NASIS data to EDIT

Nathan Roe

2/10/2022

This is the final step in the ‘EDIT ecosite data’ workflow. It will produce reports for all the ecosites that you provide to the corresponding for_loop_file.R.

Note: You will need to change the file path on line 28. The easiest way to do that is to find the file that you created in Step 2, your NASIS ecosite report. It should be in the folder with all your other ‘EDIT ecosite data’ files. Shift + right click > copy as path. Then paste the path in, replacing the existing one. That path will have backslashes these all need to be changed to forwardslashes /.

Packages

```
library(plyr)
library(dplyr)
library(httr)
library(wordcloud)
library(wordcloud2)
library(pander)
```

Read in NASIS report

```
EDIT_updates <- read.csv("C:/Users/Nathan.Roe/Documents/PES4ever/NASIS_data_for_EDIT.csv")
```

Remove miscellaneous areas

```
EDIT_updates <- EDIT_updates %>% filter(compkind != "miscellaneous area")
```

Define ecosite of interest

```
EDIT_updates_filtered <- EDIT_updates %>% filter(ecosite_id == ecosites_of_interest)
```

Physiographic features tab:

Landform

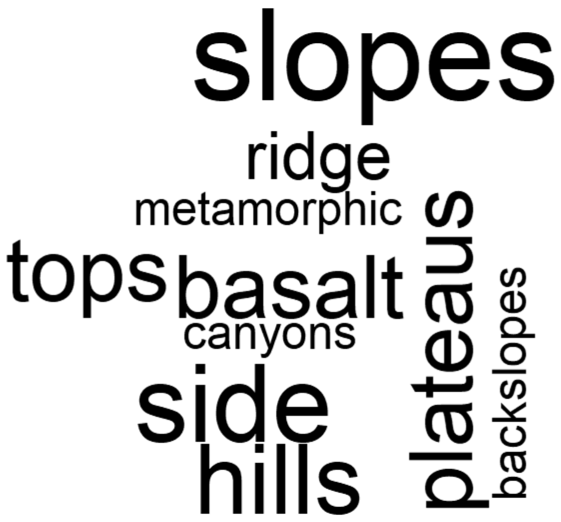
Add the most relevant landforms. Try to keep it to 3-4.

```
landform <- as.data.frame(table(EDIT_updates_filtered$landform)) %>%
  dplyr::rename(Landform = Var1) %>% arrange(desc(Freq)) %>% mutate_if(is.factor, as.character)

landform
```

```
##                                                 Landform Freq
## 1                                   Backslopes in canyons    3
## 2                           backslopes on basalt plateaus    2
## 3               benches and side slopes on basalt plateaus    2
## 4       Ridge tops and and footslopes on metamorphic hills    2
## 5                          side slopes on basalt plateaus    2
## 6                                     Backslopes on hills    1
## 7               Benches on side slopes on basalt plateaus    1
## 8                 In fractures on tops of basalt plateaus    1
## 9                                    nose slopes on hills    1
## 10     Ridge tops and shoulder slopes on volcanic ridges    1
## 11       Ridge tops and side slopes on metamorphic hills    1
## 12                      Ridge tops on metavolcanic hills    1
## 13                         Ridge tops on volcanic ridges    1
## 14                                Side slopes in canyons    1
## 15                        Side slopes on basalt plateaus    1
## 16                                  side slopes on hills    1
## 17 Side slopes on metamorphic and intrusive igneous hills    1
## 18                     Side slopes on metavolcanic hills    1
## 19                                  Toeslopes in canyons    1
## 20                               Tops of basalt plateaus    1
```

Landform words

```
landform_text <- paste(landform$Landform, collapse = " ")

words <- strsplit(landform_text, "[[:space:]]+")[[1]]

if(length(words) == 0){print("No landform descriptions")} else{wordcloud::wordcloud(words)
}
```



Runoff class

```
# This field is obsolete and will be removed from EDIT. Enter medium for all values.
```

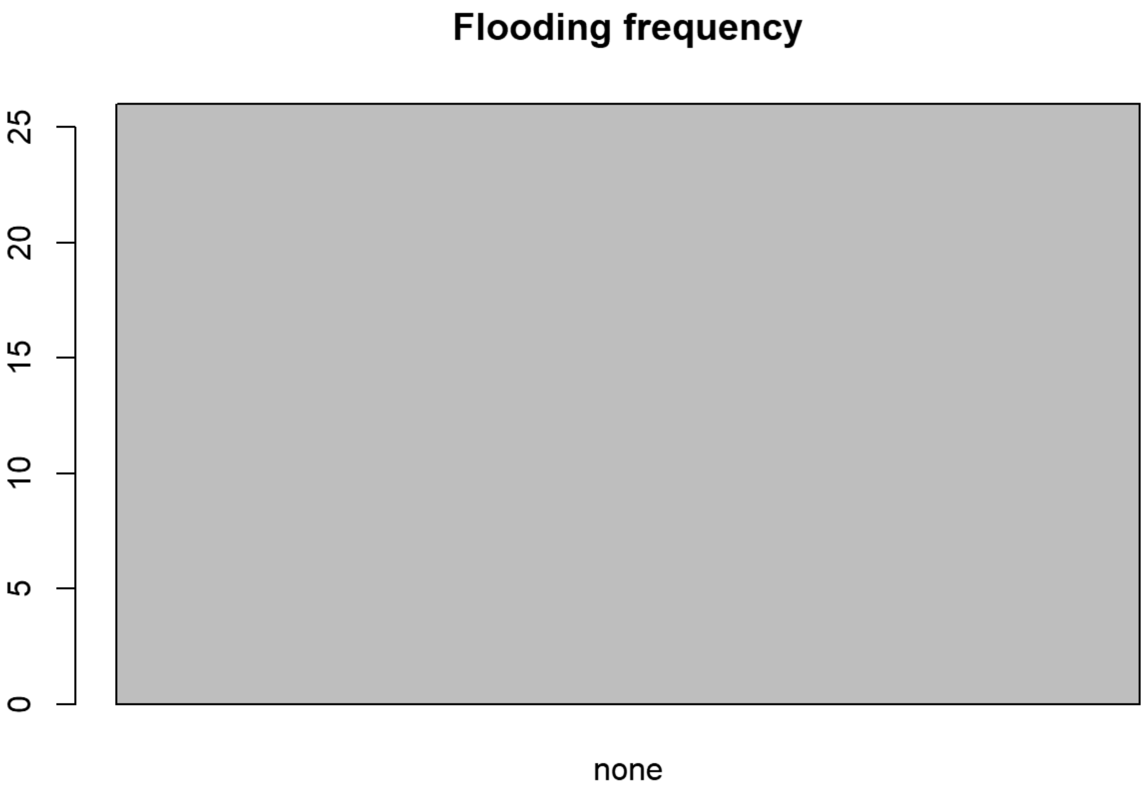
Flooding

Put “–” for duration if frequency is “none”

```
Flood.freq <- ifelse(EDIT_updates_filtered$flood_freq == "", "none", EDIT_updates_filtered$flood
_freq)

as.data.frame(table(Flood.freq)) %>%
  arrange(desc(Freq))
```

```
##   Flood.freq Freq
## 1       none   26
```

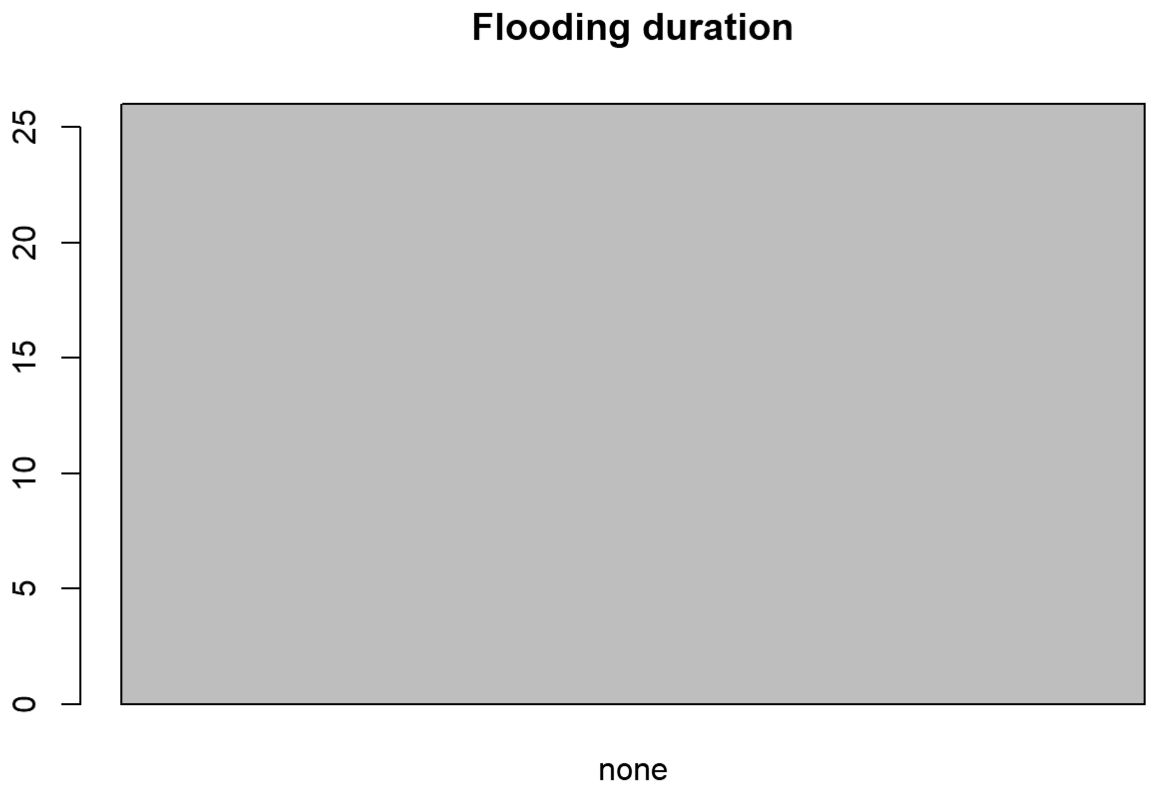
```
barplot(table(Flood.freq), main = "Flooding frequency")
```



```
Flood.duration <- ifelse(EDIT_updates_filtered$flood_dur == "", "none", EDIT_updates_filtered$fl
ood_dur)


as.data.frame(table(Flood.duration)) %>% arrange(desc(Freq))
```

```
##   Flood.duration Freq
## 1           none   26
```

```
barplot(table(Flood.duration), main = "Flooding duration")
```


