Exercise: New Jersey Hospital Locations

Prepared for PAP New Jersey Trainings

Kelly Pierce, Texas Advanced Computing Center

February 10, 2022

Contents

1. Download "Hospitals in NJ" from the NJGIN Open Data portal: https://njogis-newjersey.opendata.arcgis.com/datasets/hospitals-in-nj/explore?location=40.145600%2C-74.726050%2C8.85&showTable=true

2. Load and inspect the data.

```
library(sf)
```

```
## Linking to GEOS 3.8.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
WGS84 = 4326
NJ_PLANAR = 'ESRI:102311'
hosp <- st_read('data/Hospitals_in_NJ.shp')
```

```
## Reading layer `Hospitals_in_NJ' from data source
##   `/Users/kpierce/data_trainings/PAP-TACC-2022/data/Hospitals_in_NJ.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 244 features and 22 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 220776.9 ymin: 92758.93 xmax: 639050.8 ymax: 864630.7
## Projected CRS: NAD83 / New Jersey (ftUS)
```

```r
head(hosp)
```

```
## Simple feature collection with 6 features and 22 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 281276.6 ymin: 217356.6 xmax: 291225.1 ymax: 227635.6
## Projected CRS: NAD83 / New Jersey (ftUS)
##   OBJECTID                                                           NAME
## 1        1                                             Bridgeton RiteCare
## 2        2               CompleteCare Adult & Women’s Medical Professionals
## 3        3 CompleteCare Bridgeton - Medical and Dental Health Professionals
## 4        4                                     CompleteCare Health Network
## 5        5                           Kids’ Corner - Bridgeton Middle School
## 6        6                              Teen Center - Bridgeton High School
##               ADDRESS      CITY STATE   ZIP         TELEPHONE ALT_PHONE
## 1   1000 PEARL STREET BRIDGETON    NJ 08302      856-451-4700      <NA>
## 2      70 COHANSEY ST BRIDGETON    NJ 08302              <NA>      <NA>
## 3  105 MANHEIM AVENUE BRIDGETON    NJ 08302 856-451-4700X2320      <NA>
## 4  53 SOUTH LAUREL ST BRIDGETON    NJ 08302      856-451-4700      <NA>
## 5 251 W. BROAD STREET BRIDGETON    NJ 08302      856-453-1233      <NA>
## 6  111 N. WEST AVENUE BRIDGETON    NJ 08302      856-451-4440      <NA>
##                               TYPE     COUNTY COUNTYFIPS ST_FIPS COUNTRY
## 1 FEDERALLY QUALIFIED HEALTH CENTERS CUMBERLAND      34011      34      US
## 2 FEDERALLY QUALIFIED HEALTH CENTERS CUMBERLAND      34011      34      US
## 3 FEDERALLY QUALIFIED HEALTH CENTERS CUMBERLAND      34011      34      US
## 4 FEDERALLY QUALIFIED HEALTH CENTERS CUMBERLAND      34011      34      US
## 5 FEDERALLY QUALIFIED HEALTH CENTERS CUMBERLAND      34011      34      US
## 6 FEDERALLY QUALIFIED HEALTH CENTERS CUMBERLAND      34011      34      US
##   REGION LATITUDE LONGITUDE STATE_ID NAICS_CODE OWNER TTL_STAFF    BEDS TRAUMA
## 1  SOUTH 39.45620 -75.21154     <NA>       <NA>  <NA>      <NA> 9999999   <NA>
## 2  SOUTH 39.43046 -75.23435     <NA>       <NA>  <NA>      <NA> 9999999   <NA>
## 3  SOUTH 39.42937 -75.21816     <NA>       <NA>  <NA>      <NA> 9999999   <NA>
## 4  SOUTH 39.42783 -75.23519     <NA>       <NA>  <NA>      <NA> 9999999   <NA>
## 5  SOUTH 39.42937 -75.24476     <NA>       <NA>  <NA>      <NA> 9999999   <NA>
## 6  SOUTH 39.43598 -75.24656     <NA>       <NA>  <NA>      <NA> 9999999   <NA>
##                   geometry
## 1 POINT (291225.1 227635.6)
## 2 POINT (284708.3 218313.7)
## 3 POINT (289278.8 217877.2)
## 4 POINT (284463.4 217356.6)
## 5 POINT (281764.5 217940.5)
## 6 POINT (281276.6 220351.5)
```

