

Term project draft

Kipling Klimas

2021-04-23

Contents

1	Introduction	5
2	Data description and database structure	7
3	Database creation	9
4	Database connection	15
5	Data visualization	17

Chapter 1

Introduction

Wildfires can have significant impacts on both ecosystems and human assets in forested environments.

In the fire-prone mixed conifer/aspen forests of the Intermountain West (IW) fuel reduction treatments including stand thinning, mastication, aspen promotion and prescribed fire are implemented to reduce the risk, incidence and/or extent of destructive, high severity wildfire. Due to the high value of forested resources in the IW and the risk of post-fire sediment transport or debris flows downstream from burned areas, fuel reduction treatments are important landscape-scale tools that forest managers use to protect forest assets. However, the complex and pyrophytic nature of IW forests, along environmental and societal constraints, lends to barriers against spatially targeted implementation of fuel reduction treatments in high-value forested watersheds (Larson & Churchill, 2012).

Modeling the effect of fuel reduction treatments on fire behavior in fire-prone forests of the western United States has shown that targeted fuel reduction treatments are more effective at altering fire behavior (i.e. rate of spread, flame length) than randomly or indiscriminately applied treatments (Kim et al., 2009; Minas et al., 2014; Prichard et al., 2020). The spatial distribution of treatments must then follow a prioritization hierarchy, such as concentrated, dispersed or regularly arranged treated areas within a forest (Kim et al., 2009), but the economic trade-off and efficacy of treatment distribution warrants greater examination in IW forests, where fuel treatments are applied to reduce the risk of high severity wildfire (Barros et al., 2019). By altering fire behavior through landscape-scale treatments, forest managers hope to reduce the extent of high-severity burned areas and mitigate downstream consequences of high-severity wildfire, such as the development of debris flows which can have negative impacts on high value assets such as reservoirs or fish habitat.

Chapter 2

Data description and database structure

Forest structure and fuels data are needed to model fire behavior and predict burn severity spatially.

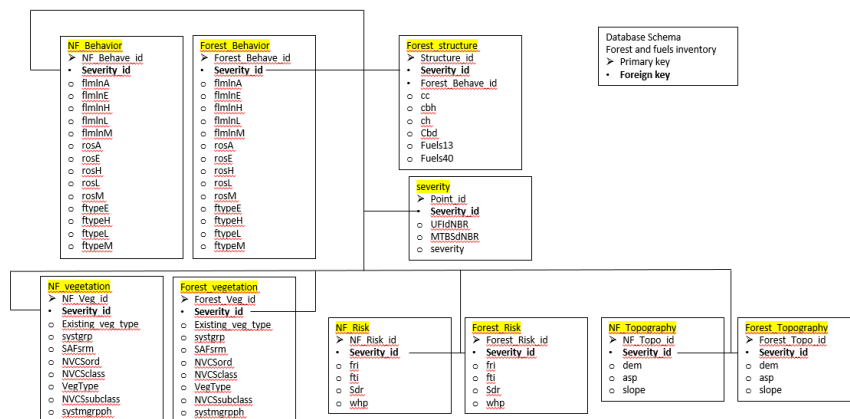


Figure 1: database structure for forest and fuels inventory.

This database structure is compatible with tree and stand level forest simulation software such as the Forest Vegetation Simulator (FVS) which can be used to model forest structure over time. An advantage of this approach and structure is these data allow detailed simulation of mortality during fire.

Tree level data are related to plot polygons that allow for stand-level simulation. Fuels data is required for modeled fire behavior, such as flame length, rate or spread or fire type (surface, passive or active).

Whether or not these tables become populated remains to be seen- a disadvantage of this approach is that fuel reduction treatments are not readily applied to the landscape in question. It is also only useable for fire behavior modeling as an ArcMap toolbar extension.

Chapter 3

Database creation

Start

Install and open packages RSQLite

```
install.packages("RSQLite")  
library(RSQLite)
```

Establish database connection

Connecting to the term_project database

```
term_project <- dbConnect(drv = RSQLite::SQLite(),  
                          "C:/Users/A02345823/Box/Lab_Group/Kipling/Classes/WILD6900_EcoRepSci/Term
```

Create severity table Term project database creation based on structure in figure 1

```
dbExecute(term_project, "CREATE TABLE severity(  
severity_id integer NOT NULL PRIMARY KEY,  
MTBSdNBR float,  
severity float  
);")
```

Create forest behavior table

```
dbExecute(term_project, "CREATE TABLE forest_behavior(  
forest_behave_id integer NOT NULL PRIMARY KEY,  
severity_id float,  
flmlnA float,
```

```

flmlnE float,
flmlnH float,
flmlnL float,
flmlnM float,
ftype_E float,
ftype_H float,
ftype_L float,
ftype_M float,
ros float,
rosE float,
rosH float,
rosL float,
rosM float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")

```

Create forest risk table

```

dbExecute(term_project, "CREATE TABLE forest_risk(
forest_risk_id integer NOT NULL PRIMARY KEY,
severity_id float,
fri float,
fti float,
sdr varchar(12),
WHP float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")

```

Create forest_structure table

```

dbExecute(term_project, "CREATE TABLE forest_structure(
str_id integer NOT NULL PRIMARY KEY,
severity_id float,
wwacbd float,
wwacbh float,
wwacc float,
wwach float,
wwafuel13 float,
wwafuel40 float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")

```

Create forest_topo table

```
dbExecute(term_project, "CREATE TABLE forest_topo(
forest_topo_id integer NOT NULL PRIMARY KEY,
severity_id float,
asp varchar(5),
dem float,
slope float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")
```