Flooding duration

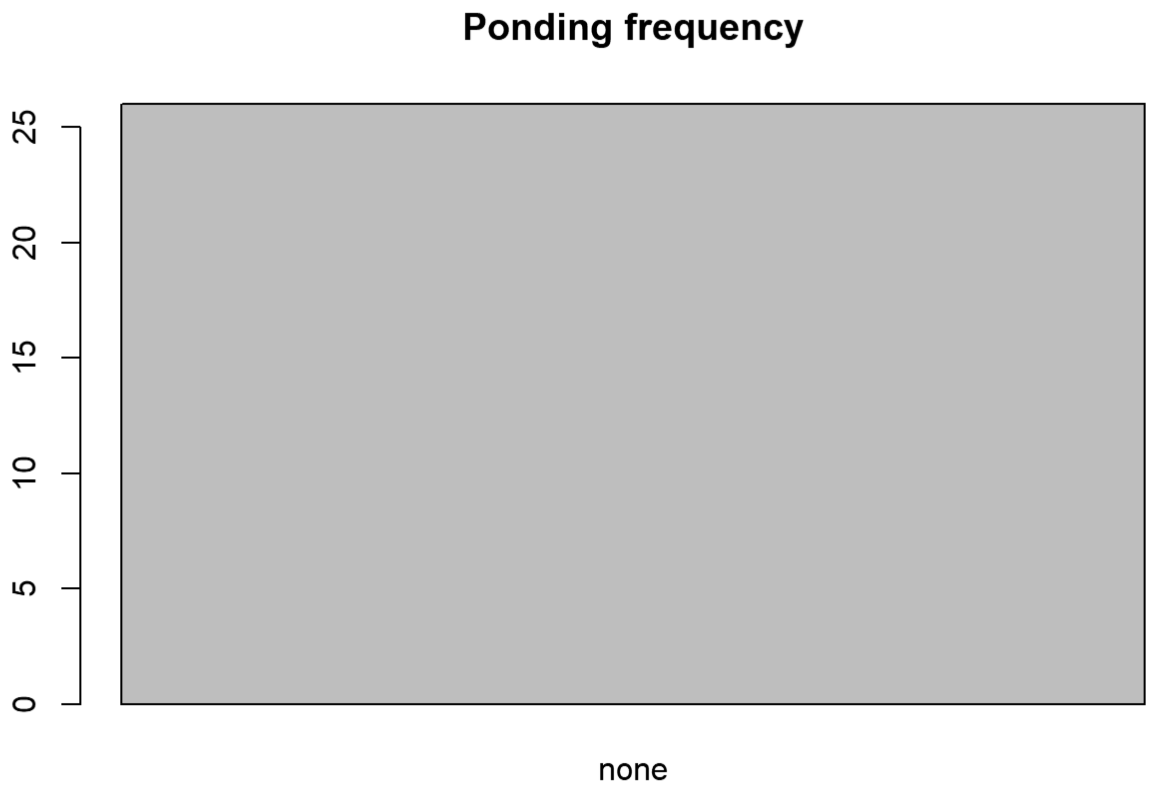
Ponding

Put “–” for duration if frequency is “none”

```
as.data.frame(table(EDIT_updates_filtered$pond_freq)) %>%
  rename("Ponding Freq" = Var1) %>% arrange(desc(Freq))
```

```
##   Ponding Freq Freq
## 1         none   26
```

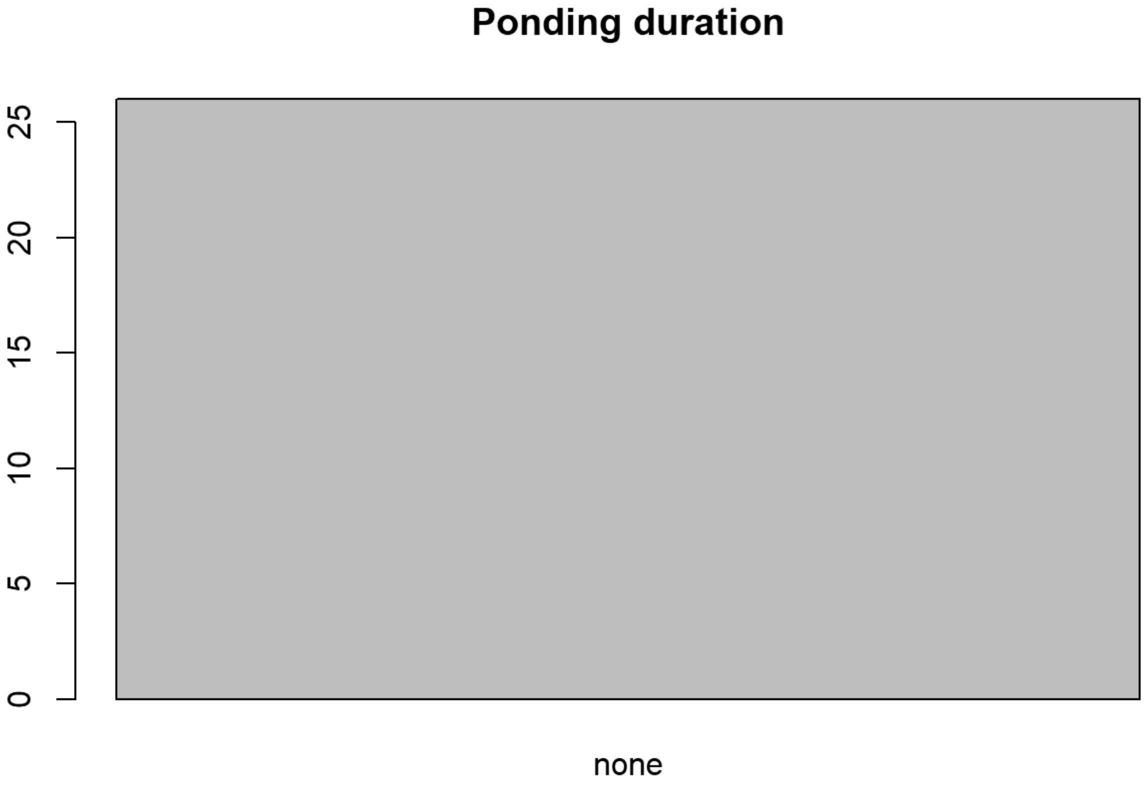
```
barplot(table(EDIT_updates_filtered$pond_freq), main = "Ponding frequency")
```



```
Pond.dur <- ifelse(EDIT_updates_filtered$pond_dur == "", "none", EDIT_updates_filtered$pond_dur)

as.data.frame(table(EDIT_updates_filtered$pond_dur)) %>%
  rename("Ponding Duration" = Var1) %>% arrange(desc(Freq))
```

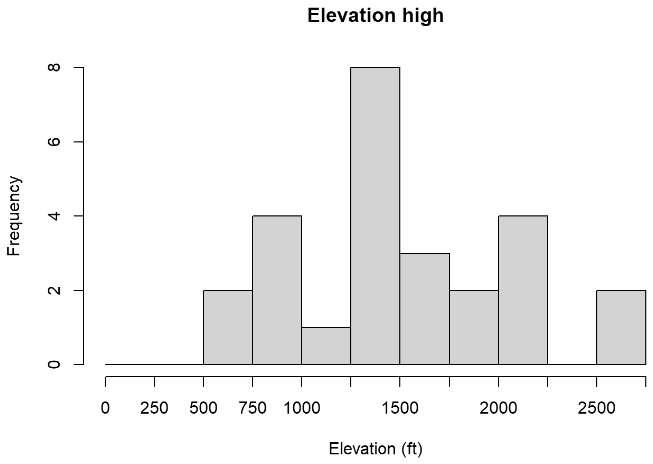
```
##   Ponding Duration Freq
## 1                    26
```

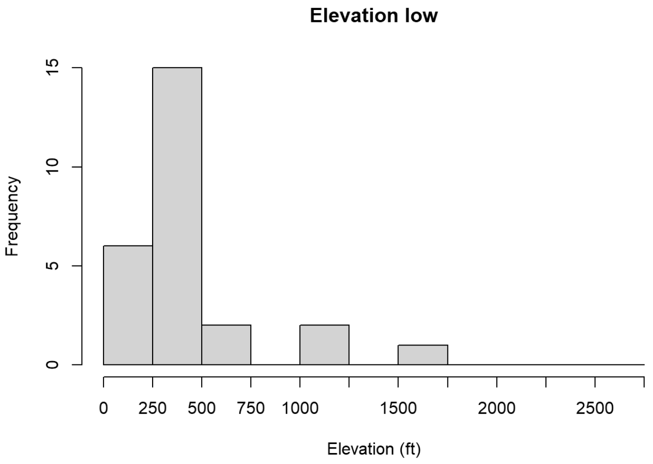
```
barplot(table(Pond.dur), main = "Ponding duration")
```