a. What is the CRS used?

```r
st_crs(hosp)
```

```
## Coordinate Reference System:
##   User input: NAD83 / New Jersey (ftUS)
##   wkt:
## PROJCRS["NAD83 / New Jersey (ftUS)",
##     BASEGEOGCRS["NAD83",
```

```
##         DATUM["North American Datum 1983",
##             ELLIPSOID["GRS 1980",6378137,298.257222101,
##                 LENGTHUNIT["metre",1]]],
##         PRIMEM["Greenwich",0,
##             ANGLEUNIT["degree",0.0174532925199433]],
##         ID["EPSG",4269]],
##     CONVERSION["SPCS83 New Jersey zone (US Survey feet)",
##         METHOD["Transverse Mercator",
##             ID["EPSG",9807]],
##         PARAMETER["Latitude of natural origin",38.8333333333333,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8801]],
##         PARAMETER["Longitude of natural origin",-74.5,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8802]],
##         PARAMETER["Scale factor at natural origin",0.9999,
##             SCALEUNIT["unity",1],
##             ID["EPSG",8805]],
##         PARAMETER["False easting",492125,
##             LENGTHUNIT["US survey foot",0.304800609601219],
##             ID["EPSG",8806]],
##         PARAMETER["False northing",0,
##             LENGTHUNIT["US survey foot",0.304800609601219],
##             ID["EPSG",8807]]],
##     CS[Cartesian,2],
##         AXIS["easting (X)",east,
##             ORDER[1],
##             LENGTHUNIT["US survey foot",0.304800609601219]],
##         AXIS["northing (Y)",north,
##             ORDER[2],
##             LENGTHUNIT["US survey foot",0.304800609601219]],
##     USAGE[
##         SCOPE["Engineering survey, topographic mapping."],
##         AREA["United States (USA) - New Jersey - counties of Atlantic; Bergen; Burlington; Camden; Ca
##         BBOX[38.87,-75.6,41.36,-73.88]],
##     ID["EPSG",3424]]
```

b. How many columns are in the data set?

```
names(hosp)
```

```
##  [1] "OBJECTID"   "NAME"       "ADDRESS"    "CITY"       "STATE"
##  [6] "ZIP"        "TELEPHONE"  "ALT_PHONE"  "TYPE"       "COUNTY"
## [11] "COUNTYFIPS" "ST_FIPS"    "COUNTRY"    "REGION"     "LATITUDE"
## [16] "LONGITUDE"  "STATE_ID"   "NAICS_CODE" "OWNER"      "TTL_STAFF"
## [21] "BEDS"       "TRAUMA"     "geometry"
```

c. How many rows?

```
dim(hosp)
```

```
## [1] 244  23
```

d. What are the data types as understood by R? Do those data types match your understanding of the data?

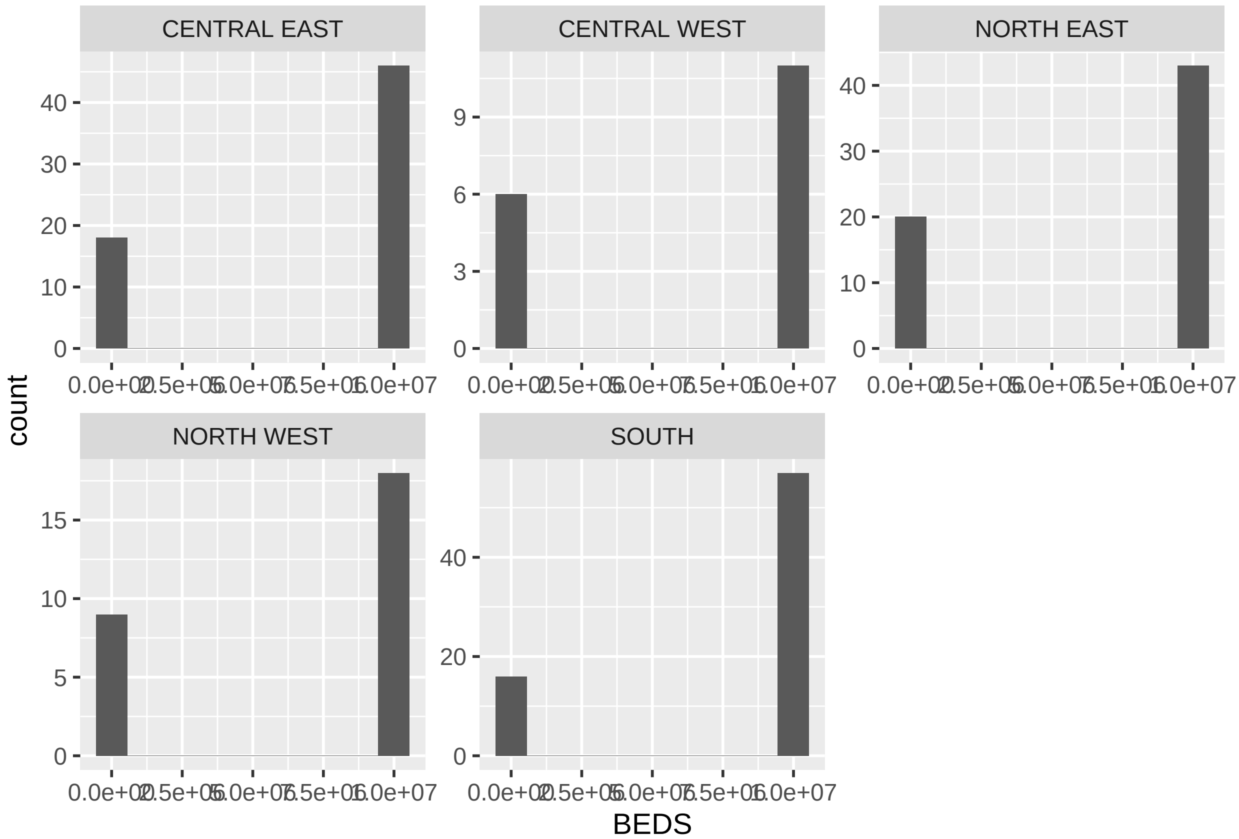
```
for(i in seq_along(names(hosp))){
  print(paste(names(hosp)[i], typeof(hosp[[i]][0]), sep=": "))
}
```

```
## [1] "OBJECTID: integer"
## [1] "NAME: character"
## [1] "ADDRESS: character"
## [1] "CITY: character"
## [1] "STATE: character"
## [1] "ZIP: character"
## [1] "TELEPHONE: character"
## [1] "ALT_PHONE: character"
## [1] "TYPE: character"
## [1] "COUNTY: character"
## [1] "COUNTYFIPS: character"
## [1] "ST_FIPS: character"
## [1] "COUNTRY: character"
## [1] "REGION: character"
## [1] "LATITUDE: double"
## [1] "LONGITUDE: double"
## [1] "STATE_ID: character"
## [1] "NAICS_CODE: character"
## [1] "OWNER: character"
## [1] "TTL_STAFF: character"
## [1] "BEDS: integer"
## [1] "TRAUMA: character"
## [1] "geometry: list"
```

