Visualisation of the wind energy power plant repowering potential in Rhineland Palatinate with R using Leaflet and Shiny

Kelly Lasisi, Anja Weiler, Elias Cuadra

10 4 2021

Contents

1. Introduction

The project "Municipal greenhouse gas accounting and regional climate protection portals in Rhineland-Palatinate (KomBiReK)", funded by the European Regional Development Fund (EFRE) and the State of Rhineland-Palatinate supports the creation of municipal climate protection measures in order to achieve the climate protection goals of the municipalities and the state and thereby increase regional added value, ensure sustainability and thus improve the quality of life of all citizens. When developing municipal climate protection, a sound strategy is required with regard to the legally anchored striving for climate neutrality in the state of Rhineland-Palatinate (LKSG paragraph 4, 2014). The energy transition is a cornerstone of this strategy and Rhineland-Palatinate wants to play a pioneering role in the implementation of the energy transition. With this in mind, the state government publishes on its website that Rhineland-Palatinate will cover 100% of its electricity needs from renewable energies by 2030. In addition to energy from the sun, water and biomass, two thirds of the electricity generated in 2030 should come from wind power and is therefore the subject of the following considerations (state government Rhineland-Palatinate, 2021).

There are two ways of increasing the amount of electricity generated by wind energy. On the one hand, areas that are still available can be identified and built on with new wind energy power plants (WEPP). On the other hand, existing old WEPP with low electricity fed in, can be replaced by new and more efficient systems through what is known as "repowering". To identify, compare and visualise the old WEPP in Rhineland-Palatinate that could be potentially suitable for a repowering is the task of this work. To accomplish this goal the necessary data needs to be prepared followed by a visualisation process with Leaflet and finally an interactive Shiny application will allow to select different date ranges and search and find WEPP and their related technical information and the repowering potential that could be achieved.

2. Software and Data

For the data wrangling and processing as well as the creation of the application R version 4.0.3 (2020-10-10) – "Bunny-Wunnies Freak Out" and RStudio Version 1.3.1093 was used.

2.1 Data from the transmission system operator Amprion and the market master data register (MaStR) The data originated partly from a dataset of the transmission network operator Amprion from 2019 (https://www.amprion.net/), responsible for Rhineland-Palatinate. These consist of system master data including energy source, system type, performance, date of commissioning, system key and movement data including amount of electricity fed in and remuneration of each wind energy power plant in Rhineland-Palatinate. The second source is an abstract, taken in October 2020, of the market master data register (Marktstammdatenregister (MaStR), https://www.marktstammdatenregister.de/MaStR) maintained by the Federal Network Agency since the beginning of 2017. In general, all active systems connected to the grid for generating electricity or gas plus the market players involved must be registered here. Important information on the place, the municipality, technical information and, above all, the EEG (Renewable Energy Act) system key in combination with the geo-reference of all registered WEPP in Rhineland-Palatinate can be viewed in the MaStR. Since the dataset of Amprion does not have certain variables which the abstract of the MaStR has, these two datasets were compiled, processed and merged together by the EEG system key by the Energy Agency of Rhineland-Palatinate. The processing included a calculation of the full load hours, electricity yield in 2019, electricity yield after repowering, the increase of electricity yield in percent through repowering, emission reduction given the emission factor of the federal electricity mix in 2019 and then an addition to the previously created data table. Thus, the used data was provided as a csv file with the name msb19_new.csv including the described information of 1750 WEPPs out of which 416 WEPP had incorrect or missing geographical references.

2.2 Data preparation To create a geographical reference for all WEPP the geocode_OSM function was uses to find the WEPP according to their address. Because there was a lot of data wrangling and iterative processing, the extensive code is not displayed in this report but for the purpose of reproducibility it is to be found in the original rmarkdown file (repowering_report.Rmd) also the coding blocks can be seen **here**. After filtering for the WEPP that were built before 2006, seven WEPP still remained without geographical reference. This generated data frame was then saved as msb19_before2005_geocoded.csv and was the basis for the further usage in the Leaflet map and the Shiny application.

