Only`

UNION ALL

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Senegal\_MPA\_Only`

The results are saved in table named « GFW\_CSRP\_Countries\_MPA\_2020\_2024». The table contains 5864 rows.

**Filtering MPAs of Benin in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| **Benin** | **BEN** |
| Côte d’Ivoire | CIV |
| Ghana | GHA |
| Liberia | LBR |
| Nigeria | NGA |
| Togo | TGO |

We paste this expression in order to filter MPAs of Benin after the click on Select Features by Expression.

Code :

"ISO3" IN ('BEN')

**After running, no matching features were found.**

**Filtering MPAs of Côte d’Ivoire in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Benin | BEN |
| **Côte d’Ivoire** | **CIV** |
| Ghana | GHA |
| Liberia | LBR |
| Nigeria | NGA |
| Togo | TGO |

We paste this expression in order to filter MPAs of Côte d’Ivoire after the click on Select Features by Expression.

Code :

"ISO3" IN ('CIV')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Côte\_Ivoire.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Côte\_Ivoire layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Côte\_Ivoire» → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Côte\_Ivoire CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Côte\_Ivoire file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Côte\_Ivoire table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Côte\_Ivoire\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Côte\_Ivoire`;

**Tagging Global Fishing Watch data of Cabo Verde with MPA name of Côte d’Ivoire**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Côte\_Ivoire ` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Côte\_Ivoire\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Côte\_Ivoire\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Côte d’Ivoire**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Côte\_Ivoire\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

**No result was found.**

**Filtering MPAs of Ghana in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Benin | BEN |
| Côte d’Ivoire | CIV |
| **Ghana** | **GHA** |
| Liberia | LBR |
| Nigeria | NGA |
| Togo | TGO |

We paste this expression in order to filter MPAs of Ghana after the click on Select Features by Expression.

Code :

"ISO3" IN ('GHA')

**After running, no matching features were found.**

**Filtering MPAs of Liberia in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Benin | BEN |
| Côte d’Ivoire | CIV |
| Ghana | GHA |
| **Liberia** | **LBR** |
| Nigeria | NGA |
| Togo | TGO |

We paste this expression in order to filter MPAs of Guinea after the click on Select Features by Expression.

Code :

"ISO3" IN ('LBR')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Liberia.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Liberia layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Liberia » → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Liberia CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Liberia file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Liberia table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Liberia\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Liberia`;

**Tagging Global Fishing Watch data of Liberia with MPA name of Liberia**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Liberia` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Liberia\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Liberia\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Liberia**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Liberia\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

**No result was found.**

**Filtering MPAs of Nigeria in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Benin | BEN |
| Côte d’Ivoire | CIV |
| Ghana | GHA |
| Liberia | LBR |
| **Nigeria** | **NGA** |
| Togo | TGO |

We paste this expression in order to filter MPAs of Nigeria after the click on Select Features by Expression.

Code :

"ISO3" IN ('NGA')

We export selected MPAs:

Right-click on the « Merged\_MPA\_World » layer in Layers panel → Choose Export → Save Selected Features As → Format: GeoJSON → File name: MPA\_Nigeria.geojson → CRS: EPSG:4326 → OK

**Exporting MPA\_Nigeria layer in CSV format for preparing work in BigQuery**

We right click on the geojson layer « MPA\_Nigeria» → Export → Save Features As → Format: CSV → CRS: EPSG:4326 (WGS84) → Geometry: AS\_WKT. A csv file is generated with a WKT column.

**Uploading MPA\_Nigeria CSV file in BigQuery**

We go to Create Table. We upload the MPA\_Nigeria file in BigQuery. For the Schema we check auto detect.

**Converting the column WKT of MPA\_Nigeria table from STRING type to GEOGRAPHY type in BigQuery**

CREATE OR REPLACE TABLE `oceanic-grin-466919-s4.GFW2.MPA\_Nigeria\_with\_geom` AS

SELECT

  \*,

  ST\_GEOGFROMTEXT(WKT) AS geom

FROM

  `oceanic-grin-466919-s4.GFW2.MPA\_Nigeria`;

**Tagging Global Fishing Watch data of Nigeria with MPA name of Nigeria**

SELECT

  g.\*,

  COALESCE(m.NAME, 'Not MPA') AS mpa\_name

FROM

  `oceanic-grin-466919-s4.GFW2.GFW2\_1\_Nigeria` AS g

LEFT JOIN

  `oceanic-grin-466919-s4.GFW2.MPA\_Nigeria\_with\_geom` AS m

ON

  ST\_WITHIN(ST\_GEOGPOINT(g.cell\_ll\_lon, g.cell\_ll\_lat), m.geom)

The results are saved in a table named « GFW\_Nigeria\_MPA\_And\_Not\_MPA ».

**Retaining only fishing activity within Marine Protected Areas of Nigeria**

SELECT \*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_Nigeria\_MPA\_And\_Not\_MPA`

WHERE mpa\_name != 'Not MPA'

**No result was found.**

**Filtering MPAs of Togo in QGIS**

We go to edit at the top menu of QGIS. We clik on Select → Select Features by Expression in the QGIS. We filter the MPAs of CSRP member countries by using the ISO3 (3-letter country codes).

| **Country** | **ISO3** |
| --- | --- |
| Benin | BEN |
| Côte d’Ivoire | CIV |
| Ghana | GHA |
| Liberia | LBR |
| Nigeria | NGA |
| **Togo** | **TGO** |

We paste this expression in order to filter MPAs of Mauritania after the click on Select Features by Expression.

Code :

"ISO3" IN ('TGO')

**After running, no matching features were found.**

## Classification of foreign and local fishing flags with SQL

### Case of CSRP RFMO

* Local fishing is defined as the fishing done by flags of CSRP countries members meaning flag vessel is equal to CPV or MRT or SEN or GMB or GNB or GIN or SLE. When flag vessel is different to these ones, the fishing is defined as foreign.
* From the table GFW\_CSRP\_Countries\_2020\_2024, with SQL, I generate a column named « local\_vs\_foreign » which contain the following categorical data local and foreign. The goal is to classify data according to the fact that the fishing is done by a vessel registered to one CSRP countries or the contrary. The following script is used.

SELECT \*,

  CASE

    WHEN flag IN ('CPV', 'MRT', 'SEN', 'GMB', 'GNB', 'GIN', 'SLE') THEN 'local'

    ELSE 'foreign'

  END AS local\_vs\_foreign

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CSRP\_Countries\_2020\_2024`

The results table is named GFW\_CSRP\_Countries\_2020\_2024\_Local\_vs\_Foreign\_Flag

### Case of CPCO RFMO

* Local fishing is defined as the fishing done by flags of CSRP countries members meaning flag vessel is equal to BEN or CIV or GHA or LBR or NGA or TGO. When flag vessel is different to these ones, the fishing is defined as foreign.
* From the table GFW\_CPCO\_Countries\_2020\_2024, with SQL, I generate a column named « local\_vs\_foreign » which contain the following categorical data local and foreign. The goal is to classify data according to the fact that the fishing is done by a vessel registered to one CSRP countries or the contrary. The following script is used.

SELECT \*,

  CASE

    WHEN flag IN ('BEN', 'CIV', 'GHA', 'LBR', 'NGA', 'TGO') THEN 'local'

    ELSE 'foreign'

  END AS local\_vs\_foreign

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CPCO\_Countries\_2020\_2024`

The results table is named GFW\_CPCO\_Countries\_2020\_2024\_Local\_vs\_Foreign\_Flag

## Filtering data related to trawlers with SQL

SELECT

\*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CSRP\_Countries\_2020\_2024\_Local\_vs\_Foreign\_Flag`

WHERE geartype = 'trawlers'

The results table is named GFW\_CSRP\_Countries\_2020\_2024\_Trawlers

SELECT

\*

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CSRP\_Countries\_2020\_2024\_Local\_vs\_Foreign\_Flag`

WHERE geartype = 'trawlers'

The results table is named GFW\_CPCO\_Countries\_2020\_2024\_Trawlers

## Grouping fishing hours for trawlers by month and year using SQL

SELECT month, year, SUM (fishing\_hours) AS total\_fishing\_hours

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CSRP\_Countries\_2020\_2024\_Trawlers`

GROUP BY month, year

ORDER BY total\_fishing\_hours DESC

The results table is named GFW\_CSRP\_Countries\_2020\_2024\_Trawlers\_Month

SELECT month, year, SUM (fishing\_hours) AS total\_fishing\_hours

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CPCO\_Countries\_2020\_2024\_Trawlers`

GROUP BY month, year

ORDER BY total\_fishing\_hours DESC

The results table is named GFW\_CPCO\_Countries\_2020\_2024\_Trawlers\_Month

## Grouping fishing hours for trawlers by country and month for each year using SQL

SELECT country, year, month, SUM (fishing\_hours) AS total\_fishing\_hours

FROM `oceanic-grin-466919-s4.GFW2.GFW\_CSRP\_Countries\_2020\_2024\_Trawlers`

GROUP BY country, year, month

ORDER BY total\_fishing\_hours DESC

The results table is named GFW\_CSRP\_per\_Country\_2020\_2024\_Trawlers\_Month

SELECT country, year, month, SUM (fishing\_hours) AS total\_fishing\_hours

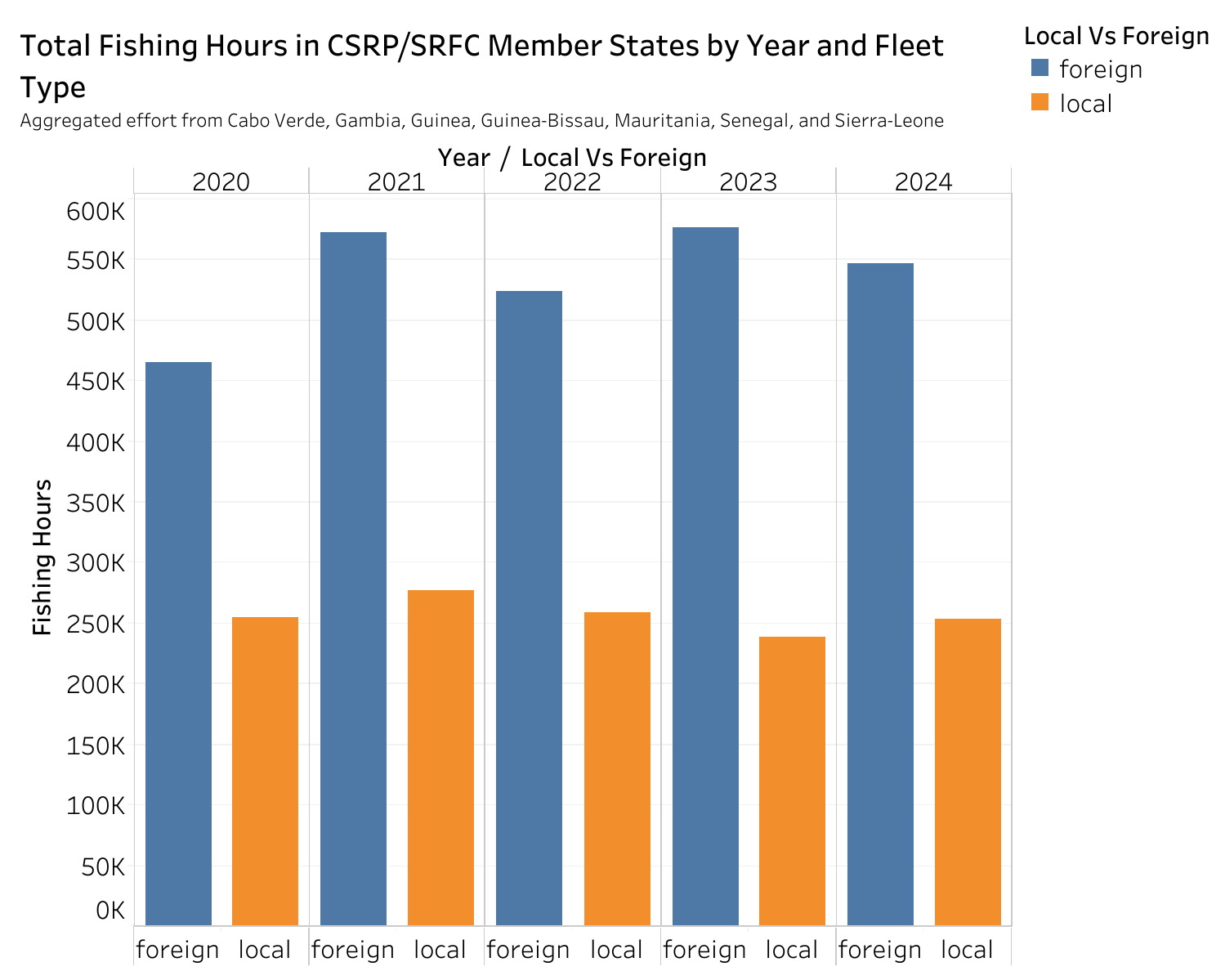
FROM `oceanic-grin-466919-s4.GFW2.GFW\_CPCO\_Countries\_2020\_2024\_Trawlers`

