

MIIN Part 1a: Import, standardize, and clean the paper data

Marissa Lee

June 1, 2015

Filename: MIIN_1_paperData.Rmd This markdown file does the following tasks: 1. Minimally cleans raw data related to paper selection (papers) and the data collected within papers (observations, measures, cover, species, traits)

2. For data collected within papers (measures, cover, traits)... A. Standardize variance measurements B. Aggregate values within an observation (i.e. measAgg, covAgg, traitAgg) C. Convert values to a common unit
3. Look for issues and outliers in all datasets
4. Export all cleaned datasets to DATA_SYNTHESIZED/paperData folder and histogram plots to FIGURES_TABLES/paperData

```
knitr::opts_chunk$set(cache=TRUE)
library(reshape2)
library(ggplot2)
library(ggthemes)
source('CODE/mytheme.R')
```

```
## Loading required package: grid
```

```
figuresPath<-file.path(getwd()[1], "FIGURES_TABLES", "paperData") #where to put the saved plots
fig.height<-2.5 #inches
fig.width<- 2.5 #inches
fig.res<-300
synthdataPath<-file.path(getwd()[1], "DATA", "DATA_SYNTHESIZED", "paperData") #where to put the clean d
```

1. LOAD RAW DATA (.TXT) AND DO SOME MINIMAL CLEANING:

Cleaning involves: (a) Fix the format of NAs, (b) Make the row keys numeric in dataframe (e.g. obsID, xAggNum), (c) Check that there are no duplicate obsIDs, (d) Add row keys where needed (e.g. paperID, aggID: combine obsID and xAggNum, spID: combine obsID and spEntryID), (e) Check that trait and cover-generated spIDs match the 'species' dataframe's spID by species names, (f) Make data values numeric in each dataset

```
source('CODE/paperData/script_load.R') #TASK= Load and clean raw data structure; NEEDS= files in rawDat
warnings() #warning messages that say 'NAs introduced' are okay
```

```
## NULL
```

```
measures<-measures1
measAgg<-measAgg1
```

2A. STANDARDIZE VARIANCE MEASUREMENTS

Convert variance measures (e.g. SE, SD, 95CI) to variance (VAR) in the following datasets: cover, covAgg, traits, traitAgg, measures, measAgg

```
source('CODE/paperData/script_stdVar.R') #TASK= Standardize variance; NEEDS= cover, covAgg, traits, traitAgg, measures, measAgg
```

2B. AGGREGATE VALUES WITHIN AN OBSERVATION

Aggregate mean and variance data from the xAgg files to complete the measure, cover, and trait datasets

```
# Review of some basic variance properties things:
# - SD (standard deviation) = square.root(VAR (variance))
# - SE (standard error) = SD (standard deviation) / square.root(N)
# - CV (coefficient of variation) = SD (standard deviation) / mean
# - Variance of a product of k random, independent variables: Product of (Var(x_k) + X^2) - Product of
# - Variance of a sum of k random, independent variables: Sum of (Var(x_k))

### aggregate cover #####
source('CODE/paperData/script_agg_cover.R')
##TASK= Aggregate the 'agg' files into the main dataframe
##NEEDS= cover, covAgg
##MAKES= adds new cols to cover

### aggregate traits #####
source('CODE/paperData/script_agg_traits.R')
##TASK= Aggregate the 'agg' files into the main dataframe
##NEEDS= traits, traitAgg
##MAKES= adds new cols to traits

### aggregate measures #####
source('CODE/paperData/script_agg_measures.R')
##TASK= Aggregate the 'agg' files into the main dataframe
##NEEDS= measures, measAgg
##MAKES= adds new cols to measures
```

2C. CONVERT VALUES TO A COMMON UNIT

Attach these values as 'standardized' mean and var. If the value can not be converted to the standardized unit, then simply enter NA.

```

# Remember that:
# - Var(aX) = a^2 * Var(X)
# - C:N values need to be in molC/molN; gC/gN * (14.0067/12.0107) -> molC/molN

source('CODE/paperData/script_std.R') #TASK= Convert values to a common unit; NEEDS= cover, traits, mea

# cover.new[1:10,c('covInvMean','stdmeanInv','covNatMean','stdmeanNat', 'covUnit','stdunit')]
# cover.new[1:10,c('covInvVar_VAR','stdvarInv','covNatVar_VAR','stdvarNat', 'covUnit','stdunit')]
# traits.new[1:10,c('traitMean','stdmean', 'traitUnit','stdunit')]
# traits.new[1:10,c('traitVar_VAR','stdvar','traitUnit','stdunit')]
# measures.new[1:10,c('measInvMean','stdmeanInv','measNatMean','stdmeanNat', 'measUnit','stdunit')]
# measures.new[1:10,c('measInvVar_VAR','stdvarInv','measNatVar_VAR','stdvarNat', 'measUnit','stdunit')]

