Assignment 9: Spatial Analysis in R

Keanu Valibia

Contents

OVERVIEW 1
Directions 2
DATA WRANGLING 2
Set up your session 2
Read (and filter) county features into an sf dataframe and plot 2
Read in gage locations csv as a dataframe, then display the column names it contains 5
Convert the dataframe to a spatial features ("sf") dataframe 5
Plot the gage locations on top of the counties 6
Read in the gage height data and join the site location data to it. 7
Map the pattern of gage height data 8
SPATIAL ANALYSIS 9
Transform the counties and gage site datasets to UTM Zone 14N 9
Select the gages falling within a given county 10

List of Figures

1 Nebraska Counties 4
2 NWIS Gage Locations In Nebraska 6
3 NWIS Gage Locations in Nebraska by Height 8
4 Gage Locations in Nebraska (UTM) 9
5 Gage Locations in Lancaster County, Nebraska 10

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on spatial analysis.

Directions

1. Rename this file `<FirstLast>_A09_SpatialAnalysis.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
4. Work through the steps, **creating code and output** that fulfill each instruction.
5. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
6. When you have completed the assignment, **Knit** the text and code into a single HTML file.

DATA WRANGLING

Set up your session

1. Import libraries: tidyverse, sf, leaflet, here, and mapview
2. Execute the `here()` command to display the current project directory

```
# 1.

library(tidyverse)
library(sf)
library(leaflet)
library(here)
library(mapview)
library(ggplot2)

# 2.

here()
```

```
## [1] "/home/guest/R/R Projects/EDA_Spring2024"
```

```
myTheme <- theme_classic(base_size = 11) + theme(axis.text = element_text(color = "black"),
    axis.line = element_line(color = "black"),
    panel.background = element_rect(fill = "#EDE6E3"),
    panel.grid.major = element_line(color = "#36382E",
        linetype = "dotted"), plot.title = element_text(size = 15),
    axis.title.x = element_text(size = 13), axis.title.y = element_text(size = 13),
    legend.position = "right")

theme_set(myTheme)
```

Read (and filter) county features into an sf dataframe and plot

In this exercise, we will be exploring stream gage height data in Nebraska corresponding to floods occurring there in 2019. First, we will import from the US Counties shapefile we've used in lab lessons, filtering it this time for just Nebraska counties. Nebraska's state FIPS code is `31` (as North Carolina's was `37`).

3. Read the `cb_2018_us_county_20m.shp` shapefile into an sf dataframe, filtering records for Nebraska counties (State FIPS = 31)

4. Reveal the dataset's coordinate reference system
5. Plot the records as a map (using `mapview` or `ggplot`)

```
# 3. Read in Counties shapefile into an sf
# dataframe, filtering for just NE counties

nebraska.counties <- st_read(here("~/R/R Projects/EDA_Spring2024/Data/Spatial/cb_2018_us_county_20m.shp
    filter(STATEFP == 31)
```

```
## Reading layer 'cb_2018_us_county_20m' from data source
##   '/home/guest/R/R Projects/EDA_Spring2024/Data/Spatial/cb_2018_us_county_20m.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 3220 features and 9 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -179.1743 ymin: 17.91377 xmax: 179.7739 ymax: 71.35256
## Geodetic CRS:  NAD83
```

```
# 4. Reveal the CRS of the counties features

st_crs(nebraska.counties)
```

```
## Coordinate Reference System:
##   User input: NAD83
##   wkt:
## GEOGCRS["NAD83",
##     DATUM["North American Datum 1983",
##         ELLIPSOID["GRS 1980",6378137,298.257222101,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["latitude",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["longitude",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     ID["EPSG",4269]]
```

```
# 5. Plot the data

plot(nebraska.counties)
```

6. What is the EPSG code of the Counties dataset? Is this a geographic or a projected coordinate reference system? (Or, does this CRS use angular or planar coordinate units?) To what datum is this CRS associated? (Tip: look for the EPSG code on https://spatialreference.org and examine the link for "Well Known Text as HTML" or "Human-Readable OGC WKT"...)

   ANSWER: The ESPG code is 4269. This is a geographic coorddinate reference system, associated with the North American Dataum 1983.

