



United States Department of Agriculture

Expanding the Toolkit for Soil Scientists

The Algorithms for Quantitative Pedology {aqp}
R package



Natural
Resources
Conservation
Service

A close-up photograph of a soil sample, showing a dark, crumbly texture with some organic matter. The text "Natural Resources Conservation Service" is overlaid in white.

Andrew G. Brown

 andrew.g.brown@usda.gov

Dylan E. Beaudette

 dylan.beaudette@usda.gov

SSSA 2020 — Virtual Meeting
Big Data with Soil Survey,
Capacity Building

Natural
Resources
Conservation
Service

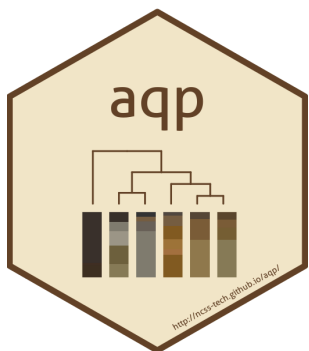
nrcs.usda.gov/

The {aqp} R package



R is a freely available language and environment for statistical computing and graphics.

The {aqp} **R** package provides functions to support data-driven tasks such as visualization, aggregation, and classification of soil profiles. The code is open-source and under active development by members of the National Cooperative Soil Survey.



Project Homepage: <http://ncss-tech.github.io/AQP/>

{aqp} on Comprehensive R Archive Network (CRAN; *stable*)

- <http://cran.r-project.org/web/packages/aqp/>

{aqp} on GitHub (*development*):

- <http://github.com/ncss-tech/aqp/>



Soil Data Inputs



You can load data from any source that **R** supports!

The `{soilDB}` package has several ways to get soil data!

```

columnlookup
columnlookuphist
comonth
comparativeyielddata
comparativeyieldrefquadrats
component
dmuidref : int
segnum : smallint
compct_l : smallint
compct_r : smallint
compct_h : smallint
compname : varchar
compname_s : smallint
localphase : varchar
compkind : smallint
majcompflag : bit
otherph : varchar
slope_l : real
slope_r : real
slope_h : real
slopeusage_l : smallint
slopeusage_r : smallint
slopeusage_h : smallint
runoff : smallint
runoff_s : smallint
tfact : smallint
tfact_s : smallint
wei : smallint
wei_s : smallint
weg : smallint
weg_s : smallint
erochl : smallint
earthcovkind1 : smallint
earthcovkind2 : smallint
hydricon : smallint
hydricrating : smallint
drainagecl : smallint
hydrologystatus : smallint
elev_l : real
elev_r : real
elev_h : real
aspectcwise : smallint
aspectrep : smallint

```

- `fetchSDA`, `fetchSDA_spatial`, `SDA_query` & `SDA_spatialQuery` for **SSURGO** from **Soil Data Access (SDA)**
- `fetchKSSL` for querying a snapshot of the **Kellogg Soil Survey Laboratory (KSSL)** database
- `fetchOSD` for series type location profiles and narratives from **Official Series Descriptions (OSDs)**
- `fetchNASIS` for **NASIS** pedons / components from a local database



{aqp} SoilProfileCollection

```
library(aqp) # load aqp package  
  
# load sample dataset CA serpentinite Soils  
# (McGahan et al., 2009)  
data(sp4, package = "aqp") # see ?sp4 for metadata
```



{aqp} SoilProfileCollection

```
library(aqp) # load aqp package

# load sample dataset CA serpentinite Soils
# (McGahan et al., 2009)
data(sp4, package = "aqp") # see ?sp4 for metadata
```

```
head(sp4, n = 8)
```

##	id	name	top	bottom	K	Mg	Ca	CEC_7	ex_Ca_to_Mg	sand	silt	clay	CF
## 1	colusa	A	0	3	0.3	25.7	9.0	23.0	0.35	46	33	21	0.12
## 2	colusa	ABt	3	8	0.2	23.7	5.6	21.4	0.23	42	31	27	0.27
## 3	colusa	Bt1	8	30	0.1	23.2	1.9	23.7	0.08	40	28	32	0.27
## 4	colusa	Bt2	30	42	0.1	44.3	0.3	43.0	0.01	27	18	55	0.16
## 5	glenn	A	0	9	0.2	21.9	4.4	18.8	0.20	54	20	25	0.55
## 6	glenn	Bt	9	34	0.3	18.9	4.5	27.5	0.20	49	18	34	0.84
## 7	kings	A	0	4	0.2	12.1	1.4	23.7	0.58	43	55	3	0.50
## 8	kings	Bt1	4	13	0.6	12.1	7.0	18.0	0.51	36	49	15	0.75



