ASSESSMENT

1- Summarize historic conditions in Wisconsin

1. Download Biophysical Settings raster layer from LANDFIRE.gov (LF2016_BPS_200_CONUS). Download Wisconsin state boundary shapefile
2. Clip to Wisconsin state boundary (Wisconsin_State_Boundary_24K); save to gdb
3. Symbolize by BpS_Name; Export map = map of historic vegetation conditions
4. Symbolize by mFRI; Export map = map of historic fire regime
5. Symbolize by FRG_NEW (i.e. fire regime groups. Check documentation at: https://landfire.gov/frg.php) = map of historic fire severity
6. Export attribute table as wi200bps.csv
7. Calculate area and fire regime summaries in R (bps_stats.R)

```
# BpS summaries
# Our analysis used BpS LANDFIRE 2016, version 2.0.0 (i.e. 200)

library(dplyr)
library(tidyverse)
library(readr)
library(ggplot2)

bps <- read_csv("d:\\wi200bps.csv") # csv is exported attribute table from the wisconsin BpS raster layer

# Part 1 - Area summaries and historical fire regimes

# create variable p as percent of total count, per row.  sum = total count everything combined
bps$sum <- sum(bps$COUNT)
print(bps$sum)

bps <- within(bps, p <- as.numeric(COUNT/bps$sum*100))
head(bps)
head(bps$p)

# Sort data table
# Because our dataset crosses model zones, there are on occasion multiple rows for a BpS for each zone. We want to take the attributes from the BpS-Zone most common in our project extent.
bpsOrder <- bps %>%
  arrange(desc(BPS_CODE), desc(COUNT))
print(bpsOrder)

# Calculate counts by BpS using our sorted dataset
bpsClean <- bpsOrder %>%
  group_by(BPS_NAME) %>%
  summarise(BPS_CODE = first(BPS_CODE), COUNT = sum(COUNT), area = sum(COUNT)*30*30, acres = sum(COUNT)*30*30/4046.86, perArea = sum(p), FRI_ALLFIR = first(FRI_ALLFIR), annual = acres/FRI_ALLFIR, PRC_REPLAC = first(PRC_REPLAC), annRep = annual*PRC_REPLAC, GROUPVEG = first(GROUPVEG)) %>%   ## 30x30 M resolution. 1 acre = 4046.86 sqm Output = acres
  arrange(desc(area))
bpsClean <- bpsClean %>% mutate_if(is.numeric, list(~na_if(., Inf))) %>% replace(is.na(.), 0) # get rid of values calculated as Inf (value/0 = Inf)
print(bpsClean)
write_csv(bpsClean, "d:\\wiBpsSummarized.csv")


# Take a look at just the top 10 most common BpS's i.e. historical vegetation conditions
```

```
bps10 <- bpsClean %>% slice (1:10) %>%
  arrange(desc(area))
print(bps10)

# plot
p <- ggplot(bps10) +  geom_bar(aes(x = reorder(BPS_NAME, perArea), y = perArea), stat = 'identity') +
scale_y_continuous(limits=c(0,30))
p + coord_flip()

# Export plot and edit in Adobe Illustrator to match colors of map output

# Part 1B:
# Take a look at where fire occurred most frequently, by major vegetation group
bpsFireGroup <- subset(bpsClean, GROUPVEG %in% c("Conifer", "Grassland", "Hardwood", "Hardwood-Conifer",
"Riparian")) %>%
  group_by(GROUPVEG) %>%
  summarise(annual = sum(annual)) %>%
  arrange(desc(annual))  # annual = annual acres burned
print(bpsFireGroup)

p <- ggplot(bpsFireGroup) +  geom_bar(aes(x = reorder(GROUPVEG, annual), y = annual), stat = 'identity') +
scale_y_continuous(limits=c(0,2000000))
p + coord_flip()

# Take a look at where fire occurred most frequently, by BpS
bpsFire <- subset(bpsClean, GROUPVEG %in% c("Conifer", "Grassland", "Hardwood", "Hardwood-Conifer",
"Riparian")) %>%
  group_by(BPS_NAME) %>%
  summarise(annual = sum(annual)) %>%
  arrange(desc(annual)) # annual = annual acres burned
print(bpsFire)

bpsFire10 <- bpsFire %>% slice (1:10) %>%
  arrange(desc(annual))

p <- ggplot(bpsFire10) +  geom_bar(aes(x = reorder(BPS_NAME, annual), y = annual), stat = 'identity') +
scale_y_continuous(limits=c(0,2000000))
p + coord_flip()

# Export plot and edit in Adobe Illustrator to match colors of map output

# Part 2 - Summarize historic fire Severity

# Recode fire regime groups
bps$sev <- recode(as.character(bps$FRG_NEW),
                  "I-A" = "Low/mixed",
                  "I-B" = "Low/mixed",
                  "I-C" = "Low/mixed",
                  "II-A" = "Replacement",
                  "II-B" = "Replacement",
                  "II-C" = "Replacement",
                  "III-A" = "Low/mixed",
                  "III-B" = "Low/mixed",
                  "III-C" = "Low/mixed",
                  "IV-A" = "Replacement",
                  "IV-B" = "Replacement",
                  "IV-C" = "Replacement",
                  "V-A" = "Any/Replacement",
```

```
        "V-B" = "Any/Replacement")

head(bps$sev)


# Percent replacement by veg group - categorical, stacked bar
bpsSev <- bps %>%
  group_by(GROUPVEG, sev) %>%
  summarise(perCount = sum(p)) %>%
  arrange(GROUPVEG)
print(bpsSev, n=nrow(bpsSev))

# subset to remove open water and shrubland which as < 0.1%, and open water which is NA
p <- ggplot(subset(bpsSev, GROUPVEG %in% c("Conifer", "Grassland", "Hardwood", "Hardwood-Conifer", "Riparian")))
+
  aes(fill = sev, x = reorder(GROUPVEG, perCount), y = perCount) +
  geom_bar(position="stack", stat = "identity")
p + coord_flip()

# Export plot; edit colors in Adobe Illustrator to match map output
```

