KEEP Final Report Appendix C: KEEP Stormwater Modeling Technical Document

Katie Ferran

April 24, 2020

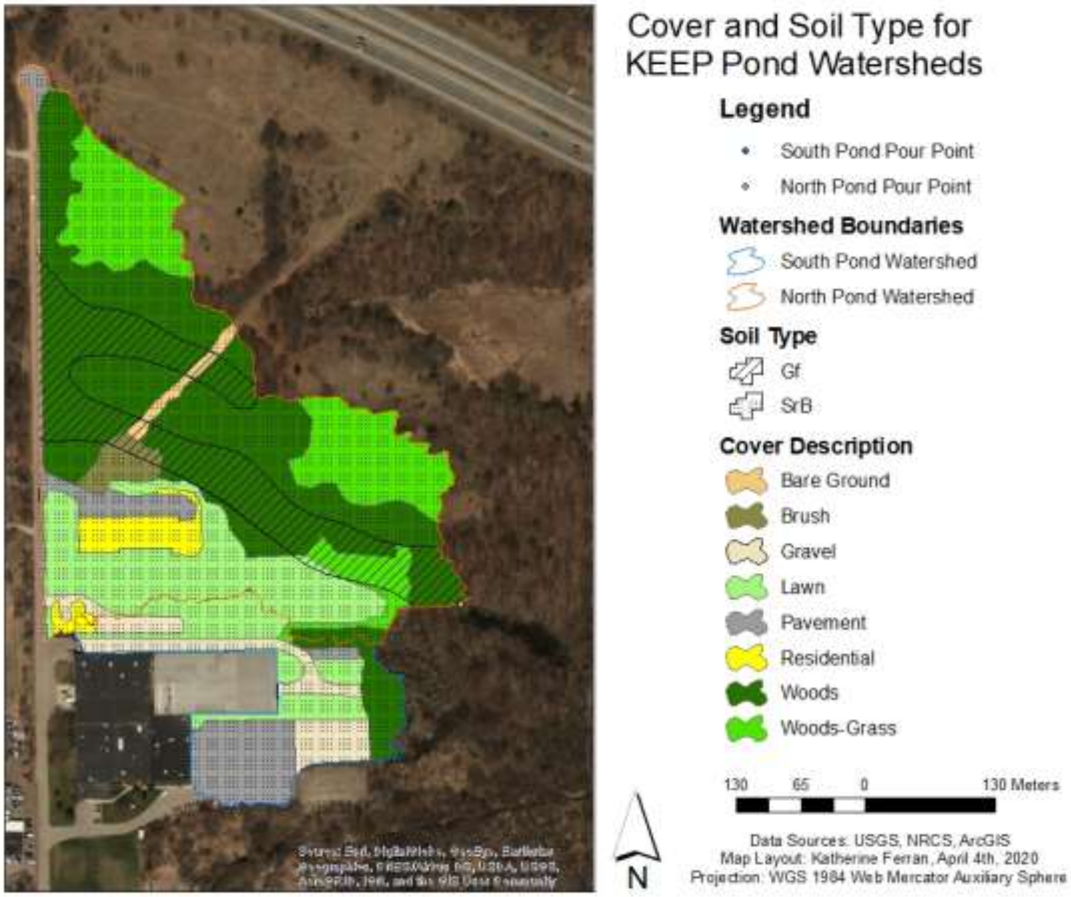
```
knitr::opts_chunk$set(echo = TRUE)
setwd("~/R")
library("ggplot2",lib.loc="~/R/win-library/3.5")
library("lubridate",lib.loc="~/R/win-library/3.5")

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

library("readr",lib.loc="~/R/win-library/3.5")
library("plotrix",lib.loc="~/R/win-library/3.5")

## Warning: package 'plotrix' was built under R version 3.5.3

library("dplyr",lib.loc="~/R/win-library/3.5")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

To set up, be sure to have the above packages installed and your working directory set to somewhere you can easily access and download data into.

The purpose of this script is to provide support to our recommendation for the installation of two stormwater collection systems on the Kiwanis of Ann Arbor property, for the purposes of improving water quality and drainage of the site’s water features. First, we establish a baseline of stormwater inflow into the property’s two ponds (North Pond and Sourth Pond) via watershed analysis for the year 2015. We will hereafter refer to this step of the analysis as the “Status Quo Inflow,” assuming that rainfall patterns of 2015 are typical for the area. Second, we simulate the altered inflow into North and South Pond were our recommended collection systems installed and operational for the year 2015. This step, we will refert to as the “Recommended Modified Inflow”.

Estimating Status Quo Inflow

To estimate the likely volume of inflow into the North and South Ponds in the year 2015, we must first import the 2015 Jackson Rd rain gauge dataset and the results of our spatial analysis of the two watersheds. This analysis yielded the area of the two pond’s watersheds,and the cover type and soil type of the areas, or sub-basins within.



Data Sources: USGS, NRCS, ArcGIS
Map Layout: Katherine Ferran, April 4th, 2020
Projection: WGS 1984 Web Mercator Auxiliary Sphere

Each of these sub-basins are further described in the following tables.