Create forest_veg table

```
dbExecute(term_project, "CREATE TABLE forest_veg(
forest_veg_id integer NOT NULL PRIMARY KEY,
severity_id float,
EVT varchar(60),
SystGrp varchar(60),
SAFSrm varchar(60),
NVCSOrd varchar(16),
NVCSclass varchar(25),
NVCSsubclass varchar(40),
SYSTEMGRPPH varchar(30),
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")
```

Create non-forest db tables

Create non-forest behavior table

```
dbExecute(term_project, "CREATE TABLE nf_behavior(
nf_behave_id integer NOT NULL PRIMARY KEY,
severity_id float,
flmlnA float,
flmlnE float,
flmlnH float,
flmlnL float,
flmlnM float,
ftype_E float,
ftype_H float,
ftype_L float,
ftype_M float,
ros float,
rosE float,
rosH float,
rosL float,
rosM float,
```

```
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")
```

Create non-forest risk table

```
dbExecute(term_project, "CREATE TABLE nf_risk(
nf_risk_id integer NOT NULL PRIMARY KEY,
severity_id float,
fri float,
fti float,
sdr varchar(12),
WHP float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")
```

Create non-forest_structure table

```
dbExecute(term_project, "CREATE TABLE nf_structure(
nf_str_id integer NOT NULL PRIMARY KEY,
severity_id float,
wwacbd float,
wwacbh float,
wwacc float,
wwach float,
wwafuel13 float,
wwafuel40 float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")
```

Create non-forest_topo table

```
dbExecute(term_project, "CREATE TABLE nf_topo(
nf_topo_id integer NOT NULL PRIMARY KEY,
severity_id float,
asp varchar(5),
dem float,
slope float,
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)
);")
```

Create non-forest_veg table

```
dbExecute(term_project, "CREATE TABLE nf_veg(  
nf_veg_id integer NOT NULL PRIMARY KEY,  
severity_id float,  
EVT varchar(60),  
SystGrp varchar(60),  
SAFSrm varchar(60),  
NVCSOrd varchar(16),  
NVCSclass varchar(25),  
NVCSsubclass varchar(40),  
SYSTMGRPPH varchar(30),  
FOREIGN KEY(severity_id) REFERENCES severity(severity_id)  
);")
```


Chapter 4

Database connection

Start

Install package RSQLite

```
install.packages("RSQLite")  
library(RSQLite)
```

Establish database connection and create tables

```
severity_db <- dbConnect(drv = RSQLite::SQLite(),  
                        "/Users/kiplingklimas/Box/Lab_Group/Kipling/Classes/WILD6900_EcoRepSci/Term_proje  
  
severity <- dbGetQuery(severity_db, "SELECT * FROM severity;")  
forest_behavior <- dbGetQuery(severity_db, "SELECT * FROM forest_behavior;")  
forest_risk <- dbGetQuery(severity_db, "SELECT * FROM forest_risk;")  
forest_structure <- dbGetQuery(severity_db, "SELECT * FROM forest_structure;")  
forest_topo <- dbGetQuery(severity_db, "SELECT * FROM forest_topo;")  
forest_veg <- dbGetQuery(severity_db, "SELECT * FROM forest_veg;")  
nf_behavior <- dbGetQuery(severity_db, "SELECT * FROM nf_behavior;")  
nf_risk <- dbGetQuery(severity_db, "SELECT * FROM nf_risk;")  
nf_structure <- dbGetQuery(severity_db, "SELECT * FROM nf_structure;")  
nf_topo <- dbGetQuery(severity_db, "SELECT * FROM nf_topo;")  
nf_veg <- dbGetQuery(severity_db, "SELECT * FROM nf_veg;")
```


Chapter 5

Data visualization

Install packages `tidyverse` and `viridis`

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.3      v purrr   0.3.4
## v tibble  3.1.1      v dplyr   1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1
```

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

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RSQLite)
library(DBI)
```

Connect to database ‘severity’ and create tables

```

severity_db <- dbConnect(drv = RSQLite::SQLite(),
                        "/Users/kiplingklimas/Box/Lab_Group/Kipling/Classes/WILD6900_EcoRepSci,

severity <- dbGetQuery(severity_db, "SELECT * FROM severity;")
forest_behavior <- dbGetQuery(severity_db, "SELECT * FROM forest_behavior;")
forest_risk <- dbGetQuery(severity_db, "SELECT * FROM forest_risk;")
forest_structure <- dbGetQuery(severity_db, "SELECT * FROM forest_structure;")
forest_topo <- dbGetQuery(severity_db, "SELECT * FROM forest_topo;")
forest_veg <- dbGetQuery(severity_db, "SELECT * FROM forest_veg;")
nf_behavior <- dbGetQuery(severity_db, "SELECT * FROM nf_behavior;")
nf_risk <- dbGetQuery(severity_db, "SELECT * FROM nf_risk;")
nf_structure <- dbGetQuery(severity_db, "SELECT * FROM nf_structure;")
nf_topo <- dbGetQuery(severity_db, "SELECT * FROM nf_topo;")
nf_veg <- dbGetQuery(severity_db, "SELECT * FROM nf_veg;")

```

This fire severity dataset shows the relationship between severity measurements and a variety of predictive variables including topographic, vegetation, structural and derived risk indices. Data was differentiated between forest and non-forest points to compare severity relationships between forests and non-forest landcover systems.

```

forest_topo %>%
  left_join(severity, by = "severity_id") %>%
  select(dem, severity) %>%
  ggplot(aes(x = dem, y = severity)) +
  geom_point()

```

