

# OKUM Certification

*TCM*

*Tuesday, July 15, 2014*

## Automated analysing submitted data for OKUM based on defined outliers

Thomas Meisel (2014-08-03)

defining the RM and measurand to be analysed

```
refmat <- 'OKUM' # defining the RM
```

general comments to the design

The data for this interlaboratory comparison based certification of property values were analysed by 36 labs following the nested design approached as proposed the IAG certification protocol. Participating labs received 3 packages of OKUM and MUH-1 respectively and one package of GAS. The latter was supplied as a “traceability” sample and is here used for quality control purposes. It was the task of the labs to prepare two independent sample preparations (i.e. digestions) of each packet and analyse the preparations on two different days. Labs thus should have submitted 12 values (3x2x2 PacketxPrepxDay). The outliers have been selected based in Youden plots, Mandel’s k and detection limit criteria. In this file the property values and the uncertainties are calculated for all analytes of a specific candidate OKUM

```
'%p%' <- function(x, y) {as.character(paste(x, y, sep = ""))}  
df <- data.frame(cbind(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0))  
names(df) <- c("date", "RM", "measurand", "mean.before", "mean.after", "median.before", "median.after",  
df <- df[!1,]  
write.table(df, "df0.txt", row.names=FALSE) # needed only the first time
```

```
# Data for certification project was gathered and joined in Excel. The files  
# were exported from Excel as xxxx.csv files to make them universally  
# readable. For this markdown the data is stored in the 'root/documents'  
# directory. Data is loaded ('GOMGather1.R') and merged ('GOMMerge.R') for  
# GAS, OKUM and MUH-1 are merged together with a methods file  
# ('OKUM.method') into a universal data.frame file named 'GOM'. All of this  
# happens in the 'Makefile.R'
```

```
source("Makefile.R")
```

importing the data and assigning factors

defining the function for plotting methods vs. measurand mass fraction. Sample preparation methods are also marked in the plot.

defining the function of Youden plots

```
## means over packets within lab
meanGOM <- function(x) mean(x, na.rm=TRUE) # defining a function for further calcuations
sdGOM <- function(x) sd(x, na.rm=TRUE) # defining a function for further calcuations, here for calculate
meanGOM.packet <- ddply(GOM, c("Lab", "Packet"), numcolwise(meanGOM)) # calculated the mean for each Pa
## mean over mean of packets within lab
GOM.mean <- ddply(meanGOM.packet, c("Lab")), numcolwise(meanGOM))
GOM.mean <- merge(GOM.mean, OKUM.methods, by="Lab")
```

```
## median over packets within lab
medianGOM <- function(x) median(x, na.rm=TRUE)
medianGOM.packet <- ddply(GOM, c("Lab", "Packet"), numcolwise(medianGOM))
GOM.sd <- ddply(medianGOM.packet, c("Lab"), numcolwise(sdGOM))
## median over median of packets within lab
GOM.median <- ddply(medianGOM.packet, c("Lab"), numcolwise(medianGOM))
GOM.median <- merge(GOM.median, OKUM.methods, by="Lab")
```

initial calculations with complete data set

plots before outlier removal and outlier removal

```
sequence <- seq(from = 1, to = length(names(OKUM.outlier)), by = 3)
col <- OKUM.outlier[,c(sequence)]
col.names <- colnames(col)
for (m in col.names) {
  measurand.name <- m
  switch(
    reformat,
    GAS = rm1 <- 2,
    MUH = rm1 <- 1,
    OKUM = rm1 <- 0
  )
  if(rm1 > 0)
  {measurand <- measurand.name %p% '.' %p% rm1
  } else
  {
    measurand <- measurand.name
  }
  MorT <- grep(measurand.name, colnames(GOM), fixed=TRUE) # finding the position of the measurand.name
  ifelse(MorT[1]< 21, MorT <- 'M', MorT<-'T') # testing if measurand is a major or trace element/componu
  ifelse(MorT == "T", unit <- 'mg/kg', unit <- 'g/100g') # testing which unit is needed

print(plot_method(measurand))

# outlier removal
```

```

outlier <- OKUM.outlier[[measurand.name]]
#outlier <- ifelse(OKUM.outlier[[measurand.name]][1]=="NA", "0", OKUM.outlier[[measurand.name]]) # does
outlier <- na.omit(outlier)
leng <- length(outlier) ## counting the number of outliers for loop
for(i in seq(leng)) ## looping
{
  GOM[[measurand]] <- ifelse(GOM$Lab==outlier[i], NA, GOM[[measurand]]) ## replacing values of outlying
  message("Lab ", outlier[i], " was removed")
# print(summary(GOM[[measurand]], na.rm=TRUE, digits=4))
}
}

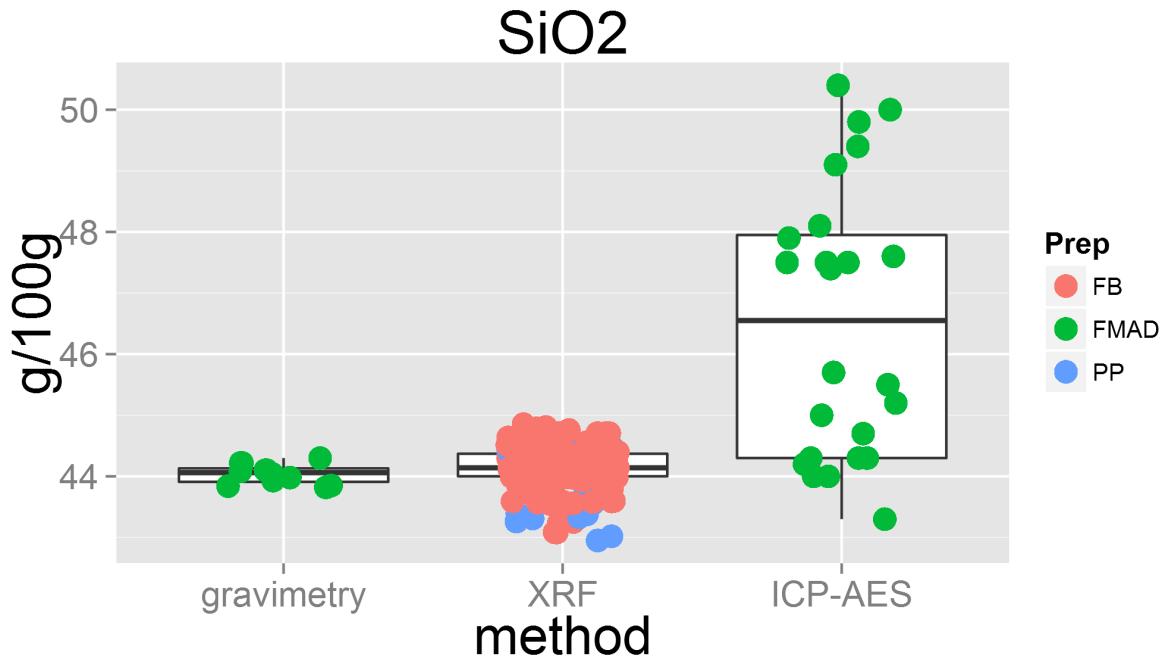
```

```

## Warning: Removed 12 rows containing missing values (geom_point).

## Lab 16 was removed

```



```

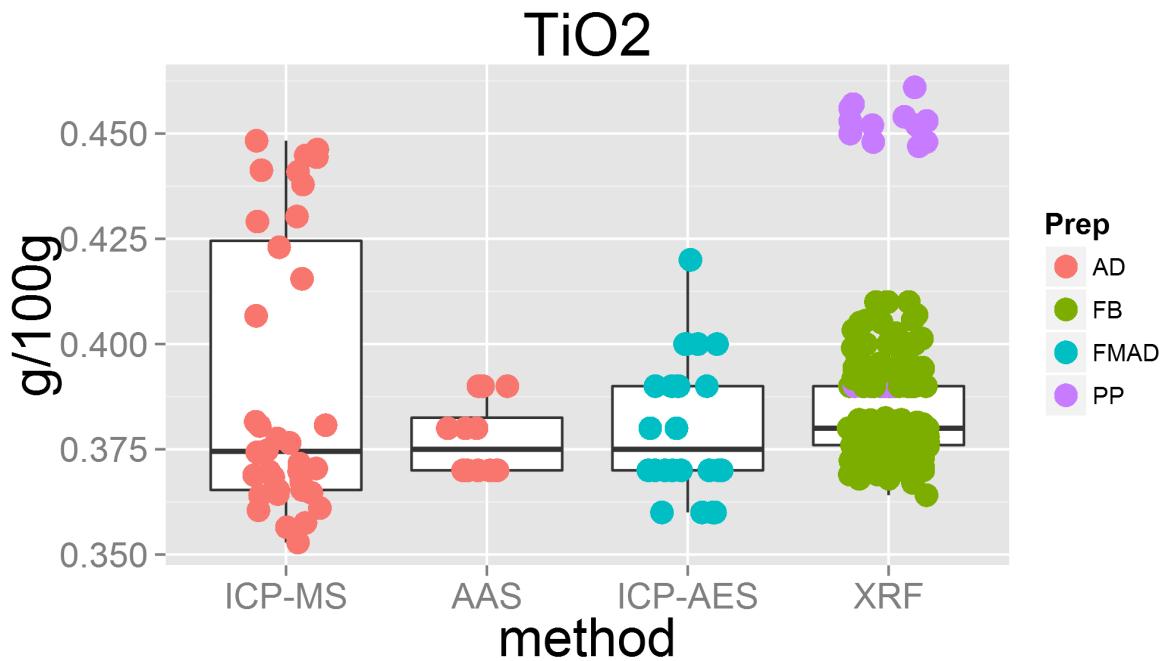
## Warning: Removed 12 rows containing missing values (geom_point).

```

```

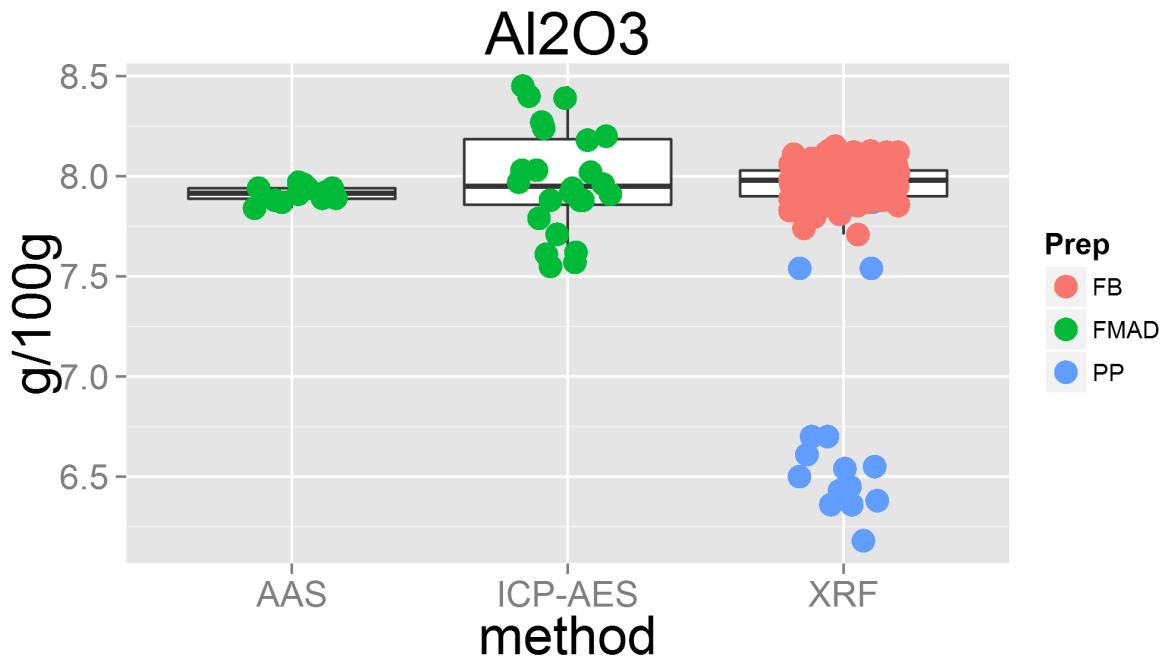
## Lab 12 was removed
## Lab 19 was removed
## Lab 23 was removed

```



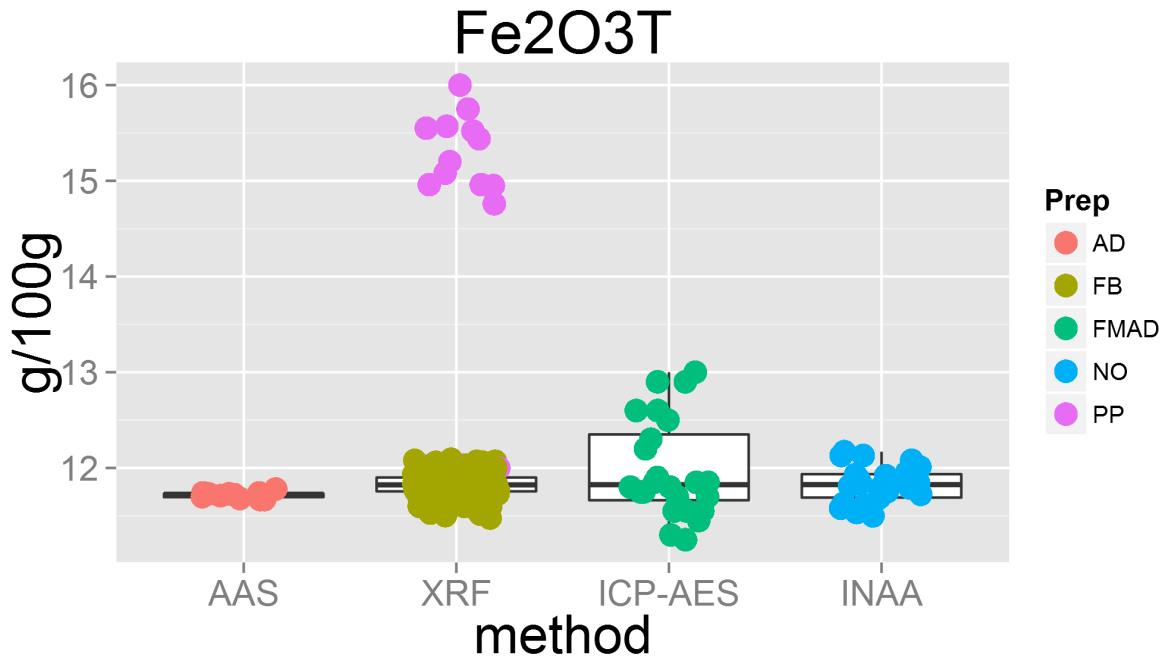
```
## Warning: Removed 14 rows containing missing values (geom_point).
```

```
## Lab 12 was removed
```

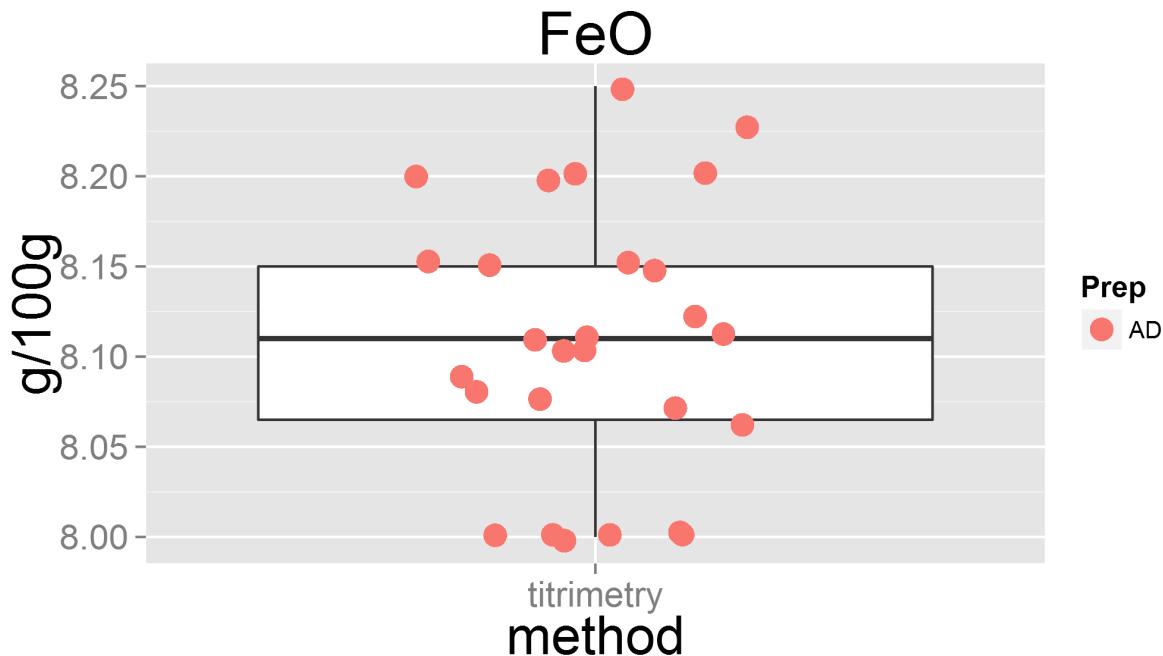


```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Lab 12 was removed  
## Lab 16 was removed  
## Lab 14 was removed
```



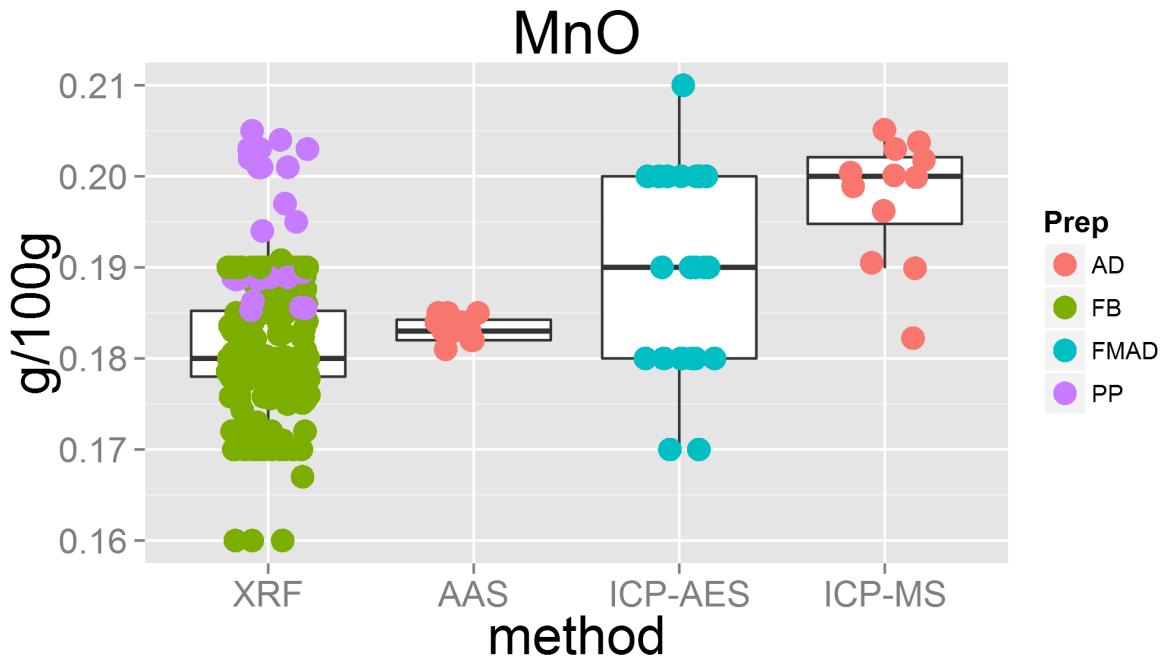
```
## Lab 0 was removed
```



```
## Warning: Removed 14 rows containing missing values (geom_point).
```

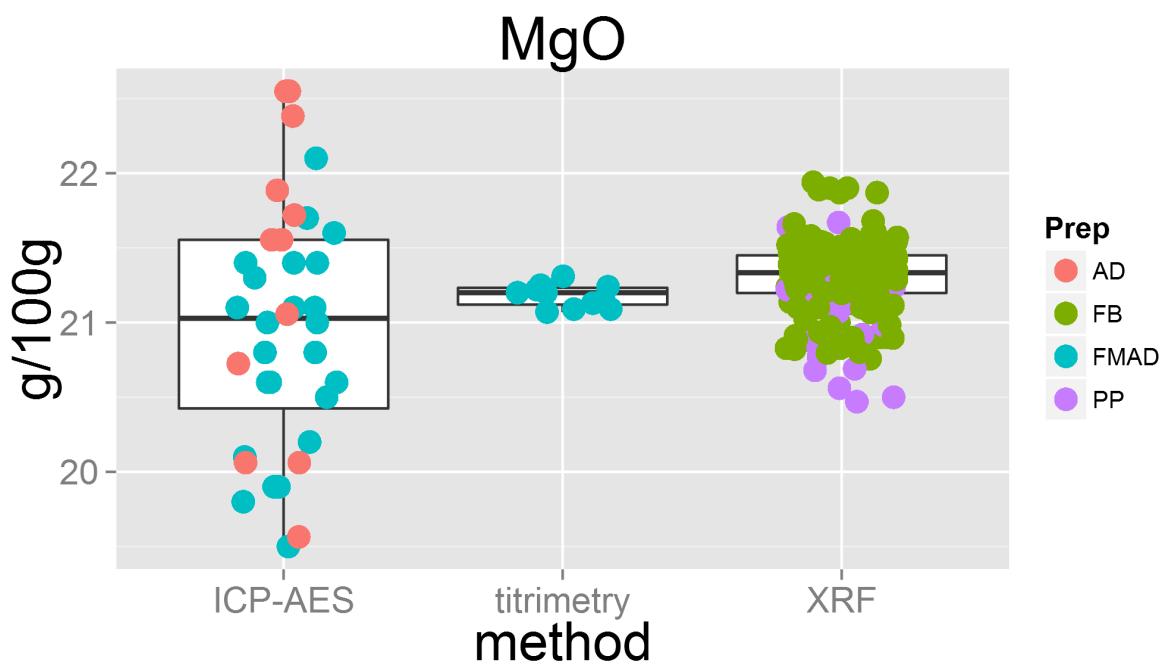
```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Lab 12 was removed  
## Lab 16 was removed
```



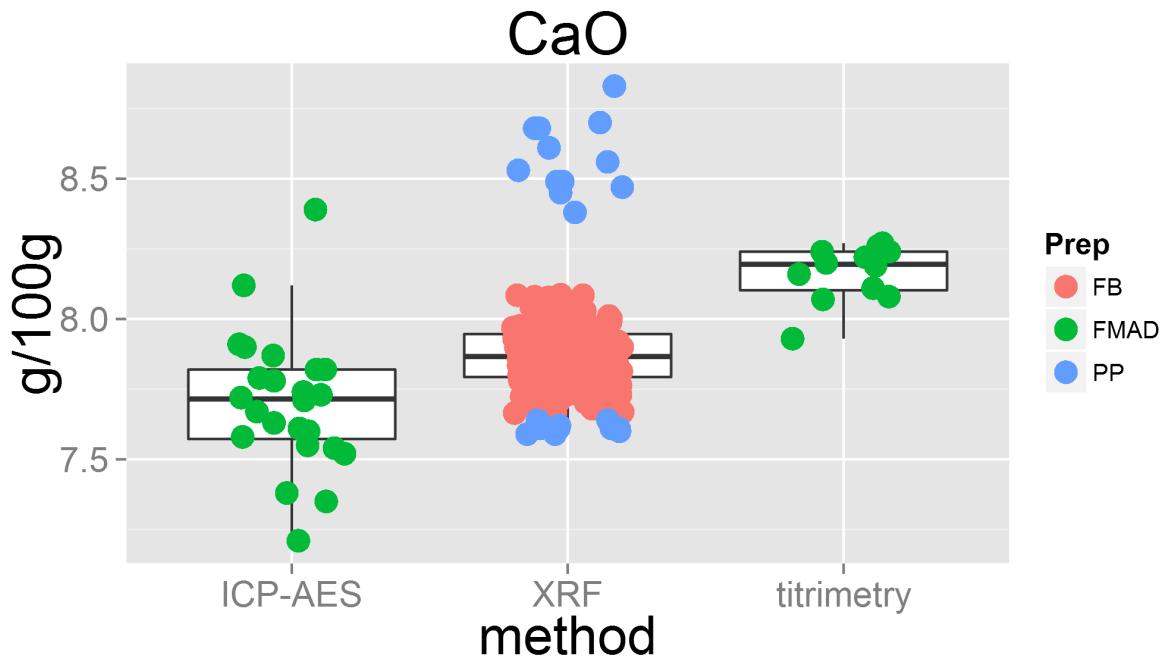
```
## Warning: Removed 15 rows containing missing values (geom_point).
```

```
## Lab 12 was removed  
## Lab 16 was removed  
## Lab 32 was removed
```



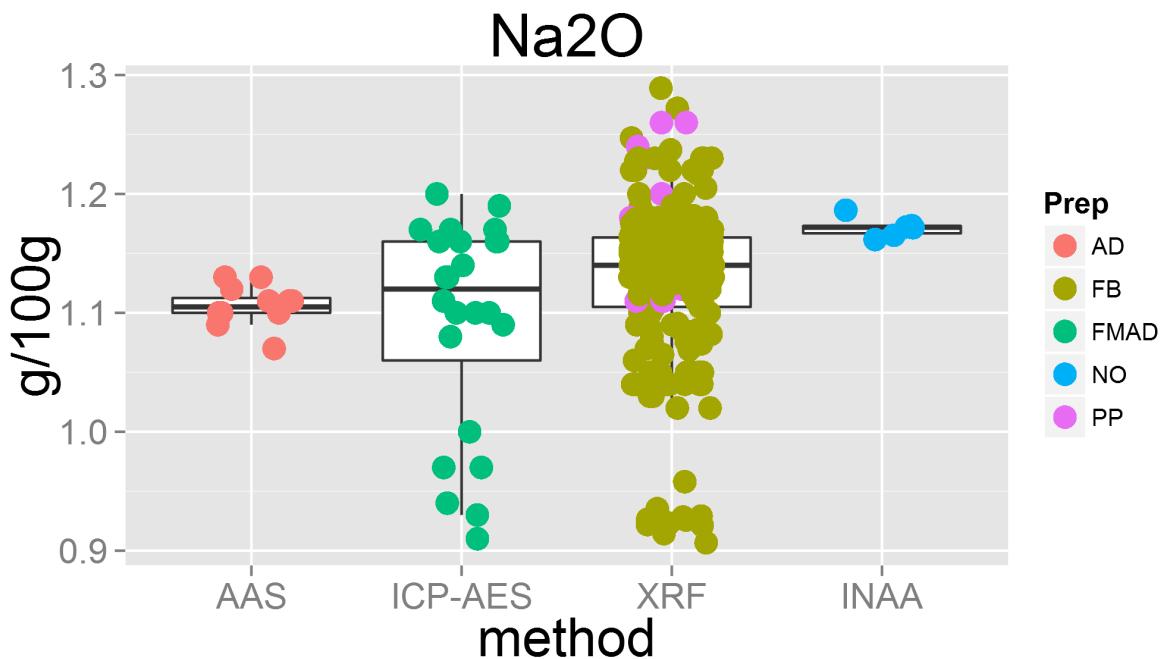
```
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 12 rows containing missing values (geom_point).
```

```
## Lab 12 was removed
## Lab 16 was removed
## Lab 30 was removed
```



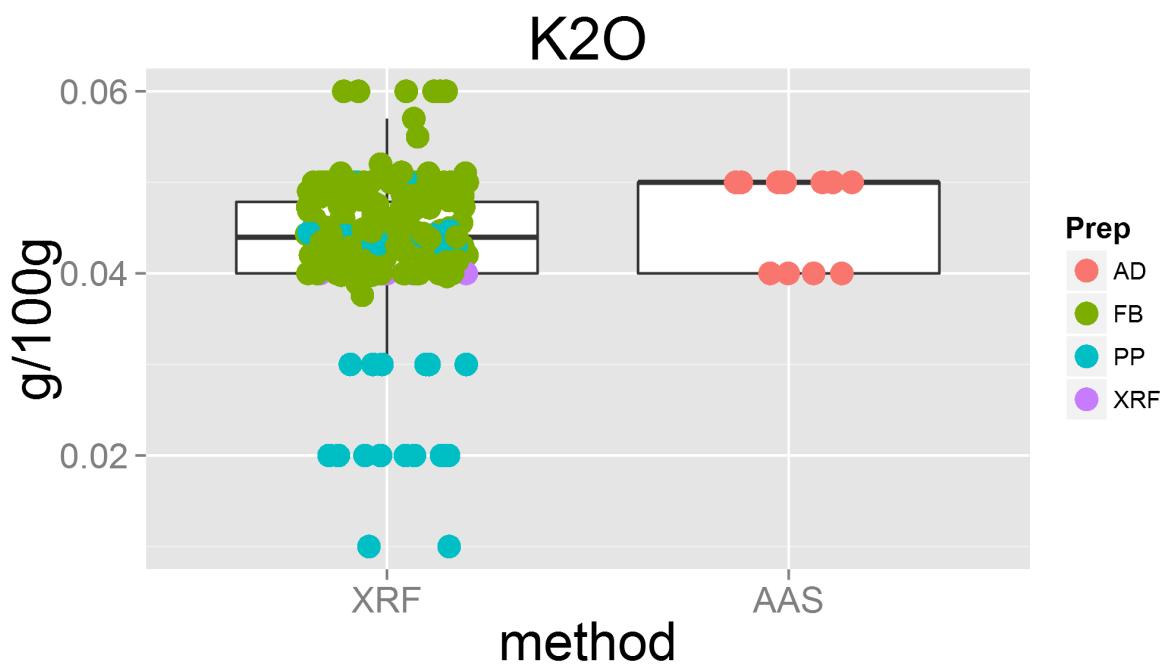
```
## Warning: Removed 1 rows containing missing values (geom_point).  
## Warning: Removed 1 rows containing missing values (geom_point).  
## Warning: Removed 16 rows containing missing values (geom_point).  
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Lab 23 was removed
```



```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Lab 4 was removed  
## Lab 12 was removed
```

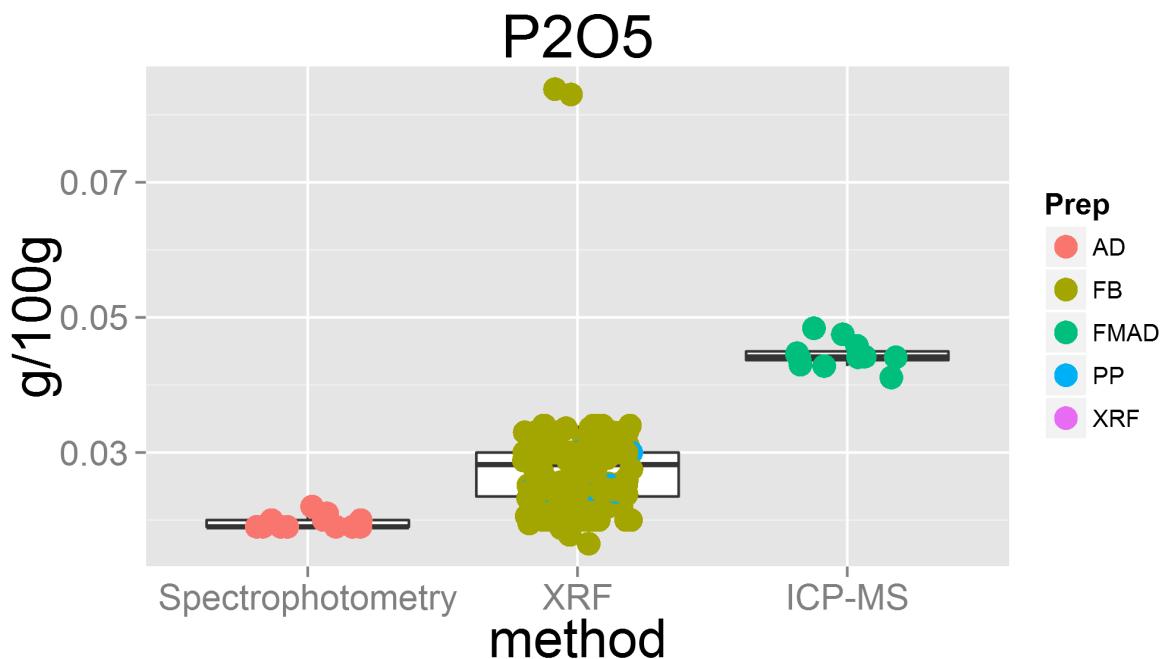


```

## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_point).
## Warning: Removed 3 rows containing missing values (geom_point).

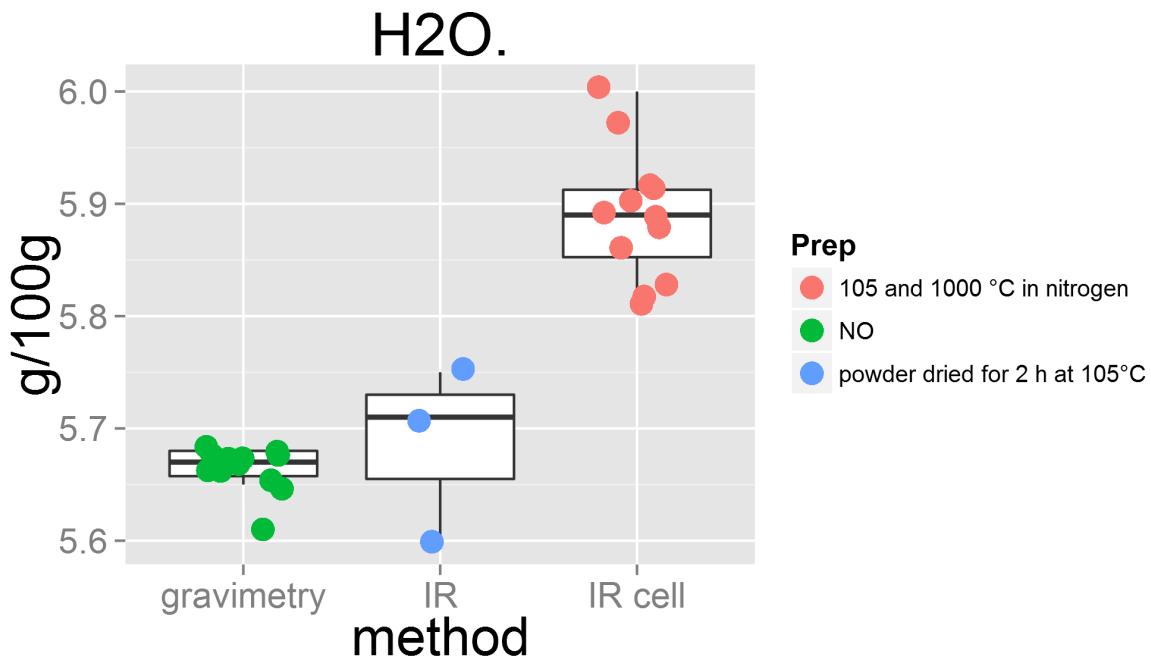
## Lab 31 was removed
## Lab 33 was removed

```



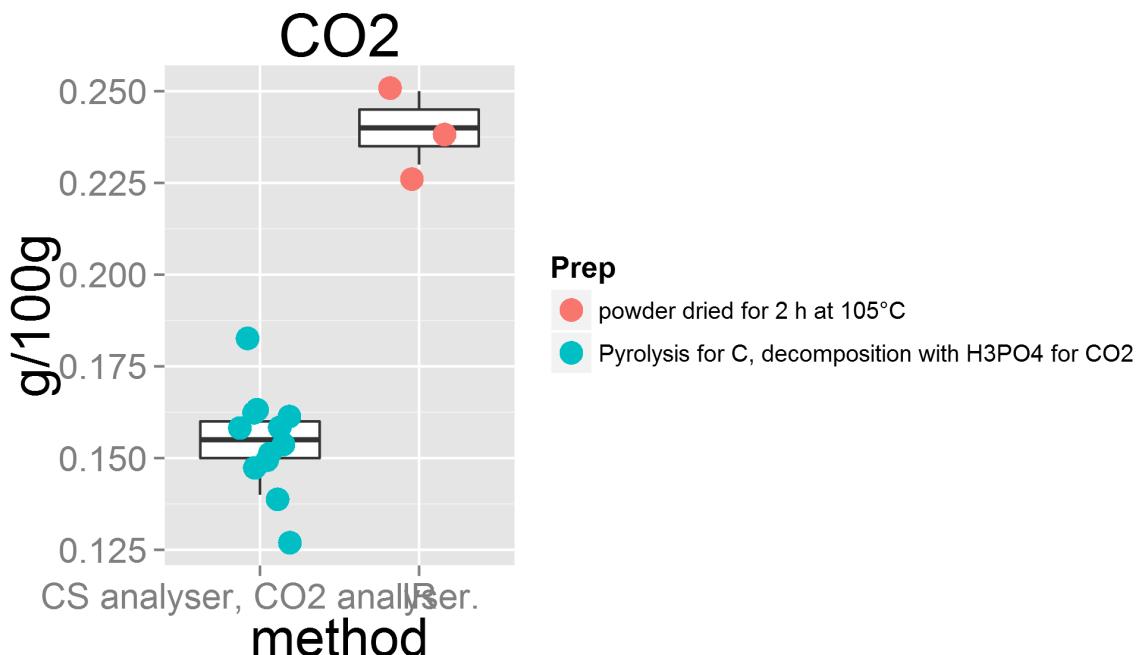
```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



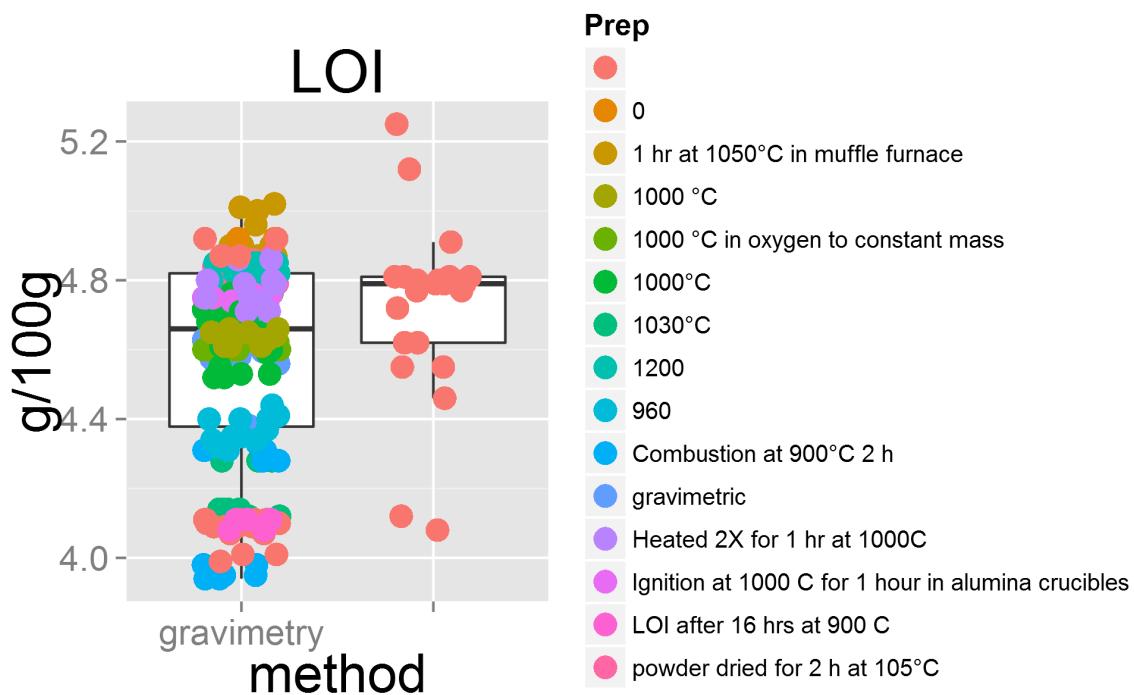
```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```

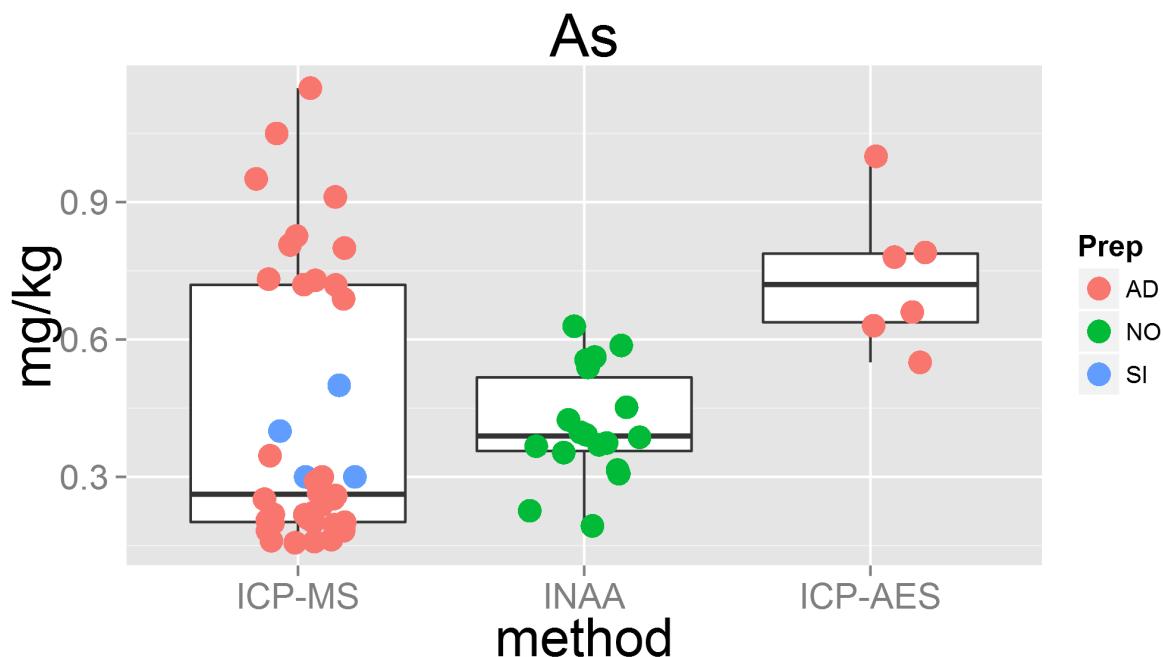


```
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