```
setwd("~/R/KEEPmodelfiles")
kw.dat.q        = read.csv("JacksonRoadRainGaugeData2015Q.csv", header = T)
kw.dat.q[,8]    = as.numeric(strptime(kw.dat.q[,1], format = "%m/%d/%Y 
%H:%M"))


HM.Landcover<- read_csv("gis.analysis.NP.csv")
```

```
## Parsed with column specification:
## cols(
##   OBJECTID = col_integer(),
##   `Watershed Code` = col_integer(),
##   MUSYM = col_character(),
##   Rating = col_character(),
##   Type = col_character(),
##   `Curve Number` = col_integer(),
##   `S value` = col_double(),
##   Shape_Length = col_double(),
##   Shape_Area = col_double()
## )
```

```
BS.Landcover<-read_csv("gis.analysis.SP.csv")
```

```
## Parsed with column specification:
## cols(
##   OBJECTID = col_integer(),
##   `Watershed Code` = col_integer(),
##   MUSYM = col_character(),
##   Rating = col_character(),
##   Type = col_character(),
##   `Curve Number` = col_integer(),
##   `S value` = col_double(),
##   Shape_Length = col_double(),
##   Shape_Area = col_double()
## )
```

```
HM.area<-sum(HM.Landcover$Shape_Area)
BS.area<-sum(BS.Landcover$Shape_Area)
total.area<-HM.area+BS.area
cubintogal  <- 0.004329
print(HM.Landcover)
```

```
## # A tibble: 24 x 9
##    OBJECTID `Watershed Code` MUSYM Rating Type  `Curve Number` `S value`
##       <int>            <int> <chr> <chr>  <chr>          <int>     <dbl>
##  1        2                1 SrB   A      Pave~             98     0.204
##  2       10                1 SrB   A      Brush             35    18.6
##  3       11                1 SrB   A      Grav~             76     3.16
##  4       12                1 SrB   A      Lawn              39    15.6
##  5       13                1 SrB   A      Pave~             98     0.204
##  6       14                1 SrB   A      Resi~             77     2.99
##  7       15                1 SrB   A      Woods             36    17.8
##  8       16                1 SrB   A      Wood~             43    13.3
##  9       17                1 SrB   A      Bare~             68     4.71
## 10       18                1 SrB   A      Pave~             98     0.204
## # ... with 14 more rows, and 2 more variables: Shape_Length <dbl>,
## #   Shape_Area <dbl>
```

