Calculating conversion tables

Cornelius Erfort

5/17/2021

Contents

1 Setting up

This script requires the shape files and election results data which are not included on GitHub. All files except the shapefiles until 2009 are available for download online. See the below for more details.

1.1 Loading packages

```
packages <- c("stringr", "readxl", "dplyr", "tidyr", "plyr", "httr", "rvest", "urltools",
    "XML", "gnumeric", "english", "gsubfn", "zoo", "haven", "xlsx", "lubridate",
    "ggplot2", "maptools", "sp", "rgdal", "tools", "readr", "spatstat", "rgeos",
    "corrgram", "ggrepel")

lapply(packages[!(packages %in% rownames(installed.packages()))], install.packages)

invisible(lapply(packages, require, character.only = T))
```

2 2017

2.1 Load 2017 county-level geodata

In a first step, the shapefiles and election results for 2017 are loaded. The years until 1953 will be converted to 2017 counties.

```
# Read shapefile
counties_17 <- readOGR(dsn = "data/county shapefiles/2013+2017/Kreisgrenzen_2017_mit_Einwohnerzahl-shp"
    layer = "Kreisgrenzen_2017_mit_Einwohnerzahl", stringsAsFactors = FALSE) %>%
    spTransform(CRS("+proj=longlat +datum=WGS84"))
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 401 features
## It has 23 fields
```

```
# Remove East Germany (including Berlin)
counties_17 <- counties_17[counties_17$AGS < 11000, ]

# Plot map
plot(counties_17)
```

```
# Add rownumber
counties_17@data$rownumber <- row(counties_17@data)[, 1]
```

2.2 Load 2017 voting results

```
# Read election results
results_17 <- read.csv2("data/Municipality-level results/btw17_wbz/btw17_wbz_zweitstimmen.csv",
    skip = 4, fileEncoding = "latin1") %>%
    filter(Land < 11)  # Remove East Germany

# Generate a county key
results_17$AGS <- str_c(str_pad(results_17$Land, 2, pad = "0"), results_17$Regierungsbezirk,
    str_pad(results_17$Kreis, 2, pad = "0"))

# Only keep relevant vars
results_17 <- select(results_17, c(Land, Gültige, Wahlberechtigte..A., CDU, CSU,
    SPD, DIE.LINKE, GRÜNE, FDP, AfD, AGS))
```

```
# Aggregate by county
results_17 <- aggregate(. ~ Land + AGS, data = results_17, FUN = sum)

# Combine CDU.CSU (and remove individual columns)
results_17$CDU.CSU <- results_17$CDU + results_17$CSU
results_17 <- select(results_17, -c(CDU, CSU)) %>%
    dplyr::rename(state = Land, valid_votes = Gültige, reg_voters = Wahlberechtigte..A.)

# Add year suffix to varnames
names(results_17)[-c(1:2)] <- str_c(names(results_17)[-c(1:2)], ".17")

# Make vote shares (party votes divided by valid votes)
results_17[, names(results_17) %>%
    str_subset("(CDU)|(CSU)|(SPD)|(FDP)|(LINKE)|(GRÜNE)|(AfD)")] <- apply(results_17[,
    names(results_17) %>%
        str_subset("(CDU)|(CSU)|(SPD)|(FDP)|(LINKE)|(GRÜNE)|(AfD)")], MARGIN = 2,
    FUN = function(x) x/results_17[, names(results_17) %>%
        str_subset("valid_votes")])
```

3 Calculate intersections with 2017

3.1 County level: 1953-1976

3.1.1 Geodata: 1953-1976

```
county_years <- c(1953, 1957, 1961, 1965, 1969, 1972, 1976)

for (year in county_years) {
    print(year)

    # Get 2-digit year
    (shyear <- substr(year, 3, 4) %>%
        as.numeric())

    # Load county-level geodata for that year
    counties_ <- readOGR(dsn = "data/county shapefiles/1949-1989/Federal Republic of Germany and German
        layer = str_c("Germany_", year, "_v.1.0"), stringsAsFactors = FALSE, verbose = F) %>%
        spTransform(CRS("+proj=longlat +datum=WGS84"))

    # Correct duplicate counties (usually city and surroundings, e.g. Augsburg and
    # Augsburg (Stadt))
    for (county in counties_@data$GEN[duplicated(counties_@data$GEN)]) {
        # cat(county, ', ')
        counties_@data$GEN[counties_@data$AREA == min(counties_@data$AREA[counties_@data$GEN ==
            county]) & counties_@data$GEN == county] <- counties_@data$GEN[counties_@data$AREA ==
            min(counties_@data$AREA[counties_@data$GEN == county]) & counties_@data$GEN ==
            county] %>%
            str_c(., ", Stadt")
    }

    # Remove East Germany (including Berlin)
    counties_ <- counties_[counties_$KREIS_KENN < 11001000, ]
```

```r
    rownames(counties_@data) <- NULL

    # Remove Saarland for 1953 (not part of Germany at the time)
    if (year == 1953)
        counties_ <- counties_[(counties_@data$KREIS_ID %>%
            substr(1, 2)) != "10", ]

    # Add rownumber
    counties_@data$rownumber <- row(counties_@data)[, 1]

    # Find intersections between year and 2017
    intersections <- gIntersects(counties_17, counties_, byid = TRUE)  # Matrix that indicates intersec
    dim(intersections)  # Dimensions (height x width)

    # Calculate intersection shares between year and 2017
    intersection_shares <- matrix(nrow = nrow(intersections), ncol = ncol(intersections))  # Create mat
    # Go through all intersections and calculate areas Load if file already exists Go
    # through all rows cat('o') # Report progess Go through all intersecting columns
    # cat('.') # Report progess
    if (!file.exists(str_c("conversion tables/intersection_shares_", shyear, ".RData"))) {
        for (new_shape in 1:ncol(intersections)) {
            for (old_shape in which(intersections[, new_shape])) {

                intersection <- gIntersection(counties_17[new_shape, ], gBuffer(counties_[old_shape,
                  ], byid = TRUE, width = 0), checkValidity = T)

                # Area of intersection between 'old' and 2017 counties divided by area of 'old'
                # counties
                intersection_shares[old_shape, new_shape] <- gArea(gIntersection(counties_17[new_shape,
                  ], gBuffer(counties_[old_shape, ], byid = TRUE, width = 0), checkValidity = T))/gArea
                  ])
            }
        }

        # Save intersection_shares
        save(intersection_shares, file = str_c("conversion tables/intersection_shares_",
            shyear, ".RData"))

    } else load(str_c("conversion tables/intersection_shares_", shyear, ".RData"))

    # Create object for geodata for that year
    assign(str_c("counties_", shyear), counties_)
    cat("\n\n")
}
```

```
## [1] 1953
##
##
## [1] 1957
##
##
## [1] 1961
##
##
```

```
## [1] 1965
##
##
## [1] 1969
##
##
## [1] 1972
##
##
## [1] 1976
```