Optional: use data.table or tibble!

```
sp4 <- data.table::as.data.table(sp4)
head(sp4)
```

##	id	name	top	bottom	K	Mg	Ca	CEC_7	sand	silt	clay	CF
## 1:	colusa	A	0	3	0.3	25.7	9.0	23.0	46	33	21	0.12
## 2:	colusa	ABt	3	8	0.2	23.7	5.6	21.4	42	31	27	0.27
## 3:	colusa	Bt1	8	30	0.1	23.2	1.9	23.7	40	28	32	0.27
## 4:	colusa	Bt2	30	42	0.1	44.3	0.3	43.0	27	18	55	0.16
## 5:	glenn	A	0	9	0.2	21.9	4.4	18.8	54	20	25	0.55
## 6:	glenn	Bt	9	34	0.3	18.9	4.5	27.5	49	18	34	0.84



Optional: use data.table or tibble!

```
sp4 <- data.table::as.data.table(sp4)
head(sp4)
```

```
##           id name top bottom    K    Mg    Ca CEC_7 sand silt clay  CF
## 1: colusa    A   0     3 0.3 25.7 9.0 23.0  46  33  21 0.12
## 2: colusa  ABt   3     8 0.2 23.7 5.6 21.4  42  31  27 0.27
## 3: colusa  Bt1   8    30 0.1 23.2 1.9 23.7  40  28  32 0.27
## 4: colusa  Bt2  30    42 0.1 44.3 0.3 43.0  27  18  55 0.16
## 5: glenn    A   0     9 0.2 21.9 4.4 18.8  54  20  25 0.55
## 6: glenn   Bt   9    34 0.3 18.9 4.5 27.5  49  18  34 0.84
```

```
sp4 <- tibble::as_tibble(sp4)
head(sp4)
```

```
## # A tibble: 6 x 10
##   id      name    top bottom    K    Mg    Ca CEC_7  sand  silt
##   <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <dbl> <int> <int>
## 1 colusa A       0      3  0.3 25.7  9    23    46    33
## 2 colusa ABt     3      8  0.2 23.7  5.6 21.4    42    31
## 3 colusa Bt1     8     30  0.1 23.2  1.9 23.7    40    28
## 4 colusa Bt2    30     42  0.1 44.3  0.3 43    27    18
## 5 glenn  A       0      9  0.2 21.9  4.4 18.8    54    20
## 6 glenn  Bt     9     34  0.3 18.9  4.5 27.5    49    18
```



{aqp} Methods (basics)

"Promote" *data.frame*-like horizon data to a *SoilProfileCollection* (SPC) object.

```
class(sp4)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```



{aqp} Methods (basics)

"Promote" *data.frame*-like horizon data to a *SoilProfileCollection* (SPC) object.

```
class(sp4)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

```
depths(sp4) ← id ~ top + bottom # specify site ID, top and bottom depth
```



{aqp} Methods (basics)

"Promote" *data.frame*-like horizon data to a *SoilProfileCollection* (SPC) object.

```
class(sp4)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

```
depths(sp4) ← id ~ top + bottom # specify site ID, top and bottom depth
```

```
class(sp4) # sp4 promoted from tbl_df → SoilProfileCollection
```

```
## [1] "SoilProfileCollection"  
## attr(,"package")  
## [1] "aqp"
```



{aqp} Methods (basics)

"Promote" *data.frame*-like horizon data to a *SoilProfileCollection* (SPC) object.

```
class(sp4)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

```
depths(sp4) ← id ~ top + bottom # specify site ID, top and bottom depth
```

```
class(sp4) # sp4 promoted from tbl_df → SoilProfileCollection
```

```
## [1] "SoilProfileCollection"  
## attr(,"package")  
## [1] "aqp"
```

```
str(profile_id(sp4), vec.len = 3) # view first 3 profile IDs
```

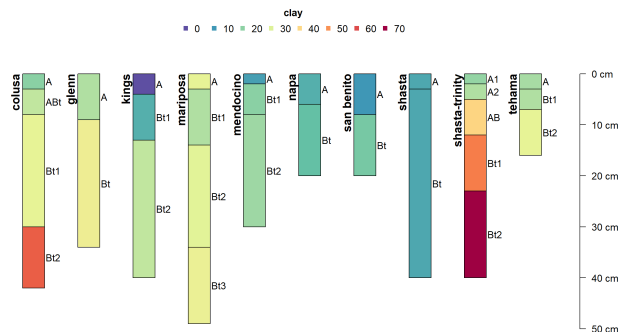
```
## chr [1:10] "colusa" "glenn" "kings" ...
```