3. Make a faceted histogram of beds by region. Adjust the histogram bin size as necessary.

Upon closer inspection of the data, the `BEDS` column appears to use '9999999" to indicate missing data (it's impossible for a hospital to have almost 10 million beds). This choice was probably made to keep the column data type numeric, but we'll need to address it to make sensible plots. Leaving this high value in the dataset obscures the true distribution of hospital beds:

```
hosp_hist <- ggplot(data=hosp, aes(x=BEDS)) +
  geom_histogram(bins=10) +
  facet_wrap(~REGION, scales="free")
hosp_hist
```



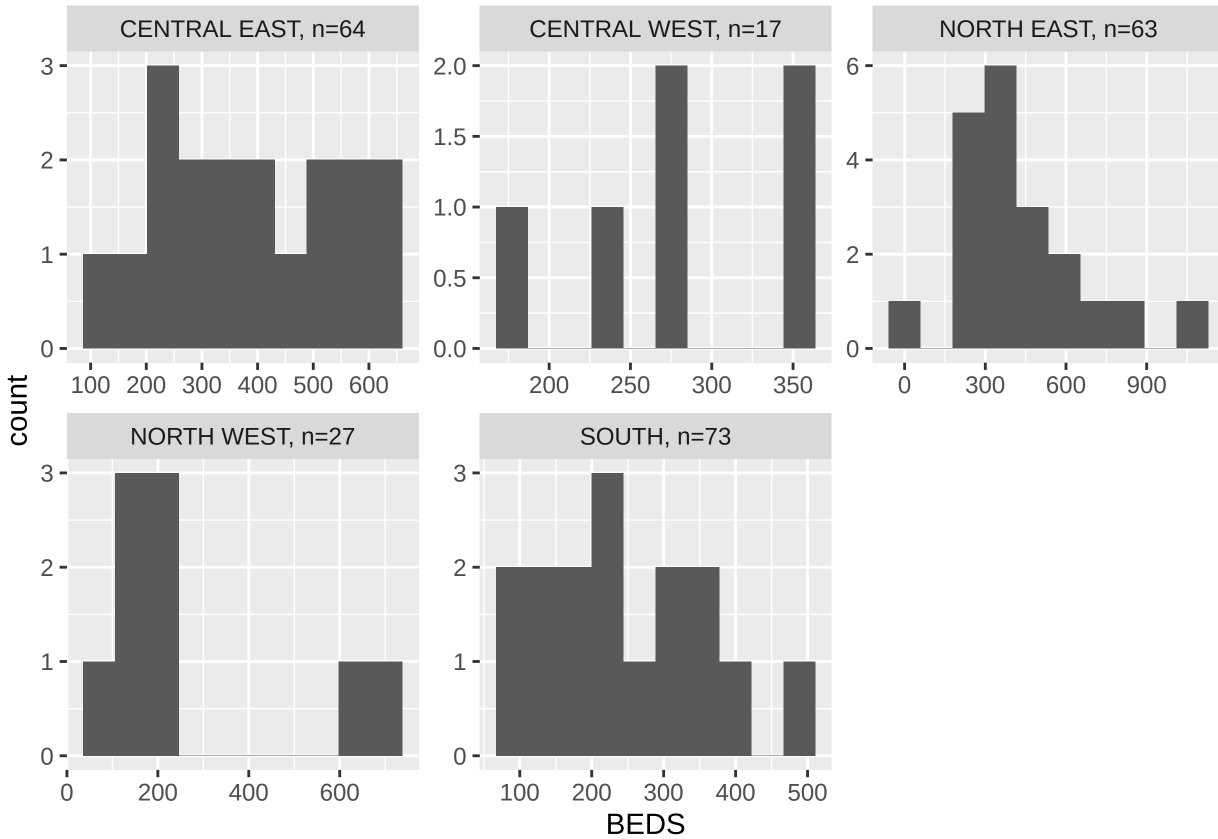
We can replace missing data using the `na_if()` function within a `mutate()` statement:

```
hosp <- hosp %>% mutate(BEDS = na_if(BEDS, 9999999))
```