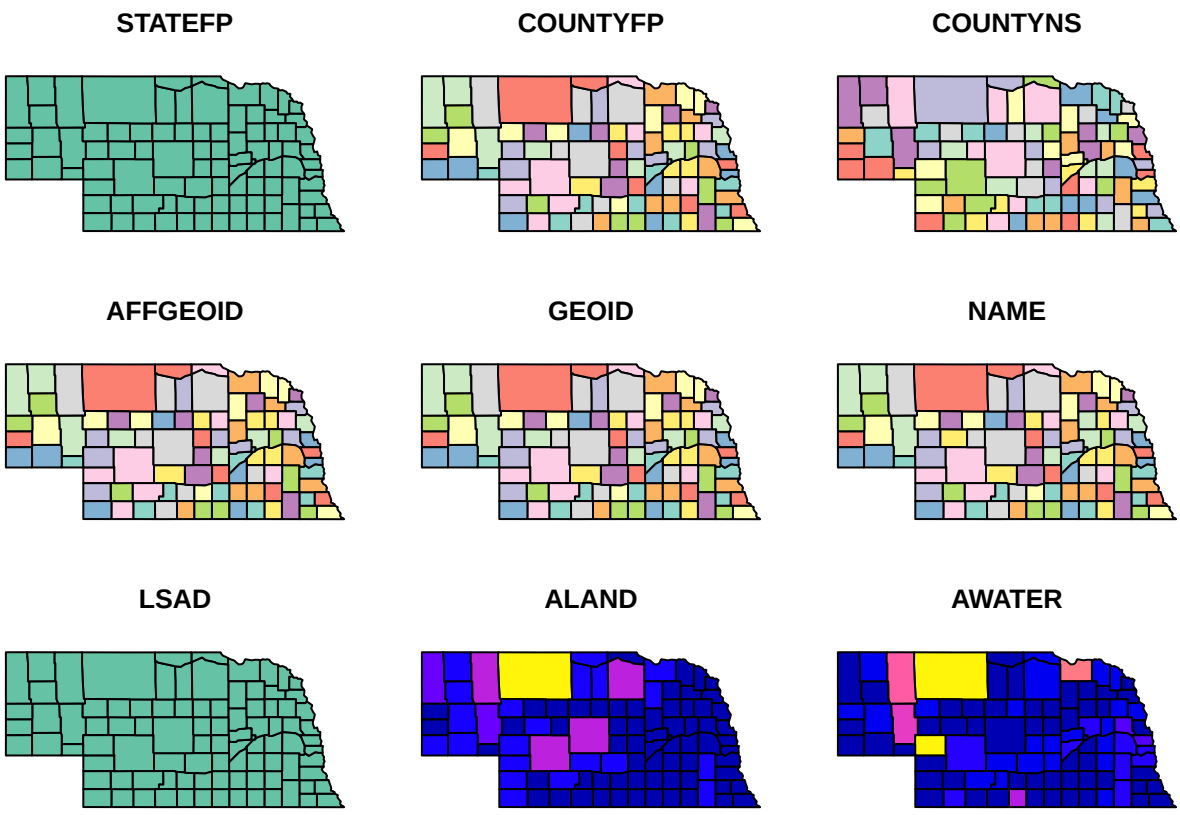

Figure 1: Nebraska Counties

Read in gage locations csv as a dataframe, then display the column names it contains

Next we'll read in some USGS/NWIS gage location data added to the `Data/Raw` folder. These are in the `NWIS_SiteInfo_NE_RAW.csv` file.(See `NWIS_SiteInfo_NE_RAW.README.txt` for more info on this dataset.)

7. Read the NWIS_SiteInfo_NE_RAW.csv file into a standard dataframe, being sure to set the `site_no` field as well as other character columns as a factor.
8. Display the column names of this dataset.

```
# 7. Read in gage locations csv as a
# dataframe

gage.locations <- read.csv(here("~/R/R Projects/EDA_Spring2024/Data/Raw/NWIS_SiteInfo_NE_RAW.csv"))
gage.locations$site_no <- as.factor(gage.locations$site_no)
gage.locations$station_nm <- as.factor(gage.locations$station_nm)
gage.locations$site_tp_cd <- as.factor(gage.locations$site_tp_cd)
gage.locations$coord_acy_cd <- as.factor(gage.locations$coord_acy_cd)
gage.locations$dec_coord_datum_cd <- as.factor(gage.locations$dec_coord_datum_cd)

# 8. Reveal the names of the columns

colnames(gage.locations)
```

```
## [1] "site_no"            "station_nm"         "site_tp_cd"
## [4] "dec_lat_va"         "dec_long_va"        "coord_acy_cd"
## [7] "dec_coord_datum_cd"
```