```
print(BS.Landcover)
```

```
## # A tibble: 18 x 9
##    OBJECTID `Watershed Code` MUSYM Rating Type  `Curve Number` `S value`
##       <int>            <int> <chr> <chr>  <chr>          <int>     <dbl>
##  1        1                0 SrB   A      Pave~             98     0.204
##  2        3                0 SrB   A      Grav~             76     3.16 
##  3        4                0 SrB   A      Lawn              39    15.6  
##  4        5                0 SrB   A      Pave~             98     0.204
##  5        6                0 SrB   A      Resi~             77     2.99 
##  6        7                0 SrB   A      Woods             36    17.8  
##  7        8                0 SrB   A      Wood~             43    13.3  
##  8        9                0 Gf    D      Woods             79     2.66 
##  9       27                0 SrB   A      Grav~             76     3.16 
## 10       29                0 SrB   A      Lawn              39    15.6  
## 11       31                0 SrB   A      Resi~             77     2.99 
## 12       33                0 SrB   A      Woods             36    17.8  
## 13       35                0 SrB   A      Wood~             43    13.3  
## 14       37                0 SrB   A      Grav~             76     3.16 
## 15       38                0 SrB   A      Woods             36    17.8  
## 16       39                0 SrB   A      Lawn              39    15.6  
## 17       40                0 SrB   A      Pave~             98     0.204
## 18       41                0 SrB   A      Lawn              39    15.6  
## # ... with 2 more variables: Shape_Length <dbl>, Shape_Area <dbl>
```

As you can see in each of these tables, each sub-basin was assigned a curve number the following the guidelines outlined in these tables from the USDA NRCS Conservation Engineering Division methodology:

Cover description		Curve numbers for hydrologic soil group			
Cover type and hydrologic condition	Average percent impervious area [2]	A	B	C	D
Fully developed urban areas (vegetation established)					
Open space (lawns, parks, golf courses, cemeteries, etc.) [3]:					
Poor condition (grass cover < 50%)		68	79	86	89
Fair condition (grass cover 50% to 75%)		49	69	79	84
Good condition (grass cover > 75%)		39	61	74	80
Impervious areas:					
Paved parking lots, roofs, driveways, etc. (excluding right-of-way)		98	98	98	98
Streets and roads:					
Paved; curbs and storm sewers (excluding right-of-way)		98	98	98	98
Paved; open ditches (including right-of-way)		83	89	92	93
Gravel (including right-of-way)		76	85	89	91
Dirt (including right-of-way)		72	82	87	89
Western desert urban areas:					
Natural desert landscaping (pervious areas only) [4]		63	77	85	88
Artificial desert landscaping (impervious weed barrier, desert shrub with 1- to 2-inch sand or gravel mulch and basin borders)		96	96	96	96
Urban districts:					
Commercial and business	85	89	92	94	95
Industrial	72	81	88	91	93
Residential districts by average lot size:					
1/8 acre or less (town houses)	65	77	85	90	92
1/4 acre	38	61	75	83	87
1/3 acre	30	57	72	81	86
1/2 acre	25	54	70	80	85
1 acre	20	51	68	79	84
2 acres	12	46	65	77	82
Developing urban areas					
Newly graded areas (pervious areas only, no vegetation) [5]		77	86	91	94
Idle lands (CN's are determined using cover types similar to those in table 2-2c).					

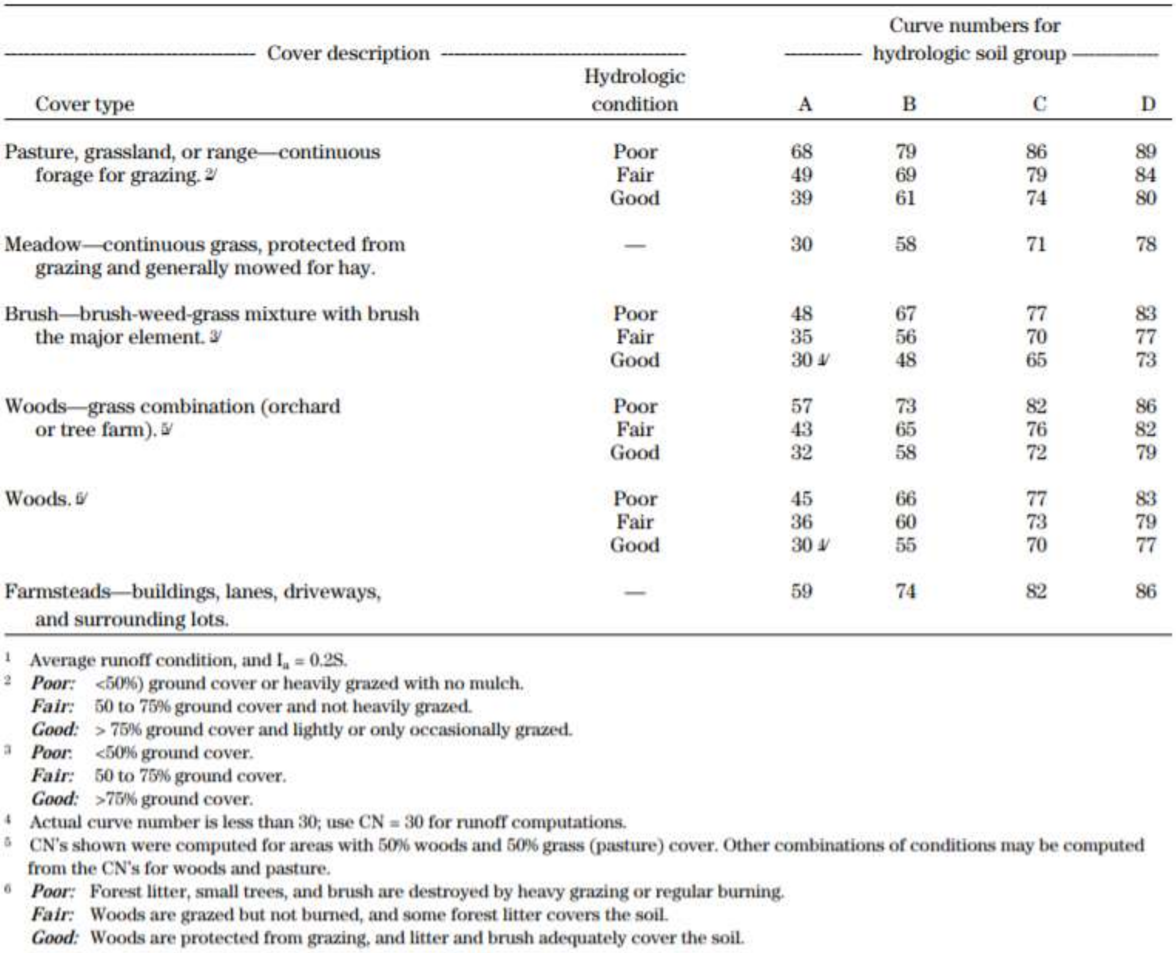
[1] Average runoff condition, and $I_a = 0.2S$.

[2] The average percent impervious area shown was used to develop the composite CN's. Other assumptions are as follows: impervious areas are directly connected to the drainage system, impervious areas have a CN of 98, and pervious areas are considered equivalent to open space in good hydrologic condition. CN's for other combinations of conditions may be computed using figure 2-3 or 2-4.

[3] CN's shown are equivalent to those of pasture. Composite CN's may be computed for other combinations of open space cover type.

[4] Composite CN's for natural desert landscaping should be computed using figures 2-3 or 2-4 based on the impervious area percentage (CN = 98) and the pervious area CN. The pervious area CN's are assumed equivalent to desert shrub in poor hydrologic condition.

[5] Composite CN's to use for the design of temporary measures during grading and construction should be computed using figure 2-3 or 2-4 based on the degree of development (impervious area percentage) and the CN's for the newly graded pervious areas.



Cover description		Curve numbers for hydrologic soil group			
Cover type	Hydrologic condition	A	B	C	D
Pasture, grassland, or range—continuous forage for grazing. [2]	Poor	68	79	86	89
	Fair	49	69	79	84
	Good	39	61	74	80
Meadow—continuous grass, protected from grazing and generally mowed for hay.	—	30	58	71	78
Brush—brush-weed-grass mixture with brush the major element. [3]	Poor	48	67	77	83
	Fair	35	56	70	77
	Good	30 [4]	48	65	73
Woods—grass combination (orchard or tree farm). [5]	Poor	57	73	82	86
	Fair	43	65	76	82
	Good	32	58	72	79
Woods. [6]	Poor	45	66	77	83
	Fair	36	60	73	79
	Good	30 [4]	55	70	77
Farmsteads—buildings, lanes, driveways, and surrounding lots.	—	59	74	82	86

[1] Average runoff condition, and I_a = 0.2S.

[2] *Poor:* <50%) ground cover or heavily grazed with no mulch.
Fair: 50 to 75% ground cover and not heavily grazed.
Good: > 75% ground cover and lightly or only occasionally grazed.

[3] *Poor:* <50% ground cover.
Fair: 50 to 75% ground cover.
Good: >75% ground cover.

[4] Actual curve number is less than 30; use CN = 30 for runoff computations.

[5] CN's shown were computed for areas with 50% woods and 50% grass (pasture) cover. Other combinations of conditions may be computed from the CN's for woods and pasture.

[6] *Poor:* Forest litter, small trees, and brush are destroyed by heavy grazing or regular burning.
Fair: Woods are grazed but not burned, and some forest litter covers the soil.
Good: Woods are protected from grazing, and litter and brush adequately cover the soil.

Runoff in inches (Q) from a surface type can be described as $Q = \frac{(P-.2S)^2}{P+.8S}$ where P is rainfall in inches. Each surface type needs to be assigned an S value, which represents potential maximum retention after runoff begins (in). The formula is $S = \frac{100}{CN-10}$, where CN is the "curve number". These formulas are used on each sub-basin for each rainfall reading in the following loop. Unfortunately, the CN method was developed to estimate runoff based on volumes typical of a 24-hour period, and volumes at the size of our 15-minute rainfall data deliver inaccurate results. One symptom of this inaccuracy around small values is that the function for Q will deliver positive values when rainfall is 0 inches.Our fix is to take our quarter-hour rainfall data and multiply it by 96 (the number of quarter-hours in a day), and then divide the resulting Q values by 96. This method is based on assumptions that may not be sound, and could generate uncertainty.

