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. Minimially 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') #this requires ggplot2 and grid
```

```
## Loading required package: ggplot2

## 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

```r
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
```

```r
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

```r
source('CODE/paperData/script_stdVar.R') #TASK= Standarize variance; NEEDS= cover, covAgg, traits, trai
```

2B. AGGREGATE VALUES WITHIN AN OBSERVATION

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

```r
# 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

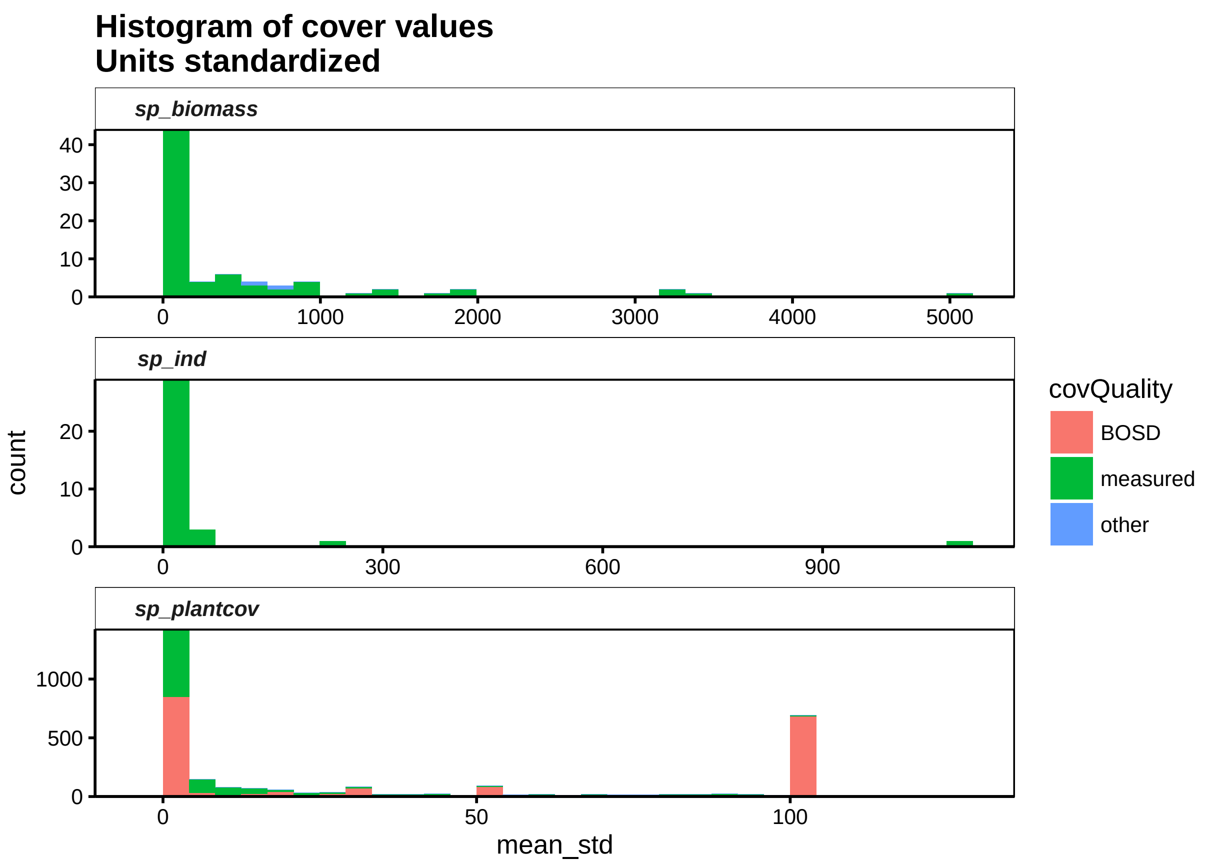
```
##########
# PAPERS
##########
#View(papers)
#length(unique(papers$paperID));dim(papers)# This looks fine for now. Potential issues that might need

##########
# 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
require(reshape2)
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','covQua
                    '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
```