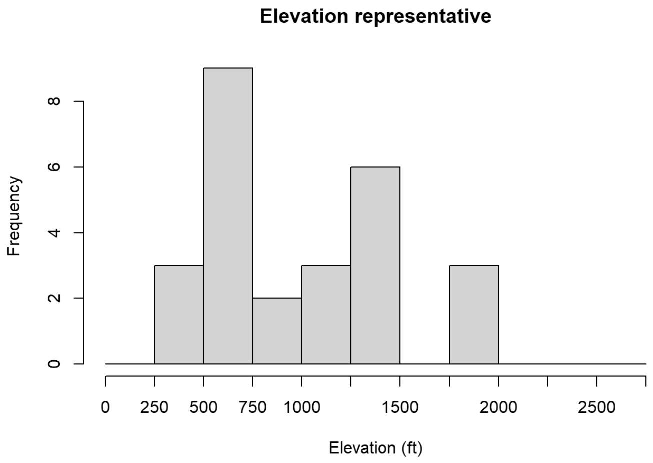


Elevation range

```
{hist(EDIT_updates_filtered$elev_h_ft, breaks = seq(round_any(min(EDIT_updates_filtered$elev_l_f
t), 250, f = floor),
                                                    round_any(max(EDIT_updates_filtered$elev_h_f
t), 250, f = ceiling), 250), xaxt = "n",
      main = "Elevation high", xlab = "Elevation (ft)")
axis(side = 1, at = seq(round_any(min(EDIT_updates_filtered$elev_l_ft), 250, f = floor),
round_any(max(EDIT_updates_filtered$elev_h_ft), 250, f = ceiling), 250),
     labels = seq(round_any(min(EDIT_updates_filtered$elev_l_ft), 250, f = floor),
round_any(max(EDIT_updates_filtered$elev_h_ft), 250, f = ceiling), 250))}

{hist(EDIT_updates_filtered$elev_l_ft, breaks = seq(round_any(min(EDIT_updates_filtered$elev_l_f
t), 250, f = floor),
                                                    round_any(max(EDIT_updates_filtered$elev_h_f
t), 250, f = ceiling), 250), xaxt = "n",
      main = "Elevation low", xlab = "Elevation (ft)")
axis(side = 1, at = seq(round_any(min(EDIT_updates_filtered$elev_l_ft), 250, f = floor),
round_any(max(EDIT_updates_filtered$elev_h_ft), 250, f = ceiling), 250), labels = seq(round_any
(min(EDIT_updates_filtered$elev_l_ft), 250, f = floor),
round_any(max(EDIT_updates_filtered$elev_h_ft), 250, f = ceiling), 250))}

{hist(EDIT_updates_filtered$elev_r_ft, breaks = seq(round_any(min(EDIT_updates_filtered$elev_l_f
t), 250, f = floor),                                                     round_any(max(EDIT_upda
tes_filtered$elev_h_ft), 250, f = ceiling), 250), xaxt = "n",
      main = "Elevation representative", xlab = "Elevation (ft)")
axis(side = 1, at = seq(round_any(min(EDIT_updates_filtered$elev_l_ft), 250, f = floor),
round_any(max(EDIT_updates_filtered$elev_h_ft), 250, f = ceiling), 250), labels = seq(round_any
(min(EDIT_updates_filtered$elev_l_ft), 250, f = floor),
round_any(max(EDIT_updates_filtered$elev_h_ft), 250, f = ceiling), 250))}
```







Elevation (high) summary

```
summary(EDIT_updates_filtered$elev_h_ft)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     640    1175    1500    1565    2000    2600 
```

Elevation (low) summary

```
summary(EDIT_updates_filtered$elev_l_ft)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   160.0   310.0   400.0   479.2   455.0  1600.0 
```

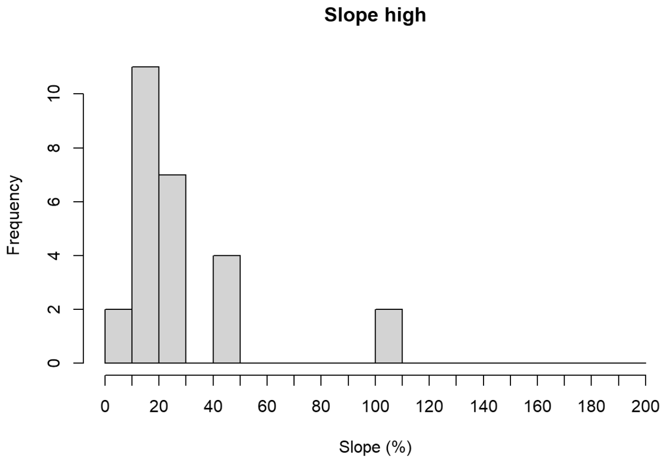
Elevation (representative) summary

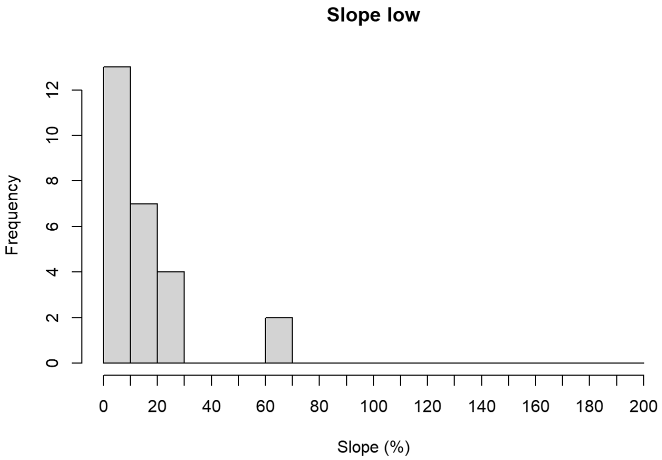
```
summary(EDIT_updates_filtered$elev_r_ft)
```

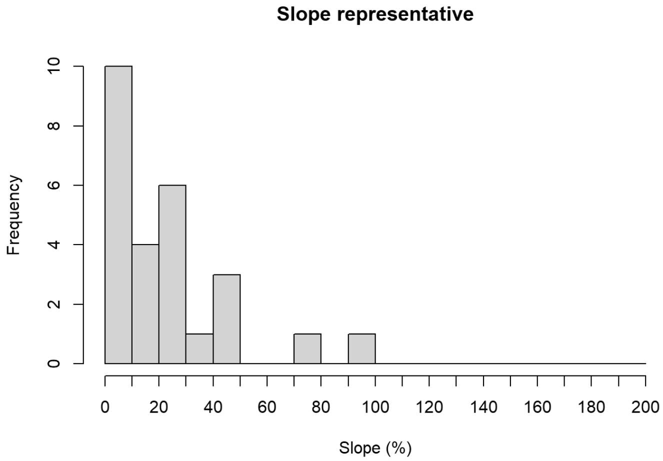
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   400.0   655.0   875.0   998.1  1300.0  1840.0 
```

Slope

```
{hist(EDIT_updates_filtered$slope_h, breaks = seq(0, 200, 10), xaxt = "n",
      main = "Slope high", xlab = "Slope (%)")
axis(side = 1, at = seq(0, 200, 10),
     labels = seq(0, 200, 10))}

{hist(EDIT_updates_filtered$slope_l, breaks = seq(0, 200, 10), xaxt = "n",
      main = "Slope low", xlab = "Slope (%)")
axis(side = 1, at = seq(0, 200, 10), labels = seq(0, 200, 10))}

{hist(EDIT_updates_filtered$slope_r, breaks = seq(0, 200, 10), xaxt = "n",
      main = "Slope representative", xlab = "Slope (%)")
axis(side = 1, at = seq(0, 200, 10), labels = seq(0, 200, 10))}
```







Slope low

```
summary(EDIT_updates_filtered$slope_l)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    3.00   11.50   15.88   15.00   70.00
```

Slope high

```
summary(EDIT_updates_filtered$slope_h)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    5.00   15.00   22.50   31.08   30.00  110.00
```

Slope representative

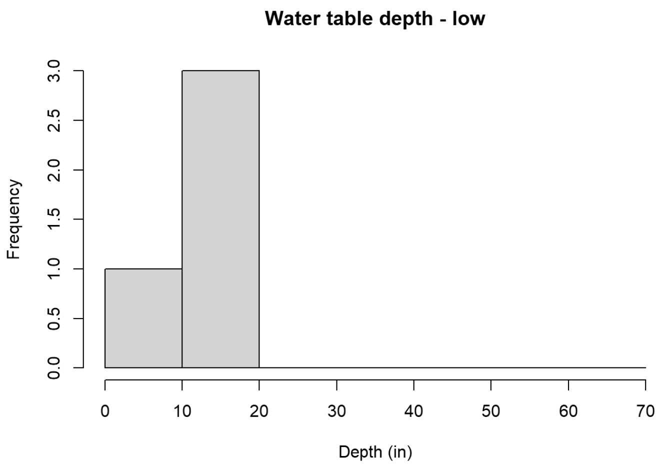
```
summary(EDIT_updates_filtered$slope_r)
```

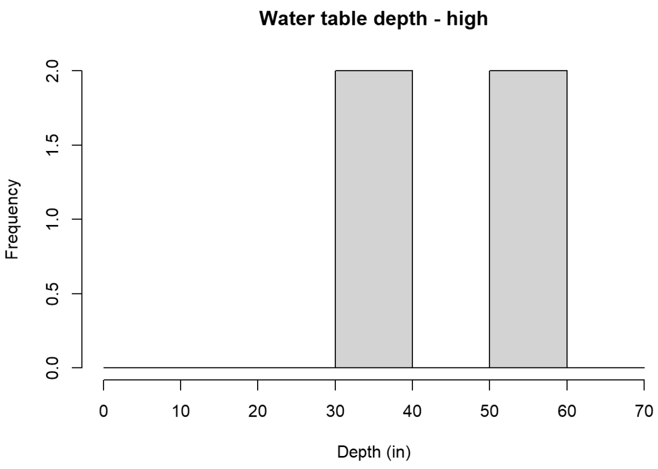
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2.00    8.00   16.00   23.15   24.25   92.00
```