```
for(i in 1:length(kw.dat.q$Rainfall)){
  for(x in 1:length(HM.Landcover$`S value`)){
    S<-HM.Landcover$`S value`[x]
    Area<-HM.Landcover$Shape_Area[x]*1550
    HM.Landcover$receive[x]<-(((((kw.dat.q$Rainfall[i]*96)-
.2*S)^2)/((kw.dat.q$Rainfall[i]*96)+.8*S))/96)*Area*cubintogal
    if(kw.dat.q$Rainfall[i]==0){
      HM.Landcover$receive[x]<-0
    }
  }
  kw.dat.q$Hm.receive[i]<-sum(HM.Landcover$receive)
  for(y in 1:length(BS.Landcover$`S value`)){
    S<-BS.Landcover$`S value`[y]
    Area<-BS.Landcover$Shape_Area[y]*1550
    BS.Landcover$receive[y]<-(((((kw.dat.q$Rainfall[i]*96)-
.2*S)^2)/((kw.dat.q$Rainfall[i]*96)+.8*S))/96)*Area*cubintogal
    if(kw.dat.q$Rainfall[i]==0){
      BS.Landcover$receive[y]<-0
    }
  }
  kw.dat.q$BS.receive[i]<-sum(BS.Landcover$receive)
}

kw.dat.q<-mutate(kw.dat.q,Date=as.Date(kw.dat.q$Reading.Date,format="%m/%d/%Y
%H:%M","%m/%d/%Y"))

NPreceive<-round(sum(kw.dat.q$Hm.receive),2)
SPreceive<-round(sum(kw.dat.q$BS.receive),2)
```

From this analysis, we estimate that the North Pond received 5.13 million gallons in 2015, and the South Pond received 1.78 million.

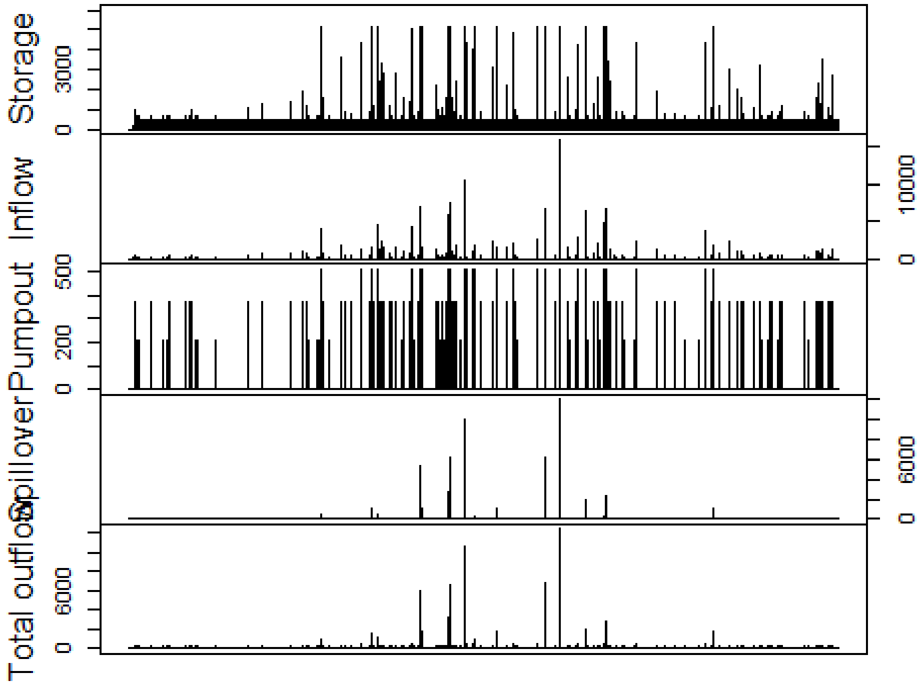
Estimating Recommended Modified Inflow

Now to estimate the possible volume of stormwater that could be collected and redirected via stormwater collection system installation. Again, we are recommending the installation of two systems: one to collect stormwater from the warehouse rooftop, and one to collect runoff that pools in the grafel parking lot. We begin with the rooftop system. This roof is approximately 44933 sq ft, but only 75% of its area will be able to be redirected, according to the clients. By multiplying rainfall by this area, we will obtain runoff in cubic inches, which we will then convert into gallons. Next, we must set the parameters of our theoretical tank and pump system. Using some equipment marketed on https://www.rainharvest.com/shop/, we start with a 5100 gallon tank, which if aboveground is about a story tall. We also assume it has a pump that can pump 25 gallons per minute (gpm) on average, and 34 gpm at most. The following loop stipulates that the average pump rate be used when the tank is more than 10% full, and to shut the pump off should the volume in the tank sink below this value. It also stipulates that the pump increase to it's faster pump rate should the tank become more than 75% full. Inflow, outflow, and overflow are recorded with each run of the loop.