Now that the missing data have been represented as `NA`, we can make more sensible plots. To help with interpretability, we're also going to add a count of the number of hospitals in each region. The pattern `groupby(variable) %>% mutate(N=n())` gives us a count of the values in each grouping. We group by `REGION`, and then create a new column with the region name and sample size together as a string. Faceting with this new column will automatically put the region + sample size information in the title of each facet plot.

```
hosp <- hosp %>%
  group_by(REGION) %>%
  mutate(N=n()) %>%
  mutate(REGION_N=paste(REGION, ", n=", N, sep=''))
hosp_hist <- ggplot(data=hosp, aes(x=BEDS), label=N) +
  geom_histogram(bins=10) +
  facet_wrap(~REGION_N, scales="free")
hosp_hist
```

```
## Warning: Removed 175 rows containing non-finite values (stat_bin).
```



4. Assign each hospital to a grid cell in the 5-mile grid.

Boilerplate to create 5-mile fishnet grid:

```
nj <- read_sf(
  'https://opendata.arcgis.com/datasets/5f45e1ece6e14ef5866974a7b57d3b95_1.geojson'
  )
nj <- nj %>% st_transform(crs=WGS84)
nj_flat <- st_transform(nj, crs=NJ_PLANAR)
fishnet_width <- 1609 * 5
nj_border <- st_make_valid(st_union(nj_flat))
net <- st_make_grid(
  x=nj_border, cellsize=fishnet_width, what='polygons', square=TRUE, crs=NJ_PLANAR
  )
net_agg <- st_as_sf(net) %>% tibble::rowid_to_column(., "net_id")
net_intersect <- st_intersects(nj_border, net_agg)
fishnet <- net_agg[unique(unlist(net_intersect)),]
```

Point-in-polygon join and aggregation:

```
hosp_net <- st_join(
  st_transform(hosp, NJ_PLANAR),
  fishnet,
  join=st_within
  ) %>%
```

```
    st_drop_geometry()

hosp_net_summary <- hosp_net %>%
  group_by(net_id) %>%
  summarise(count=n())

hosp_net_final <- st_as_sf(
  left_join(
    fishnet,
    hosp_net_summary,
    by='net_id'
    ),
  crs=st_crs(fishnet))
```

a. Plot the corresponding choropleth map of hospitals per grid cell.

Boilerplate for basemap:

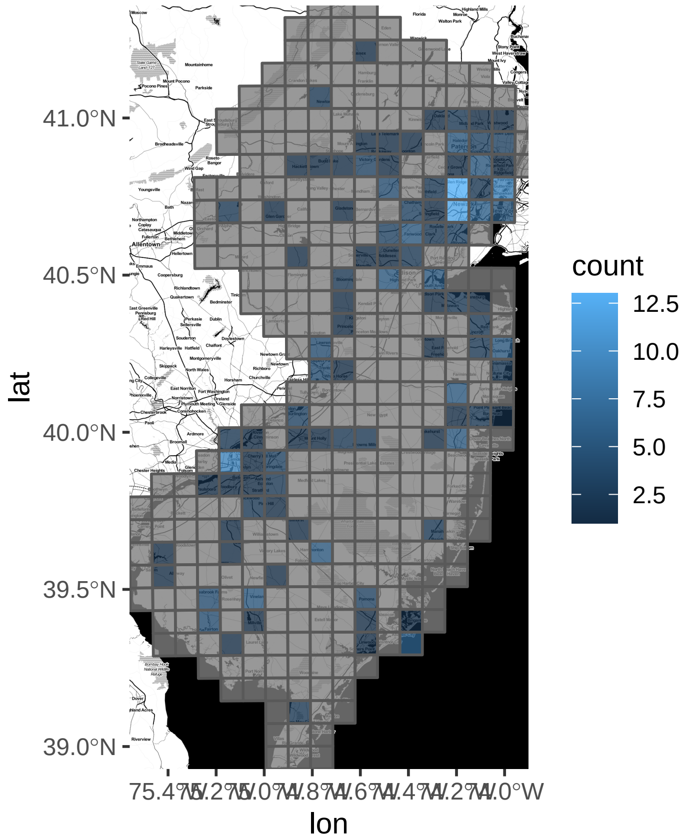
```
library(ggmap)
nj_bbox <- unname(st_bbox(nj))
nj_base_map <- get_stamenmap(bbox=nj_bbox, maptype="toner", force=TRUE)
```

