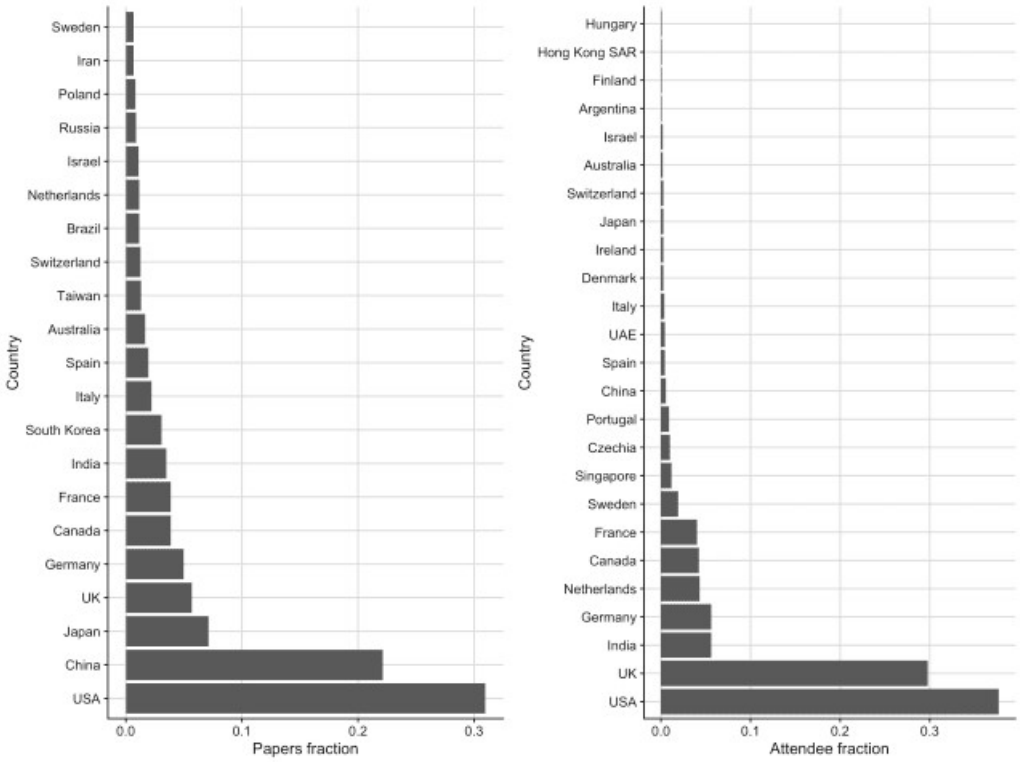
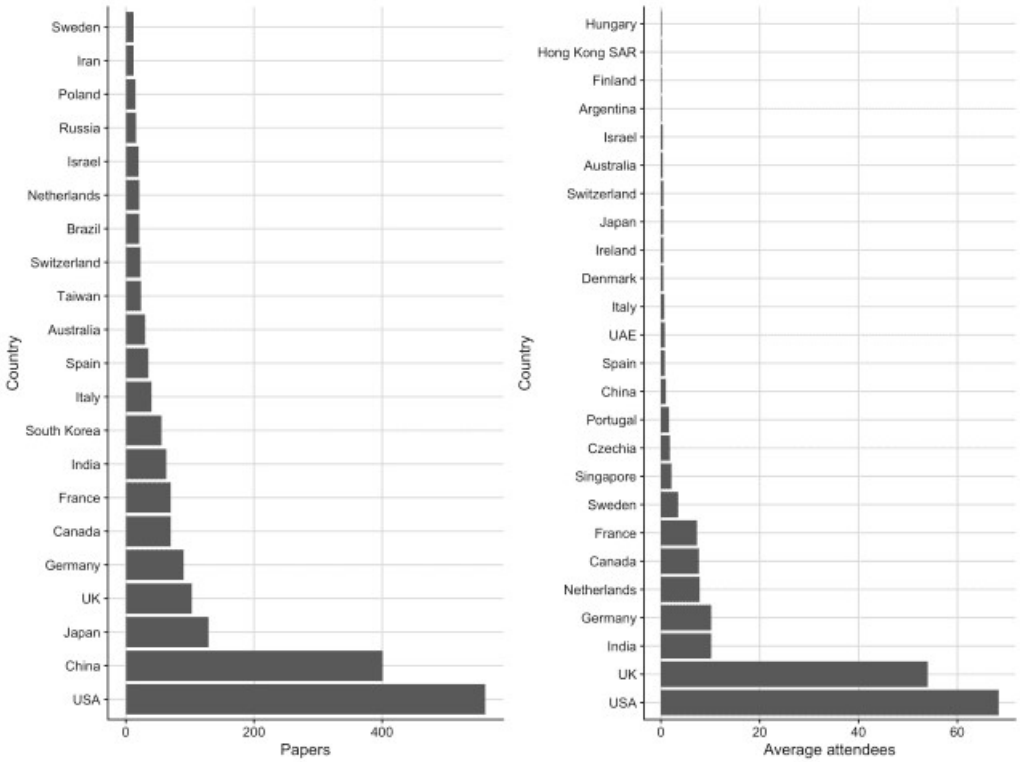