{aqp} Methods (site data)

plot

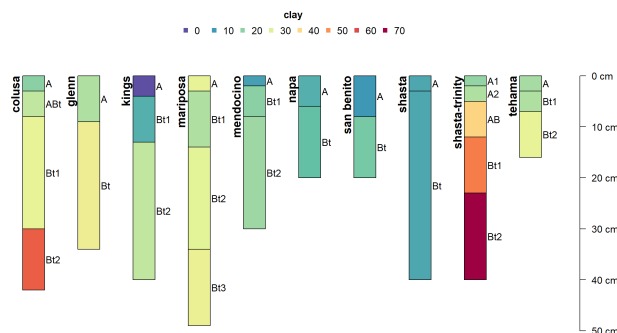
```
plot(sp4, # plot % clay content
     color = 'clay',
     id.style = 'side',
     cex.names = 1)
```



{aqp} Methods (site data)

plot

```
plot(sp4, # plot % clay content
      color = 'clay',
      id.style = 'side',
      cex.names = 1)
```



site

```
site(sp4) # "site" data
```

```
## # A tibble: 10 x 1
##   id
##   <chr>
## 1 colusa
## 2 glenn
## 3 kings
## 4 mariposa
## 5 mendocino
## 6 napa
## 7 san benito
## 8 shasta
## 9 shasta-trinity
## 10 tehama
```



{aqp} Methods (horizon data)

horizons

```
horizons(sp4) # "horizon" data
```

```
## # A tibble: 30 x 13
```

##	id	name	top	bottom	K	Mg	Ca	CEC_7	sand	silt	clay	CF	h
##	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>	<int>	<dbl>	<dbl>
##	1	colusa	A	0	3	0.3	25.7	9	23	46	33	21	0.12
##	2	colusa	ABt	3	8	0.2	23.7	5.6	21.4	42	31	27	0.27
##	3	colusa	Bt1	8	30	0.1	23.2	1.9	23.7	40	28	32	0.27
##	4	colusa	Bt2	30	42	0.1	44.3	0.3	43	27	18	55	0.16
##	5	glenn	A	0	9	0.2	21.9	4.4	18.8	54	20	25	0.55
##	6	glenn	Bt	9	34	0.3	18.9	4.5	27.5	49	18	34	0.84
##	7	kings	A	0	4	0.2	12.1	1.4	23.7	43	55	3	0.5
##	8	kings	Bt1	4	13	0.6	12.1	7	18	36	49	15	0.75
##	9	kings	Bt2	13	40	0.8	17.7	4.4	20	27	45	27	0.67
##	10	mariposa	A	0	3	0.6	28.3	5.8	29.3	42	26	32	0.25

```
## # ... with 20 more rows
```



{aqp} Methods (extract)

[i,]

```
sp4[1:2,] # i-index: first two profiles
```

```
## SoilProfileCollection with 2 profiles and 6 horizons
```

```
## profile ID: id | horizon ID: hzID
```

```
## Depth range: 34 - 42 cm
```

```
##
```

```
## ----- Horizons (6 / 6 rows | 10 / 13 columns) -----
```

```
## # A tibble: 6 x 10
```

	id	hzID	top	bottom	name	K	Mg	Ca	CEC_7	sand
	<chr>	<chr>	<int>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
## 1	colusa	1	0	3	A	0.3	25.7	9	23	46
## 2	colusa	2	3	8	ABt	0.2	23.7	5.6	21.4	42
## 3	colusa	3	8	30	Bt1	0.1	23.2	1.9	23.7	40
## 4	colusa	4	30	42	Bt2	0.1	44.3	0.3	43	27
## 5	glenn	5	0	9	A	0.2	21.9	4.4	18.8	54
## 6	glenn	6	9	34	Bt	0.3	18.9	4.5	27.5	49

```
##
```

```
## ----- Sites (2 / 2 rows | 1 / 1 columns) -----
```

```
## # A tibble: 2 x 1
```

```
## id
```

```
## <chr>
```



{aqp} Methods (extract)

[i,]

```
sp4[1:2,] # i-index: first two profiles
```

[,j]

```
sp4[,1:2] # j-index: first two horizons (of each profile!)
```

```
## SoilProfileCollection with 10 profiles and 20 horizons
```

```
## profile ID: id | horizon ID: hzID
```

```
## Depth range: 5 - 40 cm
```

```
##
```

```
## ----- Horizons (6 / 20 rows | 10 / 13 columns) -----
```

```
## # A tibble: 6 x 10
```

	id	hzID	top	bottom	name	K	Mg	Ca	CEC_7	sand
	<chr>	<chr>	<int>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
## 1	colusa	1	0	3	A	0.3	25.7	9	23	46
## 2	colusa	2	3	8	ABt	0.2	23.7	5.6	21.4	42
## 3	glenn	5	0	9	A	0.2	21.9	4.4	18.8	54