2.3 Creation of a Leaflet map of WEPP in Rhineland-Palatinate built before 2006 In order to build an interactive application a Leaflet map with the described dataset is created first. The visualisation consists of the OpenStreetMap maptiles, a marker for every WEPP with cluster options and the borders of the state county and municipalities which can be switched on and off. The code to generate the map as well as a link to the interactive map is provided below.

To the repowering map

```
################################################################################
#load packages and import data                                                 #
################################################################################
Sys.setenv(LANG = "en")
pacman::p_load(rio, data.table, tidyverse, tidyr, purrr, magrittr, compare,
               ggplot2, leaflet, sp, raster, rgdal, ggmap, htmltools,
               htmlwidgets, tmaptools)
msb19_before2005_geocoded <- read.csv("data//msb19_before2005_geocoded.csv",
                                      header=TRUE)
################################################################################
#change leistung_s to numeric and indatum_s to date
```

```
###############################################################################
msb19_before2005_geocoded$leistung_s <- as.numeric(msb19_before2005_geocoded$
                                                     leistung_s)
msb19_before2005_geocoded$indatum_s <- as.Date(msb19_before2005_geocoded$
                                                 indatum_s, "%Y-%m-%d")
###############################################################################
#load borders of Rhineland-Palatinate as shape files                          #
###############################################################################
land <- readOGR("Grenzen/Landesgrenze_RLP.shp")
landkreise <- readOGR("Grenzen/Landkreise_RLP.shp")
gemeinden <- readOGR("Grenzen/Verbandsgemeinde_RLP.shp")
###############################################################################
#create map                                                                   #
###############################################################################
m <- leaflet(data = msb19_before2005_geocoded) %>%
  addTiles() %>%

  addPolygons(data = land,
              color = "#5DADE2",
              weight = 2,
              opacity = 0.6,
              fillColor = "#5DADE200",
              highlight = highlightOptions(weight = 7,
                                           color = "#5DADE2",
                                           fillColor = "#5DADE2",
                                           fillOpacity = 0.3,
                                           bringToFront = TRUE),
              label = "Rheinland-Pfalz",
              group = "Rheinland-Pfalz") %>%

  addPolygons(data = landkreise,
              color = "#000fff",
              weight = 2,
              opacity = 0.6,
              fillColor = "#000fff00",
              highlight = highlightOptions(weight = 7,
                                           color = "#000fff",
                                           fillColor = "#000fff",
                                           fillOpacity = 0.3,
                                           bringToFront = TRUE),
              label = landkreise$ldkreis,
              group = "Landkreise") %>%

  addPolygons(data = gemeinden,
              color = "#D93F0D",
              weight = 1,
              opacity = 0.6,
              fillColor = "#D93F0D00",
              highlight = highlightOptions(weight = 7,
                                           color = "#D93F0D",
                                           fillColor = "#D93F0D",
                                           fillOpacity = 0.3,
                                           bringToFront = TRUE),
              label = gemeinden$vgname,
              group = "Verbandsgemeinden") %>%
  addLayersControl(overlayGroups = c("Rheinland-Pfalz", "Landkreise",
                                     "Verbandsgemeinden"),
                   options = layersControlOptions(collapsed = FALSE)) %>%
  addMarkers(lng = msb19_before2005_geocoded$lon_wgs84,
             lat = msb19_before2005_geocoded$lat_wgs84,
             clusterOptions = markerClusterOptions(disableClusteringAtZoom = 10),
             popup = ~paste("<h3> Daten der Windkraftanlage</h3>",
                            "<b>Landkreis (LK):</b>",lk_name, "<br>",
                            "<b>Verbandsgemeinde:</b>", vg_name, "<br>",
                            "<b>EEG-Nr.:</b>", eeg_nr,"<br>",
                            "<b>Leistung [kW]:</b>", leistung_s, "<br>",
                            "<b>Nabenhoehe [m]:</b>", nabe, "<br>",
                            "<b>Rotordurchmesser [m]:</b>", rotor, "<br>",
                            "<b>Stromertrag 2019 [MWh]:</b>", Ertrag2019_MWh, "<br>",
                            "<b>Volllaststunden im LK 2019 [h]:</b>",
                            round(menge_kwh/leistung_s), "<br>",
                            "<b style ='color: red'>Prognose Volllaststunden [h]:</b>",
                            lk_volllast, "<br>",
                            "<b style ='color: red'>Prognose Stromertrag nach
                            <br>Repowering [MWh]:</b>", ErtragRepowert, "<br>",
                            "<b style ='color: red'>Prognose Ertragssteigerung
                            [%]:</b>", Ertragssteigerung, "<br>",
                            "<b style ='color: red'>Emissionsminderung nach Repowering
                            bei Bundesstrommix 2019 [t/a]:</b>",
                            round(Emissionsminderung), "<br>"
             ),
             label = ~as.character(vg_name))
################################################################################
#print map                                                                     #
################################################################################
m
################################################################################
#save map as html file                                                         #
################################################################################
saveWidget(m, file="repowering_map.html")
```

