

Missionaries: Catalysts for Aid? Evidence from Africa

Introduction

Aid allocation is ideally based on need, yet often it is influenced by political and strategic considerations. This research investigates how colonial ties, particularly those related to Christian missionaries who preceded modern local-level aid projects and assimilate them in terms of function, impact aid distribution today (Alpino and Hammersmark, 2021).

Data

The data for all aid projects across all 54 countries in Africa was provided by the International Aid Transparency Initiative (IATI) and retrieved from d-portal (link already queries for all 54 countries). In total, 269,058 activities were recorded across 1148 IATI organisations of which 26% had exact geo-locations. Subject to this analysis is the subsample of aid projects that have exact geo-locations, totalling 65521 projects across 278 organisations.

Missionary data comes from Hedde - von Westernhagen and Becker (2022) and is retrieved from Harvard Dataverse. To date, this is the most extensive colonial missionary dataset on Africa, surpassing Nunn (2010) whose data is often used for similar studies in terms of coverage.

Country boundaries and sub-national boundaries are retrieved from GADM (see “whole world” and “whole world as six separate layers”) and ArcGis. Boundaries on climate zones come from Regional Centre for Mapping of Resource for Development.

```
## Warning: package 'sf' was built under R version 4.2.3

## Warning: package 'openxlsx' was built under R version 4.2.3

## Warning: package 'zoo' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'spatstat' was built under R version 4.2.3

## Warning: package 'spatstat.data' was built under R version 4.2.3

## Warning: package 'spatstat.geom' was built under R version 4.2.3

## Warning: package 'spatstat.random' was built under R version 4.2.3

## Warning: package 'spatstat.explore' was built under R version 4.2.3

## Warning: package 'spatstat.model' was built under R version 4.2.3
```

```

## Warning: package 'spatstat.linnet' was built under R version 4.2.3

## Warning: package 'terra' was built under R version 4.2.3

## Warning: package 'tmap' was built under R version 4.2.3

## Warning: package 'spatialreg' was built under R version 4.2.3

## Warning: package 'spData' was built under R version 4.2.3

## Warning: package 'Matrix' was built under R version 4.2.3

## Warning: package 'spdep' was built under R version 4.2.3

```

Loading Aid Data

The downloaded aid data from the link above is loaded and converted into a spatial data frame.

```

aiddata <- read.csv("data/dportal_map_activities.csv")
aiddata$day_start <- ymd(айдата$day_start) # using lubridate's ymd function to parse dates
айдата_sf <- st_as_sf(айдата, coords = c("location_longitude", "location_latitude"), crs = 4326)

```

Loading Africa boundaries

The sub-national, district boundaries at level 1 are loaded from GADM and filtered for African countries only. The code checks if boundaries for all countries were found. GADM provided level 1-4. Here, level 1 is chosen because it has Africa-wide coverage, while more granular levels are not available for all African countries.

Furthermore, a shape file for the whole African continent is loaded, and the areas that are desert are subtracted. The remaining area can be considered the habitable land in Africa and is used later on to filter out raster cells that are located in inhabitable land when running spatial regressions.

```

#Level 1 boundaries
boundaries <- st_read("data/gadm404-levels.gpkg", layer = "level1")
африка_boundaries <- boundaries[boundaries$COUNTRY %in% африка_countries,]

found <- unique(африка_boundaries$COUNTRY)
missing_countries <- setdiff(африка_countries, found)
print(missing_countries)

# Country boundaries
boundaries_ctr <- st_read("data/Africa_Boundaries.shp")
#crs_info <- st_crs(boundaries_ctr)
#print(crs_info)

#Continent boundaries
continents <- st_read("data/World_continents.shp")
аfrican_continent <- continents[continents$CONTINENT == "Africa",]
qtm(аfrican_continent)

```



```
#Desert boundaries
biomes <- st_read("data/Africa_Biomes_Dataset.shp")
biomes <- st_transform(biomes, crs = 4326)
qtm(biomes)
```



```
desert <- biomes[biomes$FID == 1,]  
qtm(desert)
```



```
#subtract desert from continent
habitableAfrica <- st_difference(african_continent, desert)

#save in shape file
st_write(habitableAfrica, "data/habitableAfrica.shp")

#clean up
rm(boundaries, found, crs_info, missing_countries, continents, biomes, desert)
gc()
```

Loading Missionary Data

Next the the data on the location of missions is loaded. In total 2278 missions are recorded.

```
#load missions
missions <- st_read("data/bj_missions.shp")

## Reading layer 'bj_missions' from data source
##   'C:\Users\kaius\OneDrive - Hertie School\MSc Data Science for Public Policy\Semester_4\Geospatial Data\'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 2278 features and 4 fields
## Geometry type: POINT
## Dimension:      XY
```

```

## Bounding box: xmin: -17.20426 ymin: -34.52016 xmax: 50.1411 ymax: 36.89282
## Geodetic CRS: WGS 84

str(missions)

## Classes 'sf' and 'data.frame': 2278 obs. of 5 variables:
## $ xcoord : num -5.29 -5.79 -6.18 -6.96 -7.91 ...
## $ ycoord : num 35.5 35.6 35 33.9 33.4 ...
## $ denom   : chr "Catholic" "Catholic" "Catholic" "Catholic" ...
## $ vacated : chr "no" "no" "no" "no" ...
## $ geometry:sfc_POINT of length 2278; first list element: 'XY' num -5.29 35.49
## - attr(*, "sf_column")= chr "geometry"
## - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA NA
## ..- attr(*, "names")= chr [1:4] "xcoord" "ycoord" "denom" "vacated"

```

Summary statistics

Mapping missionary data

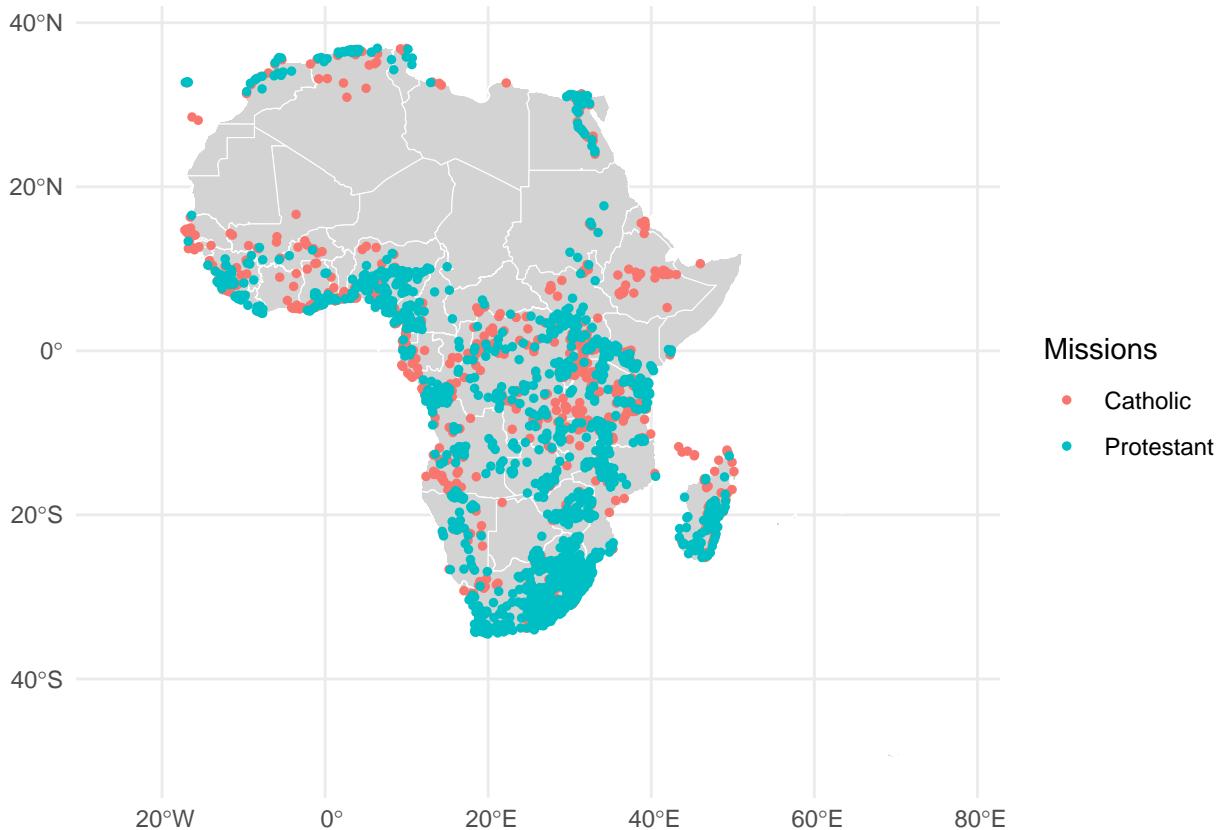
This map displays mission locations by denomination on a map of national boundaries.

```

#plot missions
missions_plot <- ggplot() +
  geom_sf(data = boundaries_ctr, fill = "lightgray", color = "white") +
  geom_sf(data = missions, aes(color = denom), size = 1) +
  labs(color = "Missions" ) +
  theme_minimal()

print(missions_plot)

```



```
ggsave("output/missions.png", missions_plot, width = 10, height = 7)
rm(missions_plot)
```

Mapping missionary data against aid projects

The line of code below performs a spatial join between the aid data and the African district boundaries by merging point location of the aid projects with polygons indicating the district boundaries, specifically where the points intersect the polygons. This operation creates a new dataset that adds district names and further district information to the aid projects.

```
#spatial merge of missions on level-1 boundaries
master <- st_join(aiddata_sf, africa_boundaries, join = st_intersects)
```

This plot shows the location point of missions and number of aid projects by level 1 boundary. To do so, a new dataset was created with the number of aid projects by level 1 district. This dataset was merged with the level 1 boundary dataset to create a spatial dataframe including the polygons of the districts and the number of projects in each district. This data is plotted along side the point locations of missions, where mission points are coloured by their denomination.