GROUP BY country, year, month

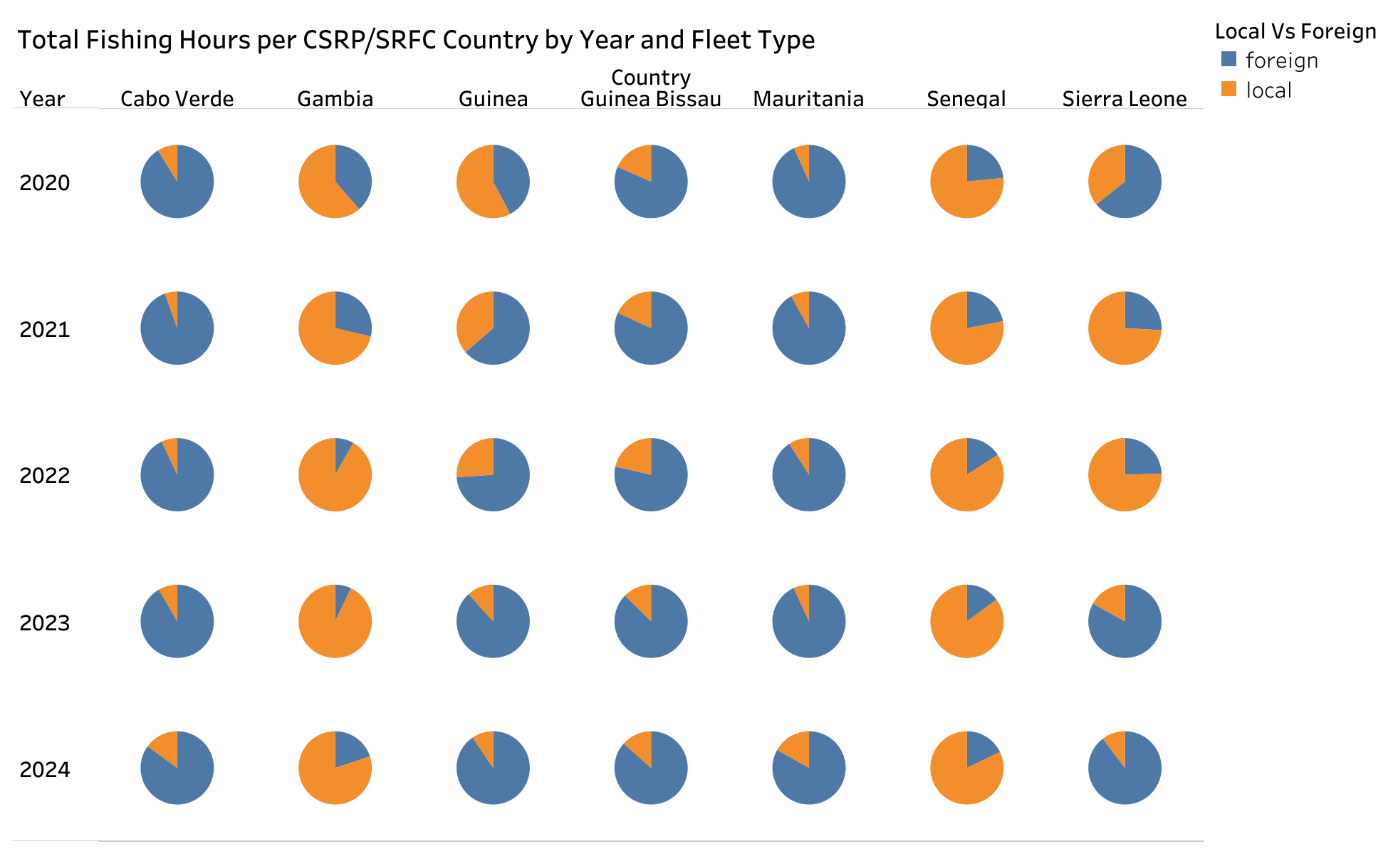
ORDER BY total\_fishing\_hours DESC

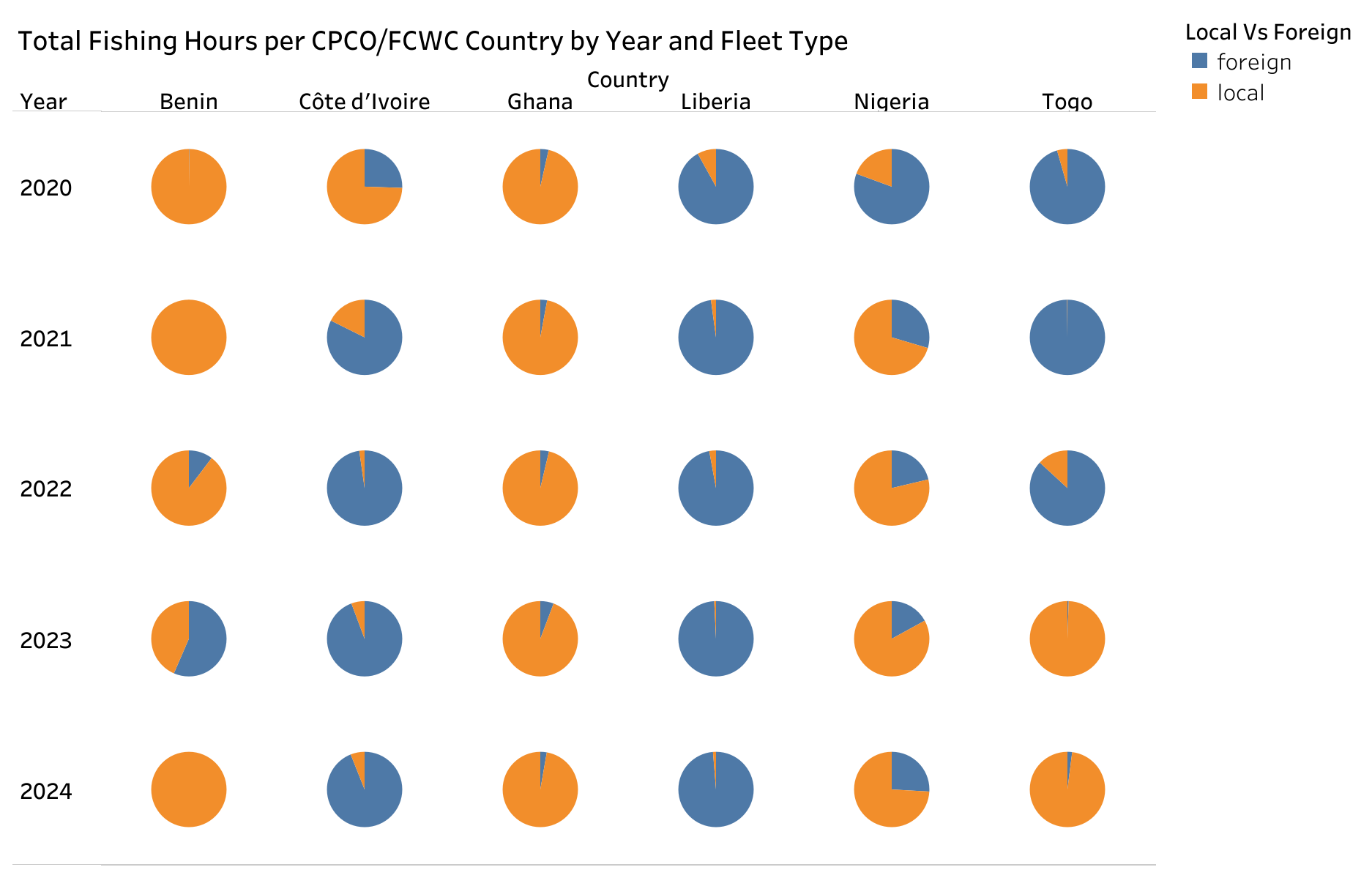
The results table is named GFW\_CPCO\_per\_Country\_2020\_2024\_Trawlers\_Month