Correct data with hand coded lists

3.1.2 Election data: 1953-1976

```
vote <- data.frame()
counties_GEN <- data.frame()

for (year in county_years) {
    print(year)

    # 2-digit year
    (shyear <- substr(year, 3, 4) %>%
        as.numeric())

    # Load voting results for that year
    vote <- read.csv2(str_c("data/County-level results/btw", year, "kreis.csv"),
        skip = ifelse(shyear %in% c(57, 61, 65), 5, 4), fileEncoding = ifelse(shyear ==
            76, "latin1", "UTF-8")) %>%
        filter(!is.na(Lfd..Nr.)) %>%
        select(c(Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis, Gültige,
            Wahlberechtigte, CDU, SPD, F.D.P., CSU)) %>%
        dplyr::rename(FDP = F.D.P.) %>%
        mutate(year = year) %>%
        rbind.fill(vote)  # Select and rename vars

    counties_GEN <- rbind.fill(counties_GEN, data.frame(GEN = get(str_c("counties_",
        shyear))@data$GEN, year = year))

}
```

```
## [1] 1953
## [1] 1957
## [1] 1961
## [1] 1965
## [1] 1969
## [1] 1972
## [1] 1976
```

Correct data with hand coded lists because some names do not match the geodata.

```
# Correct data
vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[vote$Kreisfreie.Stadt.bzw..Stadtkreis..Land
    "Friedberg" & vote$CSU > 0] <- "Friedberg, Stadt"  # There are two counties with this name (one is

vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[vote$Kreisfreie.Stadt.bzw..Stadtkreis..Land
```

```
    "Konstanz, Stadt"] <- "Konstanz"  # Konstanz is only one county in the geodata
vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landk
    "Lüdenscheid, Stadt"] <- "Lüdenscheid"  # Lüdenscheid is only one county in the geodata
vote <- aggregate(. ~ Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis + year,
    data = vote, FUN = sum)

vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[str_detect(vote$Kreisfreie.Stadt.bzw..Stadtk
    "(Neuss)|(Neuß)]")] <- "Neuss"

votenames <- read.xlsx("county names/53-76votenames.xlsx", sheetIndex = 1)
for (i in 1:nrow(votenames)) vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[vote$Kreisfrei
    votenames$name1[i]] <- votenames$name2[i]

# Combine CDU.CSU
vote <- mutate(vote, CDU.CSU = CDU + CSU) %>%
    select(-c(CDU, CSU)) %>%
    dplyr::rename(valid_votes = Gültige, reg_voters = Wahlberechtigte)

# Export for manual matching
table(vote$year)
```

```
##
## 1953 1957 1961 1965 1969 1972 1976
##  556  565  565  564  549  428  342
```

```
table(counties_GEN$year)
```

```
##
## 1953 1957 1961 1965 1969 1972 1976
##  556  565  565  564  549  427  342
```

```
vote <- vote[!(vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis == "Donaueschingen" &
    vote$year == 1972), ]

for (year in county_years) {
    print(year)

    (shyear <- substr(year, 3, 4) %>%
        as.numeric())  # 2-digit year
    code_GEN <- counties_GEN$GEN[counties_GEN$year == year] %>%
        unique %>%
        sort
    code_VOTE <- vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[vote$year ==
        year] %>%
        unique %>%
        sort

    data.frame(GEN = code_GEN, VOTE = code_VOTE) %>%
        mutate(check = !(GEN == VOTE)) %>%
        write.xlsx(str_c("county names/", shyear, "_kreis2shapefile.xlsx"))

}
```

```
## [1] 1953
## [1] 1957
## [1] 1961
## [1] 1965
## [1] 1969
## [1] 1972
## [1] 1976
```

```
for (year in county_years) {
    print(year)

    (shyear <- substr(year, 3, 4) %>%
        as.numeric())  # 2-digit year
    code_GEN <- counties_GEN$GEN[counties_GEN$year == year] %>%
        unique %>%
        sort
    code_VOTE <- vote$Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis[vote$year ==
        year] %>%
        unique %>%
        sort

    data.frame(GEN = code_GEN, VOTE = code_VOTE) %>%
        mutate(check = !(GEN == VOTE)) %>%
        write.xlsx(str_c("county names/", shyear, "_kreis2shapefile.xlsx"))

}
```

```
## [1] 1953
## [1] 1957
## [1] 1961
## [1] 1965
## [1] 1969
## [1] 1972
## [1] 1976
```