Water table depth

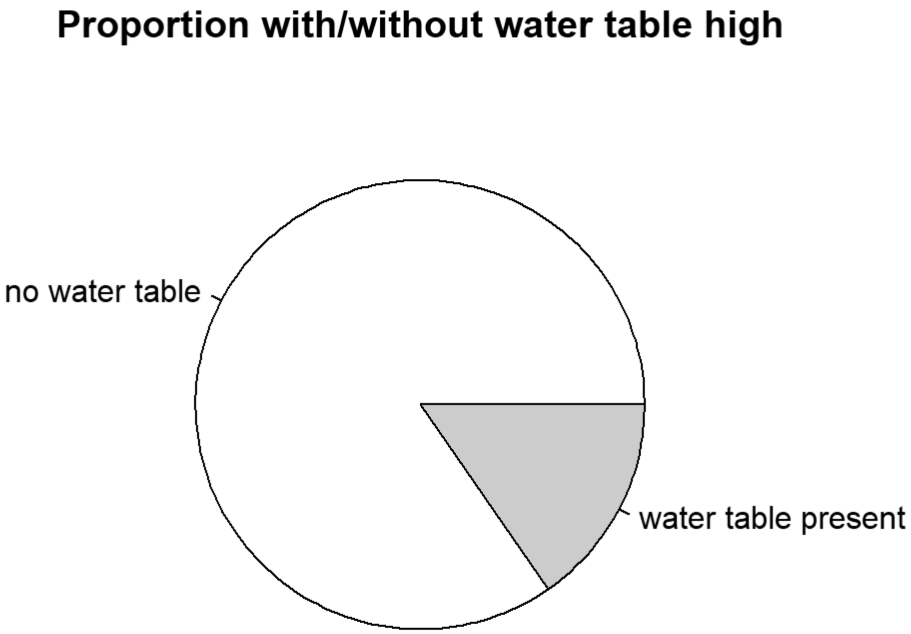
If ecosite does not seem to be associated with a water table, enter “NA”

```
if (all(is.na(EDIT_updates_filtered$water_table_l_in))) {print("No recorded values for water tab
le low. Enter NA.")

} else{
  {hist(EDIT_updates_filtered$water_table_l_in, breaks = seq(plyr::round_any(min(EDIT_updates_fi
ltered$water_table_l_in, na.rm = TRUE), 10, f = floor),
                                                    plyr::round_any(max(EDIT_updates_filtered$wa
ter_table_h_in + 1, na.rm = TRUE), 10, f = ceiling), 10), xaxt = "n",
      main = "Water table depth - low", xlab = "Depth (in)")
axis(side = 1, at = seq(round_any(min(EDIT_updates_filtered$water_table_l_in, na.rm = TRUE), 10,
f = floor),
                                                    round_any(max(EDIT_updates_filtered$water_ta
ble_h_in + 1, na.rm = TRUE), 10, f = ceiling), 10),
     labels = seq(round_any(min(EDIT_updates_filtered$water_table_l_in, na.rm = TRUE), 10, f = f
loor),
                                                    round_any(max(EDIT_updates_filtered$water_ta
ble_h_in + 1, na.rm = TRUE), 10, f = ceiling), 10))}

}


if (all(is.na(EDIT_updates_filtered$water_table_h_in))) {print("No recorded values for water tab
le high. Enter NA.")

} else{
  {hist(EDIT_updates_filtered$water_table_h_in, breaks = seq(round_any(min(EDIT_updates_filtered
$water_table_l_in, na.rm = TRUE), 10, f = floor),
                                                    round_any(max(EDIT_updates_filtered$water_ta
ble_h_in + 1, na.rm = TRUE), 10, f = ceiling), 10), xaxt = "n",
      main = "Water table depth - high", xlab = "Depth (in)")
axis(side = 1, at = seq(round_any(min(EDIT_updates_filtered$water_table_l_in, na.rm = TRUE), 10,
f = floor),
                                                    round_any(max(EDIT_updates_filtered$water_ta
ble_h_in + 1, na.rm = TRUE), 10, f = ceiling), 10),
     labels = seq(round_any(min(EDIT_updates_filtered$water_table_l_in, na.rm = TRUE), 10, f = f
loor),
                                                    round_any(max(EDIT_updates_filtered$water_ta
ble_h_in + 1, na.rm = TRUE), 10, f = ceiling), 10))}

}
```

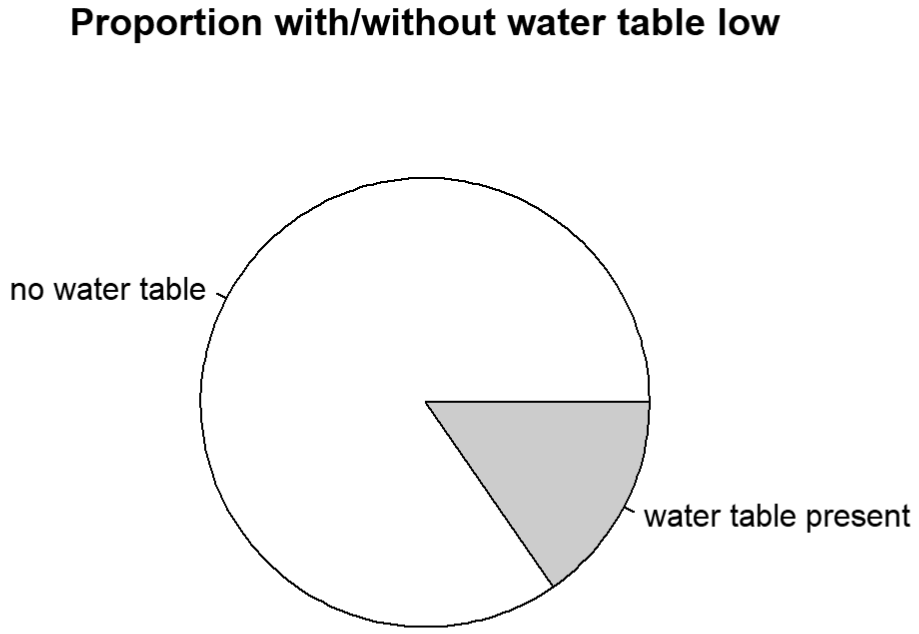




```
EDIT_updates_filtered_for_pie_h <- ifelse(is.na(EDIT_updates_filtered$water_table_h_in) | EDIT_updates_filtered$water_table_h_in == "", "no water table", "water table present")
pie(table(EDIT_updates_filtered_for_pie_h), main = "Proportion with/without water table high")
```



```
EDIT_updates_filtered_for_pie_l <- ifelse(is.na(EDIT_updates_filtered$water_table_l_in) | EDIT_updates_filtered$water_table_l_in == "", "no water table", "water table present")
pie(table(EDIT_updates_filtered_for_pie_l), main = "Proportion with/without water table low")
```



```
as.data.frame(table(EDIT_updates_filtered$water_table_h_in, useNA = "always")) %>%
  rename('Water table depth high' = Var1) %>% arrange(desc(Freq))
```

```
##   Water table depth high Freq
## 1                   <NA>   22
## 2                     40    2
## 3                     60    2
```

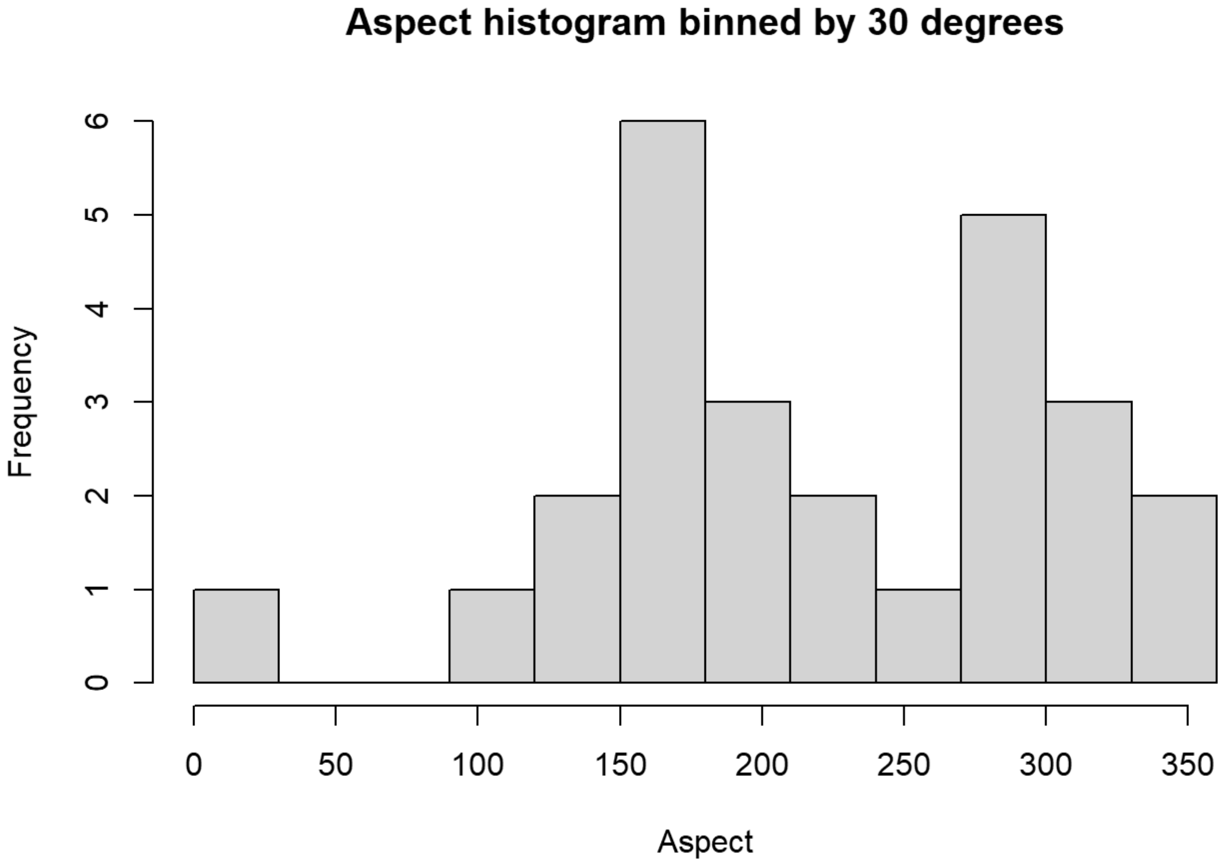
```
as.data.frame(table(EDIT_updates_filtered$water_table_l_in, useNA = "always")) %>%
  rename('Water table depth low' = Var1) %>% arrange(desc(Freq))
```

```
##   Water table depth low Freq
## 1                  <NA>   22
## 2                     3    1
## 3                    18    1
## 4                    19    1
## 5                    20    1
```

Aspect

If aspect is not strongly associated with a particular aspect, select “not applicable”

```
hist(EDIT_updates_filtered$aspect_rep, breaks = seq(0, 360, 30),
     main = "Aspect histogram binned by 30 degrees", xlab = "Aspect")
```



Climate features tab:

This tab should have climate stations selected and data should be autopopulated. If the climate stations are not selected, notify Nate.