{aqp} Methods (accessors)

\$

```
sp4$clay      # get clay data
```

```
## [1] 21 27 32 55 25 34  3 15 27 32 25 31 33 13 21 23 15 17 12 19 14 14 22 25 40
```



{aqp} Methods (accessors)

\$

```
sp4$clay      # get clay data
```

```
## [1] 21 27 32 55 25 34  3 15 27 32 25 31 33 13 21 23 15 17 12 19 14 14 22 25 40
```

[[

```
sp4[["clay"]] # using expression for name, not symbol
```

```
## [1] 21 27 32 55 25 34  3 15 27 32 25 31 33 13 21 23 15 17 12 19 14 14 22 25 40
```



{aqp} Methods (setters)

\$← and [[←

Calculate Ca:Mg ratio

```
sp4$ex_Ca_to_Mg      ← sp4$Ca / sp4$Mg  
sp4[["ex_Ca_to_Mg"]] ← sp4$Ca / sp4$Mg
```



{aqp} Methods (setters)

\$← and [[←

Calculate Ca:Mg ratio

```
sp4$ex_Ca_to_Mg      ← sp4$Ca / sp4$Mg  
sp4[["ex_Ca_to_Mg"]] ← sp4$Ca / sp4$Mg
```

Initialize a new column with a single value

```
site(sp4)$new_var ← 2  
horizons(sp4)$new_hz_var ← 3
```



{aqp} Methods (setters)

\$← and [[←

Calculate Ca:Mg ratio

```
sp4$ex_Ca_to_Mg      ← sp4$Ca / sp4$Mg  
sp4[["ex_Ca_to_Mg"]] ← sp4$Ca / sp4$Mg
```

Initialize a new column with a single value

```
site(sp4)$new_var ← 2  
horizons(sp4)$new_hz_var ← 3
```

```
length(sp4$new_var) # 10 sites, 10 values  
length(sp4$new_hz_var) # 30 horizons, 30 values  
sp4$new_var ← NULL # remove a column
```



{aqp} Methods (subset)

subset is the {aqp} method for extracting profiles that meet certain logical criteria at the site or horizon level.

```
# site property filtering, using base  
sub.sp4 ← subset(sp4, id %in% c("colusa", "mariposa", "shasta"))
```



{aqp} Methods (subset)

subset is the {aqp} method for extracting profiles that meet certain logical criteria at the site or horizon level.

```
# site property filtering, using base  
sub.sp4 ← subset(sp4, id %in% c("colusa", "mariposa", "shasta"))
```

```
# or dplyr-like syntax: filter  
sub.sp4 ← filter(sp4, id %in% c("colusa", "mariposa", "shasta"))
```



{aqp} Methods (subset)

subset is the {aqp} method for extracting profiles that meet certain logical criteria at the site or horizon level.

```
# site property filtering, using base
sub.sp4 ← subset(sp4, id %in% c("colusa", "mariposa", "shasta"))
```

```
# or dplyr-like syntax: filter
sub.sp4 ← filter(sp4, id %in% c("colusa", "mariposa", "shasta"))
```

```
sub.sp4
```

```
## SoilProfileCollection with 3 profiles and 10 horizons
```

```
## profile ID: id | horizon ID: hzID
```

```
## Depth range: 40 - 49 cm
```

```
##
```

```
## ----- Horizons (6 / 10 rows | 10 / 15 columns) -----
```

```
## # A tibble: 6 x 10
```

##	id	hzID	top	bottom	name	K	Mg	Ca	CEC_7	sand
##	<chr>	<chr>	<int>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
##	1 colusa	1	0	3	A	0.3	25.7	9	23	46
##	2 colusa	2	3	8	ABt	0.2	23.7	5.6	21.4	42
##	3 colusa	3	8	30	Bt1	0.1	23.2	1.9	23.7	40
##	4 colusa	4	20	42	Bt2	0.1	14.2	0.2	42	27



{aqp} Methods (subset)

```

# horizon properties (two simultaneous logical expressions)
sub.sp4 ← subset(sp4, clay > 30, ex_Ca_to_Mg < 0.05)
sub.sp4