3.1.3 Project the 1953-1976 election results onto 2017

```
# Start all results dataframe with 2017 results
results <- results_17

# Go through years
for (year in county_years) {
    print(year)

    (shyear <- substr(year, 3, 4) %>%
        as.numeric())  # 2-digit year

    # Load geodata and vote results for that year
    counties_ <- get(str_c("counties_", shyear))
    vote_ <- vote[vote$year == year, ]

    # Add geodata county names to results
    vote_ <- merge(vote_, read.xlsx(str_c("county names/", shyear, "_kreis2shapefile_.xlsx"),
        sheetIndex = 1, colIndex = 2:3), by.x = "Kreisfreie.Stadt.bzw..Stadtkreis..Landkreis.bzw..Kreis
        by.y = "VOTE", all.x = T)

    # Add results to geodata
    counties_@data <- merge(counties_@data, vote_, by = "GEN", all.x = T)

    # Bring in original order
    counties_ <- counties_[order(counties_@data$rownumber), ]

    # Load intersection shares
    load(str_c("conversion tables/intersection_shares_", shyear, ".RData"))

    # Construct hypothetical results by 2017 counties
    t_results <- data.frame(AGS = counties_17@data$AGS, valid_votes = colSums(counties_@data$valid_votes
        intersection_shares, na.rm = T), reg_voters = colSums(counties_@data$reg_voters *
        intersection_shares, na.rm = T), CDU.CSU = colSums(counties_@data$CDU.CSU *
        intersection_shares, na.rm = T), SPD = colSums(counties_@data$SPD * intersection_shares,
        na.rm = T), FDP = colSums(counties_@data$FDP * intersection_shares, na.rm = T))

    # Write year in varnames
    names(t_results)[-1] <- str_c(names(t_results)[-1], ".", shyear)

    # Make vote shares
    t_results[, names(t_results) %>%
        str_subset("(CDU)|(CSU)|(SPD)|(FDP)")] <- apply(t_results[, names(t_results) %>%
        str_subset("(CDU)|(CSU)|(SPD)|(FDP)")], MARGIN = 2, FUN = function(x) x/t_results[,
        names(t_results) %>%
            str_subset("valid_votes")])

    # Add this year to all results dataframe
    results <- merge(results, t_results, by = "AGS", all = T)

}
```

```
## [1] 1953
## [1] 1957
## [1] 1961
## [1] 1965
## [1] 1969
## [1] 1972
## [1] 1976
```

```
# Set 1953 results for Saarland to NA
results[results$state == "10", names(results) %>%
    str_subset("\\.53")] <- NA
```

3.2 Municipality level: 1980-2013

3.2.1 Geodata: 1980-2013

```
munip_years <- c(1980, 1983, 1987, 1990, 1994, 1998, 2002, 2005, 2009, 2013)

for (year in munip_years) {
    print(year)

    (shyear <- substr(year, 3, 4))  # 2-digit year

    # Load county-level geodata for that year
    if (year < 1990)
        counties_ <- readOGR(dsn = "data/county shapefiles/1949-1989/Federal Republic of Germany and Ge
            layer = str_c("Germany_", ifelse(year == 1983, "1981_83", ifelse(year ==
                1987, "1984_87", year)), "_v.1.0"), stringsAsFactors = FALSE) %>%
            spTransform(CRS("+proj=longlat +datum=WGS84"))  # Read layer/shapefile

    if (year >= 1990 & year <= 2009)
        counties_ <- readOGR(dsn = "data/county shapefiles/1990-2009/Federal Republic of Germany 1990-2
            layer = str_c("Germany_", ifelse(year == 2005, "2004_06", ifelse(year ==
                2002, "2001_02", year)), ifelse(year == 1990, "_91", ""), "_v.1.0"),
            stringsAsFactors = FALSE) %>%
            spTransform(CRS("+proj=longlat +datum=WGS84"))  # Read layer/shapefile

    if (year == 2013) {
        counties_ <- readOGR(dsn = "data/county shapefiles/2013+2017/Kreisgrenzen_2013-shp",
            layer = "KRS_2013", stringsAsFactors = FALSE) %>%
            spTransform(CRS("+proj=longlat +datum=WGS84"))  # Read layer/shapefile
        names(counties_)[names(counties_) == "AGS"] <- "KREIS_KENN"
    }

    # Make var AGS
    counties_$AGS <- substr(counties_$KREIS_KENN, 1, 5)

    # Remove East Germany (including Berlin)
    counties_ <- counties_[as.numeric(counties_$KREIS_KENN) < 1.1e+07, ]
    rownames(counties_@data) <- NULL

    # Add rownumber
    counties_@data$rownumber <- row(counties_@data)[, 1]

    # Find intersections between year and 2017
    intersections <- gIntersects(counties_17, counties_, byid = TRUE)  # Matrix that indicates intersec
    dim(intersections)  # Dimensions (height x width)

    # Calculate intersection shares between year and 2017
    intersection_shares <- matrix(nrow = nrow(intersections), ncol = ncol(intersections))  # Create mat
    # Go through all intersections and calculate areas Load if file already exists
    if (!file.exists(str_c("conversion tables/intersection_shares_", shyear, ".RData"))) {
        for (new_shape in 1:ncol(intersections)) {
            cat("o")
            for (old_shape in which(intersections[, new_shape])) {
                cat(".")
                intersection <- gIntersection(counties_17[new_shape, ], gBuffer(counties_[old_shape,
                  ], byid = TRUE, width = 0), checkValidity = T) %>%
                  suppressWarnings()
                intersection_shares[old_shape, new_shape] <- gArea(gIntersection(counties_17[new_shape,
                  ], gBuffer(counties_[old_shape, ], byid = TRUE, width = 0), checkValidity = T))/gArea
                  ])  # Area of intersection between 'old' and 2017 counties divided by area of 'old' c
            }
        }
        # Save intersection_shares
        save(intersection_shares, file = str_c("conversion tables/intersection_shares_",
            shyear, ".RData"))
    } else load(str_c("conversion tables/intersection_shares_", shyear, ".RData"))

    # Create object for geodata for that year
    assign(str_c("counties_", shyear), counties_)
    cat("\n\n")

}
```

```
## [1] 1980
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 547 features
## It has 7 fields
##
##
## [1] 1983
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 547 features
## It has 6 fields
##
##
## [1] 1987
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 547 features
## It has 6 fields
##
##
## [1] 1990
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 543 features
## It has 6 fields
##
##
## [1] 1994
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 444 features
## It has 7 fields
##
##
## [1] 1998
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 440 features
## It has 7 fields
##
##
## [1] 2002
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 439 features
## It has 7 fields
##
##
## [1] 2005
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 439 features
## It has 7 fields
##
##
## [1] 2009
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 412 features
## It has 7 fields
##
##
## [1] 2013
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/cornelius/Desktop/DYNAMICS/projects/GitHub/germany-53-17-districts/data/county shape
## with 402 features
## It has 17 fields
```