Water features tab:

Canned statement for both ‘Water features narrative’ and ‘Wetland description’: Due to the topographic position, this site does not have water features or wetlands.

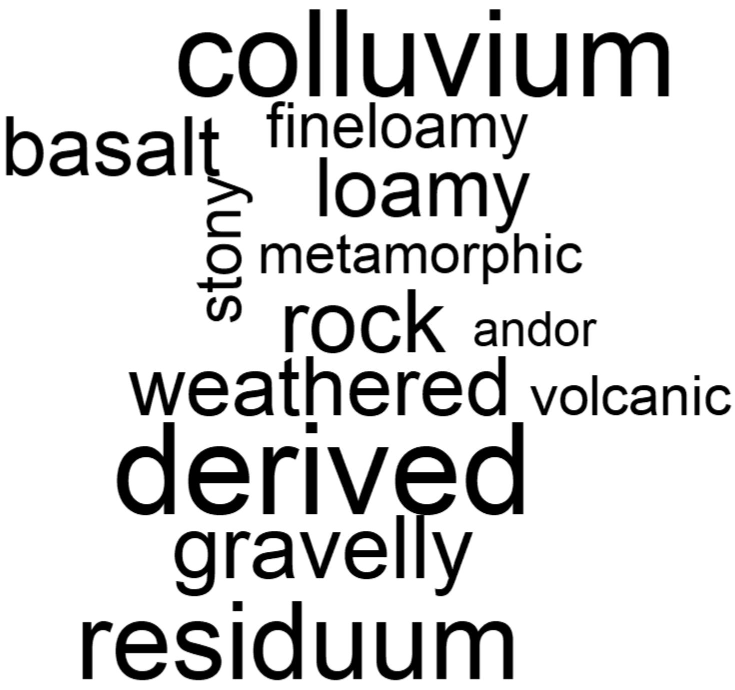
Soil features tab:

Parent material

```
pm_table <- as.data.frame(table(EDIT_updates_filtered$parent_material)) %>%
  rename(PM = Var1) %>% arrange(desc(Freq))

pandoc.table(pm_table, split.cells = c(60, 10))
```

```
##
## --------------------------------------------------------------------
##                               PM                                Freq
## ------------------------------------------------------------- ------
##       stony and gravelly colluvium derived from basalt          6
##
##         fine-loamy residuum weathered from sandstone            2
##
##    gravelly loamy colluvium derived from metamorphic rock       2
##
##  loamy residuum and/or colluvium derived from volcanic rock     2
##
##          loamy residuum weathered from volcanic rock            2
##
##  fine-loamy colluvium and/or residuum weathered from igneous    1
##                     and metamorphic rock
##
##  fine-loamy colluvium derived from volcanic and sedimentary     1
##      rock over clayey residuum weathered from claystone
##
##       fine-loamy residuum and/or colluvium derived from         1
##                       metamorphic rock
##
##      fine-loamy residuum weathered from metamorphic rock        1
##
##            gravelly colluvium derived from basalt               1
##
##    gravelly loamy residuum weathered from metamorphic rock      1
##
##            gravelly residuum weathered from basalt              1
##
##              loamy alluvium derived from basalt                 1
##
##  loamy residuum and/or colluvium derived from metavolcanics     1
##
##          loamy residuum weathered from metavolcanics            1
##
##              silty alluvium derived from basalt                 1
##
##   stony fine-loamy colluvium derived from volcanic breccia      1
## --------------------------------------------------------------------
```

```
PM <- paste(EDIT_updates_filtered$parent_material, collapse = " ")

pm_words <- strsplit(PM, "[[:space:]]+")[[1]]

wordcloud::wordcloud(pm_words)
```



Surface texture

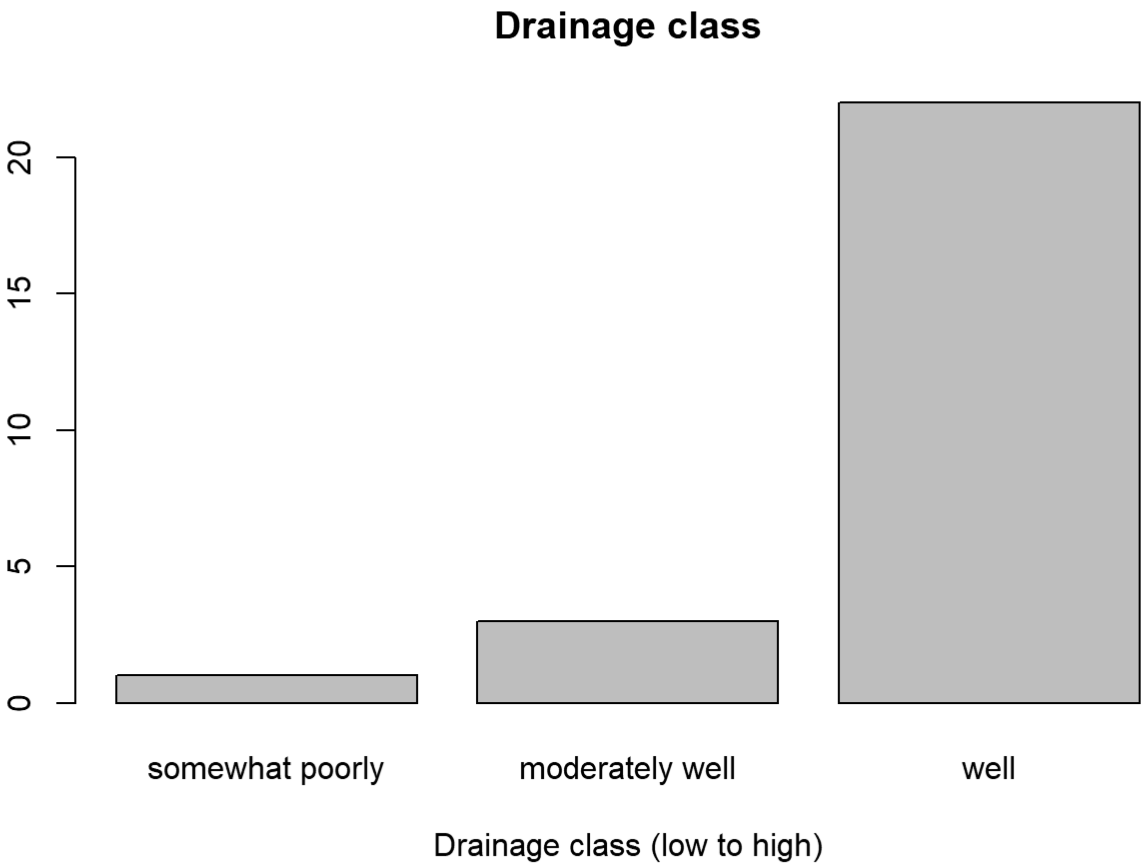
```
as.data.frame(table(EDIT_updates_filtered$surf_texture)) %>%
  rename("Surface texture" = Var1) %>% arrange(desc(Freq))
```

```
##   Surface texture Freq
## 1            GR-L   11
## 2               L    6
## 3           CBV-L    4
## 4             FSL    1
## 5          GR-SIL    1
## 6           GRV-L    1
## 7           GRX-L    1
## 8              SL    1
```

Drainage class

Drainage class does not have low, high, and representative. Use your best judgement about defining representative and actual values.

```
drainage_class_levels <- c("very poorly", "poorly", "somewhat poorly", "moderately well", "well"
, "somewhat excessively",
                           "excessively")


drainage_factor <- factor(EDIT_updates_filtered$drainage_class, levels = drainage_class_levels[d
rainage_class_levels %in% EDIT_updates_filtered$drainage_class])

barplot(table(drainage_factor), main = "Drainage class", xlab = "Drainage class (low to high)")
```



```
as.data.frame(table(EDIT_updates_filtered$drainage_class)) %>%
  rename("Drainage class" = Var1) %>% arrange(desc(Freq))
```

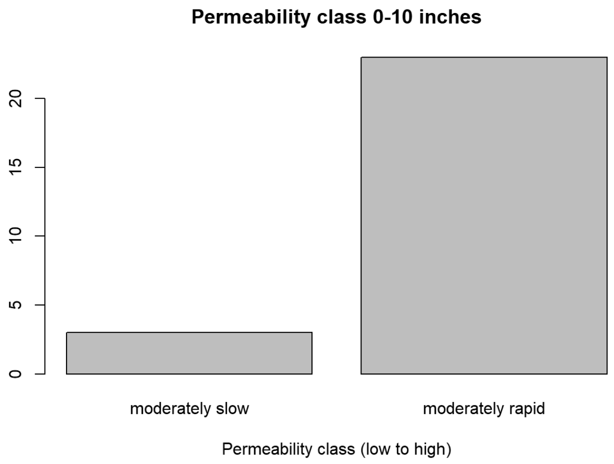
```
##    Drainage class Freq
## 1            well   22
## 2 moderately well    3
## 3 somewhat poorly    1
```

Permeability class

Permeability class does not have low, high, and representative. Use your best judgement about defining representative and actual values.