```

```
## SoilProfileCollection with 2 profiles and 9 horizons
```

```
## profile ID: id | horizon ID: hzID
```

```
## Depth range: 40 - 42 cm
```

```
##
```

```
## ----- Horizons (6 / 9 rows | 10 / 15 columns) -----
```

```
## # A tibble: 6 x 10
```

	id	hzID	top	bottom	name	K	Mg	Ca	CEC_7	sand
	<chr>	<chr>	<int>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
## 1	colusa	1	0	3	A	0.3	25.7	9	23	46
## 2	colusa	2	3	8	ABt	0.2	23.7	5.6	21.4	42
## 3	colusa	3	8	30	Bt1	0.1	23.2	1.9	23.7	40
## 4	colusa	4	30	42	Bt2	0.1	44.3	0.3	43	27
## 5	shasta-trinity	23	0	2	A1	0.2	18.8	6.6	23	34
## 6	shasta-trinity	24	2	5	A2	0.2	25.5	4.1	21.5	33

```
## [ ... more horizons ... ]
```

```
##
```

```
## ----- Sites (2 / 2 rows | 2 / 2 columns) -----
```

```
## # A tibble: 2 x 2
```

	id	new_var
	<chr>	<dbl>
## 1	colusa	2



{aqp} Methods (split SPC -> list)

If you need to operate on groups, splitting into `list` is a good option.

```
a.list ← split(sp4, f = idname(sp4))
```

```
str(a.list, max.level = 1)
```

```
## List of 10
## $ colusa      :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ glenn       :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ kings       :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ mariposa    :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ mendocino   :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ napa        :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ san benito  :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ shasta      :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ shasta-trinity:Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
## $ tehama      :Formal class 'SoilProfileCollection' [package "aqp"] with 9 sl
```



{aqp} Methods (split SPC -> list)

If you need to operate on groups, splitting into `list` is a good option.

```
a.list ← split(sp4, f = idname(sp4))
```

```
str(a.list, max.level = 1)
```

```
a.list[[1]]
```

```
## SoilProfileCollection with 1 profiles and 4 horizons
## profile ID: id | horizon ID: hzID
## Depth range: 42 - 42 cm
##
## ----- Horizons (4 / 4 rows | 10 / 15 columns) -----
## # A tibble: 4 x 10
##   id      hzID    top bottom name      K      Mg      Ca CEC_7  sand
##   <chr> <chr> <int> <int> <chr> <dbl> <dbl> <dbl> <dbl> <int>
## 1 colusa 1         0      3 A      0.3  25.7   9     23     46
## 2 colusa 2         3      8 ABt    0.2  23.7   5.6   21.4    42
## 3 colusa 3         8     30 Bt1    0.1  23.2   1.9   23.7    40
## 4 colusa 4        30     42 Bt2    0.1  44.3   0.3   43     27
##
## ----- Sites (1 / 1 rows | 2 / 2 columns) -----
## # A tibble: 1 x 2
```



{aqp} Methods (iteration)

```
sub.sp4$soil_depth ← profileApply(sub.sp4, estimateSoilDepth)
sub.sp4$soil_depth
```

```
##          colusa shasta-trinity
##          42          40
```



{aqp} Methods (iteration)

```
sub.sp4$soil_depth ← profileApply(sub.sp4, estimateSoilDepth)
sub.sp4$soil_depth
```

```
##           colusa shasta-trinity
##           42           40
```

```
profileApply(sp4, frameify = TRUE,
  function(p) {
    data.frame(id = profile_id(p),
               soil_depth = estimateSoilDepth(p))
  })
```

```
## # A tibble: 10 x 2
```

##	id	soil_depth
##	<chr>	<int>
##	1 colusa	42
##	2 glenn	34
##	3 kings	40
##	4 mariposa	49
##	5 mendocino	30
##	6 napa	20
##	7 san benito	20
##	8 shasta	40
##	9 shasta-trinity	40



{aqp} Methods (combine; list -> SPC)

Recombine list elements into "original" SoilProfileCollection with `combine`

```
combine(a.list)
```

```
## SoilProfileCollection with 10 profiles and 30 horizons
## profile ID: id | horizon ID: hzID
## Depth range: 16 - 49 cm
##
## ----- Horizons (6 / 30 rows | 10 / 15 columns) -----
## # A tibble: 6 x 10
##   id      hzID    top bottom name      K      Mg      Ca CEC_7  sand
##   <chr> <chr> <int> <int> <chr> <dbl> <dbl> <dbl> <dbl> <int>
## 1 colusa 1         0      3 A      0.3  25.7    9     23     46
## 2 colusa 2         3      8 ABt    0.2  23.7    5.6  21.4     42
## 3 colusa 3         8     30 Bt1    0.1  23.2    1.9  23.7     40
## 4 colusa 4        30     42 Bt2    0.1  44.3    0.3   43     27
## 5 glenn  5         0      9 A      0.2  21.9    4.4  18.8     54
## 6 glenn  6         9     34 Bt     0.3  18.9    4.5  27.5     49
## [ ... more horizons ... ]
##
## ----- Sites (6 / 10 rows | 2 / 2 columns) -----
## # A tibble: 6 x 2
##   id      new_var
##   <chr> <dbl>
```