```

3. LOOK FOR ISSUES AND OUTLIERS

```

#####
# PAPERS
#####
#View(papers)
#length(unique(papers$paperID));dim(papers)# This looks fine for now. Potential issues that might need

#####
# OBSERVATIONS
#####
#View(observations)
#length(unique(observations$obsID));dim(observations)# This looks fine for now.
#get rid of the columns with long notes. they have odd symbols in there that make it difficult to read/
removeCols<-colnames(observations) %in% c('obsMeasNote','obsCovNote','obsTraitNote','obsSpNote')
observations1<-observations[,!removeCols]

#####
# SPECIES
#####
#View(species)
#length(unique(species$spID));dim(species)# This looks fine for now.

#####
# COVER
#####
#View(cover)
cover$covID<-paste(cover$obsID, as.numeric(cover$covEntryID), sep=".") #identify each row
#length(unique(cover$covID));dim(cover)
#tmp<-ddply(cover, ~covID, summarise, n = length(covID)) #there seem to be duplicate cover IDs
#sum(tmp$n > 1) # this should be 0. If not, there will be duplicate covIDs
#morethan1<-which(tmp$n > 1)
#tmp[morethan1,]
# FOR COVER - Create and reshape so that inv and nat area is 1 factor column
df<-cover
m.df<-melt(df, id.vars=c("covID","aggID","covCat","covDescript","covInvasive","covMultiGrowth","covMult.

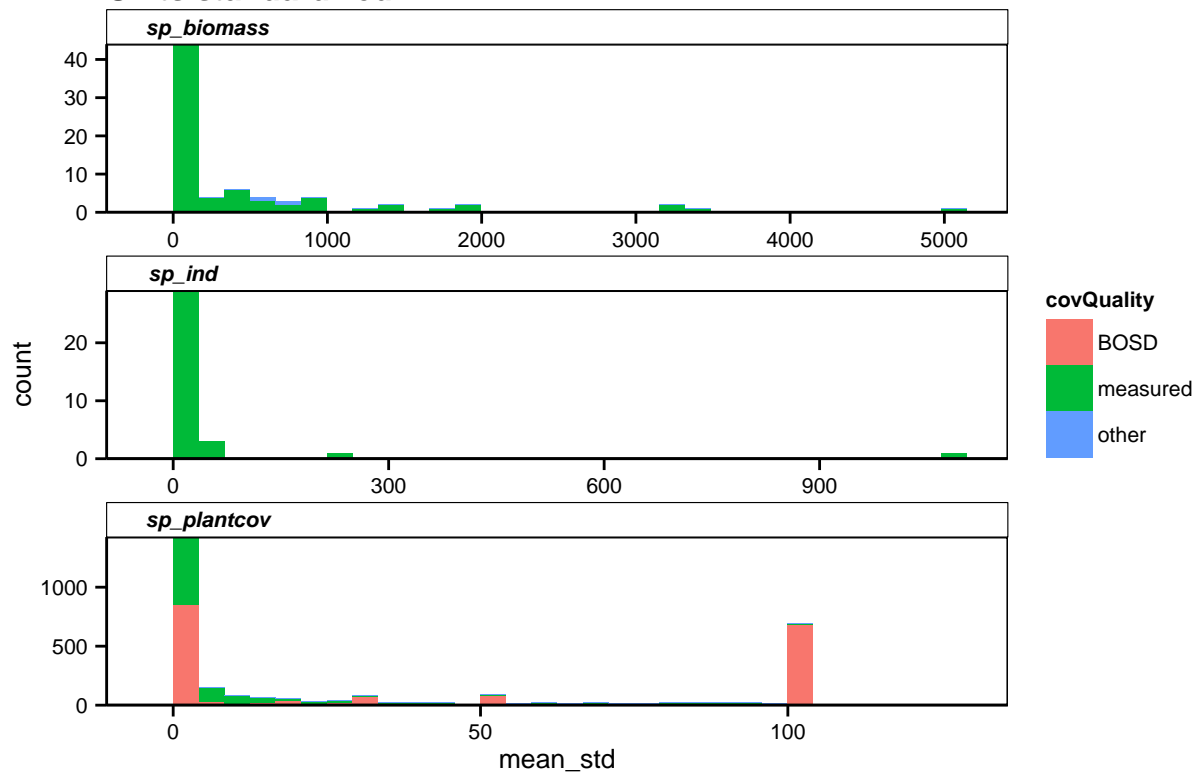
```

```

      "covSpEntryID", "covNumSpp", "spID", "covVarType", "covRef", "covImageFile", "covNote", "
measure.vars=c('covInvMean', 'covInvVar', 'covInvN', 'covNatMean', 'covNatVar', 'covNatN',
               'covInvVar_VAR', 'covNatVar_VAR',
               'stdmeanInv', 'stdmeanNat', 'stdvarInv', 'stdvarNat')) #melt
#add column to differentiate between inv and nat
m.df$invType<-rep(NA,length(dim(m.df)[1])) #add column to differentiate between inv and nat
m.df[grepl("Inv",m.df$variable),'invType']<-'inv'
m.df[grepl("Nat",m.df$variable),'invType']<-'nat'
#add column to differentiate between mean_std, var_std, mean, var, var_VAR, n
m.df$valueType<-rep(NA,length(dim(m.df)[1])) #add column to differentiate between value types
m.df[grepl("Mean",m.df$variable),'valueType']<-'mean'
m.df[grepl("Var",m.df$variable),'valueType']<-'var'
m.df[grepl("VAR",m.df$variable),'valueType']<-'var_VAR'
m.df[grepl("stdmean",m.df$variable),'valueType']<-'mean_std'
m.df[grepl("stdvar",m.df$variable),'valueType']<-'var_std'
m.df[m.df$variable %in% c('covNatN','covInvN'),'valueType']<-'n'
#cast
c.df<-dcast(m.df, covID+invType~ valueType)
idCols<-grepl("Inv",colnames(df)) | grepl("Nat",colnames(df))
idCols[colnames(df)=='covInvasive']<-FALSE
df1<-merge(df,!idCols],c.df, by=c('covID'))
#reorganize columns
cover.clean<-df1[,c('covID','invType', #unique row identifiers
                    'paperID','obsID','aggID', #IDs
                    'covCat','covUnit','stdunit','covVarType','covRef','covImageFile','covNote','covQual
                    'covDescript','covSpEntryID','covNumSpp','spID','covInvasive','covMultiGrowth','cov
                    'mean','var','n','var_VAR','mean_std','var_std')] #data
#View(cover.clean)
# FOR COVER - Plot the histograms for the standardized cover values
pHist_cover_Std<-ggplot(data=cover.clean, aes(x=mean_std, fill=covQuality)) + mytheme +
  facet_wrap(~covCat, scales='free', ncol=1) + geom_histogram()+
  scale_y_continuous(expand = c(0,0)) + ggtitle('Histogram of cover values\nUnits standardized')
pHist_cover_Std

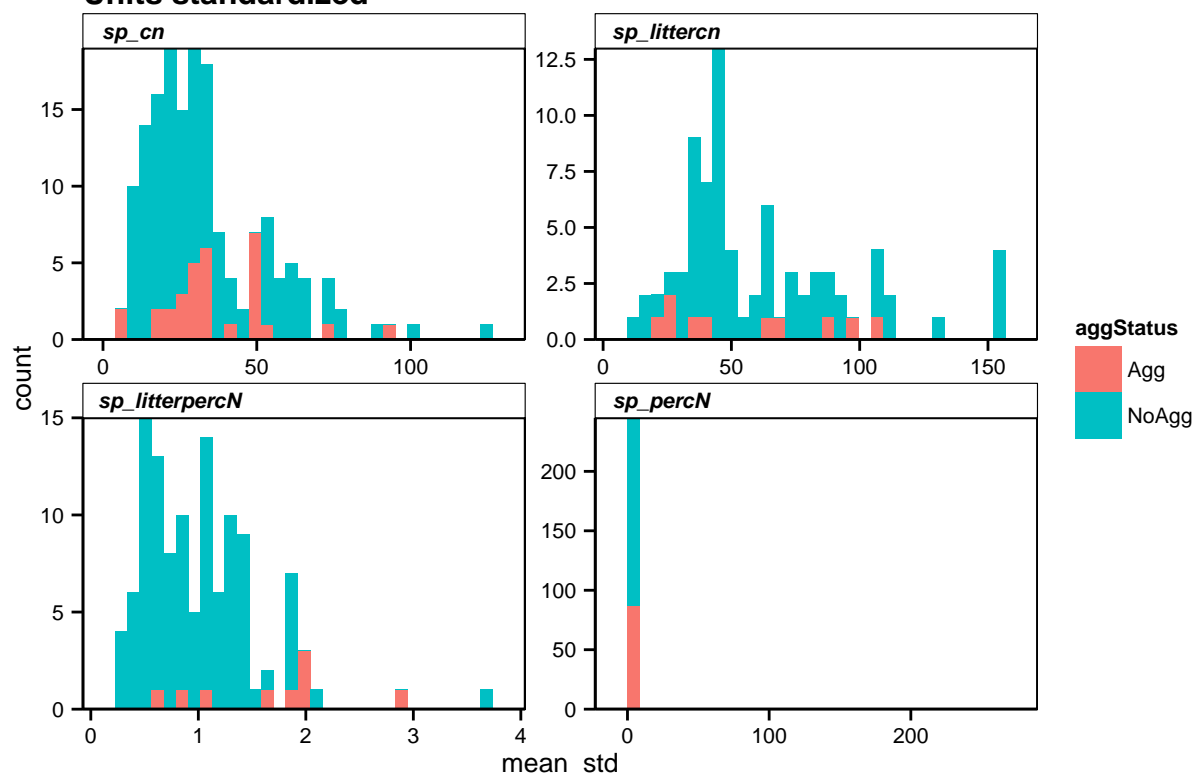
```