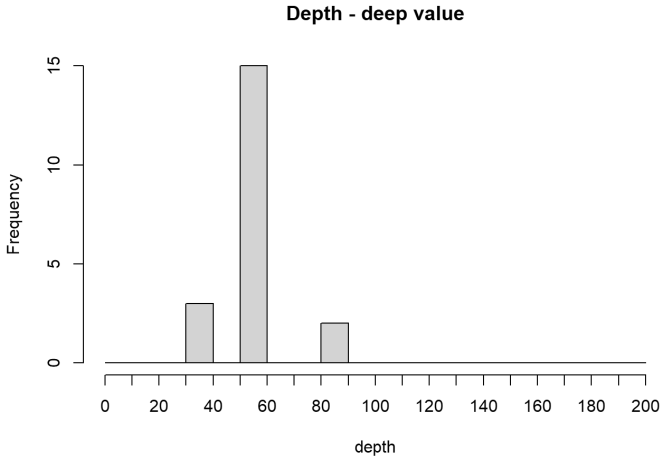
```
my_levels <- c("very slow", "slow", "moderately slow", "moderate", "moderately rapid", "rapid",
"very rapid")

perm_0_10 <- mapvalues(EDIT_updates_filtered$ksat_l_0_10_in, from = c("very low", "low", "modera
tely low", "moderate", "moderately high",
                                                                        "high", "very high"), to =
c("very slow", "slow", "moderately slow",

"moderate", "moderately rapid", "rapid",

"very rapid"))


perm_0_10 <- factor(perm_0_10, levels = my_levels[my_levels %in% perm_0_10])


if(all(is.na(perm_0_10))){print("All permeability values NA.")} else {barplot(table(perm_0_10),
 main = "Permeability class 0-10 inches", xlab = "Permeability class (low to high)")}

as.data.frame(table(EDIT_updates_filtered$ksat_l_0_10_in)) %>%
  rename("Permeability class" = Var1) %>% arrange(desc(Freq))
```

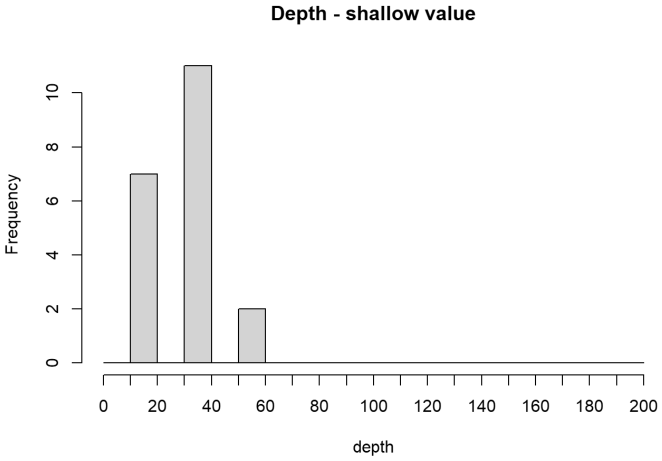
```
##   Permeability class Freq
## 1    moderately high   23
## 2     moderately low    3
```

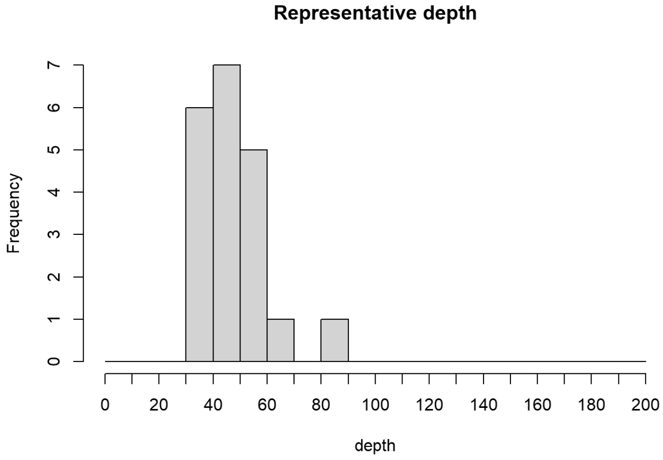


Depth to restrictive layer

```
{hist(EDIT_updates_filtered$restrict_depth_h_in, breaks = seq(0,200, 10), xaxt = "n",
      main = "Depth - deep value", xlab = "depth")
axis(side = 1, at = seq(0, 200, 10), labels = seq(0,200, 10))}

{hist(EDIT_updates_filtered$restrict_depth_l_in, breaks = seq(0,200, 10), xaxt = "n",
      main = "Depth - shallow value", xlab = "depth")
axis(side = 1, at = seq(0, 200, 10), labels = seq(0,200, 10))}

{hist(EDIT_updates_filtered$restrict_depth_r_in, breaks = seq(0,200, 10), xaxt = "n",
      main = "Representative depth", xlab = "depth")
axis(side = 1, at = seq(0, 200, 10), labels = seq(0,200, 10))}
```







Soil depth

Soil depth is the same as depth to restrictive layer.

```
summary(EDIT_updates_filtered$restrict_depth_h_in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##      40      60      60      60      60      90       6
```

```
summary(EDIT_updates_filtered$restrict_depth_l_in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##      20      20      40      35      40      60       6
```

```
summary(EDIT_updates_filtered$restrict_depth_r_in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   31.00   40.00   45.00   47.35   52.00   90.00       6
```

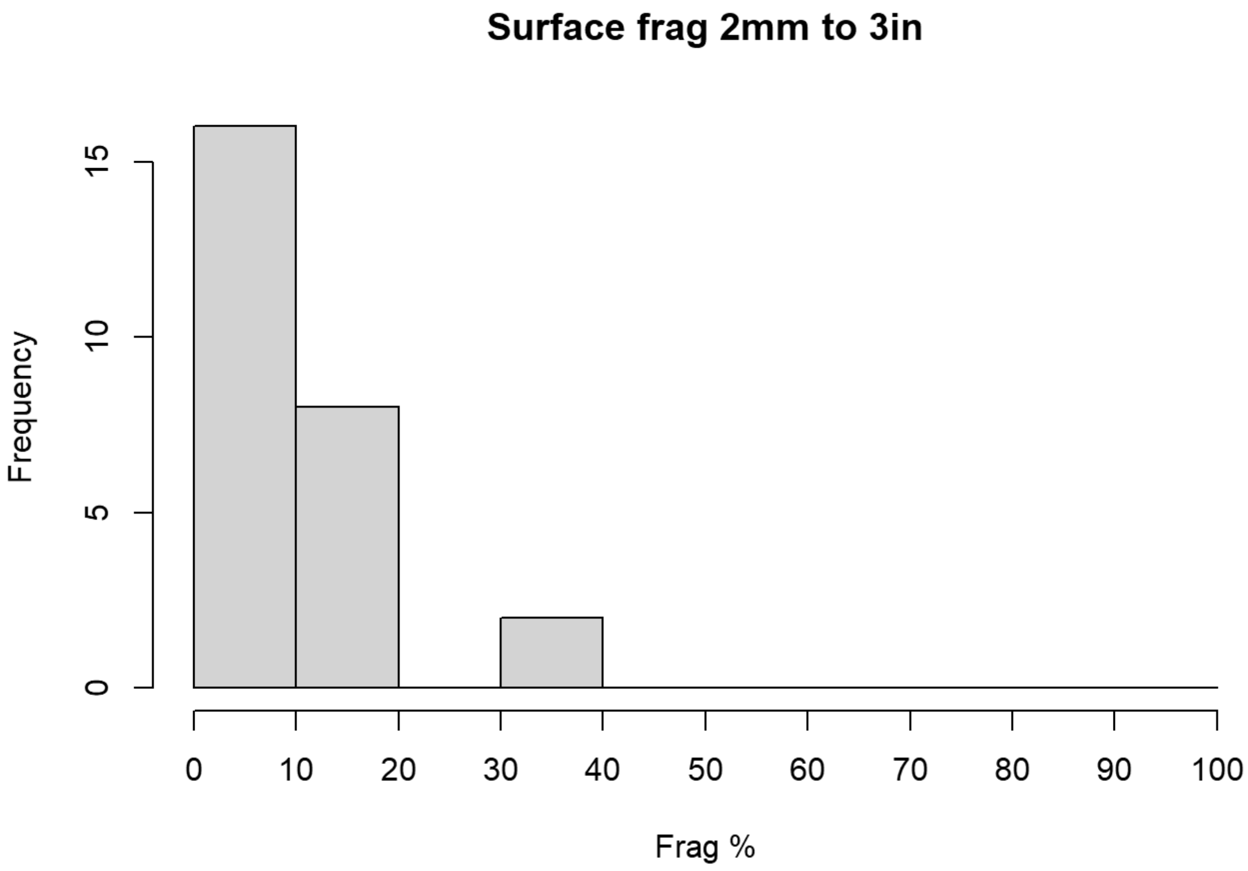
```
EDIT_updates_filtered$dept
```

```
## NULL
```

Surface frag cover

Surface frag does not have low, high, and representative. Use your best judgement about defining representative and actual values. NOTE: Notice that there is a calculation of what % of frags are zero. This can’t be seen in the histogram because it is binned. It is possible that the representative values could be a low of zero and a high of zero if a high percentage are zero. I would say over 75% 0’s would be sufficient to have representative 0 to 0.

```
frags2mm3in <- ifelse(is.na(EDIT_updates_filtered$frags_r_2mm_3in_on_surf), 0,
                EDIT_updates_filtered$frags_r_2mm_3in_on_surf)

frags3plus <- ifelse(is.na(EDIT_updates_filtered$frags_r_gt_3in_on_surf), 0,
                     EDIT_updates_filtered$frags_r_gt_3in_on_surf)

{hist(frags2mm3in, breaks = seq(0, 100, 10), xaxt = "n", main = "Surface frag 2mm to 3in",
     xlab = "Frag %")
axis(side = 1, at = seq(0, 100, 10), labels = seq(0,100, 10))}
```



