Difficulty of Trails at Macleish Station

CODE

SDS 192: MP4

Emily Rhyu

Smith College

Alicia Chen

Smith College

Elda Cervantes

Smith College

June 30, 2022

Prompt

MacLeish has approximately four miles of built trail for use by field station visitors. We would like to be able to classify these trails as "Easy," "Moderate," or "Difficult" based on their length and change in elevation. Use the spatial data to develop a heuristic that will classify these (or any other) trails based on their difficulty. You might want to consult existing trail rating systems. It would be cool if you could create elevation profiles (e.g., this one). Be sure to consult the extended example in MDSR, which walks you through creating an elevation map for one trail. This map has some noted weaknesses. Your goal should be to improve on it and/or make similar maps for all trails.

Supporting data includes: - Existing trails layer (trails) - 30' contour elevation data (elevation) - 3 meter contours can be retrieved from MassGIS (see help("mass_gis")) - Property boundary shapefile (boundary)

How We Did It

Our group chose Prompt 1, which asked us to classify the trails of MacLeish into levels of difficulty using their length and change in elevation. We began by exploring the MacLeish package in R Studio and familiarized ourselves with the dataset, "macleish_layers." Our next step was figuring out the length of each trail and calculating each one's elevation. We ran into some problems while creating a function to graph the elevation of all the trails. We first tested out the function with the longest trail in the dataset, and then tested it with other, randomly selected trails. This seemed to work, but we ran into another issue after doing this. Although our function worked when we passed separate trails in as arguments, it didn't work when we tried to use the map() function to do the same with the entire "trails_full" dataset. We then attempted to split the dataset into separate rows and use a for loop to call our function for each trail, but this gave us more errors (albeit different ones). Ultimately, we decided to graph every trail at once and facet wrap it by name to produce the elevation maps of all the trails. We encountered an issue where some of the trails' graphs were cut off, but were able to fix it after some help from our peers in the questions channel in the class Slack.

The ratings for the trails weren't very distinctive initially. All the trail levels were "Easy" according to the Shenandoah rating system as the area of MacLeish is pretty much flat and even the longest trail is only about 1.6 miles. Therefore, we created our own rating system in order to accommodate for the smaller scale: ratings 0-5 are ranked easy, ratings 5-10 are ranked moderate, and ratings 10-20 are ranked difficult.

Loading the data

HIDE
HIDE

```r
library(macleish)
```

HIDE
HIDE

```r
trails <- macleish_layers %>%
  pluck("trails")

contour <- mass_gis("contours250k")
```

```
## Reading layer `CONTOURS250K_ARC' from data source
##   `/private/var/folders/80/txdq3ctd01l3s67d97r4lk3w0000gn/T/Rtmpi2sj5R/contours250k'
##   using driver `ESRI Shapefile'
## Simple feature collection with 50395 features and 3 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 33874.42 ymin: 778055.9 xmax: 325713.8 ymax: 959744.8
## Projected CRS: NAD83 / Massachusetts Mainland
```

HIDE
HIDE

```r
boundary <- macleish_layers %>%
  pluck("boundary")

elevations <- macleish_layers %>%
  pluck("elevation")
```

Finding the Lengths and Elevations of all the Trails

HIDE
HIDE

```r
#find the lengths of all the trails by segments
trails_full <- trails %>%
  group_by(name) %>%
  summarize(num_segments = n()) %>%
  mutate(trail_length = st_length(geometry)) %>%
  arrange(desc(trail_length))

#finds elevation for all the trails by segments
trail_elevations <- trails_full %>%
  st_intersection(elevations) %>%
  st_cast("MULTIPOINT") %>%
  st_cast("POINT")
trail_elevations
```

name	num_segments	trail_length	OBJECTID	Shape_Leng	Shape_Area	CONTOUR_...	INDEX_ARC
<chr>	<int>	<units>	<int>	<dbl>	<dbl>	<dbl>	<chr>
Snowmobile Trail	2	2576.0933 [m]	1	5893.578	1033988	870	N
Snowmobile Trail	2	2576.0933 [m]	1	5893.578	1033988	870	N
Poplar Hill Road	2	1040.4175 [m]	1	5893.578	1033988	870	N
Poplar Hill Road	2	1040.4175 [m]	1	5893.578	1033988	870	N
Snowmobile Trail	2	2576.0933 [m]	1	5893.578	1033988	810	N
Eastern Loop	2	1939.2452 [m]	1	5893.578	1033988	810	N
Eastern Loop	2	1939.2452 [m]	1	5893.578	1033988	810	N
Eastern Loop	2	1939.2452 [m]	1	5893.578	1033988	810	N
Eastern Loop	2	1939.2452 [m]	1	5893.578	1033988	810	N
Western Loop	3	1350.0538 [m]	1	5893.578	1033988	810	N