A caveat of this map is that larger districts in terms of area tend to have more projects and therefore show up in brighter colour. In order to address this issue, the concentration of aid project is revisited in the analysis section by plotting the number of aid projects and missions on grid cells of 0.5x0.5 degrees (approximately 55kmx55km) following Alpino and Hammersmark (2021).

```

# Create a dataset with number of projects by level 1 district
regional_projects <- master %>%
  group_by(NAME_1) %>%
  summarise(Num_Projects = n()) %>%
  st_set_geometry(NULL)

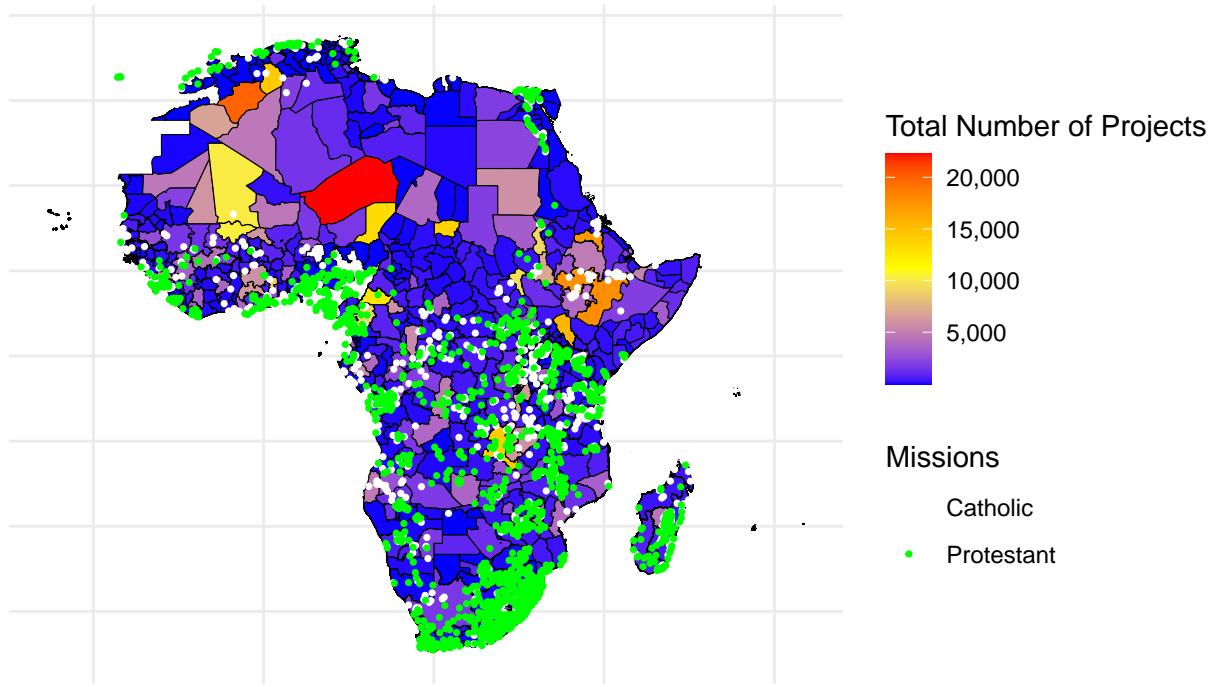
# Merge number of aid project by district with district polygons
map_data <- merge(africa_boundaries, regional_projects, by = "NAME_1")

# Ensure the merged data is still an sf object
map_data <- st_as_sf(map_data)

# Plotting the map with ggplot2
missionandaid <- ggplot(data = map_data) +
  geom_sf(aes(fill = Num_Projects), color = "black", size = 0.5) +
  geom_sf(data = missions, aes(color = denom), size = 0.6) +
  scale_fill_gradientn(colors = c("blue", "yellow", "red"),
                        values = scales::rescale(c(0, 10000, 20000)),
                        labels = comma,
                        guide = "colourbar") +
  scale_color_manual(values = c("Catholic" = "white", "Protestant" = "green"),
                     labels = c("Catholic", "Protestant")) +
  labs(fill = "Total Number of Projects",
       color = "Missions" ) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.background = element_blank())

print(missionandaid)

```



```
ggsave("output/missionsandaid.png", missionandaid, width = 10, height = 7)
rm(missionandaid)
```

Summary statistics on the aid data

Graph 1: Number of project by institution where institution names are cut to the first four words and institutions with less than 500 projects are filtered out. One can observe that international organisations and bilateral donors have the highest number of reported geo-tagged projects in Africa. The organisation AidData is not a donor, but a data provider for geo-tagged aid projects.

This reporting error brings to light a more fundamental issue with the data. The dataset classifies the agency that reports the data to IATI as donor, regardless of the implementing organisation. This means, that projects could be double counted if the reporting agency, e.g. Agence Française de Développement, reports projects that finance multi-lateral organisations, e.g. UNICEF, who in turn report the project, given they implement it.

For the Federal Ministry of Economics Cooperation and Development (the organisation with the second most projects in this dataset) (BMZ), the universe of aid projects financed by the ministry was sourced from BMZ. For the BMZ, I found that close to all projects are reported in my data, and that the data includes only bilateral aid, i.e. it excludes funding to other international organisations. For other donors similar research could be conducted to refine the understanding about the quality of data provided here.

```
# Summarize data to count number of projects per reporting institution
project_counts <- aiddata %>%
  mutate(reporting = str_split(reporting, " "), .by = reporting) %>%
  group_by(reporting) %>%
```

```

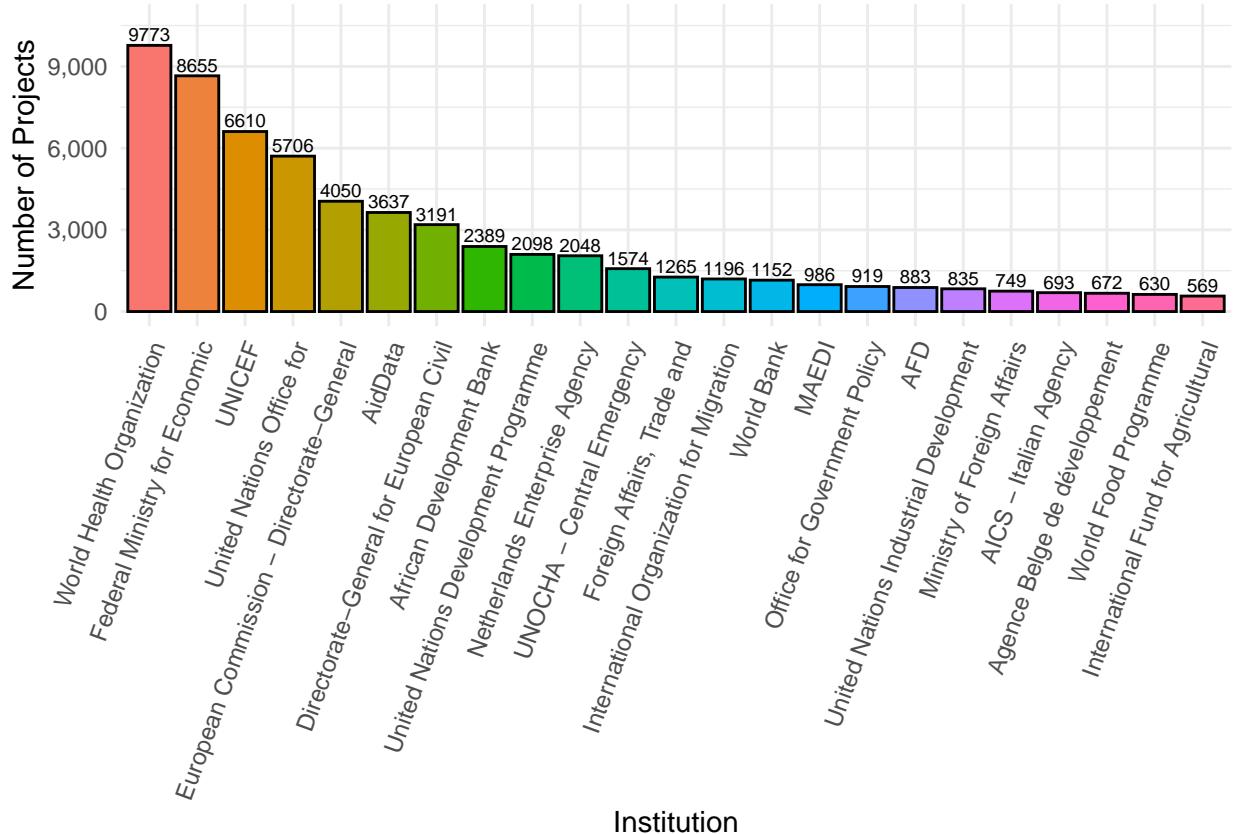
summarise(Number_of_Projects = n_distinct(aid), .groups = 'drop') %>%
filter(Number_of_Projects >= 500) #

#Sort institutions by number of projects
prepared_data <- project_counts %>%
  mutate(reporting = factor(reporting, levels = project_counts$reporting[order(-project_counts$Number_of_Projects)]))

# Create the plot
projectByInstitution <- ggplot(prepared_data, aes(x = reporting, y = Number_of_Projects, fill = reporting)) +
  geom_bar(stat = "identity", color = "black") + # Create the bars
  geom_text(aes(label = Number_of_Projects), vjust = -0.3, color = "black", size = 2.3) +
  theme_minimal() +
  labs(x = "Institution",
       y = "Number of Projects") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1)) +
  guides(fill = FALSE) +
  scale_y_continuous(labels = label_comma(), limits = c(NA, max(prepared_data$Number_of_Projects) * 1.1))

# Display the plot
print(projectByInstitution)