Histogram of cover values Units standardized



```
#####
# TRAITS
#####
traits$traitID<-paste(traits$obsID, as.numeric(traits$traitEntryID), sep=".") #identify each row
#length(unique(traits$traitID));dim(traits)
#tmp<-ddply(traits, ~traitID, summarise, n = length(traitID)) #there seem to be duplicate cover IDs
#sum(tmp$n > 1) # this should be 0. If not, there will be duplicate covIDs
#morethan1<-which(tmp$n > 1)
# FOR TRAITS - clarify quality
traits$aggStatus<-rep(NA, dim(traits)[1])
traits[is.na(traits$traitAggNum), 'aggStatus']<-'NoAgg'
traits[!is.na(traits$traitAggNum), 'aggStatus']<-'Agg'
# FOR TRAITS - reorganize columns
traits.clean<-traits[,c('traitID', #unique row identifiers
                        'paperID', 'obsID', 'aggID', #IDs
                        'traitCat', 'traitUnit', 'stdunit', 'traitVarType', 'traitRef', 'traitImageFile', #IDs
                        'spID', 'aggStatus', #IDs
                        'traitMean', 'traitVar', 'traitN', 'traitVar_VAR', 'stdmean', 'stdvar')] #data
colNums<-which(colnames(traits.clean) %in% c('traitMean', 'traitVar', 'traitN', 'traitVar_VAR', 'stdmean', 'stdvar'))
colnames(traits.clean)[colNums]<-c('mean', 'var', 'n', 'var_VAR', 'mean_std', 'var_std')
# FOR TRAITS - plot histogram for the standardized trait values
pHist_traits_Std<-ggplot(data=traits.clean, aes(x=mean_std, fill=aggStatus)) + mytheme +
  facet_wrap(~traitCat, scales='free', ncol=2) + geom_histogram() +
  scale_y_continuous(expand = c(0,0)) +
  ggtitle('Histogram of species trait values extracted from original papers\nUnits standardized')
pHist_traits_Std
```

Histogram of species trait values extracted from original papers Units standardized



```
# FOR TRAITS - examine outliers
#percN
sub<-subset(traits.clean, traitCat == 'sp_percN' & !is.na(mean_std) & mean_std > 5)
#sub
paste('Excluded percN values greater than 5')
```

```
## [1] "Excluded percN values greater than 5"
```