1-10 of 43 rows | 1-8 of 10 columns Previous 1 2 3 4 5 Next

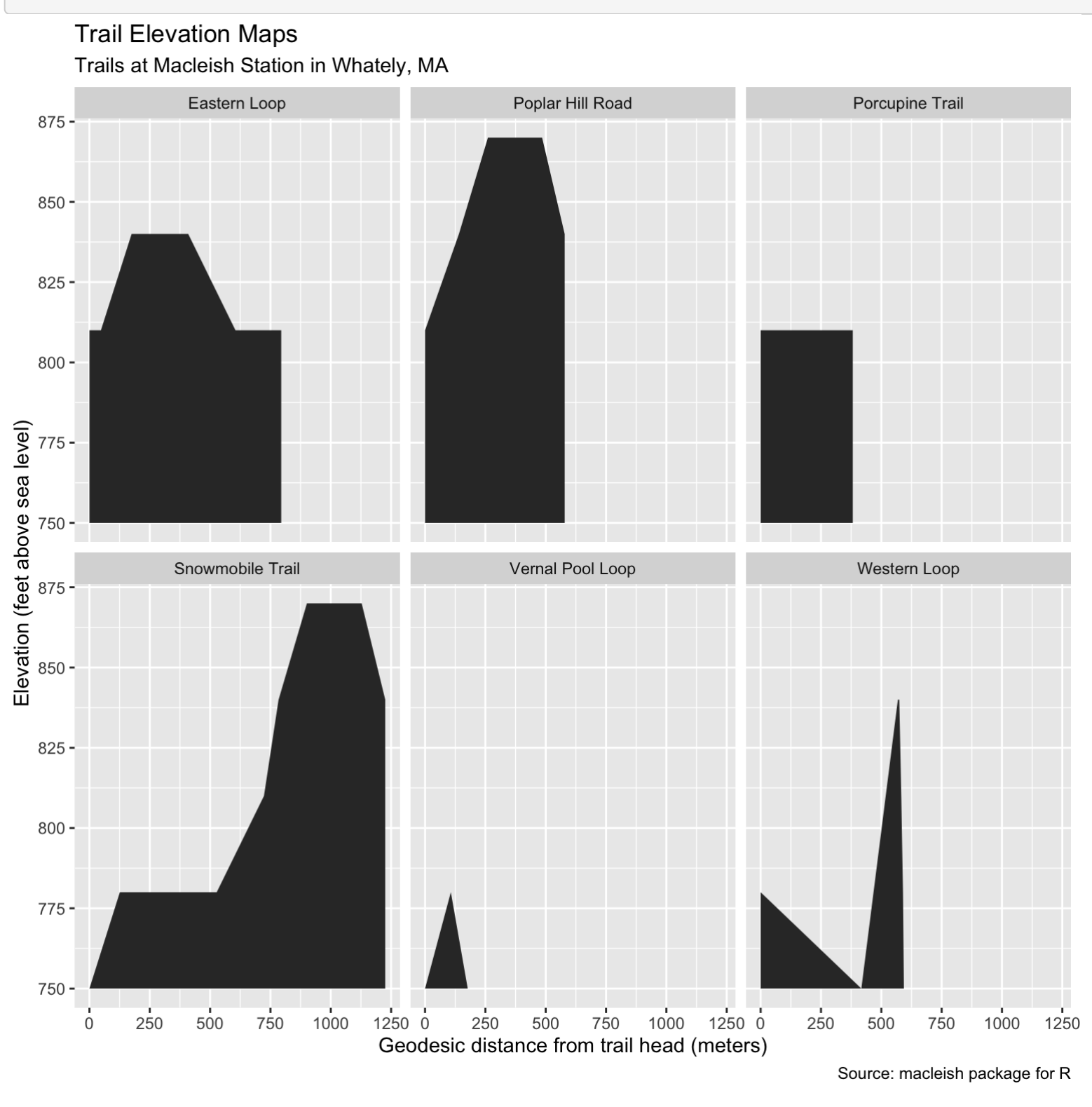
Elevation Maps

HIDE
HIDE

```r
elevationGrapher <- function(trail_elevations) {
  trail_elevations <- trail_elevations %>%
    group_by(name) %>%
    mutate(lat = st_coordinates(geometry)[, 2]) %>%
    arrange(lat) %>%
    mutate(distance_from_start = as.numeric(st_distance(geometry)[, 1]))


  ggplot(trail_elevations, aes(x = distance_from_start)) +
    geom_ribbon(aes(ymax = CONTOUR_FT, ymin = 750)) +
    scale_y_continuous("Elevation (feet above sea level)") +
    scale_x_continuous("Geodesic distance from trail head (meters)") +
    labs(title = "Trail Elevation Maps",
         subtitle = "Trails at Macleish Station in Whately, MA",
         caption = "Source: macleish package for R") +
    facet_wrap( ~ name)
}

elevationGrapher(trail_elevations)
```


Trail Elevation Maps
Trails at Macleish Station in Whately, MA
Eastern Loop
Poplar Hill Road
Porcupine Trail
Snowmobile Trail
Vernal Pool Loop
Western Loop
875
850
825
800
775
750
Elevation (feet above sea level)
0
250
500
750
1000
1250
Geodesic distance from trail head (meters)
Source: macleish package for R

The Trail Ratings using Shenandoah system

HIDE
HIDE

```r
trail_ratings <- trail_elevations %>%
  group_by(name) %>%
  summarize(
    gain = max(CONTOUR_FT) - min(CONTOUR_FT),
    trail_length = max(units::set_units(trail_length, "miles")),
    rating = sqrt(gain * 2 * as.numeric(trail_length))
  )

trail_ratings <- trail_ratings %>%
  mutate(difficulty = case_when(
    between(rating, 0, 5) ~ "Easy",
    between(rating, 5, 10) ~ "Moderate",
    between(rating, 10, 20) ~ "Difficult",
  ))
```