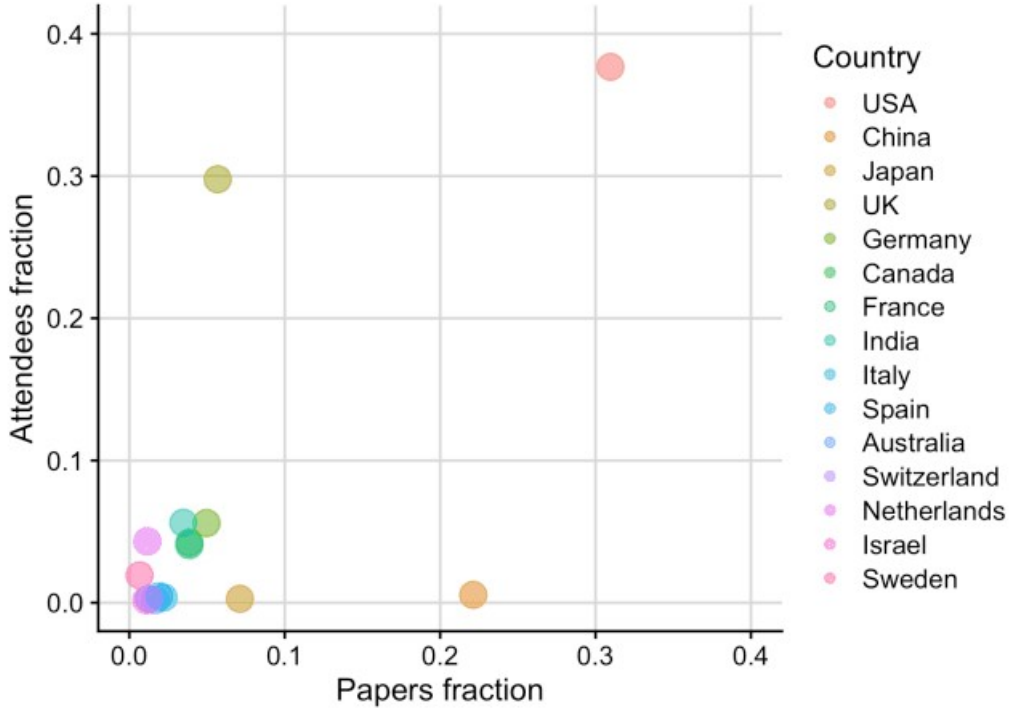
What did we find?