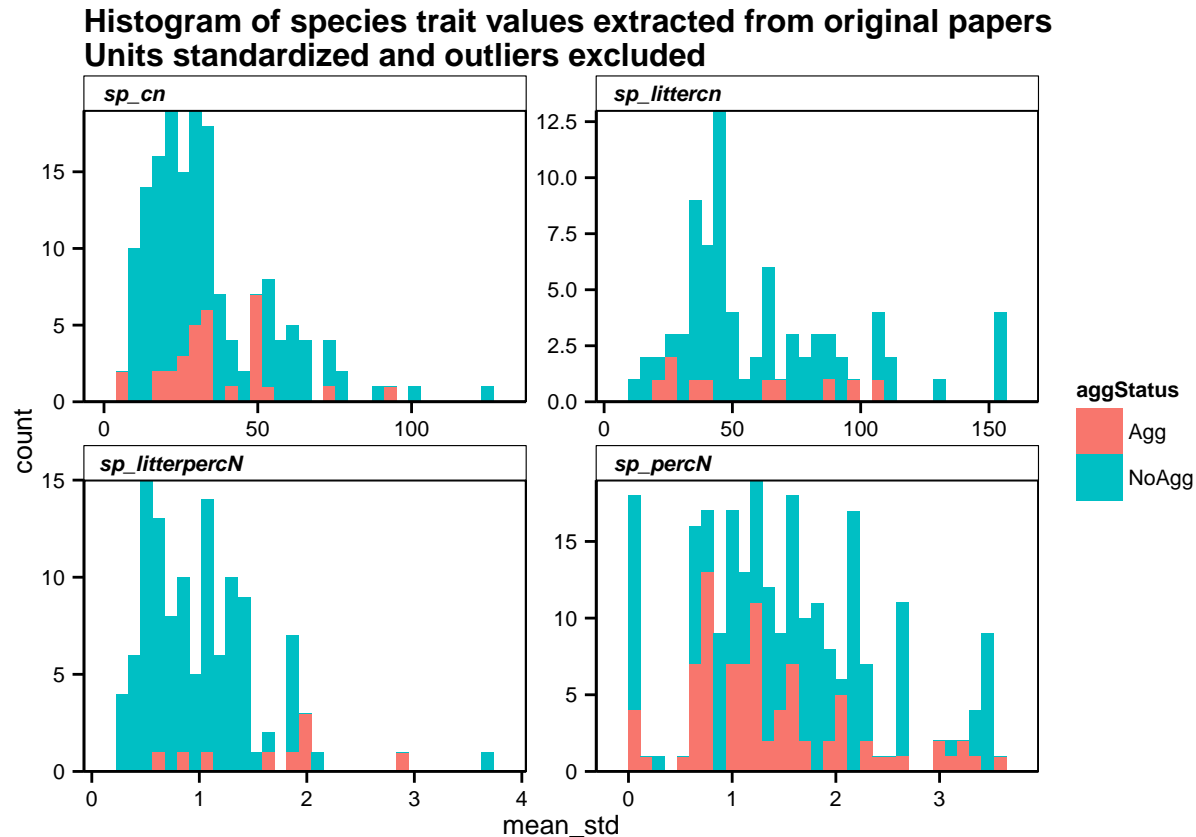
```
traits.clean[traits.clean$traitID %in% sub$traitID & traits.clean$traitCat=='sp_percN',] # exclude
```

```
##      traitID paperID  obsID    aggID traitCat traitUnit stdunit
## 362 222.01.1      222 222.01 222.01.NA sp_percN      g/g      %
## 364 222.01.3      222 222.01 222.01.NA sp_percN      g/g      %
## 671 720.01.1      720 720.01 720.01.1 sp_percN      %      %
## 683 720.02.1      720 720.02 720.02.1 sp_percN      %      %
##      traitVarType traitRef traitImageFile    spID aggStatus mean var n
## 362      <NA>      p.901      <NA> 222.01.2   NoAgg 0.54 NA 1
## 364      <NA>      p.901      <NA> 222.01.1   NoAgg 2.58 NA 1
## 671      Var    Table 2      <NA> 720.01.1     Agg 7.20 NA 4
## 683      Var    Table 2      <NA> 720.02.1     Agg 7.20 NA 4
##      var_VAR mean_std var_std
## 362      NA      54.0      NA
## 364      NA     258.0      NA
## 671 12.85787      7.2 12.85787
## 683 12.85787      7.2 12.85787
```

```

exrows.percN<-which(traits.clean$traitID %in% sub$traitID & traits.clean$traitCat=='sp_percN')
# FOR TRAITS - exclude data where necessary
exrows<-c(exrows.percN)
traits.clean<-traits.clean[-exrows,]
# FOR TRAITS - re-plot histogram for the standardized trait values
pHist_traits_Std_OR<-ggplot(data=traits.clean, aes(x=mean_std,fill=aggStatus)) + mytheme +
  facet_wrap(~traitCat, scales='free', ncol=2) + geom_histogram() +
  scale_y_continuous(expand = c(0,0)) +
  ggtitle('Histogram of species trait values extracted from original papers\nUnits standardized and outliers excluded')
pHist_traits_Std_OR