```



Graph2: This figure shows the number of aid projects across countries. Given we are not working with a sample of aid data, it is important to understand where aid projects in the data are concentrated. It seems that Ethiopia, Somalia, Congo and South Sudan have many projects which surprised the author.

```

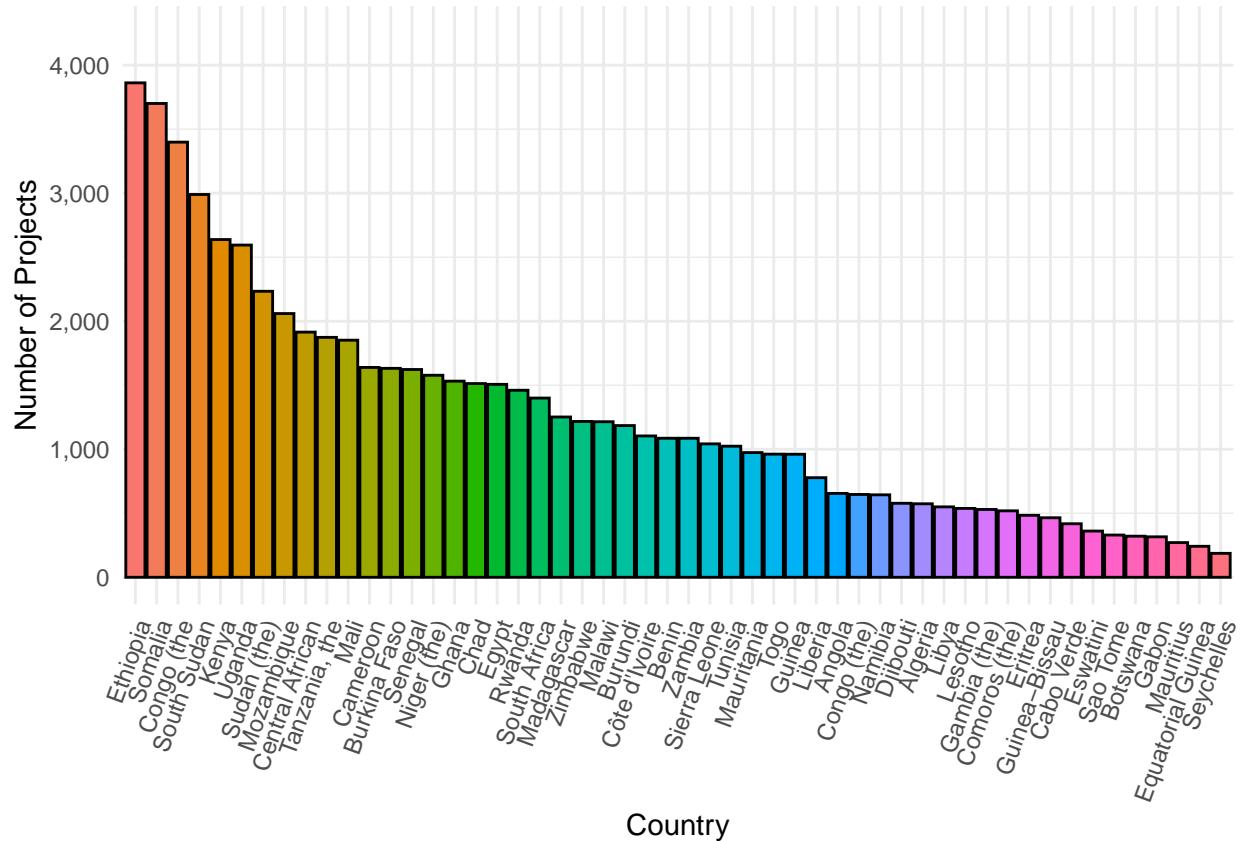
# Summarize data to count number of projects per country
project_counts <- aiddata %>%
  mutate(country_code = str_split(country_code, " "), .by_group = TRUE) %>%
  summarise(Number_of_Projects = n_distinct(aid), .groups = 'drop')

#Sort countries by number of projects
prepared_data <- project_counts %>%
  mutate(country = factor(country_code, levels = project_counts$country_code[order(-project_counts$Number_of_Projects)]))

# Create the plot
projectByInstitution <- ggplot(prepared_data, aes(x = country, y = Number_of_Projects, fill = country))
  geom_bar(stat = "identity", color = "black") +
  theme_minimal() +
  labs(x = "Country",
       y = "Number of Projects") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1)) +
  guides(fill = FALSE) +
  scale_y_continuous(labels = label_comma(), limits = c(NA, max(prepared_data$Number_of_Projects) * 1.1))

# Display the plot
print(projectByInstitution)

```



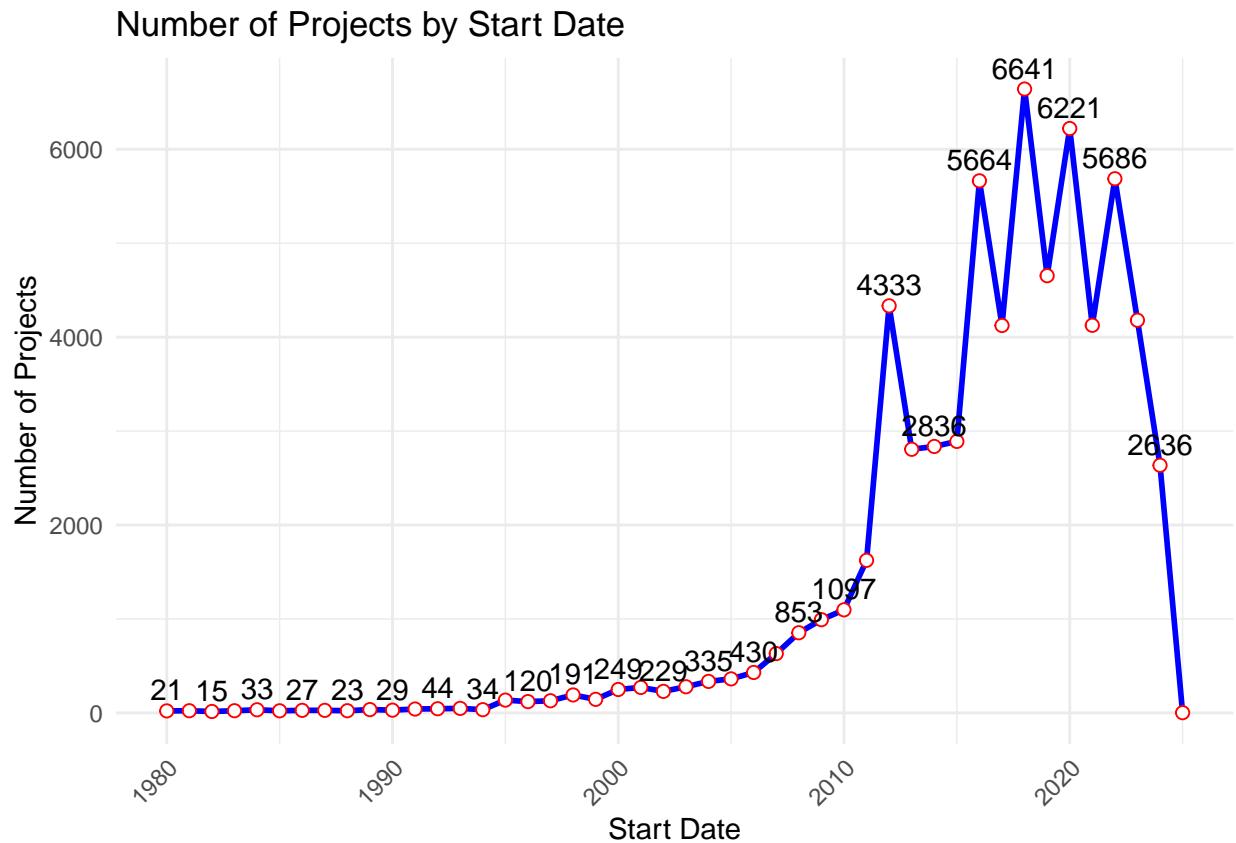
Graph3: This graph plots the number of project by start date in order to better understand the temporal dimension of the data. The large increase in projects after 2010 suggests that many more projects with

geotags were reported to IATI, albeit it stays unclear whether this is due to an increase in aid projects or an improvement in project reporting.

```
# Summarize data to count the number of projects starting each year
daily_projects <- aiddata %>%
  mutate(year = year(day_start))%>%
  group_by(year) %>%
  summarise(Number_of_Projects = n_distinct(aid), .groups = 'drop') %>%
  arrange(year) %>%
  filter(year >= 1980)

# Create the line chart
line_chart <- ggplot(daily_projects, aes(x = year, y = Number_of_Projects)) +
  geom_line(group = 1, color = "blue", size = 1) +
  geom_point(color = "red", size = 2, shape = 21, fill = "white") +
  geom_text(data = filter(daily_projects, year %% 2 == 0), aes(label = Number_of_Projects),
            vjust = -0.5) +
  labs(title = "Number of Projects by Start Date",
       x = "Start Date",
       y = "Number of Projects") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display the plot
print(line_chart)
```



Graph4: Next, the quality of the geotags are evaluated. As explained by Alpino and Hammersmark (2021),

the variable “location_precision” ranges from level 1-8, where 1 signifies a exact location specification, and 2 signifies a location accuracy up to 25km of the project location. All other precision levels are more coarse or not suited for geospatial analysis, and therefore projects with other precision levels are not reported with geo-tags in d-portal.