- 

- 



The USA produces the most papers on motors and they represent the biggest chunk of the Motors in Quarantine attendees. UK is ve much over-represented at the seminars, whereas as China and Japan are completely under-represented.

The seminars are at 4pm UK time, which may be quite late for Asia. Recordings are available for viewing at a more convenient time b it seems that China and Japan are not opting for this either.

There are some countries that have a consistent presence at the seminars despite not producing a large share of the papers. It's possible that a country has just one or two labs working on motors, so number of papers is quite low but lots of lab members are showing up at the seminars. Another possibility is that lab members working in the USA are residing in another country during the pandemic.

> The analysis is useful to take look at where development can be done to increase representation from all labs interested in motors.

Code walkthrough – rationale

The challenge is that we want to know where research activity on motors is happening. There are several ways to do this, but a simpl approach is to use the production of papers as a readout of research activity.

I retrieved a PubMed XML file for this query

```
(kinesin[tiab] OR myosin[tiab] OR dynein[tiab]) AND 2019[pdat]
```

This will give us all papers in PubMed published in 2019 that had in the title or abstract: either kinesin, myosin or dynein. This was 20 papers.

Extracting the data

Getting the data from this file into R is relatively straightforward, but how do we know which country produced a given paper?

PubMed has affiliation info for all authors for most papers. I came up with a way to retrieve country information for the last author and used this to determine the country for the paper. My first thought was to extract countries from all affiliations. This seemed like a good idea, because attendees at the seminar series are students and postdocs as well as PIs. However, this would mean that number of authors on a paper could skew the geographic distribution. Also, some authors have multiple affiliations. So, taking the final affiliation, i.e. just one per paper, was a good solution.

```
require(XML)
require(ggplot2)
require(cowplot)

filename <- "Data/pubmed_result.xml"
```

To parse the XML file I use an edited form of pubmedXML code.

```
extract_xml <- function(theFile) {
  library(XML)
  newData <- xmlParse(theFile)
  records <- getNodeSet(newData, "//PubmedArticle")
  pmid <- xpathSApply(newData,"//MedlineCitation/PMID", xmlValue)
  doi <- lapply(records, xpathSApply, ".//ELocationID[@EIdType = \"doi\"]", xmlValue)
  doi[sapply(doi, is.list)] <- NA
  doi <- unlist(doi)
  authLast <- lapply(records, xpathSApply, ".//Author/LastName", xmlValue)
  authLast[sapply(authLast, is.list)] <- NA
  authInit <- lapply(records, xpathSApply, ".//Author/Initials", xmlValue)
  authInit[sapply(authInit, is.list)] <- NA
  authors <- mapply(paste, authLast, authInit, collapse = "|")
  affiliations <- lapply(records, xpathSApply, ".//Author/AffiliationInfo/Affiliation",
xmlValue)
  affiliations[sapply(affiliations, is.list)] <- NA
  affiliations <- sapply(affiliations, paste, collapse = "|")
  year <- lapply(records, xpathSApply, ".//PubDate/Year", xmlValue)
  year[sapply(year, is.list)] <- NA
  year <- unlist(year)
  articletitle <- lapply(records, xpathSApply, ".//ArticleTitle", xmlValue)
  articletitle[sapply(articletitle, is.list)] <- NA
  articletitle <- unlist(articletitle)
  journal <- lapply(records, xpathSApply, ".//ISOAbbreviation", xmlValue)
  journal[sapply(journal, is.list)] <- NA
  journal <- unlist(journal)
  volume <- lapply(records, xpathSApply, ".//JournalIssue/Volume", xmlValue)
  volume[sapply(volume, is.list)] <- NA
  volume <- unlist(volume)
  issue <- lapply(records, xpathSApply, ".//JournalIssue/Issue", xmlValue)
  issue[sapply(issue, is.list)] <- NA
  issue <- unlist(issue)
  pages <- lapply(records, xpathSApply, ".//MedlinePgn", xmlValue)
  pages[sapply(pages, is.list)] <- NA
  pages <- unlist(pages)
  abstract <- lapply(records, xpathSApply, ".//Abstract/AbstractText", xmlValue)
  abstract[sapply(abstract, is.list)] <- NA
  abstract <- sapply(abstract, paste, collapse = "|")
  recdatey <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'received']/Year",
xmlValue)
  recdatey[sapply(recdatey, is.list)] <- NA
  recdatem <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'received']/Month",
xmlValue)
  recdatem[sapply(recdatem, is.list)] <- NA
  recdated <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'received']/Day",
xmlValue)
  recdated[sapply(recdated, is.list)] <- NA
  recdate <- mapply(paste, recdatey, recdatem, recdated, collapse = "|")
  accdatey <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'accepted']/Year",
xmlValue)
  accdatey[sapply(accdatey, is.list)] <- NA
  accdatem <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'accepted']/Month",
xmlValue)
  accdatem[sapply(accdatem, is.list)] <- NA
  accdated <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'accepted']/Day",
xmlValue)
  accdated[sapply(accdated, is.list)] <- NA
  accdate <- mapply(paste, accdatey, accdatem, accdated, collapse = "|")
  # use pubmed date as the published date. This seems safe for older records.
  pubdatey <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'pubmed']/Year",
xmlValue)
  pubdatey[sapply(pubdatey, is.list)] <- NA
  pubdatem <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'pubmed']/Month",
xmlValue)
  pubdatem[sapply(pubdatem, is.list)] <- NA
```

```
  pubdated <- lapply(records, xpathSApply, ".//PubMedPubDate[@PubStatus = 'pubmed']/Day",
xmlValue)
  pubdated[sapply(pubdated, is.list)] <- NA
  pubdate <- mapply(paste, pubdatey, pubdatem, pubdated, collapse = "|")
  ptype <- lapply(records, xpathSApply, ".//PublicationType", xmlValue)
  ptype[sapply(ptype, is.list)] <- NA
  ptype <- sapply(ptype, paste, collapse = "|")
  theDF <- data.frame(pmid, doi, authors, affiliations, year, articletitle, journal, volume,
issue, pages, abstract, recdate, accdate, pubdate, ptype, stringsAsFactors = FALSE)
  ## convert the dates
  theDF$recdate <- as.Date(theDF$recdate, format="%Y %m %d")
  theDF$accdate <- as.Date(theDF$accdate, format="%Y %m %d")
  theDF$pubdate <- as.Date(theDF$pubdate, format="%Y %m %d")
  return(theDF)
}

## make the dataframe
theData <- extract_xml(filename)
## have a look at a few titles
theData[sample(nrow(theData), 5), "articletitle"]
```