Choropleth map:

```
hosp_net_map <- ggmap(nj_base_map) +
  geom_sf(
    data=st_transform(hosp_net_final, crs=WGS84),
    aes(fill=count),
    inherit.aes = FALSE, alpha=0.8
    )
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

```
hosp_net_map
```



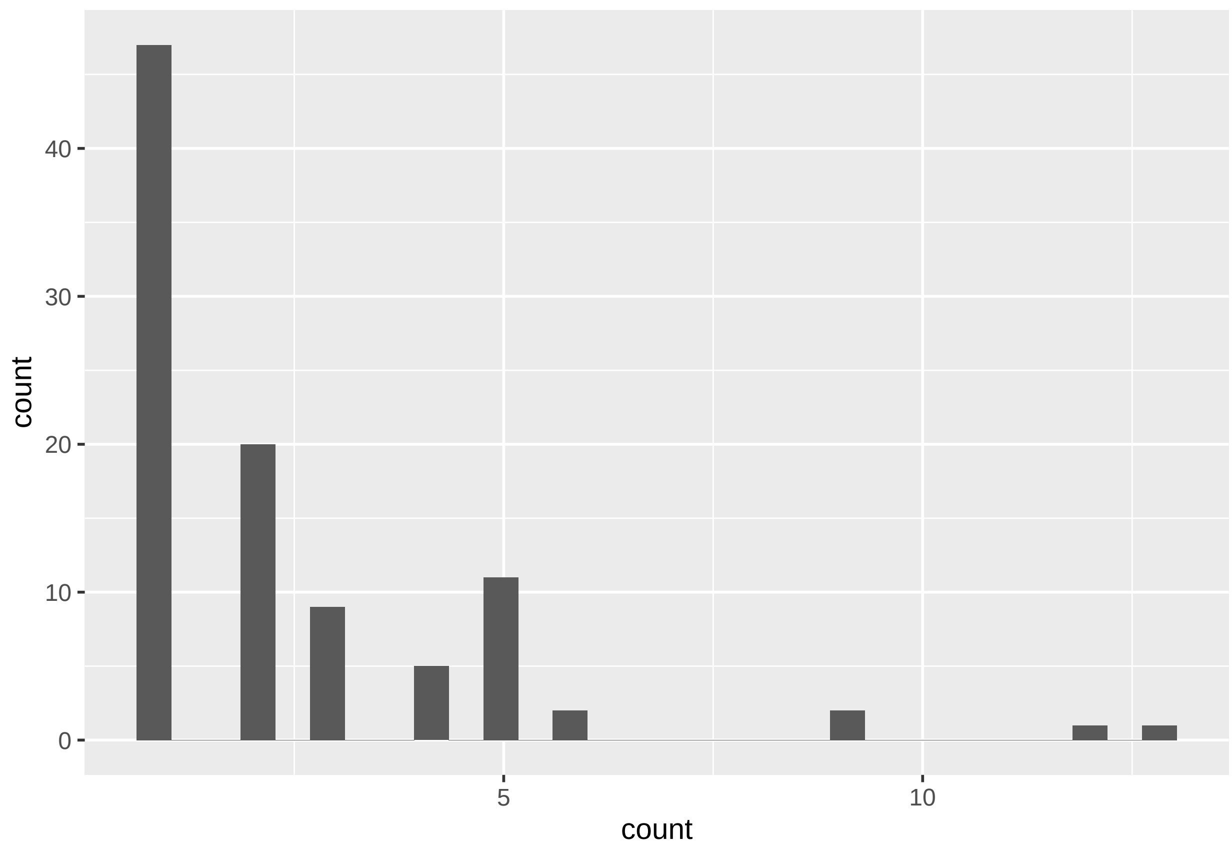
b. Plot a histogram of the hospital per grid cell count.

Most cells do not have any hospitals, and those cells have `NA` in the count colum. The resulting histogram does not reflect the true zeros in the data:

```
hosp_grid_hist <- ggplot(data=hosp_net_final, aes(x=count)) +
  geom_histogram()
hosp_grid_hist
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 275 rows containing non-finite values (stat_bin).
```

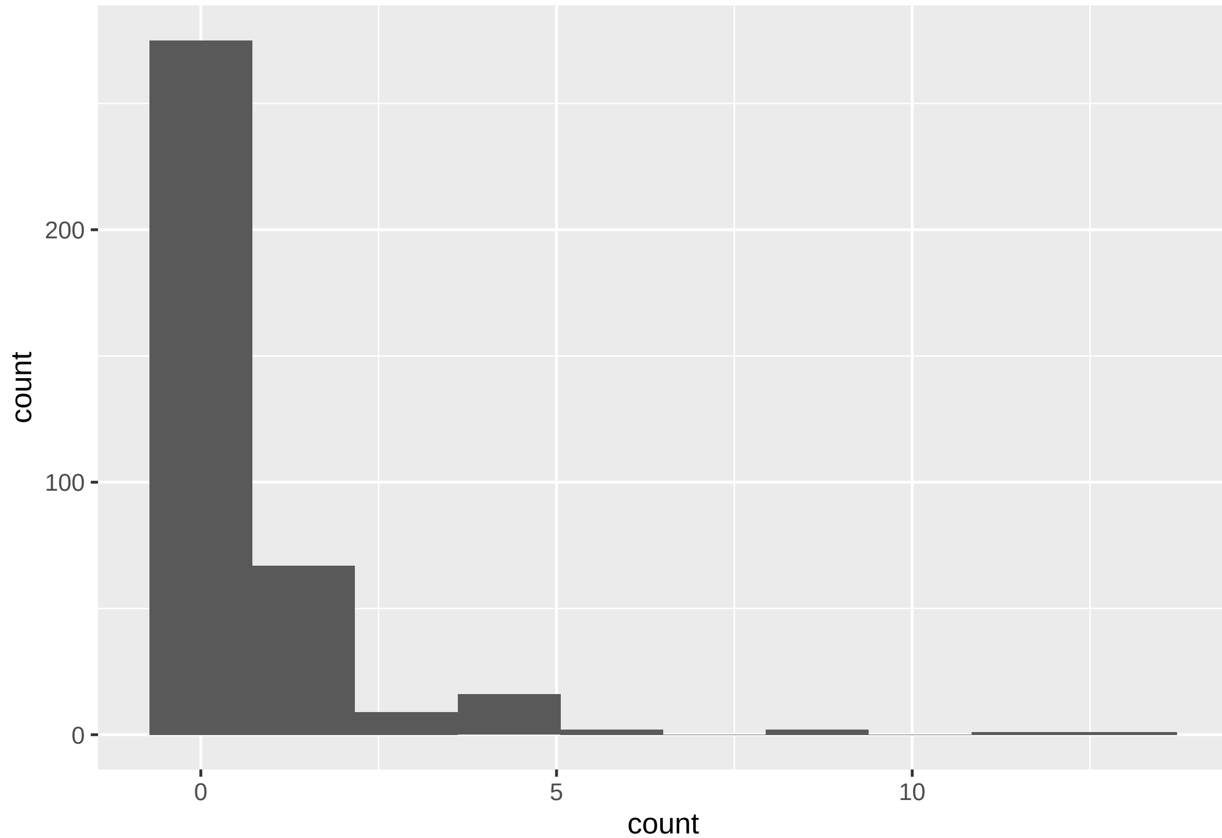


We can replace the `NA` values with zeros to help visualize the frequency of zero-hospital grid cells in the state:

```
hosp_net_final$count <- hosp_net_final$count %>% replace_na(0)
```

The new histogram gives more information on the zeros in the dataset:

```
hosp_grid_hist_zeros <- ggplot(data=hosp_net_final, aes(x=count)) +
  geom_histogram(bins=10)
hosp_grid_hist_zeros
```



5. Make a county-level fishnet.

a. Use the `filter()` method to subset a single county's data from the hospital dataset.

