NBI_report

Martin Menninger

8/16/2019

Download the data

This chunk downloads the data that you need to run the rest of the report. You should only need to run it once.

Import the data

```
years <- 1992:2018
state <- "IL"

Format <- read_csv("~/R/NBI/Inputs/Format.csv")
```

```
## Parsed with column specification:
## cols(
##   ITEM_NO = col_character(),
##   ITEM_NAME = col_character(),
##   `ITEM POSITION` = col_character(),
##   ITEM_LENGTHTYPE = col_character(),
##   ITEM_LENGTH = col_double(),
##   Precision = col_logical(),
##   TYPE = col_character(),
##   Skip = col_logical(),
##   Column_Name = col_character(),
##   F_Class = col_character(),
##   Same = col_character()
## )
```

```
fcol_types <- paste0(rep(Format$Same), collapse = "")

##Needed to set up loop. Should rewrite to eliminate need for this.
NBIAll<- data.frame(read_fwf(file = paste("NBI", state, "2018", ".txt", sep = ""),
                             fwf_widths(Format$ITEM_LENGTH, Format$Column_Name),
                             col_types = fcol_types))
NBIAll$NBI_Year = 2018 #add a column that indicates inventory year

for (i in years){
  NBIAll <- filter(NBIAll, NBIAll$NBI_Year != i) #cleares out old data
  temp <- read_fwf( file = paste("NBI", state, i, ".txt", sep = ""),
                    fwf_widths(Format$ITEM_LENGTH, Format$Column_Name),
                    col_types = fcol_types)

  temp$NBI_Year = i #add a column that indicates inventory year
  print(i)
  print(Sys.time())
```

```
  NBIAll <- bind_rows(NBIAll, temp)
}
```

```
## [1] 1992
## [1] "2019-08-16 16:30:01 CDT"
## [1] 1993
## [1] "2019-08-16 16:30:02 CDT"
## [1] 1994
## [1] "2019-08-16 16:30:03 CDT"
## [1] 1995
## [1] "2019-08-16 16:30:04 CDT"
## [1] 1996
## [1] "2019-08-16 16:30:05 CDT"
## [1] 1997
## [1] "2019-08-16 16:30:06 CDT"
## [1] 1998
## [1] "2019-08-16 16:30:07 CDT"
## [1] 1999
## [1] "2019-08-16 16:30:08 CDT"
## [1] 2000
## [1] "2019-08-16 16:30:09 CDT"
## [1] 2001
## [1] "2019-08-16 16:30:11 CDT"
## [1] 2002
## [1] "2019-08-16 16:30:12 CDT"
## [1] 2003
## [1] "2019-08-16 16:30:14 CDT"
## [1] 2004
## [1] "2019-08-16 16:30:16 CDT"
## [1] 2005
## [1] "2019-08-16 16:30:18 CDT"
## [1] 2006
## [1] "2019-08-16 16:30:19 CDT"
## [1] 2007
## [1] "2019-08-16 16:30:22 CDT"
## [1] 2008
## [1] "2019-08-16 16:30:24 CDT"
## [1] 2009
## [1] "2019-08-16 16:30:26 CDT"
## [1] 2010
## [1] "2019-08-16 16:30:28 CDT"
## [1] 2011
## [1] "2019-08-16 16:30:30 CDT"
## [1] 2012
## [1] "2019-08-16 16:30:33 CDT"
## [1] 2013
## [1] "2019-08-16 16:30:35 CDT"
## [1] 2014
## [1] "2019-08-16 16:30:38 CDT"
## [1] 2015
## [1] "2019-08-16 16:30:40 CDT"
## [1] 2016
## [1] "2019-08-16 16:30:43 CDT"
```

```
## [1] 2017
## [1] "2019-08-16 16:30:46 CDT"
## [1] 2018
## [1] "2019-08-16 16:30:48 CDT"
```

```
rm(Format)
rm(i, fcol_types, years, state)
#saveRDS(NBIAll, file = "nbiIL92_18.RDS")
```