{aqp} %>% (pipes)



Use {magrittr} "pipes" (%>% infix operator) to chain operations.

```
f(x, y) = x %>% f(y)
```



{aqp} %>% (pipes)



Use {magrittr} "pipes" (%>% infix operator) to chain operations.

```
f(x, y) = x %>% f(y)
```

```
your.data2 ← operation1(your.data, argument1, argument2)  
result ← operation1(your.data2, argument3, argument4)
```

becomes...

```
result ← your.data %>%  
  operation1(argument1, argument2) %>%  
  operation2(argument3, argument4)
```



{aqp} %>% (pipes)



1. Truncate all profiles to 0 - 15 cm interval
2. Calculate NH₄OAc (pH 7) Ca (0 - 15 cm depth-weighted average cmol/kg)
3. Plot horizon-level values, in order of increasing site-level average



{aqp} %>% (pipes)



1. Truncate all profiles to 0 - 15 cm interval
2. Calculate NH₄OAc (pH 7) Ca (0 - 15 cm depth-weighted average cmol/kg)
3. Plot horizon-level values, in order of increasing site-level average

```
sp4 %>%  
  trunc(0, 15) %>%  
  mutate_profile(dwt = bottom - top / sum(bottom - top),  
                 dwt_Ca = sum(Ca * dwt)) %>%  
  plot(plot.order = order(.$dwt_Ca), color = "Ca", cex.names = 1)
```

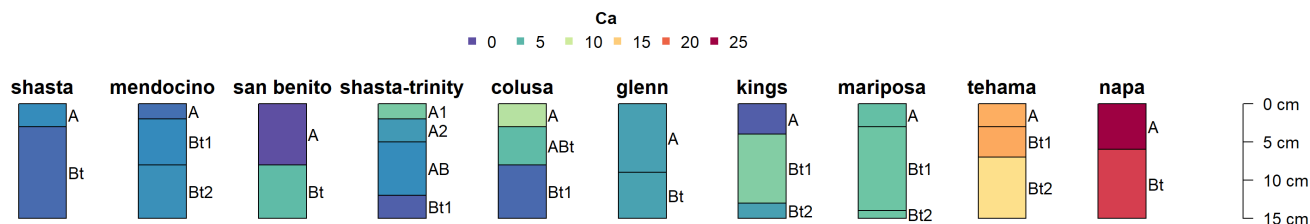


{aqp} %>% (pipes)



1. Truncate all profiles to 0 - 15 cm interval
2. Calculate NH₄OAc (pH 7) Ca (0 - 15 cm depth-weighted average cmol/kg)
3. Plot horizon-level values, in order of increasing site-level average

```
sp4 %>%
  trunc(0, 15) %>%
  mutate_profile(dwt = bottom - top / sum(bottom - top),
                 dwt_Ca = sum(Ca * dwt)) %>%
  plot(plot.order = order(.$dwt_Ca), color = "Ca", cex.names = 1)
```



{aqp} Methods (left joins)

site←

```
site(sp4) ← data.frame(id = c("mariposa", "mendocino"),  
                        site_grp = "ingroup")
```

```
head(site(sp4), 5)
```

```
## # A tibble: 5 x 3  
##   id          new_var site_grp  
##   <chr>        <dbl> <chr>  
## 1 colusa         2 <NA>  
## 2 glenn          2 <NA>  
## 3 kings          2 <NA>  
## 4 mariposa       2 ingroup  
## 5 mendocino      2 ingroup
```



{aqp} Methods (left joins)

horizons←

```
horizons(sp4) ← data.frame(id = c("mariposa", "mendocino"), hz_grp = "group")
h ← horizons(sp4)
```

```
h[5:14, c(idname(sp4), "hz_grp")]
```