3. Shiny application of the WEPP in Rhineland-Palatinate that are built before 2006

With the interactive Leaflet map and the data about the WEPP built before 2006 a Shiny application that has the ability to filter the data for a date range between 1991 and 2005 and shows the respective output in the map and as a data table. The code to construct this application is shown below. **To run the application please open the file with the name repowering_app.R and click the run button.**

```
# import and formatting
msb19_before2005 <- read.csv("data//msb19_before2005_geocoded.csv", header=TRUE)
msb19_before2005 <- msb19_before2005[,-1]
msb19_before2005$indatum_s <- as.Date(msb19_before2005$indatum_s, "%Y-%m-%d")
msb19_before2005 <- msb19_before2005 %>%
    mutate(year = as.numeric(format(msb19_before2005$indatum_s, "%Y")))

# load shapes
land <- readOGR("Grenzen/Landesgrenze_RLP.shp")
landkreise <- readOGR("Grenzen/Landkreise_RLP.shp")
gemeinden <- readOGR("Grenzen/Verbandsgemeinde_RLP.shp")


# Define UI for application that draws a histogram
ui <- dashboardPage(
    skin = "red",
    dashboardHeader(title = "Repowerin in RLP"),
    dashboardSidebar(
        sliderInput(inputId = "daterange",
                    label = "Date Range",
                    min = min(msb19_before2005$year),
                    max = max(msb19_before2005$year),
                    value = c(min(msb19_before2005$year), max(msb19_before2005$year)),
                    sep = "",
                    step = 1
        )
    ),

    dashboardBody(
        fluidRow(box(width = 12, leafletOutput(outputId = "mymap"))),
        fluidRow(box(width = 12, dataTableOutput(outputId = "subdata")))
    )

)
# Define server logic required to draw a histogram
server <- function(input, output) {

    data_input <- reactive({

        msb19_before2005 %>%
            filter(year >= input$daterange[1]) %>%
            filter(year <= input$daterange[2])
    })

    output$mymap <- renderLeaflet(
        leaflet(data = data_input()) %>%
            addTiles() %>%

            addPolygons(data = land,
                        color = "#5DADE2",
                        weight = 2,
                        opacity = 0.6,
                        fillColor = "#5DADE200",
                        highlight = highlightOptions(weight = 7,
                                                     color = "#5DADE2",
                                                     fillColor = "#5DADE2",
                                                     fillOpacity = 0.3,
                                                     bringToFront = TRUE),
                        label = "Rheinland-Pfalz",
                        group = "Rheinland-Pfalz") %>%

            addPolygons(data = landkreise,
                        color = "#000fff",
                        weight = 2,
                        opacity = 0.6,
                        fillColor = "#000fff00",
                        highlight = highlightOptions(weight = 7,
                                                     color = "#000fff",
                                                     fillColor = "#000fff",
                                                     fillOpacity = 0.3,