8. Formalize maps and associated figures in Adobe Illustrator.

2- Visualize current conditions in Wisconsin, unmodified model output

9. Download Existing Vegetation raster layer from LANDFIRE.gov (LF2016_EVT_200_CONUS). Download Wisconsin state boundary shapefile
10. Clip to Wisconsin state boundary (Wisconsin_State_Boundary_24K); save to gdb
11. Symbolize by EVT_PHYS; Export map = map of current vegetation conditions (unmodified model output)

3- Summarize current conditions in Wisconsin, modified output

Create the modified map of fire-dependent vegetation in Wisconsin#####

12. The most time-consuming aspect of a fire needs assessment is creating a three-way crosswalk schema to integrate current vegetation conditions (e.g. EVT layer) with fire regime information from the BpS layer and local plant community descriptions (Wisconsin Natural Heritage Inventory Plant Communities). There is not a one:one relationship between EVT_NAME and BPS_NAME, therefore, subjective connections are made based on the analyst's research and knowledge. Simultaneously, we crosswalked the EVT-BPS crosswalk to the Wisconsin Natural Plant Community descriptions as listed in the Natural Heritage Inventory (part of the national NatureServe classification), however, a single EVT was paired with one or more Natural Plant Communities. The crosswalk was manually compiled in excel for ease of manipulation.
13. Build the crosswalk spreading starting with EVT_NAME and EVT_VALUE. Manually assign an associated BPS_NAME to each EVT_NAME based on best match to BPS

attributes, specifically fire regime. Manually assign associated Wisconsin Natural Plant Communities and Rarity Rankings (State Rarity Ranking). Where multiple WI Natural Plant Communities were assigned to a row, average the State Rarity Ranking. The final Rarity ranking reflected the inverse of the State Rarity Ranking such that higher values were associated with higher rarity. (6 - ave(State Rarity Ranking)). In the scenario that fire regime information obtained from best-fit BPS attributes does not reflect fire regime characteristics for the current vegetation (evt) of interest, the fire regime parameters can be manually adjusted based on expert opinion and best available scientific research. For simplicity, our analysis used research and stakeholder input to assign BPS associations, but we did not manipulate fire regime parameters directly. Add a column to distinguish natural vegetation and non-natural vegetation. Non-natural vegetation will be excluded from the analysis