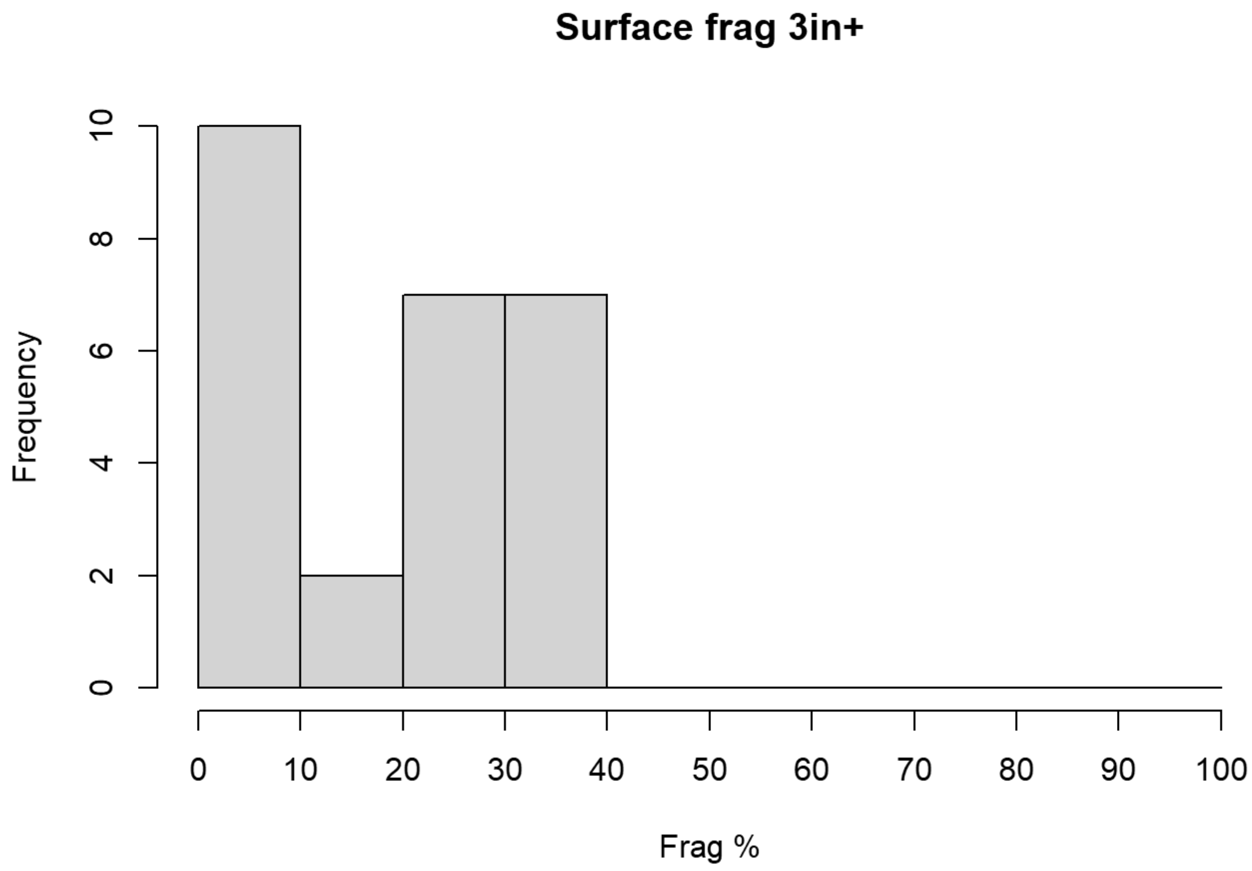
```
summary(frags2mm3in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   5.000   5.000   9.962  15.000  40.000
```

```
# Percent of frags 2mm - 3in values equal to zero
length(frags2mm3in[frags2mm3in == 0])*100/length(frags2mm3in)
```

```
## [1] 11.53846
```

```
{hist(frags3plus, breaks = seq(0, 100, 10), xaxt = "n", main = "Surface frag 3in+",
     xlab = "Frag %")
axis(side = 1, at = seq(0, 100, 10), labels = seq(0,100, 10))}
```



```
summary(frags3plus)
```

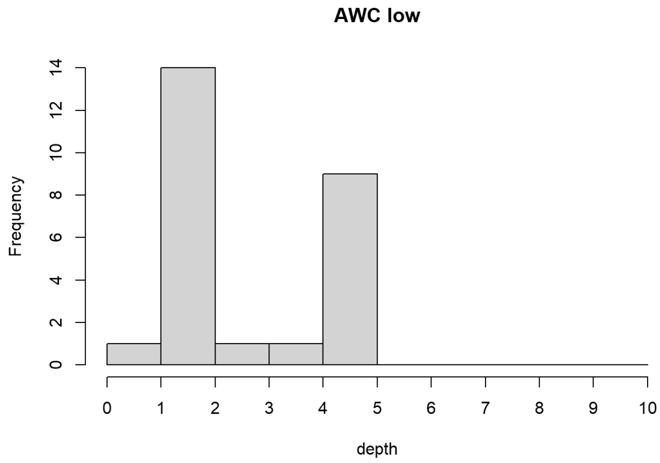
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.00   22.00   17.27   31.50   35.00
```

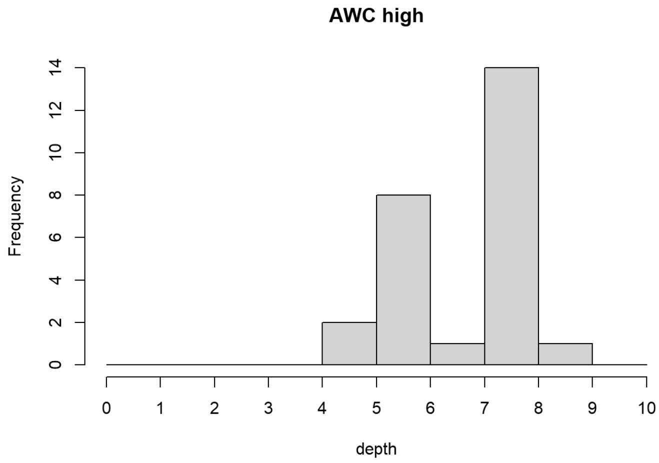
```
# Percent of frags 3in+ equal to zero
length(frags3plus[frags3plus == 0])*100/length(frags3plus)
```

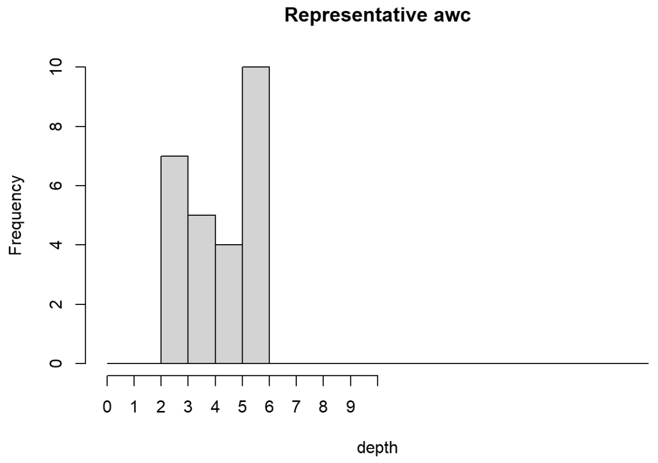
```
## [1] 19.23077
```

Available water capacity

```
{hist(EDIT_updates_filtered$awc_l_0_40_in, breaks = seq(0,10, 1), xaxt = "n",
      main = "AWC low", xlab = "depth")
axis(side = 1, at = seq(0, 10, 1), labels = seq(0, 10, 1))}

{hist(EDIT_updates_filtered$awc_h_0_40_in, breaks = seq(0, 10, 1), xaxt = "n",
      main = "AWC high", xlab = "depth")
axis(side = 1, at = seq(0, 10, 1), labels = seq(0, 10, 1))}

{hist(EDIT_updates_filtered$awc_r_0_40_in, breaks = seq(0,20, 1), xaxt = "n",
      main = "Representative awc", xlab = "depth")
axis(side = 1, at = seq(0, 10, 1), labels = seq(0,10, 1))}
```


AWC low


AWC high


Representative awc

```
summary(EDIT_updates_filtered$awc_l_0_40_in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.500   1.700   1.900   2.715   4.525   4.700
```

```
summary(EDIT_updates_filtered$awc_h_0_40_in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   5.000   5.200   7.100   6.623   7.600   8.200
```

```
summary(EDIT_updates_filtered$awc_r_0_40_in)
```

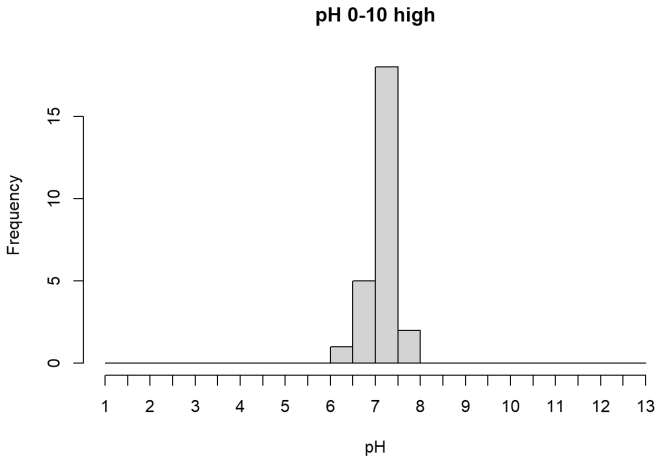
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   2.200   3.025   4.350   4.177   5.200   5.900 
```

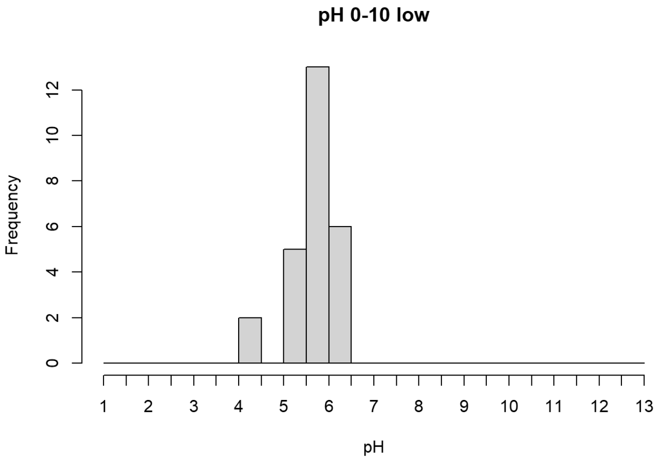
Soil reaction

There is no pH representative value. Use your best judgement.