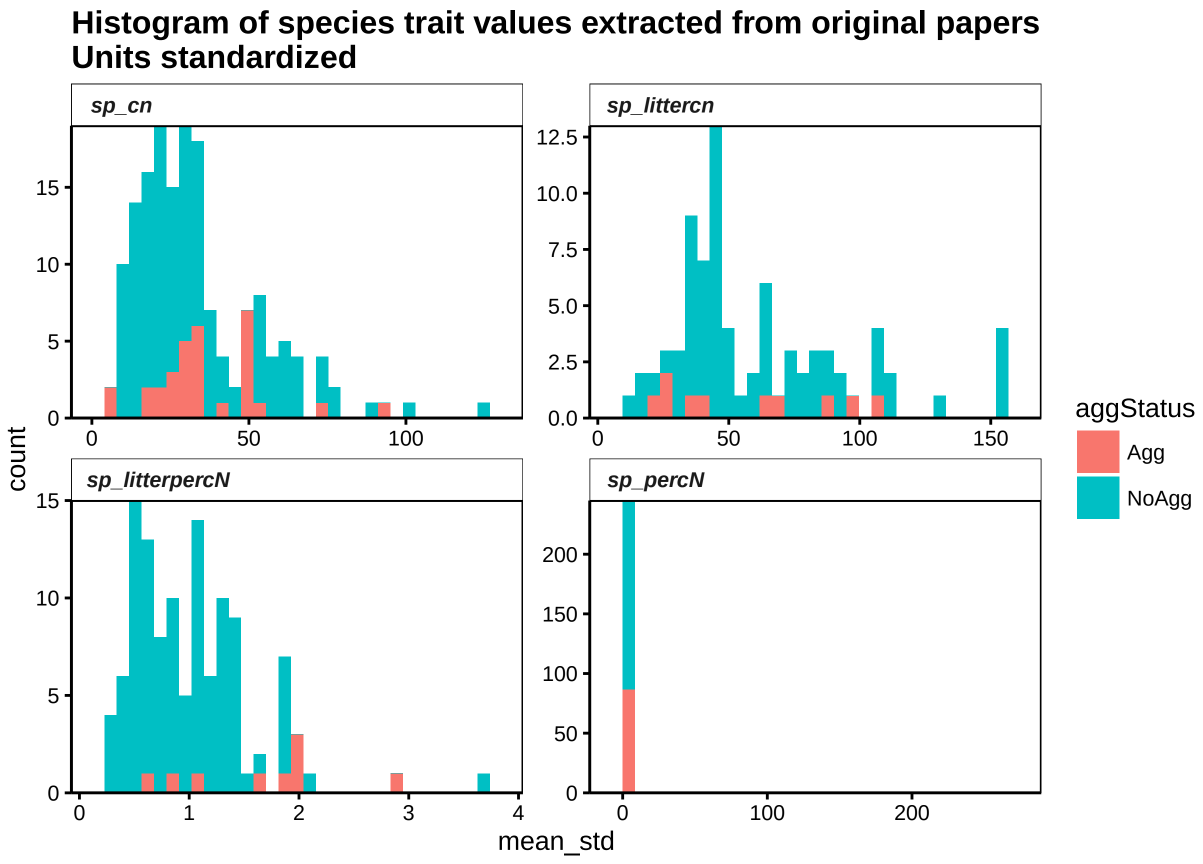


```
#summary of the number obsIDs per each cover measurement type and cover quality type
ddply(cover.clean, ~covCat+covQuality, summarise,
      nObs=length(obsID))
```

```
##        covCat covQuality nObs
## 1  sp_biomass   measured  166
## 2  sp_biomass      other    4
## 3      sp_ind       BOSD   42
## 4      sp_ind   measured  130
## 5 sp_plantcov       BOSD 1960
## 6 sp_plantcov   measured 1048
## 7 sp_plantcov      other   12
```

```
##########
# 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','s
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
```



```
#summary of the number obsIDs per each species' trait measurement type and whether or not it needed to
ddply(traits.clean, ~traitCat+aggStatus, summarise,
      nObs=length(obsID))
```

```
##         traitCat aggStatus nObs
## 1          sp_cn       Agg   31
## 2          sp_cn     NoAgg  133
## 3    sp_littercn       Agg   18
## 4    sp_littercn     NoAgg   69
## 5 sp_litterpercN       Agg   19
## 6 sp_litterpercN     NoAgg  121
## 7       sp_percN       Agg  138
## 8       sp_percN     NoAgg  175
```