14. Following protocol from the 2014 Wisconsin Fire Needs Assessment (Hmielowski et al. 2016), we identified the following vegetation types requiring further manipulation in arcGIS:
    a. "Recently logged…" and "Recently burned…" vegetation types will be converted to adjacent land cover (see step 18)
    b. Non-natural vegetation (e.g. Developed-roads, Developed-low intensity, Developed-medium intensity, Developed-high intensity) will be removed.
    c. Agricultural vegetation will be excluded with the exception of EVT's representing possible managed grassland (e.g. "Eastern Cool Temperate Pasture and Hayland", "Eastern Cool Temperate Developed Ruderal Grassland"). Grassland, hereafter, "Managed Grassland" will be separated from hayland if occurring within the historical extent of tallgrass prairie and savanna, *and*, if not classified as Row Crop agriculture in the 2013 or 2019 National Land Cover Database. The resulting "Managed Grassland" EVT to be included in the fire needs assessment represents plausible permanent grassland or CRP parcels providing potential habitat for native prairie species and grassland birds (Hmielowski et al. 2016). See Step 18 for methodology
15. Join the crosswalk csv to the EVT attributes and BPS attributes. (crosswalkJoin.R)

```
library(dplyr)
library(tidyverse)
library(readr)
library(ggplot2)

# Select bps name row with the highest count; use for FRI information ** So, using most abundant bps--zone row for FRI.
This is preferable to averaging or selecting just a single zone
# bpsFRI <- bps %>%
#   group_by(BPS_NAME) %>%
#   slice(which.max(COUNT))
# print(bpsFRI)
# write_csv(bpsFRI, "d:\\wi200bpsFRI.csv")

# JOIN EVT ATTRIBUTES

cross <- read_csv("d:\\crosswalkAug9.csv")  # use final crosswalk file. see master sheet on google docs
```

```
evtAtt <- read_csv("d:\\wi200evtT.csv")  # this is the output of EVT raster joined with EVT attributes. Info gathered from join is Count info

# join evt attributes

# left join keeps all rows from x (crosswalk), and all columns from x and y (evt attributes)
# join on evt numerical code (VALUE, EVT_VALUE)


join <- cross %>% left_join(evtAtt, by = c("EVT_VALUE" = "VALUE"))

# Drop unwanted rows

join <- select (join,-c(OBJECTID,VALUE_1,EVT_N,COUNT,R,G,B,RED,GREEN,BLUE))
#join <- join %>% .[1:108,]  # odd excel bug: spreadsheet has 1000 rows; most empty
print(join)


# JOIN BPS ATTRIBUTES

bpsFRI <- read_csv("d:\\wi200bpsFRI.csv")

# join bpsFRI data to evt table. bpsFRI dataset has a single row per BPS_NAME

# left join keeps all rows from x (evt), and all columns from x and y (bps)

#inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),...)

join2 <- join %>% left_join(bpsFRI, by = c("BPS_NAME" = "BPS_NAME"))

# Drop unwanted rows

join2 <- select (join2,-c(OBJECTID,VALUE,COUNT,OID_1,VALUE_1,R,G,B,RED,GREEN,BLUE,sum,p))
join2 <- join2 %>% .[1:108,]  # remove extra rows (excel bug)
join2 <- rename(join2, EVT_NAME = EVT_NAME.x)  # ditch join appendage
Join2 <- join2 %>% replace(is.na(.), 0) # replace NA's with 0's
print(join2)
write_csv(join2, "d:\\crosswalkJoinAug9.csv")

# csv output has all EVT200 and BPS200 attributes excluding duplicate attributes, count, oid, objectid, and color scheme (Most information is not needed)
```

16. The joined crosswalk attribute table will later be joined with our evt raster.
17. Identify natural vegetation in Wisconsin by modifying the EVT raster layer. Exclude open water, developed areas, mines, agricultural except "Eastern Cool Temperate Pasture and Hayland" (see step 24), and recently burned and logged areas (see step 20). See step 21 for separating 'pasture' from 'hayland'.

18. Exclude "Recently Burned…" and "Recently Logged…" and convert to the vegetation landcover type (EVT_NAME) that shares the greatest perimeter.
    ArcGIS Shrink Tool using a 9x9 neighborhood.
    Using arcpy Python environment:

```
import arcpy
```

```
from arcpy import env
from arcpy.sa import *

out_raster = arcpy.sa.Shrink("wi200evt", 9, [7195,7196,7197,7198,7199,7200,7191,7192,7193],
"MORPHOLOGICAL"); out_raster.save(r"C:\Users\Stacey
Marion\Documents\ArcGIS\Projects\fna\fna.gdb\Shrink_wi200evt")
```