9. What columns in the dataset contain the x and y coordinate values, respectively?

   ANSWER:"dec_long_va" and "dec_lat_va"

Convert the dataframe to a spatial features ("sf") dataframe

10. Convert the dataframe to an sf dataframe.

- Note: These data use the same coordinate reference system as the counties dataset

11. Display the column names of the resulting sf dataframe

```
# 10. Convert to an sf object

gage.sf <- gage.locations %>%
    st_as_sf(coords = c("dec_long_va", "dec_lat_va"),
        crs = 4269)

# 11. Re-examine the column names

colnames(gage.sf)
```

```
## [1] "site_no"            "station_nm"         "site_tp_cd"
## [4] "coord_acy_cd"       "dec_coord_datum_cd" "geometry"
```

12. What new field(s) appear in the sf dataframe created? What field(s), if any, disappeared?

    ANSWER: The "geometry" field was created. The "dec_long_va" and "dec_lat_va" fields disappeared.

Plot the gage locations on top of the counties

13. Use `ggplot` to plot the county and gage location datasets.

- Be sure the datasets are displayed in different colors
- Title your plot "NWIS Gage Locations in Nebraska"
- Subtitle your plot with your name

```
# 13. Plot the gage locations atop the
# county features

gage.county.plot <- ggplot() + geom_sf(data = nebraska.counties,
    color = "#320E3B") + geom_sf(data = gage.sf,
    color = "#935435") + labs(title = "NWIS Gage Locations in Nebraska",
    subtitle = "Keanu Valibia") + xlab("Longitude") +
    ylab("Latitude")

gage.county.plot
```

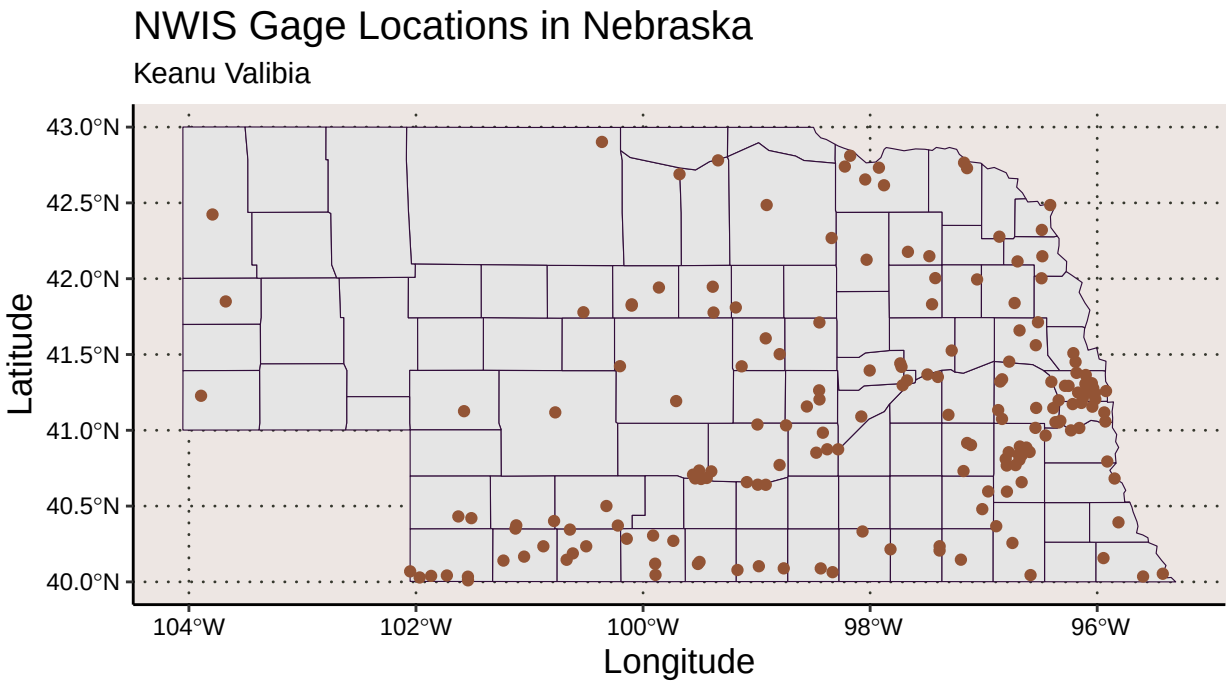

Figure 2: NWIS Gage Locations In Nebraska

Read in the gage height data and join the site location data to it.