```
roof.model<-
data.frame("startvol"=NA,"inflow"=NA,"pumpout"=NA,"grossvol"=NA,"spillover"=N
A,"endvol"=NA)
vol<-5100
fastrate<-34 #This is the fastest pump rate of the Rainflo Flow Inducer, in
gallons per quarter hour
```

```
medrate<-25 #This is the average pump rate of the Rainflo Flow Inducer, in
gallons per quarter hour
roofareasqin<-6470352*.75 #This is the area of the roof in square inches

for(i in 1:length(kw.dat.q$Rainfall)){
  if(i == 1){roof.model[i,1] = 0} #this stipulates that the model start at a
volume of zero
  if(i > 1){roof.model[i,1] = roof.model[i-1,6]} #this stipulates that the
model use the previous value for vol in tank is used
  if(roof.model[i,1] <= vol*0.10)   {pump.rate = 0}         # Column 1:
initial volume
  if(   roof.model[i,1] > vol*0.10 &
      roof.model[i,1] <= vol*0.75){pump.rate = medrate}
  if(roof.model[i,1] > vol*0.75)    {pump.rate = fastrate}
  roof.model[i,2] = kw.dat.q[i,2]*roofareasqin*cubintogal       # Column 2:
inflow
  roof.model[i,3] = pump.rate*15                            # Column 3:
pumpout. Make pumpout rate match timestep!!
  roof.model[i,4] =     roof.model[i,1]+                    # Column 4: gross
volume
    roof.model[i,2]-
    roof.model[i,3]
  if(roof.model[i,4] <= vol*0.1){
    roof.model[i,3] = roof.model[i,3]-(vol*0.1-roof.model[i,4])
    roof.model[i,5] = 0
    roof.model[i,6] = vol*0.1}
  if(roof.model[i,3]<=0){
    roof.model[i,3]=0
    roof.model[i,6]=roof.model[i,4]
  }
  if(roof.model[i,4] > vol){
    roof.model[i,5] = roof.model[i,4] - vol     # Column 5: spillover
    roof.model[i,6] = vol}
  if(   roof.model[i,4] <= vol &
      roof.model[i,4] > vol*0.1){
    roof.model[i,5] = 0
    roof.model[i,6] = roof.model[i,4]}          # Column 6: end volume
}


par(mfrow = c(5,1))
par(mar = c(0,0,0,0))
par(oma = c(4,4,1,4))


plot(roof.model[,1], axes = F, type = "h", ylim = c(0,6000)); box(); axis(2);
mtext("Storage", side = 2, line = 2.5)
plot(roof.model[,2], axes = F, type = "h"); box(); axis(4); mtext("Inflow",
side = 2, line = 2.5)
```

```
plot(roof.model[,3], axes = F, type = "h"); box(); axis(2); mtext("Pumpout", 
side = 2, line = 2.5)
plot(roof.model[,5], axes = F, type = "h"); box(); axis(4); 
mtext("Spillover", side = 2, line = 2.5)
plot(roof.model[,5]+roof.model[,3], axes = F, type = "h"); box(); axis(2); 
mtext("Total outflow", side = 2, line = 2.5)
```

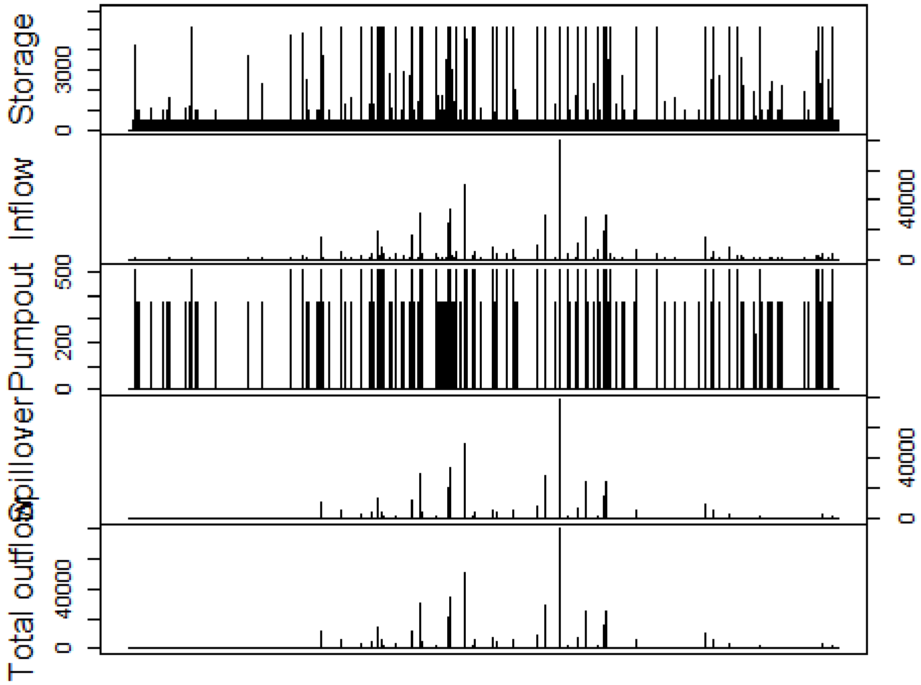


```
roofspillover<-round(sum(roof.model$spillover),2)
roofpumpout<-round(sum(roof.model$pumpout),2)
```

From this, we see some overflow taking place, about 104 thousand gallons. Total volume of water pumped from the system was about 543 thousand gallons for the year of 2015.

Next, we now modify the watershed of the South Pond to include the overflow from the roof model. We again assume a 5100 gallon tank that we recommend be installed underground at the South Pond collection point on the eastern edge of the gravel parking lot, and a pump that can pump 25 gallons per minute (gpm) on average, and 34 gpm at most.