```

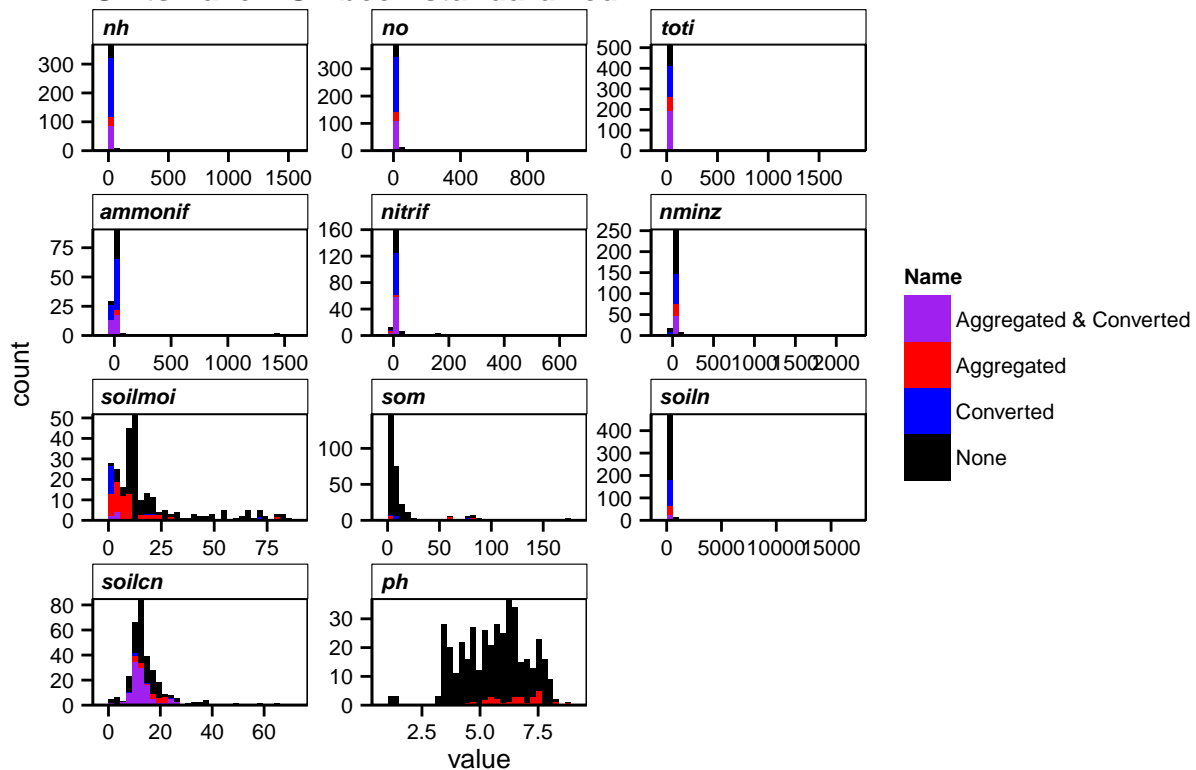


```

#####
# MEASURES (ALL)
#####
# FOR MEASURES - clarify quality
measures$AggYN<-rep("NoAgg", dim(measures)[1]) #No aggregation
measures[!is.na(measures$measAggNum), 'AggYN']<-"Agg" # Aggregation
measures$UnitConvYN<-rep("NoConv", dim(measures)[1]) # No unit conversion
measures[measures$measUnit != as.character(measures$stdunit) & !is.na(measures$stdunit), 'UnitConvYN']<-
measures$YN<-paste(measures$AggYN, measures$UnitConvYN, sep=".")
# FOR MEASURES - streamline 'measures' dataframe
measures.clean<-measures[,c('obsID', 'measEntryID2', 'measCat', 'AggYN', 'UnitConvYN', 'YN',
                             'measUnit', 'stdunit',
                             'measInvMean', 'measNatMean',
                             'measInvVar_VAR', 'measNatVar_VAR',
                             'measInvN', 'measNatN',

```


Histogram of measurement values Units have NOT been standardized



```
#nh: none, good
sub<-subset(meas, measCat == 'nh' & value ==0)
#no
sub<-subset(meas, measCat == 'no' & value < 0)
paste('there are negative values because this is off a resin bag, so, OK')
```

```
## [1] "there are negative values because this is off a resin bag, so, OK"
```

```
#toti: none, good
sub<-subset(meas, measCat == 'toti' & value < 0)
#ammonif
sub<-subset(meas, measCat == 'ammonif')
range(sub$value) #there are some really large values in here, which may be b/c units are not standardiz
```

```
## [1] -28.011 1538.070
```

```
#nitrif
sub<-subset(meas, measCat == 'nitrif')
range(sub$value) #there are some really large values in here, which may be b/c units are not standardiz
```

```
## [1] -2.25 639.59
```

```
#nminz
sub<-subset(meas, measCat == 'nminz')
range(sub$value) #there are some really large values in here, which may be b/c units are not standardiz
```