3.2.2 Election data: 1980-2013

```
for (year in munip_years) {
    print(year)

    (shyear <- substr(year, 3, 4))  # 2-digit year

    if (year == 1980)
        vote_ <- read.csv2(list.files(str_c("data/Municipality-level results/btw",
            shyear, "_wbz"), full.names = T) %>%
            str_subset("eitst.*\\.csv"), skip = 4, fileEncoding = ifelse(year %in%
            c(1980), "UTF-8", "latin1")) %>%
            suppressWarnings() else {
        vote_ <- read_xlsx(list.files(str_c("data/Municipality-level results/btw",
            substr(year, 3, 4), "_wbz"), full.names = T) %>%
            str_subset("eitst.*\\.xlsx"), skip = 4, col_types = "text") %>%
            suppressWarnings()
        vote_[, (str_detect(names(vote_), "Wahlberechtigte") %>%
            which %>%
            min):ncol(vote_)] <- apply(vote_[, (str_detect(names(vote_), "Wahlberechtigte") %>%
            which %>%
            min):ncol(vote_)], MARGIN = 2, FUN = function(x) as.numeric(x))
    }

    vote_ <- filter(vote_, as.numeric(Land) < 11)  # Remove East Germany

    # Reduce to relevant vars
    vote_$AGS <- str_c(str_pad(vote_$Land, 2, pad = "0"), vote_$Regierungsbezirk,
        str_pad(vote_$Kreis, 2, pad = "0"))
    vote_ <- select(vote_, c(Land, names(vote_) %>%
        str_subset("Gültig(e)?"), names(vote_) %>%
        str_subset("Wahlberechtigte(..A.)|( \\(A\\))"), AGS, names(vote_) %>%
        str_subset("(CDU)|(CSU)|(SPD)|(LINKE)|(PDS)|(GRÜNE)|(FDP)|(AfD)")))
    vote_ <- aggregate(. ~ Land + AGS, data = vote_, FUN = sum)
    # vote_ <- vote_[-1, ]

    vote_ <- dplyr::rename(vote_, state = Land, valid_votes = names(vote_) %>%
        str_subset("Gültig(e)?"), reg_voters = names(vote_) %>%
        str_subset("Wahlberechtigte(..A.)|( \\(A\\))"))

    # Rename party vars
    if ("CDU/CSU" %in% names(vote_))
        names(vote_)[names(vote_) == "CDU/CSU"] <- "CDU.CSU"
    if ("B90/Gr" %in% names(vote_))
        names(vote_)[names(vote_) == "B90/Gr"] <- "GRÜNE"
    if ("PDS" %in% names(vote_))
        names(vote_)[names(vote_) == "PDS"] <- "DIE.LINKE"
    if ("DIE LINKE" %in% names(vote_))
        names(vote_)[names(vote_) == "DIE LINKE"] <- "DIE.LINKE"

    # Combine CDU.CSU (if separate)
    if ("CDU" %in% names(vote_)) {
        vote_$CDU.CSU <- vote_$CDU + vote_$CSU
        vote_ <- select(vote_, -c(CDU, CSU))
    }

    # names(vote_)[-c(1:2)] <- str_c(names(vote_)[-c(1:2)], '.', shyear)

    vote_$KREIS_KENN <- str_c(vote_$AGS, "000")
    vote_$year <- year

    vote <- rbind.fill(vote, vote_)
}
```

```
## [1] 1980
## [1] 1983
## [1] 1987
## [1] 1990
## [1] 1994
## [1] 1998
## [1] 2002
## [1] 2005
## [1] 2009
## [1] 2013
```