19. From “Eastern Cool Temperate Pasture and Hayland”, separate hay/fallow field from plausible managed grassland by: excluding areas not within the historic extent of prairie/savanna (orig_veg_cover_dissbyoakgrass.shp), and excluding areas classified as “Cultivated Crop” in either the National Land Cover Database (NLCD) imagery in 2013 or 2019. Rename the remaining area “Managed Grassland” and assign an EVT_VALUE of 1.

```
HistVeg = Raster('HistVeg_Raster')

EVT = Raster('Shrink_wi200evt')

nlcd = Raster('Extract_NLCD1')  # this is the name of the .tiff clipped to wi

MaskHist = Con(((HistVeg==1) & (EVT==7977)), 1, EVT)  # EVT value 1 for “Managed Grassland”

MaskHist.save("MaskHist")

MaskNLCD = Con(((MaskHist==1) & (nlcd!=82)), 1, MaskHist)

MaskNLCD.save("MaskNLCD")
```

20. Manually assign EVT_NAME “Managed Grassland” to row where EVT_VALUE = 1
21. Join our raster output (i.e. manipulated raster) with crosswalk attribute table (csv)

```
#arcpy.management.JoinField(in_data, in_field, join_table, join_field, {fields})

inData = MaskNLCD

inField = "Value"

joinTable = "crosswalkJoinAug15.csv"

joinField = "EVT_VALUE"

Join = arcpy.JoinField_management(inData, inField, joinTable, joinField)

Join.save(‘wiModifiedEvt’) # saves raster to working environment (i.e. geodatabase)

# Load Join to map
```

22. Calculate area summaries using the modified landcover type dataset. Export Join attribute table as csv.

23. Calculate area summaries in R (evt_stats.R)

```
# EVT summaries
# Our analysis used EVT LANDFIRE 2016, version 2.0.0 (i.e. 200)

library(dplyr)
library(tidyverse)
library(readr)
library(ggplot2)

evt <- read_csv("d:\\wiModifiedEvt.csv") # csv is exported attribute table from the wisconsin EVT raster
layer

# Note:  these summaries reflect the "as model output" version of evt summaries prior to any crosswalk
modification
# therefore, any areas later parsed out from agriculture as managed grassland are still counted within
"Agricultural" in this summary

# create variable p as percent of total count, per row
evt$sum <- sum(evt$COUNT)
print(evt$sum)

evt <- within(evt, p <- as.numeric(COUNT/evt$sum*100))
head(evt)
head(evt$p)

# Primary summaries of interest are the most abundant evts and evt vegetation groups present on today's
landscape.
# In Wisconsin, for example, much of the landscape is represented by Developed, Agricultural, Ruderal, or
Exotic communities

# Summarize by most common evt's
evtClean <- evt %>%
  group_by(EVT_NAME) %>%
  summarise(area = sum(COUNT)*30*30, acres = sum(COUNT)*30*30/4046.86, perArea = sum(p),
EVT_PHYS) %>%   ## 30x30 M resolution. 1 acre = 4046.86 sqm Output = acres
  arrange(desc(area))
print(evtClean)

evt10 <- evtClean %>% slice (1:10)
print(evt10)

p <- ggplot(evt10) +  geom_bar(aes(x = reorder(EVT_NAME, perArea), y = perArea), stat = 'identity') +
scale_y_continuous(limits=c(0,20))
p + coord_flip()

# Summarize by evt vegetation groups (e.g. developed, agricultural, exotic, riparian, etc.)
evtGroup <- evt %>%
  select (everything()) %>%
  mutate(Group = case_when (
    EVT_PHYS == "Agricultural" ~ "Agricultural",
```

```
   EVT_PHYS == "Riparian" ~ "Riparian",
   EVT_PHYS == "Hardwood" ~ "Hardwood",
   EVT_PHYS == "Developed" ~ "Ruderal",  # see EVT descriptions.  "Developed" includes vegetated
communities with no historical reference, e.g. ruderal
   EVT_PHYS == "Conifer" ~ "Conifer",
   EVT_PHYS == "Conifer-Hardwood" ~ "Conifer-Hardwood",
   EVT_PHYS == "Open Water" ~ "Open Water",
   EVT_PHYS == "Developed-Roads" ~ "Developed",
   EVT_PHYS == "Exotic Herbaceous" ~ "Exotic",
   EVT_PHYS == "Developed-Low Intensity" ~ "Developed",
   EVT_PHYS == "Developed-Medium Intensity" ~ "Developed",
   EVT_PHYS == "Developed-High Intensity" ~ "Developed",
   EVT_PHYS == "Grassland" ~ "Grassland",
   EVT_PHYS == "Exotic Tree-Shrub" ~ "Exotic",
   EVT_PHYS == "Shrubland" ~ "Shrubland",
   EVT_PHYS == "Quarries-Strip Mines-Gravel Pits-Well and Wind Pads" ~ "Quarries-Strip Mines-Gravel
Pits-Well and Wind Pads",
   EVT_PHYS == "Sparsely Vegetated" ~ "Sparsely Vegetated"
   ))

evtGroup <- evtGroup %>%
  group_by(Group) %>%
  summarise(area = sum(COUNT)*30*30, acres = sum(COUNT)*30*30/4046.86, perArea = sum(p)) %>%   ##
30x30 M resolution. 1 acre = 4046.86 sqm Output = acres
  arrange(desc(area))
print(evtGroup)

evt10 <- evtGroup %>% slice (1:10)

p <- ggplot(evt10) +  geom_bar(aes(x = reorder(Group, perArea), y = perArea), stat = 'identity') +
scale_y_continuous(limits=c(0,35))
p + coord_flip()

# Export plot and edit in Adobe Illustrator
```