```
lot.model<-
data.frame("startvol"=NA,"inflow"=NA,"pumpout"=NA,"grossvol"=NA,"spillover"=N
A,"endvol"=NA)
vol<-5100
for(i in 1:length(kw.dat.q$Rainfall)){
  if(i == 1){lot.model[i,1] = 0} #this stipulates that the model start at a 
volume of zero
  if(i > 1){lot.model[i,1] = lot.model[i-1,6]} #this stipulates that the
model use the previous value for vol in tank is used
  if(lot.model[i,1] <= vol*0.10)    {pump.rate = 0}         # Column 1:
initial volume
  if(   lot.model[i,1] > vol*0.10 &
      lot.model[i,1] <= vol*0.75){pump.rate = medrate}
  if(lot.model[i,1] > vol*0.75) {pump.rate = fastrate}
  lot.model[i,2] = kw.dat.q$BS.receive[i]       # Column 2: inflow
  lot.model[i,3] = pump.rate*15                         # Column 3: pumpout.
Make pumpout rate match timestep!!
  lot.model[i,4] =  lot.model[i,1]+                 # Column 4: gross volume
    lot.model[i,2]-
    lot.model[i,3]
  if(lot.model[i,4] <= vol*0.1){
    lot.model[i,3] = lot.model[i,3]-(vol*0.1-lot.model[i,4])
    lot.model[i,5] = 0
    lot.model[i,6] = vol*0.1}
  if(lot.model[i,3]<=0){
    lot.model[i,3]=0
    lot.model[i,6]=lot.model[i,4]
  }
  if(lot.model[i,4] > vol){
    lot.model[i,5] = lot.model[i,4] - vol       # Column 5: spillover
    lot.model[i,6] = vol}
  if(   lot.model[i,4] <= vol &
      lot.model[i,4] > vol*0.1){
    lot.model[i,5] = 0
    lot.model[i,6] = lot.model[i,4]}            # Column 6: end volume
}

par(mfrow = c(5,1))
par(mar = c(0,0,0,0))
par(oma = c(4,4,1,4))