3.2.3 Project the 1980-2013 election results onto 2017

```
# Go through years
for (year in munip_years) {
    print(year)

    (shyear <- substr(year, 3, 4))  # 2-digit year

    # Load geodata and vote results for that year
    counties_ <- get(str_c("counties_", shyear))
    vote_ <- vote[vote$year == year, ]

    # Add results to geodata
    counties_@data <- merge(counties_@data, vote_, by = "KREIS_KENN", all.x = T)

    # Bring original order
    counties_ <- counties_[order(counties_@data$rownumber), ]

    # Load intersection shares
    load(str_c("conversion tables/intersection_shares_", shyear, ".RData"))

    # Construct hypothetical results by 2017 counties
    t_results <- data.frame(AGS = counties_17@data$AGS, valid_votes = colSums(counties_@data$valid_votes
        intersection_shares, na.rm = T), reg_voters = colSums(counties_@data$reg_voters *
        intersection_shares, na.rm = T), CDU.CSU = colSums(counties_@data$CDU.CSU *
        intersection_shares, na.rm = T), SPD = colSums(counties_@data$SPD * intersection_shares,
        na.rm = T), FDP = colSums(counties_@data$FDP * intersection_shares, na.rm = T))

    if (year >= 1980)
        t_results$GRÜNE <- colSums(counties_@data$GRÜNE * intersection_shares,
            na.rm = T)
    if (year >= 1990)
        t_results$DIE.LINKE <- colSums(counties_@data$DIE.LINKE * intersection_shares,
            na.rm = T)
    if (year >= 2013)
        t_results$AfD <- colSums(counties_@data$AfD * intersection_shares, na.rm = T)

    # Write year in varnames
    names(t_results)[-1] <- str_c(names(t_results)[-1], ".", shyear)

    # Make vote shares
    t_results[, names(t_results) %>%
        str_subset("(CDU)|(CSU)|(SPD)|(FDP)|(GRÜNE)|(LINKE)|(PDS)|(AfD)")] <- apply(t_results[,
        names(t_results) %>%
            str_subset("(CDU)|(CSU)|(SPD)|(FDP)|(GRÜNE)|(LINKE)|(PDS)|(AfD)")],
        MARGIN = 2, FUN = function(x) x/t_results[, names(t_results) %>%
            str_subset("valid_votes")])

    # Add this year to all results dataframe
    results <- merge(results, t_results, by = "AGS", all = T)

}
```

```
## [1] 1980
## [1] 1983
## [1] 1987
## [1] 1990
## [1] 1994
## [1] 1998
## [1] 2002
## [1] 2005
## [1] 2009
## [1] 2013
```

4 Creating the dataset

```
# Prepare var names for pivoting
names(results) <- str_replace_all(names(results), "\\.", "_")
names(results) <- str_replace(names(results), "_(?=[:alpha:])", "-")

save(results, file = "data/results-53-17.RData")
load("data/results-53-17.RData")

# Convert dataset to long format
results_long <- pivot_longer(results, cols = 3:ncol(results))

# Add year var
results_long$year <- results_long$name %>%
    str_extract("[:digit:]{2}$") %>%
    as.numeric()
results_long$year[results_long$year > 20] <- results_long$year[results_long$year >
    20] + 1900
results_long$year[results_long$year < 20] <- results_long$year[results_long$year <
    20] + 2000
unique(results_long$year)
```

```
##  [1] 2017 1953 1957 1961 1965 1969 1972 1976 1980 1983 1987 1990 1994 1998 2002
## [16] 2005 2009 2013
```

```
results_long$name <- results_long$name %>%
    str_remove("_[:digit:]{2}$")

results_long <- pivot_wider(results_long, id_cols = c("AGS", "state", "year"))

results_long <- pivot_longer(results_long, cols = names(results_long)[!str_detect(names(results_long),
    "(AGS)|(state)|(valid-votes)|(reg-voters)|(year)")], values_to = "vote_share",
    names_to = "party")

results_long <- merge(results_long, counties_17@data %>%
    select(c(AGS, GEN)) %>%
    dplyr::rename(countyname = GEN), all.x = T, by = "AGS")

save(results_long, file = "election-results-53-17.RData")

# Make varnames compatible with Stata
names(results_long) <- names(results_long) %>%
    str_replace_all("(_)|(-)|(\\.)", "")
write_dta(results_long, "election-results-53-17.dta")

load("election-results-53-17.RData")
```

5 Evaluation

```
# Look at intersections between 1953 and 2017

year <- 1953
shyear <- 53
```

```r
load(str_c("conversion tables/intersection_shares_", shyear, ".RData"))

# Correct duplicates
for (county in counties_53@data$GEN[duplicated(counties_53@data$GEN)]) {
    cat(county)
    counties_53@data$GEN[counties_53@data$AREA == min(counties_53@data$AREA[counties_53@data$GEN ==
        county]) & counties_53@data$GEN == county] <- counties_53@data$GEN[counties_53@data$AREA ==
        min(counties_53@data$AREA[counties_53@data$GEN == county]) & counties_53@data$GEN ==
        county] %>%
        str_c(., ", Stadt")
}

# Remove East Germany (including Berlin)
counties_53 <- counties_53[counties_53$KREIS_KENN < 11001000, ]
rownames(counties_53@data) <- NULL

# Remove Saarland for 1953
if (year == 1953) counties_53 <- counties_53[(counties_53@data$KREIS_ID %>%
    substr(1, 2)) != "10", ]  # (not part of Germany at the time)

# Add rownumber
counties_53@data$rownumber <- row(counties_53@data)[, 1]

# Add geographical center of districts (centroids)
centroids <- data.frame()
counties_53@data$long_mean <- NA
counties_53@data$lat_mean <- NA
for (i in 1:nrow(counties_53)) {
    centroids <- gCentroid(counties_53[i, ])@coords %>%
        as.data.frame()
    counties_53@data$long_mean[i] <- centroids$x
    counties_53@data$lat_mean[i] <- centroids$y
}
counties_17@data$long_mean <- NA
counties_17@data$lat_mean <- NA
for (i in 1:nrow(counties_17)) {
    centroids <- gCentroid(counties_17[i, ])@coords %>%
        as.data.frame()
    counties_17@data$long_mean[i] <- centroids$x
    counties_17@data$lat_mean[i] <- centroids$y
}


intersections <- gOverlaps(counties_17, counties_53, byid = TRUE)  # Matrix that indicates intersection

# Example , add = T i <- i + 1 # 23, 26, 27, 30, 35, 95

i <- 95
print(i)
```

```
## [1] 95
```

```r
# 2017 county name
counties_17$GEN[i]
```