Clean Up the Data

```
library(readr)
  Owner_code <- read_csv("~/R/NBI/Inputs/Owner_code.csv")
```

```
## Parsed with column specification:
## cols(
##   Code = col_character(),
##   Description = col_character()
## )
```

```
  condition_levels <- factor( levels = c("Good", "Fair", "Poor"), ordered = TRUE)


#Build NBI file ----
  nbi_trend <- NBIAll %>%
    filter(RECORD_TYPE_005A =="1") %>%
    select(STATE_CODE_001, COUNTY_CODE_003, LAT_016, LONG_017, WATERWAY_EVAL_071, HIGHWAY_SYSTEM_104, #
           APPR_WIDTH_MT_032, STRUCTURE_LEN_MT_049, DECK_WIDTH_MT_052,   ##Size
           DECK_COND_058, SUPERSTRUCTURE_COND_059, SUBSTRUCTURE_COND_060, CULVERT_COND_062,  ## Conditi
           OWNER_022, FUNCTIONAL_CLASS_026, YEAR_BUILT_027, ADT_029, SUFFICIENCY_RATING, CAT10, NBI_Yea
    left_join(maps::state.fips, by = c("STATE_CODE_001" = "fips")) %>% ## Join in the State
    mutate(county_code = as.integer(paste(STATE_CODE_001, COUNTY_CODE_003, sep = ""))) %>%  ##Format fo
    left_join(maps::county.fips, by = c("county_code" = "fips")) %>% ## Join in the County
    mutate(water = ifelse(WATERWAY_EVAL_071 == "N", 0, 1)) %>%
    mutate(age = 2019 - YEAR_BUILT_027) %>%
    left_join(Owner_code, by = c("OWNER_022" = "Code")) %>%
    #mutate(CAT10 = fct_recode(CAT10, "Good" = "G","Fair" = "F","Poor" = "P")) %>%
    mutate(    area_m = ifelse(DECK_WIDTH_MT_052 > 0,
                               STRUCTURE_LEN_MT_049 *DECK_WIDTH_MT_052, STRUCTURE_LEN_MT_049 *APPR_WIDTH
               min_con = pmin(DECK_COND_058,SUPERSTRUCTURE_COND_059,
                              SUBSTRUCTURE_COND_060, CULVERT_COND_062, na.rm = TRUE),
               Rating = ifelse(min_con<=4, "Poor", ifelse(min_con>=7, "Good", "Fair")))
```

```
## Warning in evalq(as.integer(paste(STATE_CODE_001, COUNTY_CODE_003, sep =
## "")), : NAs introduced by coercion
```

```
#nbi_trend$Rating <- factor(nbi_trend$Rating, levels = condition_levels)
nbi_trend$Rating <- factor(nbi_trend$Rating, levels = c("Good", "Fair", "Poor"))

#Latitude----
```

```
  deg <- as.numeric(stringr::str_sub(nbi_trend$LAT_016, 1, 2))
  min <- as.numeric(stringr::str_sub(nbi_trend$LAT_016, 3, 4))
  sec <- as.numeric(stringr::str_sub(nbi_trend$LAT_016, 5, 8))/100
  nbi_trend$latitude <- deg + min/60 + sec/3600

#Longitude----
  deg <- as.numeric(stringr::str_sub(nbi_trend$LONG_017, 1, 3))
  min <- as.numeric(stringr::str_sub(nbi_trend$LONG_017, 4, 5))
  sec <- as.numeric(stringr::str_sub(nbi_trend$LONG_017, 6, 9))/100

  nbi_trend$longitude <- -1 * (deg + min/60 + sec/3600)

  nbi <- nbi_trend %>% filter(NBI_Year == max(nbi_trend$NBI_Year))

   nbi <- nbi %>%
    filter(longitude > -92) %>%
    filter(latitude > 36) %>%
    filter(latitude < 50)

rm(condition_levels, deg, min, sec)
rm(Owner_code)
#rm(NBIAll)
rm(temp)
```

Add geography information for summary

```
library(sf)
```

```
## Linking to GEOS 3.6.1, GDAL 2.2.3, PROJ 4.9.3
```