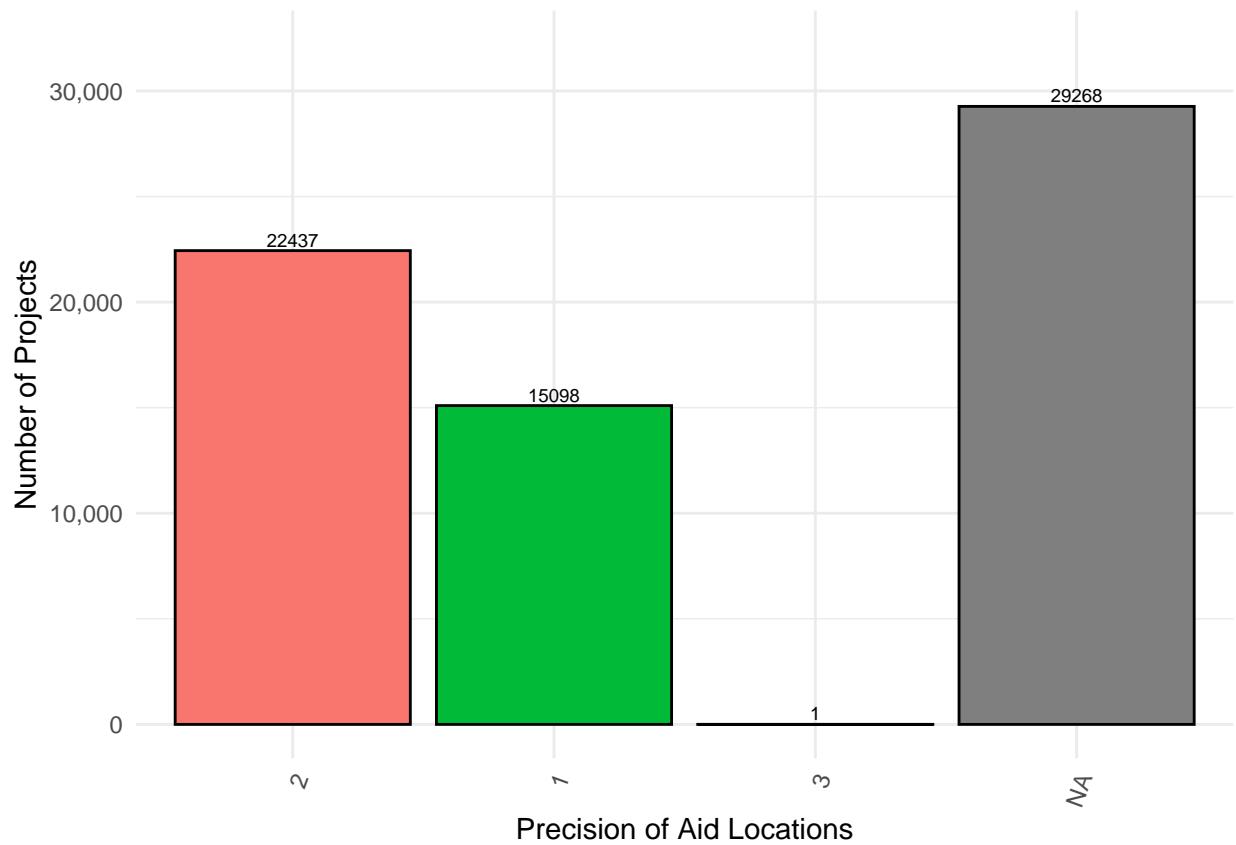
Given this context, we can observe from Graph4 that about half of the projects have a precision level of 1 or 2 and the rest has no reported precision level.

```
# Summarize data to count number of projects by location precision
project_counts <- aiddata %>%
  group_by(location_precision) %>%
  summarise(Number_of_Projects = n_distinct(aid), .groups = 'drop')

#Sort by number od projects
prepared_data <- project_counts %>%
  mutate(location_precision = factor(location_precision, levels = project_counts$location_precision[order(project_counts$Number_of_Projects)]))

# Create the plot
precision <- ggplot(prepared_data, aes(x = location_precision, y = Number_of_Projects, fill = location_precision))
precision +
  geom_bar(stat = "identity", color = "black") +
  geom_text(aes(label = Number_of_Projects), vjust = -0.3, color = "black", size = 2.3) +
  theme_minimal() +
  labs(x = "Precision of Aid Locations",
       y = "Number of Projects") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1)) +
  guides(fill = FALSE) +
  scale_y_continuous(labels = label_comma(), limits = c(NA, max(prepared_data$Number_of_Projects) * 1.1))

# Display the plot
print(precision)
```



```
rm(precision)
```

Graph5 and 6 reports the number of projects by institution that have no reported precision levels against those that do (Graph6). Comparing Graph5 to Graph6, we can observe that donors either report their project locations using the location_precision variable, or they do not. For example, WHO which was ranked largest donor in Graph1 has no project with location_precision specified. In contrast, all BMZ projects consistently come with a location precision.

These findings could provoke future researcher to focus on donors that report precision levels and/or have high reporting coverage of their projects like the BMZ. Since reducing the scope of aid projects to a single donor comes with the drawback that it likely represents the population of aid project even less, this project sticks with using the entire data.

```
# Filter data to include only rows where location_precision is NA
filtered_data <- aiddata %>%
  filter(is.na(location_precision))

# Summarize filtered data to count number of projects by institution
project_counts <- filtered_data %>%
  mutate(reporting = sapply(str_split(reporting, " "), function(x) paste(x[1:min(length(x), 4)]), collapse = "")) %>%
  group_by(reporting) %>%
  summarise(Number_of_Projects = n_distinct(aid), .groups = 'drop') %>%
  filter(Number_of_Projects >= 500) # Remove institutions with less than 500 projects

# Sort institution by number of projects
prepared_data <- project_counts %>%
```

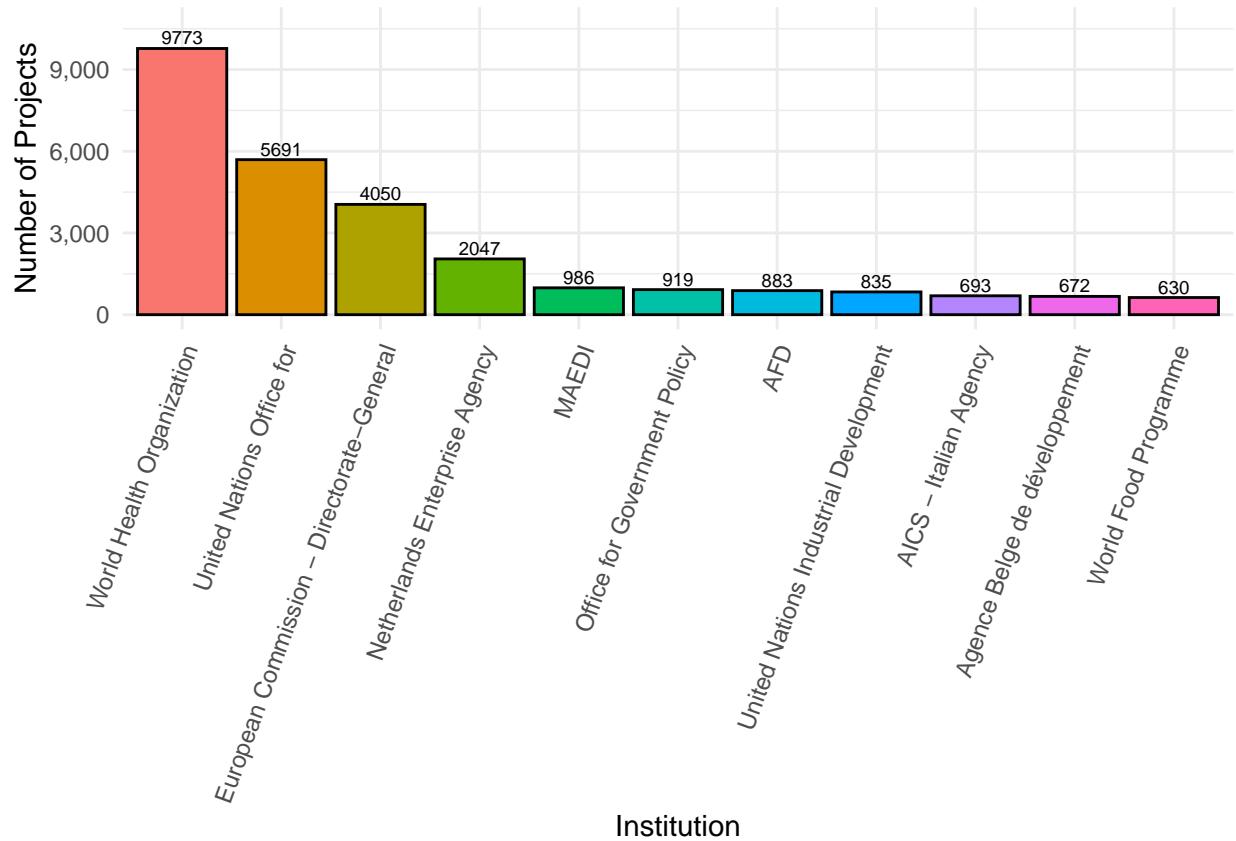
```

mutate(reporting = factor(reporting, levels = project_counts$reporting[order(-project_counts$Number_of_Projects)]))

# Create the plot
institution_plot <- ggplot(prepared_data, aes(x = reporting, y = Number_of_Projects, fill = reporting))
geom_bar(stat = "identity", color = "black") + # Create the bars
geom_text(aes(label = Number_of_Projects), vjust = -0.3, color = "black", size = 2.3) +
theme_minimal() +
labs(x = "Institution",
y = "Number of Projects") +
theme(axis.text.x = element_text(angle = 70, hjust = 1)) +
guides(fill = FALSE) +
scale_y_continuous(labels = label_comma(), limits = c(NA, max(prepared_data$Number_of_Projects) * 1.1))

# Display the plot
print(institution_plot)

```



```

rm(institution_plot)

# Filter data to include only rows where location_precision is not NA
filtered_data <- aiddata %>%
  filter(!is.na(location_precision))

# Summarize filtered data to count number of projects by institution
project_counts <- filtered_data %>%
  mutate(reporting = sapply(str_split(reporting, " "), function(x) paste(x[1:min(length(x), 4)], collapse = "")))