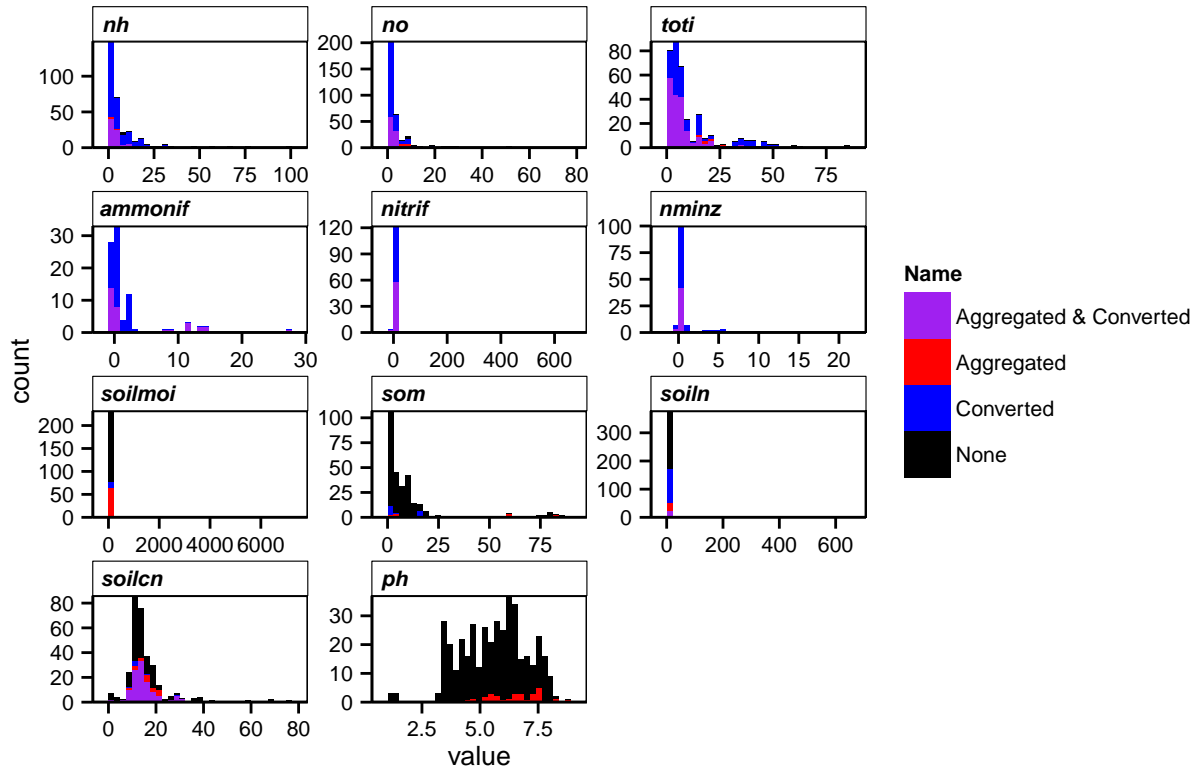
```
## [1] -8.912 2162.440
```

```
#soilmoi: none, good
sub<-subset(meas, measCat == 'soilmoi' & value < 0)
#som: none, good
sub<-subset(meas, measCat == 'som' & value < 0)
sub<-subset(meas, measCat == 'som' & value > 75)
#soiln: none, good
sub<-subset(meas, measCat == 'soiln' & value < 0)
#soilcn: none, good
sub<-subset(meas, measCat == 'soilcn' & value < 0)
#pH - looks fine
# FOR MEASURES (Non-STANDARDIZED UNITS) - exclude data where necessary
NA

#####
# MEASURES (STANDARDIZED UNITS)
#####
# FOR MEASURES (STANDARDIZED UNITS) - reshape the unit-standardized means so that inv and nat area is 1
df1<-measures[,c('obsID', 'measEntryID2', 'measCat', 'YN',
                 'inv_mean_std', 'nat_mean_std')] # standardized means
m.df1<-melt(df1, id=c('obsID', 'measEntryID2', 'measCat', 'YN')) #melt
# FOR MEASURES (STANDARDIZED UNITS) - add column to differentiate between inv and nat
m.df1$invType<-rep(NA, length(dim(m.df1)[1])) #add column to differentiate between inv and nat
m.df1[m.df1$variable=='inv_mean_std', 'invType']<-'inv'
m.df1[m.df1$variable=='nat_mean_std', 'invType']<-'nat'
meas<-m.df1[,c('obsID', 'measEntryID2', 'measCat', 'variable', 'invType', 'YN', 'value')]
# FOR MEASURES (STANDARDIZED UNITS) - plot histograms
df.sub<-subset(meas, measCat %in% c("nh", "no", "toti", "ammonif", "nitrif", "nminz", "soilmoi", "som", "soiln", "soilcn", "pH"))
df.sub <- transform(df.sub, measCat = factor(measCat, levels=c("nh", "no", "toti", "ammonif", "nitrif", "nminz", "soilmoi", "som", "soiln", "soilcn", "pH")))
pHist_meas_Std<-ggplot(data=df.sub, aes(x=value, fill=YN)) + mytheme +
  facet_wrap(~measCat, scales='free', ncol=3) + geom_histogram() +
  scale_y_continuous(expand = c(0,0)) + ggtitle('Histogram of measurement values\nUnits standardized') +
  scale_fill_manual(name = "Name",
                    labels = c("Aggregated & Converted",
                               "Aggregated",
                               "Converted",
                               "None"),
                    values=c("Agg.Conv" = "purple",
                              "Agg.NoConv" = "red",
                              "NoAgg.Conv" = "blue",
                              "NoAgg.NoConv" = "black"))

pHist_meas_Std
```