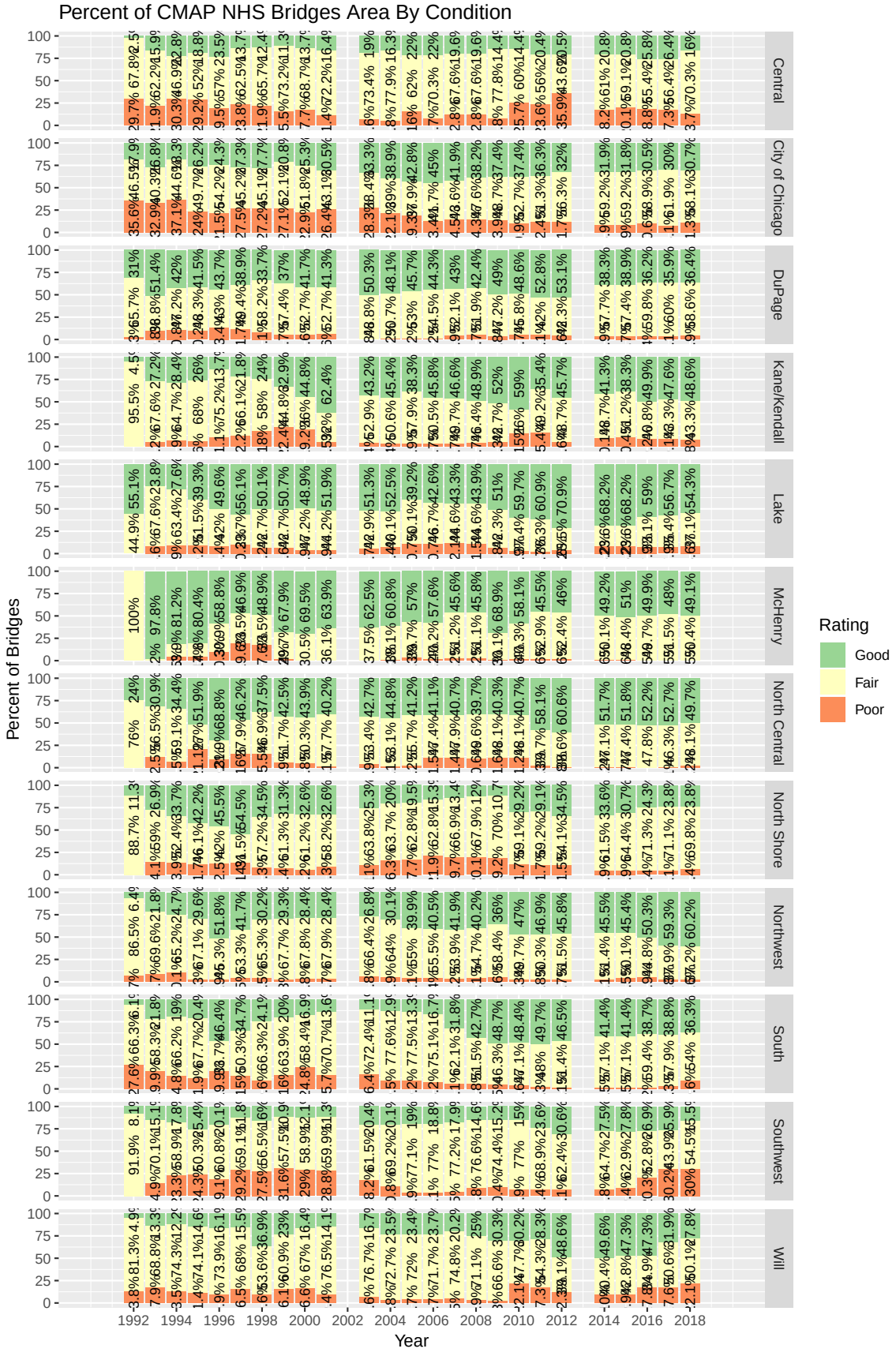
```
library(sp)
COM <- st_read("~/R/NBI/Inputs/CoM_CMAP_201303.shp", stringsAsFactors = FALSE)    #Local Council of May
```

```
## Reading layer `CoM_CMAP_201303' from data source `C:\Users\mmenninger.CMAP\Documents\R\NBI\Inputs\Col
## Simple feature collection with 12 features and 5 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 880400.2 ymin: 1651672 xmax: 1205878 ymax: 2123358
## epsg (SRID):    NA
## proj4string:    +proj=tmerc +lat_0=36.66666666666666 +lon_0=-88.33333333333333 +k=0.9999749999999999
```

```
#plot(COM)


#Make the NBI spatial
nbisp <- subset(nbi_trend, is.na(latitude) == F)
nbisp <- subset(nbisp, is.na(longitude) == F)
#nbisp<- filter(nbisp, COUNTY_CODE_003 == "031" )
#coordinates(nbisp) <- c("longitude", "latitude")
nbisp<- st_as_sf(nbisp, coords = c("longitude", "latitude"))
```

```r
nbisp <-st_set_crs(nbisp,  "+proj=longlat +ellps=GRS80 +datum=NAD83 +no_defs")
#nbisp<- st_as_sf(nbisp)
nbisp<- st_transform(nbisp, st_crs(COM))
z <- st_join(nbisp, COM)


#plot(subset(z, z$Council=="City of Chicago"))
```

```r
  z %>%
  filter(HIGHWAY_SYSTEM_104 == 1) %>%
  #filter(COUNTY_CODE_003 %in% c("031", "043", "089", "093", "097", "111", "197")) %>%
  filter(Rating != "NA") %>%
  group_by(NBI_Year, Rating, Council) %>%
  summarise(countt = n(), area = sum(area_m)) %>%
  group_by(NBI_Year, Council) %>%
  mutate(pct = area/sum(area)) %>%
  filter(Council != "NA") %>%

  ggplot( aes(x= NBI_Year, y = pct*100, fill = Rating)) +
  geom_bar(stat = "identity") +
  #geom_text(aes(label = paste(format(pct*100, digits = 1), "%")),
  geom_text(aes(label = paste(round(pct*100, digits = 1), "%", sep = "")),
            position = position_stack(vjust = 0.5), size = 3, angle = 90) +
  scale_x_continuous(breaks = seq(1992, 2018, 2), limits = c(1990, 2020))+
    scale_fill_brewer(type = "seq", palette="Spectral", direction = -1) +
  facet_grid(rows = vars(Council)) +
  #facet_wrap(~Council) +
  #coord_flip() +
  labs(x= "Year", y= "Percent of Bridges", title= "Percent of CMAP NHS Bridges Area By Condition")
```


Percent of CMAP NHS Bridges Area By Condition
Percent of Bridges
Year
Rating
Good
Fair
Poor
Central
City of Chicago
DuPage
Kane/Kendall
Lake
McHenry
North Central
North Shore
Northwest
South
Southwest
Will
0
25
50
75
100
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018