24. From the modified landcover type raster (wiModifiedEvt), create a natural vegetation raster excluding open water, developed areas, mines, and agricultural types.

```
#use extract to create new raster

#ExtractByAttributes(in_raster, where_clause)

Natural = ExtractByAttributes(Join, '"evt_natVeg" >= 1')

Natural.save('Natural') # saves raster to working environment (i.e. geodatabase)
```

*Export attribute table as final joined natural vegetation layer with all attributes needed for further analysis

Calculate quantitative summaries for fire needs

25. Export raster attribute table as final joined natural vegetation layer with all attributes needed for further analysis.

26. Calculate fire regime summaries for natural vegetation landcover types (modified EVT_NAME) (final_stats.R)

```
# Fire regime summaries

library(dplyr)
library(tidyverse)
library(readr)
library(ggplot2)

final <- read_csv("d:\\joinAug10.csv") # final fire dependent vegetation areas  (joinAug10.csv = export from raster.  new evt classification, joined with all evt, bps attributes + count data)

# create variable p as percent of total count, per row
final$sum <- sum(final$Count)
print(final$sum)

final <- within(final, p <- as.numeric(Count/final$sum*100))
head(final)
head(final$p)

# Calculate annual burning needs

# annualAll ~ based on mean FRI (FRI_ALLFIR), expected annual burning needs (units = acres)

finalFRI <- final %>%
  group_by(EVT_NAME) %>%
  summarise(WI_Nat_Comm, area = sum(Count)*30*30, acres = sum(Count)*30*30/4046.86, FRI_ALLFIR, annualALL = (acres/FRI_ALLFIR), FRI_REPLAC, PRC_REPLAC, annualREPLAC = (annualALL*PRC_REPLAC/100), FRI_MIXED, PRC_MIXED, annualMIXED = (annualALL*PRC_MIXED/100), FRI_SURFAC, PRC_SURFAC, annualSURFAC = (annualALL*PRC_SURFAC/100)) %>%
  arrange(desc(annualALL))
finalFRI <- finalFRI %>% mutate_if(is.numeric, list(~na_if(., Inf))) %>% replace(is.na(.), 0) # remove Inf (0/Value; replace with 0)
finalFRI <- finalFRI %>% arrange(desc(annualALL)) # re-sort
print(finalFRI)


write_csv(finalFRI, "d:\\natVegFRI.csv")
```

Calculate quantitative indices for prioritization

27. Convert our output raster (i.e. natural vegetation) to a polygon feature class.

#Raster to Polygon :

```
with arcpy.EnvManager(outputZFlag="Disabled", outputMFlag="Disabled"):
arcpy.conversion.RasterToPolygon("Natural", r"C:\Users\Stacey
Marion\Documents\ArcGIS\Projects\fna\fna.gdb\fireVeg", "SIMPLIFY", "Value",
"MULTIPLE_OUTER_PART", None)
```

28. For the remainder of analysis, our results will be generalized to the HUC12 watershed boundary level. LANDFIRE model output should not be interpreted at the pixel-to-pixel level (https://landfire.gov/faqprint.php).

29. First, split natural area patches (i.e. polygons) that straddle one or more HUC unit boundaries.

30. ####Determine Management Benefit Index
31.