```

```

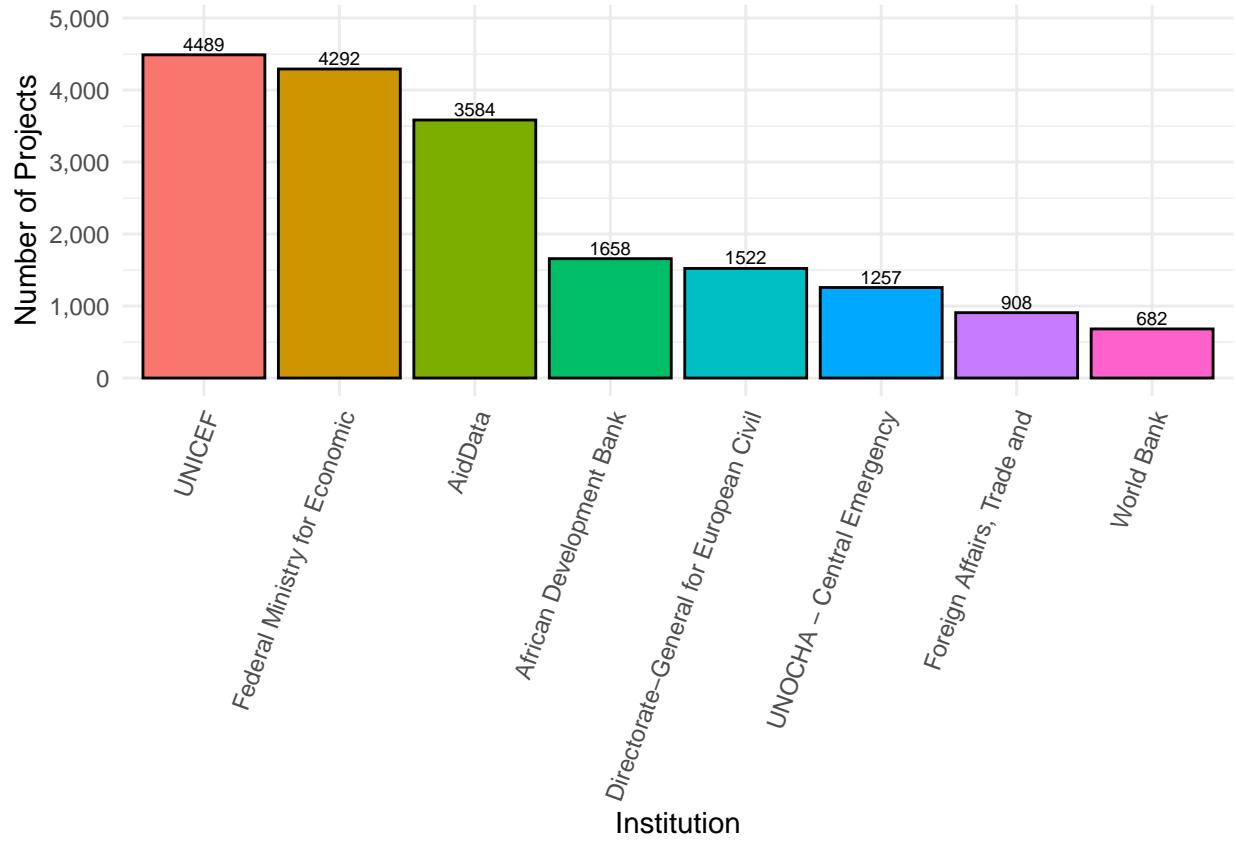
mutate(year = year(day_start))%>%
filter(year<=2016)%>%
group_by(reporting) %>%
summarise(Number_of_Projects = n_distinct(aid), .groups = 'drop')%>%
filter(Number_of_Projects >= 500) #

#Sort by number of projects
prepared_data <- project_counts %>%
  mutate(reporting = factor(reporting, levels = project_counts$reporting[order(-project_counts$Number_o))

# Create the plot
institution_plot <- ggplot(prepared_data, aes(x = reporting, y = Number_of_Projects, fill = reporting))
  geom_bar(stat = "identity", color = "black") + # Create the bars
  geom_text(aes(label = Number_of_Projects), vjust = -0.3, color = "black", size = 2.3) +
  theme_minimal() +
  labs(x = "Institution",
       y = "Number of Projects") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1)) +
  guides(fill = FALSE) +
  scale_y_continuous(labels = label_comma(), limits = c(NA, max(prepared_data$Number_of_Projects) * 1.1))

# Display the plot
print(institution_plot)

```



```
rm(institution_plot)
```

Method

Two methods are used to assess the spatial correlation between colonial mission stations and contemporary aid projects.

First, a Cross K-Function is used. This function evaluates if two sets of points, A and B, are clustered or dispersed. At radius r, it compares the clustering of A and B to a random distribution of B (Gelb (2023). “Network k Functions”).

Second, regression and spatial regression is used. To use regressions, the aid projects and missions have to be mapped to grid cells as explained in the analysis section.

Analysis

Analysis using Cross K-function

The plot below represents the baseline (red, dotted line) which represents the how K would look like if the data was distributed according to a Poision distribution, against the sample K. Because $K(\text{un})$, the cross-K function of missions to aid projects, is above $K(\text{pois})$, the random baseline, we can infer that aid projects tend to be clustered around missions by more than we would expect at random.

```
aid_points <- ppp(aiddata$location_longitude, aiddata$location_latitude, window=owin(range(aiddata$longitude), range(aiddata$latitude)))
missionary_points <- ppp(missions$xcoord, missions$ycoord, window=owin(range(missions$xcoord), range(missions$ycoord)))

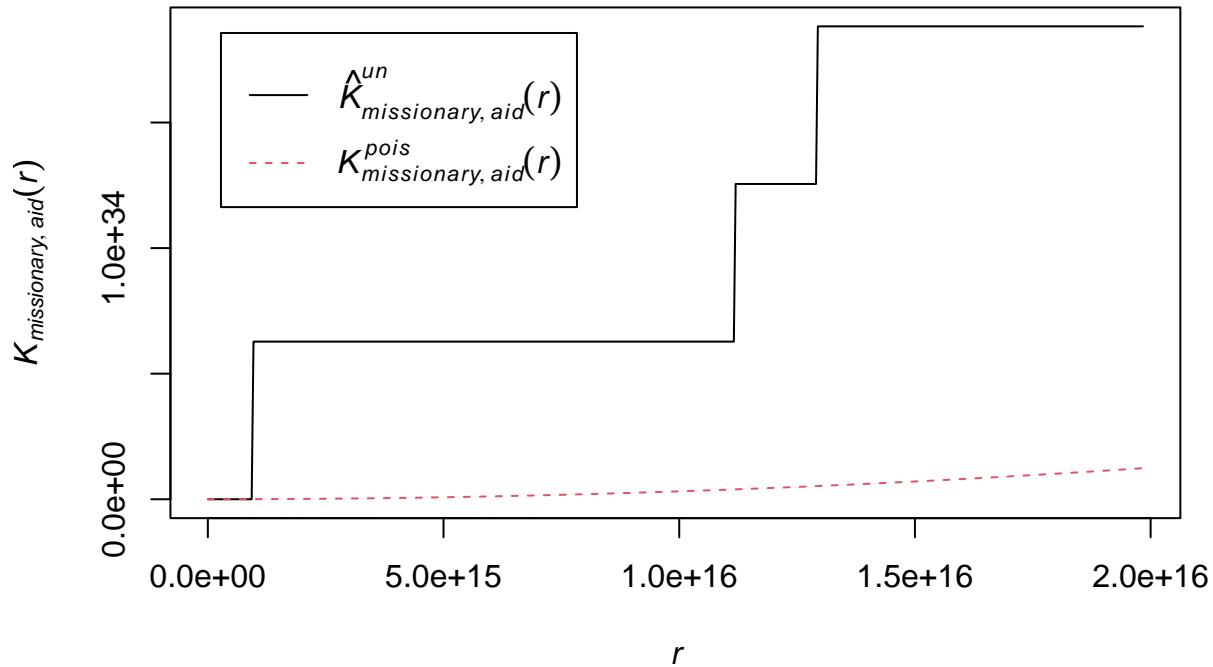
aid_points$marks <- "aid"
missionary_points$marks <- "missionary"
combined_points <- superimpose(aid=aid_points, missionary=missionary_points)

combined_points$marks <- as.character(unlist(combined_points$marks))
combined_points$marks <- factor(combined_points$marks)

# Cross K-function
cross_k <- Kcross(combined_points, i="missionary", j="aid", correction="none")

# Plotting the cross K-function
plot(cross_k, main="Cross K-function from Missionary to Aid")
```

Cross K-function from Missionary to Aid



Analysis using regression analysis

In order to run regression analysis, I aggregate the number of aid projects and missions on a grid cell level, where one grid cell has the size 0.5x0.5 degrees (approximately 55km x 55km). To achieve this, I create a raster with grid cells of the resolution 0.5x0.5 degrees bounded by the area of Africa. The coordinates were retrieved from the shape file with the African continent.

Next, the raster is mapped on the Africa shape file and the raster cells are cropped out that cover Africa. Aid data is then rasterised on this raster and the resulting plot shows the concentration of aid projects on a grid cell level over Africa.

Distributing aid projects into grid cells

```
# Creating a raster
# Define the extent using the bounding box coordinates
xmin <- -25.3618
xmax <- 77.60327
ymin <- -50.01889
ymax <- 37.55986

# Create a raster with a 0.5 degree resolution and map it on Africa
raster_05deg <- rast(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax, resolution=0.5)
xy <- xyFromCell(raster_05deg, 1:ncell(raster_05deg))
coop <- st_as_sf(as.data.frame(xy), coords = c("x", "y"),
```

```

        crs = st_crs(african_continent))
coop <- st_filter(coop, african_continent)
coop2 <- st_filter(coop, habitableAfrica)
coop_rast <- rasterize(coop, raster_05deg, field=1, fun='count')
coop_rast2 <- rasterize(coop2, raster_05deg, field=1, fun='count')
qtm(coop)

```



```

# Number of project by raster
aiddata_sf <- st_transform(aiddata_sf, crs(raster_05deg))
aiddata_vect <- vect(aiddata_sf)
aid_raster <- rasterize(aiddata_vect, raster_05deg, fun="count")
aid_raster[is.na(aid_raster)] <- 0
aid_raster <- mask(aid_raster, coop_rast, maskvalue=NA, updatevalue=NA)
aid_raster2 <- mask(aid_raster, coop_rast2, maskvalue=NA, updatevalue=NA)

# Calculate the percentile breaks
percentiles <- seq(0, 100, by = 10) # Adjust this as needed
breaks <- quantile(values(aid_raster), probs = percentiles / 100, na.rm = TRUE)

# Ensure breaks are unique and sorted (adding minimum for zero coverage)
breaks <- c(min(values(aid_raster)), na.rm = TRUE), unique(sort(breaks)))

# Ensure breaks are unique in case of duplicate percentile values
breaks <- unique(breaks)