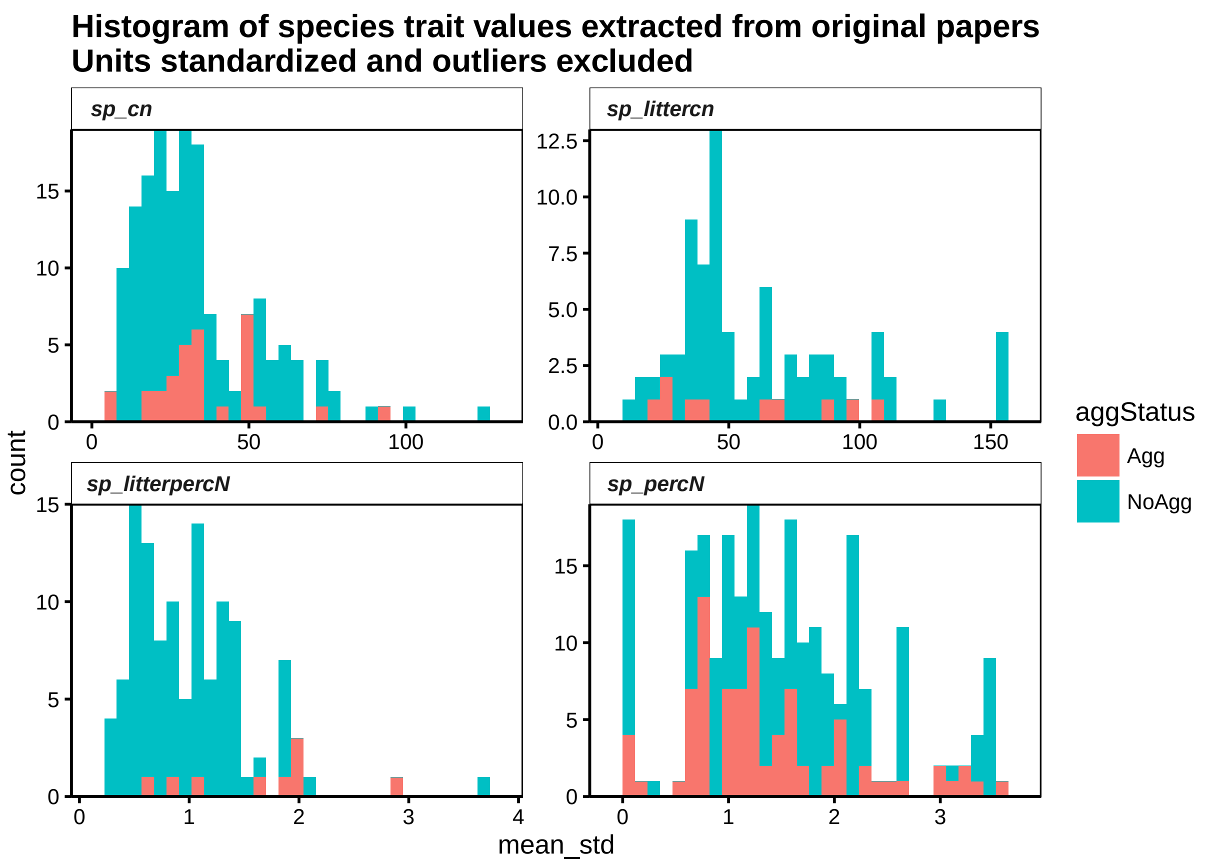
```
# FOR TRAITS - examine outliers
#percN
sub<-subset(traits.clean, traitCat == 'sp_percN' & !is.na(mean_std) & mean_std > 5)
#sub
paste('Excluded percN values greater than 5')
```

```
## [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 out
pHist_traits_Std_OR
```