                                                     bringToFront = TRUE),
                        label = landkreise$ldkreis,
                        group = "Landkreise") %>%

            addPolygons(data = gemeinden,
                        color = "#D93F0D",
                        weight = 1,
                        opacity = 0.6,
                        fillColor = "#D93F0D00",
                        highlight = highlightOptions(weight = 7,
                                                     color = "#D93F0D",
                                                     fillColor = "#D93F0D",
                                                     fillOpacity = 0.3,
                                                     bringToFront = TRUE),
                        label = gemeinden$vgname,
                        group = "Verbandsgemeinden") %>%

            addLayersControl(overlayGroups = c("Rheinland-Pfalz", "Landkreise", "Verbandsgemeinden"),
                             options = layersControlOptions(collapsed = FALSE)) %>%

            addMarkers(lng = data_input() %>% pull(lon_wgs84),
                       lat = data_input() %>% pull(lat_wgs84),
                       clusterOptions = markerClusterOptions(disableClusteringAtZoom =
                                                               10),
                       popup = ~paste("<h3> Daten der Windkraftanlage</h3>",
                                      "<b>Landkreis (LK):</b>", lk_name, "<br>",
                                      "<b>Verbandsgemeinde:</b>", vg_name, "<br>",
                                      "<b>EEG-Nr.:</b>", eeg_nr,"<br>",
                                      "<b>Leistung [kW]:</b>", leistung_s, "<br>",
                                      "<b>Nabenhoehe [m]:</b>", nabe, "<br>",
                                      "<b>Rotordurchmesser [m]:</b>", rotor, "<br>",
                                      "<b>Stromertrag 2019 [MWh]:</b>", Ertrag2019_MWh,
                                      "<br>",
                                      "<b>Volllaststunden im LK 2019 [h]:</b>",
                                      round(menge_kwh/leistung_s), "<br>",
                                      "<b style ='color: red'>Prognose Volllaststunden
                                      [h]:</b>", lk_volllast, "<br>",
                                      "<b style ='color: red'>Prognose Stromertrag nach
                                      <br>Repowering [MWh]:</b>", ErtragRepowert,
                                      "<br>",
                                      "<b style ='color: red'>Prognose
                                      Ertragssteigerung [%]:</b>", Ertragssteigerung,
                                      "<br>",
                                      "<b style ='color: red'>Emissionsminderung nach
                                      Repowering bei Bundesstrommix 2019 [t/a]:</b>",
                                      round(Emissionsminderung), "<br>"),
                       label = ~as.character(vg_name)
                       )
    )

    output$subdata <- renderDataTable(data_input(),
                                      options = list(scrollX = TRUE))


}


# Run the application
shinyApp(ui = ui, server = server)
```