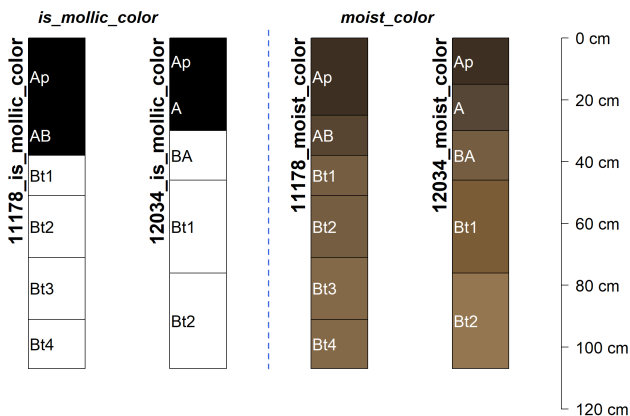
```
## # A tibble: 10 x 2
##   id      hz_grp
##   <chr>   <chr>
## 1 glenn   <NA>
## 2 glenn   <NA>
## 3 kings   <NA>
## 4 kings   <NA>
## 5 kings   <NA>
## 6 mariposa group
## 7 mariposa group
## 8 mariposa group
## 9 mariposa group
## 10 mendocino group
```

```
table(h$hz_grp, useNA = "ifany")
```

```
##
## group  <NA>
##      7    23
```



{aqp} Soil Color



{aqp} has methods for soil data in Munsell, sRGB and CIELAB.

aggregateColor, colorContrast,
colorQuantiles, contrastChart,
contrastClass,
getClosestMunsellChip,
hasDarkColors, horizonColorIndices,
huePosition, rgb2munsell,
munsell2rgb, munsell2spc,
parseMunsell, previewColors,
soilColorSignature, soilPalette



{aqp} Soil Color Opinions

```
n ← 8
hues ← c('10YR', '7.5YR', '2.5Y')
# hue
hh ← sample(hues, size = n,
            replace = TRUE,
            prob = c(0.7, 0.2, 0.1))

# value
vv ← sample(3:6, size = n,
            replace = TRUE)

# chroma
cc ← sample(c(3, 4, 6), size = n,
            replace = TRUE)
```

```
# reference soil color
m1 ← rep('10YR 4/4',
        times = n)

# opinions of soil color
m2 ← sprintf('%s %s/%s',
            hh, vv, cc)

# color contrast via dE00
cc ← colorContrast(m1, m2)

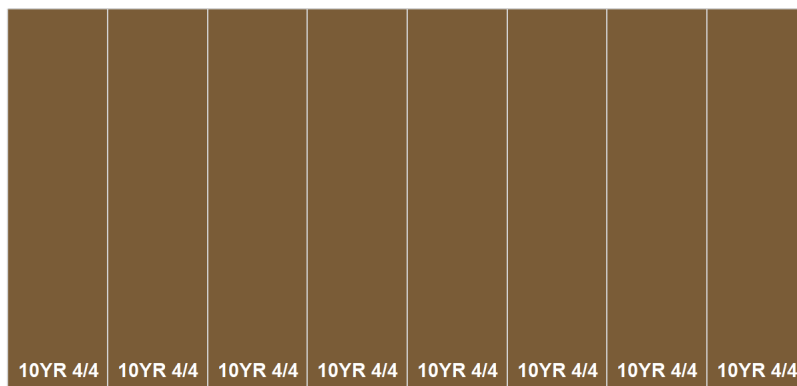
# re-order opinions
o ← order(cc$dE00)
m2 ← m2[o]
```





```
colorContrastPlot(m1, m2,
  labels = c('reference', 'opinions'),
  d.cex = 0.8, col.cex = 0.8)
```