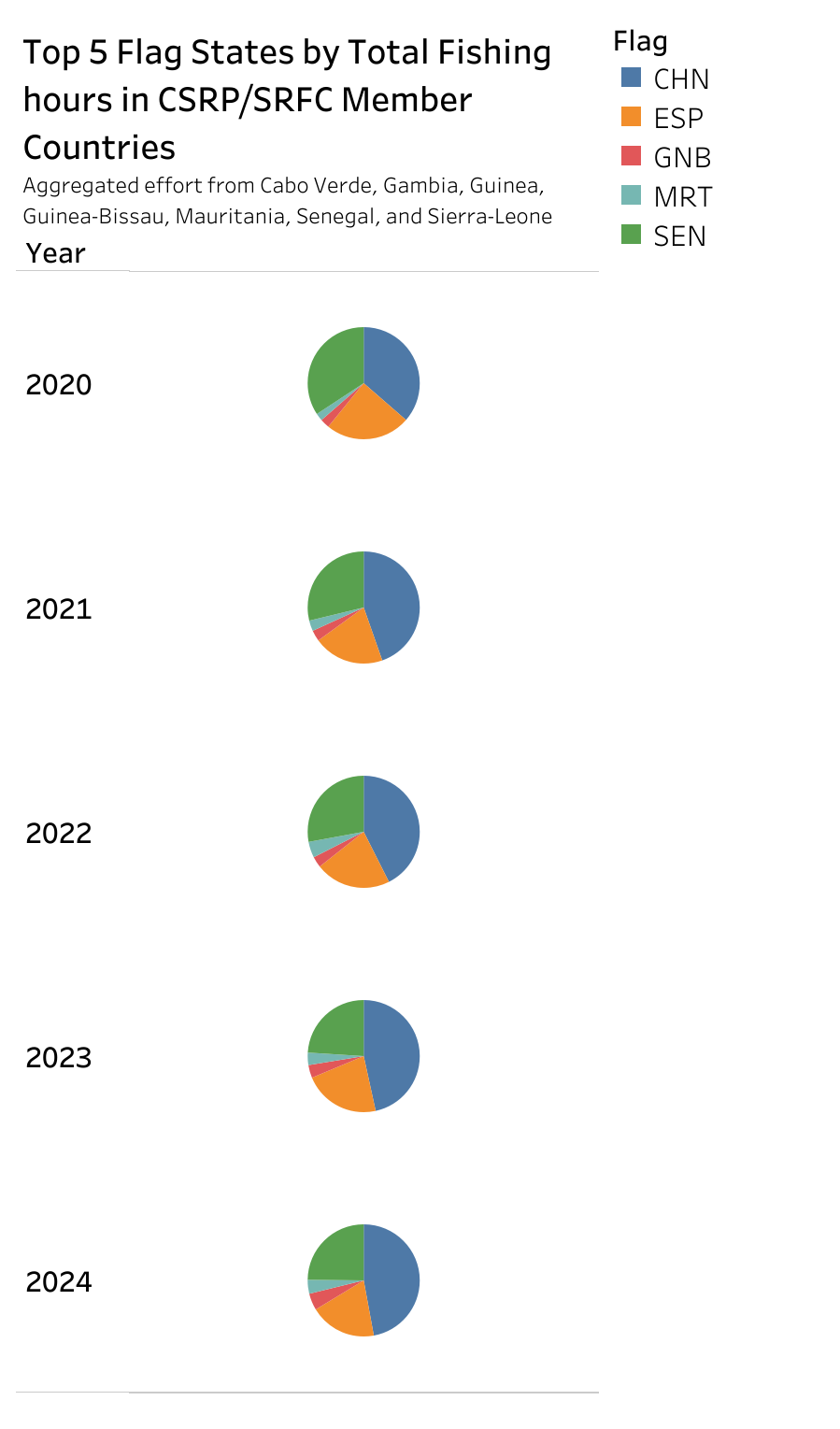
# Tableau

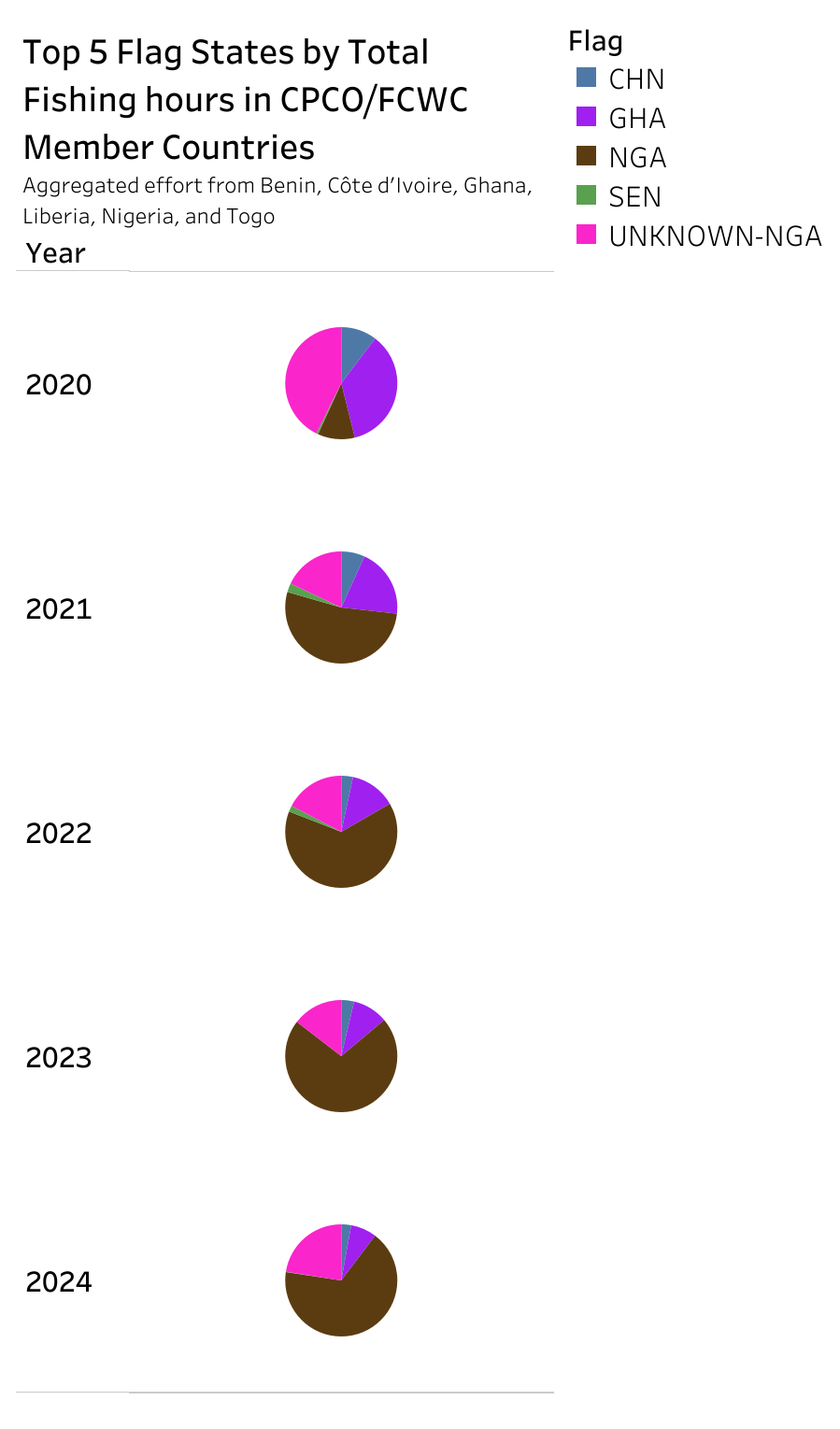


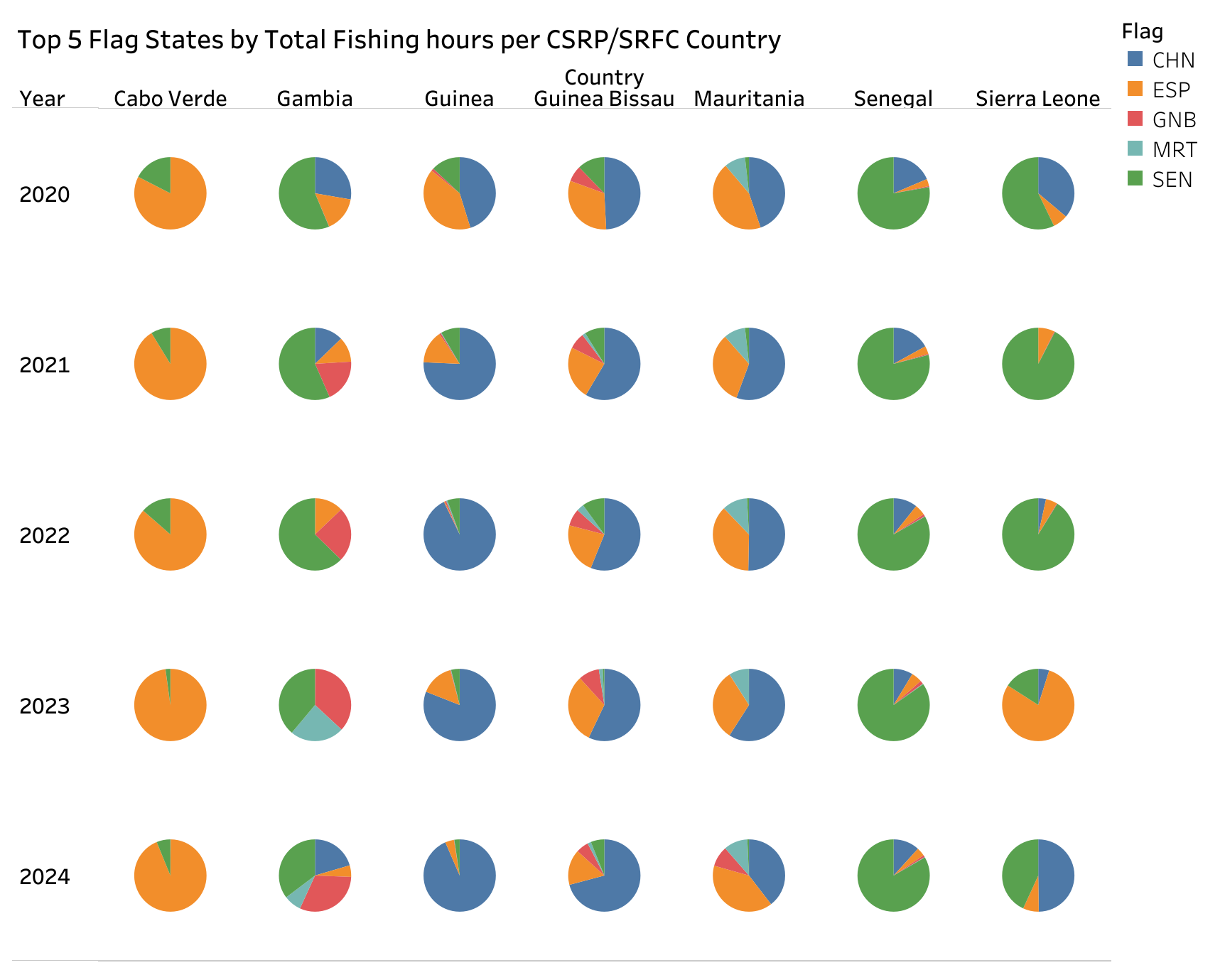


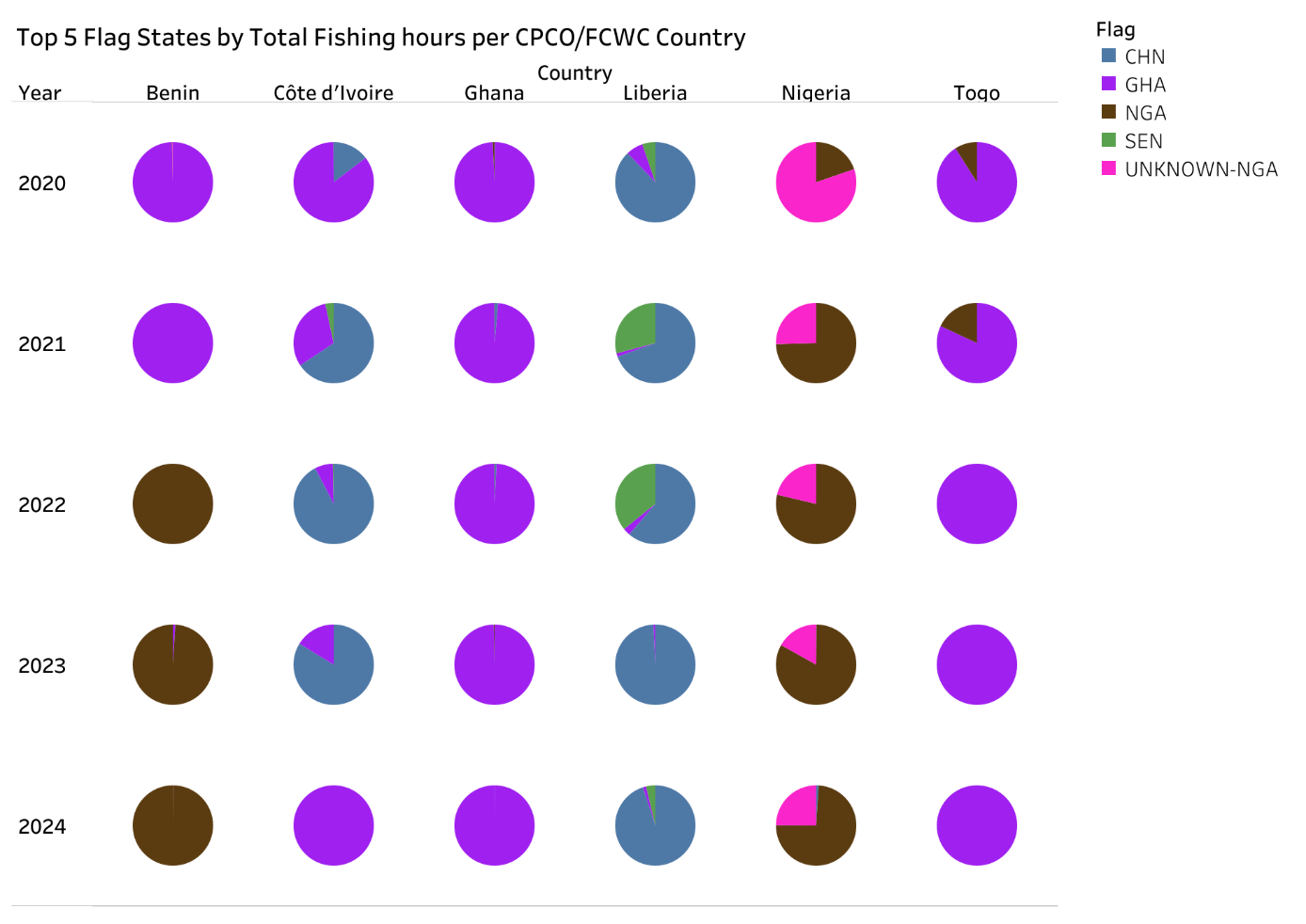


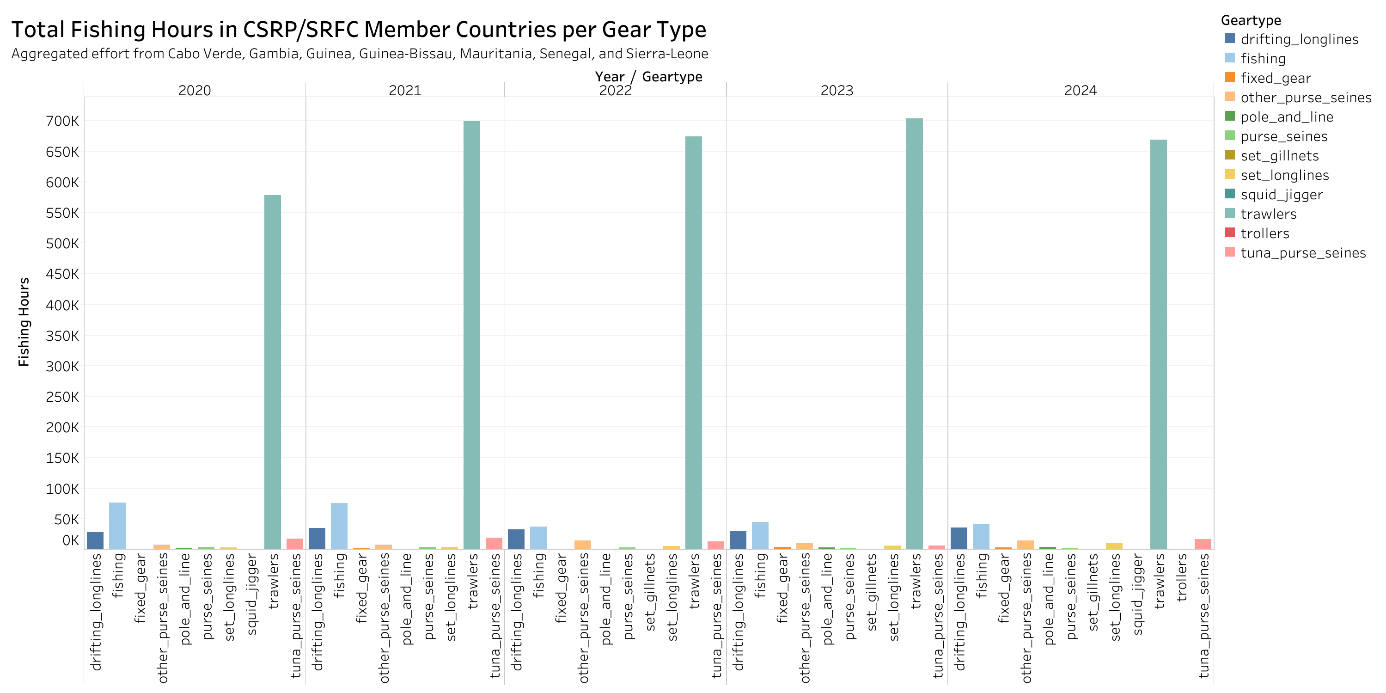


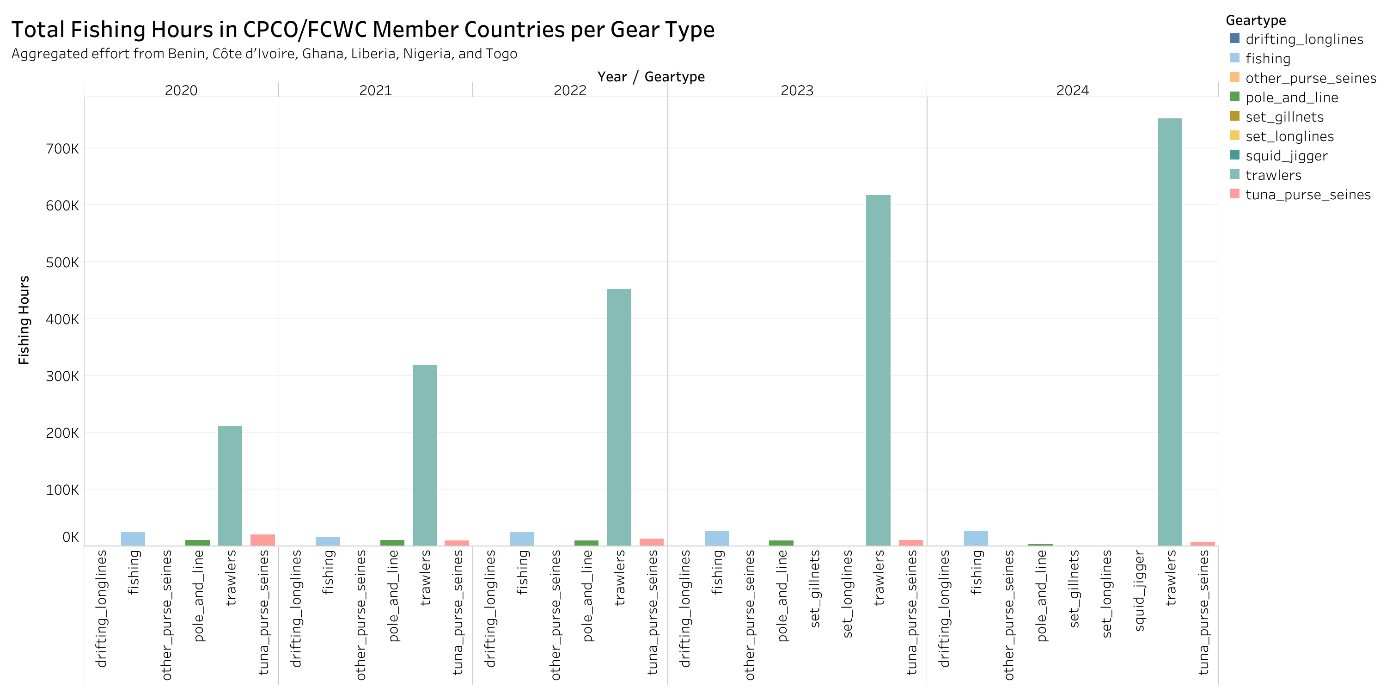


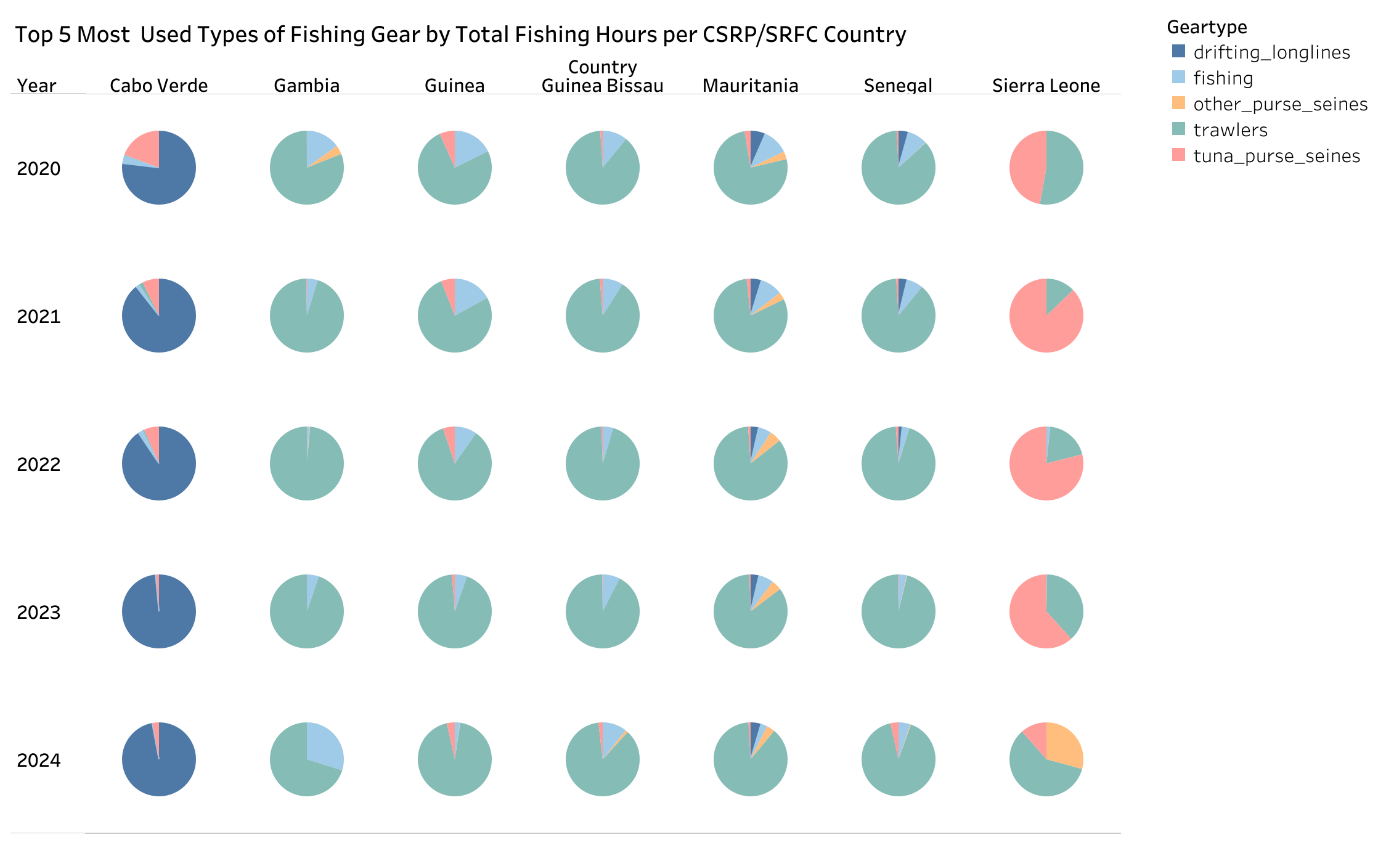


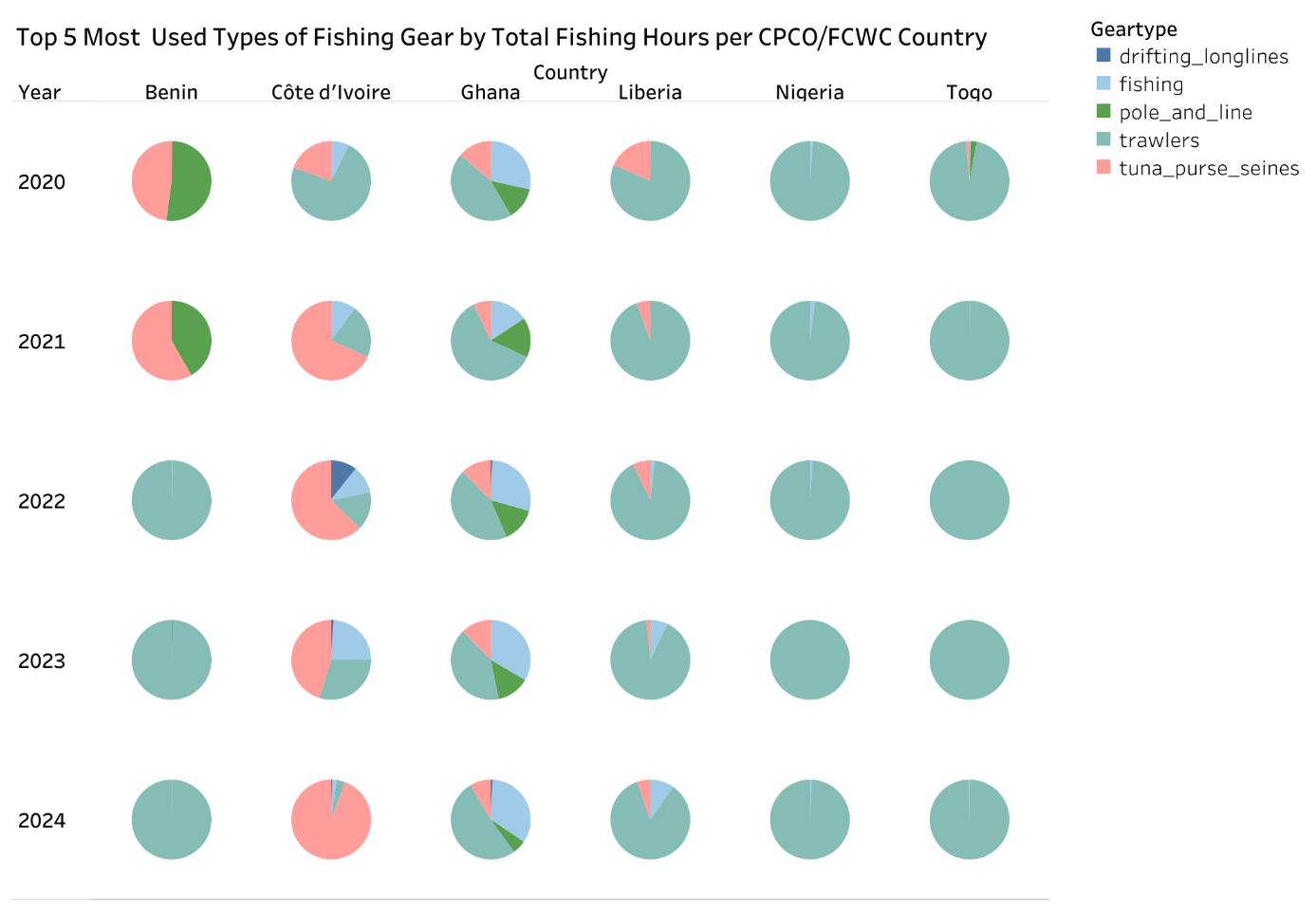


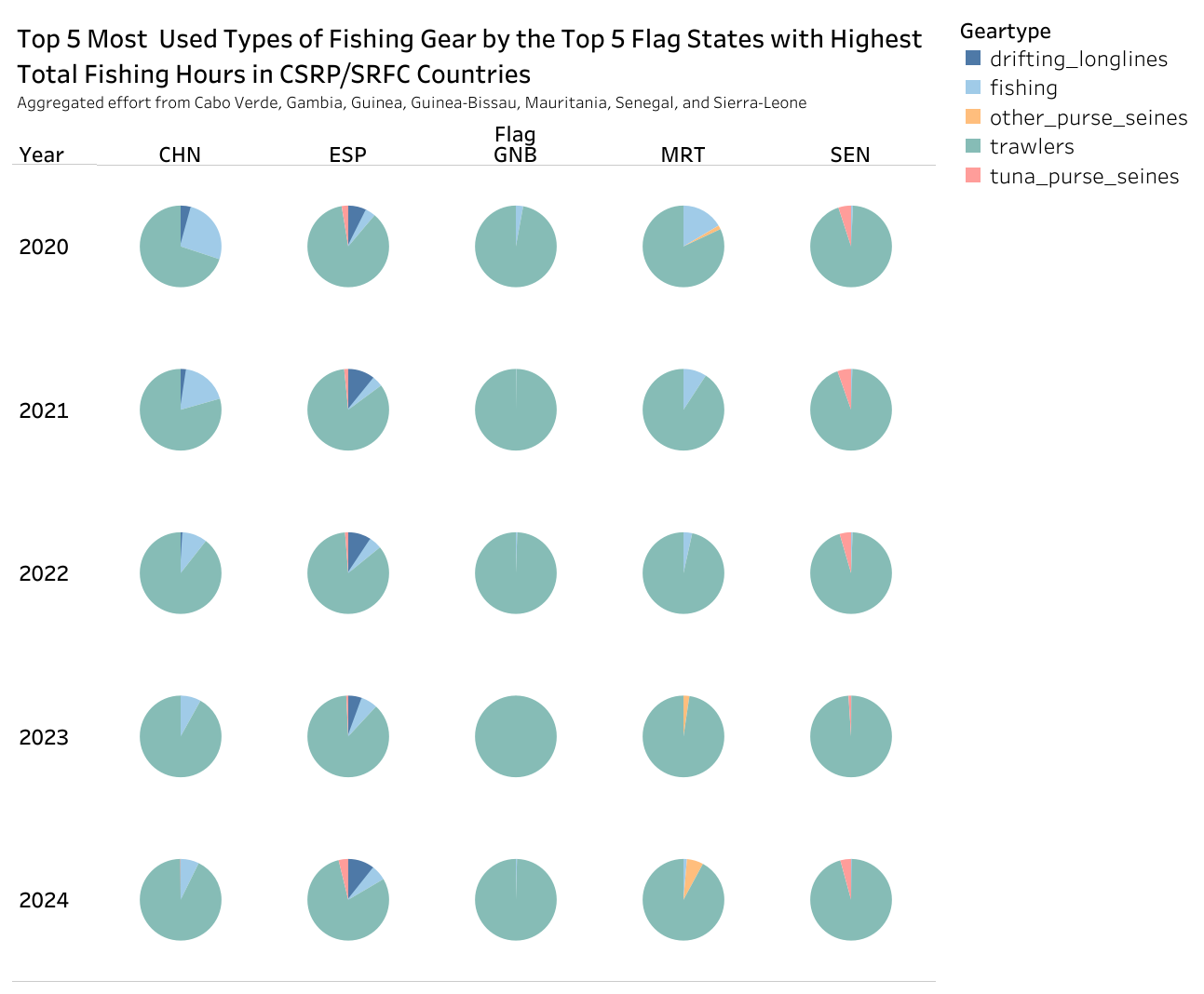


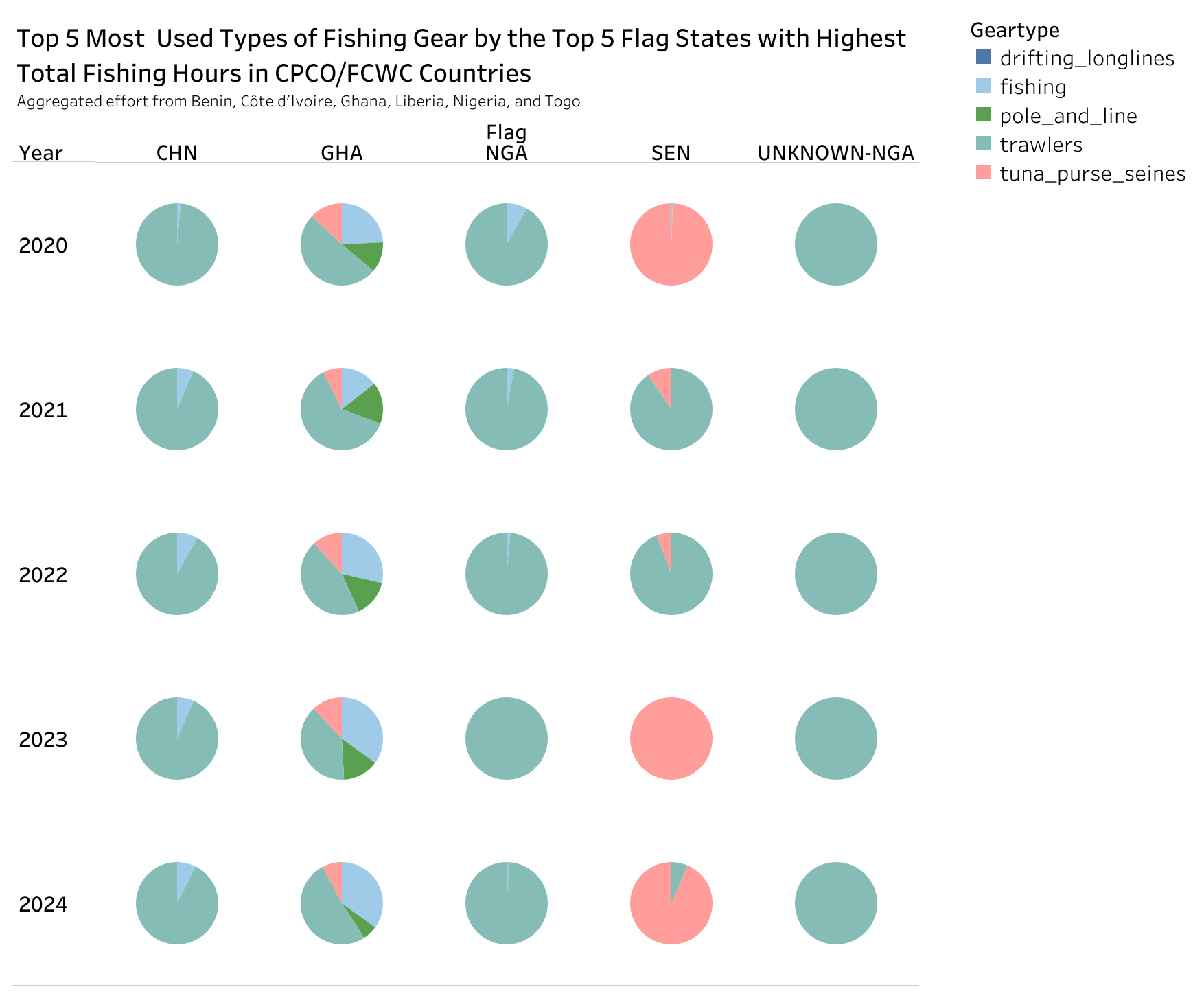


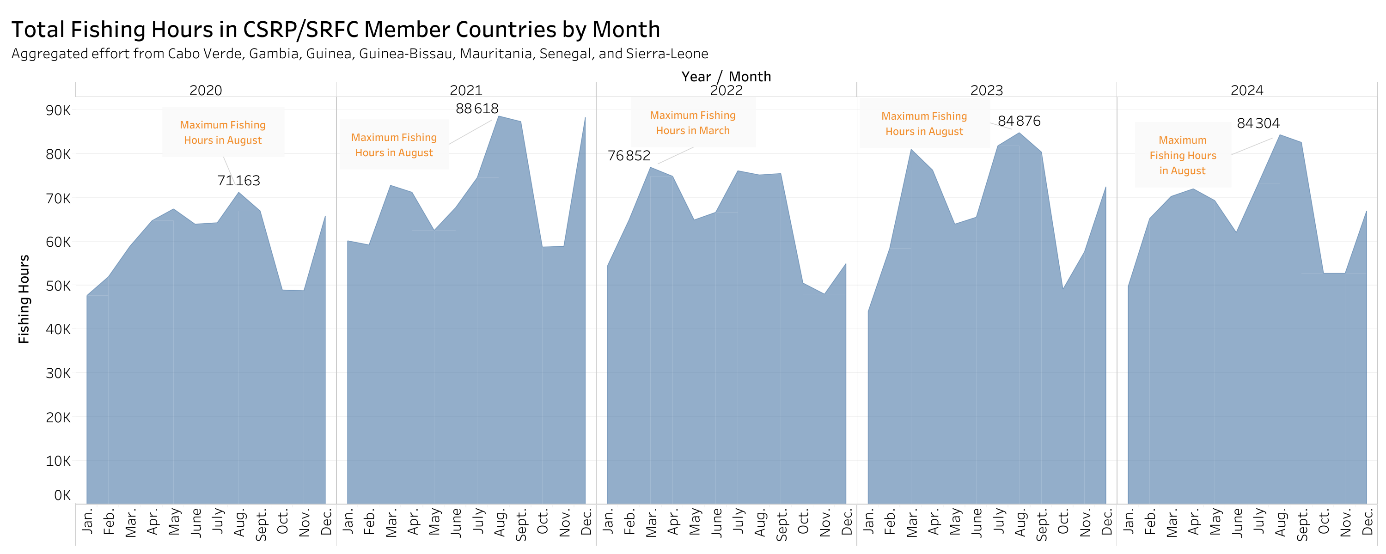


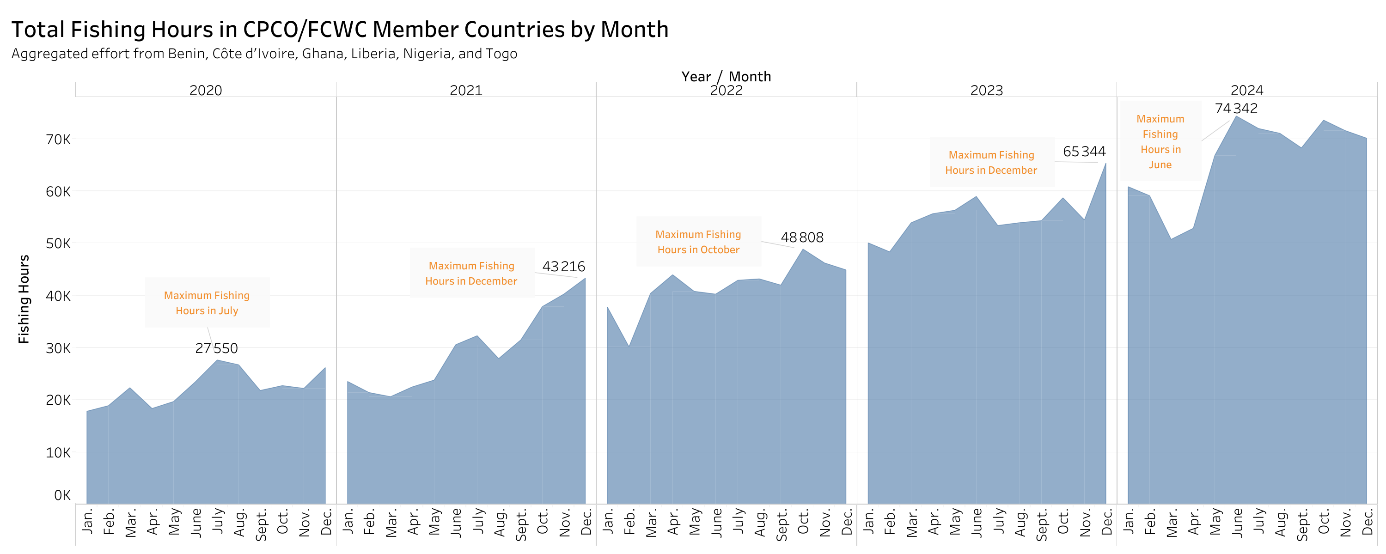


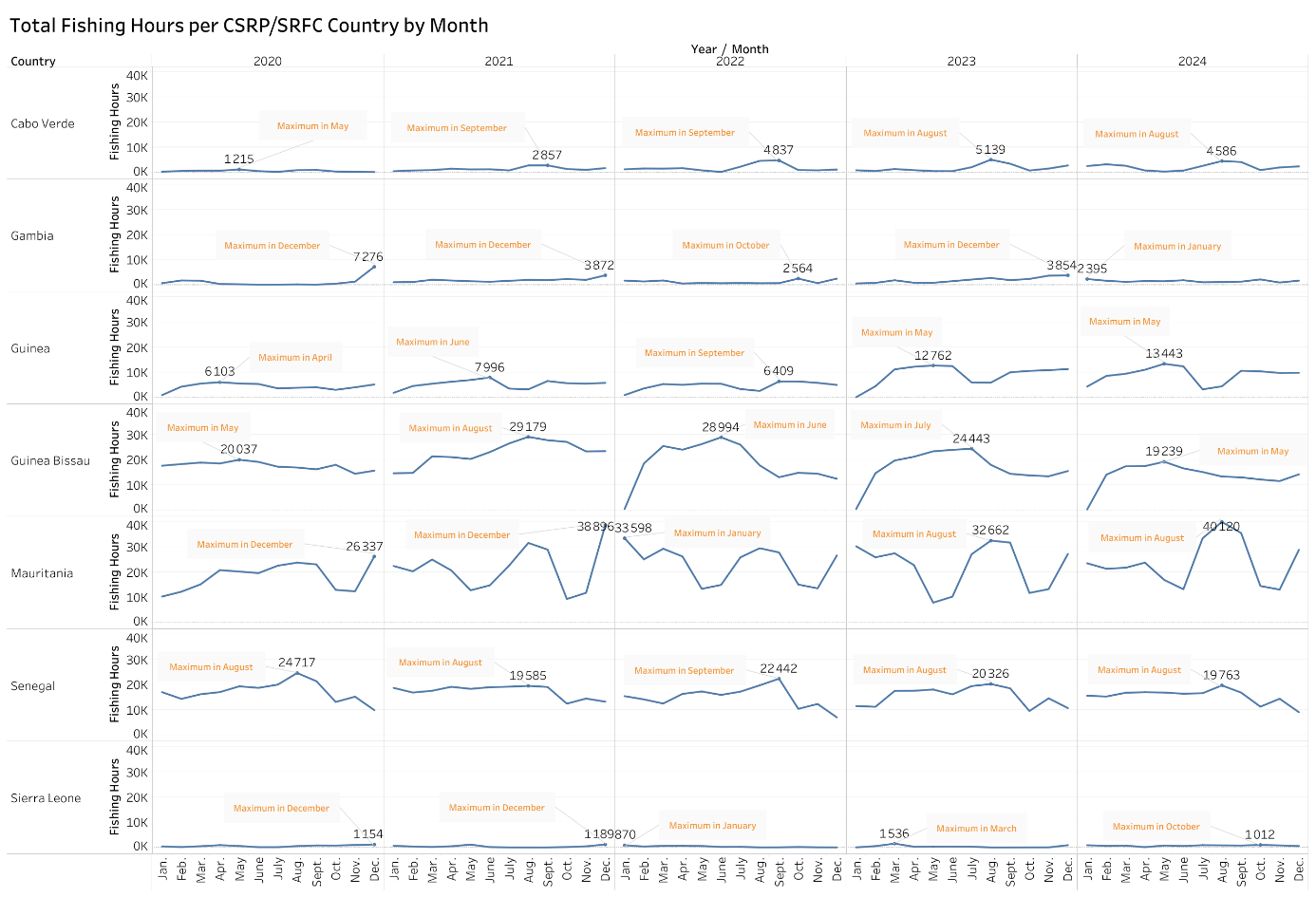


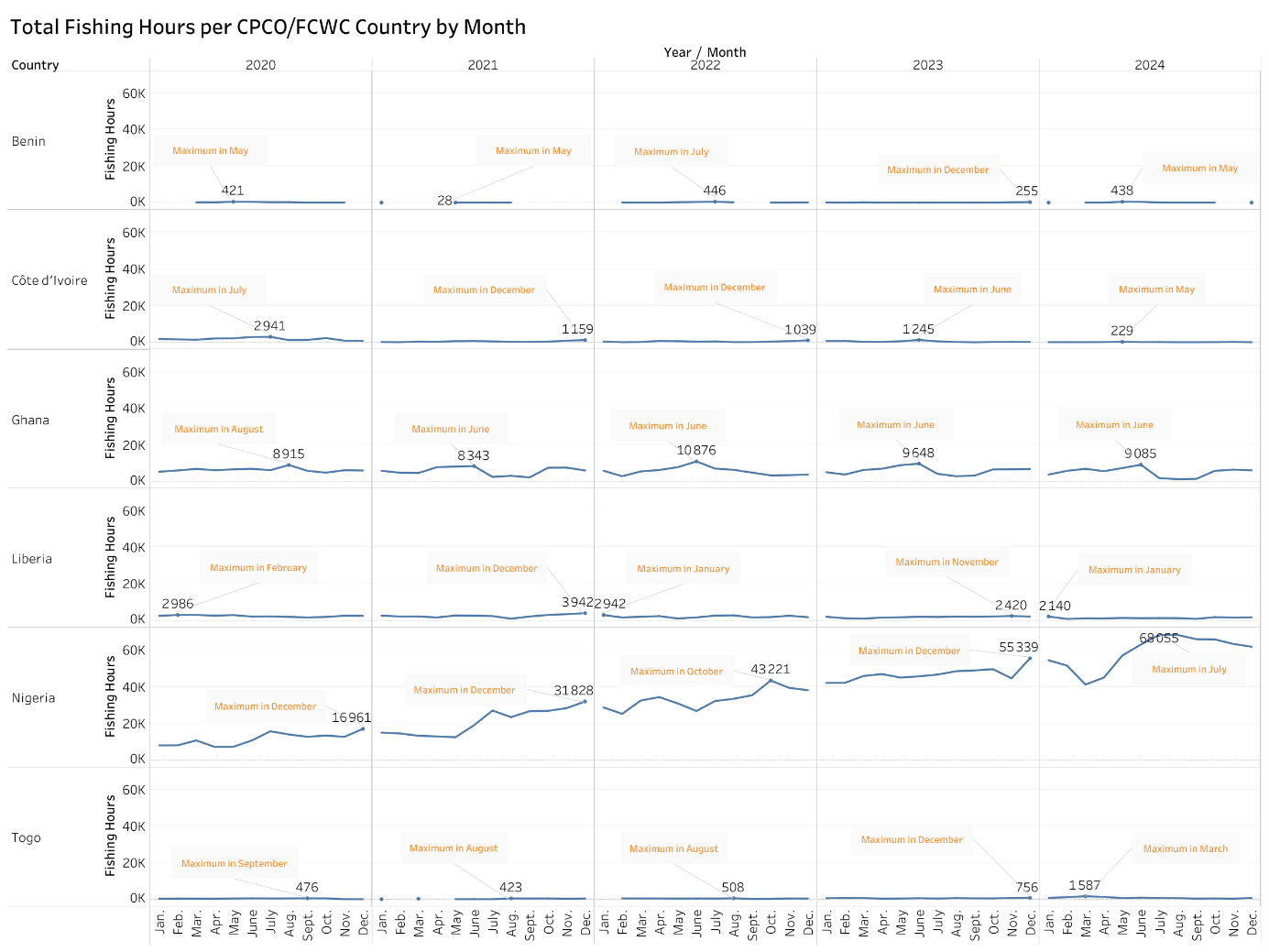


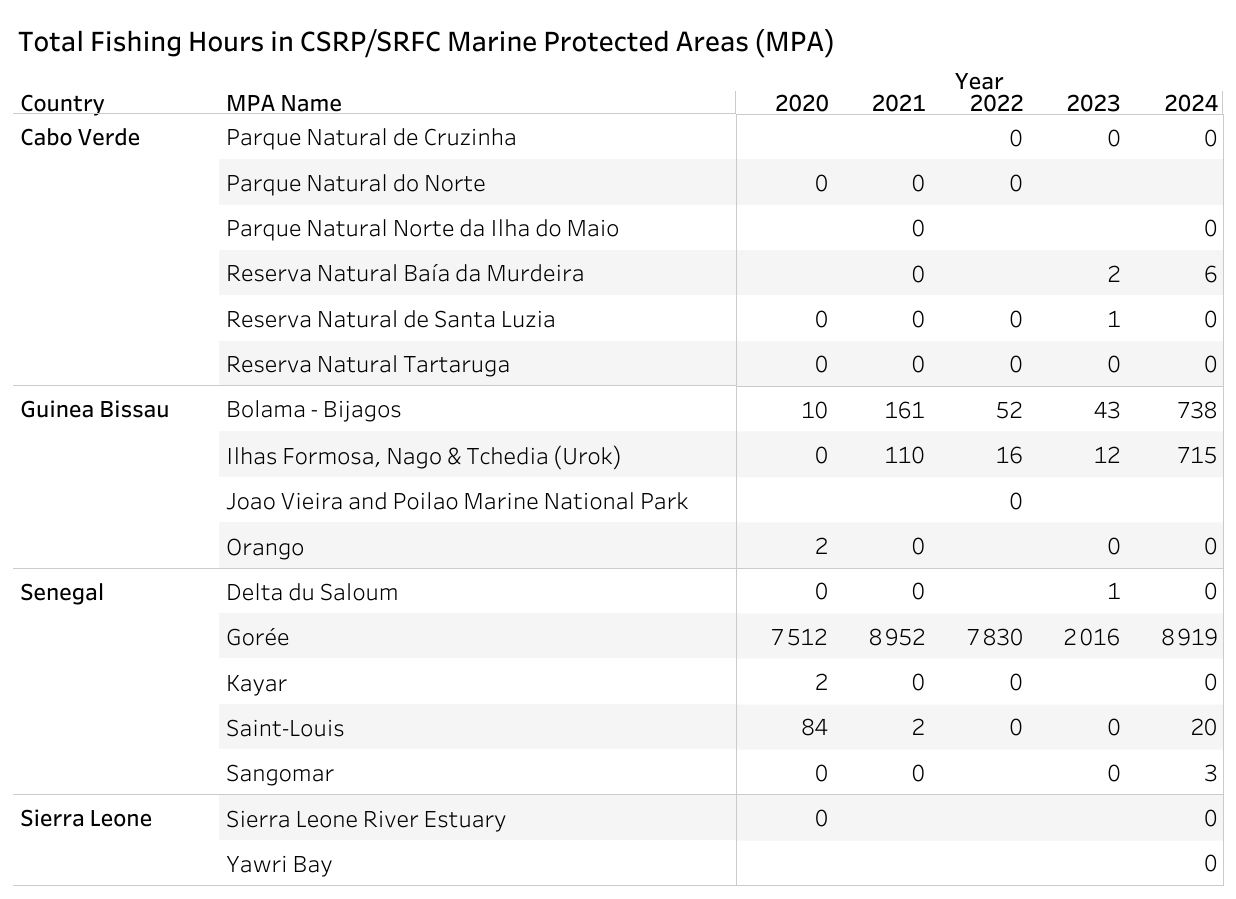












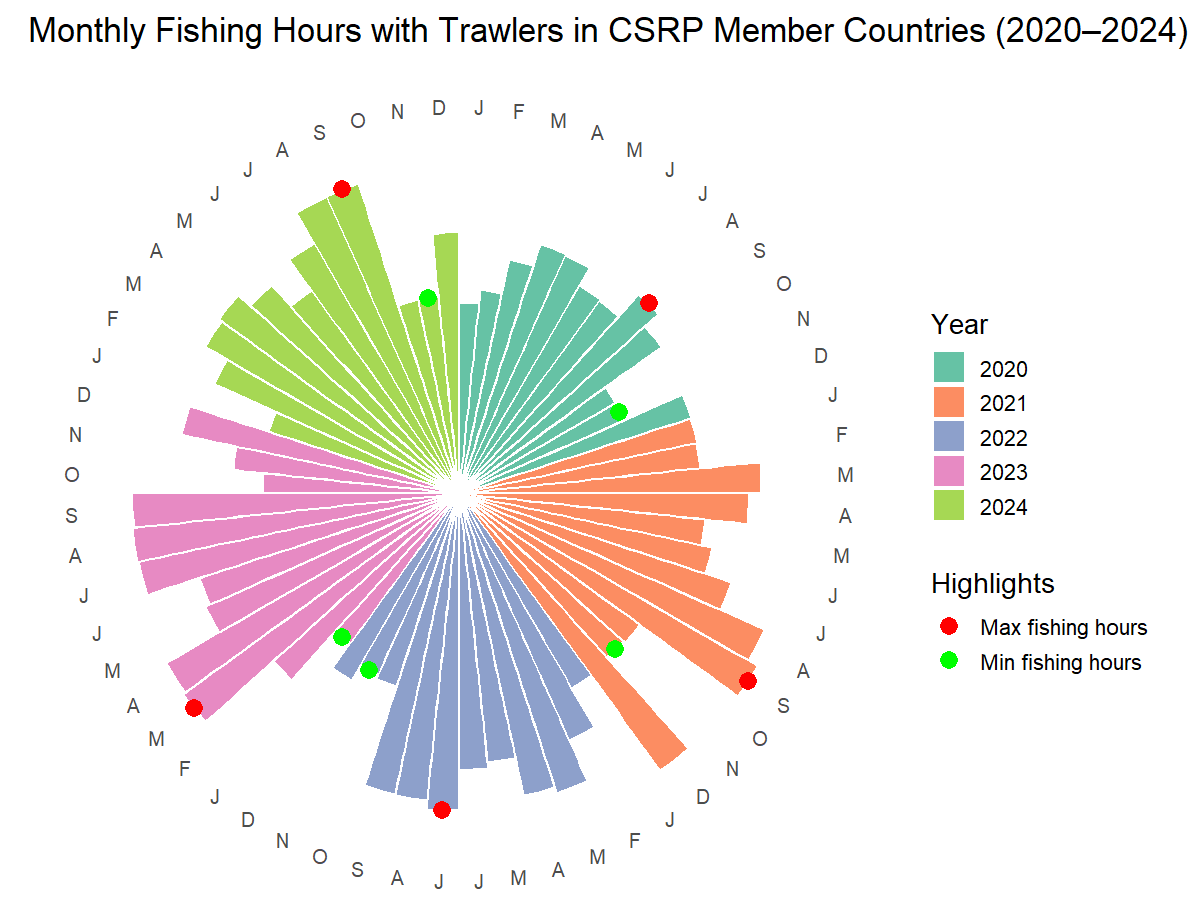