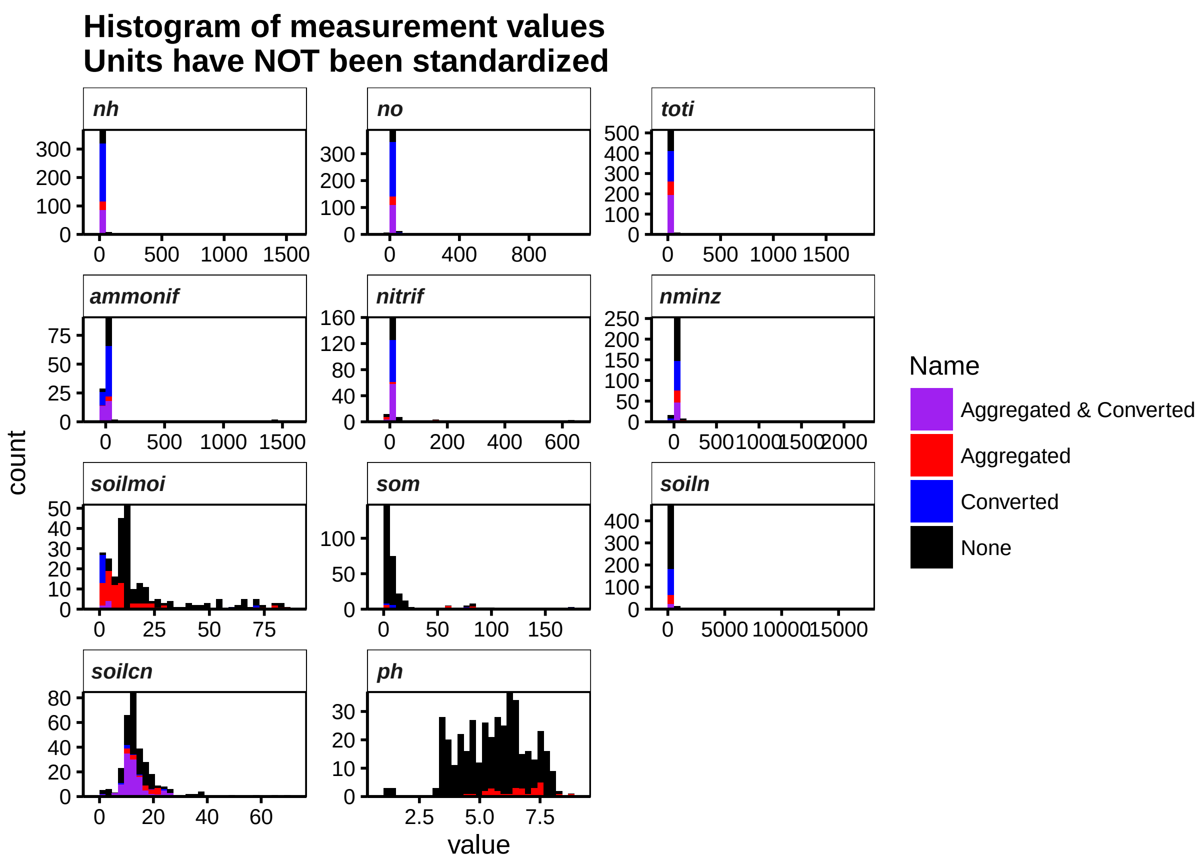


```
#summary of the number obsIDs per each species' trait measurement type and whether or not it needed to
ddply(traits.clean, ~traitCat+aggStatus, summarise,
      nObs=length(obsID))
```

```
##         traitCat aggStatus nObs
## 1          sp_cn       Agg   31
## 2          sp_cn     NoAgg  133
## 3    sp_littercn       Agg   18
## 4    sp_littercn     NoAgg   69
## 5 sp_litterpercN       Agg   19
## 6 sp_litterpercN     NoAgg  121
## 7        sp_percN       Agg  136
## 8        sp_percN     NoAgg  173
```

```
##########
# 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',
                            'stdmeanInv','stdmeanNat',
                            'stdvarInv','stdvarNat')]
colnames(measures.clean)[which(colnames(measures.clean) %in% c('measUnit','stdunit'))]<-c('unit', 'unit_
colnames(measures.clean)[which(colnames(measures.clean) %in% c('measInvMean','measNatMean',
                                                                   'measInvVar_VAR','measNatVar_VAR',
                                                                   'stdmeanInv','stdmeanNat',
                                                                   'stdvarInv','stdvarNat'))]<-c('inv_mean'
                                                                                                 'inv_var',