Histogram of measurement values Units standardized



```
#nh: none, good
sub<-subset(meas, measCat == 'nh' & value < 0)
sub<-subset(meas, measCat == 'nh' & value > 40)
#no: none, good
sub<-subset(meas, measCat == 'no' & value < 0)
sub<-subset(meas, measCat == 'no' & value > 40)
#toti: none, good
sub<-subset(meas, measCat == 'toti' & value < 0)
sub<-subset(meas, measCat == 'toti' & value > 75)
#ammonif: none, good
sub<-subset(meas, measCat == 'ammonif' & value > 10)
#nitrif
sub<-subset(meas, measCat == 'nitrif' & value > 10)
sub #these 2 observations might have very large values because of variation in incubation time length a
```

```
##      obsID measEntryID2 measCat    variable invType      YN    value
## 1308 233.04      233.04.1  nitrif inv_mean_std    inv NoAgg.Conv 150.024
## 3546 233.04      233.04.1  nitrif nat_mean_std    nat NoAgg.Conv 662.040
```

```
paste('Excluded nitrif values greater than 10')
```

```
## [1] "Excluded nitrif values greater than 10"
```

```
measures[measures$measEntryID2 %in% sub$measEntryID2 & measures$measCat=='nitrif',
c('inv_mean_std','nat_mean_std','inv_var_std','nat_var_std')]<-NA # exclude
```

```
#nminz: none, good
sub<-subset(meas, measCat == 'nminz' & value > 10)
#soilmoi
sub<-subset(meas, measCat == 'soilmoi' & value < 0)
sub<-subset(meas, measCat == 'soilmoi' & value > 100)
sub
```

```
##      obsID measEntryID2 measCat    variable invType      YN      value
## 184    29.01      29.01.5 soilmoi inv_mean_std    inv  Agg.Conv  443.3333
## 193    29.02      29.02.5 soilmoi inv_mean_std    inv  Agg.Conv  436.3333
## 1797  484.01      484.01.5 soilmoi inv_mean_std    inv NoAgg.Conv 5900.0000
## 2422   29.01      29.01.5 soilmoi nat_mean_std    nat  Agg.Conv  532.3333
## 2431   29.02      29.02.5 soilmoi nat_mean_std    nat  Agg.Conv  532.3333
## 4035  484.01      484.01.5 soilmoi nat_mean_std    nat NoAgg.Conv 7000.0000
```

```
paste('Excluded soilmoi values greater than 100%')
```

```
## [1] "Excluded soilmoi values greater than 100%"
```

```
measures[measures$measEntryID2 %in% sub$measEntryID2 & measures$measCat=='soilmoi',
          c('inv_mean_std','nat_mean_std','inv_var_std','nat_var_std')]<-NA # exclude...these values mak
#som: none, good
sub<-subset(meas, measCat == 'som' & value < 0)
sub<-subset(meas, measCat == 'som' & value > 75)
#soiln
sub<-subset(meas, measCat == 'soiln' & value < 0)
sub<-subset(meas, measCat == 'soiln' & value > 5)
sub
```

```
##      obsID measEntryID2 measCat    variable invType      YN      value
## 29      6.01      6.01.4  soiln inv_mean_std    inv  NoAgg.Conv  630.992
## 33      6.02      6.02.4  soiln inv_mean_std    inv  NoAgg.Conv  630.992
## 800    82.01      82.01.7  soiln inv_mean_std    inv NoAgg.NoConv   9.400
## 813    82.02      82.02.7  soiln inv_mean_std    inv NoAgg.NoConv  12.600
## 826    82.03      82.03.7  soiln inv_mean_std    inv NoAgg.NoConv  18.000
## 2267   6.01      6.01.4  soiln nat_mean_std    nat  NoAgg.Conv  611.570
## 2271   6.02      6.02.4  soiln nat_mean_std    nat  NoAgg.Conv  611.570
## 3038   82.01      82.01.7  soiln nat_mean_std    nat NoAgg.NoConv   9.800
## 3051   82.02      82.02.7  soiln nat_mean_std    nat NoAgg.NoConv  13.600
## 3064   82.03      82.03.7  soiln nat_mean_std    nat NoAgg.NoConv  19.000
```

```
paste('Excluded soiln values greater than 5%')
```