```

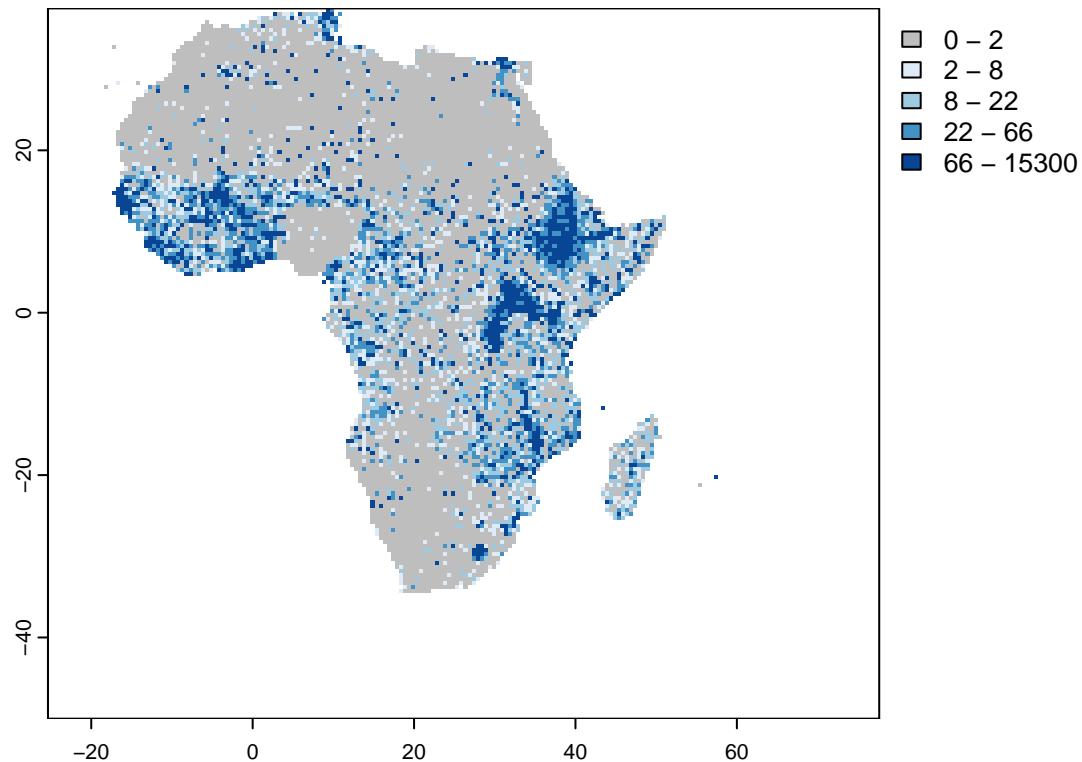
```

# Define a color palette including 'gray' for NA values
colors <- c("gray", brewer.pal(8, "Blues"))

# Plot the raster
plot(aid_raster, breaks=breaks, col=colors, main="Number of project locations per Cell by Percentiles")

```

Number of project locations per Cell by Percentiles



```

rm(invalid_geometry)
gc()

```

```

##           used   (Mb) gc trigger   (Mb) max used   (Mb)
## Ncells    9641622 515.0    20148211 1076.1  20148211 1076.1
## Vcells  118623810 905.1   186954797 1426.4 233675429 1782.9

```

Add missions to raster data

The same exercise is repeated for the missions data.

```

# rasterise mission data
# Number of missions by raster
missions <- st_transform(missions, crs(raster_05deg))
missions_vect <- vect(missions)
mission_raster <- rasterize(missions_vect, raster_05deg, fun="count")
mission_raster[is.na(mission_raster)] <- 0
mission_raster <- mask(mission_raster, coop_rast, maskvalue=NA, updatevalue=NA)

```

```

mission_raster2 <- mask(mission_raster, coop_rast2, maskvalue=NA, updatevalue=NA)

# Calculate the percentile breaks
percentiles <- seq(0, 100, by = 1) # Adjust this as needed
breaks <- quantile(values(mission_raster), probs = percentiles / 100, na.rm = TRUE)

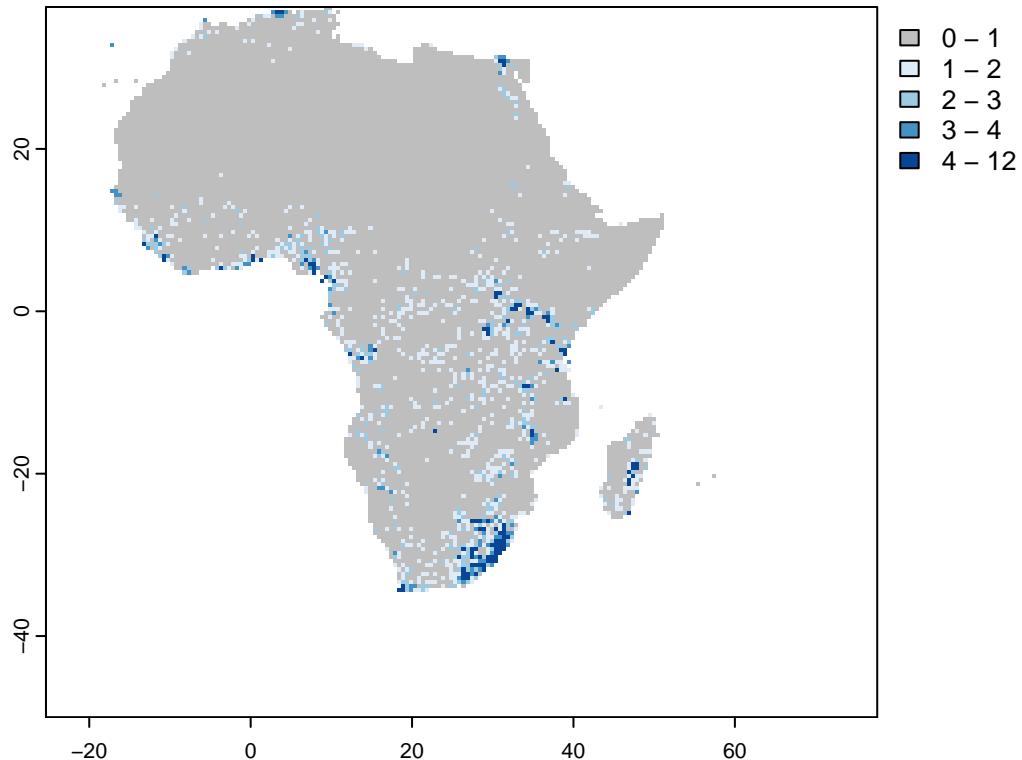
# Ensure breaks are unique and sorted (adding minimum for zero coverage)
breaks <- c(min(values(mission_raster)), na.rm = TRUE), unique(sort(breaks)))

# Ensure breaks are unique in case of duplicate percentile values
breaks <- unique(breaks)

# Define a color palette including 'gray' for NA values
colors <- c("gray", brewer.pal(8, "Blues"))
plot(mission_raster, breaks=breaks, col=colors, main="Number of Missions per Cell by Percentiles")

```

Number of Missions per Cell by Percentiles



```

# merge data
# Extract raster values to data frames
aid_values <- values(aid_raster)
aid_data <- data.frame(x = xy[,1], y = xy[,2], aid_projects = aid_values)
colnames(aid_data)[3] <- "NumAid"

mission_values <- values(mission_raster)
missions_data <- data.frame(x = xy[,1], y = xy[,2], missions = mission_values)
colnames(missions_data)[3] <- "NumMissions"

```

```

# Merge the data frames by coordinates
regression_data <- merge(aid_data, missions_data, by=c("x", "y"))

# Remove NA values which may complicate the analysis
regression_data <- na.omit(regression_data)

rm(aiddata_vect, missions_vect, aid_values, mission_values, missions_data, aid_data)
gc()

##           used   (Mb) gc trigger   (Mb) max used   (Mb)
## Ncells    9642503 515.0    20148211 1076.1  20148211 1076.1
## Vcells   118496631 904.1   186954797 1426.4  233675429 1782.9

gc()

##           used   (Mb) gc trigger   (Mb) max used   (Mb)
## Ncells    9639599 514.9    20148211 1076.1  20148211 1076.1
## Vcells   118495357 904.1   186954797 1426.4  233675429 1782.9

```

Regressions

As a initial step, a normal regression is run with the number of aid projects per cell as dependent variable and the number of missions per cell as the treatment variable. A moran's I test of the residual is conducted and the residuals plotted. Weights were estimated using neighbours up to a distance of 0.5 degrees, which basically means only the number of missions in directly neighbouring cells are used to create the weights.

The regression results show that having one mission in the grid cell is related to 34 more aid projects in the cell. The result is statistically significant at a 1 percent level, providing further evidence that colonial missions predict future aid projects. The large coefficient result reflect the large variation in number of projects between grid cells, as visible from the maps above. Particularly, outliers might be driving the results. For example, one cell had more than 15,000 projects.