It's a good idea to have a look at a few random titles from the dataset to make sure they look OK (the search criterion may need adjusting).

```
[1] "Kinesin-6 Klp9 plays motor-dependent and -independent roles in collaboration with
Kinesin-5 Cut7 and the microtubule crosslinker Ase1 in fission yeast."
[2] "MYH9 overexpression correlates with clinicopathological parameters and poor prognosis of
epithelial ovarian cancer."
[3] "PTP1B up-regulates EGFR expression by dephosphorylating MYH9 at Y1408 to promote cell
migration and invasion in esophageal squamous cell carcinoma."
[4] "MPT0G413, A Novel HDAC6-Selective Inhibitor, and Bortezomib Synergistically Exert Anti-
tumor Activity in Multiple Myeloma Cells."
[5] "Crosstalks of the PTPIP51 interactome revealed in Her2 amplified breast cancer cells by
the novel small molecule LDC3/Dynarrestin."
```

These look good. So let's carry on!

```
## now we extract the country from the last authors affiliation
theData$country <- gsub(".*, (.*)","\\1",theData$affiliations)
## remove last period
theData$country <- sub(".$", "", theData$country)
## these countries need to be cleaned up
## load country lookup
country_lookup <- read.table("Data/country_lookup.txt", sep = "\t", header = TRUE,
stringsAsFactors = FALSE)
for (row in 1:nrow(country_lookup)) {
  regex <- country_lookup$Regex[row]
  theCountry <- country_lookup$Replace[row]
  theData$country <- gsub(regex, theCountry, theData$country)
}
## save out the data
write.table(theData, file = "Output/Data/pubmed_data.txt", sep = "\t", row.names = F)
```

What's happening here? In our affiliations column of the dataframe, we have a long list of all affiliations. The last author's affiliation is the final one. Fortunately they are in a fixed format where the field ends with ", Austria.". This means we can get the country by finding the last comma-space combination and deleting the final period.