```
hosp_cumberland <- hosp %>% filter(COUNTY == 'CUMBERLAND')
```

b. Make a new fishnet grid for that county only, this time with half-mile resolution.

Fishnet boilerplate:

```
fishnet_half_mi_width <- 1609 * (1/2)
cumberland_border <- nj %>%
  filter(COUNTY=='CUMBERLAND') %>%
  st_transform(crs=NJ_PLANAR)
half_mi_net <- st_make_grid(
  x=cumberland_border,
  cellsize=fishnet_half_mi_width,
  what='polygons',
  square=TRUE,
  crs=NJ_PLANAR
  )
net_agg_half_mi <- st_as_sf(half_mi_net) %>% tibble::rowid_to_column(., "net_id")
net_intersect_half_mi <- st_intersects(cumberland_border, net_agg_half_mi)
fishnet_half_mi <- net_agg_half_mi[unique(unlist(net_intersect_half_mi)),]
```

(We've seen the fishnet boilerplate repeated a couple of times. In practice you would want to wrap this code in a helper function to ease reuse.)

c. Aggregate that county's hospital data into the half-mile fishnet grid.

Point-in-polygon join and aggregation:

```
cumberland_hosp_net <- st_join(
  st_transform(hosp_cumberland, NJ_PLANAR),
  fishnet_half_mi,
  join=st_within
  ) %>%
    st_drop_geometry()

cumberland_hosp_net_summary <- cumberland_hosp_net %>%
  group_by(net_id) %>%
  summarise(count=n())

cumberland_hosp_net_final <- st_as_sf(
  left_join(
    fishnet_half_mi,
    cumberland_hosp_net_summary,
    by='net_id'
    ),
  crs=st_crs(fishnet_half_mi))
```

b. Make the corresponding choropleth map and histogram for this county only.

We extracted the Cumberland County border above and transformed it to the `NJ_PLANAR` CRS. However, to get the corresponding basemap, we need to provide bounding box coordinates in the `WGS84` CRS.

```
cumberland_bbox <- unname(st_bbox(st_transform(cumberland_border, WGS84)))
cumberland_base_map <- get_stamenmap(bbox=cumberland_bbox, maptype="toner", force=TRUE)
```

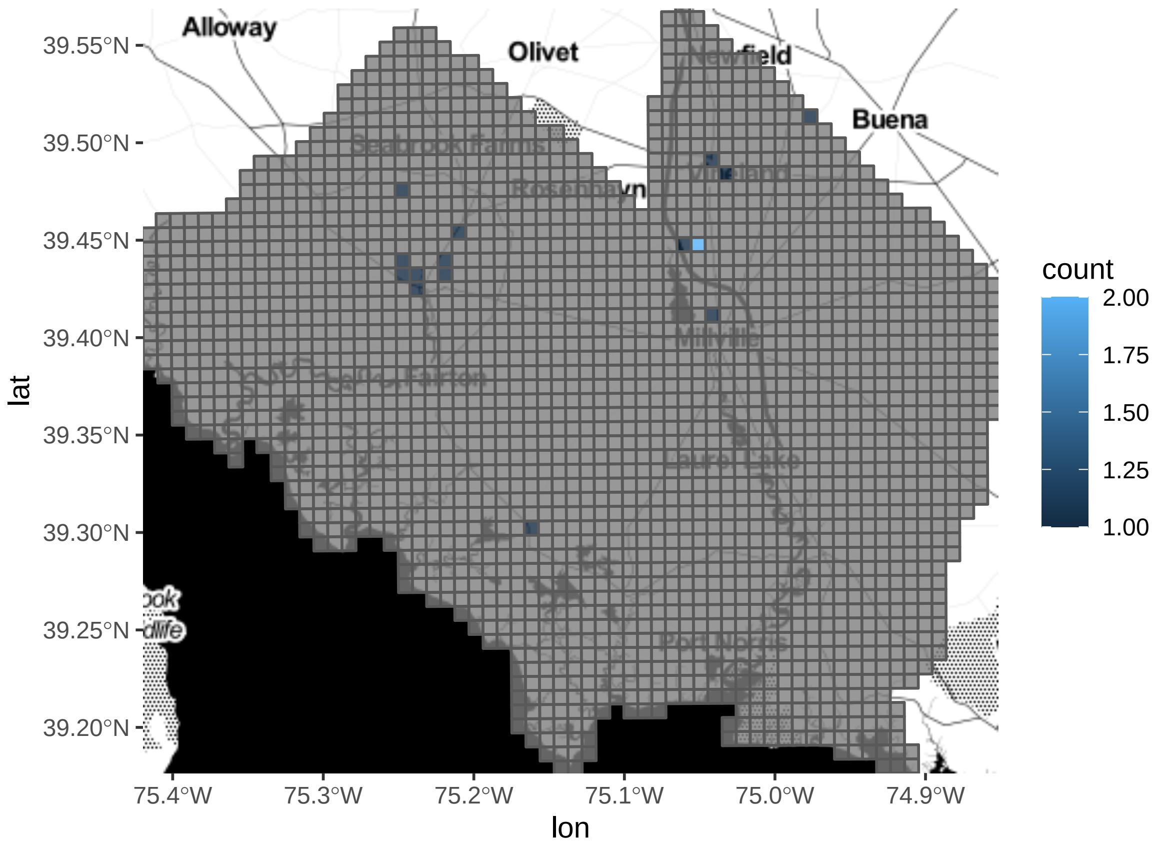
```
## Source : http://tile.stamen.com/toner/10/297/389.png