Furthermore, the Moran I of the residuals is also statistically significant with a p-value below 1% indicating spatial autocorrelation of aid projects. This is also visible from the plot of residuals against lagged residuals. This is not surprising because so far, grid cells are spread across Africa whether the land is habitable or not, so aid projects will be spatially clustered in habitable area, leading to spatial autocorrelation.

```

# Linear regression model
formula <- formula(NumAid ~ NumMissions)
model <- lm(formula = formula, data=regression_data)
summary(model)

## 
## Call:
## lm(formula = formula, data = regression_data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -369.1    -60.3    -60.3    -51.3  15218.7 
## 
## Coefficients:

```

```

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60.254      5.294 11.382 < 2e-16 ***
## NumMissions 34.312      6.901  4.972 6.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 514.7 on 10230 degrees of freedom
## Multiple R-squared:  0.002411, Adjusted R-squared:  0.002313
## F-statistic: 24.72 on 1 and 10230 DF, p-value: 6.73e-07

#Analysing correlation in error term

# Assuming each point corresponds to the center of a raster cell
coords <- regression_data[c("x", "y")]

# Create neighbors based on a distance threshold
# Calculate the appropriate distance threshold based on the raster resolution
dist_threshold <- 0.5 # Adjust 0.5 to your raster's resolution if different

# Create neighbors
nb <- dnearneigh(coords, 0, dist_threshold, longlat = FALSE)

# Convert neighbors to weights
wts <- nb2listw(nb, style="W", zero.policy=TRUE)
moran.test(model$residuals, listw=wts)

##
## Moran I test under randomisation
##
## data: model$residuals
## weights: wts
## n reduced by no-neighbour observations
##
## Moran I statistic standard deviate = 2.4322, p-value = 0.007504
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##       1.684219e-02    -9.783778e-05    4.851126e-05

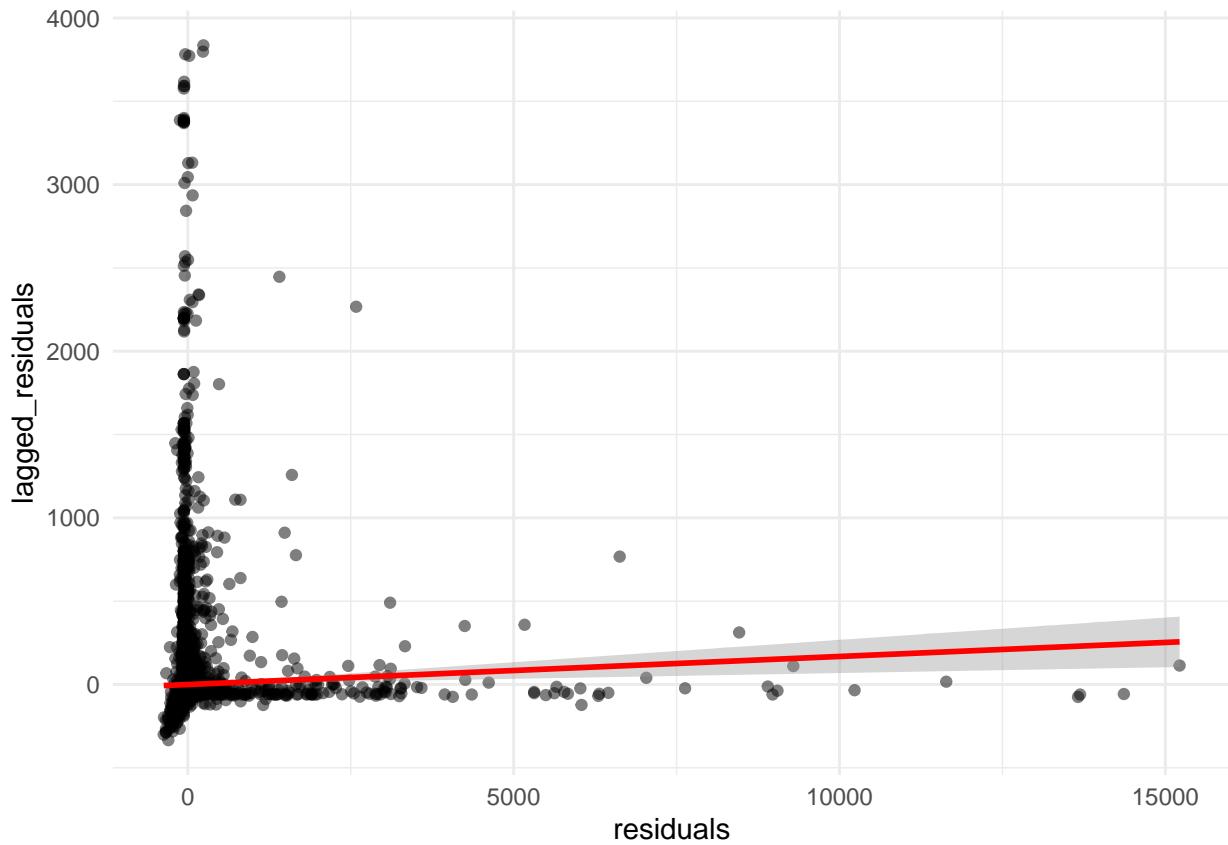
# Plot residuals
residuals_model <- residuals(model)
lagged_residuals_model <- lag.listw(wts, residuals_model)

res <- data.frame("lagged_residuals"=lagged_residuals_model, "residuals"=residuals_model)

ggplot(res, aes(x = residuals, y = lagged_residuals)) +
  theme_minimal() +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "red")

## 'geom_smooth()' using formula = 'y ~ x'

```



Spatial regression

Next, we create a lagged missions variable using weights estimated using neighbours up to a distance of 0.5 degrees as before. The lagged treatment variable was manually constructed because using inbuilt functions led to issues in calculating weights because most cells reported neither aid projects or missions. This in turn lead to the error: “NAs in lagged (in)dependent variable”

Two specifications are tested: First the lagged treatment variable is added to the initial specification. In a second model, the interaction between the missions and lagged missions is also added.

The results do not change the size or significance of the coefficient of missions, meaning that spatial autocorrelation in mission stations is not likely to be bias the estimate. The coefficient on the lag term is negative and insignificant, suggesting that the presence of missions in neighboring cells do not amplify the effect of missions on the number of aid projects.

The addition of an interaction term turn the sign of the lagged mission term around, but does not change its significance. The negative and significant coefficient estimate on the interaction term suggests that added effect of missions in the direct neighborhood is negative. In other words, missions in a given grid cell positively correlate to aid project in that cell, but there is not spillover across direct neighbors.

```
#create lag missions using neighbourhood weights
regression_data$lag_NumMissions <- lag.listw(wts, regression_data$NumMissions, zero.policy = TRUE)

# Summary to check for NA or extreme values
summary(regression_data$lag_NumMissions)
```

	##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
--	----	------	---------	--------	------	---------	------

```

##  0.0000  0.0000  0.0000  0.2079  0.2500  8.0000

# Linear model with spatial lag variable
model_with_lag <- lm(NumAid ~ NumMissions + lag_NumMissions, data = regression_data)
summary(model_with_lag)

## 
## Call:
## lm(formula = NumAid ~ NumMissions + lag_NumMissions, data = regression_data)
## 
## Residuals:
##      Min      1Q      Median      3Q      Max 
## -368.8   -60.3    -60.3   -51.2  15218.8 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 60.2671   5.4360 11.087 < 2e-16 ***
## NumMissions 34.3813   9.6053  3.579 0.000346 ***  
## lag_NumMissions -0.1323   12.7334 -0.010 0.991713  
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 514.8 on 10229 degrees of freedom
## Multiple R-squared:  0.002411, Adjusted R-squared:  0.002216 
## F-statistic: 12.36 on 2 and 10229 DF, p-value: 4.352e-06

# Linear model including main effects and interaction term
model_with_interaction <- lm(NumAid ~ NumMissions * lag_NumMissions, data = regression_data)
summary(model_with_interaction)

## 
## Call:
## lm(formula = NumAid ~ NumMissions * lag_NumMissions, data = regression_data)
## 
## Residuals:
##      Min      1Q      Median      3Q      Max 
## -352.6   -55.8    -55.8   -48.8  15217.7 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 55.831     5.617   9.939 < 2e-16 ***
## NumMissions 52.986     11.307   4.686 2.82e-06 ***  
## lag_NumMissions 21.937     14.566   1.506  0.13210  
## NumMissions:lag_NumMissions -12.404     3.981  -3.116  0.00184 ** 
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 514.5 on 10228 degrees of freedom
## Multiple R-squared:  0.003357, Adjusted R-squared:  0.003064 
## F-statistic: 11.48 on 3 and 10228 DF, p-value: 1.638e-07

```

Spatial regression restricted to habitable

To avoid grid cells in areas that are inhabitable to be included in the analysis, in this specification the same two spatial regressions as above are run on those grid cells that were located in habitable Africa - parts of the continent excluding desert (see Data Section on how this was constructed).