# R

**Monthly Fishing Hours with Trawlers in CSRP Member Countries**

**library**(ggplot2)

**library**(dplyr)

*# Load the data*  
fishing\_data <- **read.csv2**("GFW\_CSRP\_Countries\_2020\_2024\_Trawlers\_Month\_modified.csv")  
  
*# ✅ Ensure Fishing\_Hours is numeric*  
fishing\_data**$**Fishing\_Hours <- **as.numeric**(fishing\_data**$**Fishing\_Hours)  
  
*# Prepare data*  
fishing\_data <- fishing\_data **%>%**  
 **mutate**(  
 Month = **factor**(Month, levels = month.name),  
 Year\_Month = **interaction**(Year, Month, sep = "-")  
 ) **%>%**  
 **arrange**(Year, Month) **%>%**  
 **mutate**(Year\_Month = **factor**(Year\_Month, levels = **unique**(Year\_Month)))  
  
*# Identify max and min month per year*  
highlight\_points <- fishing\_data **%>%**  
 **group\_by**(Year) **%>%**  
 **mutate**(  
 is\_max = Fishing\_Hours **==** **max**(Fishing\_Hours, na.rm = TRUE),  
 is\_min = Fishing\_Hours **==** **min**(Fishing\_Hours, na.rm = TRUE)  
 ) **%>%**  
 **ungroup**() **%>%**  
 **mutate**(highlight\_type = **case\_when**(  
 is\_max **~** "Max fishing hours",  
 is\_min **~** "Min fishing hours",  
 TRUE **~** NA\_character\_  
 ))  
  
*# ✅ Ensure highlight\_points$Fishing\_Hours is numeric (again, just in case)*  
highlight\_points**$**Fishing\_Hours <- **as.numeric**(highlight\_points**$**Fishing\_Hours)  
  
*# Plot*  
**ggplot**() **+**  
 *# Bar layer for years*  