                                                                                                 'inv_mean_
                                                                                                 'inv_var_s
colnames(measures.clean)[which(colnames(measures.clean) %in% c('measInvN','measNatN'))]<-c('inv_n', 'na
#View(measures.clean)
measures<-measures.clean

##########
# MEASURES (Non-STANDARDIZED UNITS)
##########
# FOR MEASURES (Non-STANDARDIZED UNITS) - reshape the means so that inv and nat area is 1 factor column
df1<-measures[,c('obsID','measEntryID2','measCat','YN',
           'inv_mean','nat_mean')] # non-standardized means
m.df1<-melt(df1, id=c('obsID','measEntryID2','measCat','YN')) #melt
## FOR MEASURES (Non-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','invType']<-'inv'
m.df1[m.df1$variable=='nat_mean','invType']<-'nat'
meas<-m.df1[,c('obsID','measEntryID2','measCat','variable','invType','YN','value')]
# FOR MEASURES (Non-STANDARDIZED UNITS) - plot histograms
df.sub<-subset(meas, measCat %in% c("nh","no", "toti","ammonif", "nitrif","nminz","soilmoi","som","soil
df.sub <- transform(df.sub, measCat = factor(measCat, levels=c("nh", "no", "toti","ammonif", "nitrif","
pHist_meas_NStd<-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 have NOT been 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_NStd
```



```
#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
```

