

A few historical reflections

World War I was the climax of the battleship era for naval warfare. The [Battle of Jutland in 1916](#) was only the third ever – and last – full scale clash of steel battleships (the first two were a few years earlier in the Russo-Japanese war of 1904-1905). By the time of World War II, Germany did not have a large surface fleet, and the conflict with Japan was dominated by a new form of naval asset, the aircraft carrier.

In World War I, the UK and its allies dominated Germany's fleet on paper if all the assets were matched against each other in an orderly fashion. However, the Royal Navy faced a requirement to assert itself globally to protect its country's maritime lifeline, while also threatened by the German High Seas [fleet in being](#) within a day's steaming of the UK homeland. This situation led to difficulties in translating naval dominance into strategic outcomes. The UK struggled with limited success to leverage its power through blockade and (in one, large, controversial campaign) movement of troops to open a new front; but if at any moment it lost its dominance and hence the ability to control surface and submarine raiders preying on its commercial shipping, it would not be able to stay in the war.

This is what Winston Churchill (who was First Lord of the Admiralty when the war broke out) meant when he wrote later that British Admiral Sir John Jellicoe (Admiral of the Grand Fleet responsible for keeping the German fleet in check) was "the only man on either side who could lose the war in an afternoon". Even a dramatic win in a meeting of the two battle fleets would not win the war for the UK, but a dramatic loss could lose it.

The UK strategy was to try to engineer a decisive confrontation in the North Sea on favourable terms as quickly as possible, to free up assets for protecting maritime trade from commerce raiders and submarines. Whereas the German strategy was to postpone such a confrontation and concentrate on throttling the UK's mercantile marine; all the while leaving enough of a plausible threat in UK home waters to keep the UK Grand Fleet as large and anxious as possible.

I guess a motivation for a map like mine above is try to give at least a taste of the scale of the Royal Navy's global coverage at the time. While the 314 ships for which I have data is only a fraction of the ships that actually saw service, it's enough to get a good global picture.

I did find a few things interesting in actually watching my map once it was finished. Of course, I'd expected to see a wide range of operations with focus points on the historical UK naval power locations of Scapa Flow, Gibraltar, Malta, Cape Town and Alexandria; but I hadn't expected to see gunboats and other vessels operating far up-river in mainland China. Similarly, I was vaguely aware of expeditions to German colonies in what is now Samoa, Solomon Islands and Papua New Guinea but it was still a surprise to see the various (incomplete) coloured dots moving around in that area at different times.

One bit of value-add from me was to highlight the locations of key naval battles that did take place, including some of those by smaller groups of ships. The Battle of Coronel off the west coast of South America in 1914 and its sequel a few weeks later in the Falklands show up nicely, for example. In general, an annotation layer is important for turning a statistical graphic into a powerful communication tool, and never more so than in an animated map.

Making the map

All of that is by the by. How did I go about building this map? The R code to do so is [in its own](#)

[repository on GitHub](#). The code extracts below aren't self-sufficient, you'd need to clone the repository in full to make it work.

Webscraping

Getting hold of the data from the old static website was a reasonably straightforward webscraping job. Each ship gets its own page, and there is a single index page with links to all of the ships' pages. The pages themselves are fairly tidy and probably have been generated by a database somewhere (for example, dates are all in the same format). This is a fairly common pattern in webscraping; it works well if you're just re-creating data that is in a database somewhere, inaccessible to you, but which is apparent in the structure of the web pages you're getting data from.

Here's a chunk that grabs all the links to ship-specific pages. It stores the links in a character vector `all_links`, and sets up an empty list `vessel_logs_l` which is going to store, one element per page, the results of scraping each ship's page.

```
#-----Main page and all the
links-----
# Main page, which has links to all the ship-specific pages
url1 <- "https://www.naval-history.net/OWShips-LogBooksWW1.htm"
p1 <- read_html(url1)

all_links <- p1 %>%
  html_nodes("a") %>%
  str_extract("OWShips-WW1.*?\\.htm") %>%
  unique()

vessel_logs_l <- list()
all_links <- all_links[!is.na(all_links)]

# There is an error: JMS Welland, should be HMS Welland. URL
# is correct but
#
```

Now here's the main loop, which iterates through each ship's page. It extracts the vessel's name (which can be deduced from the URL - it's the 16th character to the character 5 from the end of the URL); the type of vessel (destroyer, sloop etc) which can be deduced from the HTML page title; and then the text of the log entries themselves, in the form of a big column of character strings. This is fairly simple with a few regular expressions. The trick in the pattern used below is to create TRUE/FALSE vectors that in effect label each line of the log: is this line the position of the script? is it the weather? is the description of the position (ie the English name of the location)? etc. Then these columns are used as part of the process to turn the data into one row per day (for each specified ship), with summary columns containing relevant extracts from the various lines of the log entry.

```
#-----Main loop - one ship at a
time-----

# Loop from "i" to the end means if we get interrupted we can
start
```

```

# the loop again from wherever it got up to. This loop takes
about 30-60 minutes
# to run through all 314 ships.
i = 1
for(i in 1:length(all_links)){
  cat(i, " ")
  the_url <- glue("https://www.naval-
history.net/{all_links[i]}")

  the_vessel <- str_sub(all_links[i], 16, -5) %>%
    str_replace("_", " ")

  this_ship_page <- read_html(the_url)

  vessel_type <- this_ship_page %>%
    html_nodes("title") %>%
    html_text() %>%
    drop_rn() %>%
    str_replace(" - British warships of World War 1", "") %>%
    str_replace(" - British Empire warships of World War 1",
    "") %>%
    str_replace(" - British auxiliary ships of World War 1",
    "") %>%
    str_replace(" - logbooks of British warships of World War
1", "") %>%
    str_replace(".*", "", "")

  txt <- this_ship_page %>%
    html_nodes("p") %>%
    html_text()

  d <- tibble(txt) %>%
    mutate(txt2 = drop_rn(txt)) %>%
    mutate(is_date = grepl("^[0-9]+ [a-zA-Z]+ 19[0-9][0-9]$",
txt2),
          entry_id = cumsum(is_date),
          is_position = grepl("^Lat.*Long", txt2),
          is_position_description = lag(is_date),
          is_weather = grepl("^Weather", txt2),
          last_date = ifelse(is_date, txt2, NA),
          last_date = as.Date(last_date, format = "%d %b
%Y")) %>%
    fill(last_date) %>%
    filter(entry_id >= 1)

  vessel_logs_l[[i]] <- d %>%
    group_by(entry_id) %>%
    summarise(date = unique(last_date),
              position = txt2[is_position],
              # position_description is a bit of a guess,
sometimes there are 0,
              # 1 or 2 of them (not necessarily correct), so