```
## Lab 0 was removed
```

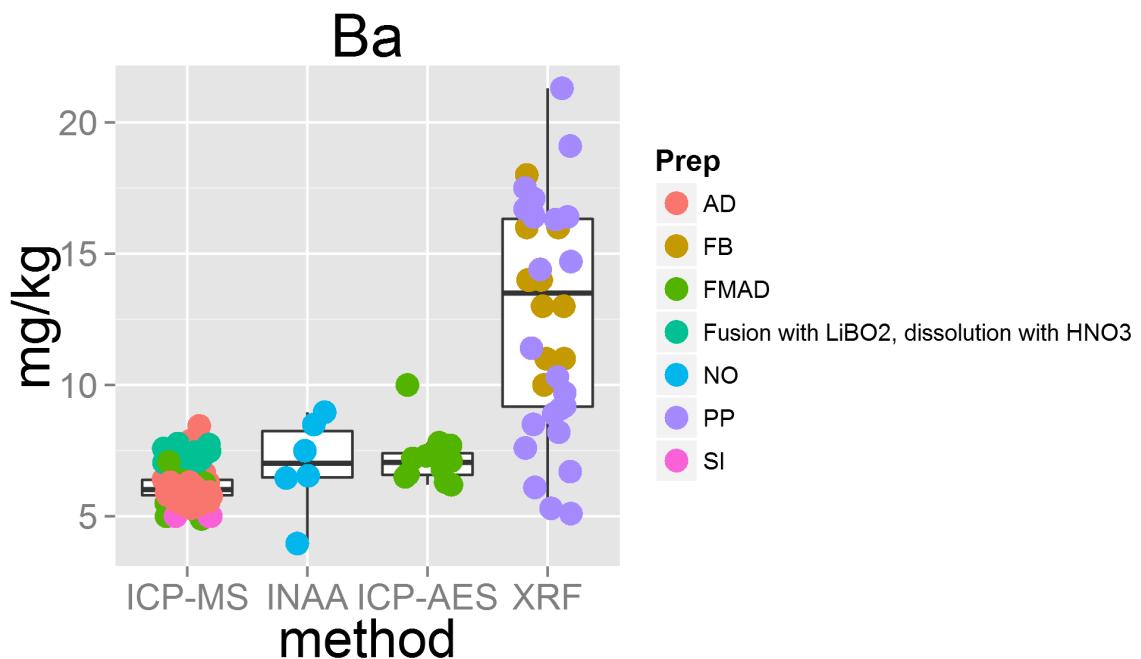


```

## Warning: Removed 14 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).

## Lab 14 was removed
## Lab 26 was removed

```

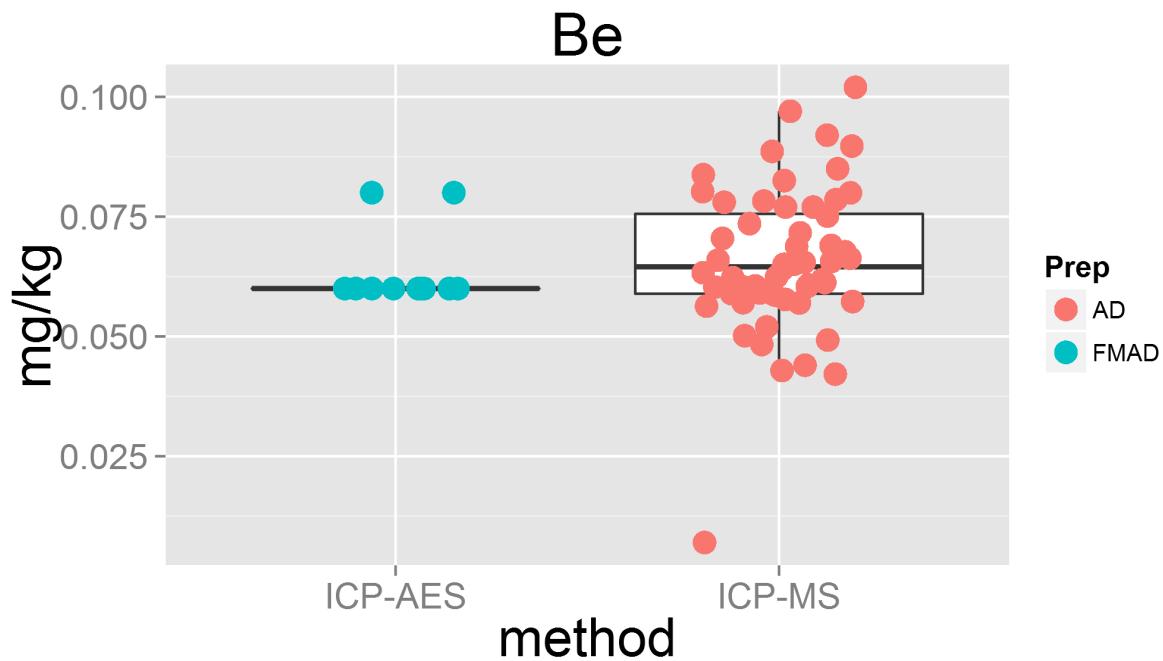


```

## Warning: Removed 2 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_point).

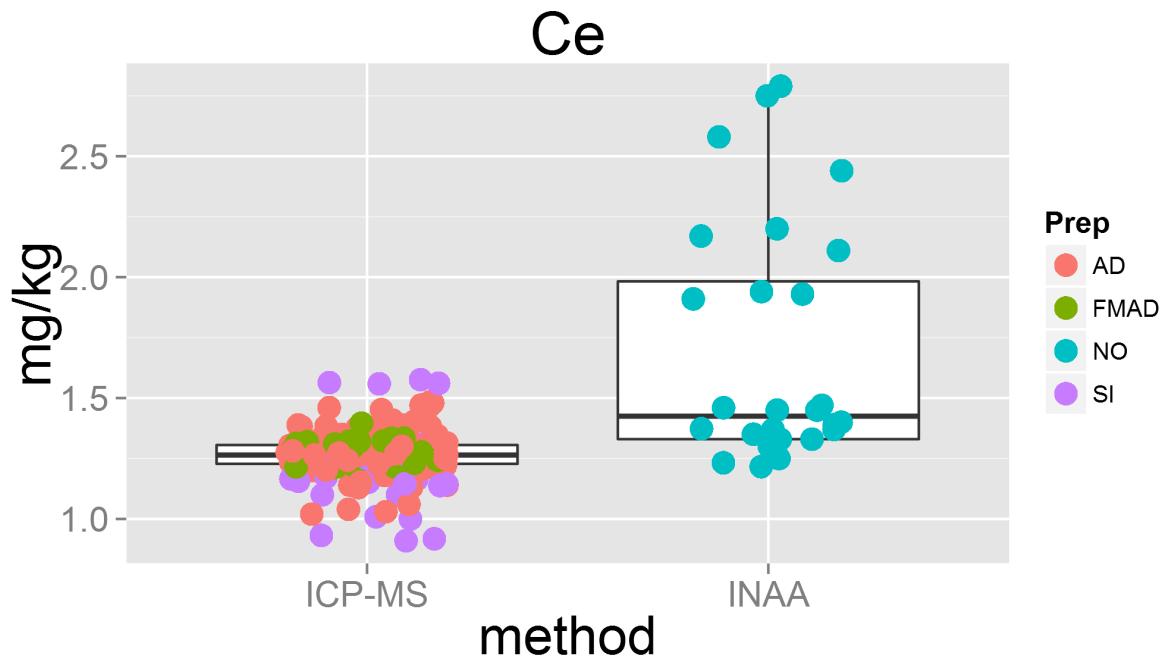
## Lab 0 was removed

```



```
## Warning: Removed 21 rows containing missing values (geom_point).
```

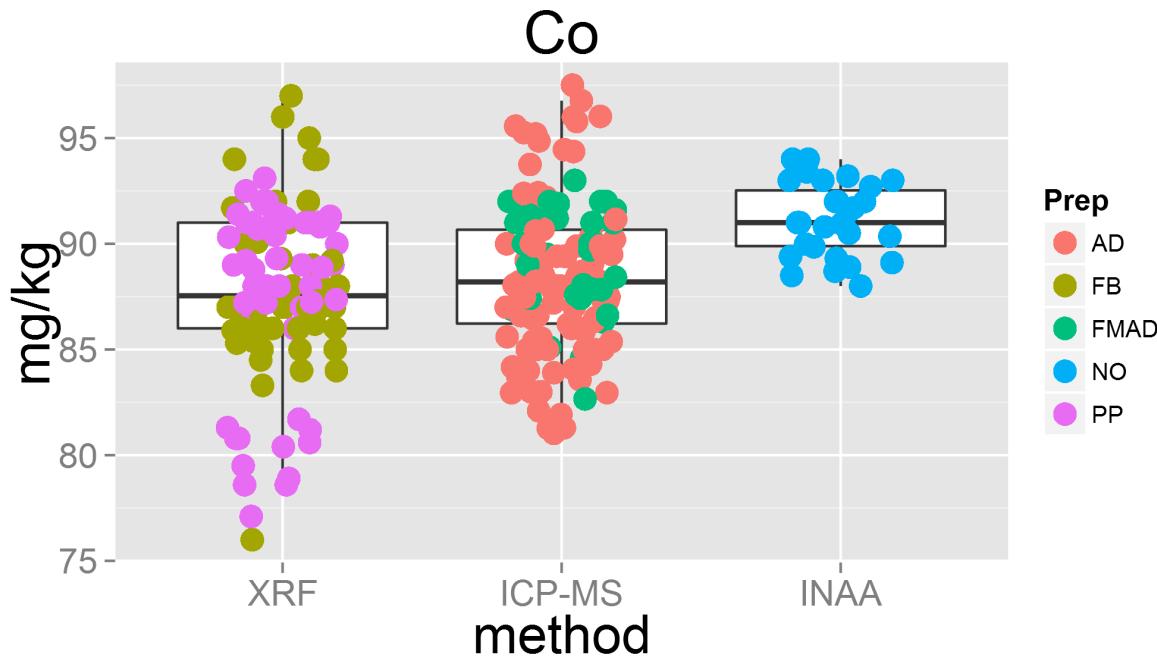
```
## Lab 15 was removed
```



```
## Warning: Removed 2 rows containing missing values (geom_point).
```

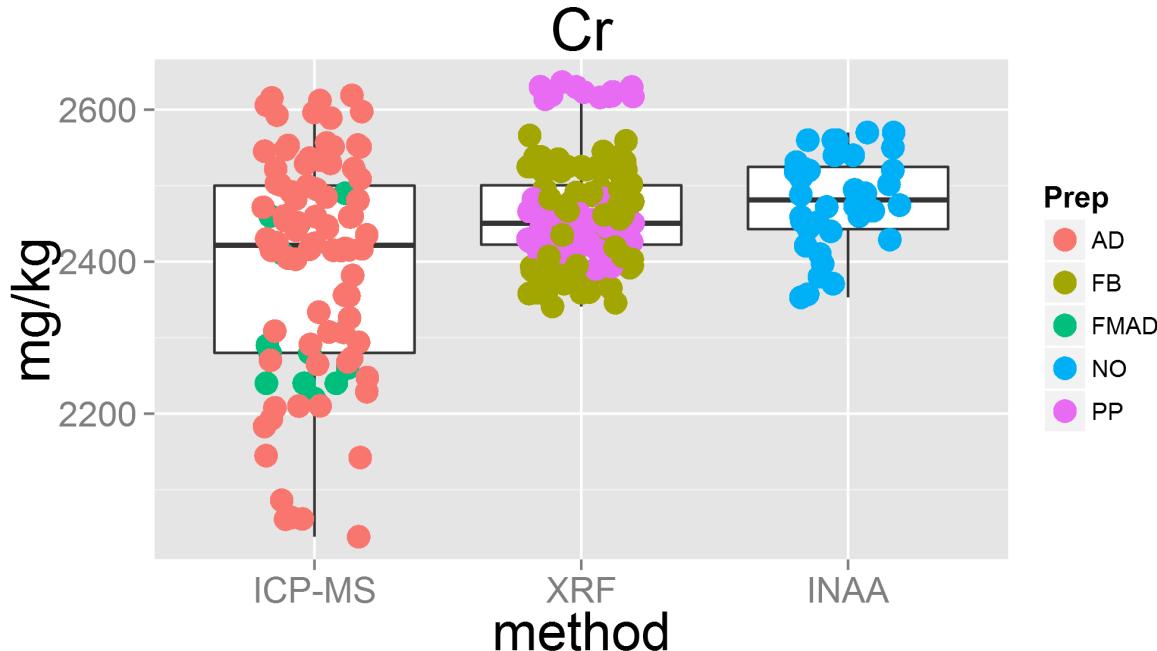
```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Lab 26 was removed
```



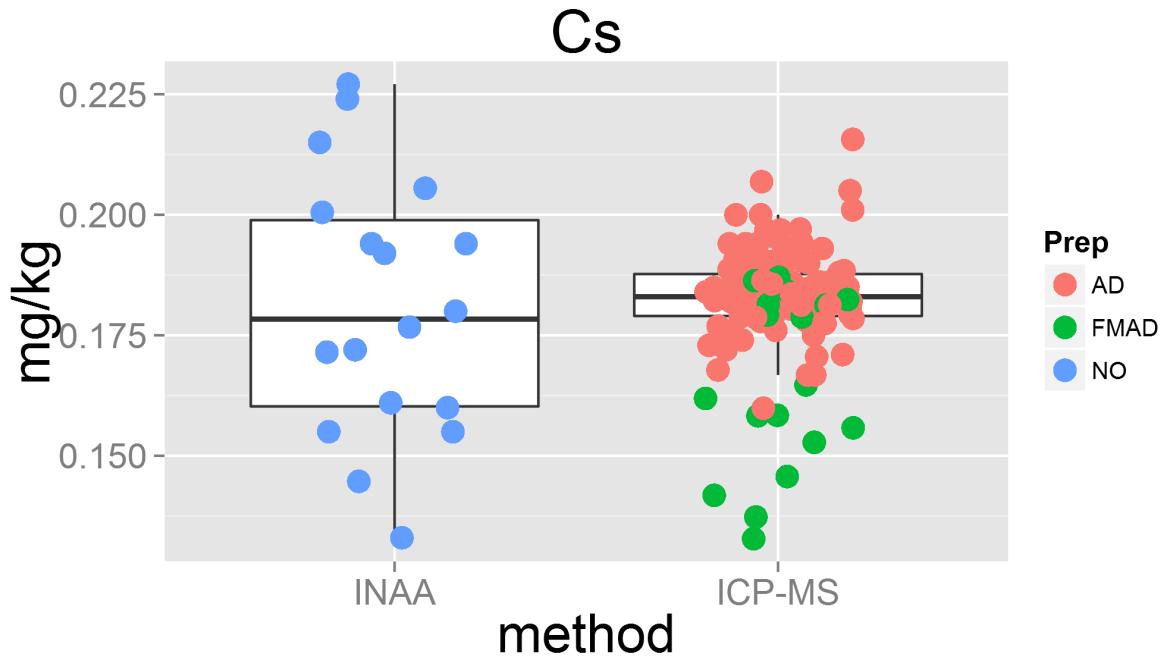
```
## Warning: Removed 9 rows containing missing values (geom_point).
```

```
## Lab 12 was removed  
## Lab 16 was removed
```



```
## Warning: Removed 15 rows containing missing values (geom_point).
```

```
## Lab 33 was removed
```



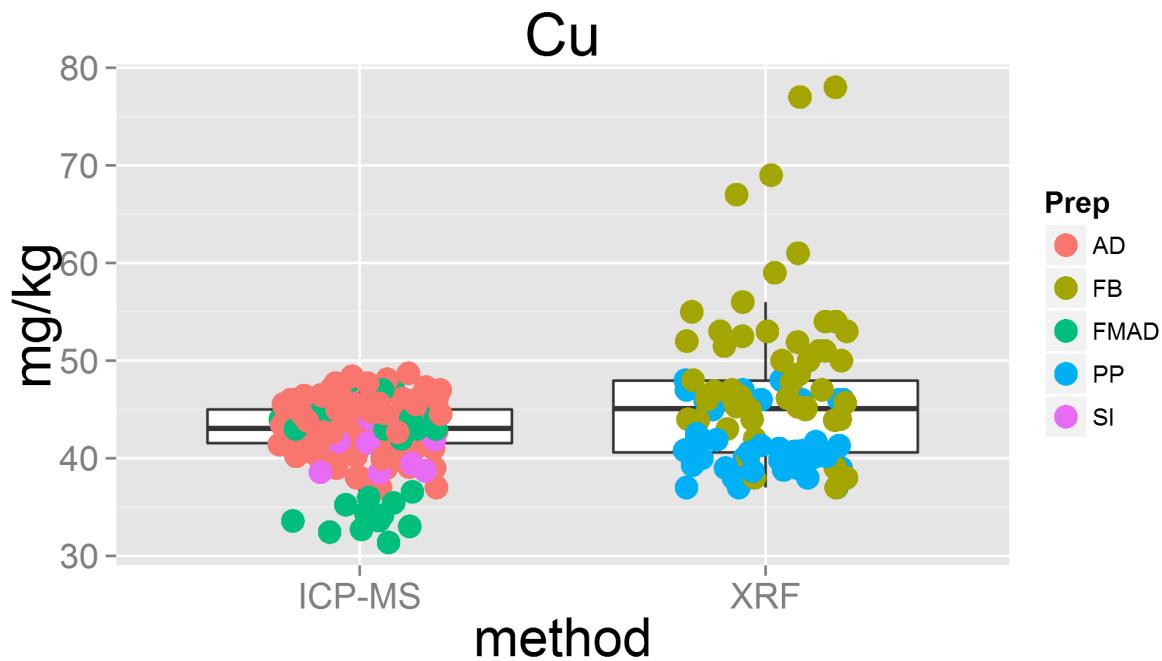
```
## Warning: Removed 11 rows containing missing values (geom_point).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Lab 3 was removed
```

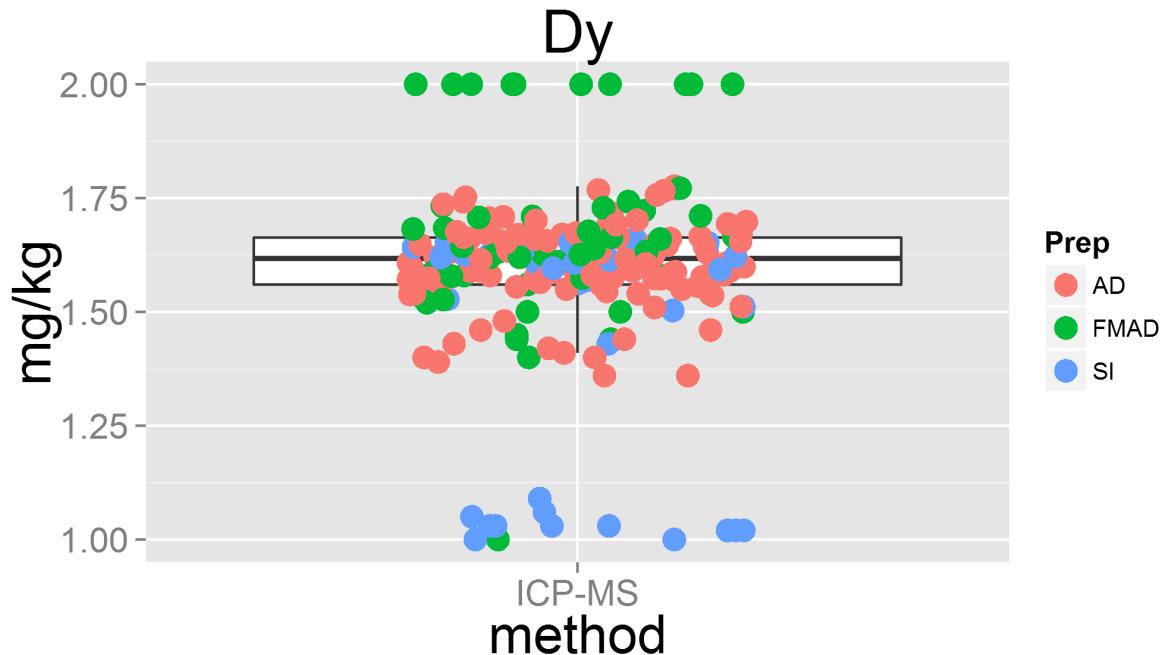
```
## Lab 33 was removed
```

```
## Lab 6 was removed
```



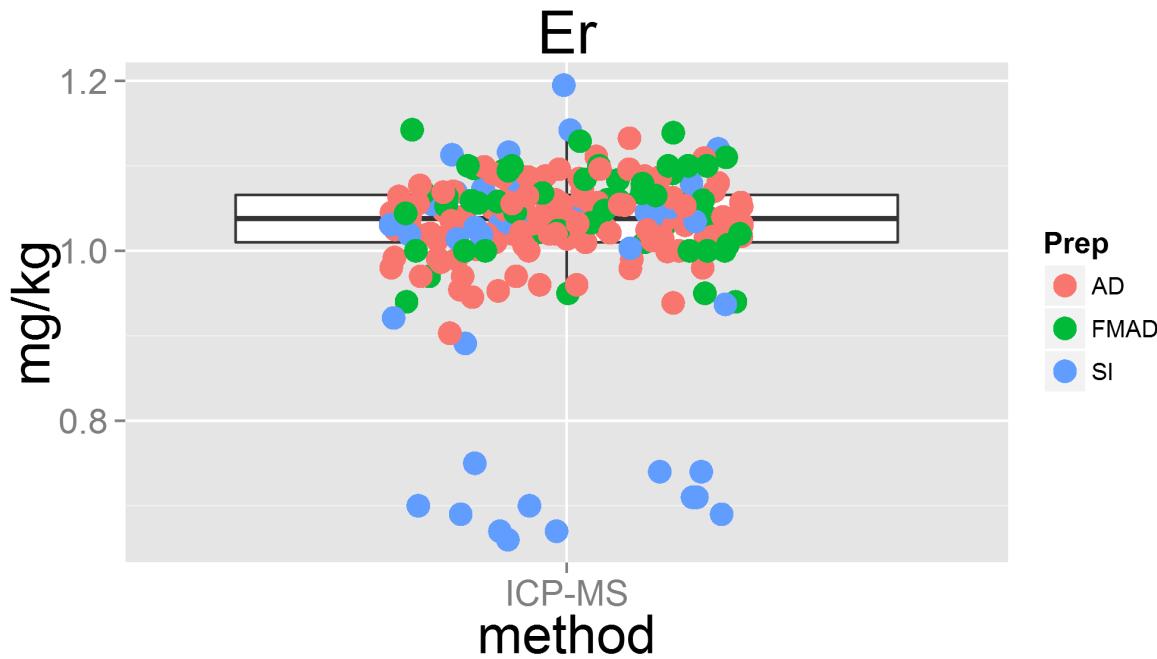
```
## Warning: Removed 30 rows containing missing values (geom_point).
```

```
## Lab 31 was removed
## Lab 27 was removed
```



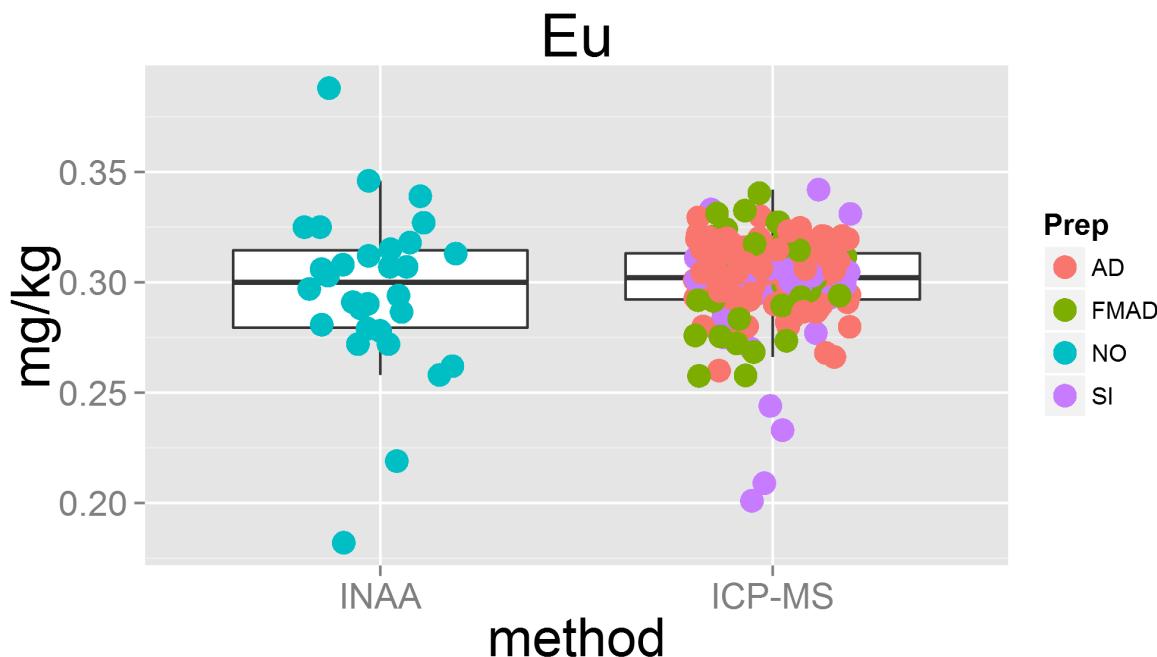
```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Lab 31 was removed
```



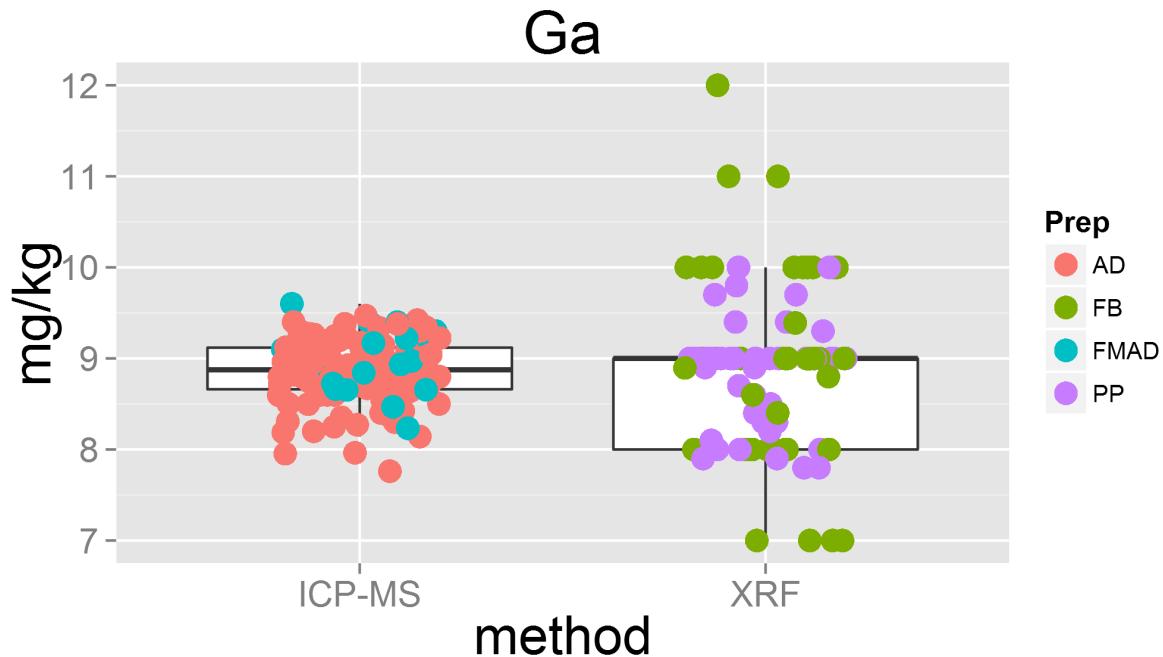
```
## Warning: Removed 3 rows containing missing values (geom_point).  
## Warning: Removed 7 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



```
## Warning: Removed 3 rows containing missing values (geom_point).  
## Warning: Removed 3 rows containing missing values (geom_point).
```

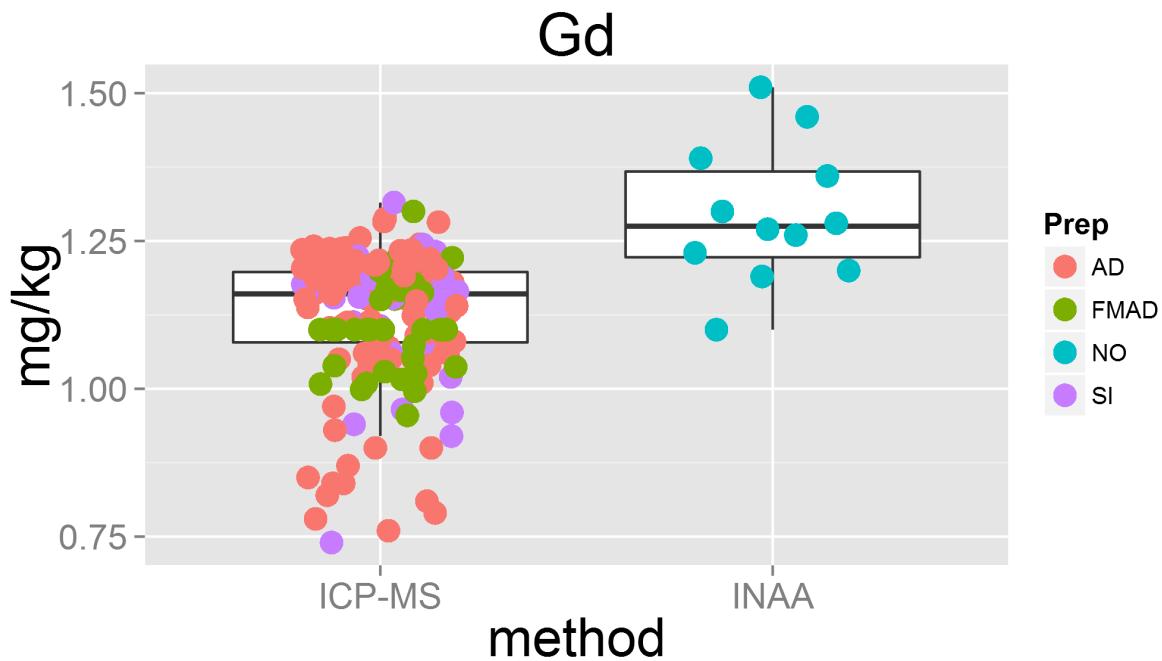
```
## Lab 6 was removed
```



```
## Warning: Removed 12 rows containing missing values (geom_point).
```

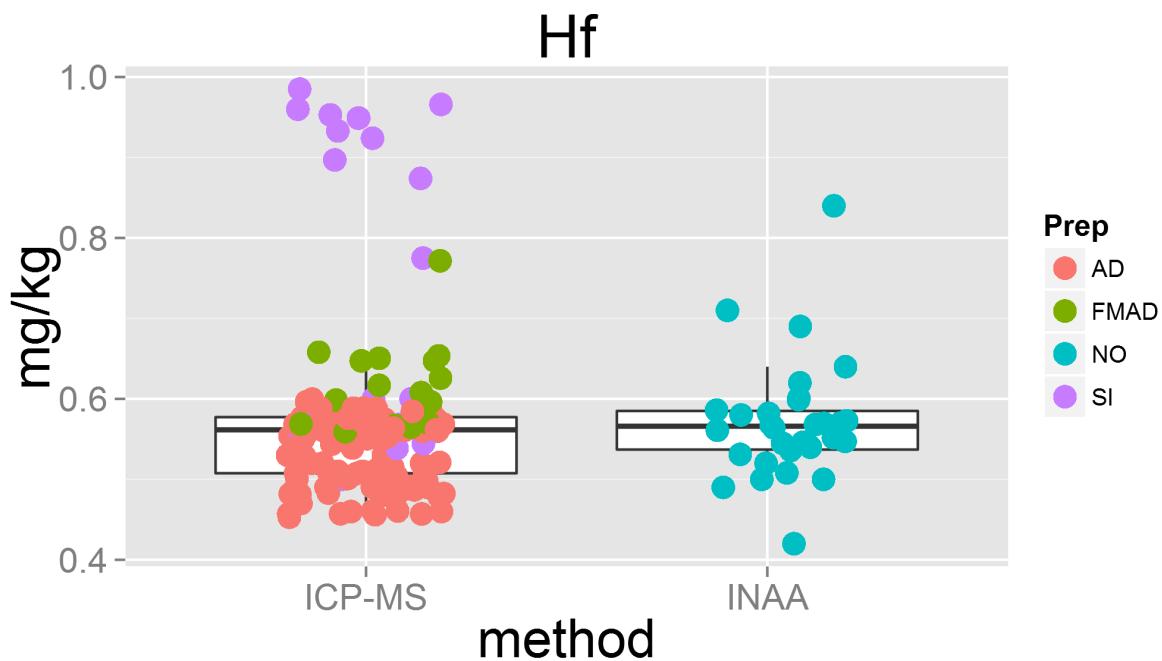
```
## Lab 8 was removed
```

```
## Lab 31 was removed
```



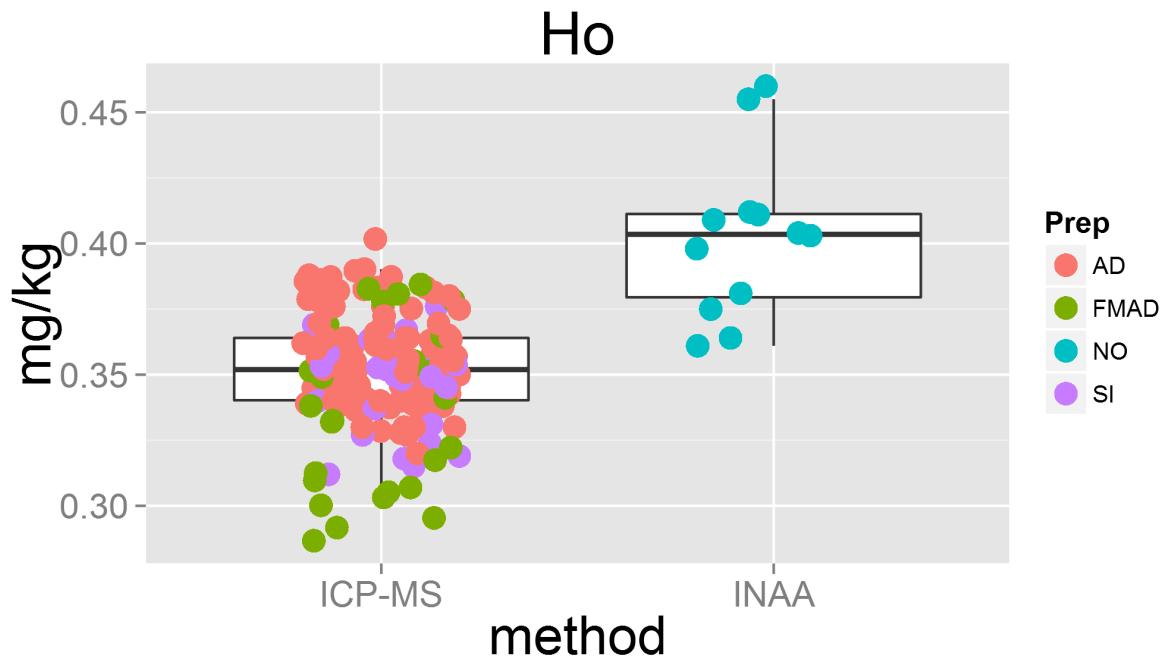
```
## Warning: Removed 12 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Lab 14 was removed
```



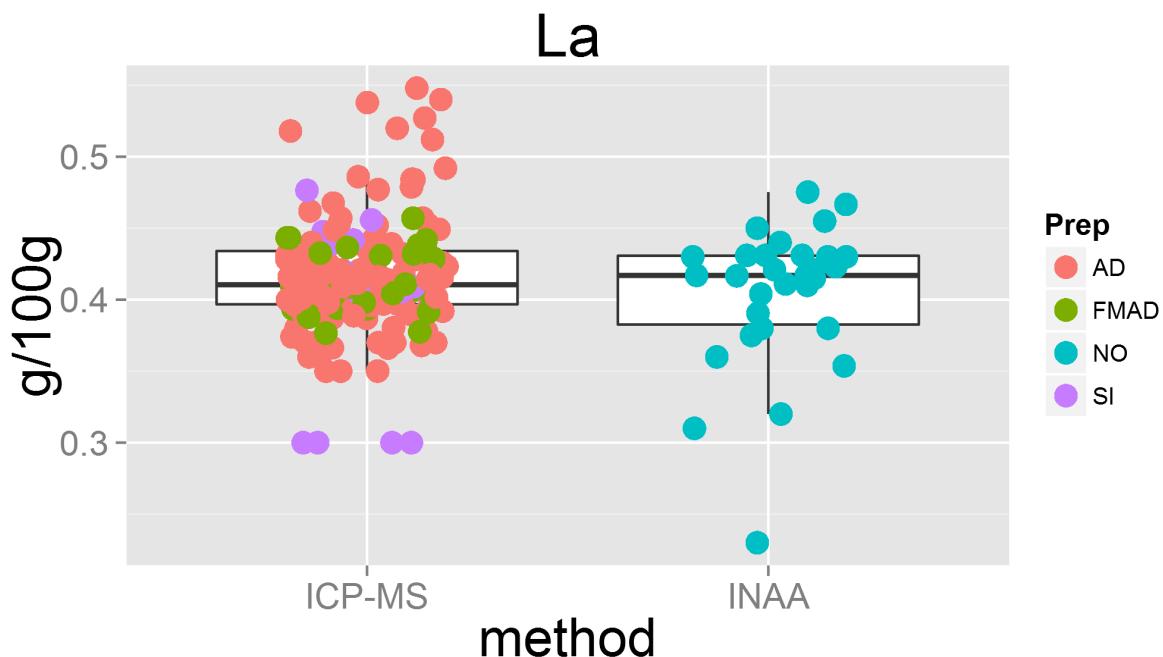
```
## Warning: Removed 6 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Lab 31 was removed  
## Lab 33 was removed
```



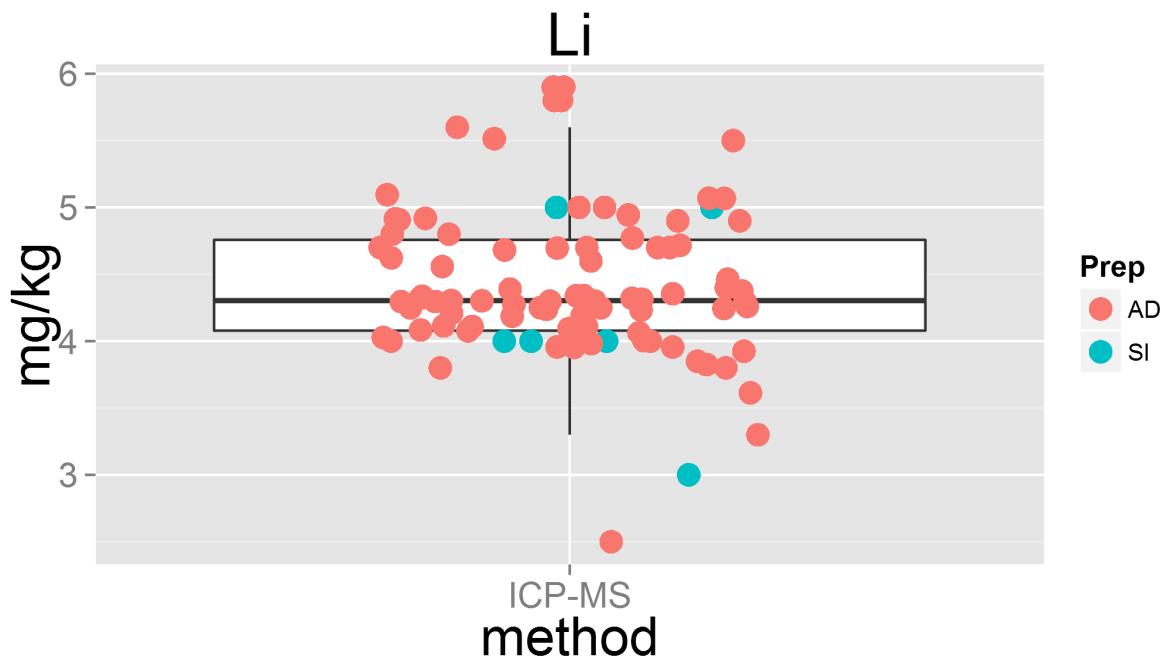
```
## Warning: Removed 12 rows containing missing values (geom_point).  
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



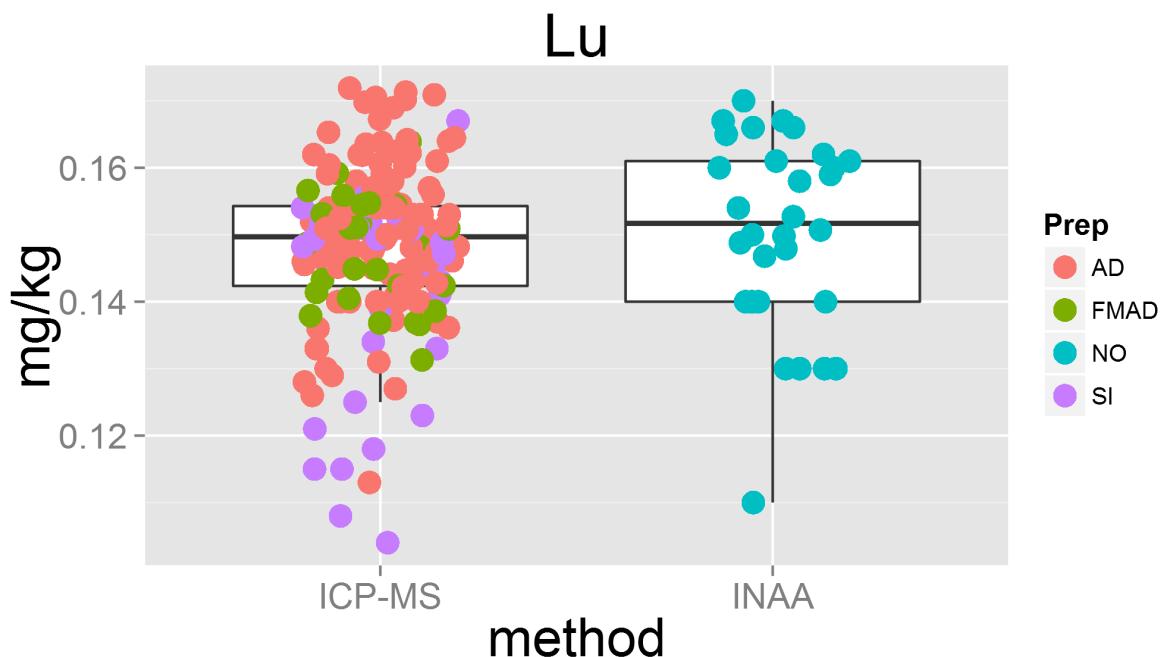
```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



```
## Warning: Removed 8 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```

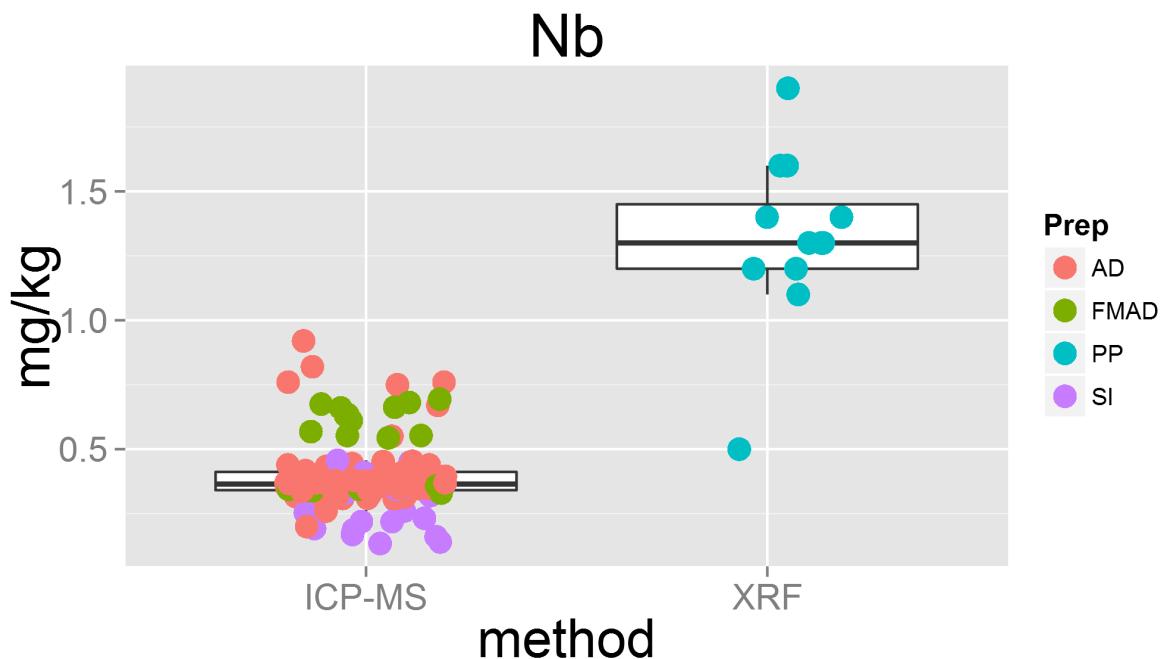


```

## Warning: Removed 29 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_point).

## Lab 26 was removed
## Lab 33 was removed

```



```

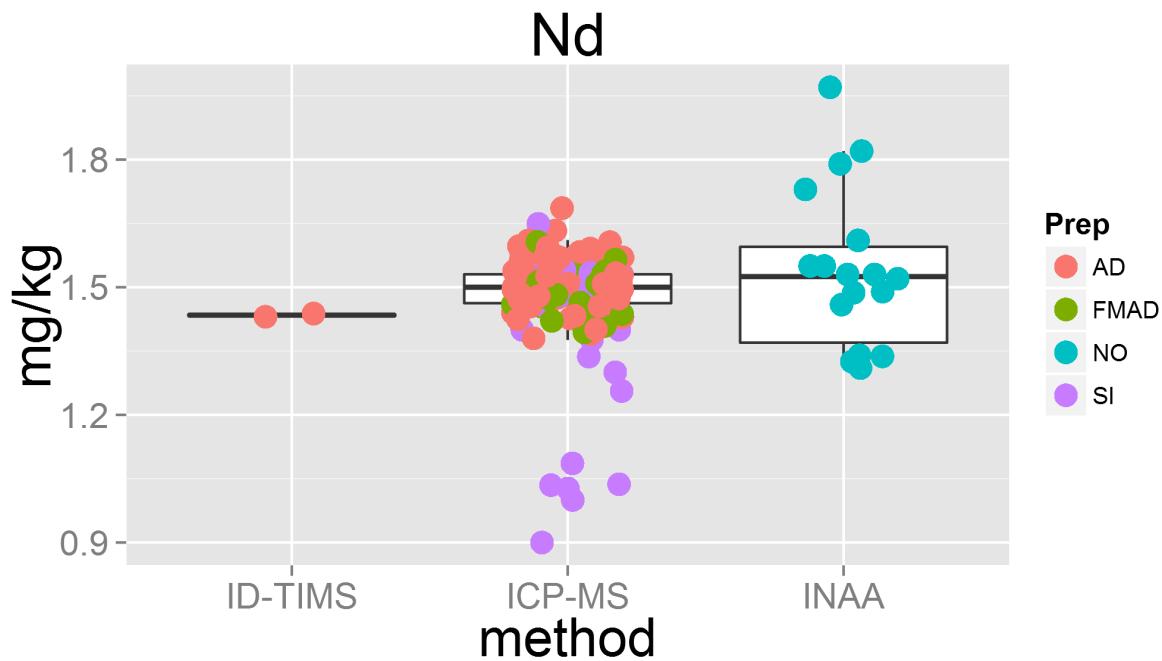
## Warning: Removed 12 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).

```