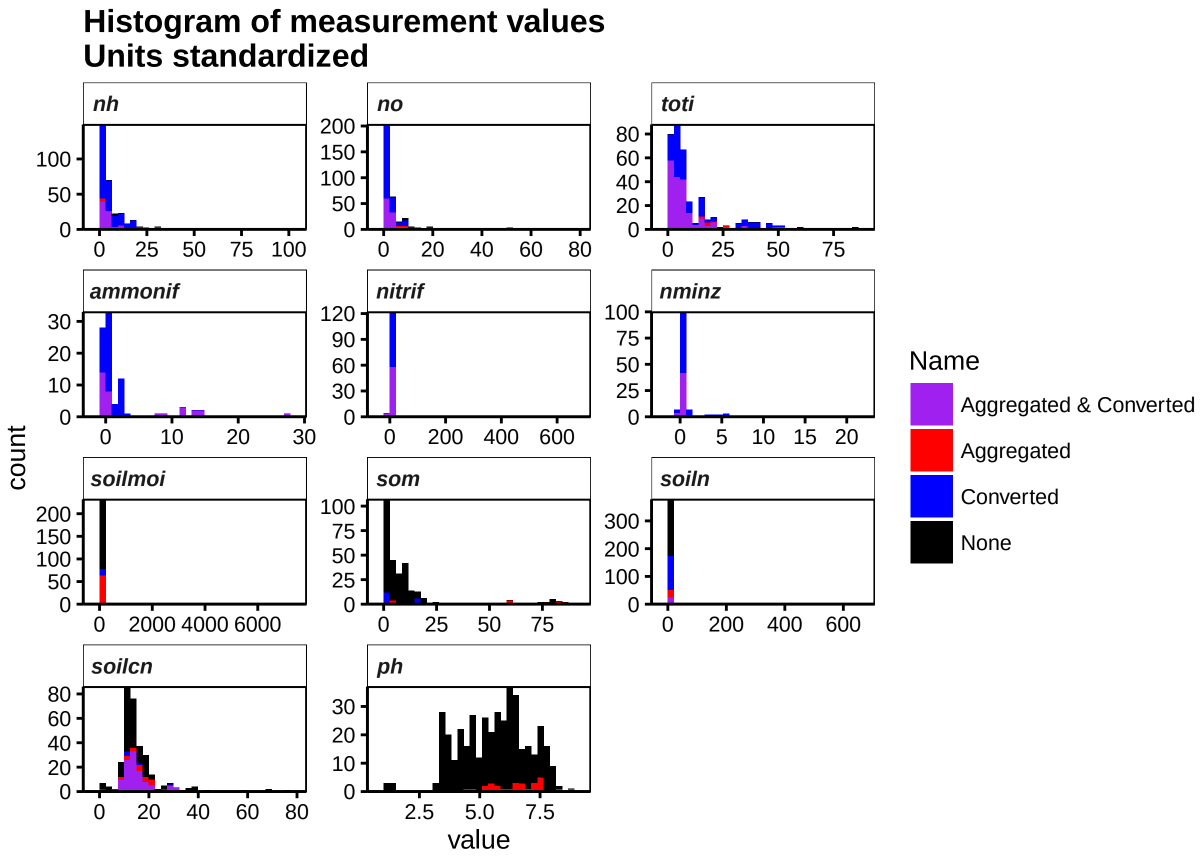
```
## [1]  -2.25 639.59
```

```
#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","soil
df.sub <- transform(df.sub, measCat = factor(measCat, levels=c("nh", "no", "toti","ammonif", "nitrif","
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
```



```r
#summary of the number obsIDs per each
ddply(df.sub, ~measCat+YN, summarise,
      nObs=length(obsID))
```

```
##     measCat           YN nObs
## 1        nh     Agg.Conv   86
## 2        nh   Agg.NoConv   74
## 3        nh   NoAgg.Conv  208
## 4        nh NoAgg.NoConv   56
## 5        no     Agg.Conv  110
## 6        no   Agg.NoConv   78
## 7        no   NoAgg.Conv  210
## 8        no NoAgg.NoConv   58
## 9      toti     Agg.Conv  196
## 10     toti   Agg.NoConv   76
## 11     toti   NoAgg.Conv  150
## 12     toti NoAgg.NoConv  114
## 13 ammonif     Agg.Conv   32
## 14 ammonif   Agg.NoConv    4
## 15 ammonif   NoAgg.Conv   56
## 16 ammonif NoAgg.NoConv   34
## 17  nitrif     Agg.Conv   62
## 18  nitrif   Agg.NoConv   12
## 19  nitrif   NoAgg.Conv   66
## 20  nitrif NoAgg.NoConv   50
## 21   nminz     Agg.Conv   52
## 22   nminz   Agg.NoConv   40
## 23   nminz   NoAgg.Conv   74
## 24   nminz NoAgg.NoConv  122
## 25 soilmoi     Agg.Conv    6
## 26 soilmoi   Agg.NoConv   72
## 27 soilmoi   NoAgg.Conv   16
## 28 soilmoi NoAgg.NoConv  162
## 29     som   Agg.NoConv   20
## 30     som   NoAgg.Conv   18
## 31     som NoAgg.NoConv  256
## 32   soiln     Agg.Conv   24
## 33   soiln   Agg.NoConv   42
## 34   soiln   NoAgg.Conv  126
## 35   soiln NoAgg.NoConv  316
## 36  soilcn     Agg.Conv  120
## 37  soilcn   Agg.NoConv   22
## 38  soilcn   NoAgg.Conv    6
## 39  soilcn NoAgg.NoConv  162
## 40      ph   Agg.NoConv   28
## 41      ph NoAgg.NoConv  386
```

```
#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
```