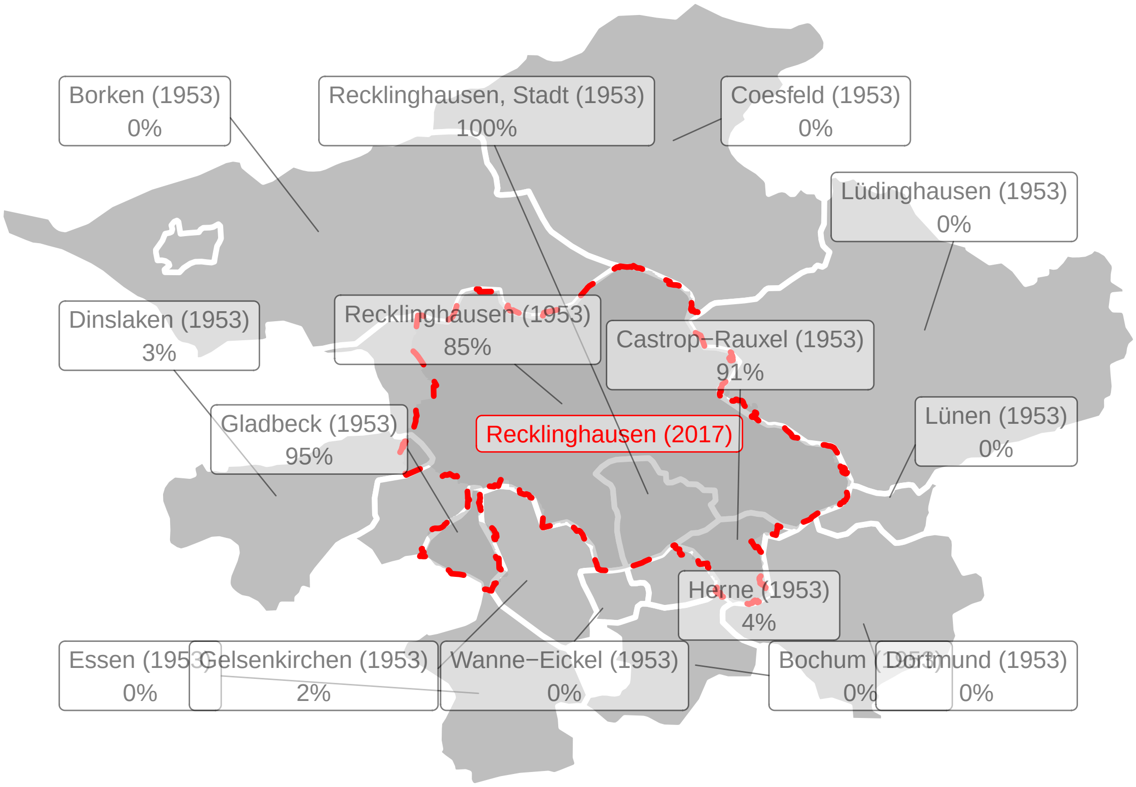
```
## [1] "Recklinghausen"
```

```
# Made up of these 1953 counties:
str_c(round(intersection_shares[which(intersections[, i]), i] * 100, digits = 1),
    "% ", counties_53$GEN[which(intersections[, i])])
```

```
##  [1] "0% Essen"                   "2.1% Gelsenkirchen"
##  [3] "94.8% Gladbeck"             "3% Dinslaken"
##  [5] "84.6% Recklinghausen"       "0% Wanne-Eickel"
##  [7] "0.1% Coesfeld"              "0.4% Lüdinghausen"
##  [9] "0.1% Bochum"                "100% Recklinghausen, Stadt"
## [11] "91.1% Castrop-Rauxel"       "0.1% Dortmund"
## [13] "0.1% Borken"                "0% Lünen"
## [15] "3.9% Herne"
```

```
ggplot() + theme_void() + geom_polygon(data = counties_53[which(intersections[, i]),
    ] %>%
    fortify, aes(long, lat, group = group), fill = "grey", size = 1, color = "white") +
    geom_polygon(data = counties_17[i, ] %>%
        fortify, aes(long, lat, group = group), fill = "dark grey", alpha = 0.5,
        size = 1, color = "red", lty = 2) + geom_label(aes(label = counties_17$GEN[i] %>%
    str_c(" (2017)"), x = counties_17$long_mean[i], y = counties_17$lat_mean[i]),
    alpha = 0.5, color = "red", size = 3) + geom_label_repel(aes(label = counties_53@data$GEN[which(int
    i])] %>%
    str_c(" (", year, ")") %>%
    str_c("\n", round(intersection_shares[intersections[, i], i] * 100, digits = 0),
        "%"), x = counties_53$long_mean[which(intersections[, i])], y = counties_53$lat_mean[which(inte
    i])]), box.padding = 3, size = 3, segment.size = 0.25, min.segment.length = 0.1,
    point.padding = 1, max.overlaps = 100, alpha = 0.5) +
ggsave("plots/conversion_example.pdf", device = cairo_pdf, width = 5 * 2^0.5, height = 5) +
    ggsave("plots/conversion_example.png", device = png, width = 1500 * 2^0.5, height = 1500,
        res = 300, limitsize = F)
```