```

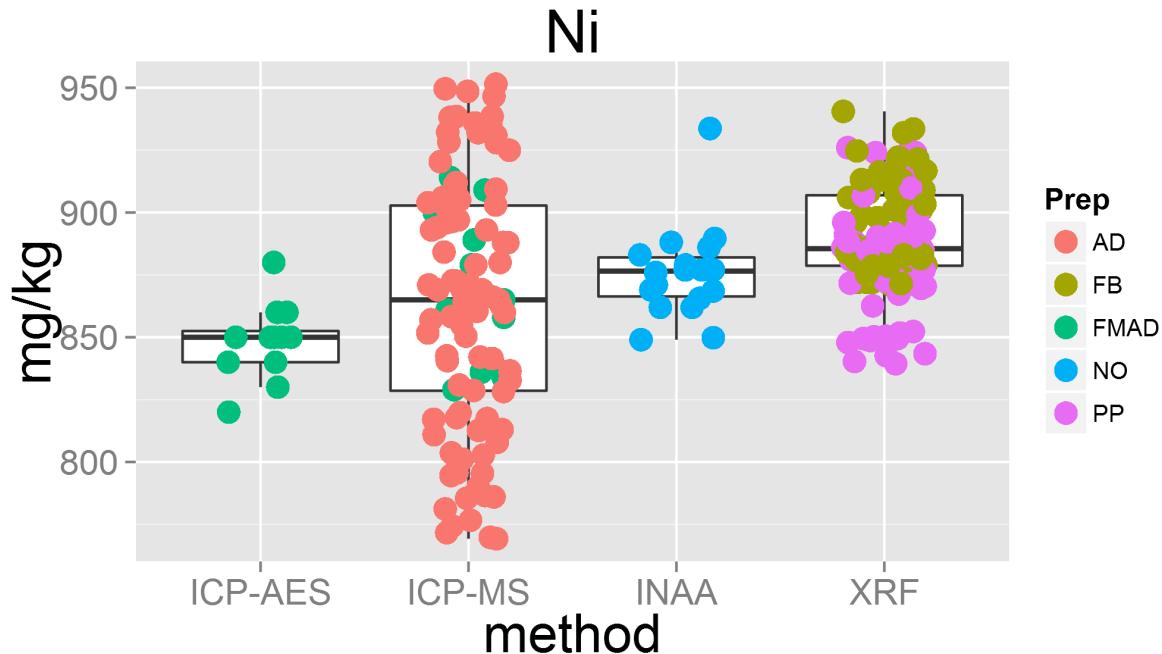
## Lab 4 was removed
## Lab 14 was removed
## Lab 31 was removed

```



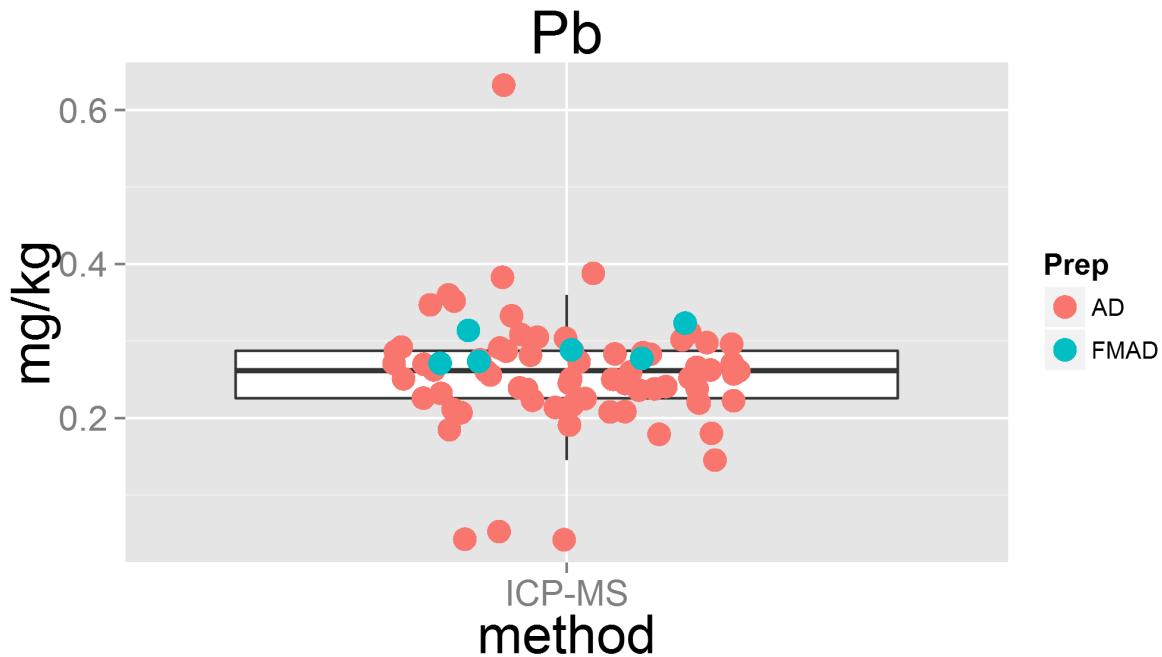
```
## Warning: Removed 2 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Lab 12 was removed
## Lab 18 was removed
## Lab 22 was removed
```



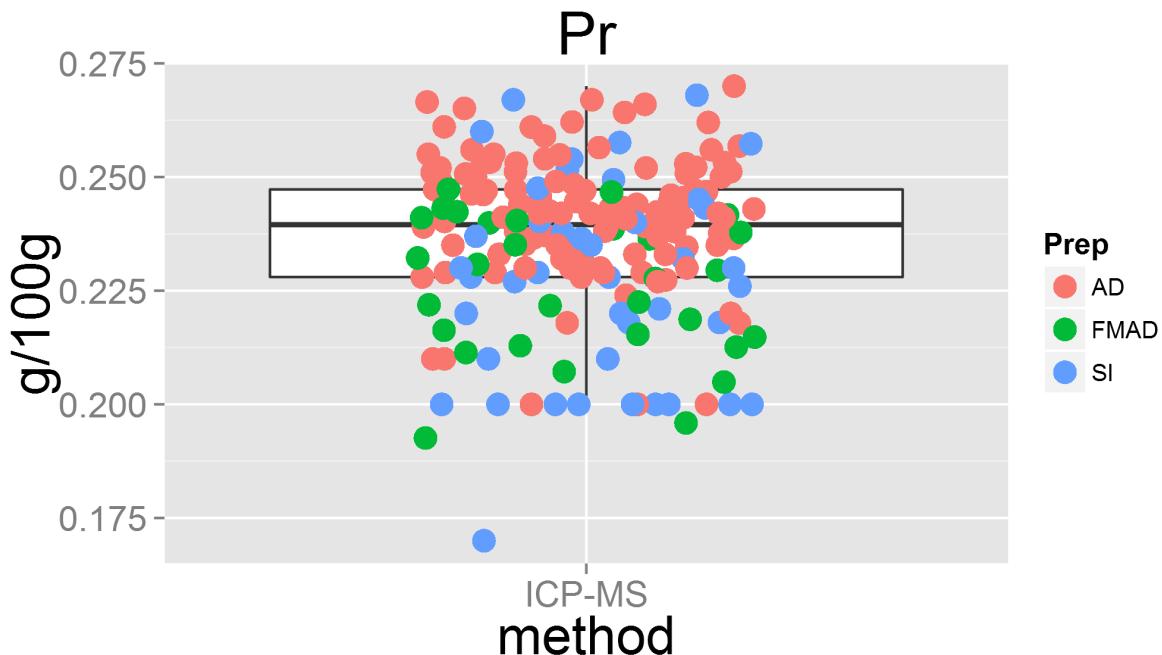
```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



```
## Warning: Removed 3 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```

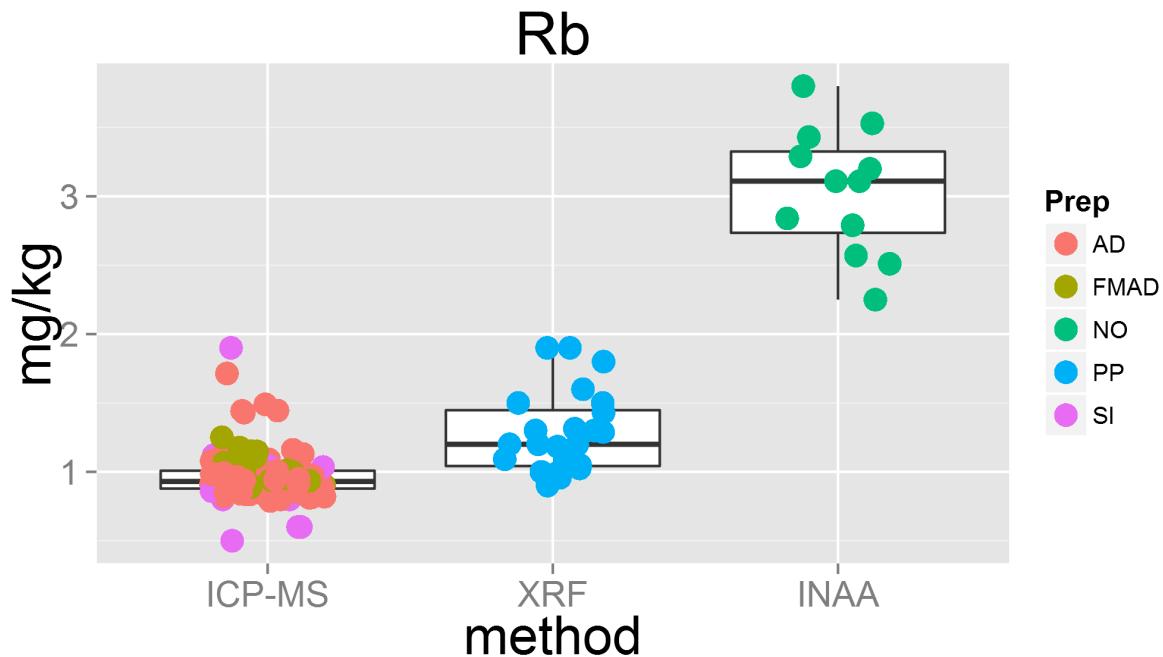


```

## Warning: Removed 10 rows containing missing values (geom_point).

## Lab 4 was removed
## Lab 24 was removed
## Lab 26 was removed
## Lab 31 was removed

```

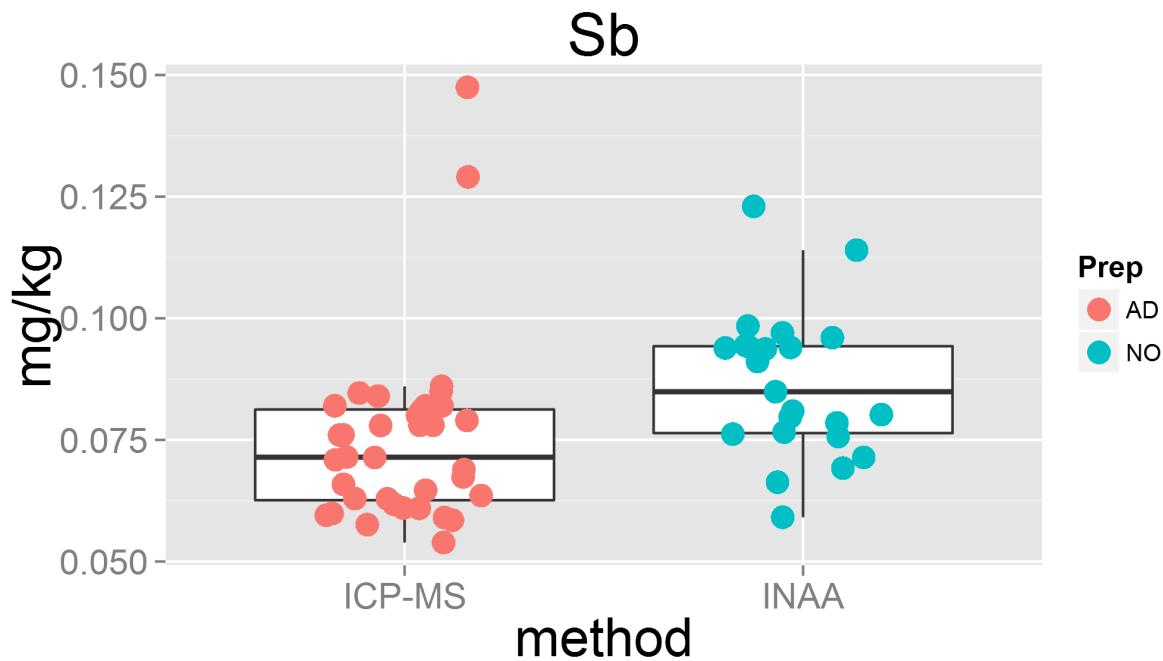


```

## Warning: Removed 2 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).

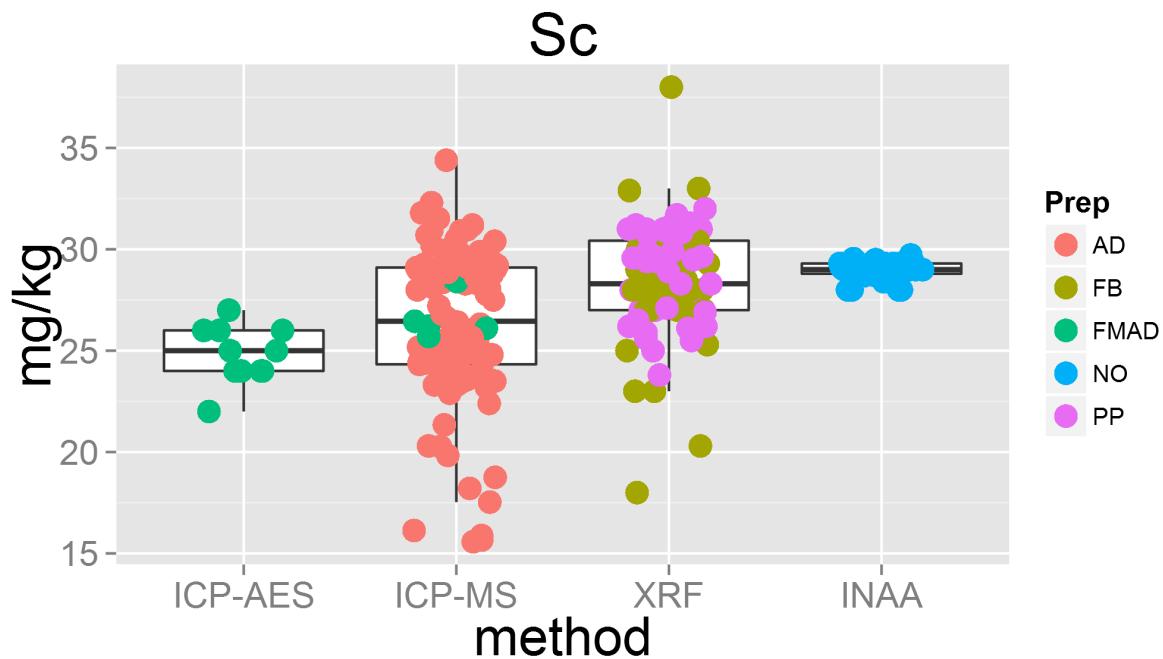
## Lab 0 was removed

```



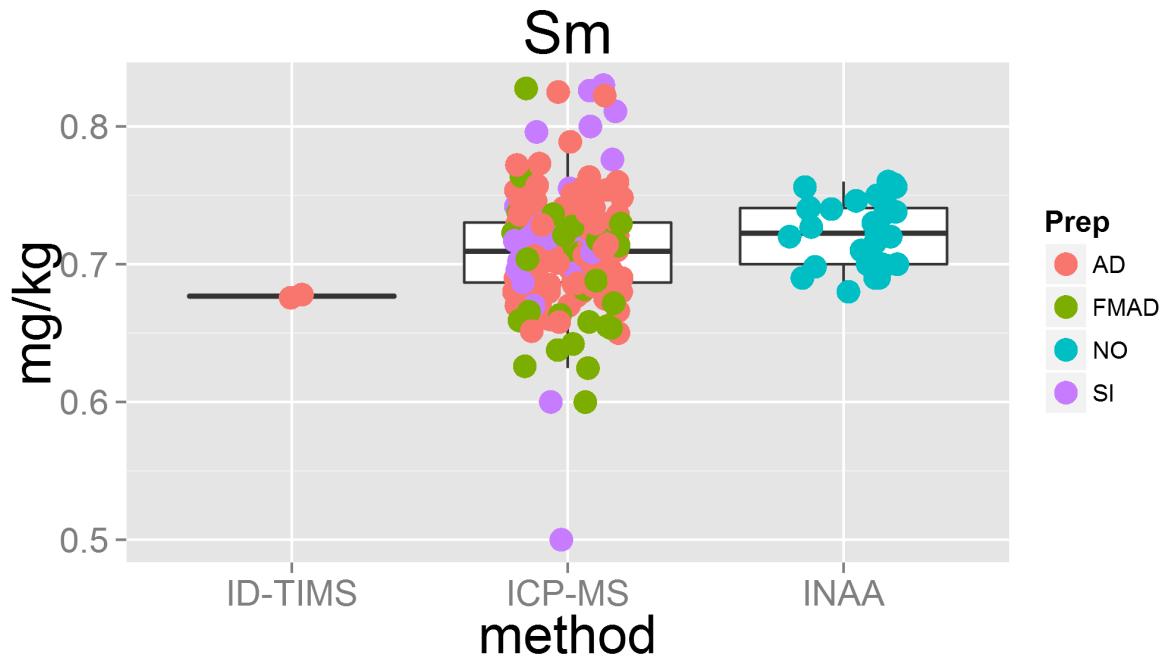
```
## Warning: Removed 4 rows containing missing values (geom_point).
## Warning: Removed 3 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Lab 18 was removed
```

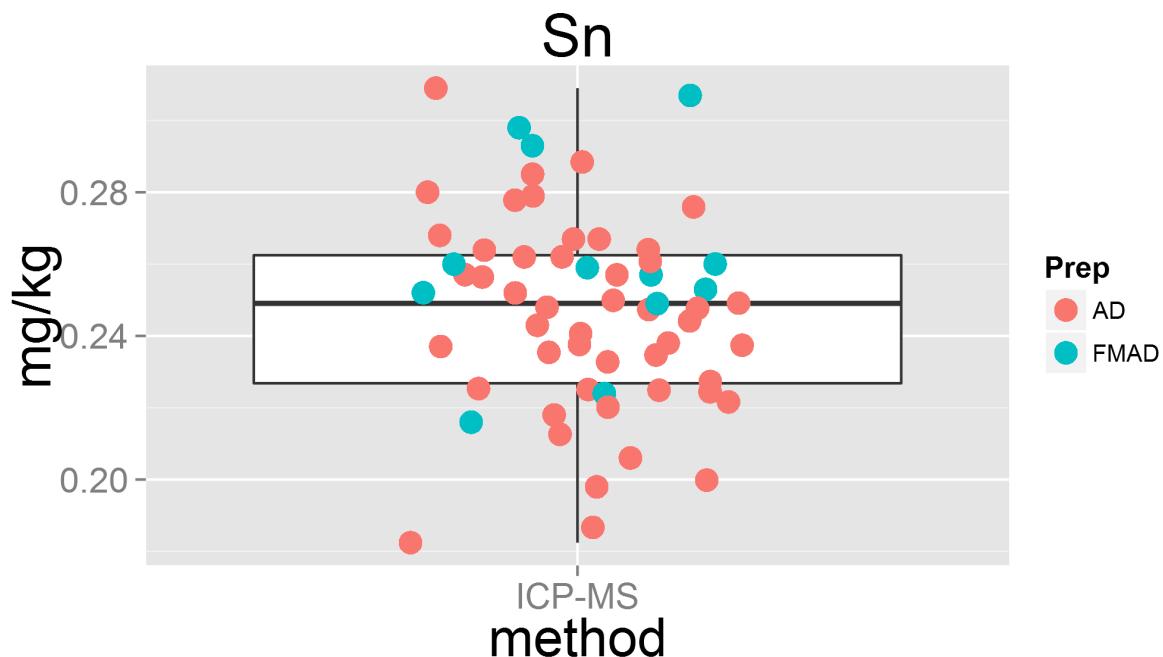


```
## Warning: Removed 11 rows containing missing values (geom_point).
```

```
## Lab 4 was removed  
## Lab 33 was removed
```



```
## Lab 0 was removed
```



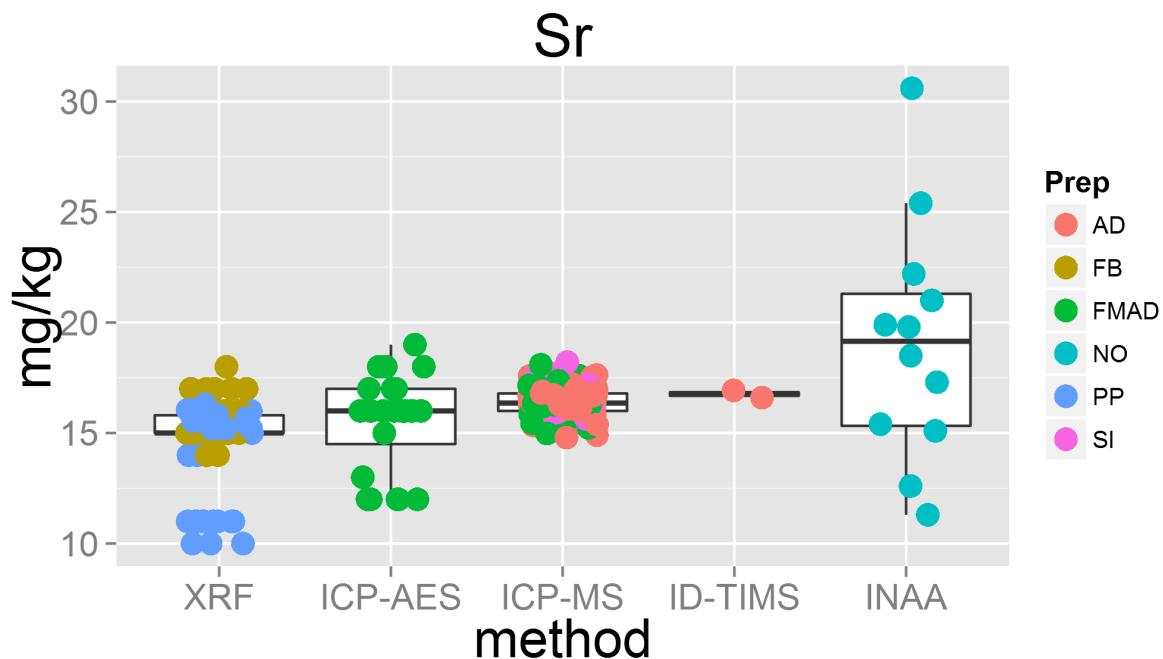
```
## Warning: Removed 14 rows containing missing values (geom_point).
```

```

## Warning: Removed 3 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).

## Lab 10 was removed
## Lab 31 was removed

```

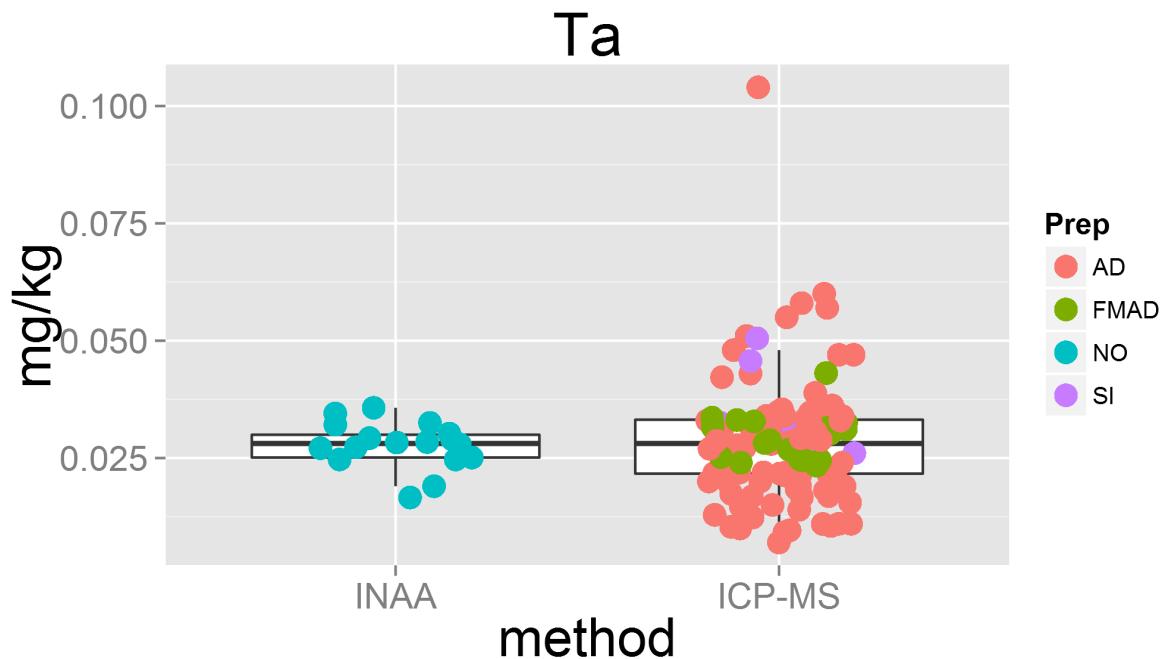


```

## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).

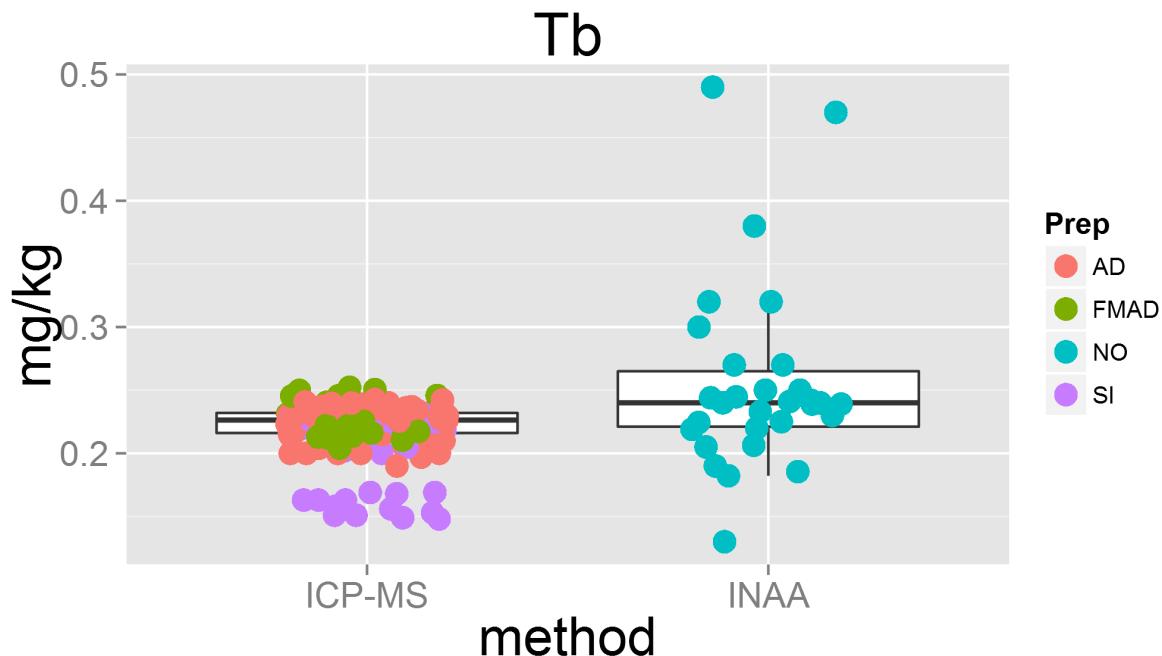
## Lab 8 was removed

```



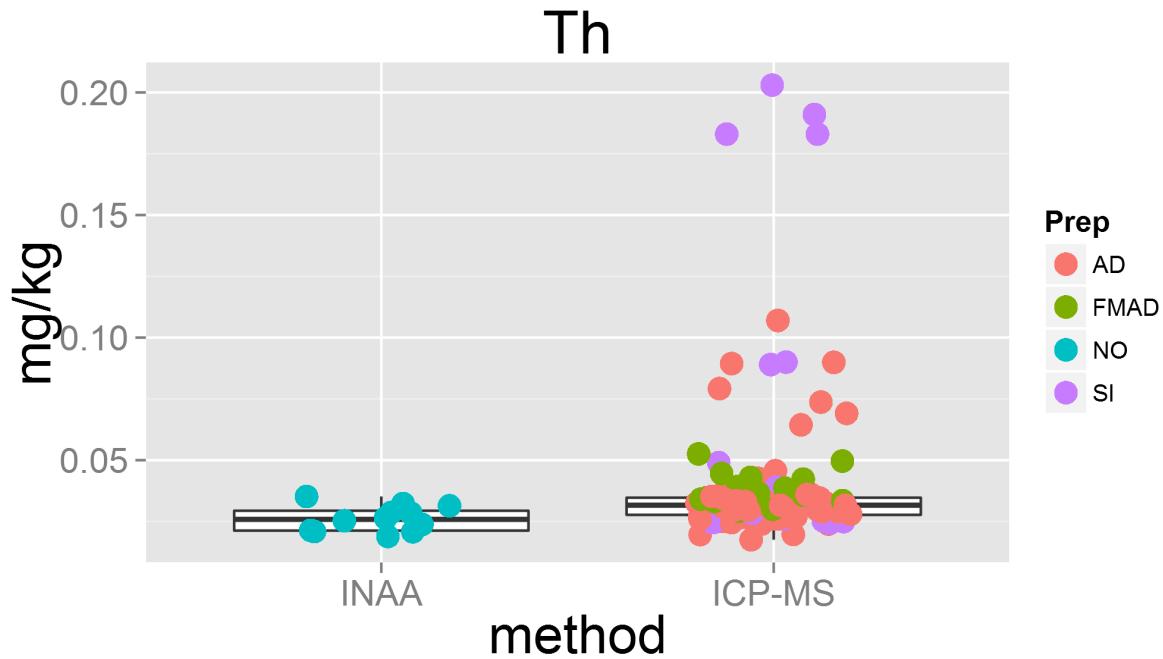
```
## Warning: Removed 14 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Lab 32 was removed
## Lab 15 was removed
```



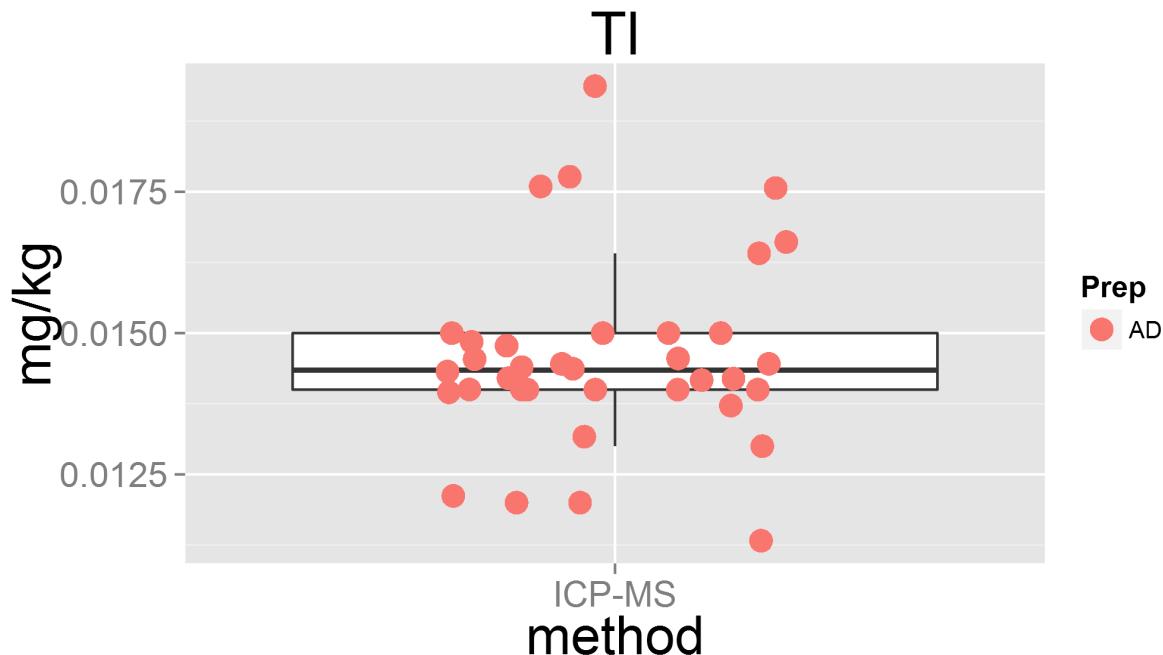
```
## Warning: Removed 17 rows containing missing values (geom_point).
```

```
## Lab 9 was removed  
## Lab 14 was removed
```



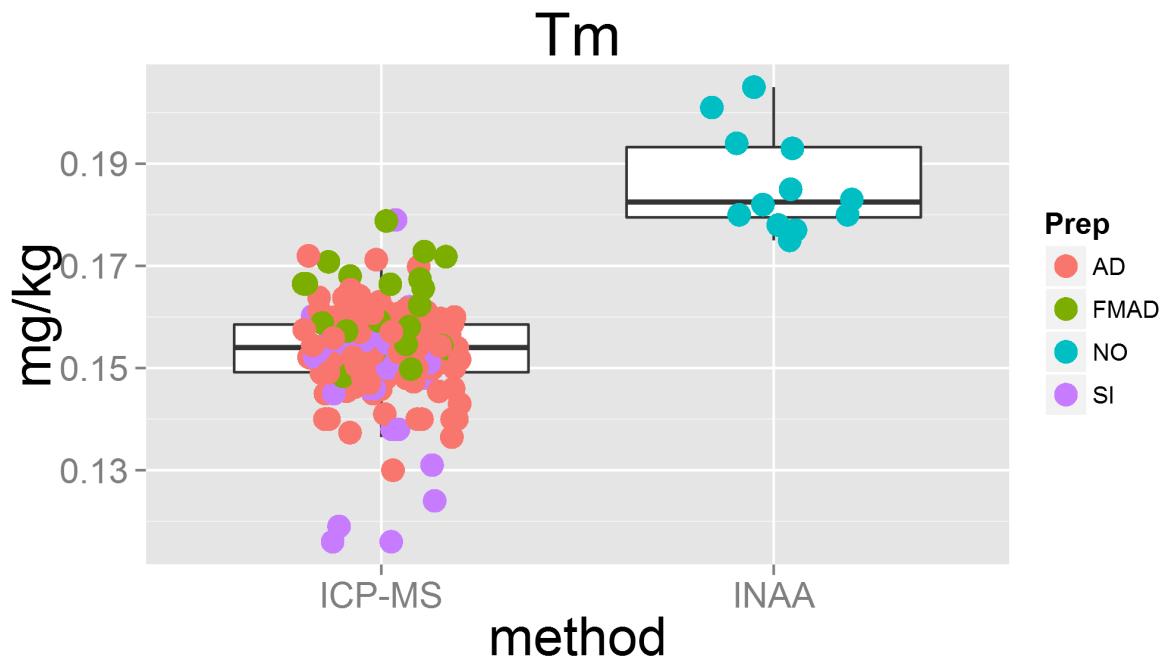
```
## Warning: Removed 9 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```



```
## Warning: Removed 9 rows containing missing values (geom_point).
```

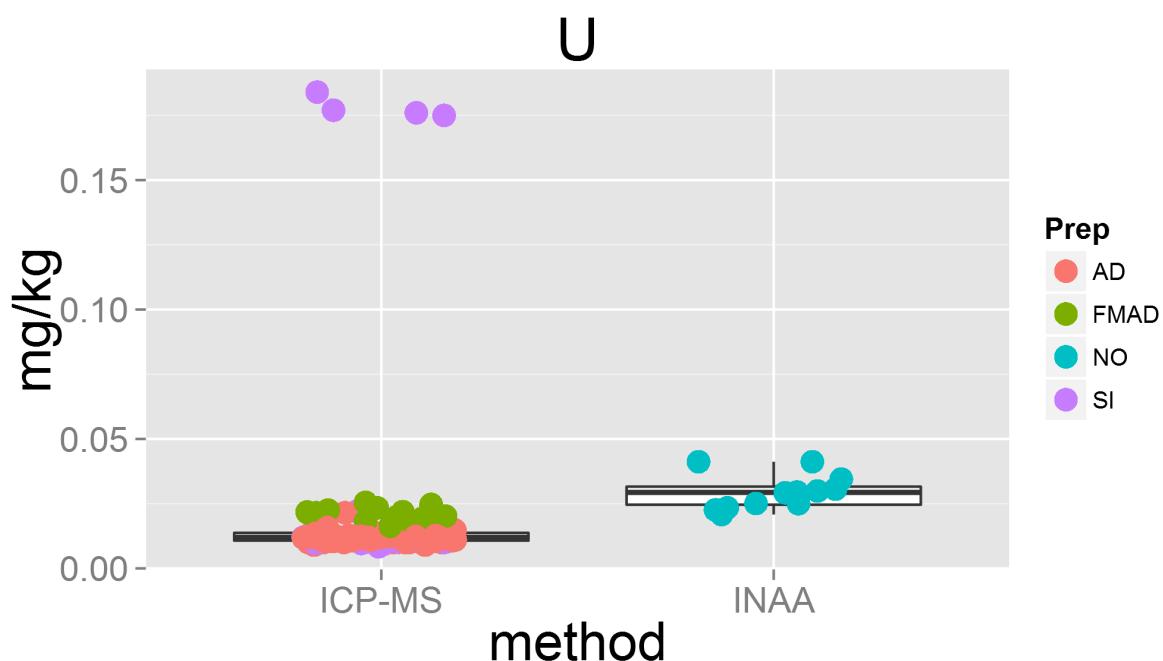
```
## Lab 31 was removed
```



```
## Warning: Removed 21 rows containing missing values (geom_point).
```

```
## Lab 14 was removed
```

```
## Lab 31 was removed
```

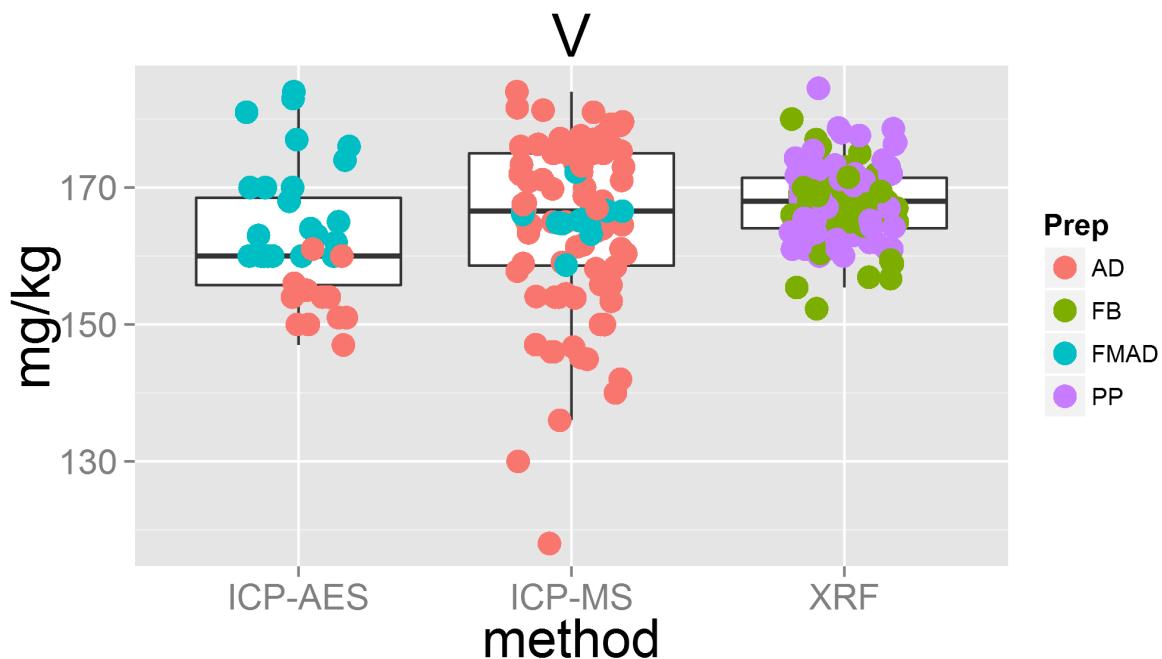


```

## Warning: Removed 2 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_point).

## Lab 31 was removed
## Lab 32 was removed

```

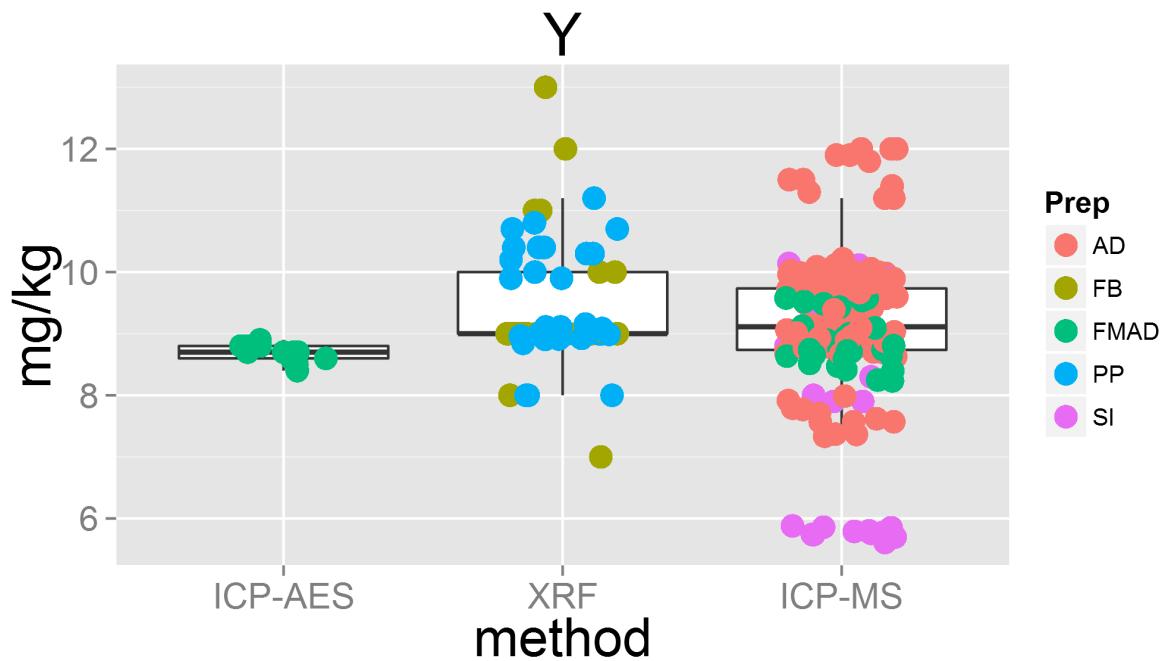


```

## Warning: Removed 3 rows containing missing values (geom_point).
## Warning: Removed 22 rows containing missing values (geom_point).

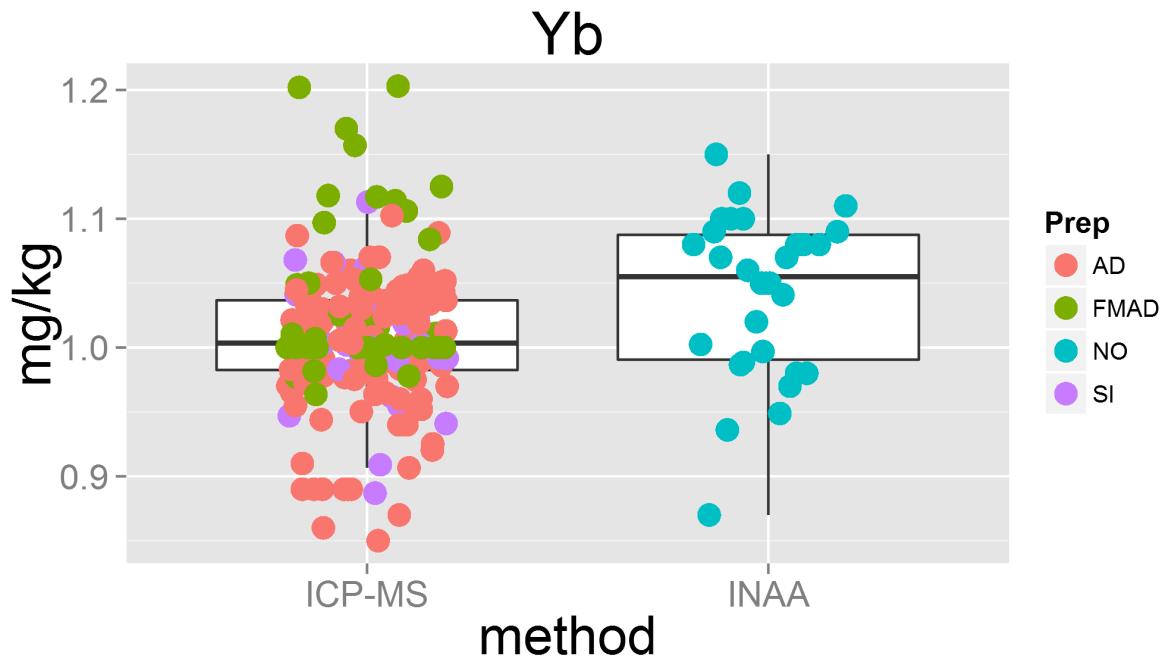
## Lab 6 was removed
## Lab 26 was removed
## Lab 31 was removed
## Lab 32 was removed