plot(lot.model[,1], axes = F, type = "h", ylim = c(0,6000)); box(); axis(2);
mtext("Storage", side = 2, line = 2.5)
plot(lot.model[,2], axes = F, type = "h"); box(); axis(4); mtext("Inflow",
side = 2, line = 2.5)
plot(lot.model[,3], axes = F, type = "h"); box(); axis(2); mtext("Pumpout",
side = 2, line = 2.5)
plot(lot.model[,5], axes = F, type = "h"); box(); axis(4); mtext("Spillover",
side = 2, line = 2.5)
plot(lot.model[,5]+lot.model[,3], axes = F, type = "h"); box(); axis(2);
mtext("Total outflow", side = 2, line = 2.5)
```



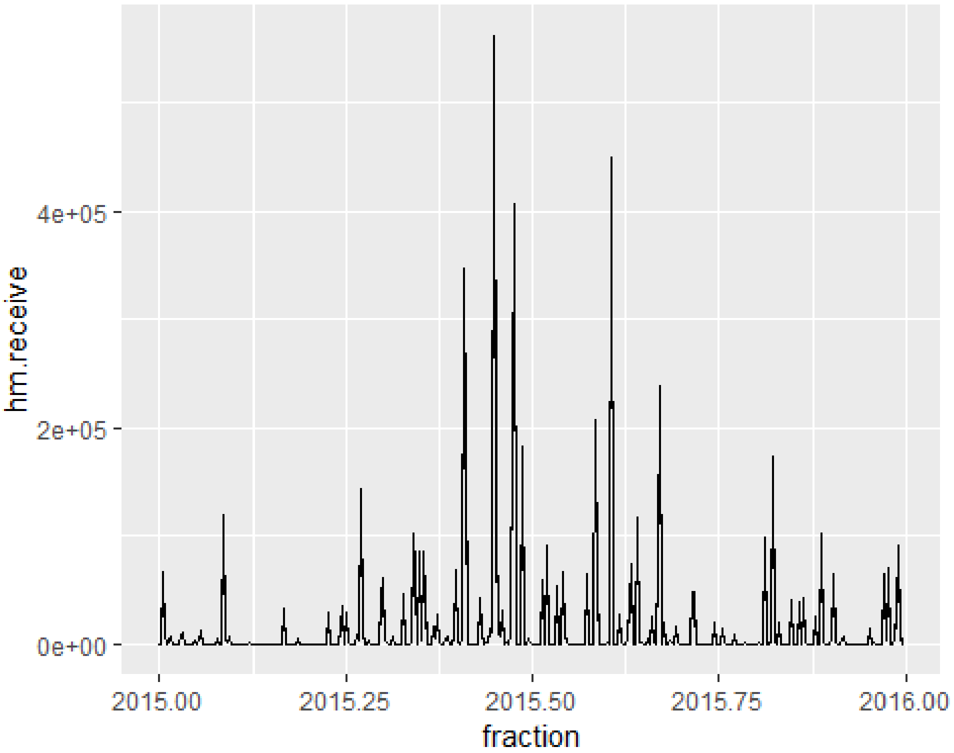
```
lotspillover<-round(sum(lot.model$spillover),2)
lotpumpout<-round(sum(lot.model$pumpout),2)
```

This system pumped out about about 850 thousand gallons, while about 933 thousand gallons were lost to overflow.

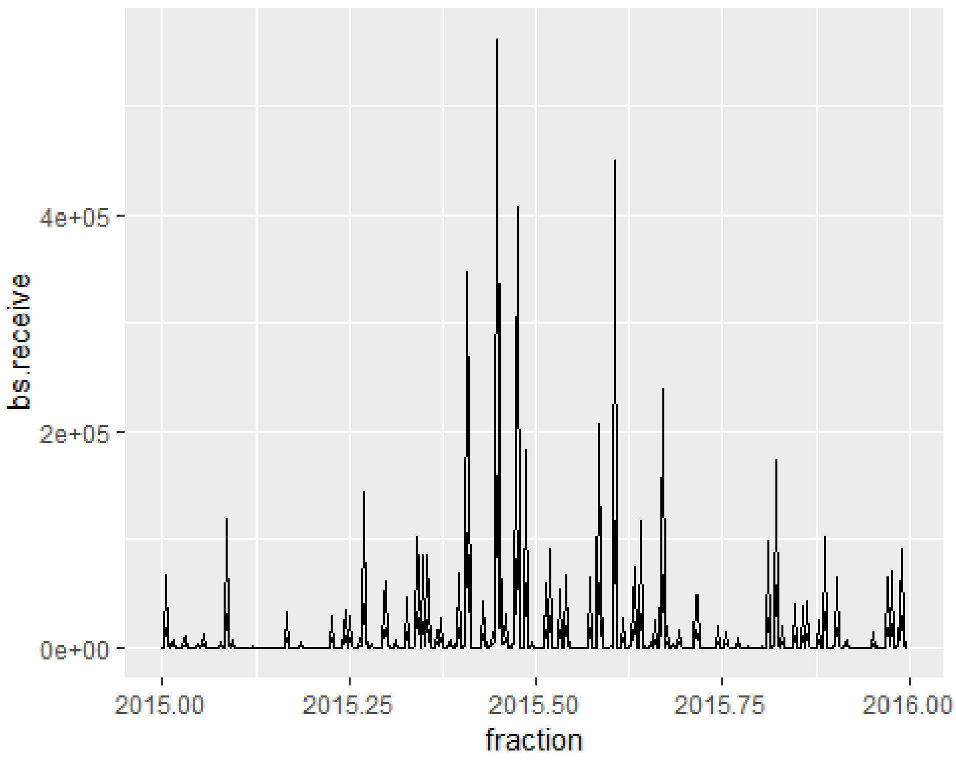
Visualizing results

Here, we summarize the status quo inflow into the two ponds, and their modified inflow if the rooftop collection system were to pump directly into the North Pond, and the parking lot collection system were to split its collection between the North and South Pond and direct its overflow directly into South Pond.

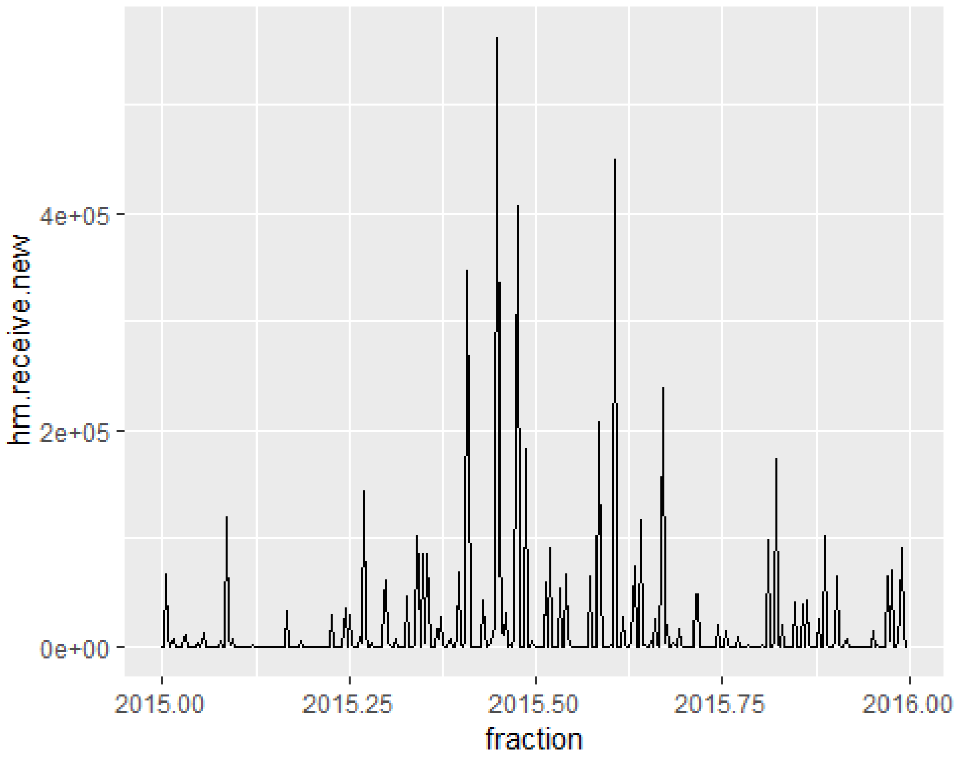
```
kw.dat.summary<-
data.frame("day"=as.Date(unique(kw.dat.q$Date),format="%m/%d/%Y","%m/%d/%Y"))
for (i in 1:length(kw.dat.summary$day)){
  date<-as.Date(kw.dat.summary$day[i],format="%m/%d/%Y","%m/%d/%Y")
  dayrain<-subset(kw.dat.q,kw.dat.q$Date==date)
  roofrain<-subset(roof.model$pumpout,roof.model$Date==date)
  lotrainp<-subset(lot.model$pumpout,lot.model$Date==date)
  lotraino<-subset(lot.model$spillover,lot.model$Date==date)
  kw.dat.summary$hm.receive[i]<-sum(dayrain$Hm.receive)
  kw.dat.summary$bs.receive[i]<-sum(dayrain$BS.receive)
  kw.dat.summary$hm.receive.new[i]<-
sum(roofrain)+sum(dayrain$Hm.receive)+(sum(lotrainp)/2)
  kw.dat.summary$bs.receive.new[i]<-sum(lotraino)+(sum(lotrainp)/2)
}
kw.dat.summary<-
mutate(kw.dat.summary,fraction=decimal_date(kw.dat.summary$day))
ggplot(data=kw.dat.summary)+geom_line(mapping=aes(x=fraction,y=hm.receive))+g
eom_line(mapping = aes(x=fraction,y=hm.receive.new))
```



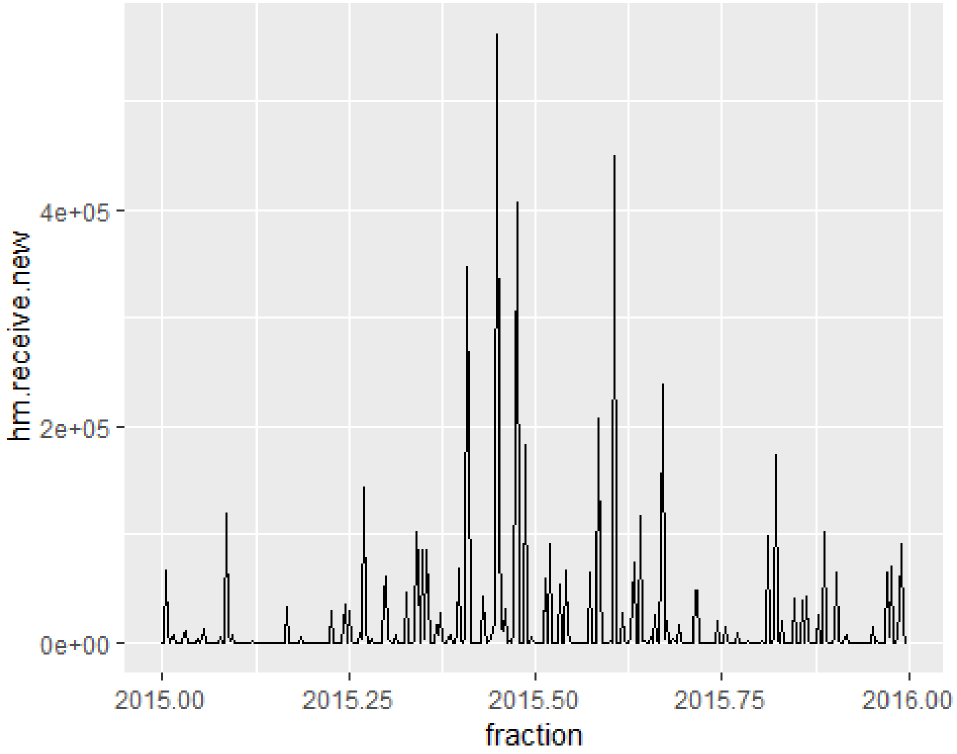
```
ggplot(data=kw.dat.summary)+geom_line(mapping=aes(x=fraction,y=bs.receive))+g
eom_line(mapping = aes(x=fraction,y=hm.receive.new))
```