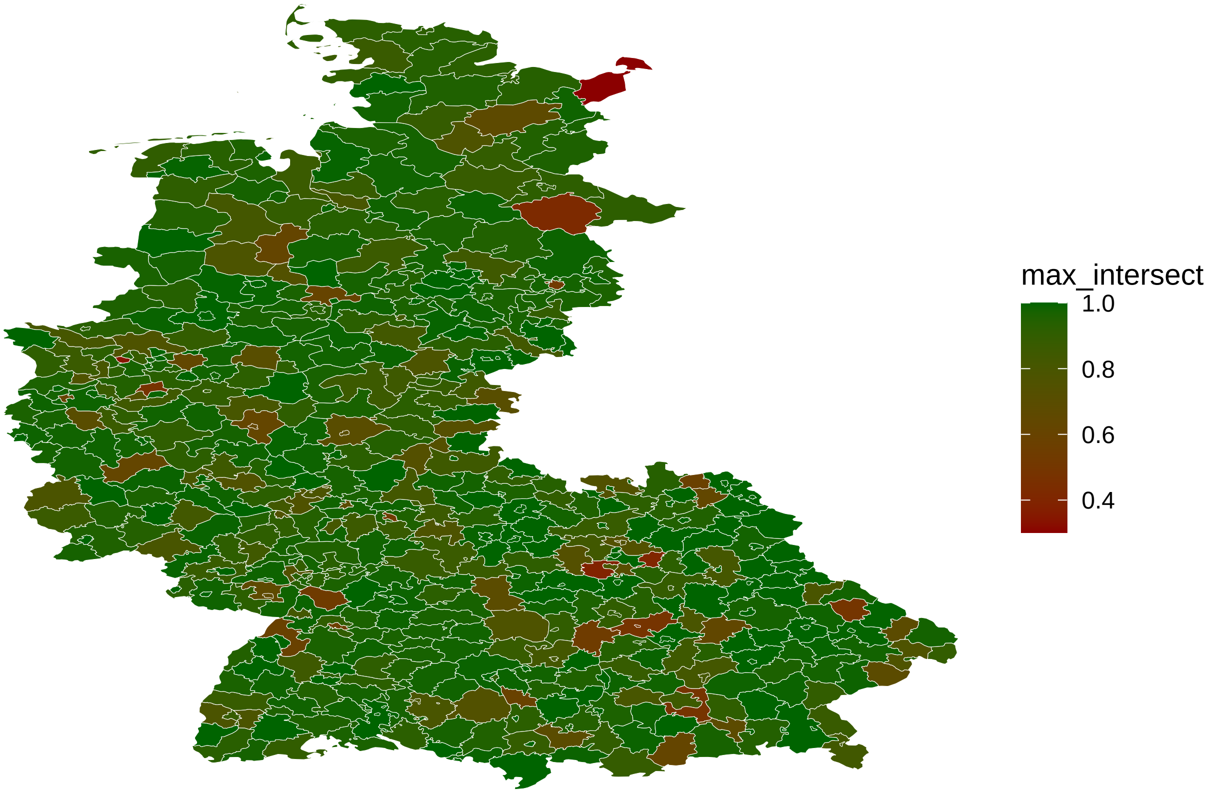
```
## Regions defined for each Polygons
## Regions defined for each Polygons
```



```r
# Coverage
counties_53@data$id <- rownames(counties_53@data)
counties_53$max_intersect <- apply(intersection_shares, 1, function(x) max(x, na.rm = T))

ggplot() + theme_void() + geom_polygon(data = counties_53 %>%
    fortify %>%
    join(counties_53@data, by = "id"), aes(long, lat, group = group, fill = max_intersect),
    size = 0.1, color = "white") + scale_fill_gradient(low = "dark red", high = "dark green",
    na.value = "dark green") + ggsave("plots/coverage_map_1953-2017.pdf", device = cairo_pdf,
    width = 4.5, height = 5) + ggsave("plots/coverage_map_1953-2017.png", device = png,
    width = 1500 * 4.5/5, height = 1500, res = 300, limitsize = F)
```

```
## Regions defined for each Polygons
```



```
png(file = "plots/corrgram_registered_voters.png", width = 3000 * 2^0.5, height = 3000,
    res = 300)
corrgram(results[, names(results) %>%
    str_subset("(reg)") %>%
    sort][, c(6:length(names(results) %>%
    str_subset("(reg)")), 1:5)], order = FALSE, upper.panel = panel.cor, text.panel = panel.txt,
    diag.panel = panel.minmax)
dev.off()
```

```
## pdf
##   2
```

```
png(file = "plots/corrgram_CDU.png", width = 3000 * 2^0.5, height = 3000, res = 300)
corrgram(results[, names(results) %>%
    str_subset("(CDU)") %>%
    sort][, c(6:length(names(results) %>%
    str_subset("(reg)")), 1:5)], order = FALSE, upper.panel = panel.cor, text.panel = panel.txt,
    diag.panel = panel.minmax)
dev.off()
```

```
## pdf
##   2
```

6 Sources

6.1 Shapefiles

1953-2009 MPIDR [Max Planck Institute for Demographic Research] and CGG [Chair for Geodesy and Geoinformatics, University of Rostock] 2011: MPIDR Population History GIS Collection (partly based on

Bundesamt für Kartographie und Geodäsie 2011) – Rostock.

Bundesamt für Kartographie und Geodäsie 2011: VG 2500 Verwaltungsgebiete 1:2 500 000. Stand 01.01.2009 – Frankfurt am Main.