3.1 Code

3.2 Result An interactive application that shows the WEPP in Rhineland-Palatinate built before 2006 could be created. It has the feature to selct the date range of the comsissioning date of the WEPP and shows technical and other information regarding their location ect. in a data table. Also the interactive Leaflet map gives the oportunity to find the location of the WEPP easily and see the repowering potential of each WEPP if repowered as well as the emission reduction potential. In terms of climate protection and the energy transistion this can be a helpful tool to gather information about the most efficient way to implement a repowering strategy. The appearance of the application is shown in the images below.

4. Sources

Chris Beely and Shitalkumar R. Sukhdeve (2018): Web Application Development with R using Shiny - Third edition, Published by Packt Publishing Ltd., https://www.packtpub.com

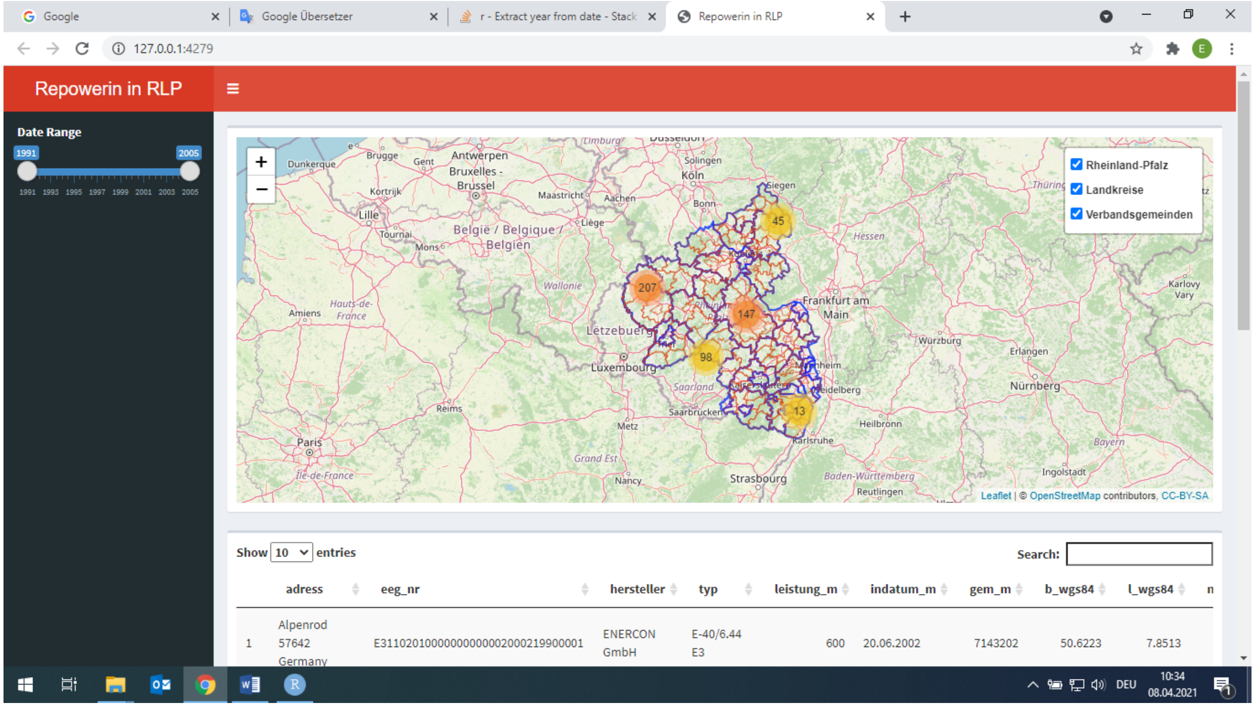
Yihui X. et al. (2021): R Markdown Cookbook, CRC Press - Taylor and Francis Group https://www.crcpress.com

Yihui X. et al. (2019): R Markdown - The Definitve Guide, CRC Press - Taylor and Francis Group https://www.crcpress.com

Landesregierung RLP (2021): Energiewende - Wind, Sonne Wasser, https://www.rlp.de/de/regierung/schwerpunkte/energiewende/#:~:text=Rheinland%2DPfalz%20wird%20bis%20zum,einem%20Viertel%20entfallen., opened on the 10.03.2021

LKSG (2014): Landesgesetz zur Foerderung des Klimaschutzes vom 19. August 2014, http://landesrecht.rlp.de/jportal/portal/t/onc/page/bsrlpprod.psml?pid=Dokumentanzeige&showdoccase=1&js_peid=Trefferliste&documentnumber=1&numberofresults=22&fromdoctodoc=yes&doc.id=jlr-KlimaSchGRPrahmen&doc.part=X&doc.price=0.0&doc.hl=1, opened on the 10.03.2021

Mac Kay JC (2009), Sustainable Energy - Without the hot air, UIT Cambridge Ltd.