Lastly, we want to attach some gage height data to our site locations. I've constructed a csv file listing many of the Nebraska gage sites, by station name and site number along with stream gage heights (in meters) recorded during the recent flood event. This file is titled `NWIS_SiteFlowData_NE_RAW.csv` and is found in the Data/Raw folder.

14. Read the `NWIS_SiteFlowData_NE_RAW.csv` dataset in as a dataframe

- Pay attention to which fields should be imported as factors!

15. Show the column names .
16. Join our site information (already imported above) to these gage height data

- The `site_no` and `station_nm` can both/either serve as joining attributes
- Construct this join so that the result only includes spatial features where both tables have data

17. Show the column names in this resulting spatial features object
18. Show the dimensions of the resulting joined dataframe

```
# 14. Read the site flow data into a data
# frame

siteFlow <- read.csv(here("~/R/R Projects/EDA_Spring2024/Data/Raw/NWIS_SiteFlowData_NE_RAW.csv"))
siteFlow$site_no <- as.factor(siteFlow$site_no)
siteFlow$station_nm <- as.factor(siteFlow$station_nm)

# 15. Show the column names

colnames(siteFlow)
```

```
## [1] "site_no"    "station_nm" "date"       "gage_ht"
```

```
# 16. Join the flow data to our NWIS gage
# location spatial dataframe

gage.site.join.sf <- inner_join(siteFlow, gage.sf,
    by = "site_no")

# 17. Show the column names of the joined
# dataset

colnames(gage.site.join.sf)
```

```
## [1] "site_no"            "station_nm.x"       "date"
## [4] "gage_ht"            "station_nm.y"       "site_tp_cd"
## [7] "coord_acy_cd"       "dec_coord_datum_cd" "geometry"
```

```
# 18. Show the dimensions of this joined
# dataset

dim(gage.site.join.sf)
```

```
## [1] 136   9
```

Map the pattern of gage height data

Now we can examine where the flooding appears most acute by visualizing gage heights spatially.

19. Plot the gage sites on top of counties (using `mapview`, `ggplot`, or `leaflet`)

- Show the magnitude of gage height by color, shape, other visualization technique.

```
# Map the points, sized by gage height

gage.height.plot <- ggplot() + geom_sf(data = nebraska.counties,
    color = "#320E3B") + geom_sf(data = gage.site.join.sf,
    aes(geometry = geometry, alpha = gage_ht),
    color = "#935435") + labs(title = "NWIS Gage Locations in Nebraska by Height",
    subtitle = "Keanu Valibia", alpha = "Gage Height") +
    xlab("Longitude") + ylab("Latitude")

gage.height.plot
```