2013 GeoBasis-DE / Bundesamt für Kartographie und Geodäsie (BKG) 2013: Verwaltungsgebiete 1 : 250 000 VG250. https://hub.arcgis.com/datasets/esri-de-content::kreisgrenzen-2013?geometry=-20.088%2C46.270%2C40.996%2C55.886

2017 GeoBasis-DE / Bundesamt für Kartographie und Geodäsie (BKG) 2017: Verwaltungsgebiete 1 : 250 000 VG250. https://hub.arcgis.com/datasets/b2e6d8854d9744ca88144d30bef06a76_1?geometry=-20.088%2C46.270%2C40.996%2C55.886

2020 GeoBasis-DE / Bundesamt für Kartographie und Geodäsie (BKG) 2020: Verwaltungsgebiete 1 : 250 000 VG250. License: dl-de/by-2-0 (www.govdata.de/dl-de/by-2-0) https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/verwaltungsgebiete/verwaltungsgebiete-1-250-000-ebenen-stand-01-01-vg250-ebenen-01-01.html

6.2 Election results

6.2.1 County level

1953 Der Bundeswahlleiter 1953: Bundestagswahl 1953, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/9d55fc40-22a5-43f2-9b50-0bd3ff3780fd/btw1953kreis.csv

1957 Der Bundeswahlleiter 1957: Bundestagswahl 1957, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/4c84aa1b-4b75-43c5-bfc7-da240208451b/btw1957kreis.csv

1961 Der Bundeswahlleiter 1961: Bundestagswahl 1961, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/d3927d97-7c88-4d9a-a08d-97f68a8ce9af/btw1961kreis.csv

1965 Der Bundeswahlleiter 1965: Bundestagswahl 1965, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/6b2137a1-01e1-481e-a221-e4de19c588bd/btw1965kreis.csv

1969 Der Bundeswahlleiter 1969: Bundestagswahl 1969, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/c5819a55-5a24-4f09-9845-d8f40da89208/btw1969kreis.csv

1972 Der Bundeswahlleiter 1972: Bundestagswahl 1972, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/3a7bbc36-c3ff-4a07-aa7c-aec1f3aa05d6/btw1972kreis.csv

1976 Der Bundeswahlleiter 1976: Bundestagswahl 1976, Ergebnisse nach kreisfreien Städten und Landkreisen. https://www.bundeswahlleiter.de/dam/jcr/91e7563c-5149-42c6-8c48-e88390962cdc/btw1976kreis.csv

6.2.2 Municipality level

1980 Der Bundeswahlleiter 1980: Bundestagswahl 1980, Ergebnisse nach Gemeinden https://www.bundeswahlleiter.de/dam/jcr/bd5c49db-8d83-463b-bab4-630b418b99cf/btw80_wbz.zip

1983 Der Bundeswahlleiter 1983: Bundestagswahl 1983, Ergebnisse nach Wahlbezirken und Gemeinden. https://www.bundeswahlleiter.de/dam/jcr/8b384f87-bad0-4b99-a7a9-88af7f5f01d3/btw83_wbz.zip

1987 Der Bundeswahlleiter 1987: Bundestagswahl 1987, Ergebnisse nach Wahlbezirken. https://www.bundeswahlleiter.de/dam/jcr/7590fb53-3965-438b-a339-d63878bf0e64/btw87_wbz.zip

1990 Der Bundeswahlleiter 1990: Bundestagswahl 1990, Ergebnisse nach Wahlbezirken und Gemeinden. https://www.bundeswahlleiter.de/dam/jcr/f57480e9-f383-4782-9488-fae3d206a84c/btw90_wbz.zip

1994 Der Bundeswahlleiter 1994: Bundestagswahl 1994, Ergebnisse nach Wahlbezirken und Gemeinden. https://www.bundeswahlleiter.de/dam/jcr/538a1f46-4407-49b4-995b-02bd9e3a8399/btw94_wbz.zip

1998 Der Bundeswahlleiter 1998: Bundestagswahl 1998, Ergebnisse nach Wahlbezirken und Gemeinden. https://www.bundeswahlleiter.de/dam/jcr/e2063bc8-4827-4826-a2db-a6cae5d9cf15/btw98_wbz.zip

2002 Der Bundeswahlleiter 2002: Bundestagswahl 2002, Ergebnisse nach Wahlbezirken. https://www.bundeswahlleiter.de/dam/jcr/16a4f1b0-fbd6-4c79-b5a6-b2792ba6df72/btw02_wbz.zip

2005 Der Bundeswahlleiter 2005: Bundestagswahl 2005, Ergebnisse nach Wahlbezirken. https://www.bundeswahlleiter.de/dam/jcr/3a8aa53f-53a1-4ad9-938f-0aa52c18863d/btw05_wbz.zip

2009 Der Bundeswahlleiter 2009: Bundestagswahl 2009, Ergebnisse nach Wahlbezirken. https://www.bundeswahlleiter.de/dam/jcr/159344fc-d466-4d34-b432-25d7891ab09f/btw09_wbz.zip

2013 Der Bundeswahlleiter 2013: Bundestagswahl 2013, Ergebnisse nach Wahlbezirken. https://www.bundeswahlleiter.de/dam/jcr/0ad35576-0c4b-4fa5-85f5-284618b8fa25/btw13_wbz.zip

2017 Der Bundeswahlleiter 2017: Bundestagswahl 2017, Ergebnisse nach Wahlbezirken. https://www.bundeswahlleiter.de/dam/jcr/a2eef6bd-0225-447c-9943-7af0f46c94d1/btw17_wbz.zip

6.3 For details on redistricting see:

before 1990 https://de.wikipedia.org/wiki/Kreisreformen_in_der_Bundesrepublik_Deutschland_bis_1990

after 1990 https://de.wikipedia.org/wiki/Kreisreformen_in_Deutschland_nach_1990