0-10

```
{hist(EDIT_updates_filtered$pH_h_0_10in, breaks = seq(1,13, 0.5), xaxt = "n",
     main = "pH 0-10 high", xlab = "pH")
axis(side = 1, at = seq(1, 13, 0.5), labels = seq(1, 13, 0.5))}

{hist(EDIT_updates_filtered$pH_l_0_10in, breaks = seq(1, 13, 0.5), xaxt = "n",
     main = "pH 0-10 low", xlab = "pH")
axis(side = 1, at = seq(1, 13, 0.5), labels = seq(1, 13, 0.5))}
```





```
summary(EDIT_updates_filtered$pH_h_0_10in)
```

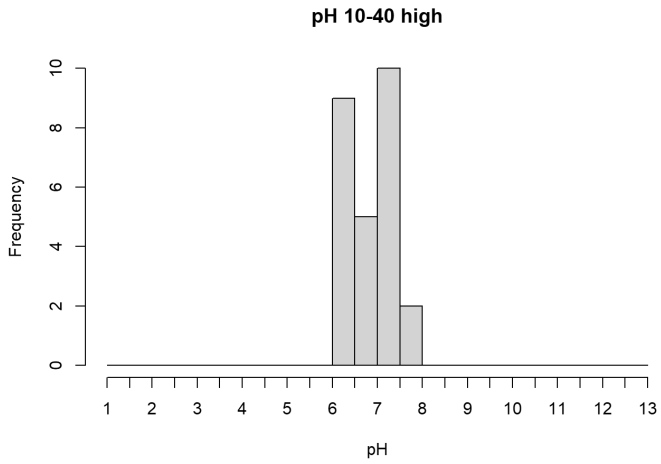
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   6.500   7.300   7.300   7.242   7.300   7.800 
```

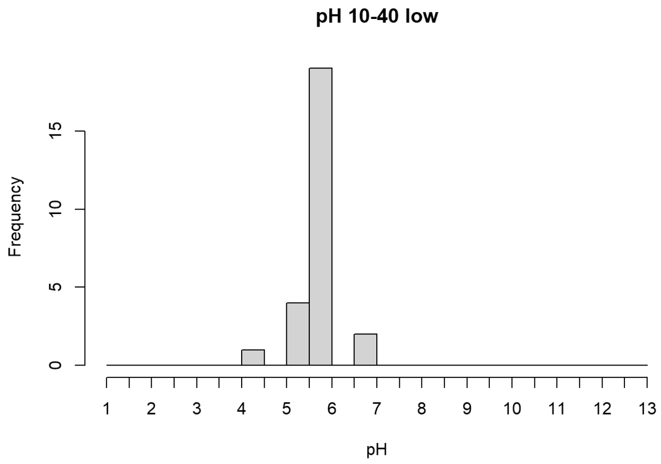
```
summary(EDIT_updates_filtered$pH_l_0_10in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   4.500   5.525   5.600   5.612   6.000   6.100 
```

10-40

```
{hist(EDIT_updates_filtered$pH_h_10_40in, breaks = seq(1,13, 0.5), xaxt = "n",
     main = "pH 10-40 high", xlab = "pH")
axis(side = 1, at = seq(1, 13, 0.5), labels = seq(1, 13, 0.5))}

{hist(EDIT_updates_filtered$pH_l_10_40in, breaks = seq(1,13, 0.5), xaxt = "n",
     main = "pH 10-40 low", xlab = "pH")
axis(side = 1, at = seq(1, 13, 0.5), labels = seq(1, 13, 0.5))}
```





```
summary(EDIT_updates_filtered$pH_h_10_40in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   6.500   6.500   7.000   6.981   7.300   7.800
```

```
summary(EDIT_updates_filtered$pH_l_10_40in)
```

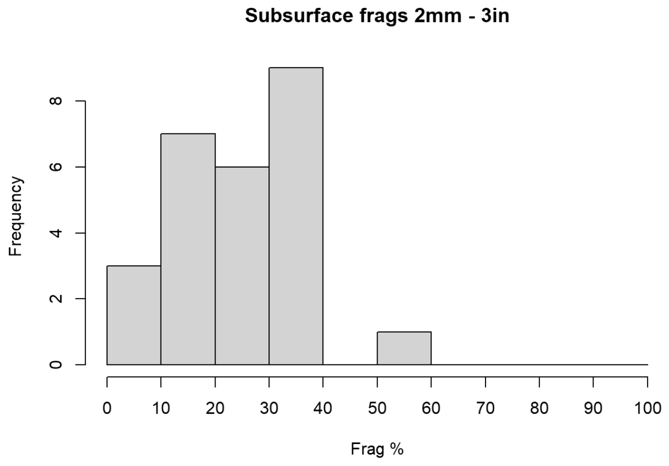
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   4.300   5.600   5.600   5.581   5.600   6.600
```

Subsurface frag

Subsurface frags are depth 0-60in.

2mm-3in

```
{hist(EDIT_updates_filtered$subsurf_frags_r_2mm.3in, breaks = seq(0, 100, 10),
     main = "Subsurface frags 2mm - 3in", xlab = "Frag %", xaxt = "n")
axis(side = 1, at = seq(0, 100, 10), labels = seq(0, 100, 10))}
```



% frags 2mm-3in equal to zero

```
length(EDIT_updates_filtered$subsurf_frags_r_2mm.3in[EDIT_updates_filtered$subsurf_frags_r_2mm.3
in == 0]) *
  100/length(EDIT_updates_filtered$subsurf_frags_r_2mm.3in)
```

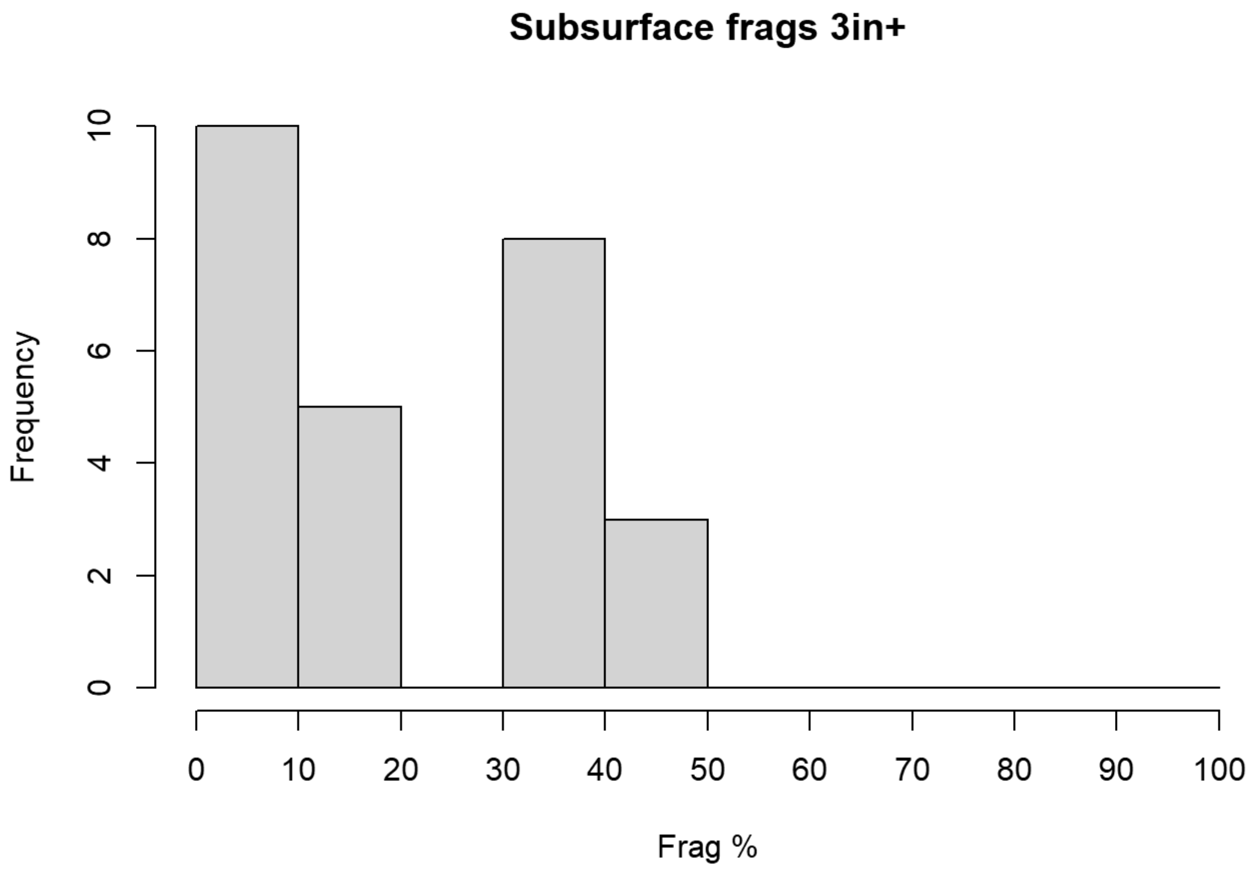
```
## [1] 0
```

```
summary(EDIT_updates_filtered$subsurf_frags_r_2mm.3in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    6.00   17.00   21.00   23.69   31.00   51.00 
```

3in+

```
{hist(EDIT_updates_filtered$subsurf_frags_r_gt_3in, breaks = seq(0, 100, 10),
     main = "Subsurface frags 3in+", xlab = "Frag %", xaxt = "n")
axis(side = 1, at = seq(0, 100, 10), labels = seq(0, 100, 10))}
```



% frags 3in+ equal to zero

```
length(EDIT_updates_filtered$subsurf_frags_r_gt_3in[EDIT_updates_filtered$subsurf_frags_r_gt_3in
== 0]) *
  100/length(EDIT_updates_filtered$subsurf_frags_r_gt_3in)
```

```
## [1] 30.76923
```

```
summary(EDIT_updates_filtered$subsurf_frags_r_gt_3in)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    0.00   16.50   20.31   38.50   47.00 
```