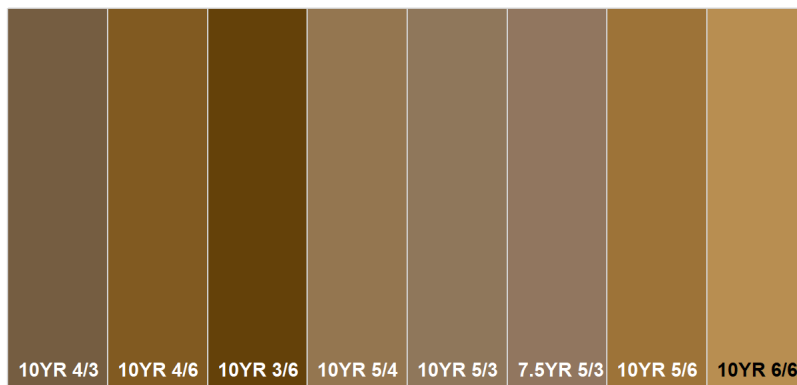
reference



Faint Distinct Distinct Faint Faint Faint Distinct Distinct

ΔE_{00} 3.11 ΔE_{00} 4.87 ΔE_{00} 9.87 ΔE_{00} 10.1 ΔE_{00} 10.7 ΔE_{00} 11.2 ΔE_{00} 11.3 ΔE_{00} 21

opinions



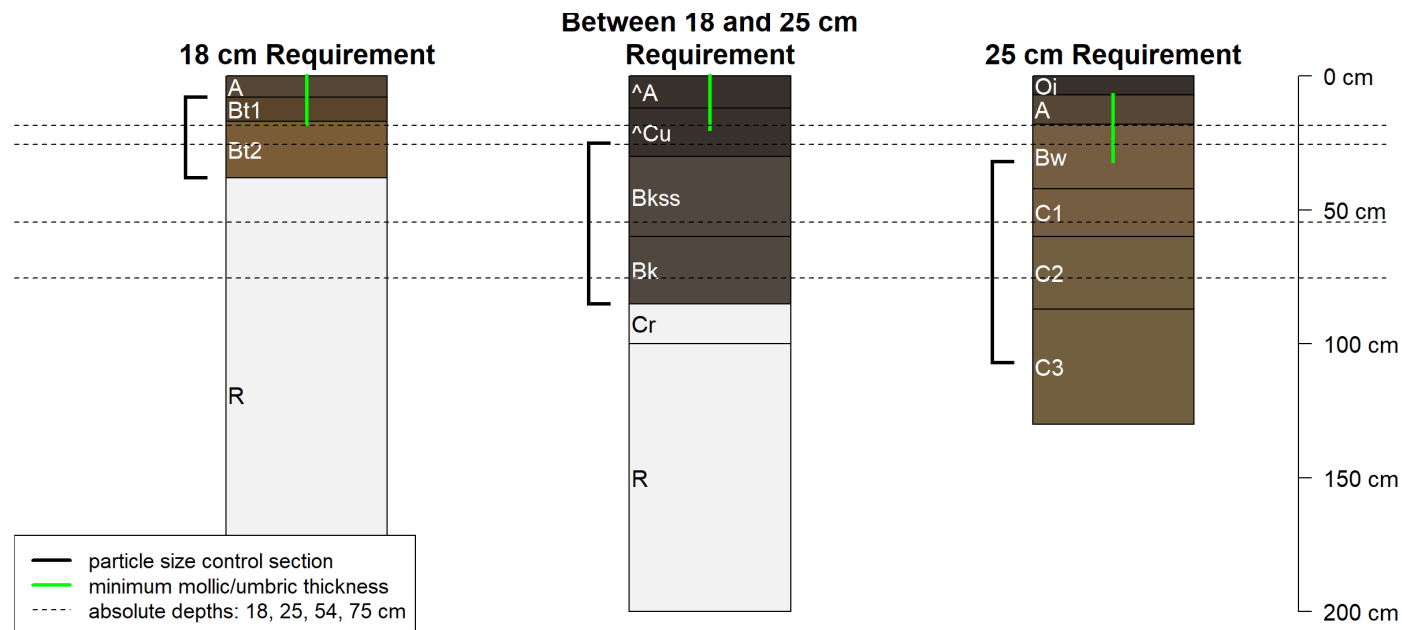
Natural
Resources
Conservation
Service

nrcs.usda.gov/



{aqp} Soil Taxonomy & Classification

{aqp} has many functions that employ heuristics about horizon designations, geometry, and diagnostic properties.



Thank you for your attention!



Andrew G. Brown, Soil Scientist, MLRA Soil Survey Office, Sonora, CA

✉ andrew.g.brown@usda.gov
 🔗 [brownag](#)
 🐦 [@humus_rocks](#)



Dylan E. Beaudette, Soil Scientist, National Soil Survey Center (duty station: Sonora, CA)

✉ dylan.beaudette@usda.gov
 🔗 [dylanbeaudette](#)
 🐦 [@dylanbeaudette](#)

USDA is an equal opportunity provider, employer, and lender.

Natural
Resources
Conservation
Service

nrcs.usda.gov/



References

Beaudette, D.E., Roudier P., and A.T. O'Geen. 2013. Algorithms for Quantitative Pedology: A Toolkit for Soil Scientists. Computers & Geosciences. 52:258 - 268.

McGahan, D.G., Southard, R.J, Claassen, V.P. 2009. Plant-Available Calcium Varies Widely in Soils on Serpentinite Landscapes. Soil Sci. Soc. Am. J. 73: 2087-2095.

Pedersen, T.L., B. Nicolae, and R. François 2020. farver: High Performance Colour Space Manipulation. R package version 2.0.3. <https://CRAN.R-project.org/package=farver>

R Core Team. 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Xie, Y. 2020. xaringan: Presentation Ninja. R package version 0.17.1. <https://github.com/yihui/xaringan>

Slides: <https://github.com/brownag/SSSA2020>