 **geom\_bar**(  
 data = fishing\_data,  
 **aes**(x = Year\_Month, y = Fishing\_Hours, fill = **factor**(Year)),  
 stat = "identity", color = "white", width = 1  
 ) **+**  
 *# Highlight points for max/min months*  
 **geom\_point**(  
 data = **filter**(highlight\_points, **!is.na**(highlight\_type)),  
 **aes**(x = Year\_Month, y = Fishing\_Hours **+** 50, color = highlight\_type),  
 size = 4, shape = 16, inherit.aes = FALSE  
 ) **+**  
 **coord\_polar**(start = 0) **+**  
 **scale\_x\_discrete**(labels = **substr**(**rep**(month.name, times = **length**(**unique**(fishing\_data**$**Year))), 1, 1)) **+**  
 **scale\_fill\_brewer**(palette = "Set2", name = "Year") **+**  
 **scale\_color\_manual**(  
 name = "Highlights",  
 values = **c**(  
 "Max fishing hours" = "red",  
 "Min fishing hours" = "green"  
 )  
 ) **+**  
 **labs**(  
 title = "Monthly Fishing Hours with Trawlers in CSRP Member Countries (2020–2024)",  
 x = "", y = ""  
 ) **+**  
 **guides**(  
 fill = **guide\_legend**(order = 1),  
 color = **guide\_legend**(order = 2, override.aes = **list**(shape = 16, size = 4))  
 ) **+**  
 **theme\_minimal**(base\_size = 14) **+**  
 **theme**(  
 axis.text.x = **element\_text**(size = 10),  
 axis.text.y = **element\_blank**(),  
 panel.grid = **element\_blank**(),  
 axis.ticks = **element\_blank**(),  
 legend.box = "vertical",  
 legend.spacing.y = **unit**(0.3, 'cm')  
 )