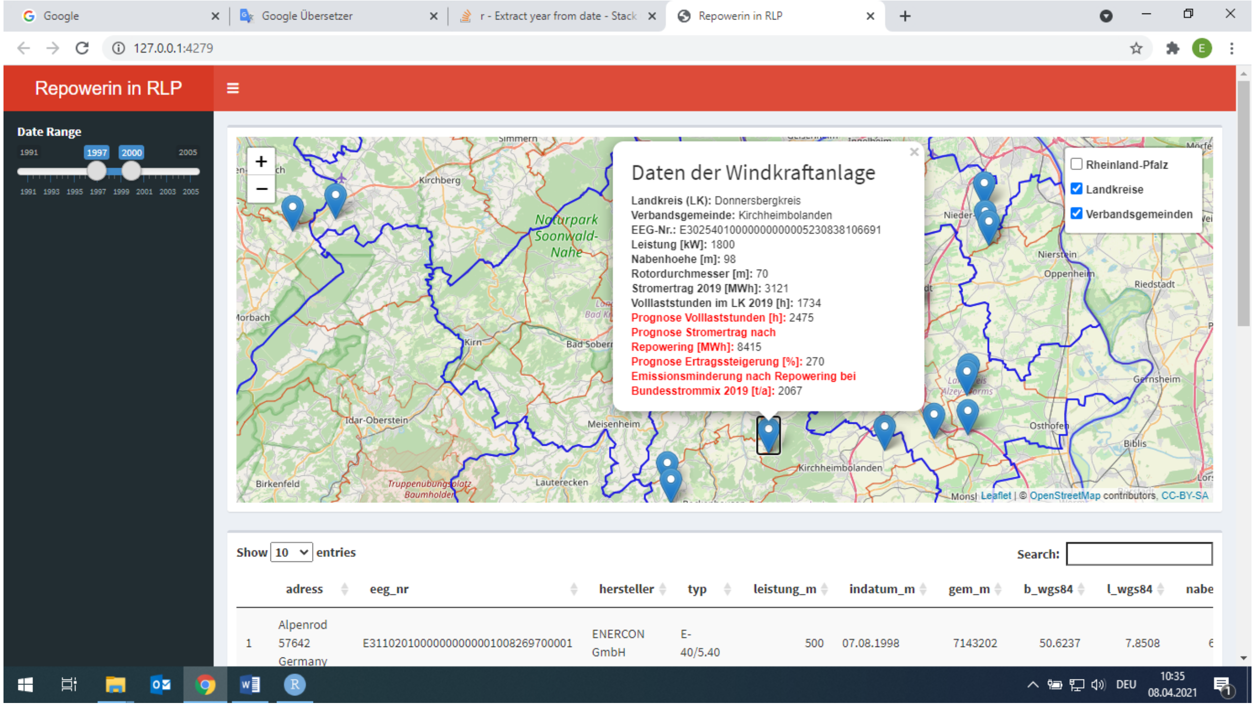


Show 10 entries

Search:

	adress	eeg_nr	hersteller	typ	leistung_m	indatum_m	gem_m	b_wgs84	l_wgs84	n
1	Alpenrod 57642 Germany	E31102010000000000002000219900001	ENERCON GmbH	E-40/6.44 E3	600	20.06.2002	7143202	50.6223	7.8513	

Figure 1: Repowering application with date range slidebar in the upper left corner, interactive Leaflet map with cluster options in the centre, data table of selected WEPP at the bottom and check box to turn county and municipality boarders on and off in the upper rigth corner



Daten der Windkraftanlage

Landkreis (LK): Donnersbergkreis
Verbandsgemeinde: Kirchheimbolanden
EEG-Nr.: E30254010000000000005230838106691
Leistung [kW]: 1800
Nabenhoehe [m]: 98
Rotordurchmesser [m]: 70
Stromertrag 2019 [MWh]: 3121
Volllaststunden im LK 2019 [h]: 1734
Prognose Volllaststunden [h]: 2475
Prognose Stromertrag nach Repowering [MWh]: 8415
Prognose Ertragssteigerung [%]: 270
Emissionsminderung nach Repowering bei Bundesstrommix 2019 [t/a]: 2067

Show 10 entries

Search:

	adress	eeg_nr	hersteller	typ	leistung_m	indatum_m	gem_m	b_wgs84	l_wgs84	nabe
1	Alpenrod 57642 Germany	E31102010000000000001008269700001	ENERCON GmbH	E-40/5.40	500	07.08.1998	7143202	50.6237	7.8508	6

Figure 2: Selected daterange from 1997 to 2000, zoom to county level and view markers for each WEPP with popup of technical information and forecasts in red letters in this case of a WEPP in Kirchheimbolanden of the Donnersbergkreis