```
ggplot(data=kw.dat.summary)+geom_line(mapping=aes(x=fraction,y=hm.receive.new
))+geom_line(mapping = aes(x=fraction,y=hm.receive.new))
```



```r
ggplot(data=kw.dat.summary)+geom_line(mapping=aes(x=fraction,y=hm.receive.new
))+geom_line(mapping = aes(x=fraction,y=hm.receive.new))
```



```r
NPreceivenew<-round(sum(kw.dat.summary$hm.receive.new),2)
SPreceivenew<-round(sum(kw.dat.summary$bs.receive.new),2)
```

If these tanks were to be installed, North Pond would have received 6.1 million gallons and South Pond 1.4 million gallons in the year 2015. We maintain that these systems will promote water quality in both ponds and improve drainage of the entire site.

Works Cited

City of Ann Arbor. Jackson Road Rain Gauge Data. https://www3.a2gov.org/RainGauges/

Soil Survey Staff, Natural Resources Conservation Service, United States Department of Agriculture. (2019). Web Soil Survey. https://websoilsurvey.sc.egov.usda.gov/

Urban Hydrology for Small Watersheds (No. 55; Technical Release, p. 1-164). (1986). United States Department of Agriculture Natural Resources Conservation Service Conservation Engineering Division.

U.S. Geological Survey. (2019). USGS NED one meter x26y469 MI 31Co Washtenaw 2016 IMG 2019: U.S. Geological Survey.