```



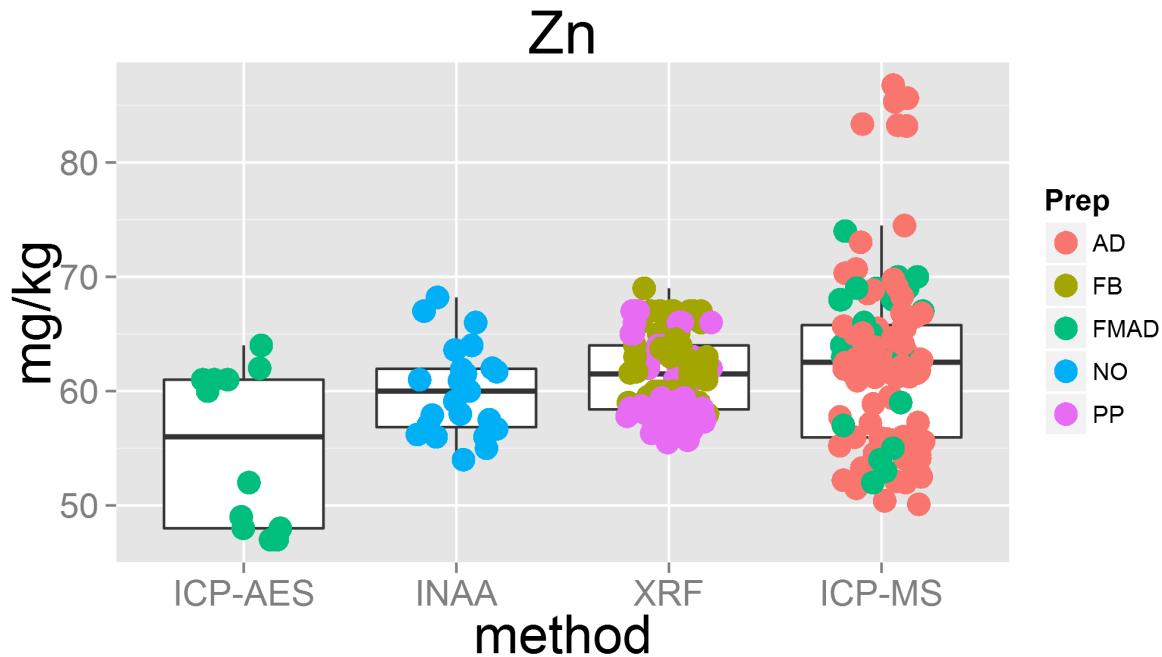
```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Lab 0 was removed
```

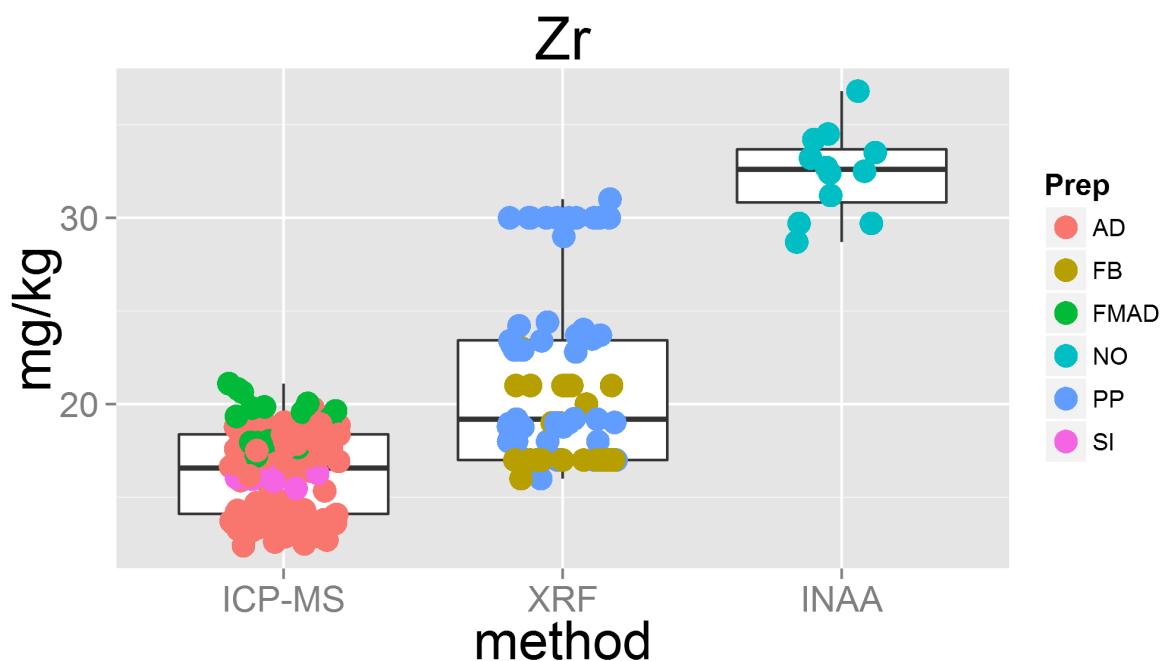


```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Lab 12 was removed  
## Lab 12 was removed
```



```
## Lab 10 was removed  
## Lab 26 was removed  
## Lab 31 was removed
```



```

medianGOM.packet.after <- ddply(GOM, c("Lab", "Packet"), numcolwise(medianGOM)) # median Lab and Packet
## median over median of packets within lab
GOM.median.after <- ddply(medianGOM.packet.after, c("Lab"), numcolwise(medianGOM)) # creating a new table
GOM.median.after <- merge(GOM.median.after, OKUM.methods, by="Lab")
GOM.median.after.df <- data.frame(apply(GOM.median.after[2:56], 2, median, na.rm=TRUE)) # creating a new table
names(GOM.median.after.df) <- c("mass fraction")
meanGOM.packet <- ddply(GOM, c("Lab", "Packet"), numcolwise(meanGOM))
meanGOM.packet.after <- ddply(GOM, c("Lab", "Packet"), numcolwise(meanGOM)) # mean Lab and Packets after
## mean over mean of packets within lab
GOM.mean.after <- ddply(meanGOM.packet.after, c("Lab"), numcolwise(meanGOM)) # creating a new table of means
GOM.mean.after <- merge(GOM.mean.after, OKUM.methods, by="Lab")
GOM.mean.after.df <- data.frame(apply(GOM.mean.after[2:56], 2, mean, na.rm=TRUE)) # creating a new table
names(GOM.mean.after.df) <- c("mass fraction")

```

### Nested random effects in data analysis: two way ANOVA

This model can be used when the results of the interlaboratory study are used to confirm the homogeneity of the material as well as to characterise it. The experimental scheme is illustrated in Fig X for the particular case of the IAG protocol. When the ILC consists of different methos, the result can be expressed by the equation

$$x_{ijk} = \mu + A_i + B_{ij} + \epsilon_{ijk}$$

where

$x_{ijk}$  is the  $k$ th result of sample unit  $j$  reported from method/laboratory  $i$ ,

$A_i$  is the error due to method/laboratory  $i$ ,

$B_{ij}$  is the error due to the  $j$ th sample unit within method/laboratory  $i$ ,  $\epsilon_{ijk}$  is the measurement error.

The parameters to be estimated are the grand mean, the between-laboratory standard deviation  $s_L$ , the between-bottle standard deviation  $s_{bb}$  and the repeatability standard deviation  $s_r$ . The are related as follows

$$s_L = \sqrt{Var(A_i)}$$

$$s_{bb} = \sqrt{Var(B_{ij})}$$

$$s_r = \sqrt{Var(\epsilon_{ijk})}$$

The formulae for computing the above-mentioned estimates read as follows. The grand mean is computed using

$$\bar{x} = \frac{1}{\sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij}} \sum_{i=1}^p \sum_{j=1}^{b_i} \sum_{k=1}^{n_{ij}} x_{ijk}$$

where  $p$  denotes the number of laboratories,  $b_i$  the number of bottles used by method/laboratory  $i$ , and  $n_{ij}$  is the number of replicates measured on bottle  $ij$ . The variances are computed as follows

$$Var(\epsilon_{ijk}) = MS_{within} = s_r^2$$

$$Var(B_{ij}) = \frac{MS_{B \subset A} - MS_{within}}{n_0} = s_b^2 b$$

$$Var(A_i) = \frac{MS_{among} - n'_0 Var(B_{ij}) - Var(\epsilon_{ijk})}{(nb)_0} = s_L^2$$

where

$$MS_{among} = \frac{\sum_{i=1}^p n_i (\bar{x}_A - \bar{x})^2}{p-1}$$

$$MS_{B \subset A} = \frac{\sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij} (\bar{x}_B - \bar{x}_A)^2}{\sum_{i=1}^p b_i - p}$$

$$MS_{within} = \frac{\sum_{i=1}^p \sum_{j=1}^{b_i} \sum_{k=1}^{n_{ij}} (x_{ijk} - \bar{x}_B)^2}{\sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij} - \sum_{i=1}^p b_i}$$

and

$$n'_0 = \frac{\sum_{i=1}^p \left( \frac{\sum_{j=1}^{b_i} n_{ij}^2}{\sum_{j=1}^{b_i} n_{ij}} \right) - \sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij}^2}{p-1}$$

$$n_0 = \frac{\sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij} - \sum_{i=1}^p \left( \frac{\sum_{j=1}^{b_i} n_{ij}^2}{\sum_{j=1}^{b_i} n_{ij}} \right)}{\sum_{i=1}^p b_i - p}$$

$$(nb)_0 = \frac{\sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij} - \frac{\sum_{i=1}^p (\sum_{j=1}^{b_i} n_{ij})^2}{\sum_{i=1}^p \sum_{j=1}^{b_i} n_{ij}}}{p-1}$$

### solutions of the above equations in R

ANOVA is calculated based on a linear model using the using “linear mixed effects models” of package nlme (lme {nlme})

```
GOM.lme <- lme(measurand ~ 1, random = ~ 1|Lab/Packet, data=DF.lme) # linear model with random effects
```

the variance components are extracted with package ape (varcomp {ape})

```
sL2 = sL2 <- varcomp(GOM.lme, FALSE, FALSE)[[1]] # between-laboratory variance
```

```
sbb2 = sbb2 <- varcomp(GOM.lme, FALSE, FALSE)[[2]] # between bottle variance
```

```
sr2 = sr2 <- varcomp(GOM.lme, FALSE, FALSE)[[3]] # repeatability standard deviation
```

The characterisation uncertainty  $u_{char}$  is calculated

```
u1 <- sqrt(sL2/p+sbb2/p+r+sr2/p/r/4)
```

which is equivalent to

$$u_1 = \sqrt{\frac{s_L^2}{p} + \frac{s_r^2}{np} + \frac{s_{bb}^2}{npr}}$$

This approach is not completely correct as it assumes 4 replicates per bottle

```
u2 <- attr(GOM.lme$fixDF, "varFixFact") # this approach takes unbalanced data into account and is used for all further calculations.
```

The between day variance is neglected here but the uncertainty component due to inhomogeneity is taken into account through the  $s_{bb}^2$  component.

```
for (m in col.names) {
  measurand.name <- m
  switch(
    refmat,
    GAS = rm1 <- 2,
    MUH = rm1 <- 1,
    OKUM = rm1 <- 0
  )
  if(rm1 > 0)
    {measurand <- measurand.name %p% '.' %p% rm1
  } else
  {
    measurand <- measurand.name
  }
  MorT <- grep(measurand.name, colnames(GOM), fixed=TRUE) # finding the position of the measurand.name
  ifelse(MorT[1]< 21, MorT <- 'M', MorT<- 'T') # testing if measurand is a major or trace element/compo
```

```

ifelse(MorT == "T", unit <- 'mg/kg', unit <- 'g/100g') # testing which unit is needed
## calculating method parameters
# '%p%' <- function(x, y) {as.character(paste(x, y, sep = ""))}
mean <- mean(tapply(GOM[[measurand]], GOM$Lab, mean, na.rm=TRUE), na.rm=TRUE)
mean.before <- mean(GOM.mean[[measurand]], na.rm=TRUE)
median.before <- median(GOM.median[[measurand]], na.rm=TRUE)
median.after <- median(GOM.median.after[[measurand]], na.rm=TRUE) # median of the individual measurand
prep <- 'Prep.'
method <- 'Method.'
anal.prep <- prep %p% measurand.name
anal.method <- method %p% measurand.name
anal <- GOM.median.after[[measurand]]
anal.prep <- GOM.median.after[[anal.prep]]
anal.method <- GOM.median.after[[anal.method]]
analyte <- data.frame(GOM.median.after$Lab, GOM.median.after$names, anal, anal.prep, anal.method )
analyte <- na.omit(analyte)

bymethod.n <- ddply(analyte, c("anal.method"), summarise,
                      N=length(anal),
                      mean = round(mean(anal), 3),
                      median = round(median(anal), 3),
                      sd = round(sd(anal),3),
                      se = round(sd/sqrt(N),3))

meanGOM.packet.after$Lab <- as.factor(meanGOM.packet$Lab) # using only the median of the 3 packages per
meanGOM.packet.after$Packet <- as.factor(meanGOM.packet$Packet)
anal <- meanGOM.packet.after[[measurand]]
DF.lme <- data.frame(meanGOM.packet.after$Lab, meanGOM.packet.after$Packet, meanGOM.packet.after[[measurand]])
DF.lme <- na.omit(DF.lme)
names(DF.lme) <- c("Lab", "Packet", "measurand")
GOM.lme <- lme(measurand ~ 1, random = ~ 1|Lab/Packet, data=DF.lme) # linear model with random effects
sL2.a <- varcomp(GOM.lme, FALSE, FALSE)[[1]] # between-laboratory variance
sbb2.a <- varcomp(GOM.lme, FALSE, FALSE)[[2]] # between bottle standard deviation
sr2.a <- varcomp(GOM.lme, FALSE, FALSE)[[3]] # repeatability standard deviation
n.p <- dim(DF.lme)[1] # number of observations
p <- length(unique(DF.lme$Lab)) # haven't found a better way how to extract the number of labs (number of
r <- length(unique(DF.lme$Packet))
t.value <- qt(0.975,df=p-1)
u1.a <- sqrt(sL2.a/p+sbb2.a/p/r+sr2.a/p/r/4) # calculating the standard uncertainty of characterization
u2.a <- attr(GOM.lme$fixDF,"varFixFact") # gives the same results as u1, amazing!
# plot(DF.lme)
analyte.noPP <- subset(analyte, analyte$anal.prep!="PP") # Removing all PP preparations for comparison
median.after.noPP <- median(analyte.noPP$anal)
outlier <- ifelse(outlier=="0", "X", outlier)
print(measurand)
qqnorm(GOM.median.after[[measurand]])
qqline(GOM.median.after[[measurand]])
reference.line <- median.after
u.Ulim <- median.after + u2.a*t.value
l.Ulim <- median.after - u2.a*t.value
bymethod <- ggplot(bymethod.n, aes(x=anal.method, y=median))+geom_point(size=4)+geom_errorbar(aes(ymin=
plot.lab <- plot_lab(measurand, MorT, horw = FALSE, u = TRUE)
# grid.arrange(bymethod, plot.lab, ncol=2)

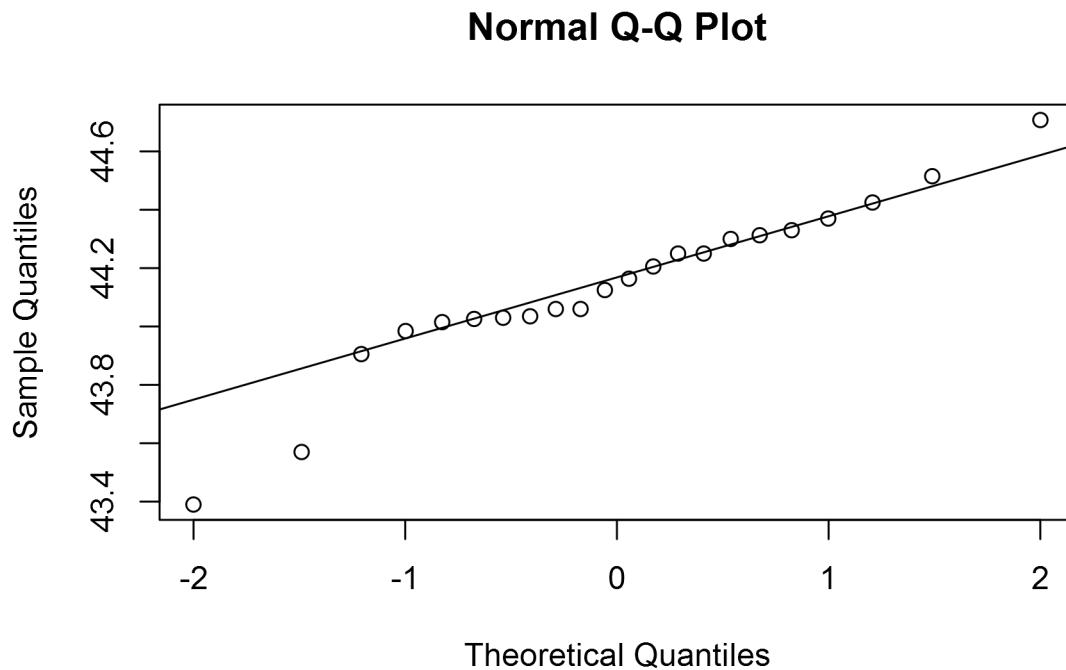
```

```

print(plot.lab)
# print(bymethod.n)
outlier.type.name <- measurand.name %p% ".outlier.type" # defining if outlier is selected ("Y" or NA)
outlier.dist.type <- measurand.name %p% ".dist.type" # defining if outlier is based on Y = Youden plot,
out.measurand <- data.frame(OKUM.outlier[[measurand.name]], OKUM.outlier[[outlier.type.name]], OKUM.out
# out.measurand <- cbind(OKUM.outlier[[measurand.name]], OKUM.outlier[[outlier.type.name]])
names(out.measurand) <- c("outlier.lab", "outlier.type", "outlier.dist")
property.value.dist.type <- out.measurand[1,3]
property.value <- ifelse(property.value.dist.type == "median", median.after, mean)
df <- data.frame(Sys.Date(), refmat, measurand.name, signif(mean.before, 4), signif(mean,4), signif(med
write.table(df, "df0.txt", row.names=FALSE, append=TRUE, col.names=FALSE)
}

## [1] "SiO2"

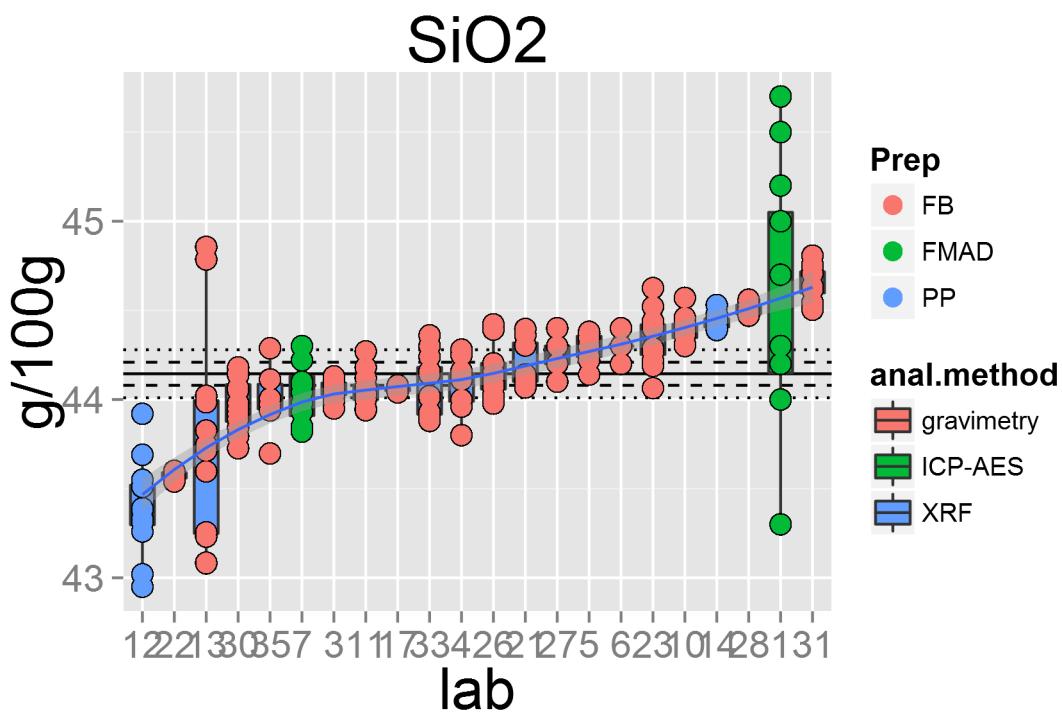
```



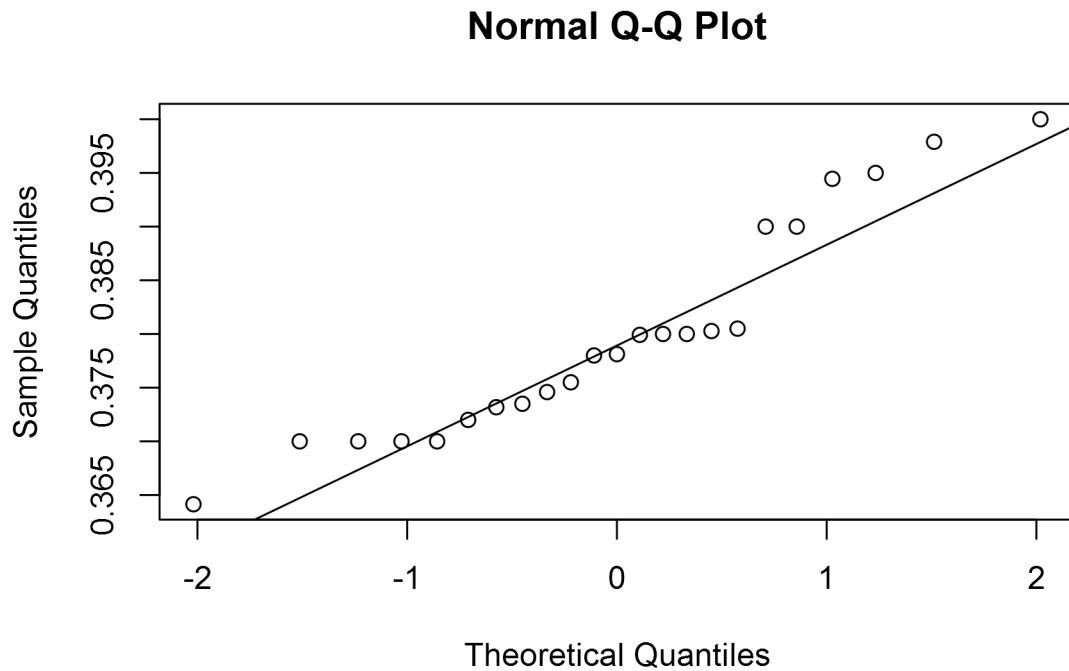
```

## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to ch

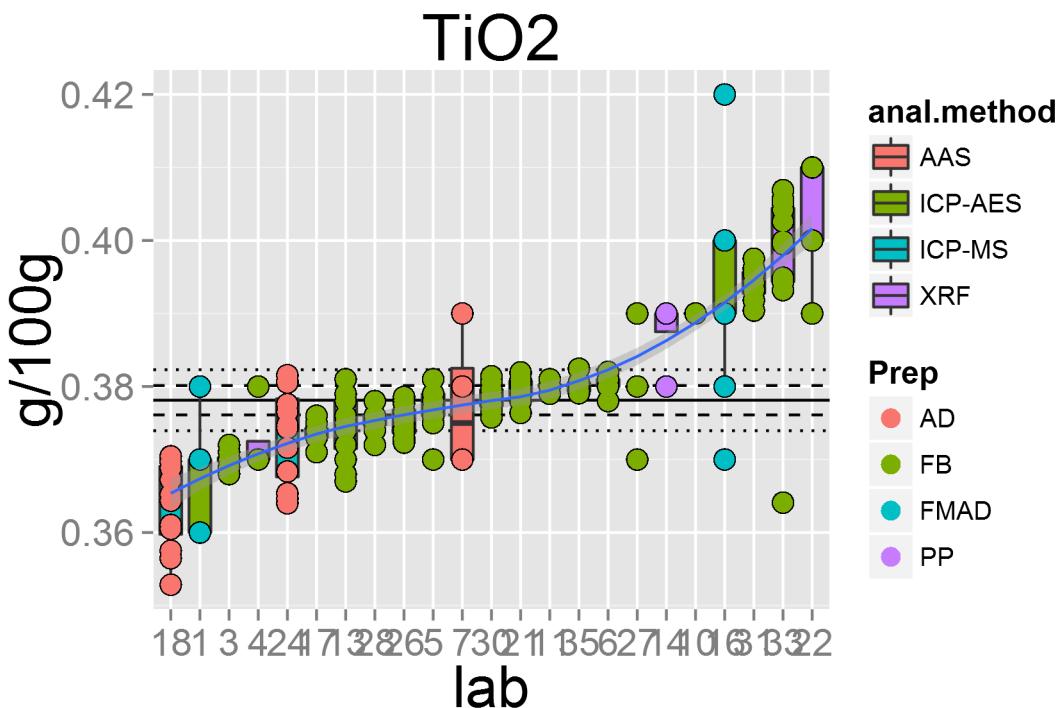
```



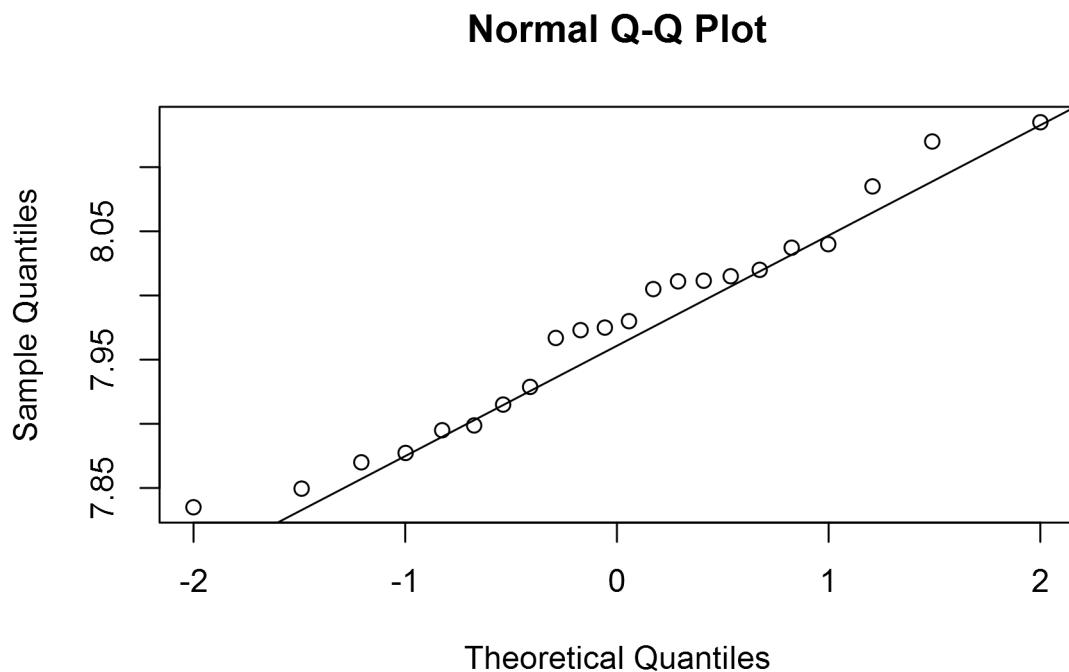
```
## [1] "TiO2"
```



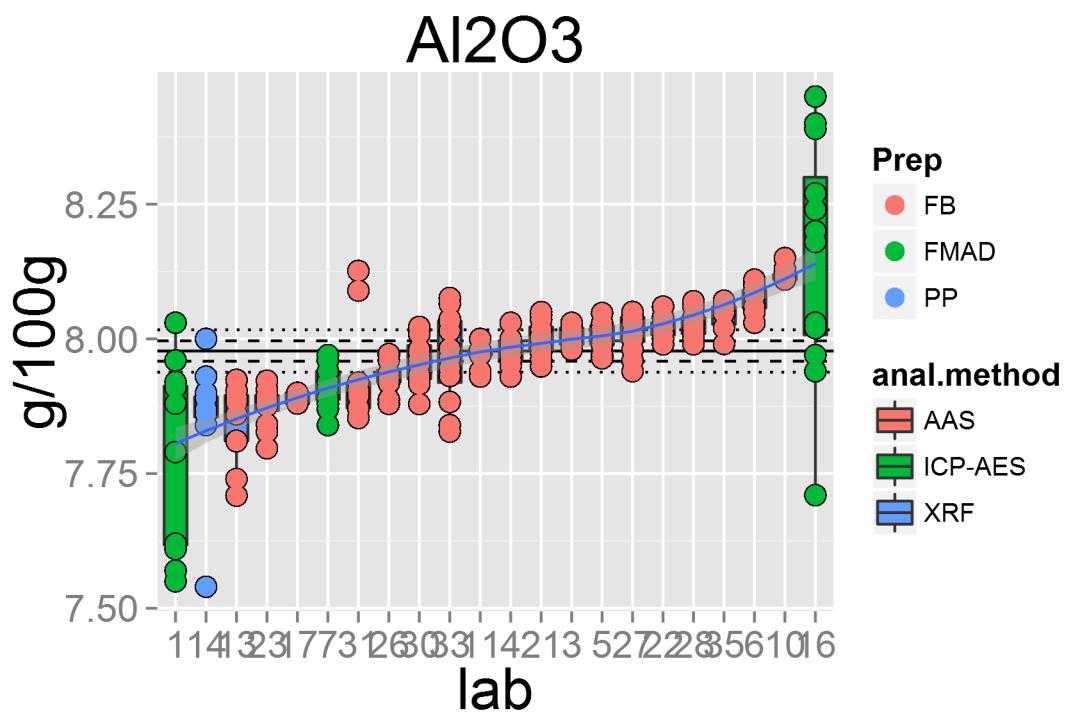
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



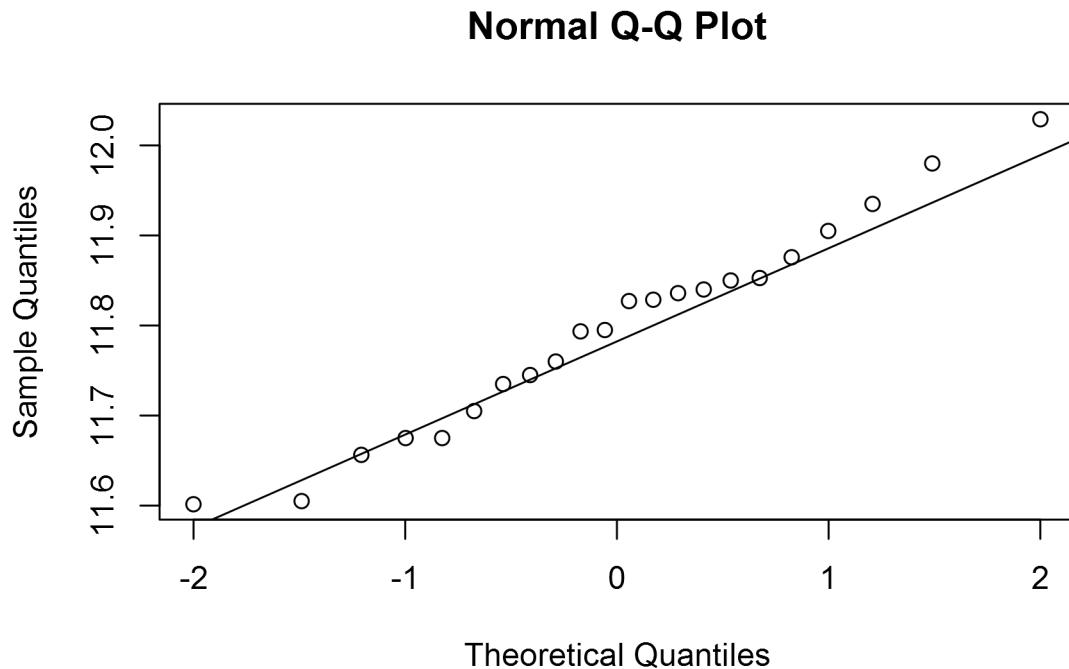
```
## [1] "Al2O3"
```



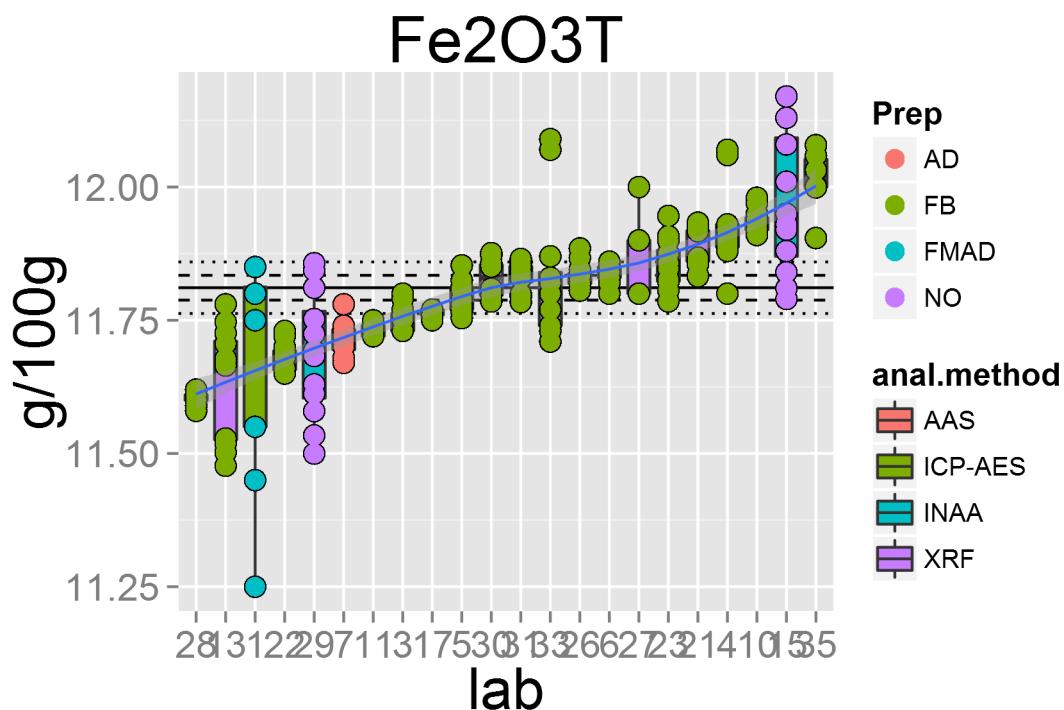
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



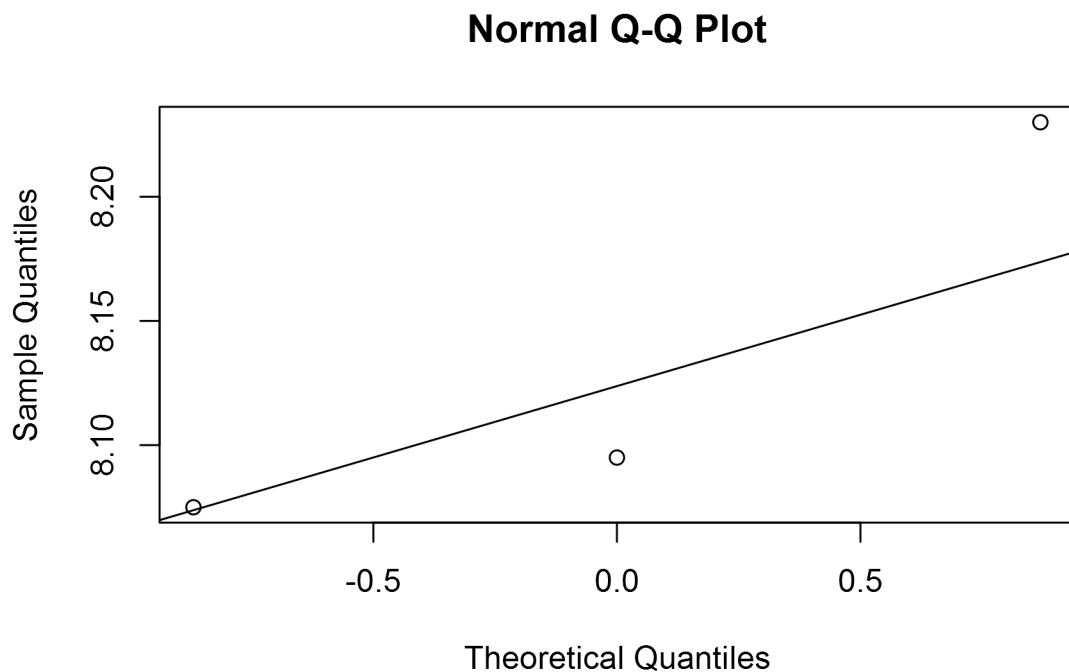
```
## [1] "Fe203T"
```



```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



```
## [1] "FeO"
```

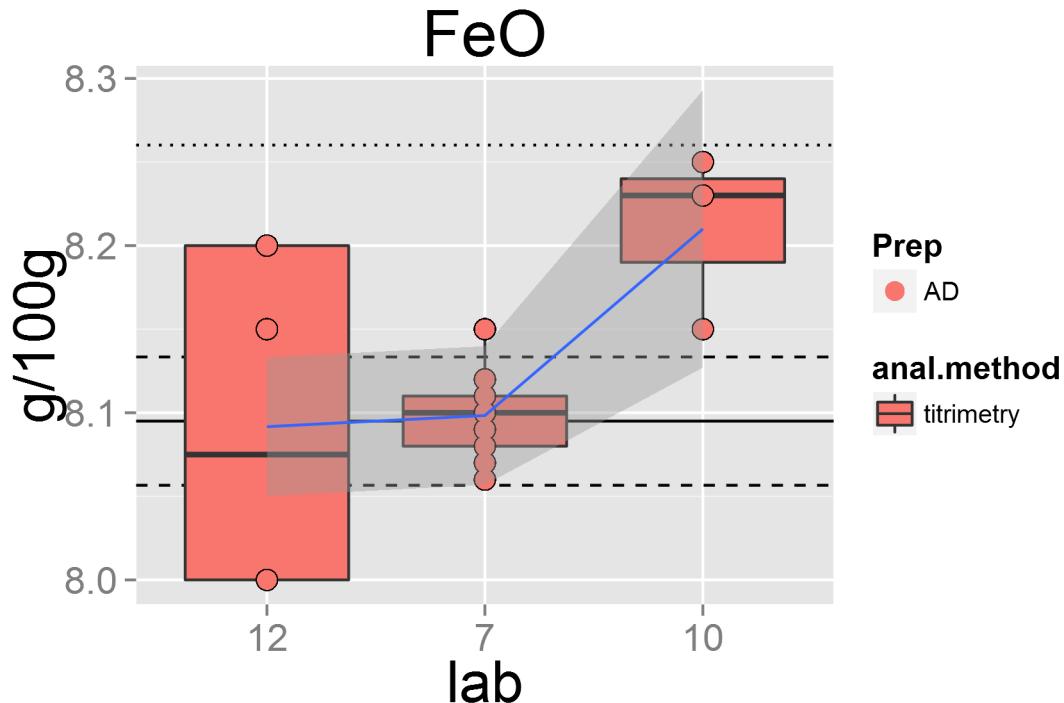


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

```

## Warning: pseudoinverse used at 0.99
## Warning: neighborhood radius 1.01
## Warning: reciprocal condition number  0
## Warning: There are other near singularities as well. 4.0401
## Warning: pseudoinverse used at 0.99
## Warning: neighborhood radius 1.01
## Warning: reciprocal condition number  0
## Warning: There are other near singularities as well. 4.0401

```

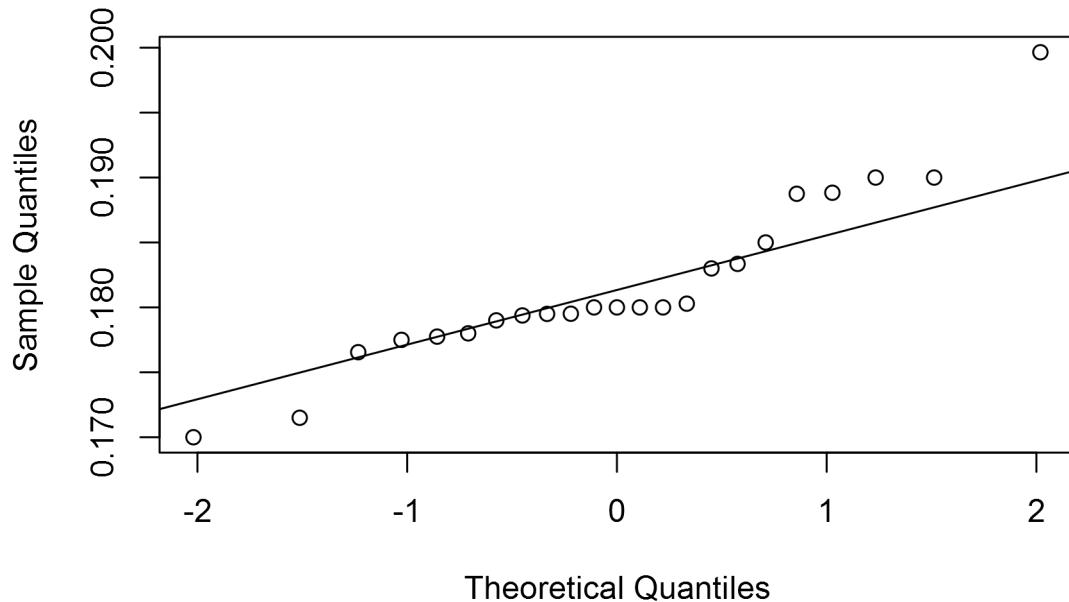