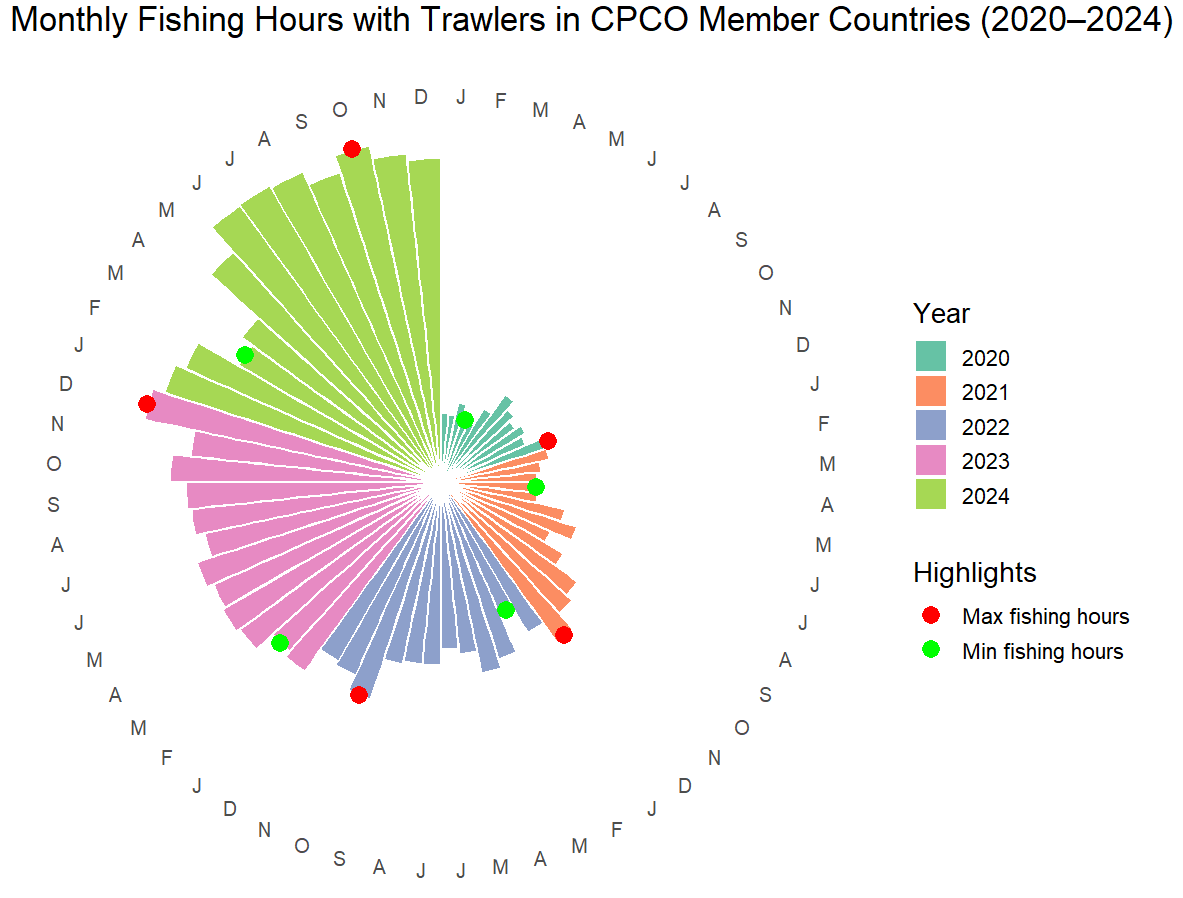


**Monthly Fishing Hours with Trawlers in CPCO Member Countries**

**library**(ggplot2)

**library**(dplyr)

*# Load the data*  
fishing\_data <- **read.csv2**("GFW\_CPCO\_Countries\_2020\_2024\_Trawlers\_Month\_modified.csv")  
  
*# ✅ Ensure Fishing\_Hours is numeric*  
fishing\_data**$**Fishing\_Hours <- **as.numeric**(fishing\_data**$**Fishing\_Hours)  
  
*# Prepare data*  
fishing\_data <- fishing\_data **%>%**  
 **mutate**(  
 Month = **factor**(Month, levels = month.name),  
 Year\_Month = **interaction**(Year, Month, sep = "-")  
 ) **%>%**  
 **arrange**(Year, Month) **%>%**  
 **mutate**(Year\_Month = **factor**(Year\_Month, levels = **unique**(Year\_Month)))  
  
*# Identify max and min month per year*  
highlight\_points <- fishing\_data **%>%**  
 **group\_by**(Year) **%>%**  
 **mutate**(  
 is\_max = Fishing\_Hours **==** **max**(Fishing\_Hours, na.rm = TRUE),  
 is\_min = Fishing\_Hours **==** **min**(Fishing\_Hours, na.rm = TRUE)  
 ) **%>%**  
 **ungroup**() **%>%**  
 **mutate**(highlight\_type = **case\_when**(  
 is\_max **~** "Max fishing hours",  
 is\_min **~** "Min fishing hours",  
 TRUE **~** NA\_character\_  
 ))  
  
*# ✅ Ensure highlight\_points$Fishing\_Hours is numeric (again, just in case)*  
highlight\_points**$**Fishing\_Hours <- **as.numeric**(highlight\_points**$**Fishing\_Hours)  
  
*# Plot*  
**ggplot**() **+**  
 *# Bar layer for years*  
 **geom\_bar**(  
 data = fishing\_data,  
 **aes**(x = Year\_Month, y = Fishing\_Hours, fill = **factor**(Year)),  
 stat = "identity", color = "white", width = 1  
 ) **+**  
 *# Highlight points for max/min months*  
 **geom\_point**(  
 data = **filter**(highlight\_points, **!is.na**(highlight\_type)),  
 **aes**(x = Year\_Month, y = Fishing\_Hours **+** 50, color = highlight\_type),  
 size = 4, shape = 16, inherit.aes = FALSE  
 ) **+**  
 **coord\_polar**(start = 0) **+**  
 **scale\_x\_discrete**(labels = **substr**(**rep**(month.name, times = **length**(**unique**(fishing\_data**$**Year))), 1, 1)) **+**  
 **scale\_fill\_brewer**(palette = "Set2", name = "Year") **+**  
 **scale\_color\_manual**(  
 name = "Highlights",  
 values = **c**(  
 "Max fishing hours" = "red",  
 "Min fishing hours" = "green"  
 )  
 ) **+**  
 **labs**(  
 title = "Monthly Fishing Hours with Trawlers in CPCO Member Countries (2020–2024)",  
 x = "", y = ""  
 ) **+**  
 **guides**(  
 fill = **guide\_legend**(order = 1),  
 color = **guide\_legend**(order = 2, override.aes = **list**(shape = 16, size = 4))  
 ) **+**  
 **theme\_minimal**(base\_size = 14) **+**  
 **theme**(  
 axis.text.x = **element\_text**(size = 10),  
 axis.text.y = **element\_blank**(),  
 panel.grid = **element\_blank**(),  
 axis.ticks = **element\_blank**(),  
 legend.box = "vertical",  
 legend.spacing.y = **unit**(0.3, 'cm')  
 )



**Monthly Fishing Hours with Trawlers in CSRP per Country, Case of Senegal (The same R code was applied for the other countries and those of CPCO)**

**library**(ggplot2)

**library**(dplyr)

*# Load the data*  
fishing\_data <- **read.csv2**("GFW\_CSRP\_Senegal\_2020\_2024\_Trawlers\_Month.csv")  
  
*# ✅ Ensure Fishing\_Hours is numeric*  
fishing\_data**$**Fishing\_Hours <- **as.numeric**(fishing\_data**$**Fishing\_Hours)  
  
*# Prepare data*  
fishing\_data <- fishing\_data **%>%**  
 **mutate**(  
 Month = **factor**(Month, levels = month.name),  
 Year\_Month = **interaction**(Year, Month, sep = "-")  
 ) **%>%**  
 **arrange**(Year, Month) **%>%**  
 **mutate**(Year\_Month = **factor**(Year\_Month, levels = **unique**(Year\_Month)))  
  
*# Identify max and min month per year*  
highlight\_points <- fishing\_data **%>%**  
 **group\_by**(Year) **%>%**  
 **mutate**(  
 is\_max = Fishing\_Hours **==** **max**(Fishing\_Hours, na.rm = TRUE),  
 is\_min = Fishing\_Hours **==** **min**(Fishing\_Hours, na.rm = TRUE)  
 ) **%>%**  
 **ungroup**() **%>%**  
 **mutate**(highlight\_type = **case\_when**(  
 is\_max **~** "Max fishing hours",  
 is\_min **~** "Min fishing hours",  
 TRUE **~** NA\_character\_  
 ))  
  
*# ✅ Ensure highlight\_points$Fishing\_Hours is numeric (again, just in case)*  
highlight\_points**$**Fishing\_Hours <- **as.numeric**(highlight\_points**$**Fishing\_Hours)  
  
*# Plot*  
**ggplot**() **+**  
 *# Bar layer for years*  
 **geom\_bar**(  
 data = fishing\_data,  
 **aes**(x = Year\_Month, y = Fishing\_Hours, fill = **factor**(Year)),  
 stat = "identity", color = "white", width = 1  
 ) **+**  
 *# Highlight points for max/min months*  
 **geom\_point**(  
 data = **filter**(highlight\_points, **!is.na**(highlight\_type)),  
 **aes**(x = Year\_Month, y = Fishing\_Hours **+** 50, color = highlight\_type),  
 size = 4, shape = 16, inherit.aes = FALSE  
 ) **+**  
 **coord\_polar**(start = 0) **+**  
 **scale\_x\_discrete**(labels = **substr**(**rep**(month.name, times = **length**(**unique**(fishing\_data**$**Year))), 1, 1)) **+**  
 **scale\_fill\_brewer**(palette = "Set2", name = "Year") **+**  
 **scale\_color\_manual**(  
 name = "Highlights",  
 values = **c**(  
 "Max fishing hours" = "red",  
 "Min fishing hours" = "green"  
 )  
 ) **+**  
 **labs**(  
 title = "Monthly Fishing Hours with Trawlers per CSRP Country, Case of Senegal (2020–2024)",  
 x = "", y = ""  
 ) **+**  
 **guides**(  
 fill = **guide\_legend**(order = 1),  
 color = **guide\_legend**(order = 2, override.aes = **list**(shape = 16, size = 4))  
 ) **+**  
 **theme\_minimal**(base\_size = 14) **+**  
 **theme**(  
 axis.text.x = **element\_text**(size = 10),  
 axis.text.y = **element\_blank**(),  
 panel.grid = **element\_blank**(),  
 axis.ticks = **element\_blank**(),  
 legend.box = "vertical",  
 legend.spacing.y = **unit**(0.3, 'cm')  
 )