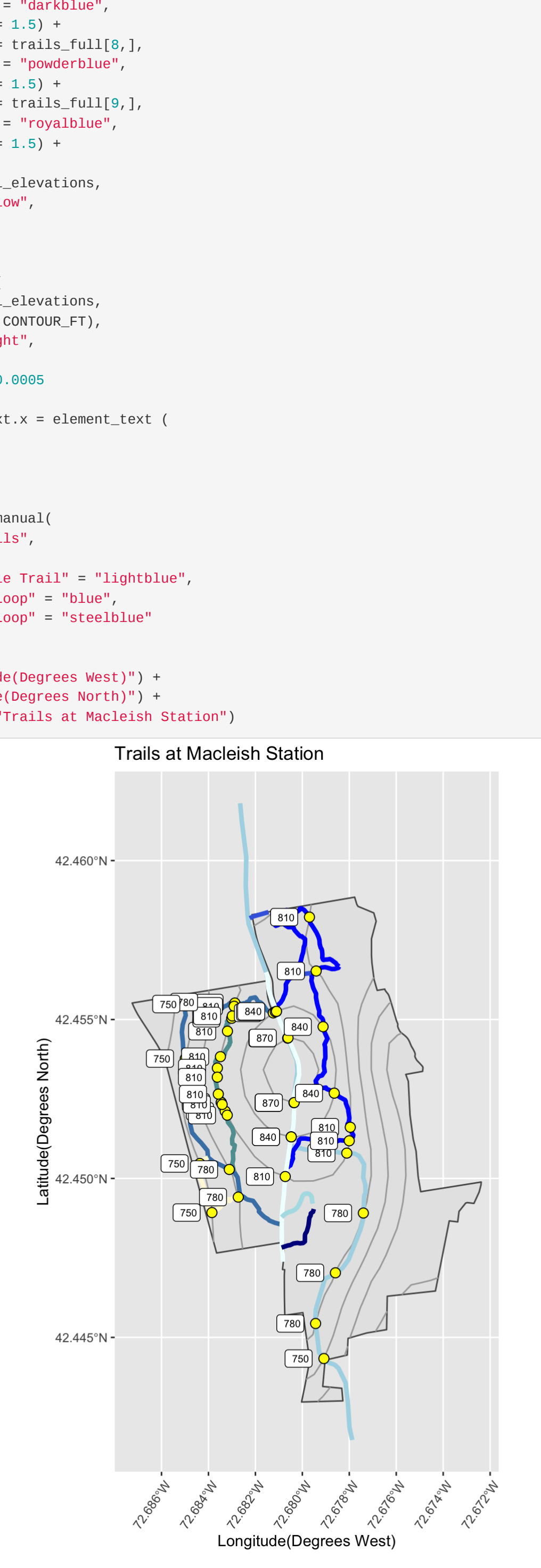
Static maps

HIDE
HIDE

```r
boundary_plot <- ggplot() +
  geom_sf(data = boundary)

bluePalette <-
  c(
    "lightblue",
    "blue",
    "steelblue",
    "azure",
    "cadetblue",
    "cornsilk",
    "darkblue",
    "powderblue",
    "royalblue"
  )

boundary_plot +
  geom_sf(data = elevations, color = "dark gray") +
  geom_sf(data = trails_full[1,],
          color = "lightblue",
          size = 1.5) +
  geom_sf(data = trails_full[2,],
          color = "blue",
          size = 1.5) +
  geom_sf(data = trails_full[3,],
          color = "steelblue",
          size = 1.5) +
  geom_sf(data = trails_full[4,],
          color = "azure",
          size = 1.5) +
  geom_sf(data = trails_full[5,],
          color = "cadetblue",
          size = 1.5) +
  geom_sf(data = trails_full[6,],
          color = "cornsilk",
          size = 1.5) +
  geom_sf(data = trails_full[7,],
          color = "darkblue",
          size = 1.5) +
  geom_sf(data = trails_full[8,],
          color = "powderblue",
          size = 1.5) +
  geom_sf(data = trails_full[9,],
          color = "royalblue",
          size = 1.5) +
  geom_sf(
    data = trail_elevations,
    fill = "yellow",
    pch = 21,
    size = 3
  ) +
  geom_sf_label(
    data = trail_elevations,
    aes(label = CONTOUR_FT),
    hjust = "right",
    size = 2.5,
    nudge_x = -0.0005
  ) +
  theme(axis.text.x = element_text (
    angle = 55,
    vjust = .89,
    hjust = .9
  )) +
  scale_colour_manual(
    name = "Trails",
    values = c(
      "Snowmobile Trail" = "lightblue",
      "Eastern Loop" = "blue",
      "Western Loop" = "steelblue"
    )
  ) +
  xlab("Longitude(Degrees West)") +
  ylab("Latitude(Degrees North)") +
  labs(title = "Trails at Macleish Station")
```