```

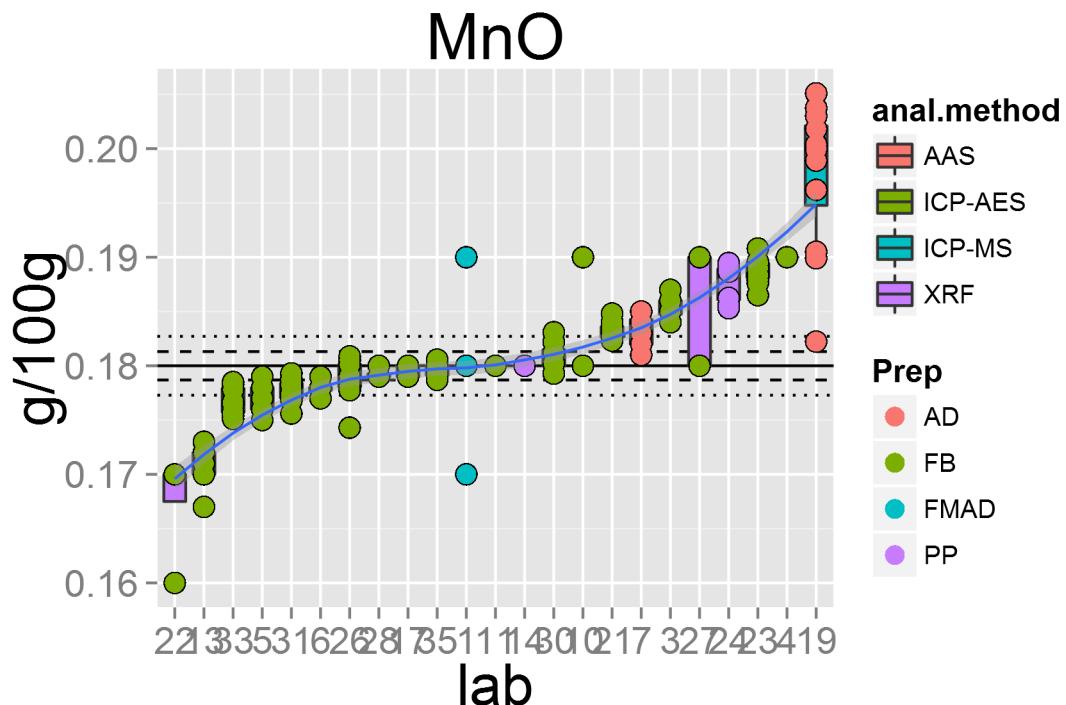
## [1] "MnO"

```

### Normal Q-Q Plot

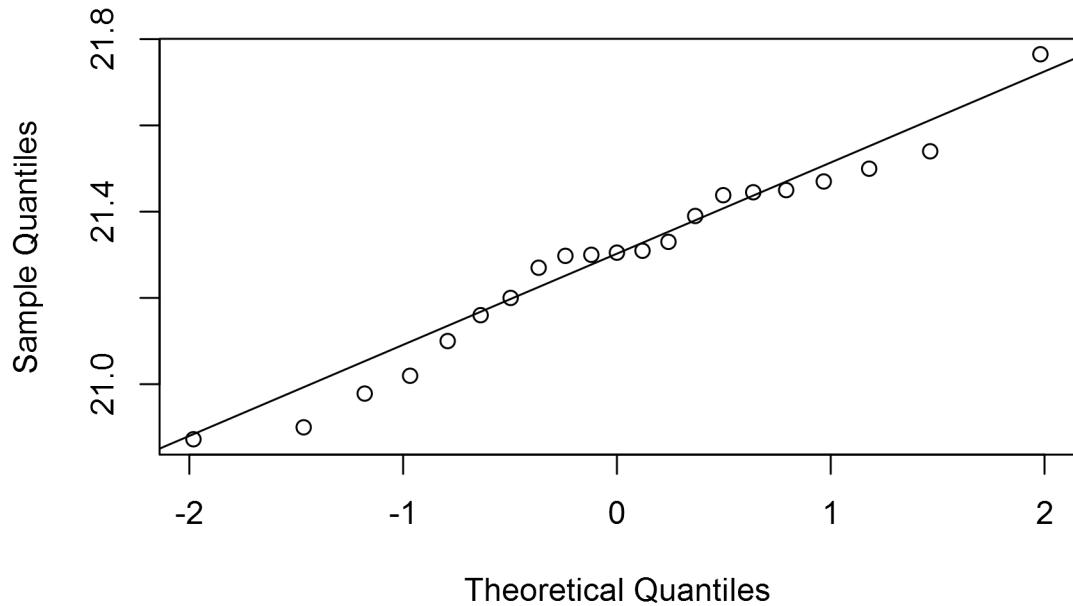


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

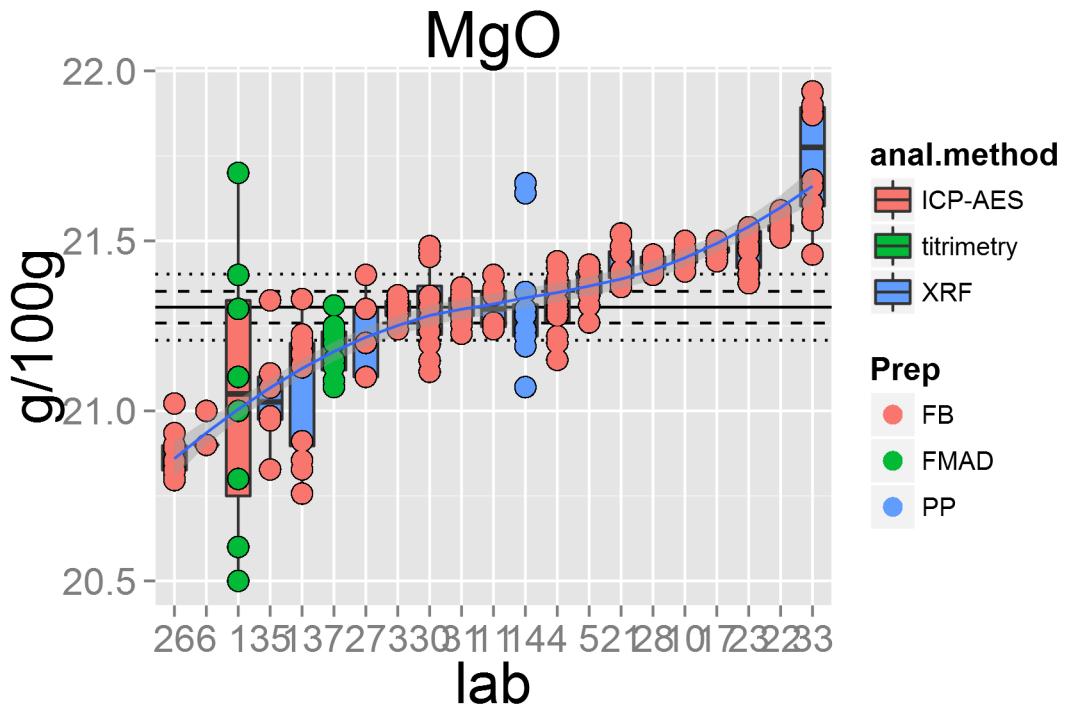


```
## [1] "MgO"
```

### Normal Q-Q Plot

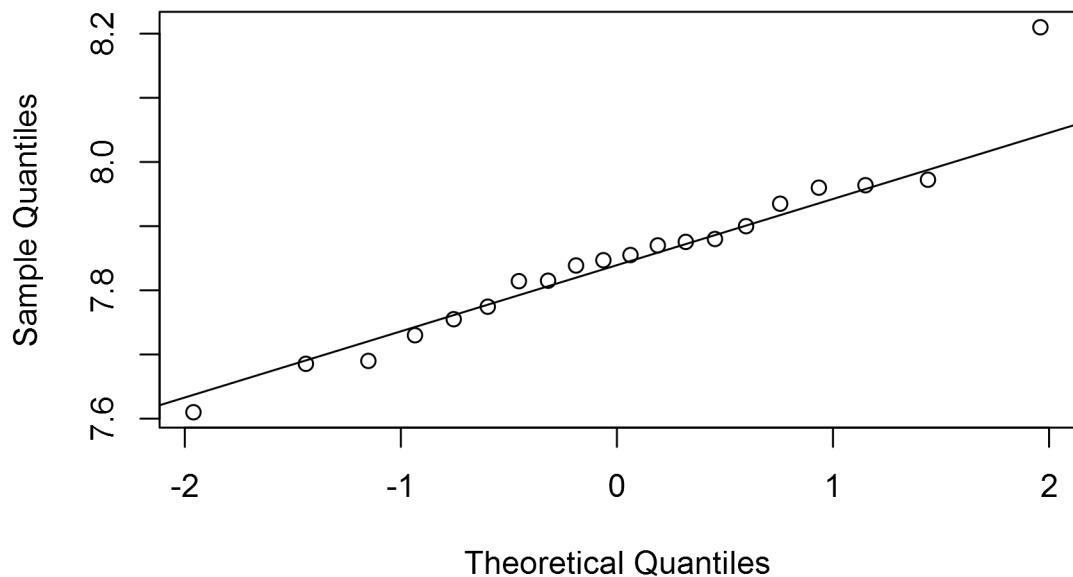


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

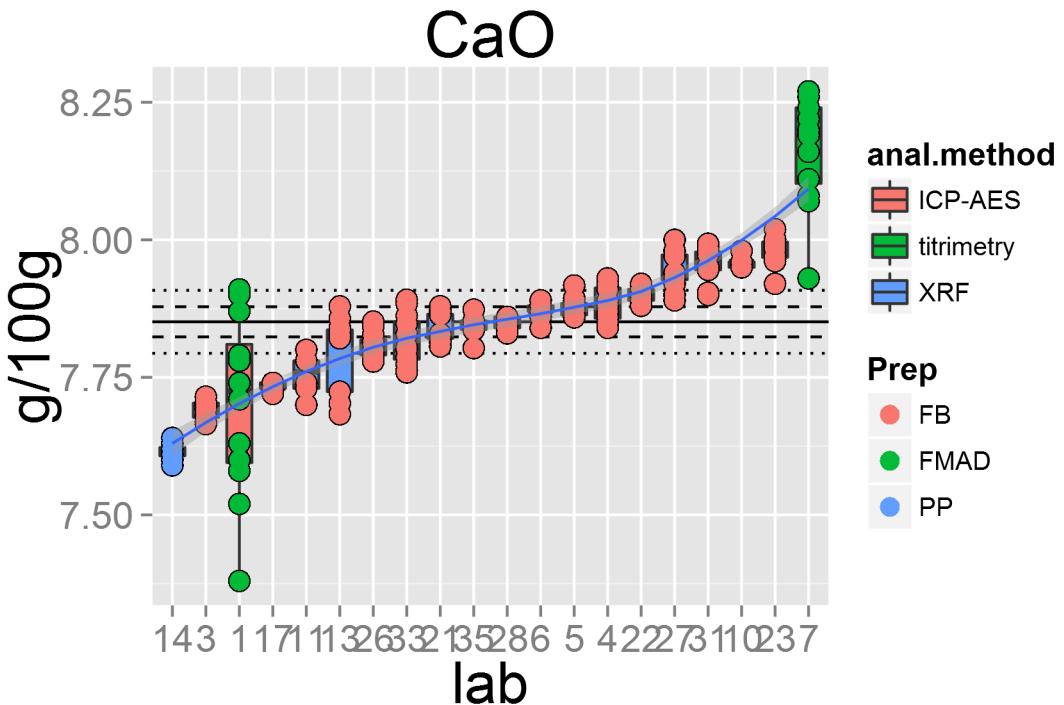


```
## [1] "CaO"
```

## Normal Q-Q Plot

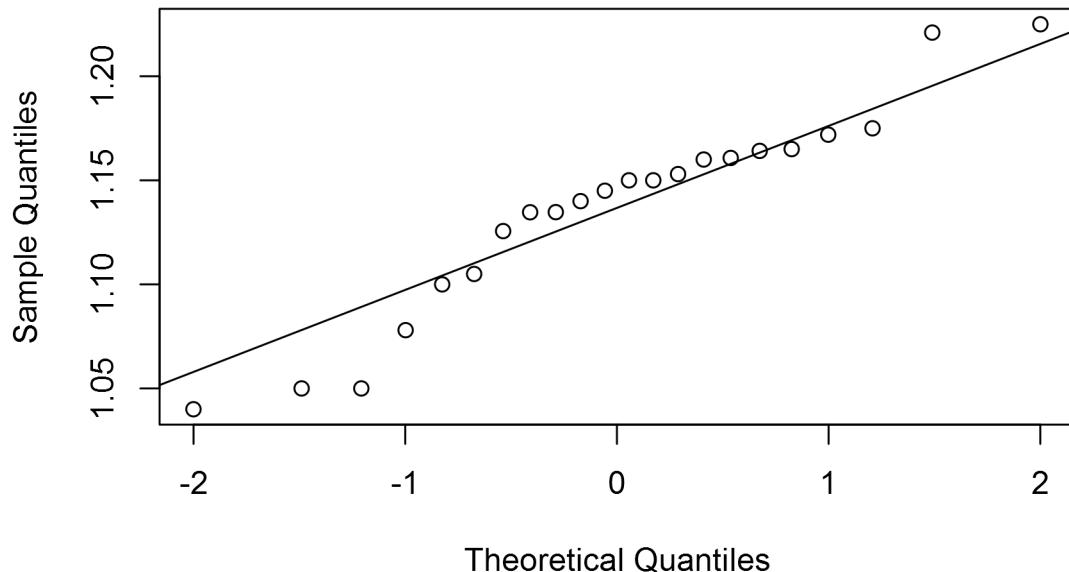


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

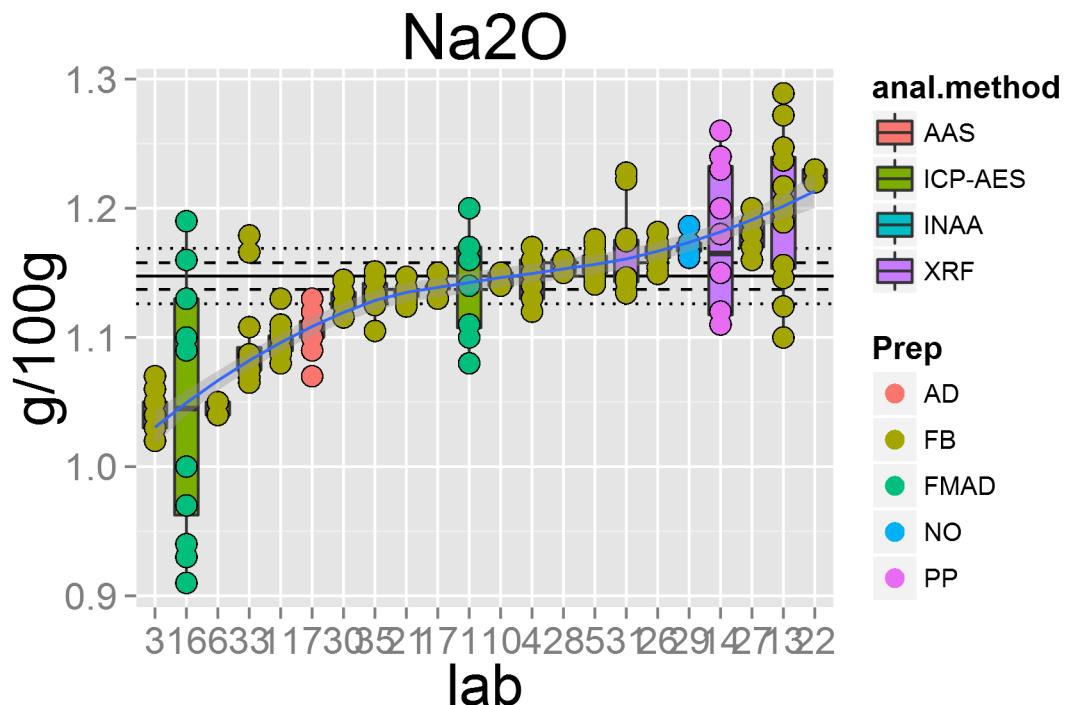


```
## [1] "Na20"
```

### Normal Q-Q Plot

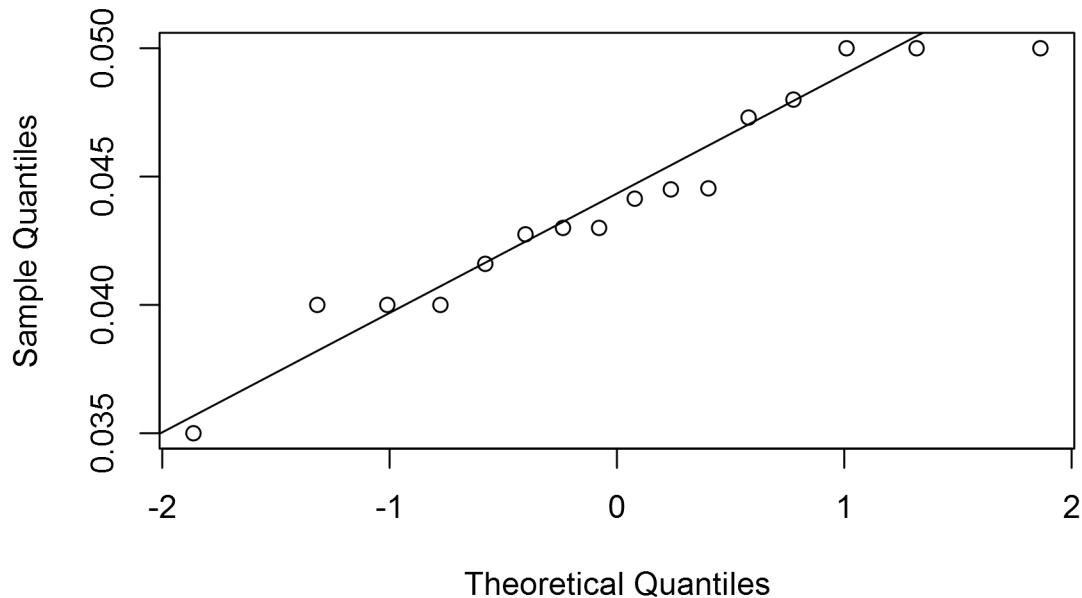


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



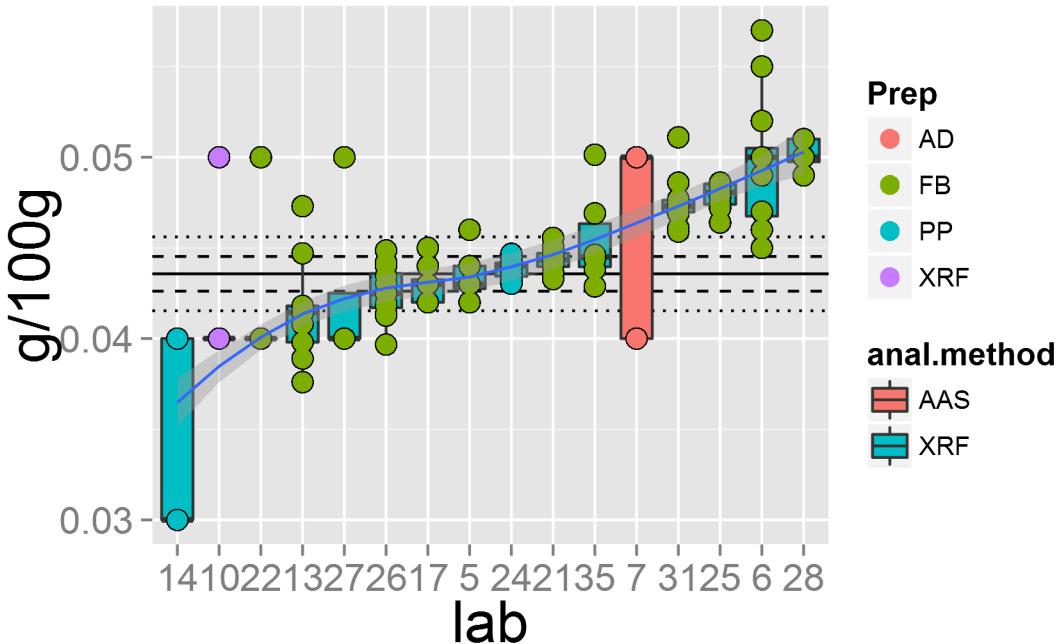
```
## [1] "K20"
```

### Normal Q-Q Plot



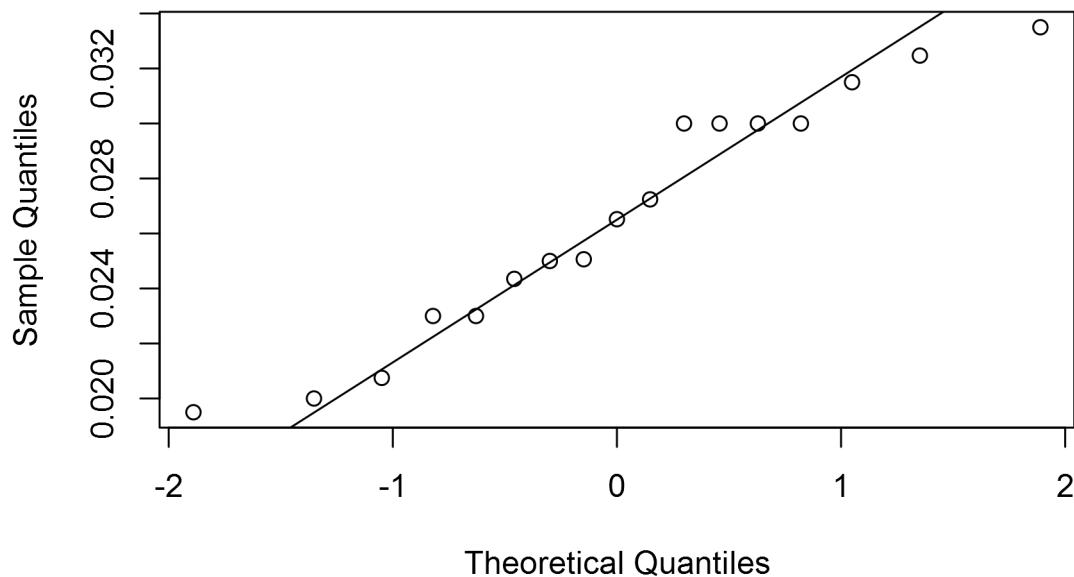
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

### K2O

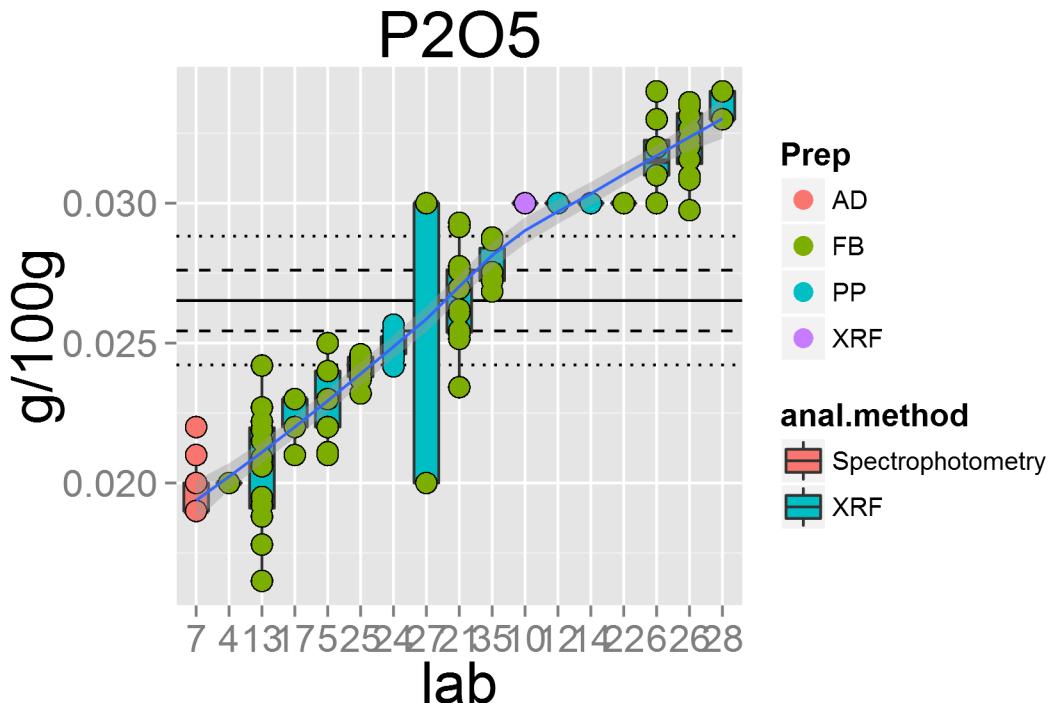


```
## [1] "P205"
```

## Normal Q-Q Plot

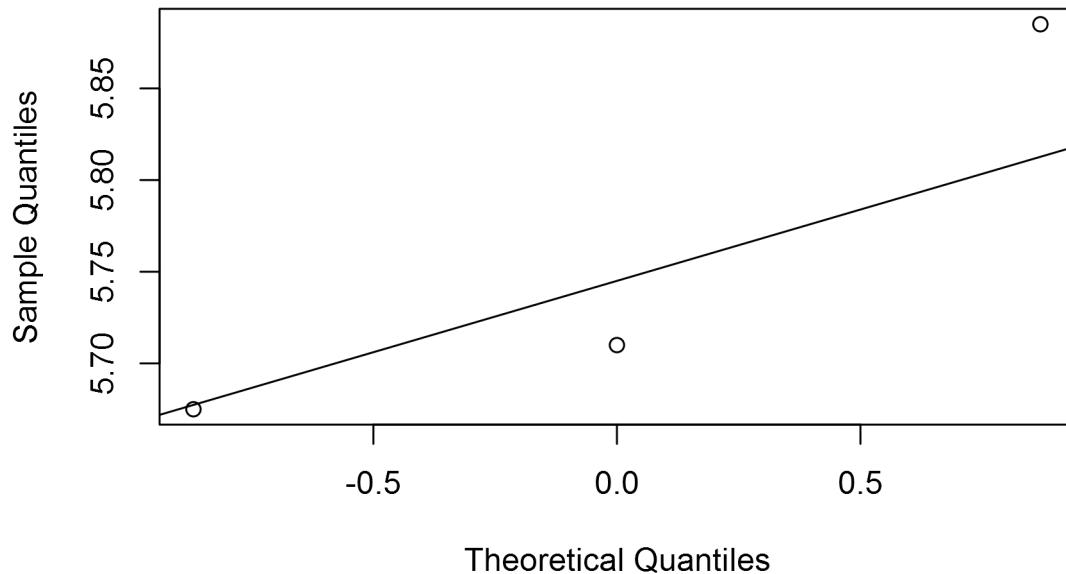


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

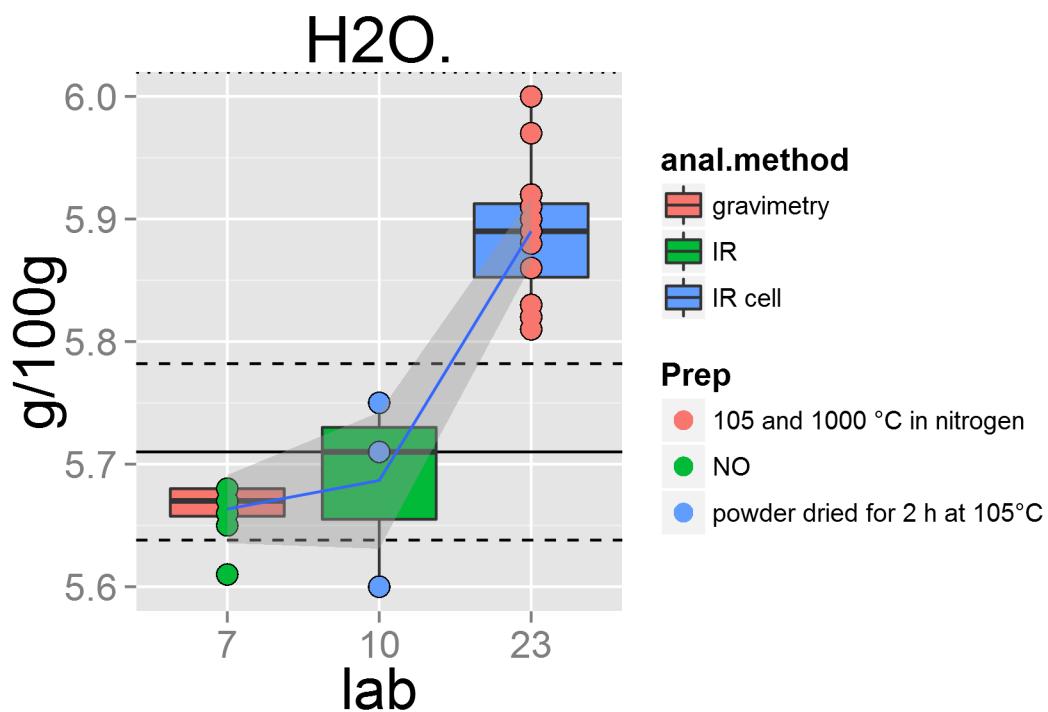


```
## [1] "H2O."
```

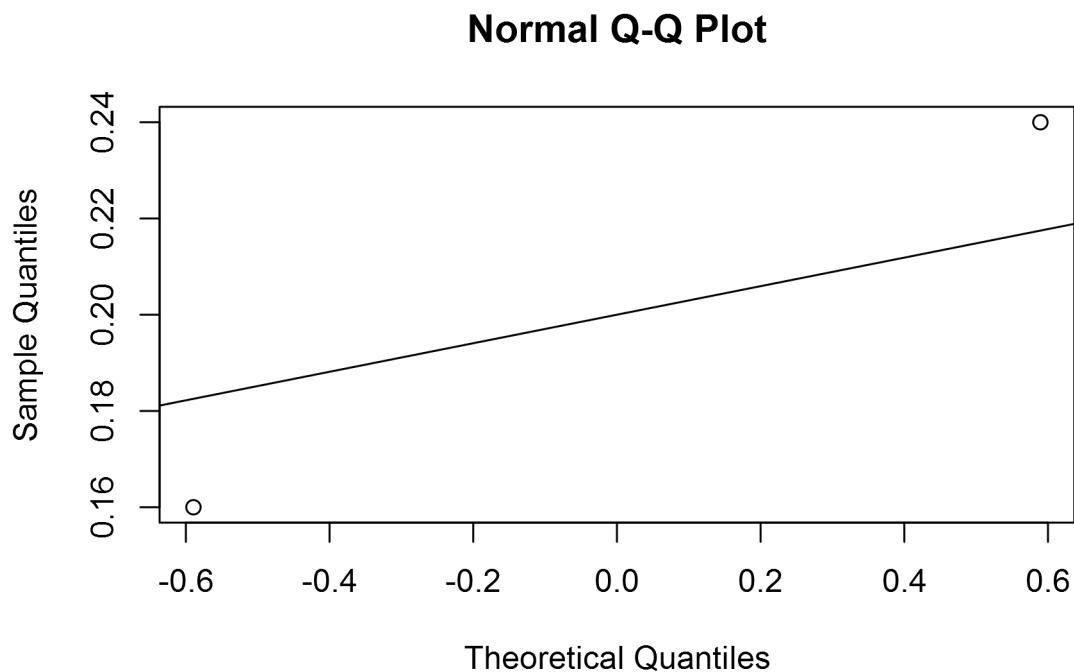
## Normal Q-Q Plot



```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
## Warning: pseudoinverse used at 0.99
## Warning: neighborhood radius 2.01
## Warning: reciprocal condition number  0
## Warning: There are other near singularities as well. 4.0401
## Warning: pseudoinverse used at 0.99
## Warning: neighborhood radius 2.01
## Warning: reciprocal condition number  0
## Warning: There are other near singularities as well. 4.0401
```



```
## [1] "CO2"
```

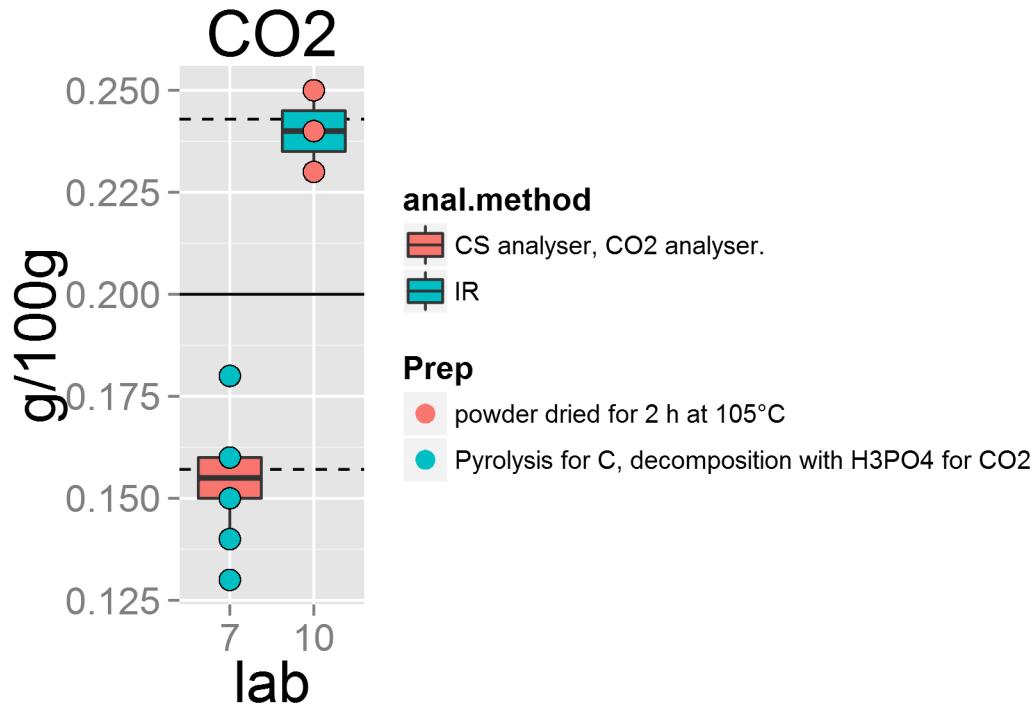


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

```

## Warning: at  0.995
## Warning: radius  2.5e-005
## Warning: all data on boundary of neighborhood. make span bigger
## Warning: pseudoinverse used at 0.995
## Warning: neighborhood radius 0.005
## Warning: reciprocal condition number  1
## Warning: There are other near singularities as well. 1.01
## Warning: zero-width neighborhood. make span bigger

```

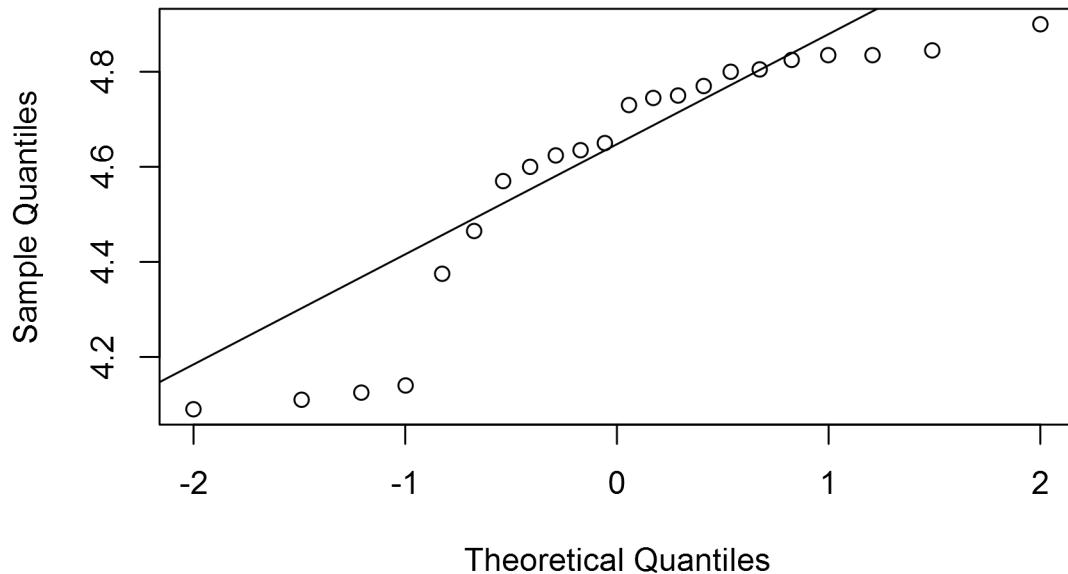


```

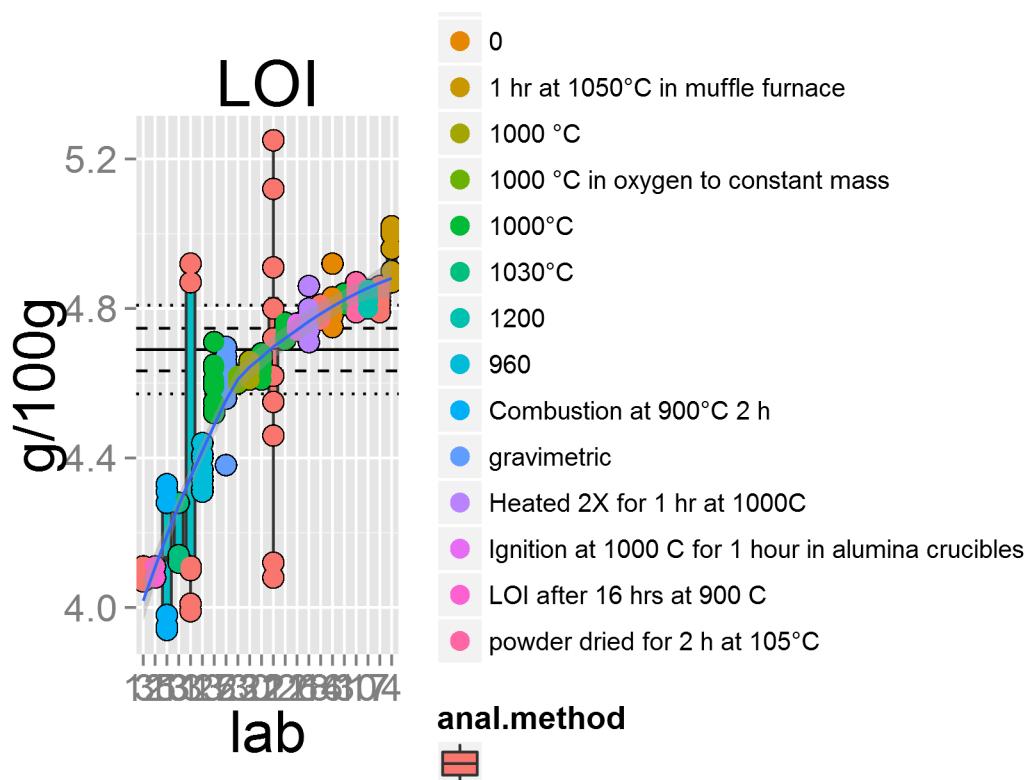
## [1] "LOI"

```

## Normal Q-Q Plot

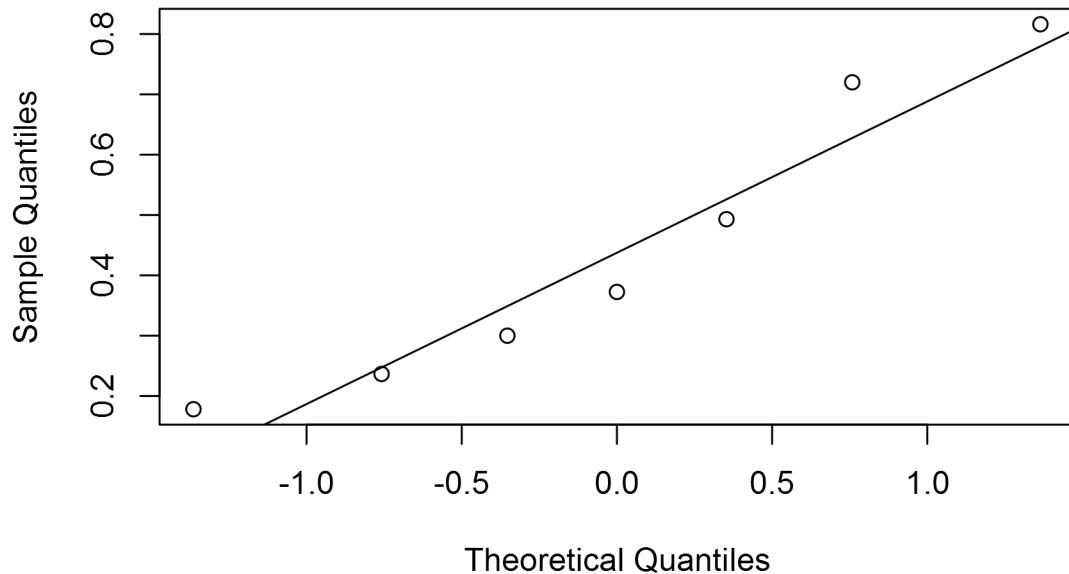


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

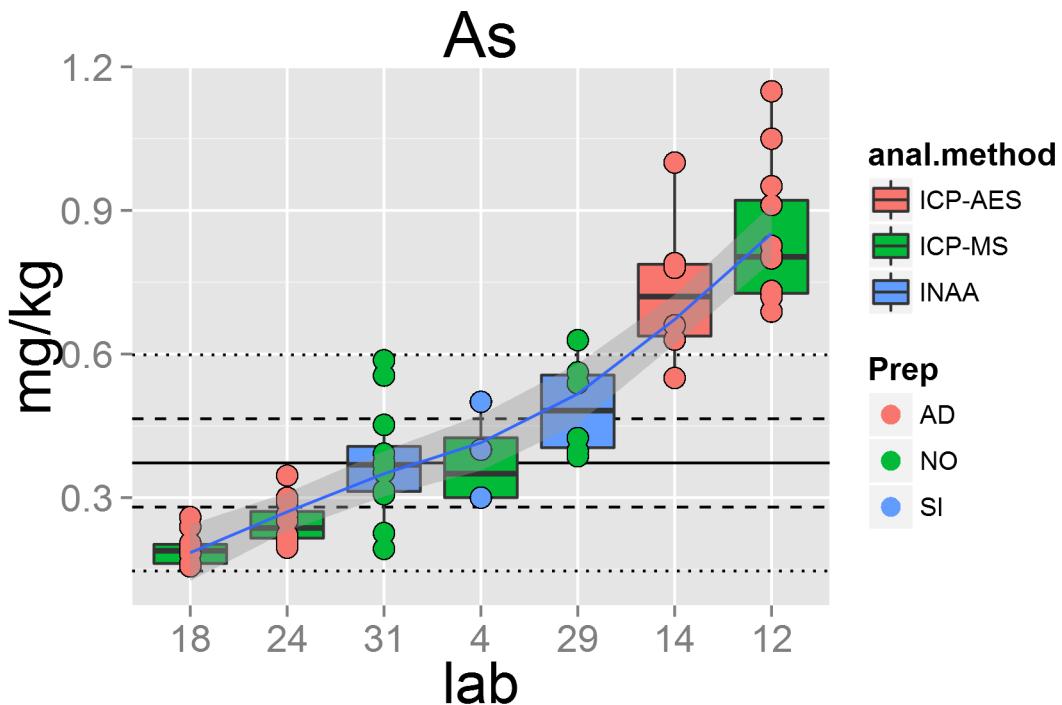


```
## [1] "As"
```

### Normal Q-Q Plot

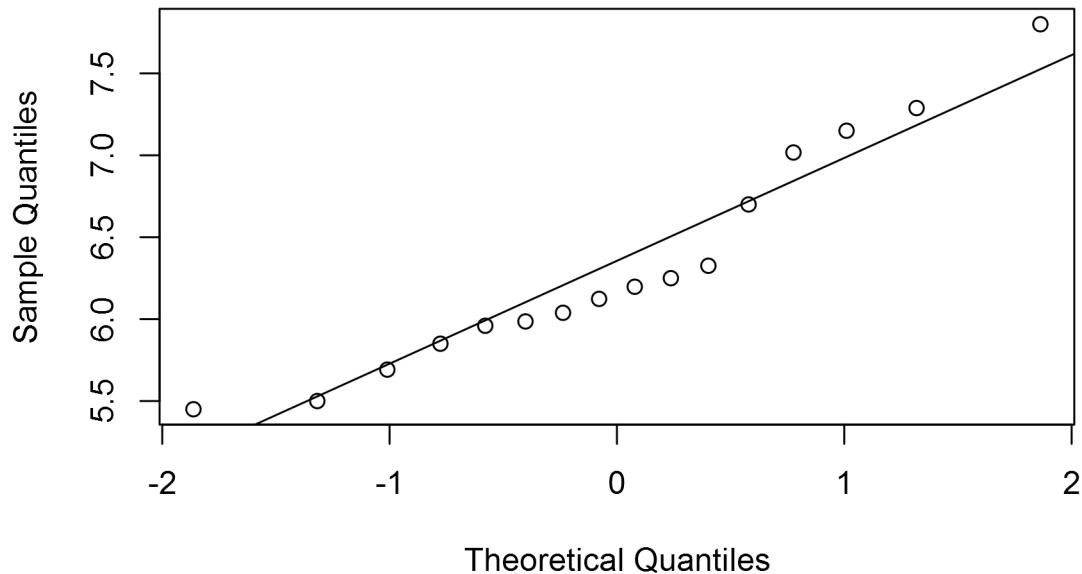


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

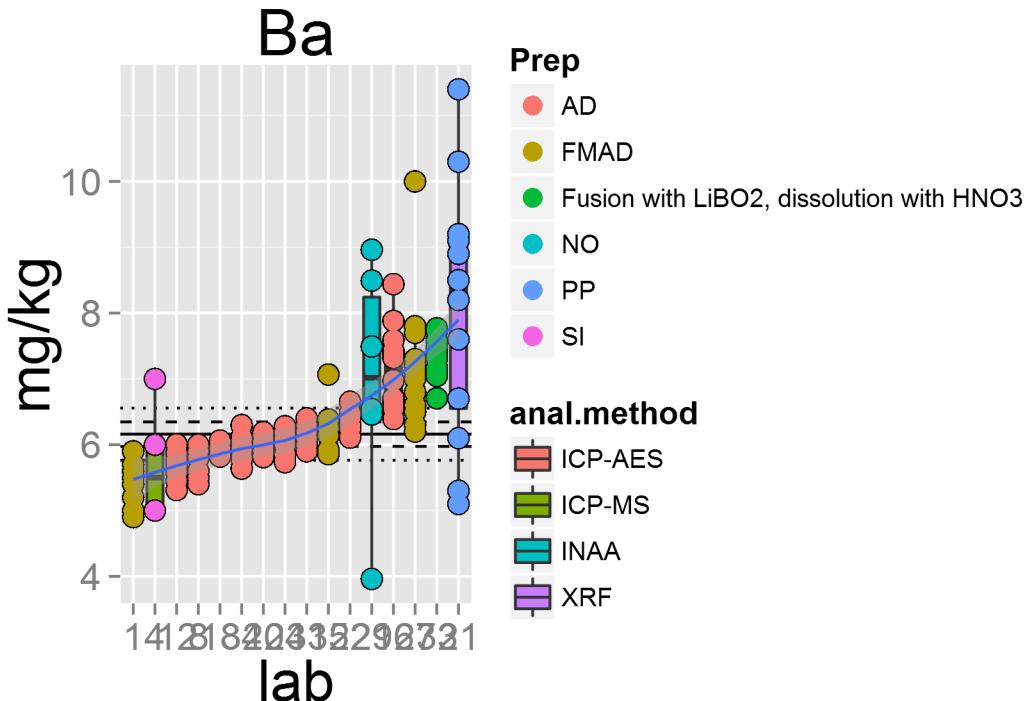