## Source : http://tile.stamen.com/toner/10/298/389.png

## Source : http://tile.stamen.com/toner/10/299/389.png

## Source : http://tile.stamen.com/toner/10/297/390.png

## Source : http://tile.stamen.com/toner/10/298/390.png

## Source : http://tile.stamen.com/toner/10/299/390.png
```

The choropleth map is a bit easier to read if we leave the zero counts as `NA` to avoid darkly shading the cells:

```
cumberland_net_map <- ggmap(cumberland_base_map) +
  geom_sf(
    data=st_transform(cumberland_hosp_net_final, crs=WGS84),
    aes(fill=count),
    inherit.aes = FALSE, alpha=0.8
    )
```

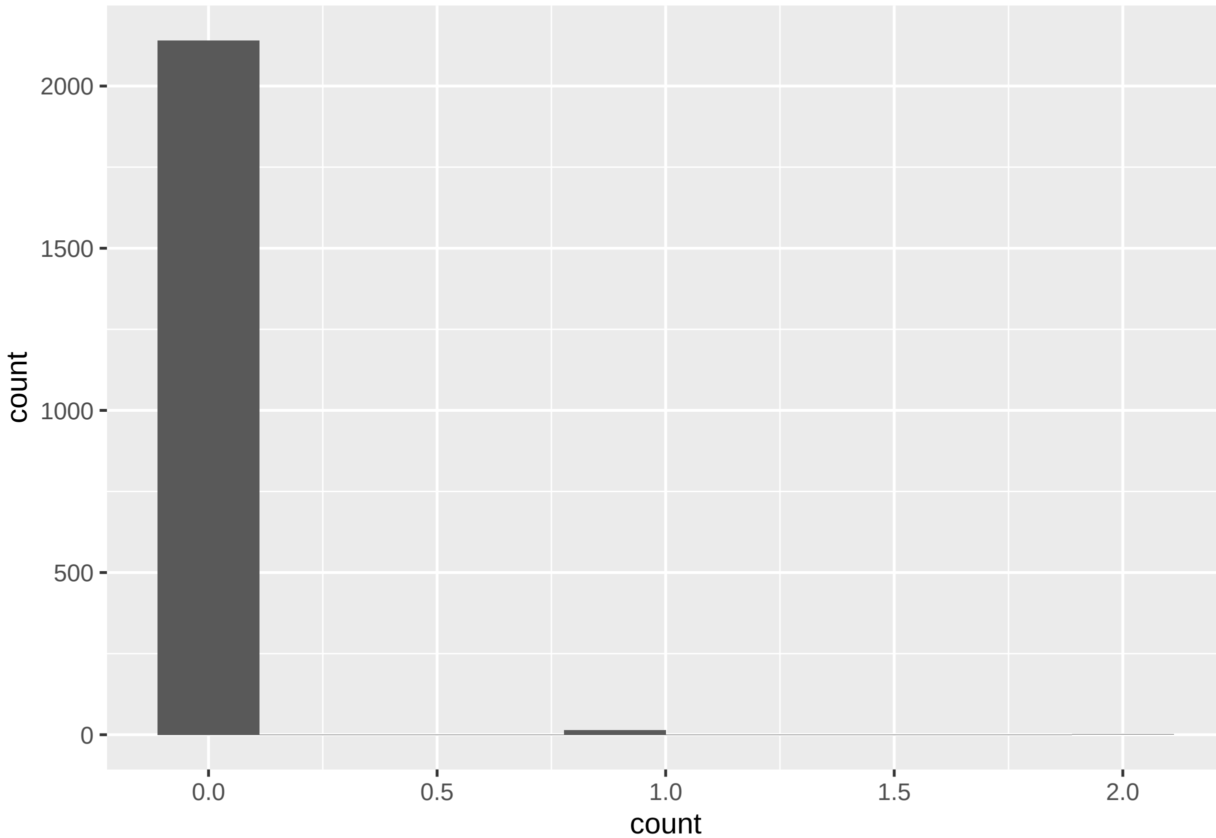
```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

```
cumberland_net_map
```



In contrast, the histogram is a bit more interpretable with the zeros represented:

```
cumberland_hosp_net_final$count <- cumberland_hosp_net_final$count %>% replace_na(0)
cumberland_grid_hist_zeros <- ggplot(data=cumberland_hosp_net_final, aes(x=count)) +
  geom_histogram(bins=10)
cumberland_grid_hist_zeros
```


count
2000
1500
1000
500
0
0.0
0.5
1.0
1.5
2.0
count