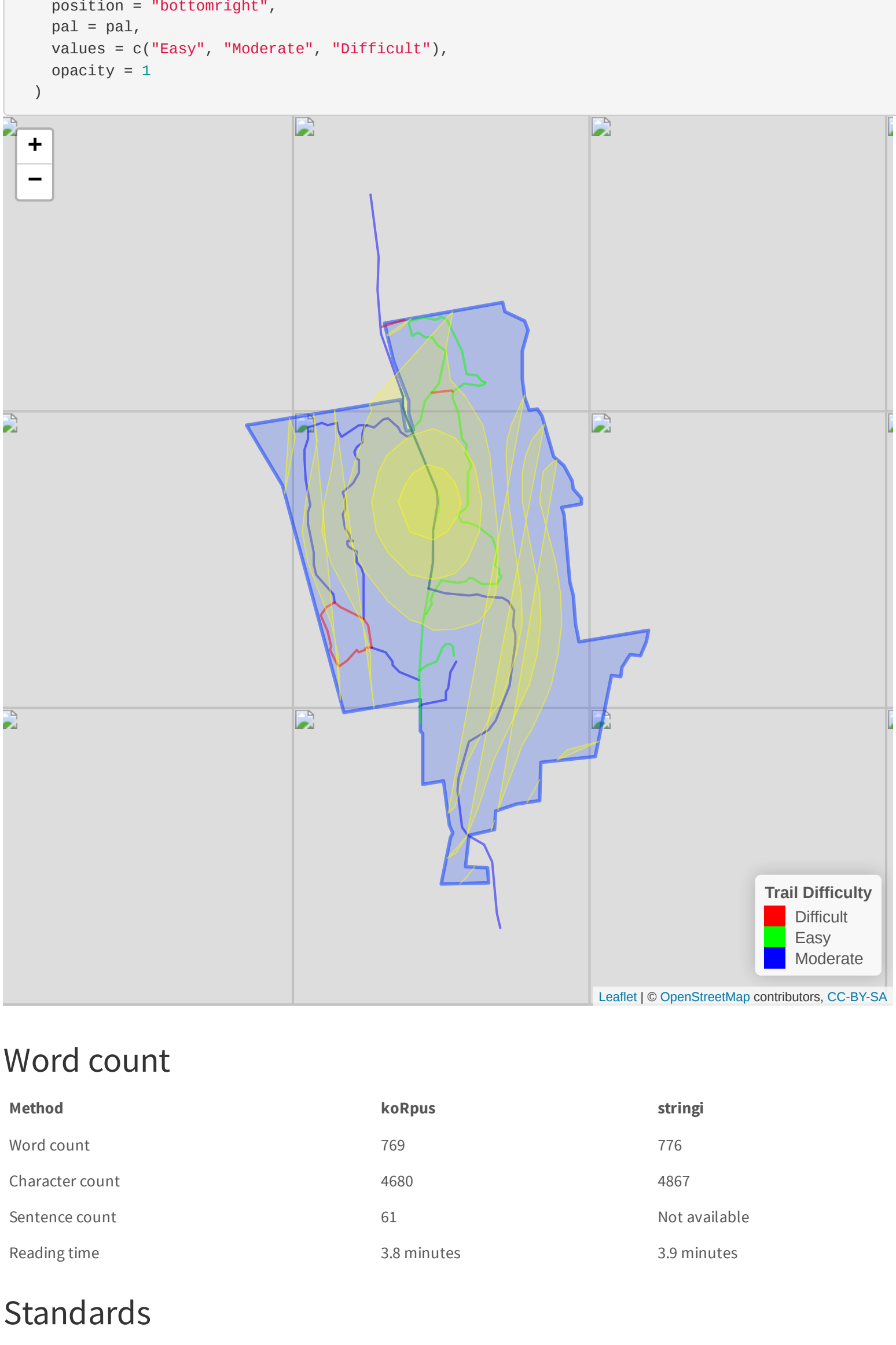
Dynamic maps

HIDE
HIDE

```r
library(leaflet)

pal <-
  colorFactor(
    palette = c("green", "blue", "red"),
    levels = c("Easy", "Moderate", "Difficult")
  )

leaflet() %>%
  addTiles() %>%
  addPolygons(data = boundary, weight = 3) %>%
  addPolylines(
    data = trails,
    weight = 2,
    color = ~ pal(trail_ratings$difficulty),
    label = trails$name
  ) %>%
  addPolygons(data = elevations,
              weight = 1,
              color = "yellow") %>%
  addLegend(
    title = "Trail Difficulty",
    position = "bottomright",
    pal = pal,
    values = c("Easy", "Moderate", "Difficult"),
    opacity = 1
  )
```



Word count

Method	koRpus	stringi
Word count	769	776
Character count	4680	4867
Sentence count	61	Not available
Reading time	3.8 minutes	3.9 minutes

Standards

In this assignment, we attempted the following standards:

- WRANGLING: We mastered the Wrangling standard since we used tools like group_by(), summarize(), and mutate() in order to manipulate our data for our purposes. We thereby optimized readability and performance.
- SPATIAL: We met the Spatial standard since we used shapefiles in conjunction with ggplot2 to create spatial plots.
- LEAFLET: We met the Leaflet standard since we created a leaflet map of the MacLeish area with geospatial layers.
- AESTHETICS: We mastered the aesthetics standard as the data graphics are customized and make use of different geoms and colors.