```
## [1] "Ba"
```

### Normal Q-Q Plot

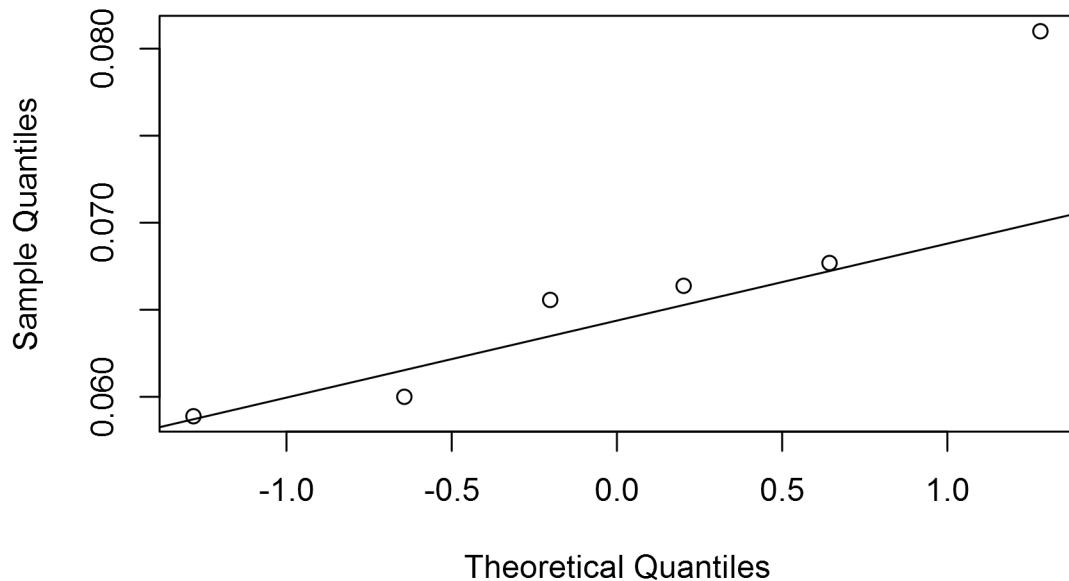


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

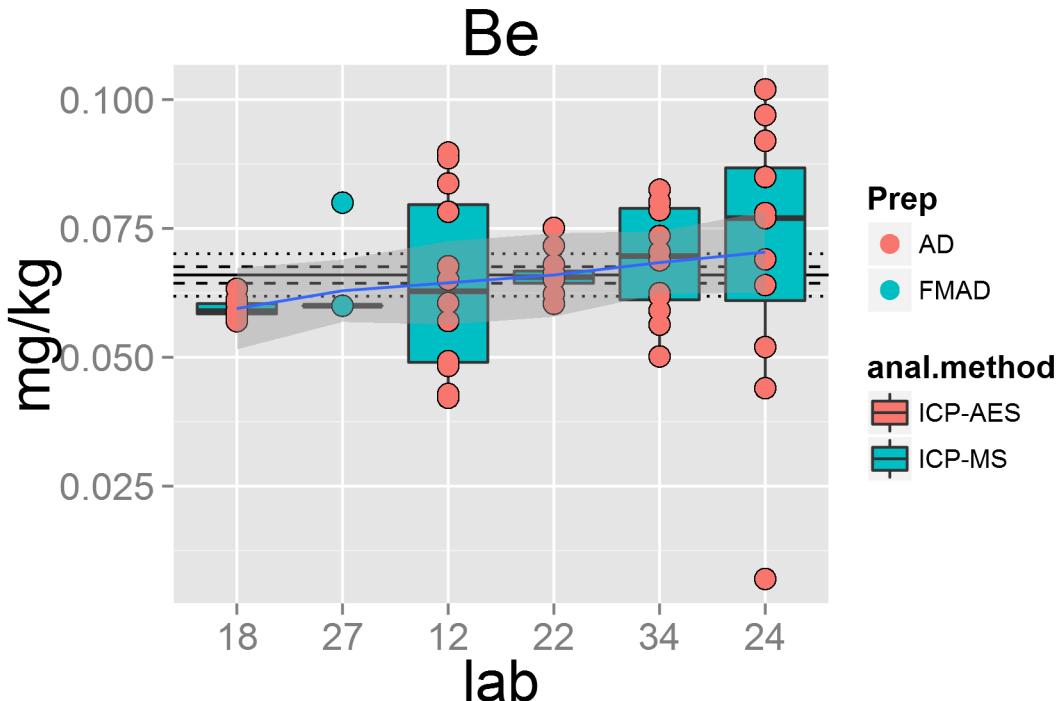


```
## [1] "Be"
```

### Normal Q-Q Plot

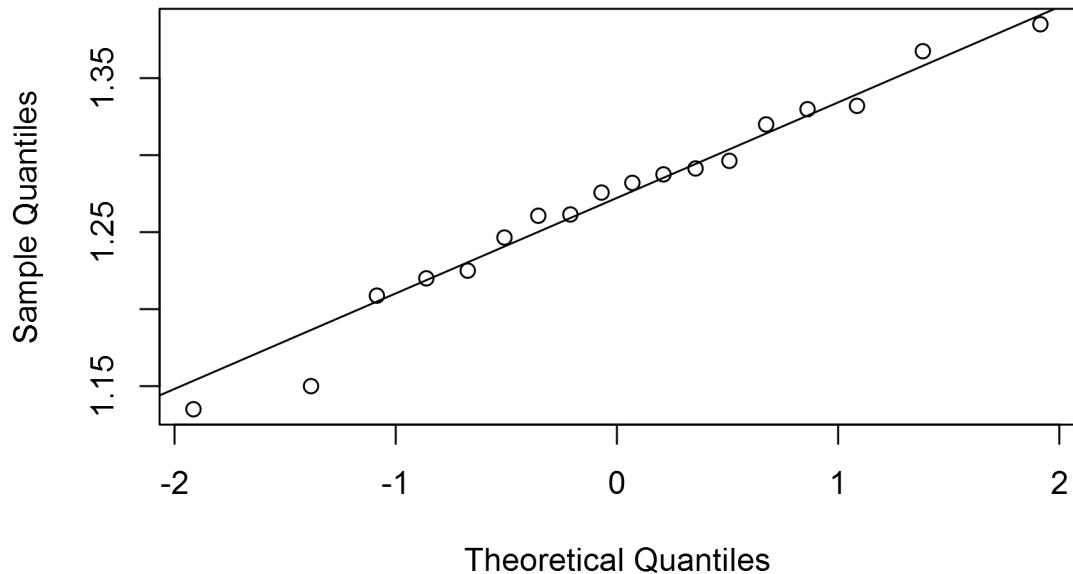


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

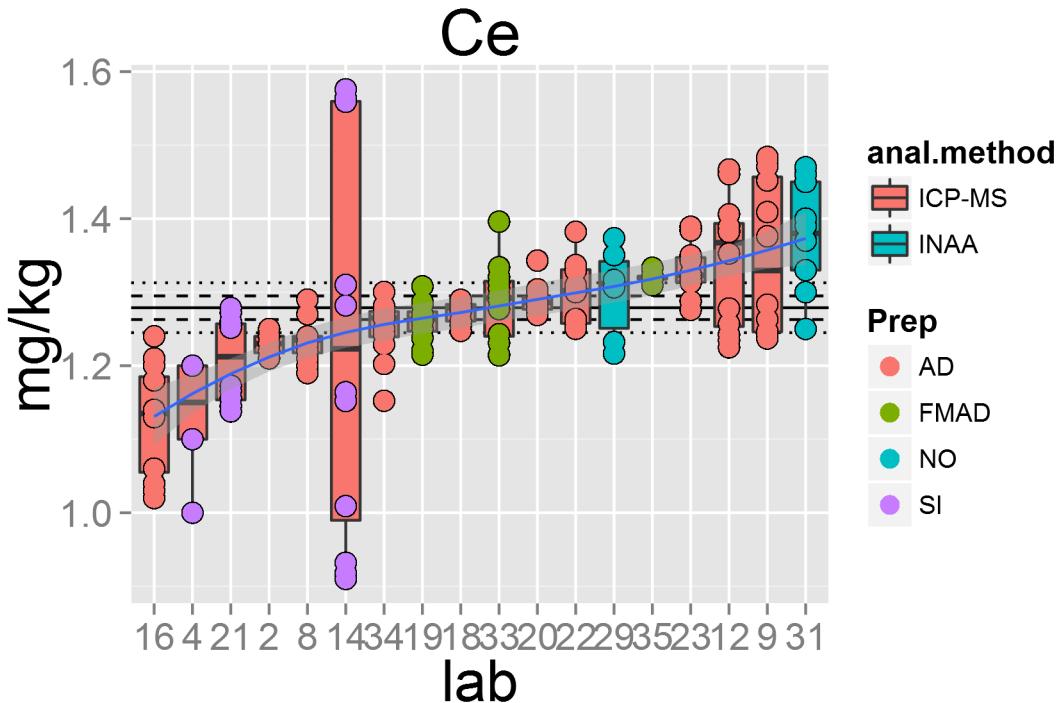


```
## [1] "Ce"
```

## Normal Q-Q Plot

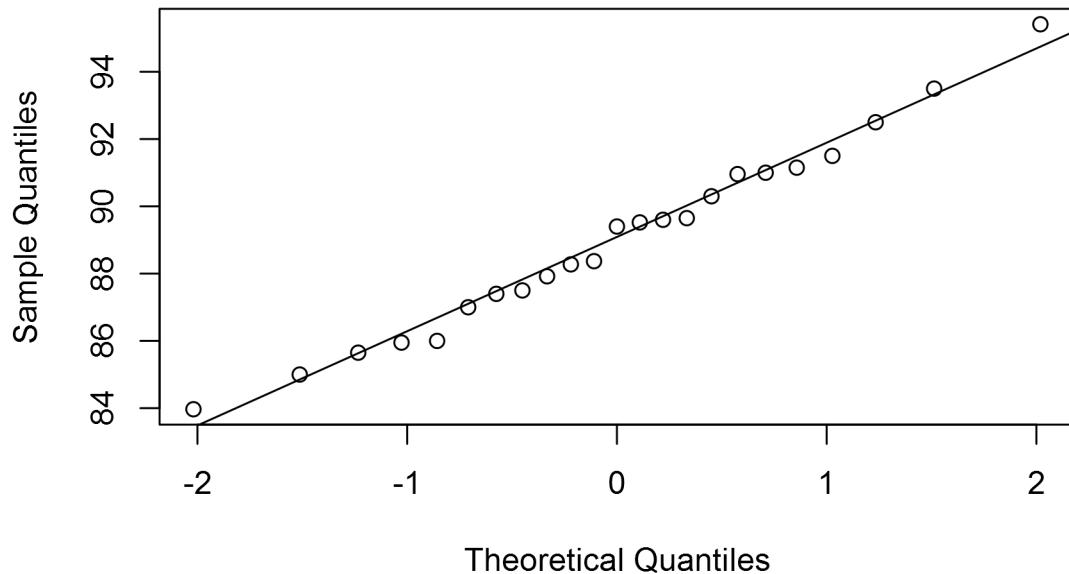


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

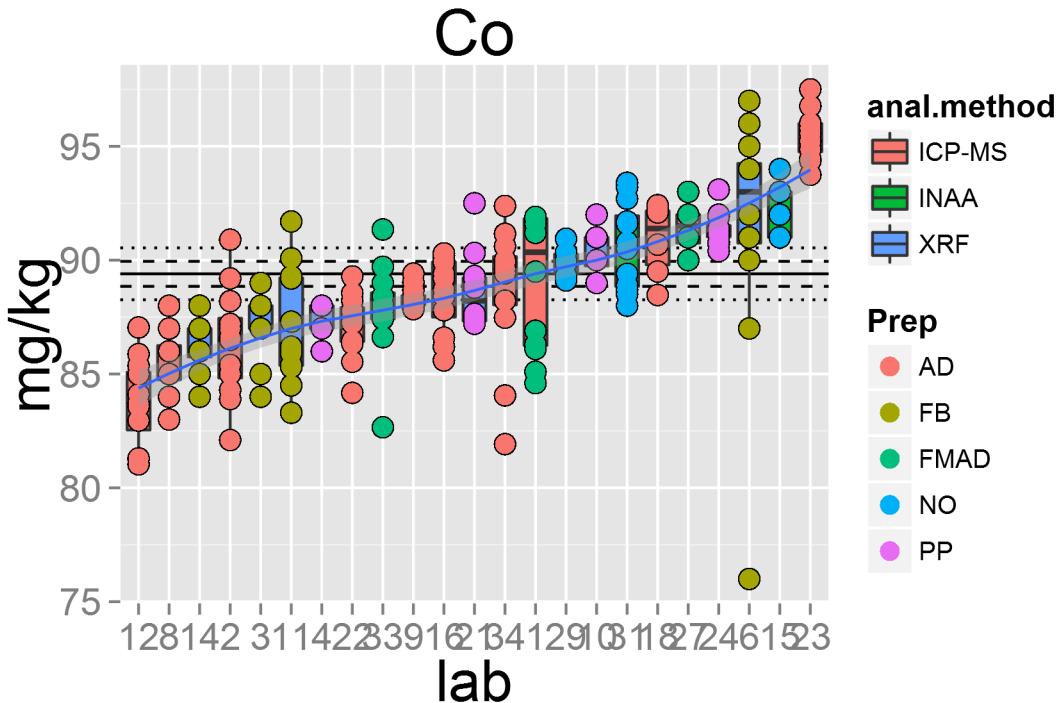


```
## [1] "Co"
```

### Normal Q-Q Plot

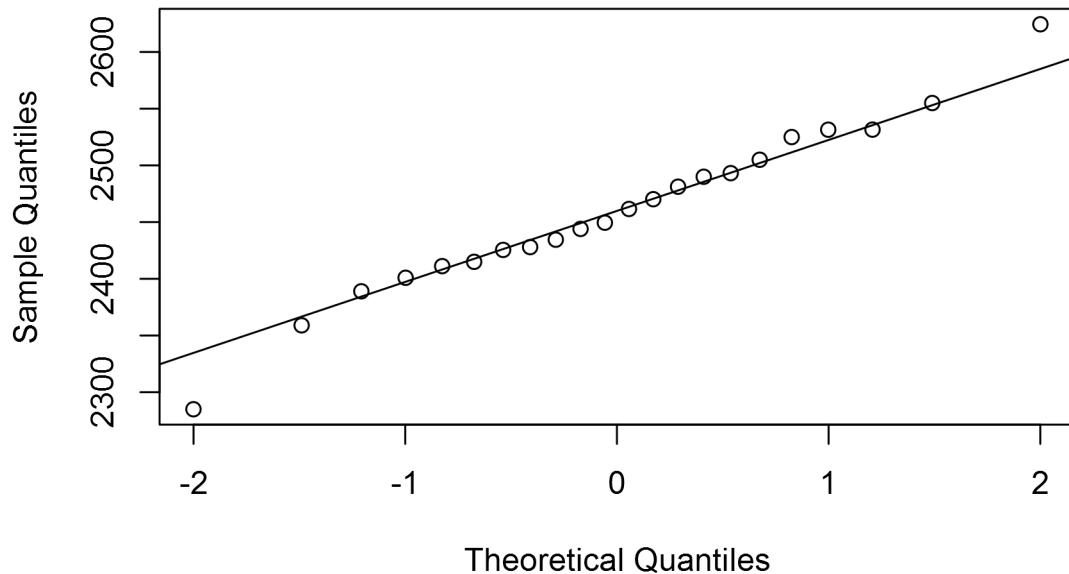


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

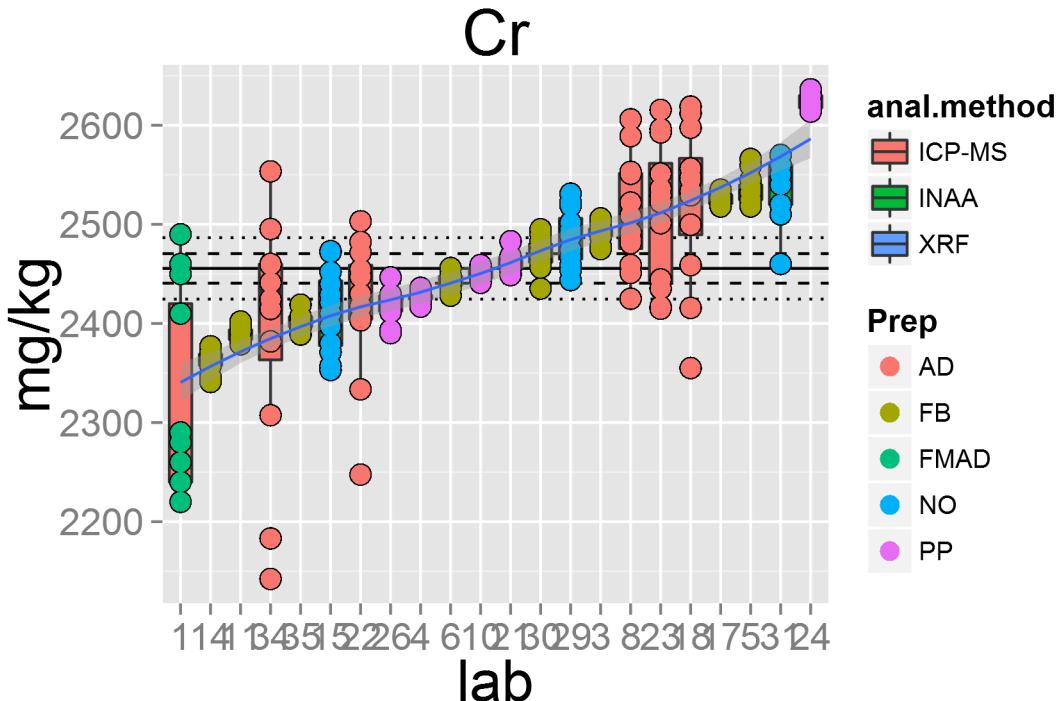


```
## [1] "Cr"
```

## Normal Q-Q Plot

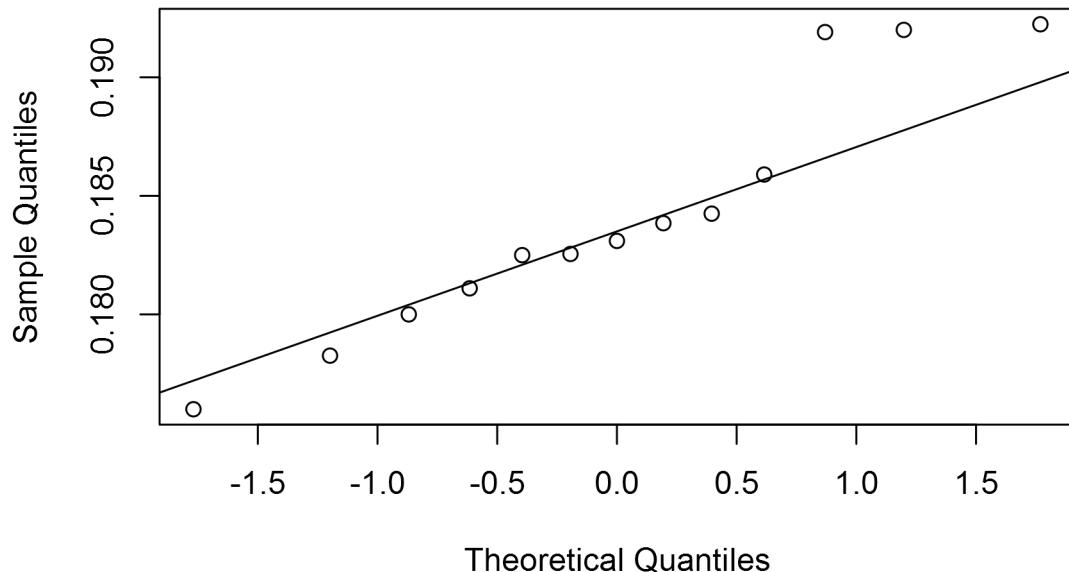


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

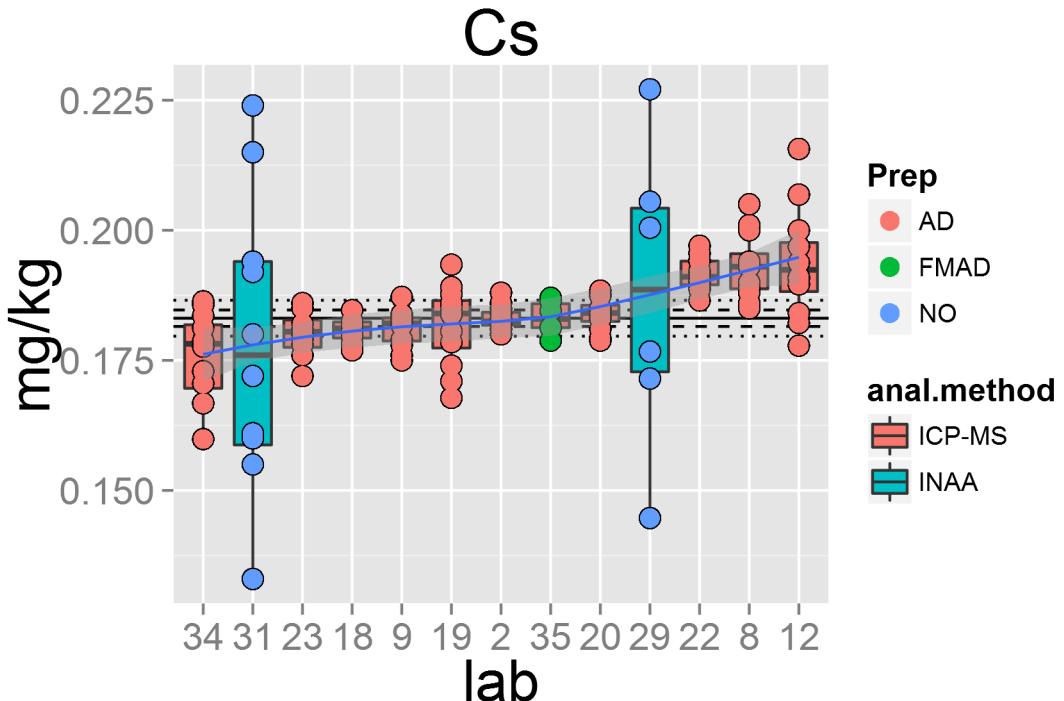


```
## [1] "Cs"
```

### Normal Q-Q Plot

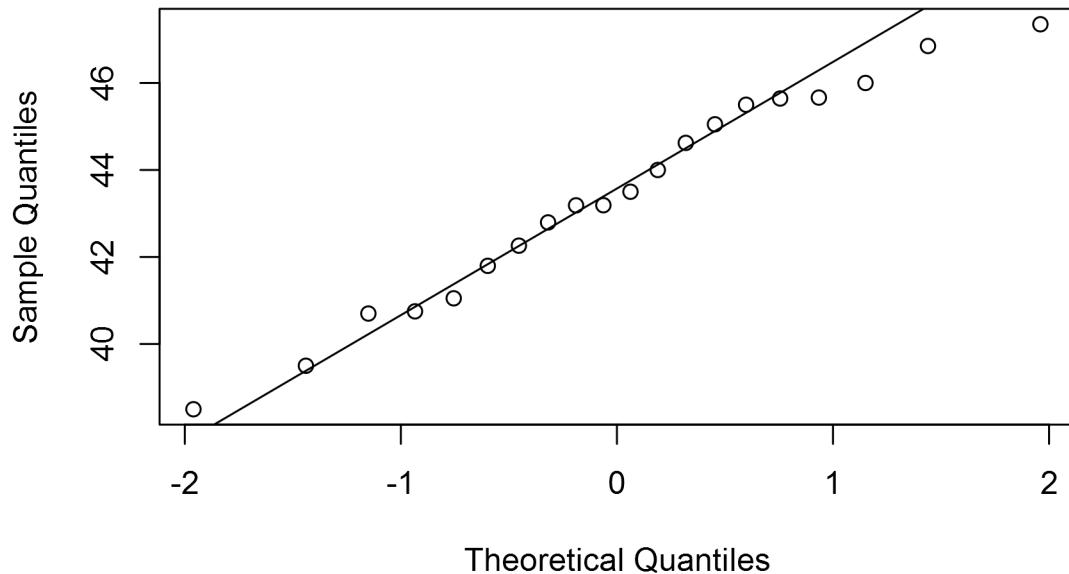


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

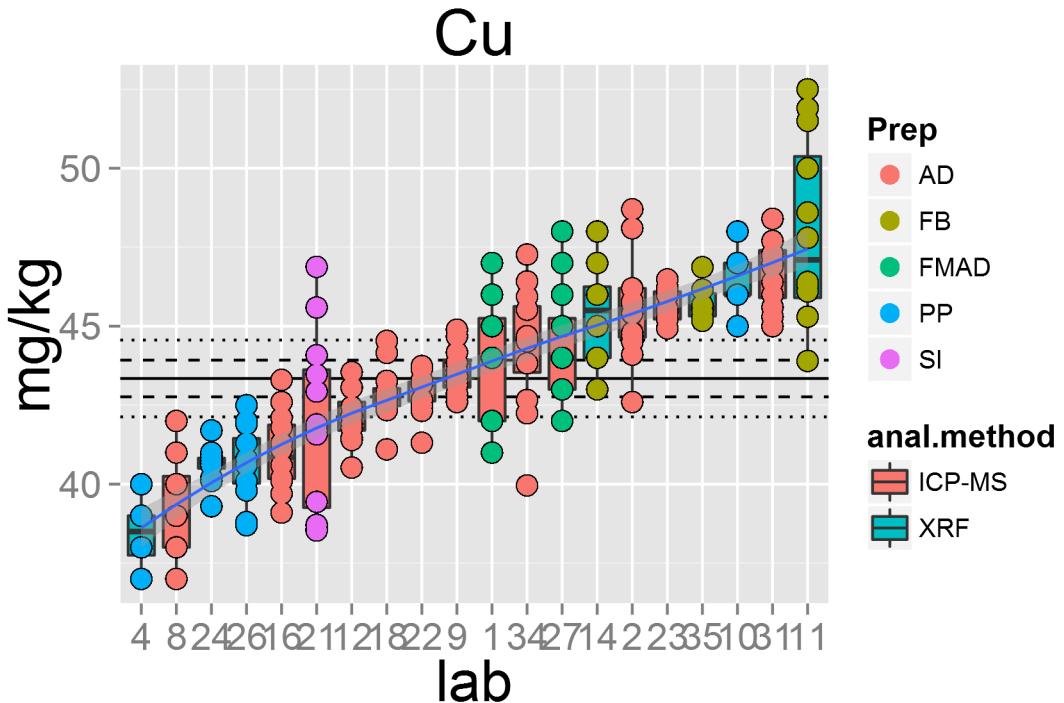


```
## [1] "Cu"
```

### Normal Q-Q Plot

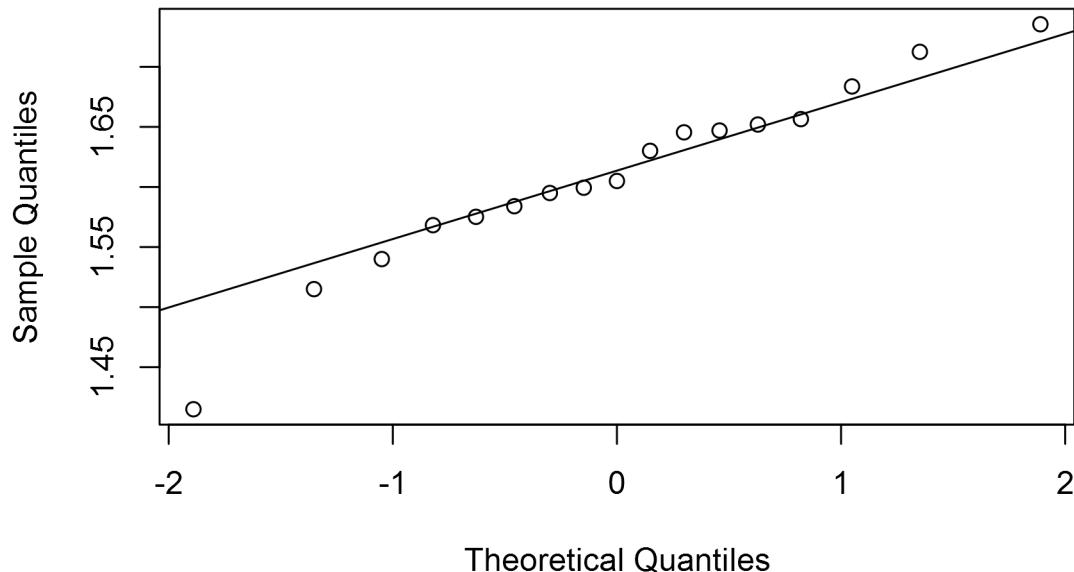


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

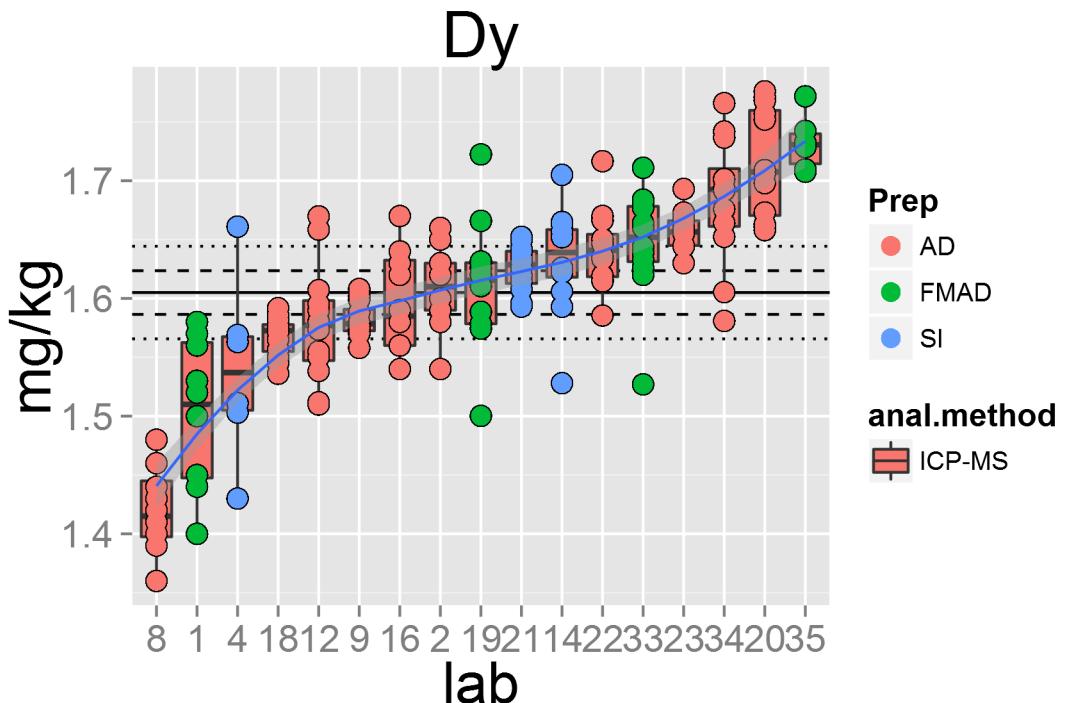


```
## [1] "Dy"
```

### Normal Q-Q Plot

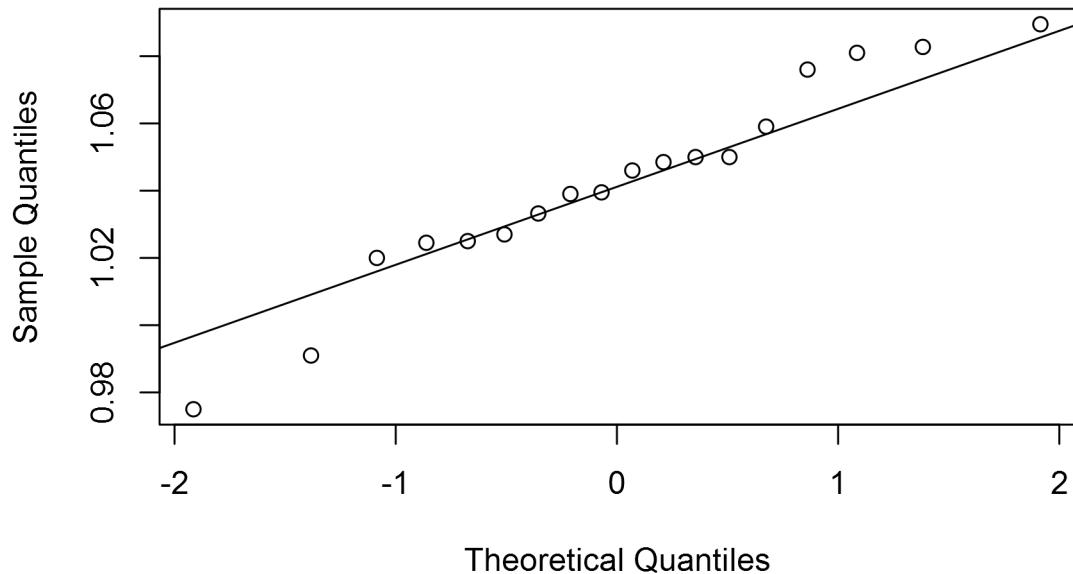


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

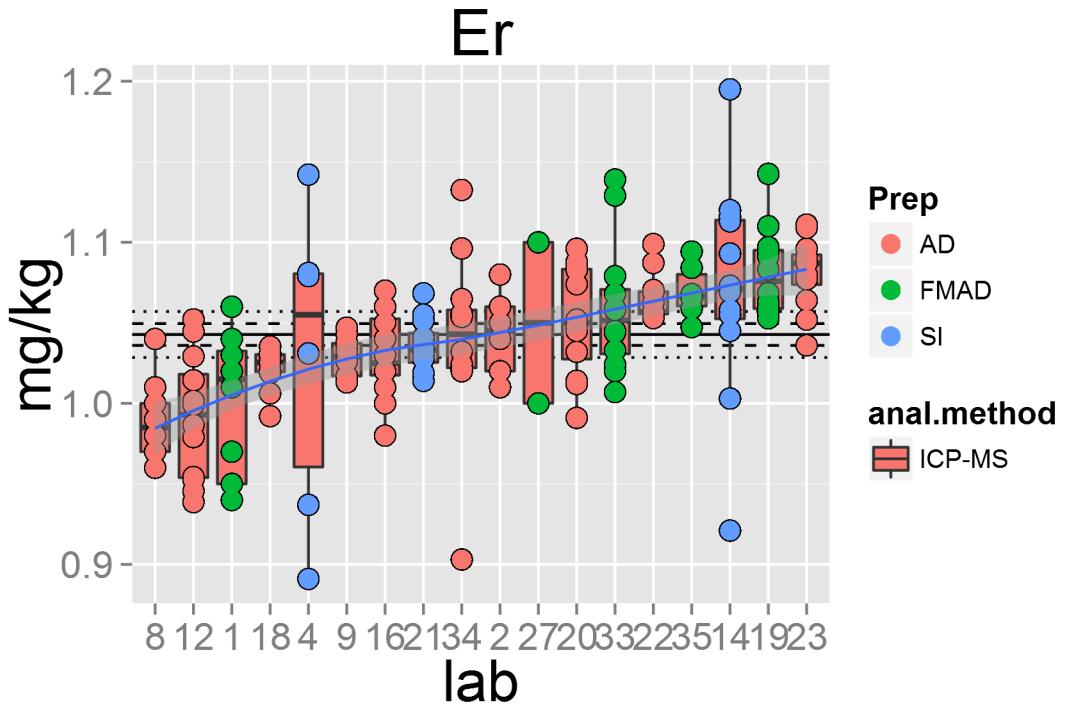


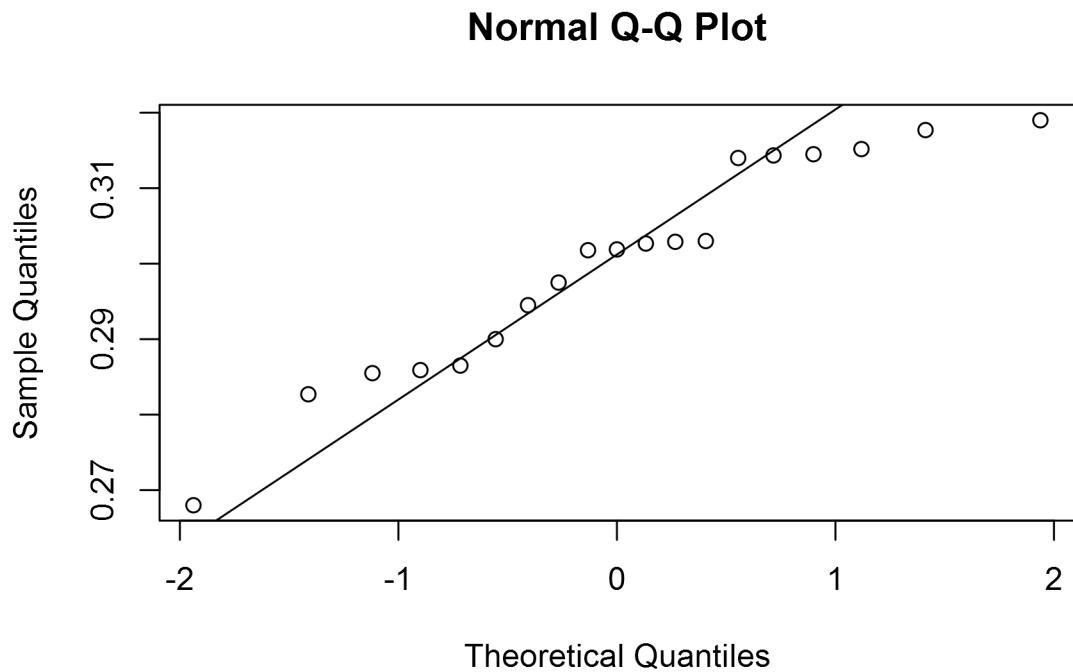
```
## [1] "Er"
```

### Normal Q-Q Plot

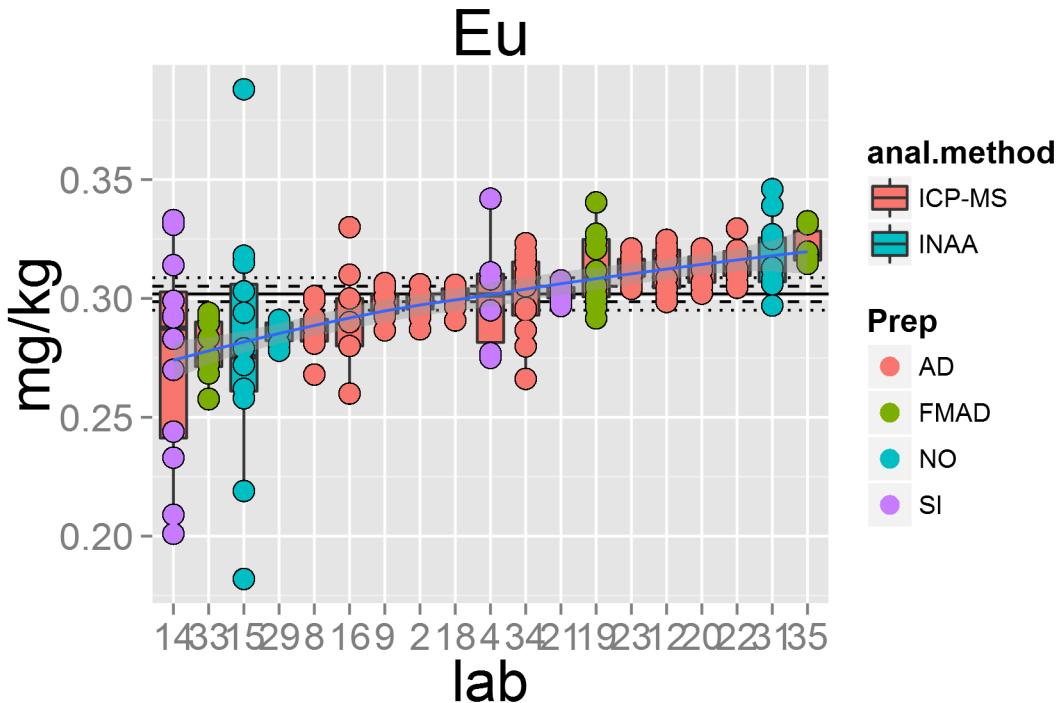


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



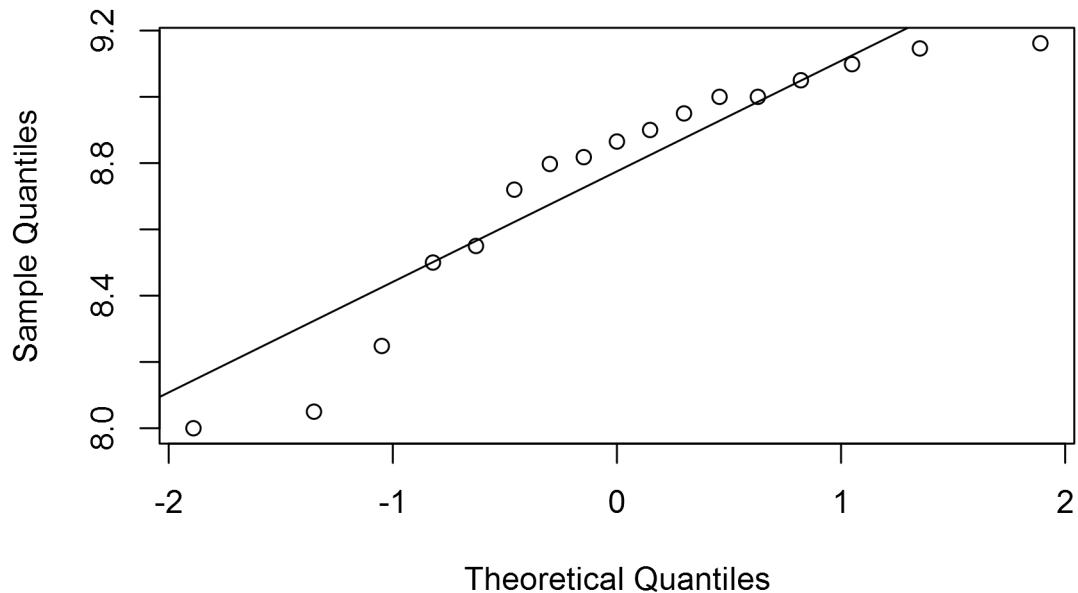


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

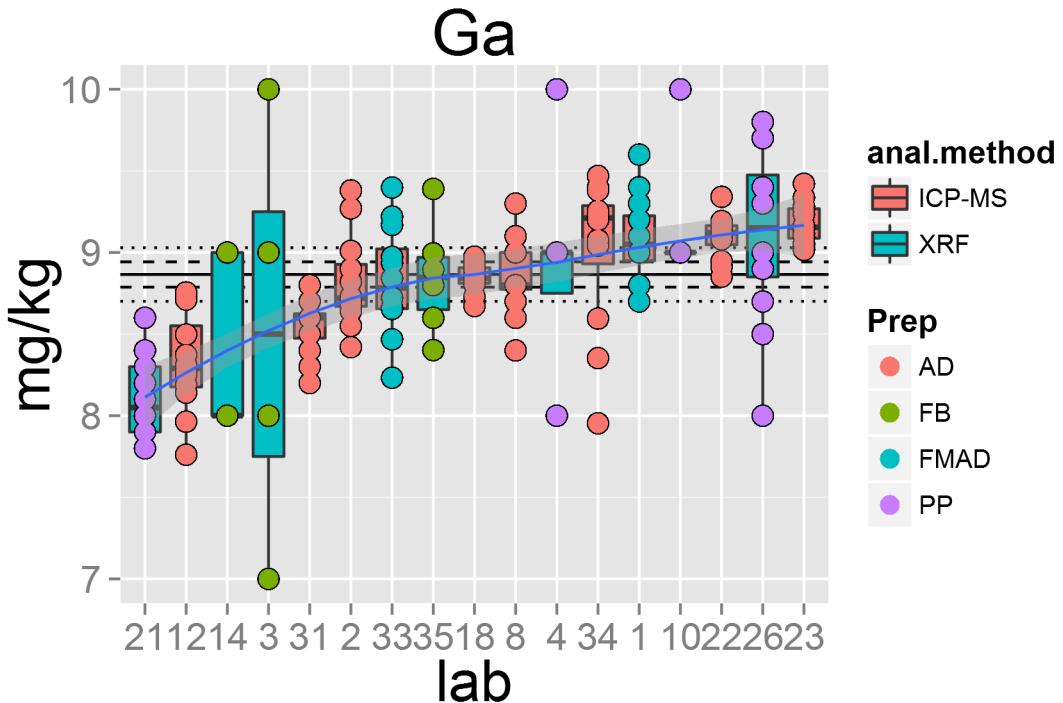