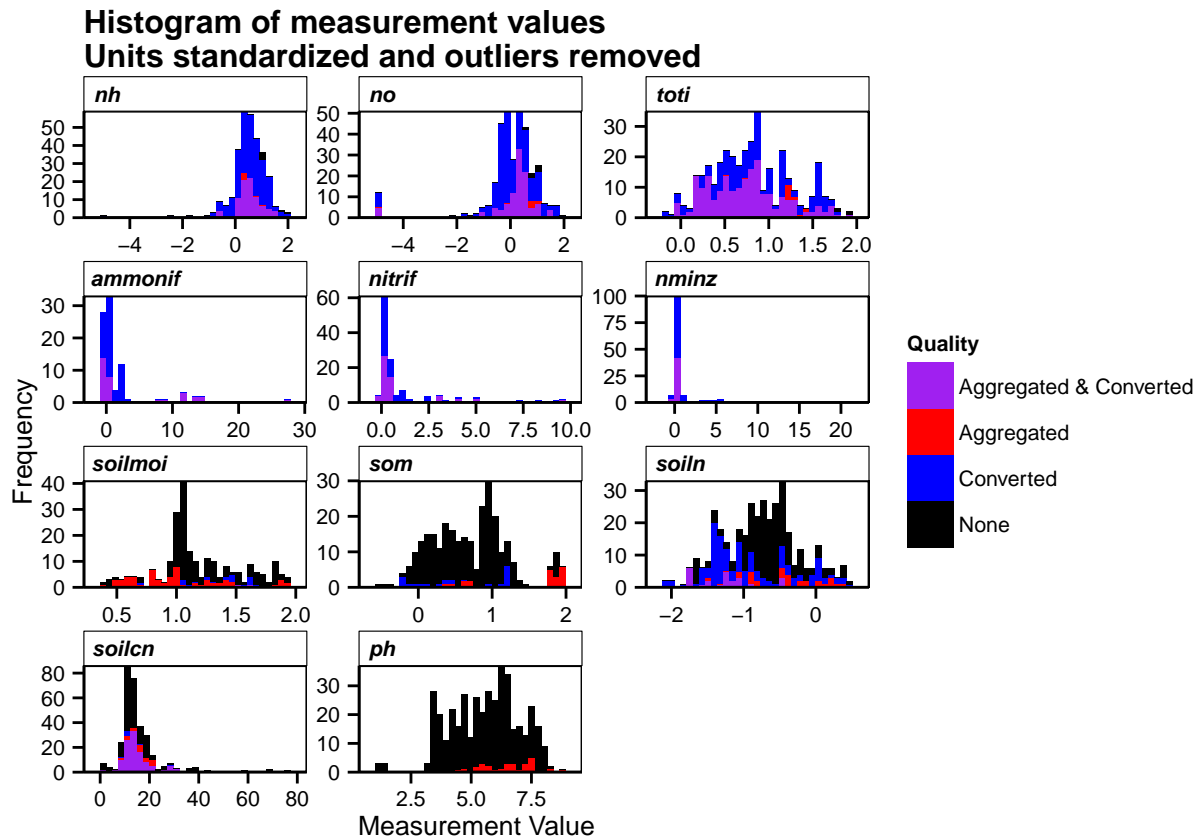
```
## [1] "Excluded soiln values greater than 5%"
```

```
measures[measures$measEntryID2 %in% sub$measEntryID2 & measures$measCat=='soiln',
          c('inv_mean_std','nat_mean_std','inv_var_std','nat_var_std')]<-NA # exclude
#soilcn: none, good
sub<-subset(meas, measCat == 'soilcn' & value < 0)
sub<-subset(meas, measCat == 'soilcn' & value > 60)
#pH: looks fine
```

```

# FOR MEASURES (STANDARDIZED UNITS) - re-do the histograms
df1<-measures[,c('obsID','measEntryID2','measCat','YN',
                'inv_mean_std','nat_mean_std')] # unit-standardized means
m.df1<-melt(df1, id=c('obsID','measEntryID2','measCat','YN')) #melt
# FOR MEASURES (STANDARDIZED UNITS) - add column to differentiate between inv and nat
m.df1$invType<-rep(NA,length(dim(m.df1)[1]))
m.df1[m.df1$variable=='inv_mean_std','invType']<-'inv'
m.df1[m.df1$variable=='nat_mean_std','invType']<-'nat'
meas<-m.df1[,c('obsID','measEntryID2','measCat','variable','invType','YN','value')]
df.sub<-subset(meas, measCat %in% c("nh","no", "toti","ammonif", "nitrif","nminz","soilmoi","som","soilc",
                                     "soilmo"))
df.sub <- transform(df.sub, measCat = factor(measCat, levels=c("nh", "no", "toti","ammonif", "nitrif","nminz",
                                     "soilmoi","som","soilc", "soilmo")))
# FOR MEASURES (STANDARDIZED UNITS) - base10 log-transform the following measures: nh, no, toti, soilmoi, soilc, soilmo
exceptions<-c('ammonif', 'nitrif','nminz','soilcn','ph')
df.sub$value.logt<-log10(df.sub$value)
df.sub[df.sub$measCat %in% exceptions, 'value.logt']<-df.sub[df.sub$measCat %in% exceptions, 'value']
# FOR MEASURES (STANDARDIZED UNITS) - plot histogram
pHist_meas_Std_OR<-ggplot(data=df.sub, aes(x=value.logt,fill=YN)) + mytheme +
  facet_wrap(~measCat, scales='free', ncol=3) + geom_histogram() +
  scale_y_continuous(expand = c(0,0)) +
  scale_fill_manual(name = "Quality",
                    labels = c("Aggregated & Converted",
                               "Aggregated",
                               "Converted",
                               "None"),
                    values=c("Agg.Conv" = "purple",
                              "Agg.NoConv" = "red",
                              "NoAgg.Conv" = "blue",
                              "NoAgg.NoConv" = "black")) +
  xlab('Measurement Value') + ylab('Frequency') +
  ggtitle('Histogram of measurement values\nUnits standardized and outliers removed')
pHist_meas_Std_OR

```



4. EXPORT ALL

```
# PAPERS
newfilename<-'papers_procd.txt'
write.table(papers, file=paste(synthdataPath,newfilename, sep='/'), sep='\t', quote=F) # the quote thing

# OBSERVATIONS
newfilename<-'observations_procd.txt'
write.table(observations1, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')

# SPECIES
newfilename<-'species_procd.txt'
write.table(species, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')

# COVER
newfilename<-'cover_procd.txt'
write.table(cover, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
newfilename<-'pHist_cover_Std.png'
png(paste(figuresPath,newfilename, sep='/'),
     units='in', width = fig.width*1.5, height = fig.height*3, res=fig.res)
pHist_cover_Std
dev.off()
```

```
## pdf
## 2
```

```
# TRAITS
newfilename<-'traits_procd.txt'
write.table(traits.clean, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
newfilename<-'pHist_traits_Std.png'
png(paste(figuresPath,newfilename, sep='/'),
     units='in', width = fig.width*2.5, height = fig.height*2, res=fig.res)
pHist_traits_Std
dev.off()
```

```
## pdf
## 2
```

```
newfilename<-'pHist_traits_Std_OR.png'
png(paste(figuresPath,newfilename, sep='/'),
     units='in', width = fig.width*2.5, height = fig.height*2, res=fig.res)
pHist_traits_Std_OR
dev.off()
```

```
## pdf
## 2
```

```
# MEASURES
newfilename<-'measures_procd.txt'
write.table(measures, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
newfilename<-'pHist_meas_NStd.png'
png(paste(figuresPath,newfilename, sep='/'),
     units='in', width = fig.width*3.5, height = fig.height*4, res=fig.res)
pHist_meas_NStd
dev.off()
```

```
## pdf
## 2
```

```
newfilename<-'pHist_meas_Std.png'
png(paste(figuresPath,newfilename, sep='/'),
     units='in', width = fig.width*3.5, height = fig.height*4, res=fig.res)
pHist_meas_Std
dev.off()
```

```
## pdf
## 2
```

```
newfilename<-'pHist_meas_Std_OR.png'
png(paste(figuresPath,newfilename, sep='/'),
     units='in', width = fig.width*3.5, height = fig.height*4, res=fig.res)
pHist_meas_Std_OR
dev.off()
```

```
## pdf
## 2
```