```

```

we just take the
      # first one and hope for the best.
      position_description =
txt2[is_position_description][1],
      weather = txt2[is_weather][1],
      log_entry = paste(txt2, collapse = "\n"),
      .groups = "drop") %>%
  mutate(url = the_url,
         vessel = the_vessel,
         vessel_type = vessel_type,
         vessel_id = i,
         lat = str_extract(position, "Lat.*?\\.[0-9]+"),
         long = str_extract(position, "Lon.*?\\.[0-9]+"),
         lat = as.numeric(gsub("Lat ", "", lat)),
         long = as.numeric(gsub("Long ", "", long)),
         weather = str_squish(gsub("Weather:", "", weather),
                                ignore.case = TRUE)))
}

# save version with all the text (about 25 MB)
vessel_logs <- bind_rows(vessel_logs_l)
save(vessel_logs, file = "data/vessel_logs.rda")

# Cut down version of the data without the original log text
# (about 2MB):
vessel_logs_sel <- select(vessel_logs, -log_entry)
save(vessel_logs_sel, file = "data/vessel_logs_sel.rda")

```

Drawing the map

Drawing each daily frame of the map itself is surprisingly easy, thanks to the wonders of `ggplot2` and neat coordinates transformation offered by simple features and `sf`. The “layered grammar of graphics” philosophy of Wickham’s `ggplot2` really comes into its own here, providing the ability to neatly specify:

- a default dataset
- six different layers including land borders, solid points for each ship, hollow circular points for any battles present on the day, text annotating those battles, and text annotations for today’s date and the description of the stage of the war
- a coordinate system to give a good presentation of the round world in a rectangle of real estate
- scales to govern the colours of the ships
- fine thematic control of background and text colours, fonts, etc

Skipping over a chunk of data management to define the times and labels used for the various annotations, here is the code for the actual drawing of the map with a single day’s data:

```

m <- ships_data %>%
  ggplot(aes(x = long, y = lat)) +
  borders(fill = "grey", colour = NA) +
  geom_point(aes(colour = vessel_type_lumped), size = 0.8)
+
  geom_point(data = battle_data,

```

```

      aes(size = point_size),
      shape = 1, colour = battle_col) +
geom_text(data = battle_data,
      aes(label = battle),
      family = main_family,
      hjust = 0,
      nudge_x = 5,
      size = 2,
      colour = battle_col) +
scale_size_identity() +
coord_sf() +
theme_void(base_family = main_family) +
# The date, in the South Atlantic:
annotate("text", x = 22, y = -64, label =
format(the_date, '%e %B %Y'),
      colour = date_col, hjust = 1) +
# Summary text next the date:
annotate("text", x = 24, y = -63,
      label = glue("{date_sum_text}:
{unique(ships_data$phase)}"),
      colour = comment_col,
      hjust = 0, size = 2.5) +
scale_colour_manual(values = pal, labels = names(pal),
drop = FALSE) +
labs(title = glue("Daily locations of Royal Navy Ships
1914 to 1919"),
      colour = "",
      caption = str_wrap("Locations of 314 UK Royal Navy
from log books compiled by
naval-history.net; map by freerangestats.info. Ships
that survived the
war and that travelled out of UK home waters were
more likely to be selected
for transcription, which was by volunteers for the
'Zooniverse Old Weather Project'."),
      # margin() theme on left and
right doesn't work for plot.caption so we add our own:
      width = 180, indent = 2, exdent =
2)) +
theme(legend.position = "bottom",
      plot.title = element_text(family = "Sarala", hjust
= 0.5),
      plot.caption = element_text(colour = "grey70", size
= 8, hjust = 0),
      legend.spacing.x = unit(0, "cm"),
      legend.text = element_text(hjust = 0, margin =
margin(l = -2, r = 15)),
      legend.background = element_rect(fill = sea_col,
colour = NA),
      panel.background = element_rect(fill = sea_col,
colour = NA))

```

Making movies

The loop that this is within draws one frame for each day, 2000 pixels wide in 16:9 ratio. I used Image Magick to create an animated GIF out of a subset of 40 of those frames, and Windows Video Editor to make the full length movie above.

So that's all, folks. Just a tiny slice of history. Oh, if you think that 6.5 minutes of video is long to watch, imagine what it was like living through. It didn't begin or end there, either. We might think 2020 has been tough, but I'd still rather what we've just gone through than many of the years in the first half of last century.