```
## [1] "Ga"
```

### Normal Q-Q Plot

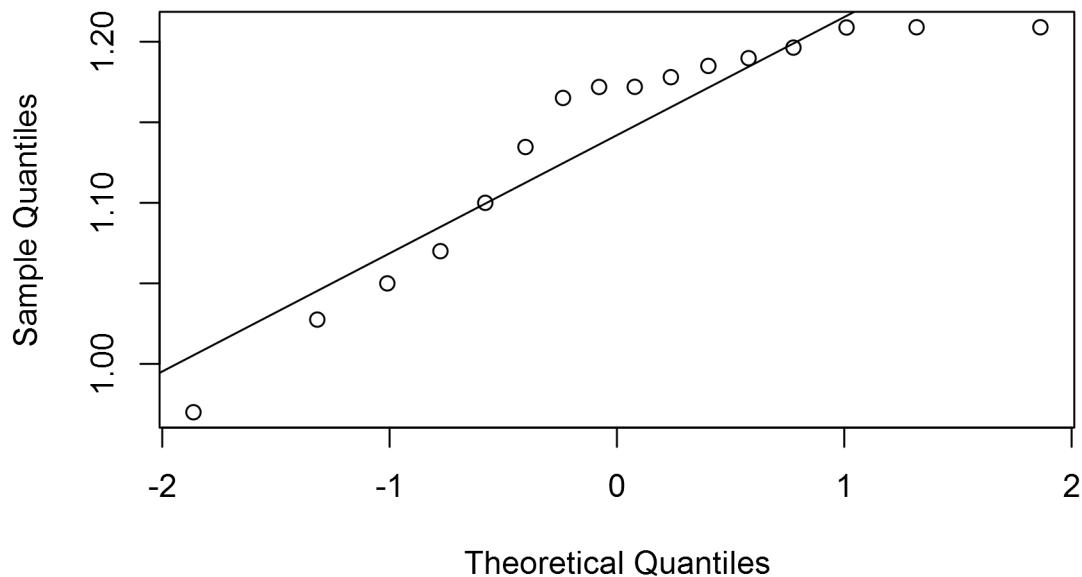


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

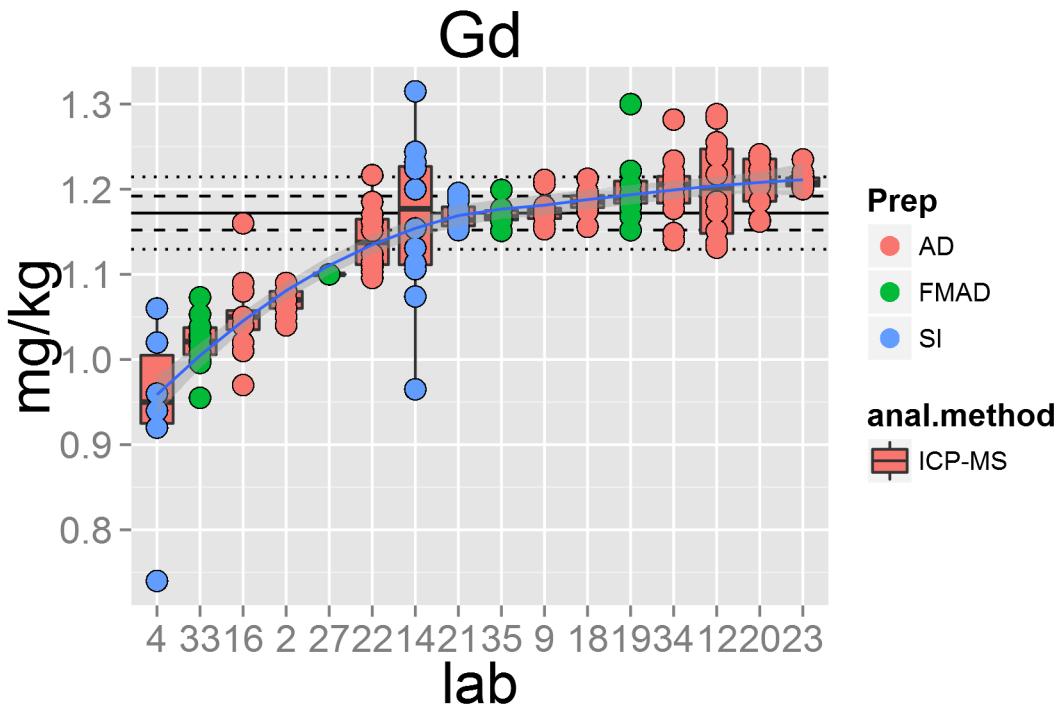


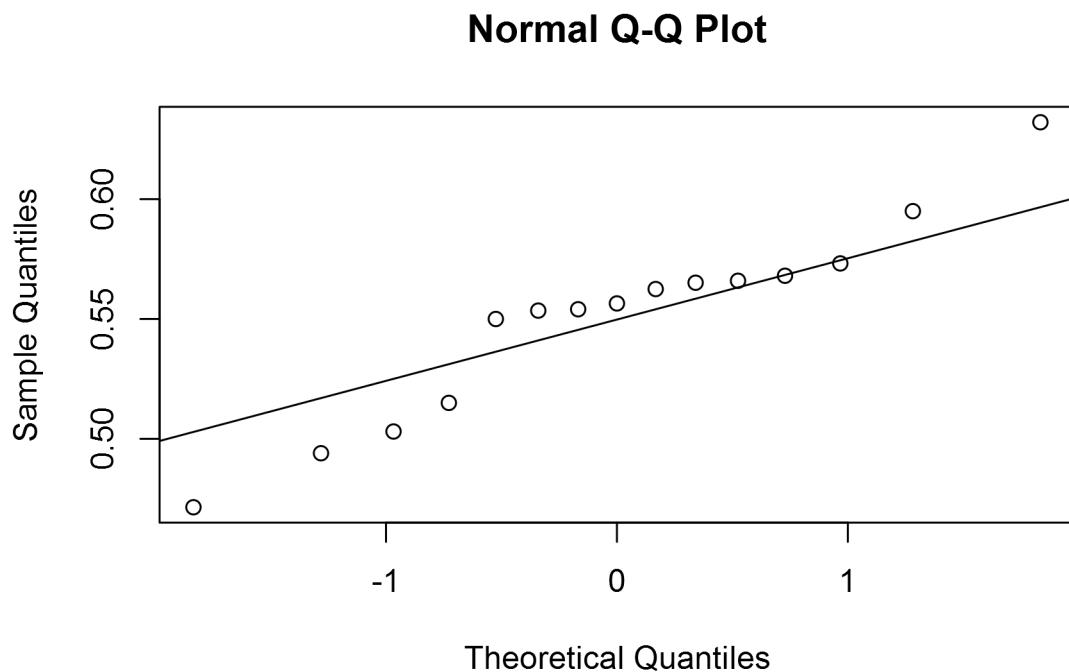
```
## [1] "Gd"
```

### Normal Q-Q Plot

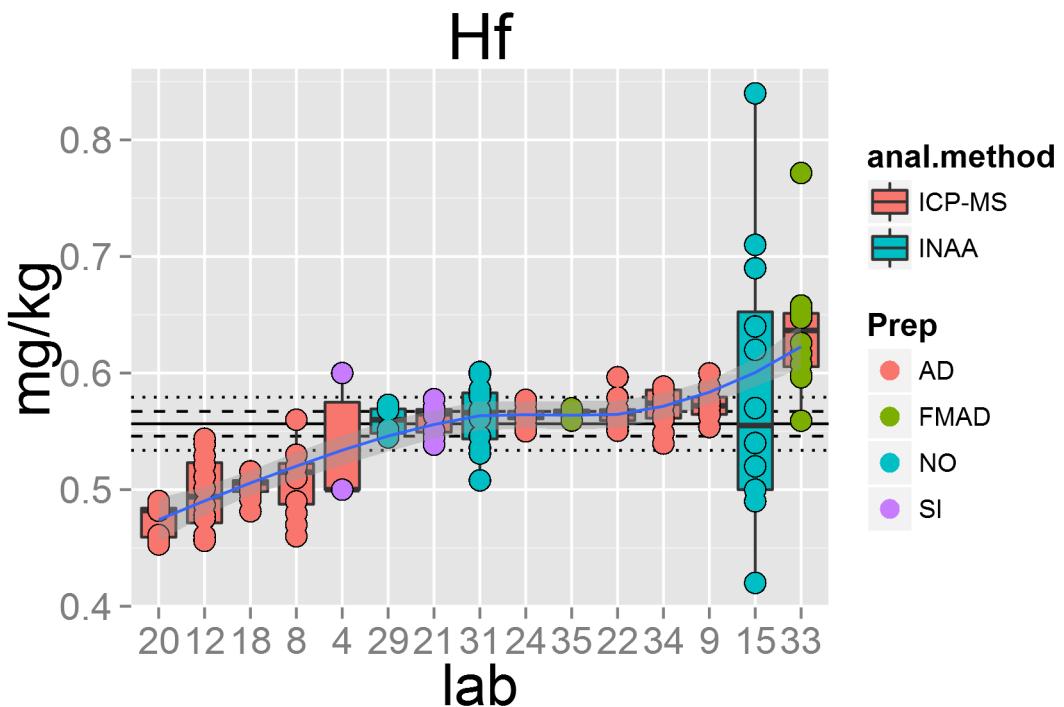


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```



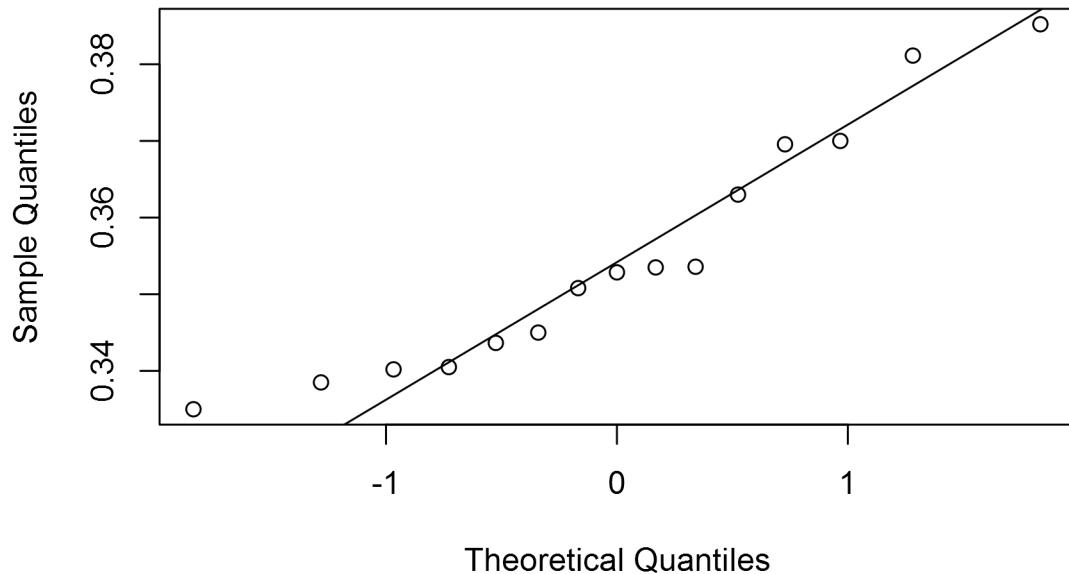


## geom\_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change.

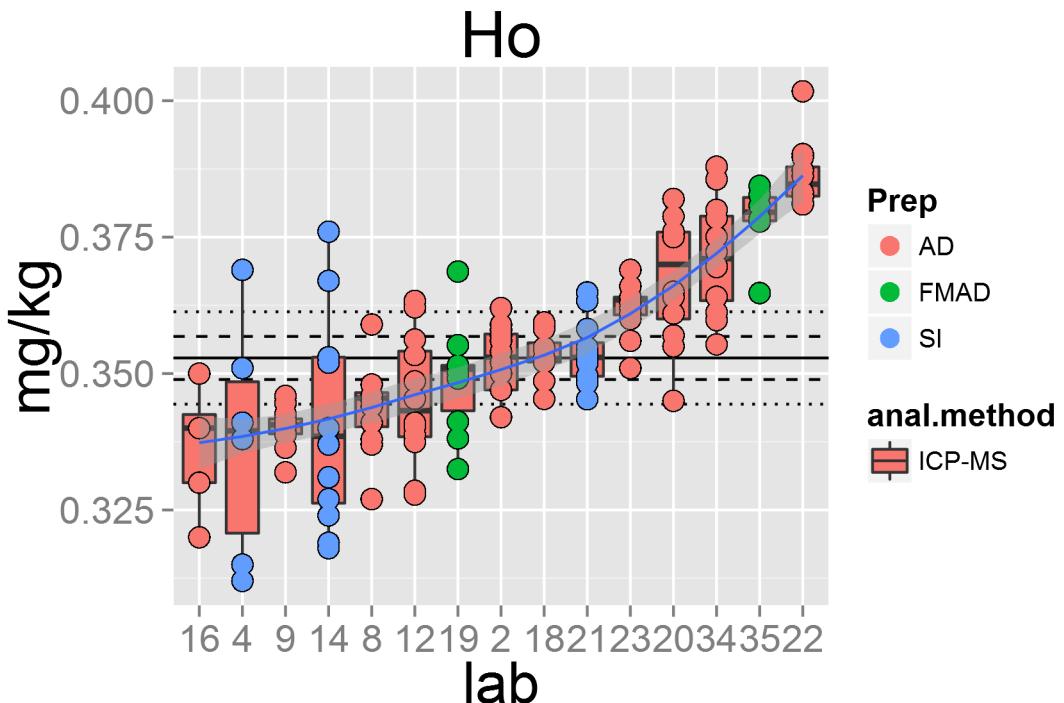


```
## [1] "Ho"
```

## Normal Q-Q Plot

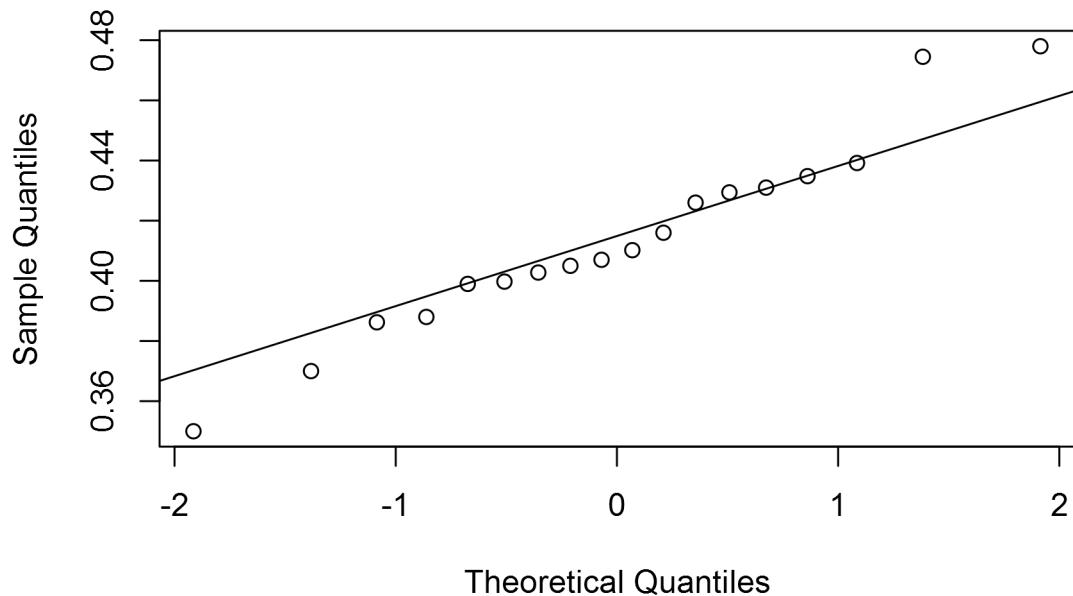


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

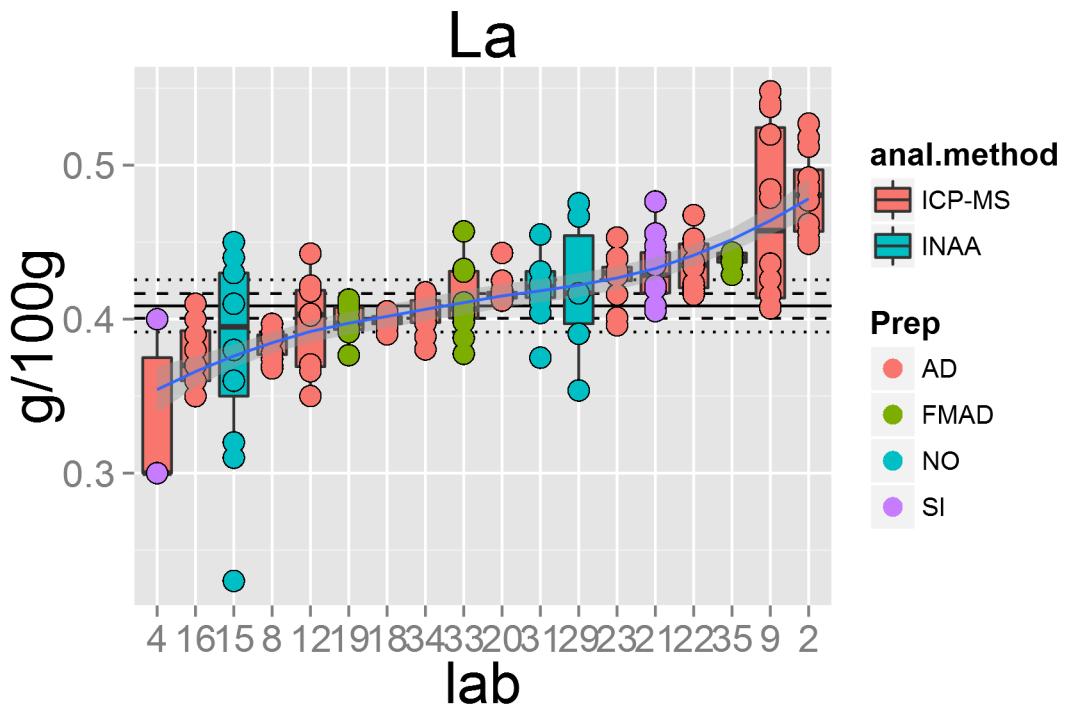


```
## [1] "La"
```

### Normal Q-Q Plot

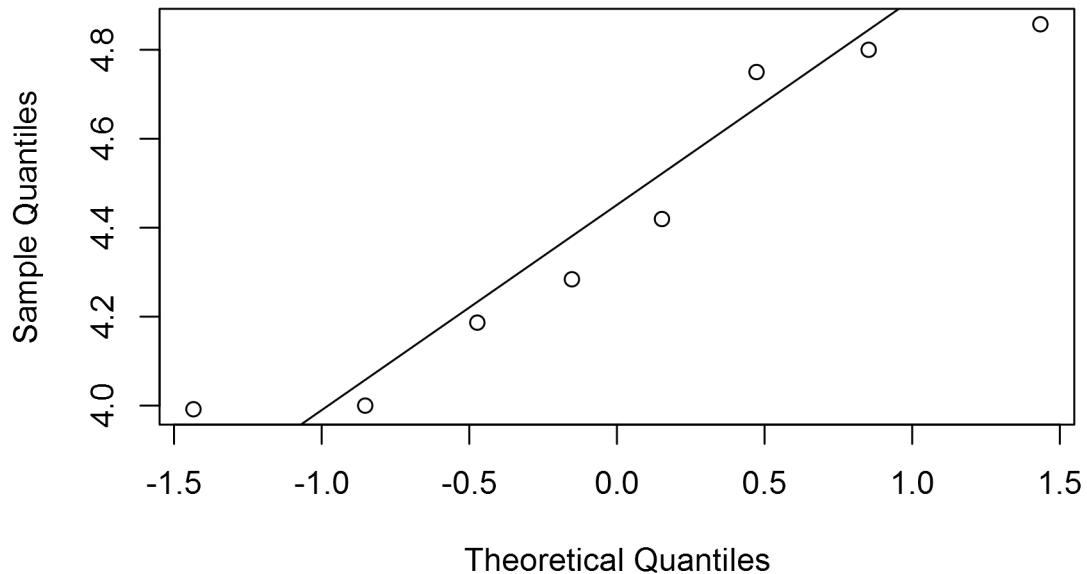


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

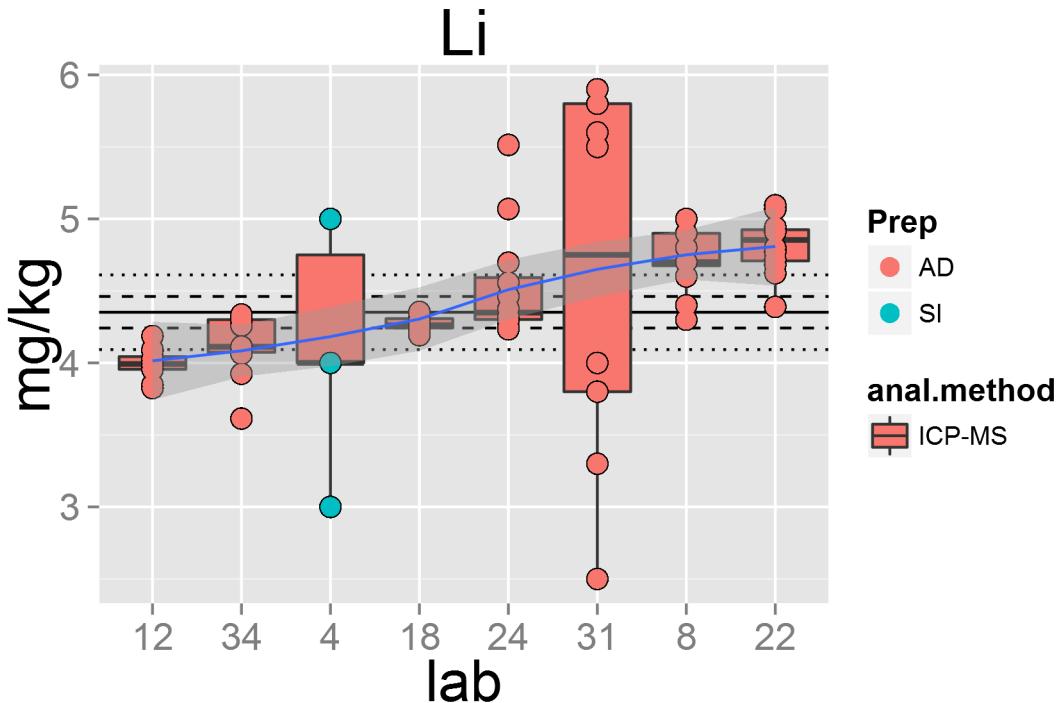


```
## [1] "Li"
```

## Normal Q-Q Plot

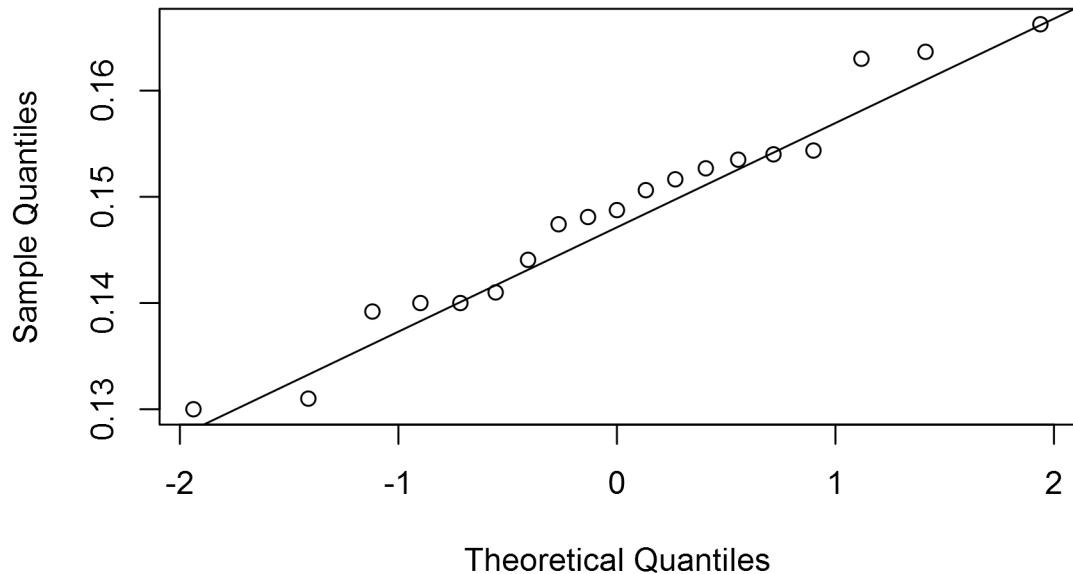


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

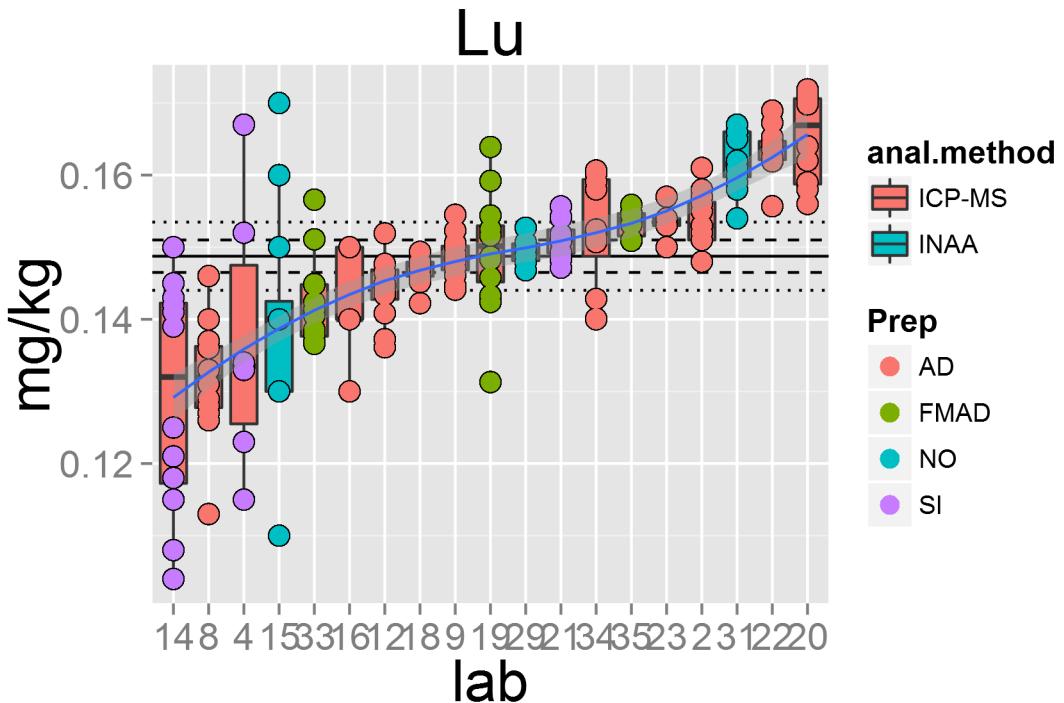


```
## [1] "Lu"
```

## Normal Q-Q Plot

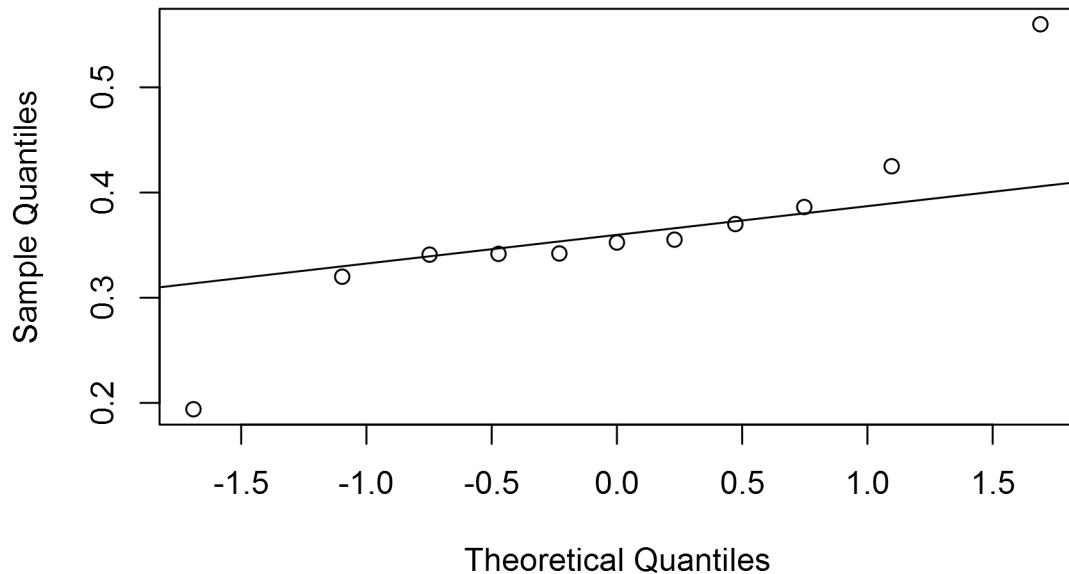


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

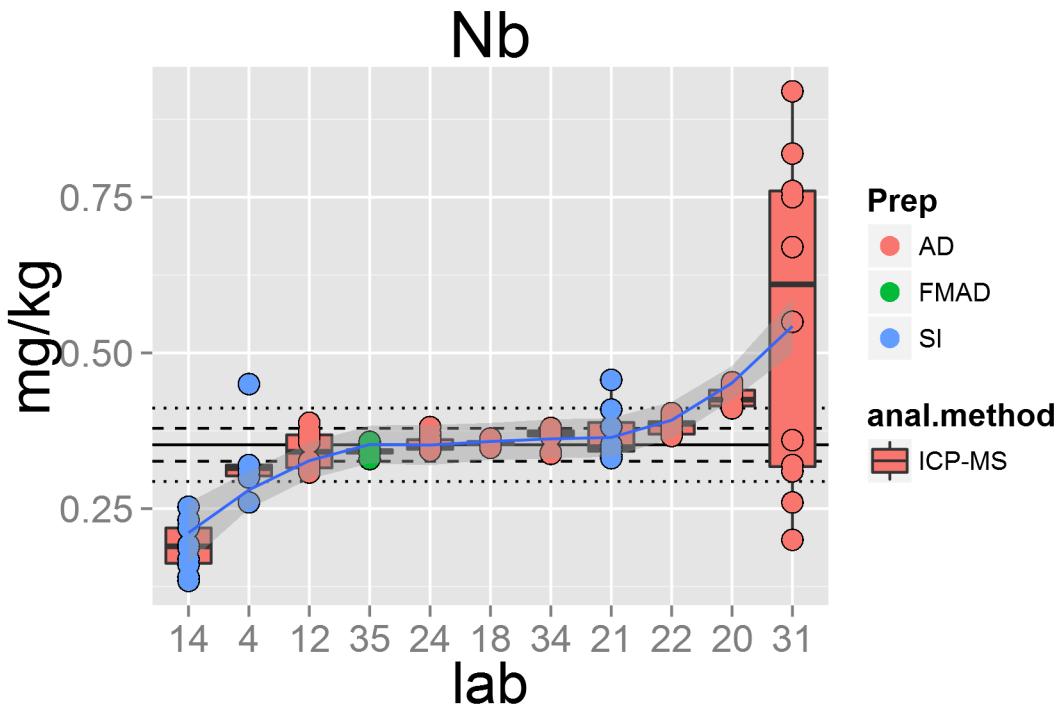


```
## [1] "Nb"
```

### Normal Q-Q Plot

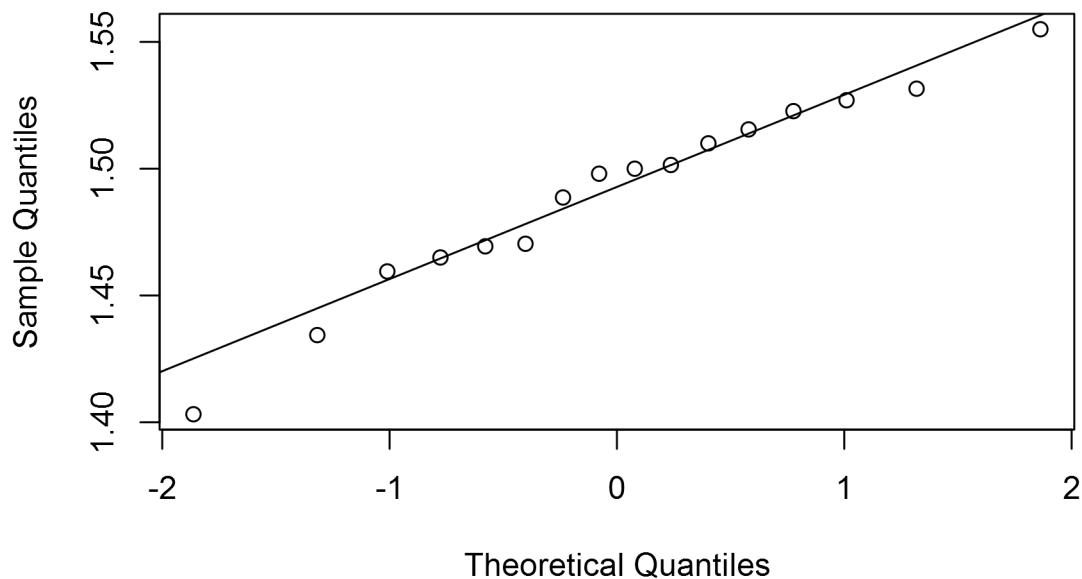


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

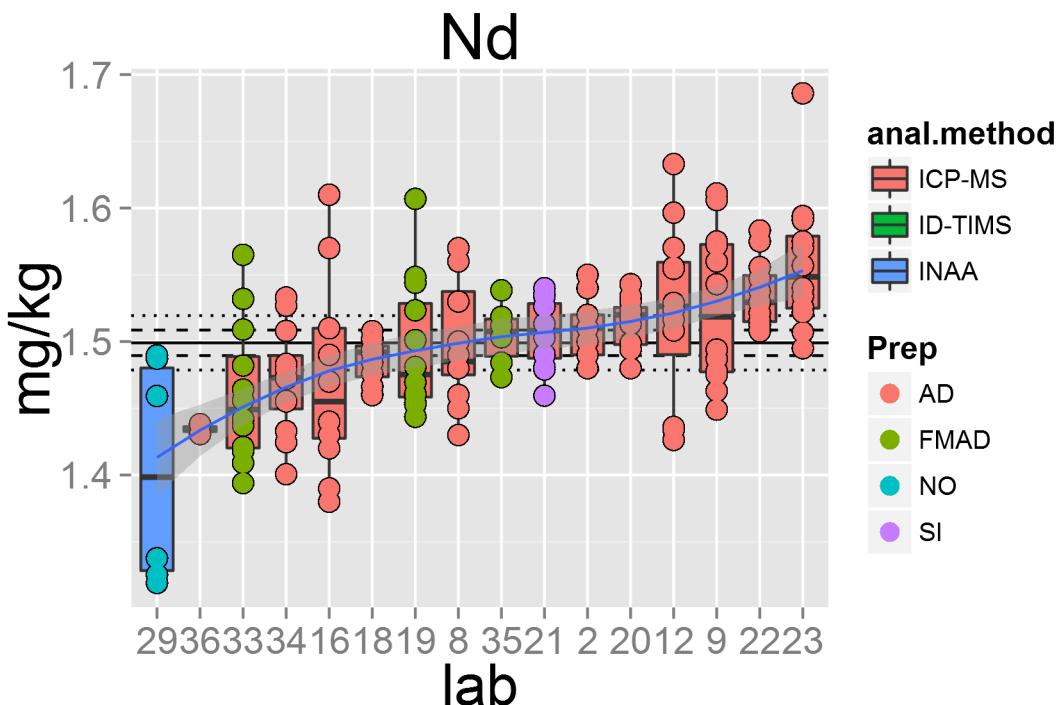


```
## [1] "Nd"
```

## Normal Q-Q Plot

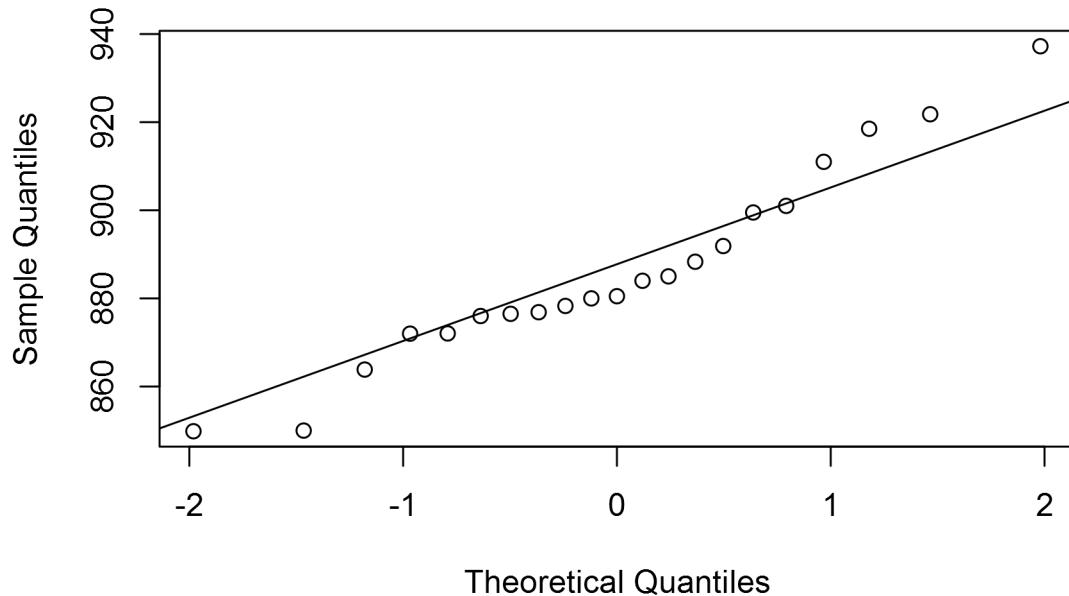


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

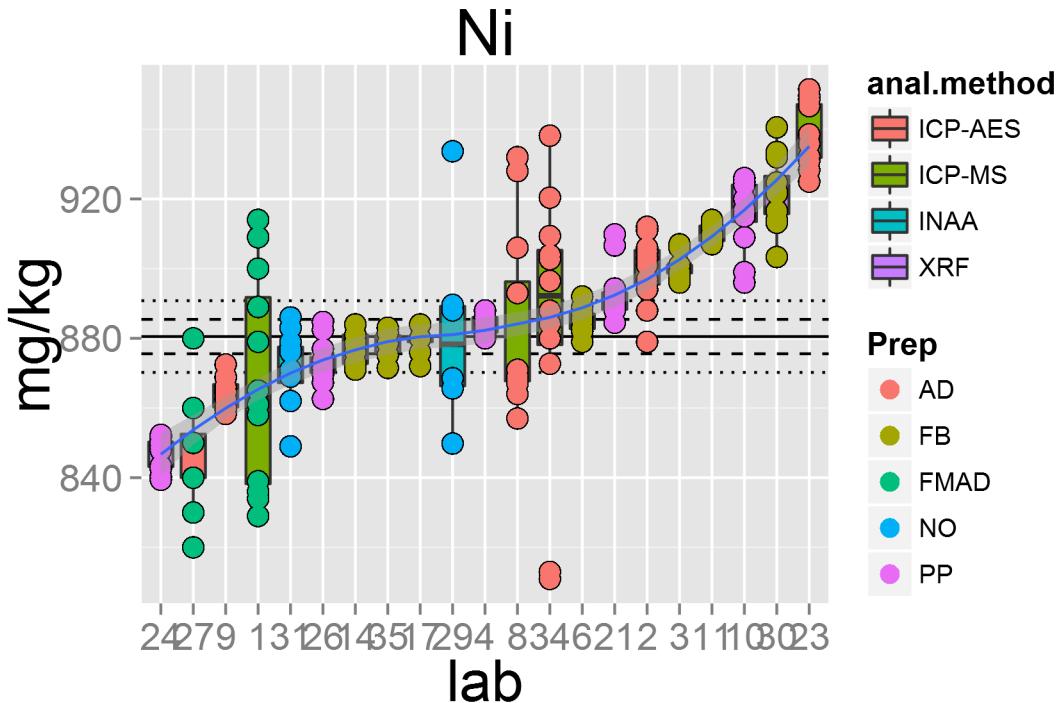


```
## [1] "Ni"
```

## Normal Q-Q Plot

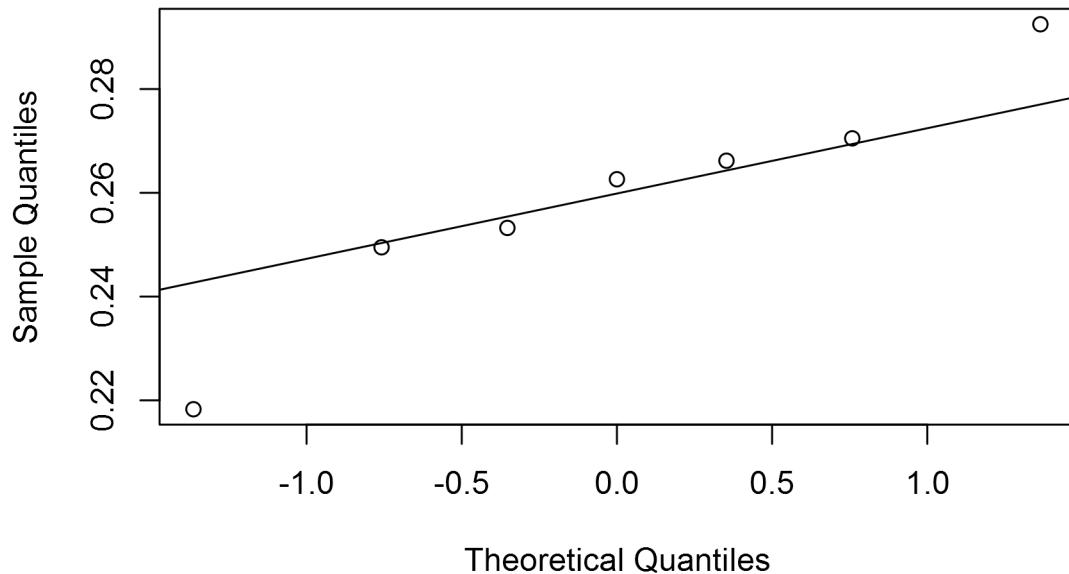


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

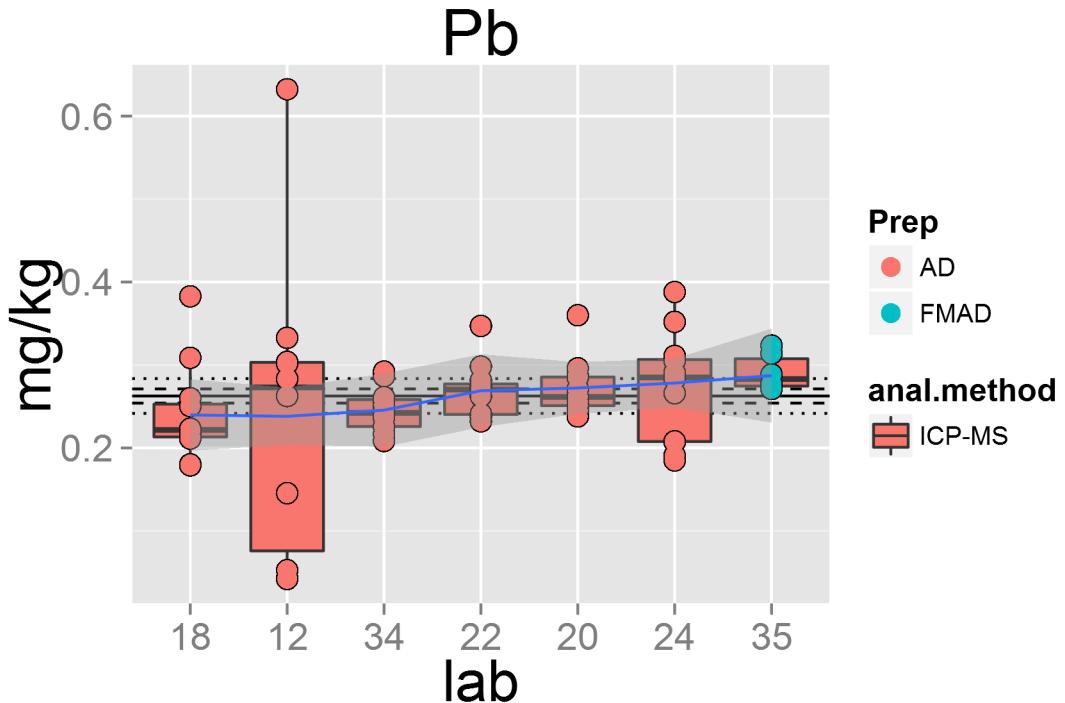