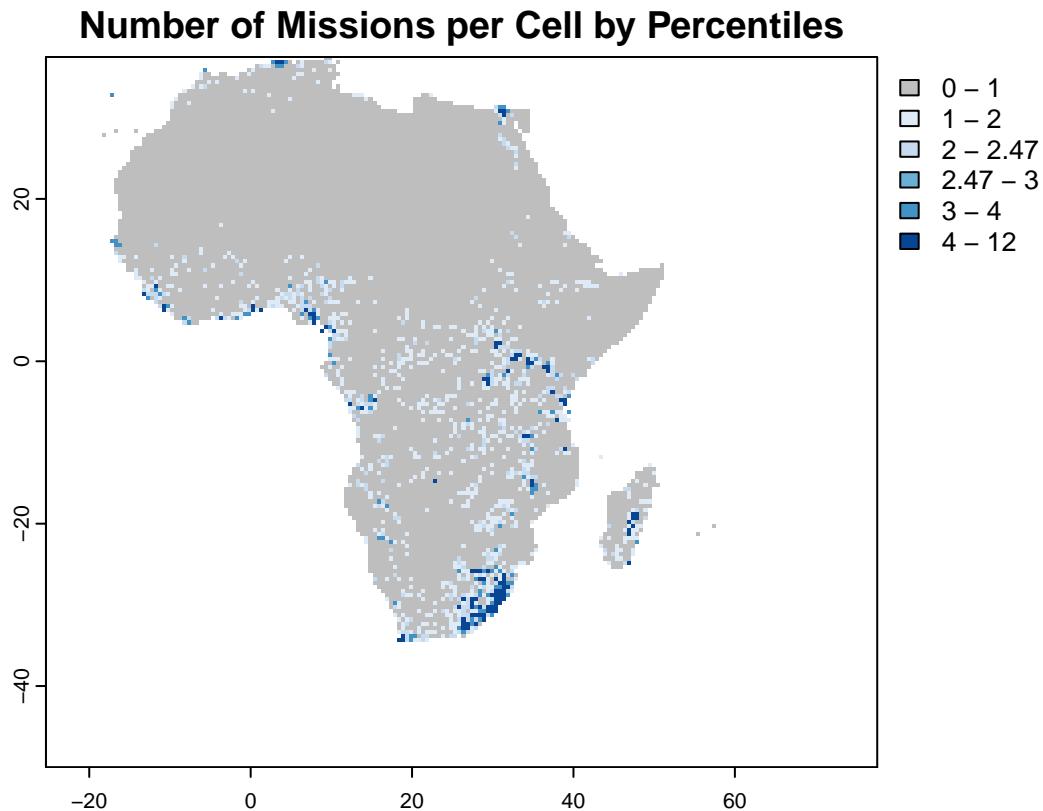
The regression results as well as the Moran's Is do not change notably.

```
### Plot aid cells excluding desert
# Calculate the percentile breaks
percentiles <- seq(0, 100, by = 1) # Adjust this as needed
breaks <- quantile(values(mission_raster2), probs = percentiles / 100, na.rm = TRUE)

# Ensure breaks are unique and sorted (adding minimum for zero coverage)
breaks <- c(min(values(mission_raster)), na.rm = TRUE), unique(sort(breaks)))

# Ensure breaks are unique in case of duplicate percentile values
breaks <- unique(breaks)

# Define a color palette including 'gray' for NA values
colors <- c("gray", brewer.pal(8, "Blues"))
plot(mission_raster, breaks=breaks, col=colors, main="Number of Missions per Cell by Percentiles")
```



```
# Extract raster values to data frames
aid_values <- values(aid_raster2)
aid_data <- data.frame(x = xy[,1], y = xy[,2], aid_projects = aid_values)
```

```

colnames(aid_data)[3] <- "NumAid"

mission_values <- values(mission_raster2)
missions_data <- data.frame(x = xy[,1], y = xy[,2], missions = mission_values)
colnames(missions_data)[3] <- "NumMissions"

# Merge the data frames by coordinates
regression_data_habitable <- merge(aid_data, missions_data, by=c("x", "y"))

# Remove NA values which may complicate the analysis
regression_data_habitable <- na.omit(regression_data)

#calculate weights

# Assuming each point corresponds to the center of a raster cell
coords <- regression_data_habitable[c("x", "y")]

# Create neighbors based on a distance threshold
# Calculate the appropriate distance threshold based on the raster resolution
dist_threshold <- 0.5 # Adjust 0.5 to your raster's resolution if different

# Create neighbors
nb <- dnearneigh(coords, 0, dist_threshold, longlat = FALSE)

# Convert neighbors to weights
wts <- nb2listw(nb, style="W", zero.policy=TRUE)

# Create lag missions using new weights
regression_data_habitable$lag_NumMissions <- lag.listw(wts, regression_data_habitable$NumMissions, zero=TRUE)

# Summary to check for NA or extreme values
summary(regression_data_habitable$lag_NumMissions)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 0.0000  0.0000  0.0000  0.2079  0.2500  8.0000

# Linear model with spatial lag variable
model_with_lag <- lm(NumAid ~ NumMissions + lag_NumMissions, data = regression_data_habitable)
summary(model_with_lag)

##
## Call:
## lm(formula = NumAid ~ NumMissions + lag_NumMissions, data = regression_data_habitable)
##
## Residuals:
##      Min       1Q     Median       3Q      Max
## -368.8    -60.3    -60.3   -51.2  15218.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60.2671    5.4360 11.087 < 2e-16 ***

```

```

## NumMissions      34.3813     9.6053    3.579 0.000346 ***
## lag_NumMissions -0.1323    12.7334   -0.010 0.991713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 514.8 on 10229 degrees of freedom
## Multiple R-squared:  0.002411, Adjusted R-squared:  0.002216
## F-statistic: 12.36 on 2 and 10229 DF, p-value: 4.352e-06

# Linear model including main effects and interaction term
model_with_interaction <- lm(NumAid ~ NumMissions * lag_NumMissions, data = regression_data_habitable)
summary(model_with_interaction)

##
## Call:
## lm(formula = NumAid ~ NumMissions * lag_NumMissions, data = regression_data_habitable)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -352.6  -55.8  -55.8  -48.8 15217.7
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  55.831     5.617   9.939 < 2e-16 ***
## NumMissions                  52.986    11.307   4.686 2.82e-06 ***
## lag_NumMissions                21.937    14.566   1.506  0.13210
## NumMissions:lag_NumMissions -12.404     3.981  -3.116  0.00184 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 514.5 on 10228 degrees of freedom
## Multiple R-squared:  0.003357, Adjusted R-squared:  0.003064
## F-statistic: 11.48 on 3 and 10228 DF, p-value: 1.638e-07

#Moran tests of residuals
moran.test(model_with_lag$residuals, listw=wts)

##
## Moran I test under randomisation
##
## data: model_with_lag$residuals
## weights: wts
## n reduced by no-neighbour observations
##
## Moran I statistic standard deviate = 2.4321, p-value = 0.007506
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
## 1.684180e-02 -9.783778e-05 4.851126e-05

moran.test(model_with_interaction$residuals, listw=wts)

```

```

## Moran I test under randomisation
## data: model_with_interaction$residuals
## weights: wts
## n reduced by no-neighbour observations
## Moran I statistic standard deviate = 2.3115, p-value = 0.0104
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##           1.600131e-02 -9.783778e-05 4.850942e-05

```

```
#
```

Spatial regression removing cells without any aid

In a final step, the same spatial regressions specifications are run with the data restricted to cells that had at least one aid project to restrict the sample even further. Under the assumptions that aid project go everywhere were humans live in Africa, this restriction has the advantage that it can control for factors beyond deserts that might affect whether the land is habitable or not. However, the restriction is not without problems because aid project might not go to areas of need. In this case, this restriction strategy is may give biased results. Still, this restriction allow to exploit the within-aid-receiving cell variation in aid and mission intensity.

The results hold up that missions are correlated with aid projects. Interstingly, the p-value of the Moran I of the residuals increases to 38% making it no longer significant, indicating that spatial autocorrelation is likely low within this sample.

```

# Removing zeros
data_no_zeros <- regression_data[regression_data$NumAid > 0, ]
coords2 <- data_no_zeros[c("x", "y")]
nb2 <- dnearneigh(coords2, 0, dist_threshold, longlat = FALSE)
listw_no_zeros <- nb2listw(nb2, style="W", zero.policy=TRUE)

# Calculating lag missions using new weights
data_no_zeros$lag_NumMissions <- lag.listw(listw_no_zeros, data_no_zeros$NumMissions, zero.policy = TRUE)

# Summary to check for NA or extreme values
summary(data_no_zeros$lag_NumMissions)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 0.0000 0.0000 0.0000 0.3187 0.3333 8.0000

# Linear model with spatial lag variable
model_with_lag <- lm(NumAid ~ NumMissions + lag_NumMissions, data = data_no_zeros)
summary(model_with_lag)

## 
## Call:
## lm(formula = NumAid ~ NumMissions + lag_NumMissions, data = data_no_zeros)
## 
```

```

## Residuals:
##      Min     1Q Median     3Q    Max
## -445.3 -139.4 -127.4 -90.6 15138.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 142.410   12.482 11.409 <2e-16 ***
## NumMissions  34.099   15.521  2.197  0.0281 *
## lag_NumMissions -8.552   20.492 -0.417  0.6765
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 761.7 on 4578 degrees of freedom
## Multiple R-squared:  0.001314, Adjusted R-squared:  0.0008775
## F-statistic: 3.011 on 2 and 4578 DF, p-value: 0.04932

# Linear model including main effects and interaction term
model_with_interaction <- lm(NumAid ~ NumMissions * lag_NumMissions, data = data_no_zeros)
summary(model_with_interaction)

##
## Call:
## lm(formula = NumAid ~ NumMissions * lag_NumMissions, data = data_no_zeros)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -497.1 -137.4 -127.4 -90.4 15139.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 140.394   13.053 10.756 <2e-16 ***
## NumMissions  40.080   19.211  2.086  0.037 *
## lag_NumMissions -2.071   23.882 -0.087  0.931
## NumMissions:lag_NumMissions -4.285    8.108 -0.528  0.597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 761.8 on 4577 degrees of freedom
## Multiple R-squared:  0.001375, Adjusted R-squared:  0.0007202
## F-statistic:  2.1 on 3 and 4577 DF, p-value: 0.09801

#Moran test of residuals
moran.test(model_with_lag$residuals, listw=listw_no_zeros)

##
## Moran I test under randomisation
##
## data: model_with_lag$residuals
## weights: listw_no_zeros
## n reduced by no-neighbour observations
##
## Moran I statistic standard deviate = 0.2952, p-value = 0.3839
## alternative hypothesis: greater

```

```

## sample estimates:
## Moran I statistic      Expectation      Variance
##          0.0037319721    -0.0002334267    0.0001804424

moran.test(model_with_interaction$residuals, listw=listw_no_zeros)

```

```

##
## Moran I test under randomisation
##
## data: model_with_interaction$residuals
## weights: listw_no_zeros
## n reduced by no-neighbour observations
##
## Moran I statistic standard deviate = 0.29277, p-value = 0.3849
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##          0.0036992607    -0.0002334267    0.0001804418

```

Discussion

Future research could add more covariates to control for other factors that predict both missions and aid projects and could bias the results. Furthermore, uninhabitable areas could have included lakes and other bodies of water or dense forest area, as well as national parks.

Conclusion

This research project analysed the spatial correlation between colonial missions and contemporary aid projects in Africa. It found that there seems to be a correlation between the two that persists over different specifications and methods.

References

- Matteo Alpino and Eivind Moe Hammersmark (2021). “The Role of Historical Christian Missions in the Location of World Bank Aid in Africa”, World Bank Policy Research Working Paper, 9146
- Gelb (2023). “Network k Functions”
- Christine Hedde-von Westernhagen and Bastian Becker (2022). “Mapping Missions: New Data for the Study of African History”, Research Data Journal for the Humanities and Social Sciences, 7(1)
- Nathan Nunn (2010). Religious conversion in colonial Africa. American Economic Review: Papers & Proceedings, 100 (2), 147–152.