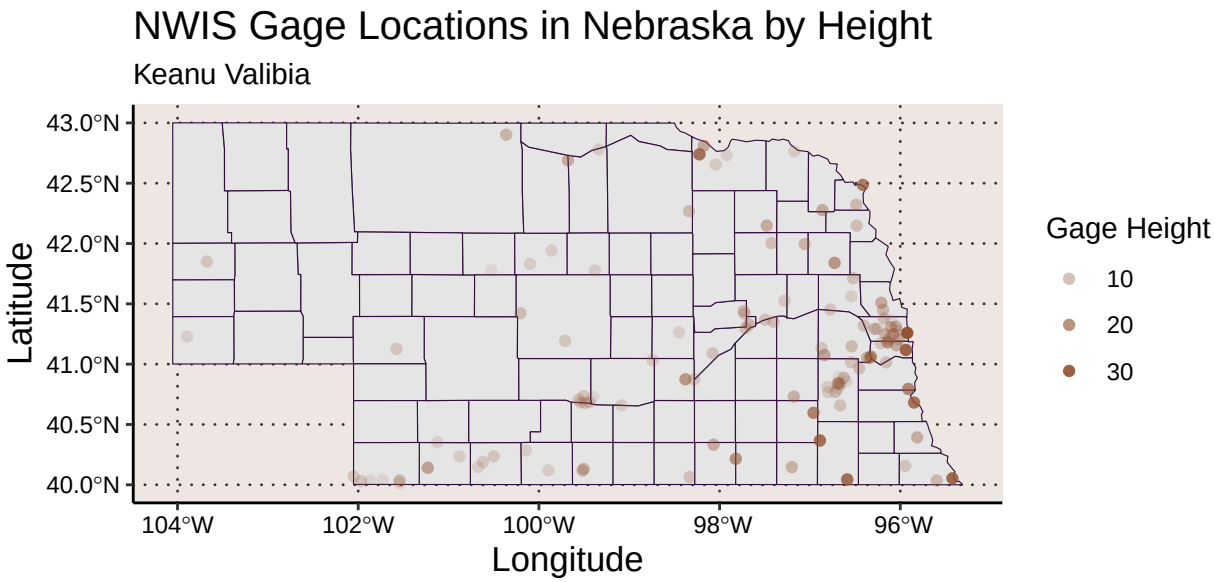

Figure 3: NWIS Gage Locations in Nebraska by Height

SPATIAL ANALYSIS

Up next we will do some spatial analysis with our data. To prepare for this, we should transform our data into a projected coordinate system. We'll choose UTM Zone 14N (EPGS = 32614).

Transform the counties and gage site datasets to UTM Zone 14N

20. Transform the counties and gage sf datasets to UTM Zone 14N (EPGS = 32614).
21. Using `mapview` or `ggplot`, plot the data so that each layer is shown with a unique color (e.g. counties blue and gages red)

```
# 20 Transform the counties and gage
# location datasets to UTM Zone 14

nebraska.utm <- st_transform(nebraska.counties,
    32614)
gage.utm <- st_transform(gage.sf, 32614)

# 21 Plot the data

ggplot() + geom_sf(data = nebraska.utm, color = "#320E3B") +
    geom_sf(data = gage.utm, color = "#935435",
        shape = "x", size = 3) + labs(title = "Gage Locations in Nebraska (UTM)",
    subtitle = "Keanu Valibia") + xlab("Longitude") +
    ylab("Latitude")
```

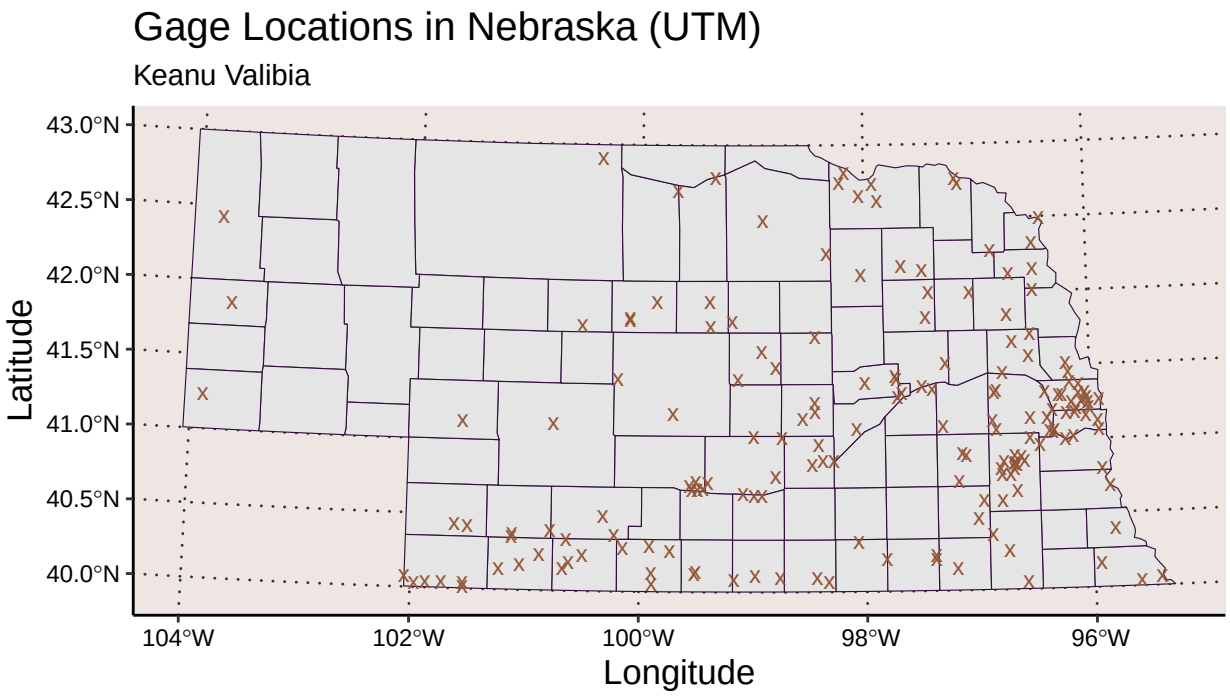

Figure 4: Gage Locations in Nebraska (UTM)

