Calculating Euclidean and along-river distance between sites

Sarah J. Hart

February 06, 2025

# Data

sites <- read\_csv(here("Data", "Jordan\_sites.csv"))   
sites.sf <- st\_as\_sf(sites, coords = c("UTM\_easting", "UTM\_northing"), crs = 32613)   
st\_write(sites.sf, here("Data", "Spatial", "sites.shp"), append=F)

## Download data from NHDPlusV2

We downloaded high resultion hydrography data from the USGS NHDPlus Version 2 using the *nhdplusTools* package ([Blodgett and Johnson 2023](#ref-nhdplusTools)).

library(nhdplusTools)  
dir.create(here("Data", "Spatial", "NHDPlusV2"))  
studyarea <- sites.sf %>% st\_buffer(100) %>% st\_union() # create buffer around sites and union to create study area  
huc06 <- get\_huc(AOI = studyarea, id = NULL, t\_srs = NULL, buffer = 0.5, type = "huc06") # this downloads all of teh data for the Rio Grande Watershed  
huc12 <- get\_huc(AOI = studyarea, id = NULL, t\_srs = NULL, buffer = 0.5, type = "huc12")  
  
download\_nhdplushr(nhd\_dir=here("Data", "Spatial", "NHDPlusV2"), hu\_list=huc06$huc6, download\_files = TRUE, archive = FALSE)  
  
nhdplus.gdbs <- list.files(here("Data", "Spatial", "NHDPlusV2", "13"), pattern=".gdb", full.names=T)  
nhdplus.flow <- lapply(nhdplus.gdbs, function(x){read\_sf(x, layer="NHDFlowline")}) %>% dplyr::bind\_rows()  
nhdplus.wbd8 <- lapply(nhdplus.gdbs, function(x){read\_sf(x, layer="WBDHU8")}) %>% dplyr::bind\_rows()   
st\_write(nhdplus.wbd8, here("Data", "Spatial", "NHDPlusV2", "NHDwbd8.shp"), append=F)  
st\_write(nhdplus.flow, here("Data", "Spatial", "NHDPlusV2", "NHDFlowline.shp"), append=F)

We the manually digitized one stream that was not included in the NHDPlusV2 dataset using QGIS.

nhdplus.wbd8.sub <- nhdplus.wbd8 %>% filter(Name=="Alamosa-Trinchera")  
streams <- st\_read(here("Data", "Spatial", "NHDPlusV2", "NHDFlowline-edit.shp")) %>% st\_crop(nhdplus.wbd8.sub) # select only streams in Alamosa-Trinchera watershed

# Calculate distance between point

## Euclidean distance

We calculated the Euclidean distance between sites using the *sf* package ([Pebesma and Bivand 2023](#ref-sf)).

sites.dist.euclidean <- st\_distance(sites.sf) %>% as.data.frame() %>% drop\_units() # calculate euclidean distance between sites in meters and drop units so that data can be saved as csv  
colnames(sites.dist.euclidean) <- sites$Site # rename columns to site names  
row.names(sites.dist.euclidean) <- sites$Site # rename rows to site names  
write.csv(sites.dist.euclidean, here("Results", "sites-euclidean.csv")) # write data to csv file   
st\_write(sites.sf, here("Data", "Spatial", "sites.shp"), append=F) # write data to shapefile for visualization

## Distance along rivers

We calculated the distance between sites along rivers using the *sfnetworks* package ([Meer et al. 2024](#ref-sfnetworks)).

streams.sub <- streams %>%   
 st\_zm() %>% # drop M dimension  
 st\_transform(st\_crs(sites.sf)) # reproject to UTM 13N  
   
streams.sub <- streams.sub %>%   
 st\_union() %>% # dissolve lines and create nodes for every end point and intersection point  
 st\_cast("LINESTRING") %>% # convert to linestring  
 as\_sfnetwork(directed = FALSE) %>% # create network object   
 st\_network\_blend(sites.sf, tolerance=100) %>% # snap sites to network  
 activate(edges) %>% # activate edges  
 st\_as\_sf() %>% # convert to sf object  
 st\_write(here("Data", "Spatial", "edges.shp"), append=F) # write edges to file for visualization  
  
river.dist <- st\_read(here("Data", "Spatial", "edges.shp")) %>%   
 st\_union() %>% # dissolve lines and create nodes for every end point and intersection point  
 st\_cast("LINESTRING") %>% # convert to linestring  
 as\_sfnetwork(directed = FALSE) %>% # create network object  
 st\_network\_cost(from=sites.sf, to=sites.sf, direction="all") # calculate distance along rivers between sites  
  
river.dist <- data.frame(river.dist, row.names=sites.sf$Site) # convert to data frame  
colnames(river.dist) <- sites.sf$Site # rename columns to site names  
write.csv(river.dist, here("Results", "sites-river.csv")) # write data to csv file

# References

Blodgett, D., and M. Johnson. 2023. [nhdplusTools: Tools for accessing and working with the NHDPlus](https://doi.org/10.5066/P97AS8JD).

Meer, L. van der, L. Abad, A. Gilardi, and R. Lovelace. 2024. [Sfnetworks: Tidy geospatial networks](https://CRAN.R-project.org/package=sfnetworks).

Pebesma, E., and R. Bivand. 2023. [Spatial data science: With applications in r](https://doi.org/10.1201/9780429459016).