The countries still needed to be cleaned up. Briefly, three problems with what we had extracted:

1. Synonyms of countries (United Kingdom, UK, U.K. etc)
2. States of USA listed rather than USA itself (in different formats)
3. Some fields had an email address appended.

To get around these problems, I made a quick regex lookup table to clean up the countries column. If you are reusing this code, it ma need to be extended as not all countries are listed (only the ones in this dataset).

```
## prepare to plot the data
```

```
countryDF <- as.data.frame(table(theData$country))
names(countryDF) <- c("Country","Count")
countryDF <- subset(countryDF, Country != "N")
countryDF <- subset(countryDF, Count >= 10)
countryDF$Country <- factor(countryDF$Country,
                  levels = countryDF$Country[order(countryDF$Count, decreasing = TRUE)])
countryDF$Fraction <- countryDF$Count / sum(countryDF$Count,na.rm = T)
```

I took the countries that had produced 10 or more papers in 2019, and organised the levels in rank order of most to least papers, reac for plotting.

```
## load MiQ data
miqDF<- read.table("Data/miq.txt", sep = "\t", header = TRUE, stringsAsFactors = FALSE)
miqDF$Country <- factor(miqDF$Country,
                        levels = miqDF$Country[order(miqDF$Attendees, decreasing = TRUE)])
miqDF$Fraction <- miqDF$Attendees / sum(miqDF$Attendees,na.rm = T)

## merge the two dataframes
compareDF <- merge(countryDF, miqDF, by="Country")
```

I loaded the attendee data. Anne had collated average attendance numbers from over six weeks of seminars. I made sure that we ha a fractional representation of papers and attendees. Finally I merged the two.

The code for plotting is shown below.

```
## make the plots
p1 <- ggplot(data = countryDF, aes(x = Country, y = Count)) +
  geom_bar(stat = "identity") +
  labs(y = "Papers") +
  theme_half_open(12) +
  background_grid() +
  coord_flip()
ggsave("Output/Plots/papers_per_country.png", p1, dpi = 300, width = 170, height = 100, units =
"mm")

p2 <- ggplot(data = miqDF, aes(x = Country, y = Attendees)) +
  geom_bar(stat = "identity") +
  labs(y = "Average attendees") +
  theme_half_open(12) +
  background_grid() +
  coord_flip()
ggsave("Output/Plots/attendees_per_country.png", p2, dpi = 300, width = 170, height = 100,
units = "mm")

p3 <- plot_grid(p1, p2)
ggsave("Output/Plots/combined.png", p3, dpi = 300)

p4 <- ggplot(data = countryDF, aes(x = Country, y = Fraction)) +
  geom_bar(stat = "identity") +
  labs(y = "Papers fraction") +
  theme_half_open(12) +
  background_grid() +
  coord_flip()
ggsave("Output/Plots/paperfraction_per_country.png", p4, dpi = 300, width = 170, height = 100,
units = "mm")

p5 <- ggplot(data = miqDF, aes(x = Country, y = Fraction)) +
  geom_bar(stat = "identity") +
  labs(y = "Attendee fraction") +
  theme_half_open(12) +
  background_grid() +
  coord_flip()
ggsave("Output/Plots/attendeefraction_per_country.png", p5, dpi = 300, width = 170, height =
100, units = "mm")

p6 <- plot_grid(p4, p5)
```

```
ggsave("Output/Plots/combined_fraction.png", p6, dpi = 300)

p7 <- ggplot(compareDF,
             aes(x = Fraction.x, y = Fraction.y, colour = Country, alpha = 0.5)) +
  geom_point(aes(size = 1.5)) +
  guides(size = FALSE, alpha = FALSE, colour = guide_legend(override.aes = list(alpha = 0.5)))
+
  xlim(c(0,0.4)) + ylim(c(0,0.4)) +
  labs(x = "Papers fraction", y = "Attendees fraction") +
  theme_half_open(12) +
  background_grid()
ggsave("Output/Plots/compare_fraction.png", p7, dpi = 300, width = 140, height = 100, units =
"mm")
```

As always, I'm happy to hear about improvements. The weakest part is probably the country clean-up step. If you have ideas about h to do that better, let me know!