- MARKDOWN: We mastered the R Markdown standard since the code we've written compiles properly, and we haven't included things that would slow down the code unnecessarily, like the loading statements.
- GITHUB: We mastered the Github standard as we used Github to effectively work together.
- CONTEXT: We mastered the Context standard as all of our data graphics are clearly labeled and tell a story.
- ITERATION: We mastered the Iteration standard since we applied a function, elevationGrapher, to multiple rows and variables.
- FUNCTION: We mastered the Function standard since we wrote a user-defined function, elevationGrapher, that takes in the data on all the trails and then gives an elevation map for each one.

References

- Benjamin S. Baumer, Rose Goueth, Wencong Li, Weijia Zhang and Nicholas Horton (NA). macleish: Retrieve Data from MacLeish Field Station. R package version 0.3.6. http://github.com/beanumber/macleish
- Baumer, Benjamin, et al. Modern Data Science with R. CRC Press, 2021.
- Majerus, Rich, et al. "Creating a Color Palette Using ColorFactor." R, Datacamp, campus.datacamp.com/courses/interactive-maps-with-leaflet-in-r/plotting-points?ex=15.
- Margalit, Sivan. "Sort Values Into Groups by Ranges." RStudio Community, 11 Apr. 2020, community.rstudio.com/t/sort-values-into-groups-by-ranges/60881.
- Ogle, Derek H. "Collapsing Categories or Values." Derek Ogle, 30 Mar. 2018, derekogle.com/fishR/2018-03-30-Collapsing_Values.