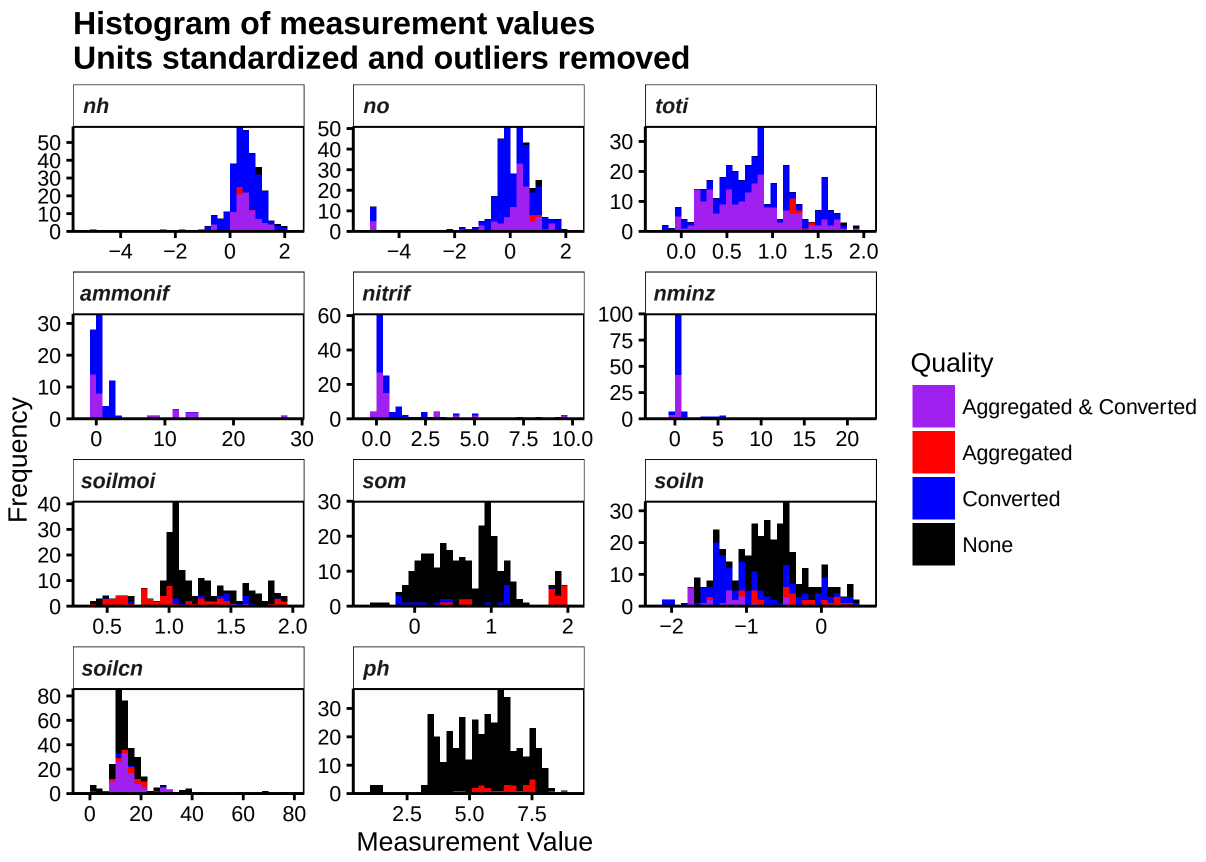
```
##      obsID measEntryID2 measCat     variable invType           YN   value
## 29    6.01       6.01.4   soiln inv_mean_std     inv   NoAgg.Conv 630.992
## 33    6.02       6.02.4   soiln inv_mean_std     inv   NoAgg.Conv 630.992
## 800  82.01      82.01.7   soiln inv_mean_std     inv NoAgg.NoConv   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'
```

```r
m.df1[m.df1$variable=='nat_mean_std','invType']<-'nat'
meas<-m.df1[,c('obsID','measEntryID2','measCat','variable','invType','YN','value')]
df.sub<-subset(meas, measCat %in% c("nh","no", "toti","ammonif", "nitrif","nminz","soilmoi","som","soil
df.sub <- transform(df.sub, measCat = factor(measCat, levels=c("nh", "no", "toti","ammonif", "nitrif","
# FOR MEASURES (STANDARDIZED UNITS) - base10 log-transform the following measures: nh, no, toti, soilmo
exceptions<-c('ammonif', 'nitrif','nminz','soilcn','ph')
df.sub$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
```



```r
#summary of the number obsIDs per each
ddply(df.sub, ~measCat, summarise,
      nObs=length(obsID))
```

```
##    measCat nObs
## 1       nh  424
## 2       no  456
## 3     toti  536
## 4  ammonif  126
## 5   nitrif  190
## 6    nminz  288
## 7  soilmoi  256
## 8      som  294
## 9    soiln  508
## 10  soilcn  310
## 11      ph  414
```

---

4. EXPORT ALL

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

# 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')

# TRAITS
newfilename<-'traits_procd.txt'
write.table(traits.clean, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')

# MEASURES
newfilename<-'measures_procd.txt'
write.table(measures, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
```