Select the gages falling within a given county

Now let's zoom into a particular county and examine the gages located there. 22. Select Lancaster county from your projected county sf dataframe 23. Select the gage sites falling `within` that county * Use either matrix subsetting or tidy filtering 24. Create a plot showing: * all Nebraska counties, * the selected county, * and the gage sites in that county

```
# 22 Select the county

nebraska.lancaster.sf <- nebraska.counties %>%
    filter(nebraska.counties$COUNTYFP == 109)

# 23 Spatially select gages within the
# selected county

gage.lancaster.sf <- st_join(gage.sf, nebraska.lancaster.sf,
    left = FALSE)

# 24 Plot

ggplot() + geom_sf(data = nebraska.utm, color = "#320E3B") +
    geom_sf(data = nebraska.lancaster.sf, color = "#320E3B",
        linewidth = 1) + geom_sf(data = gage.lancaster.sf,
    color = "#935435", shape = "x", size = 3) +
    labs(title = "Gage Locations in Lancaster County, Nebraska",
        subtitle = "Keanu Valibia") + xlab("Longitude") +
    ylab("Latitude")
```

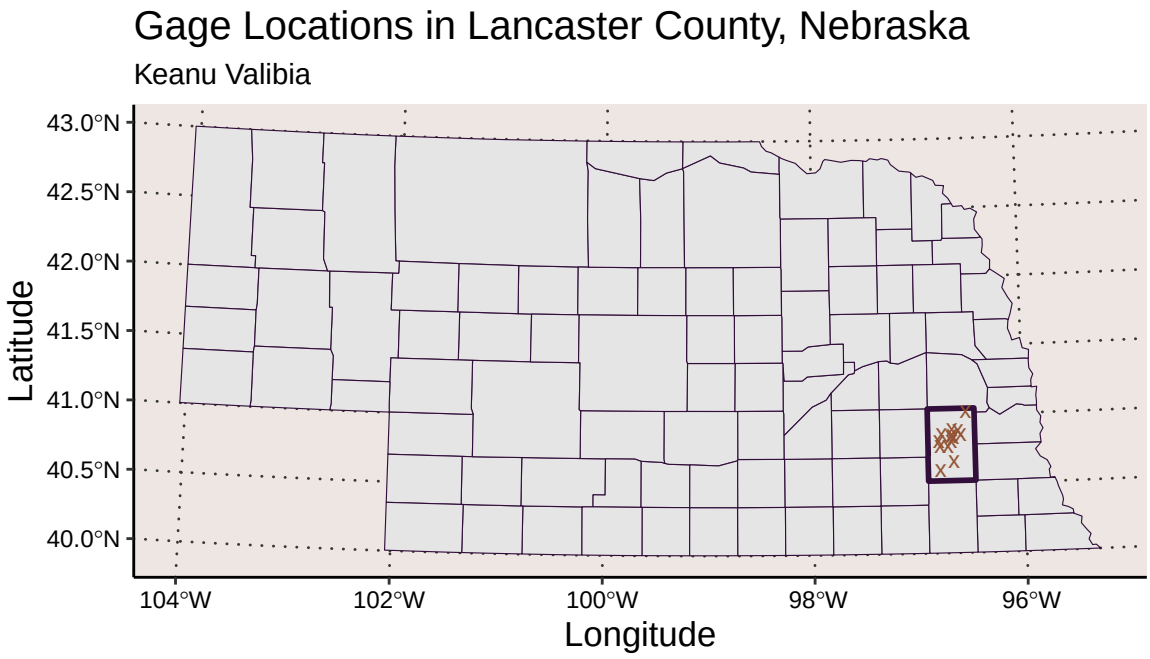

Figure 5: Gage Locations in Lancaster County, Nebraska