```
## [1] "Pb"
```

### Normal Q-Q Plot

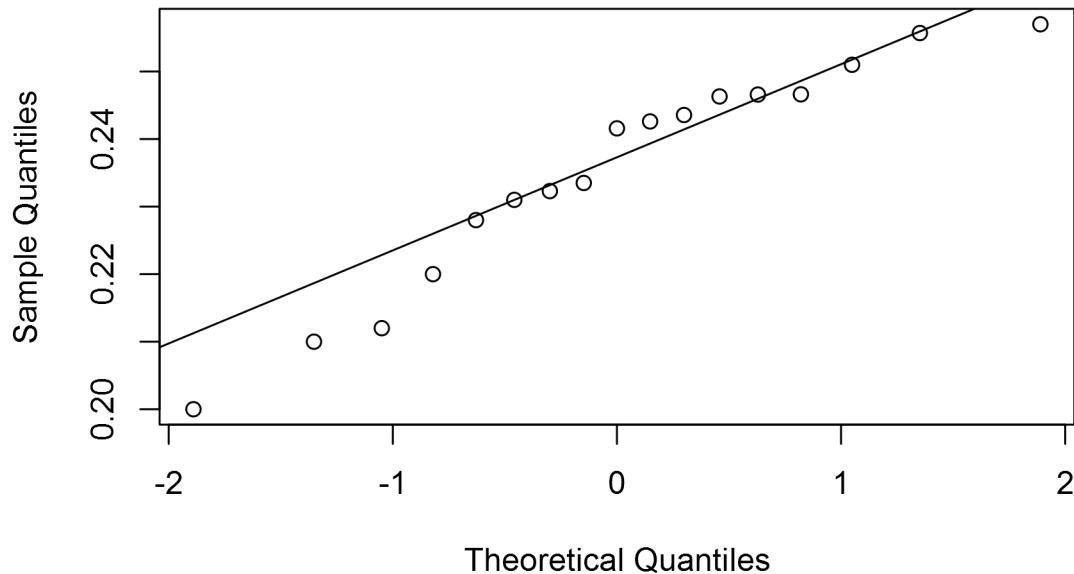


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

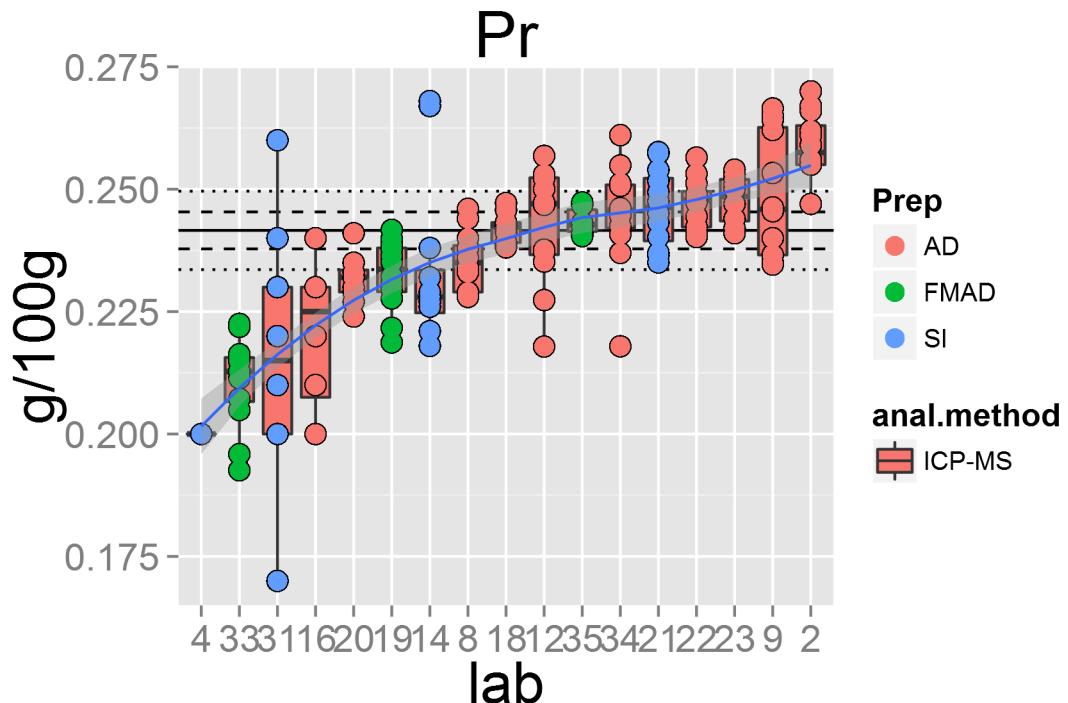


```
## [1] "Pr"
```

### Normal Q-Q Plot

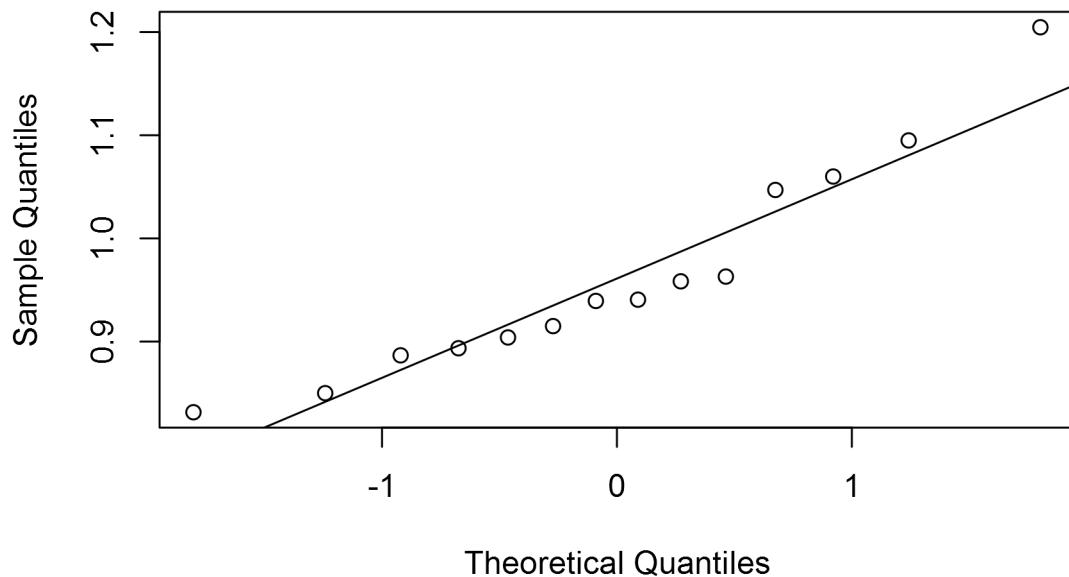


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

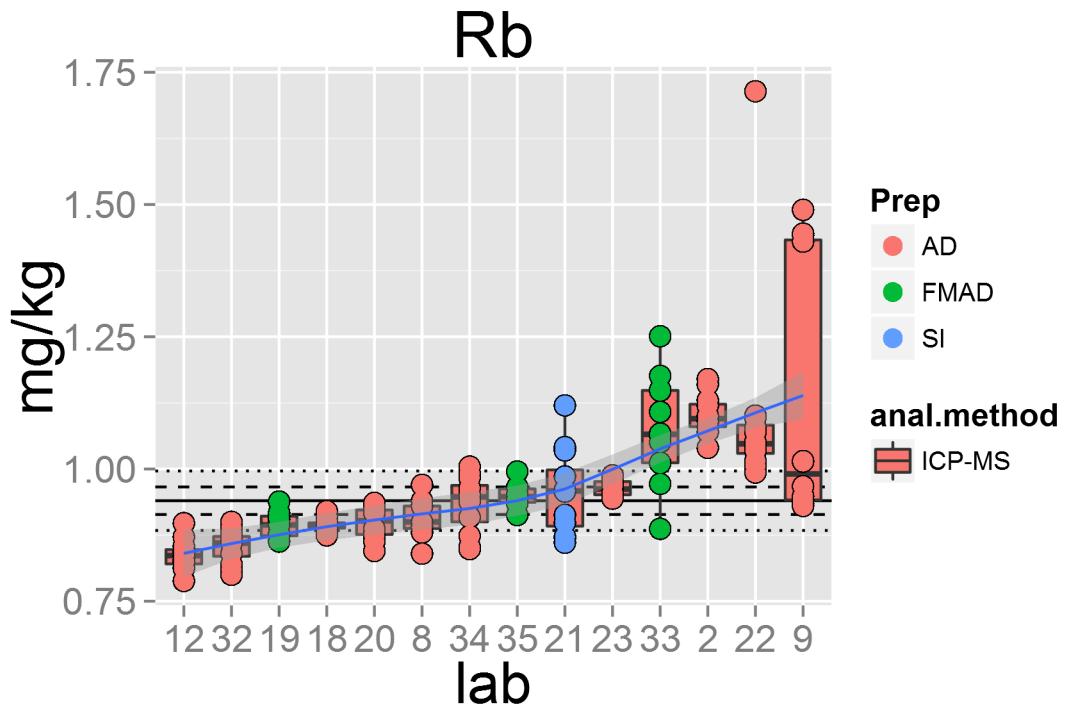


```
## [1] "Rb"
```

### Normal Q-Q Plot

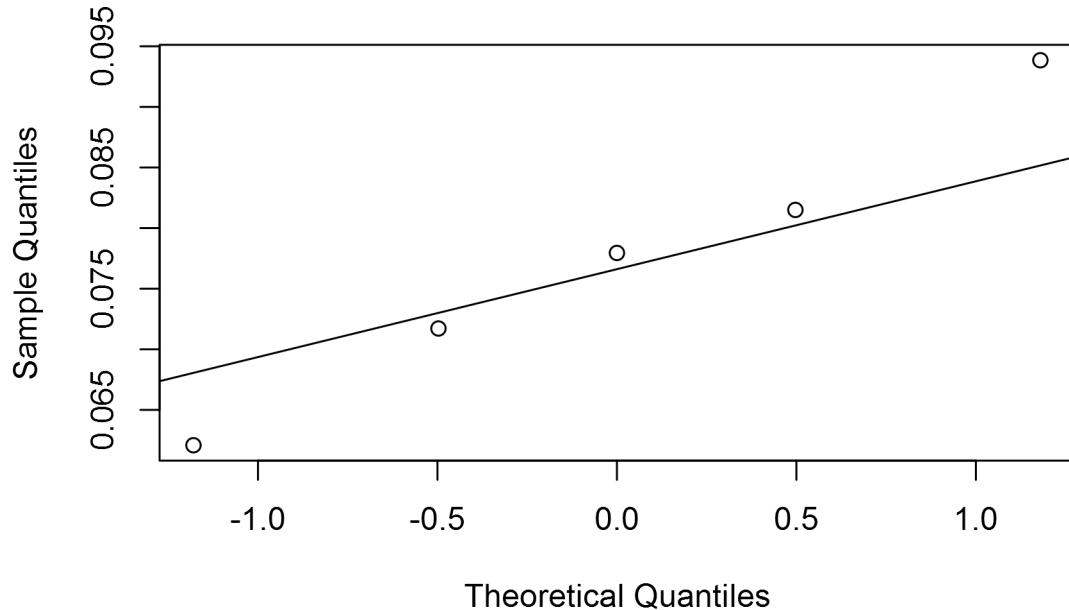


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

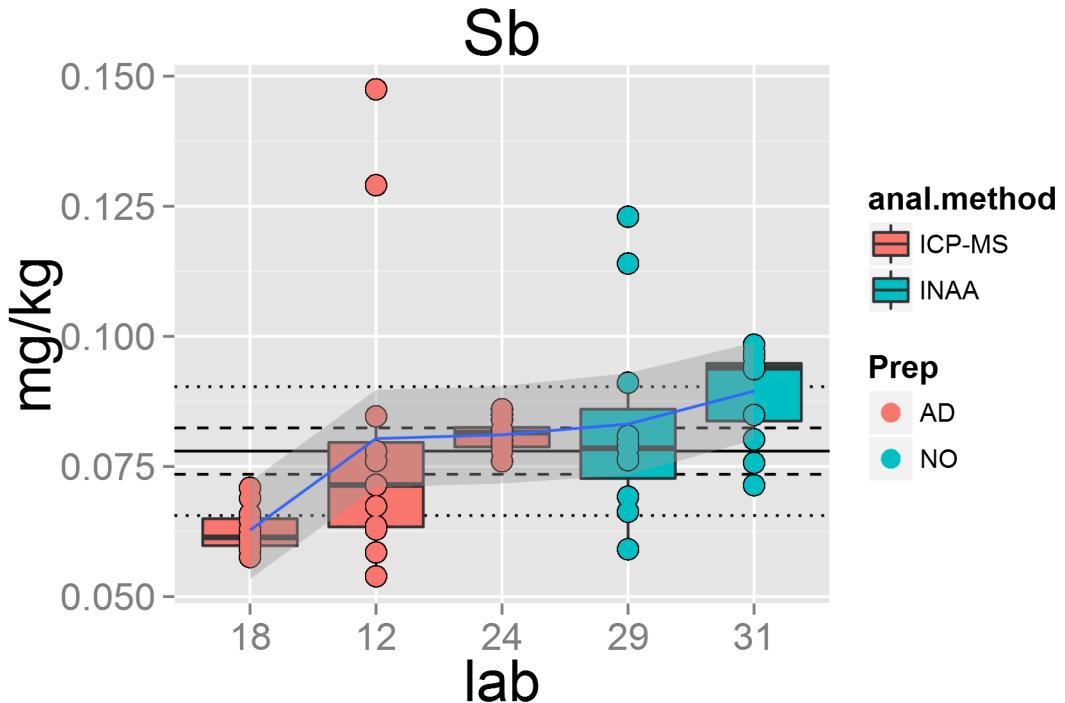


```
## [1] "Sb"
```

### Normal Q-Q Plot

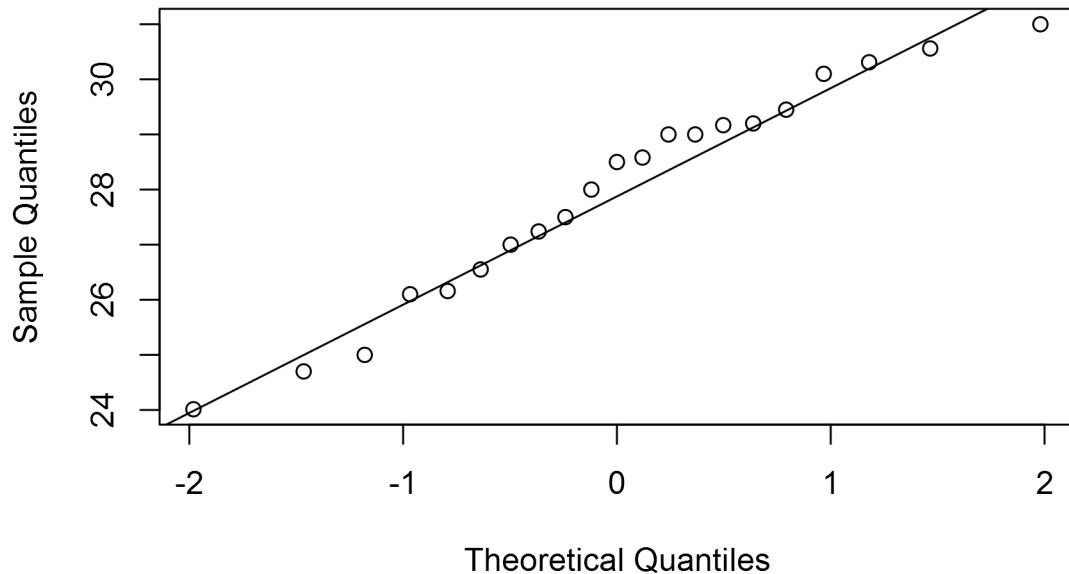


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

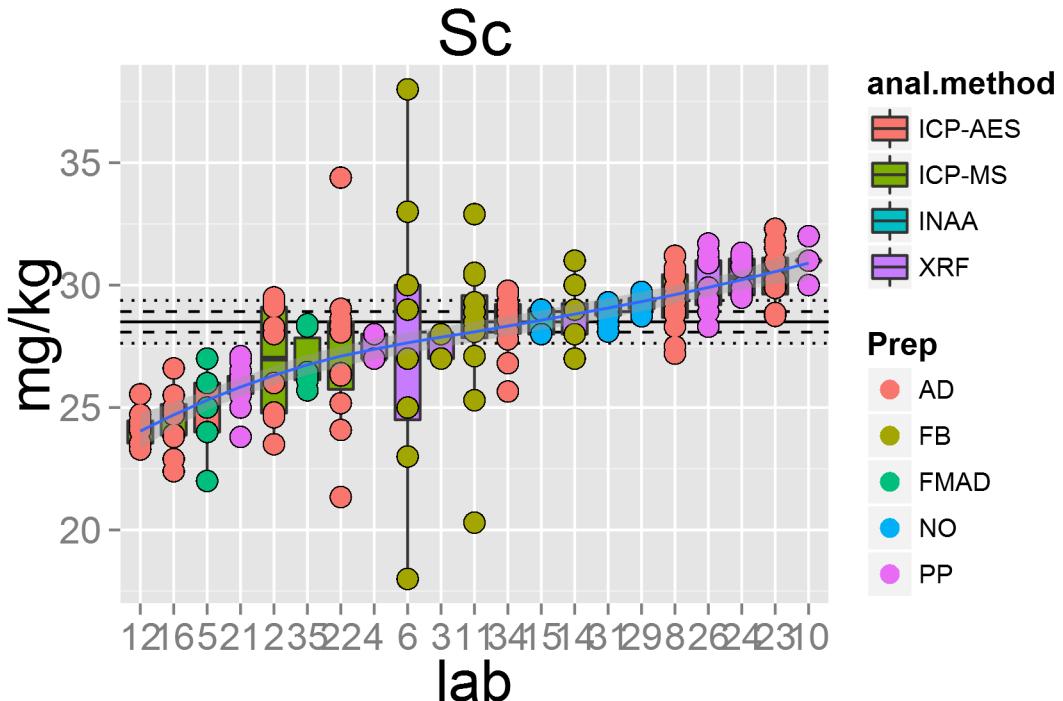


```
## [1] "Sc"
```

## Normal Q-Q Plot

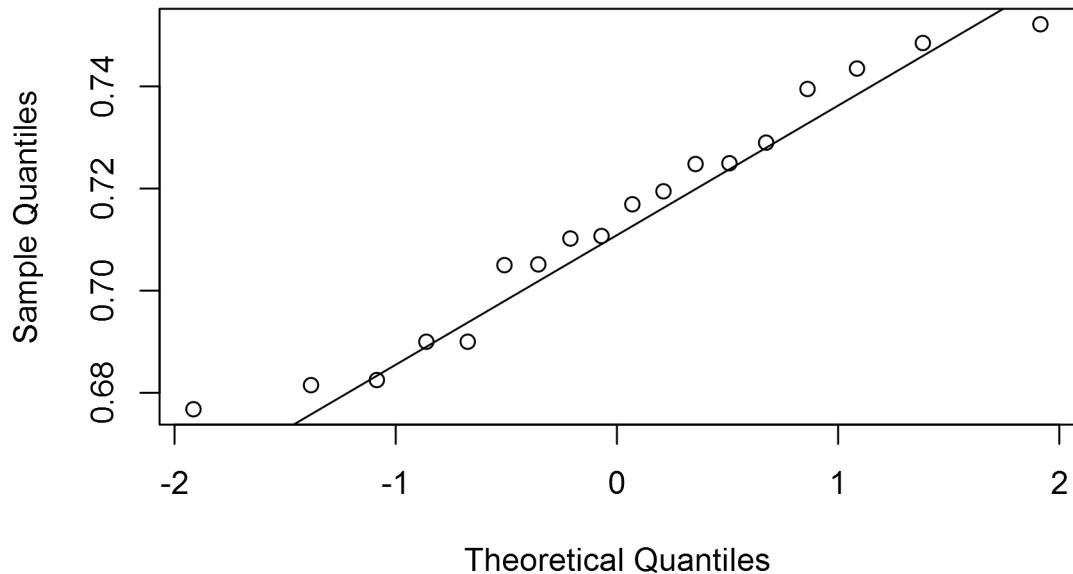


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

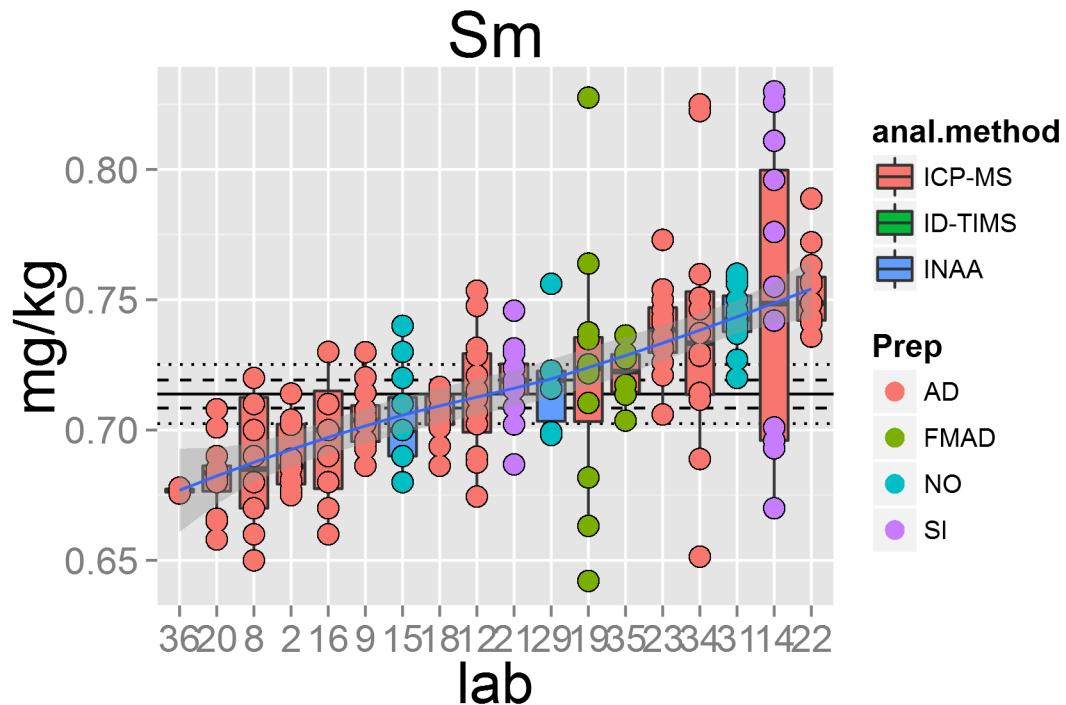


```
## [1] "Sm"
```

### Normal Q-Q Plot

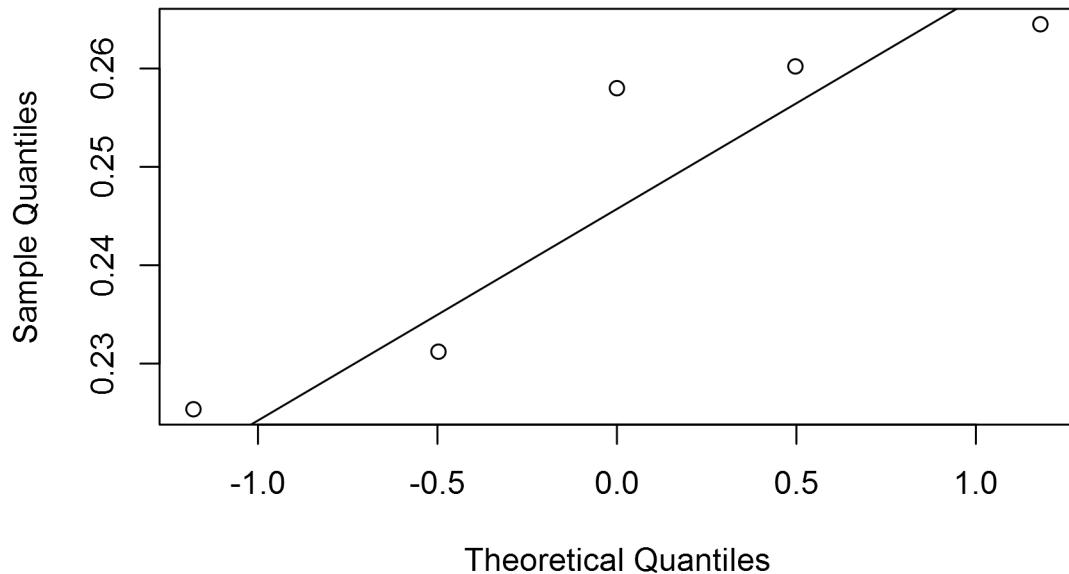


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

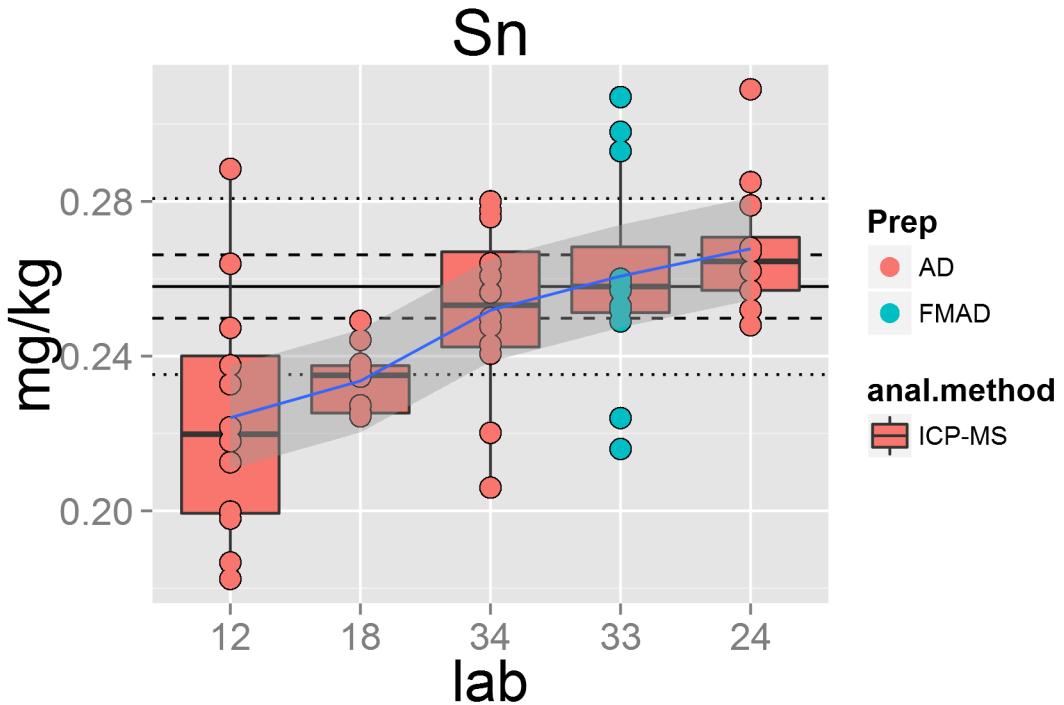


```
## [1] "Sn"
```

### Normal Q-Q Plot

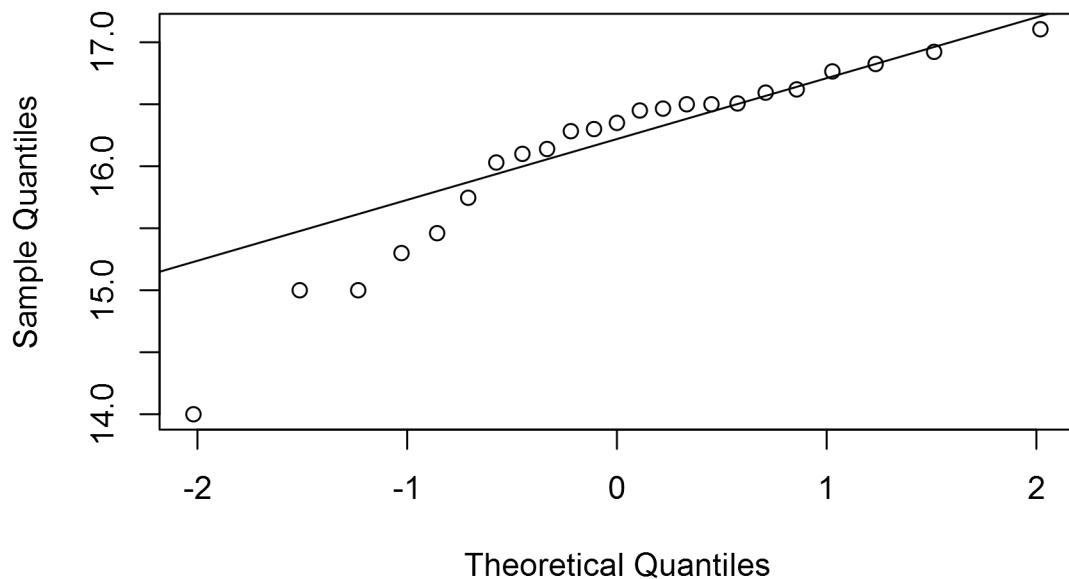


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
```

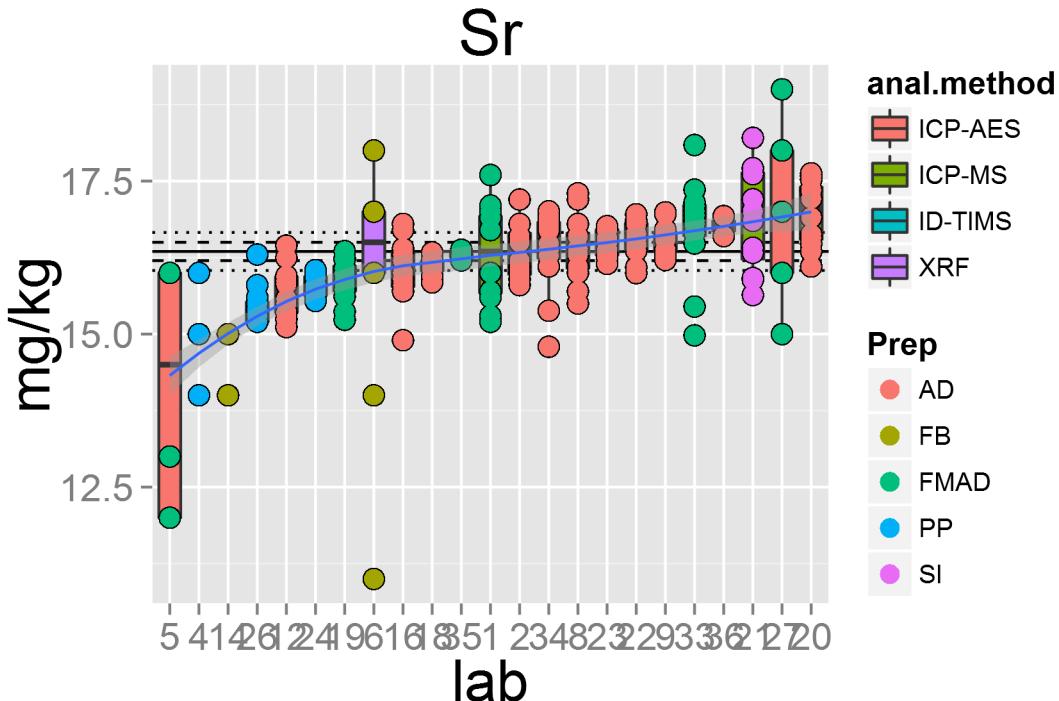


```
## [1] "Sr"
```

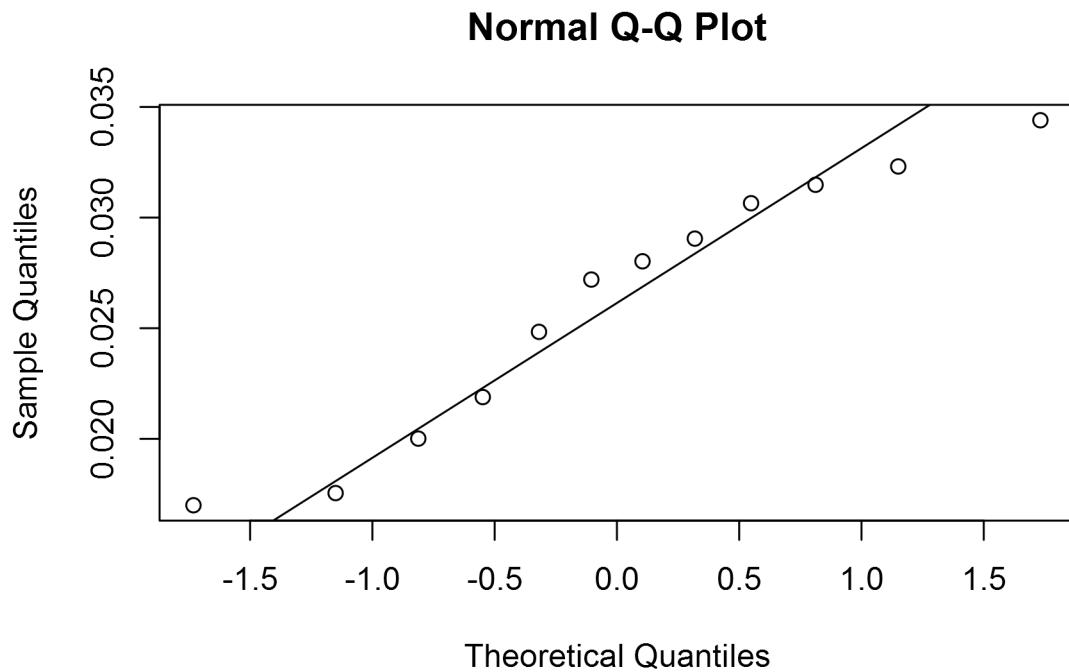
### Normal Q-Q Plot



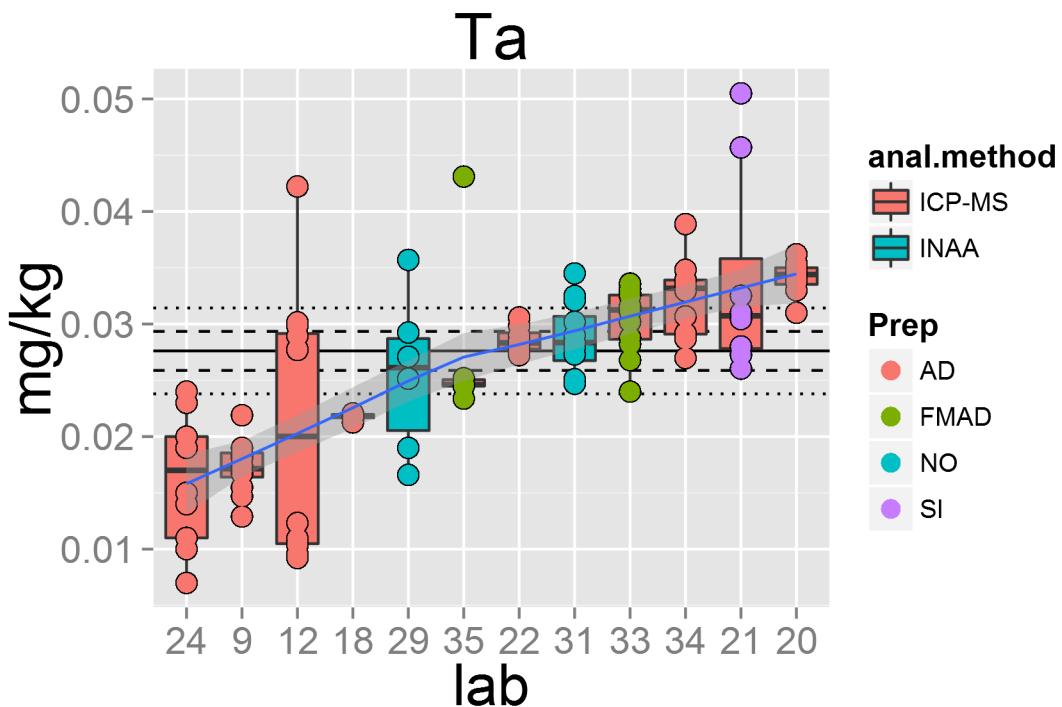
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



```
## [1] "Ta"
```

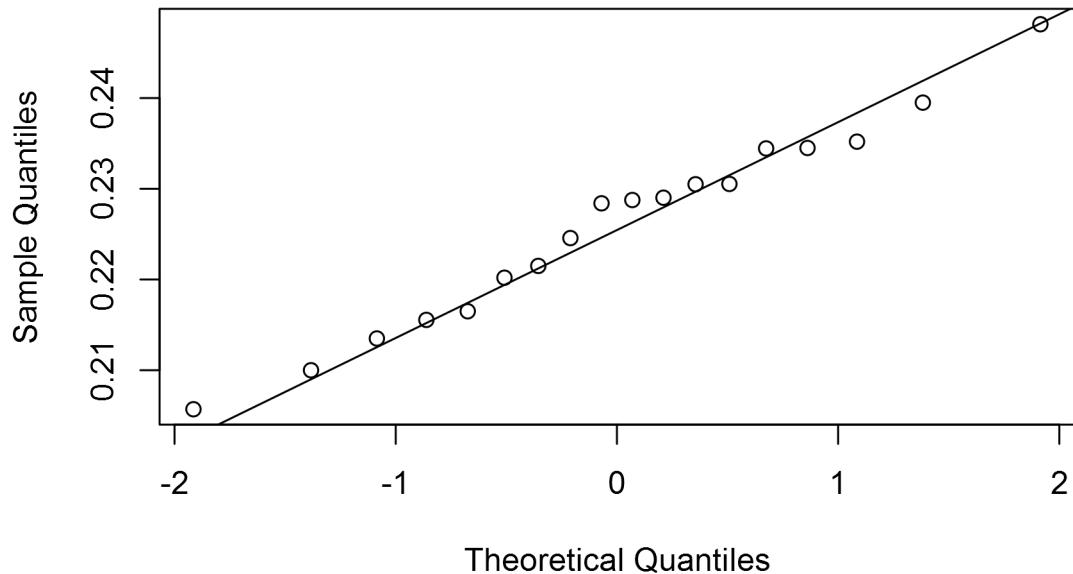


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

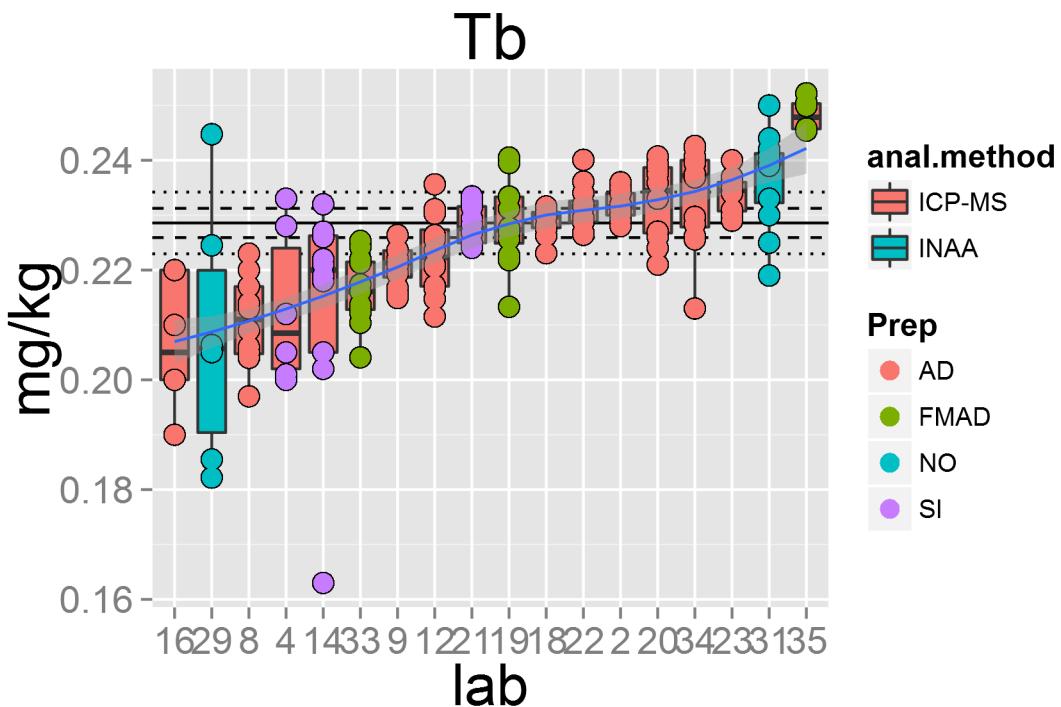


```
## [1] "Tb"
```

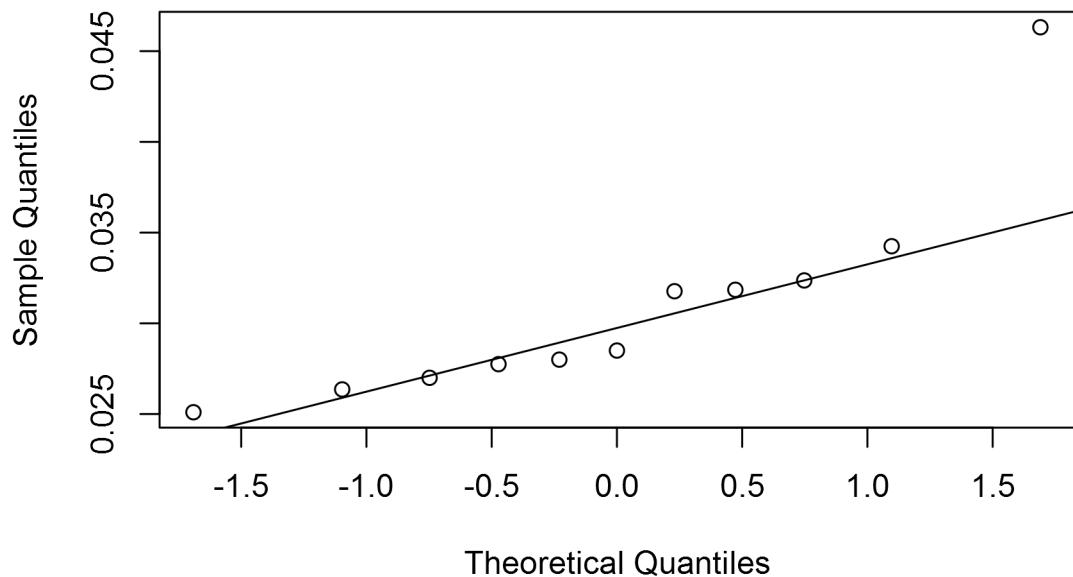
## Normal Q-Q Plot



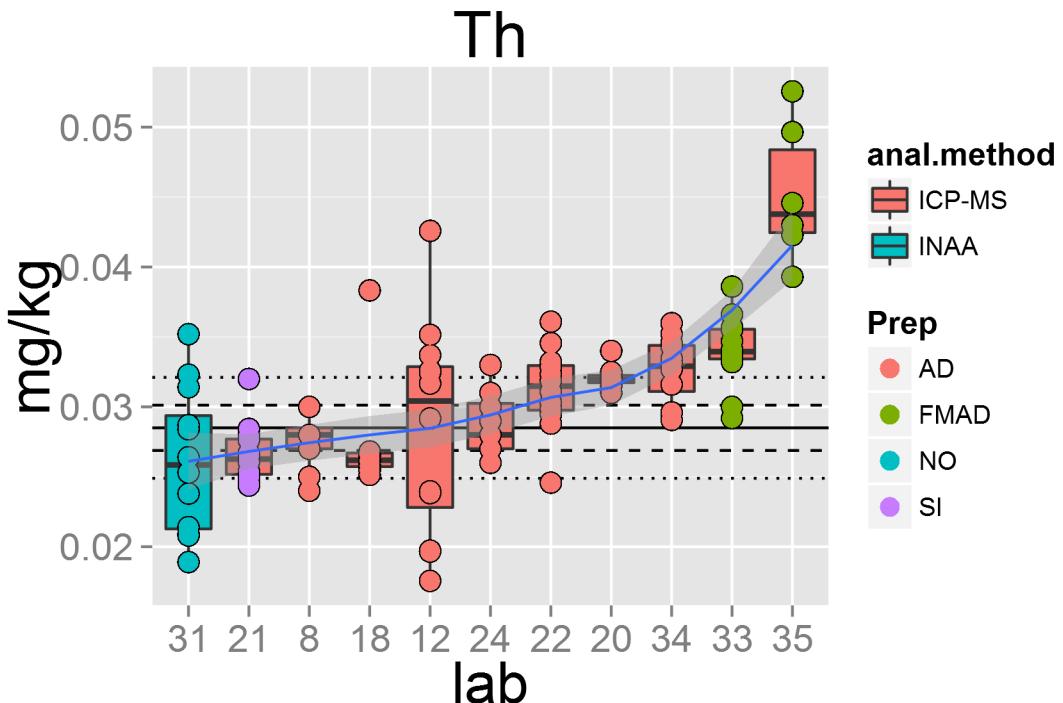
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



### Normal Q-Q Plot

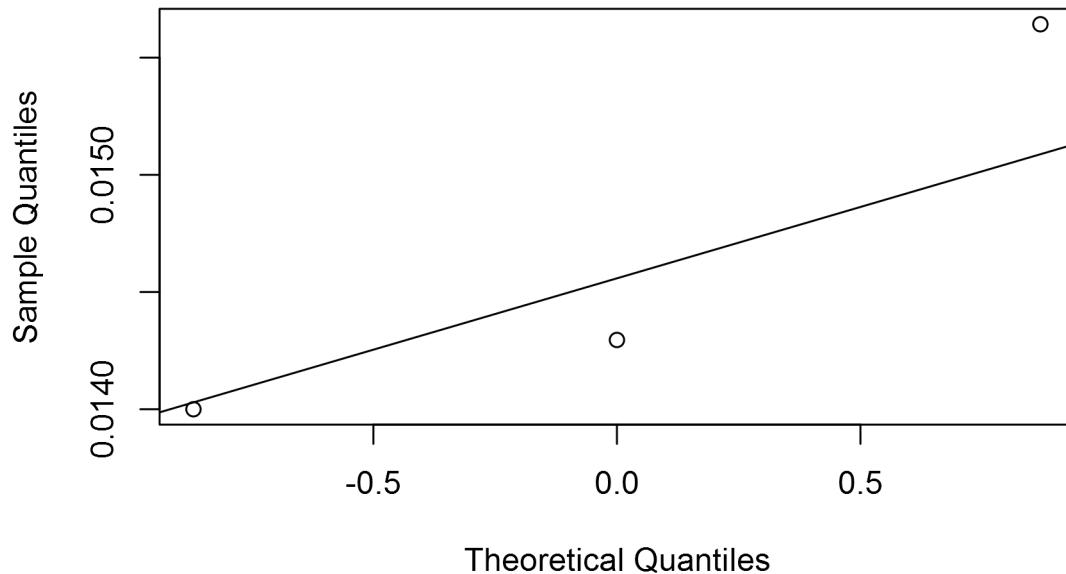


```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```

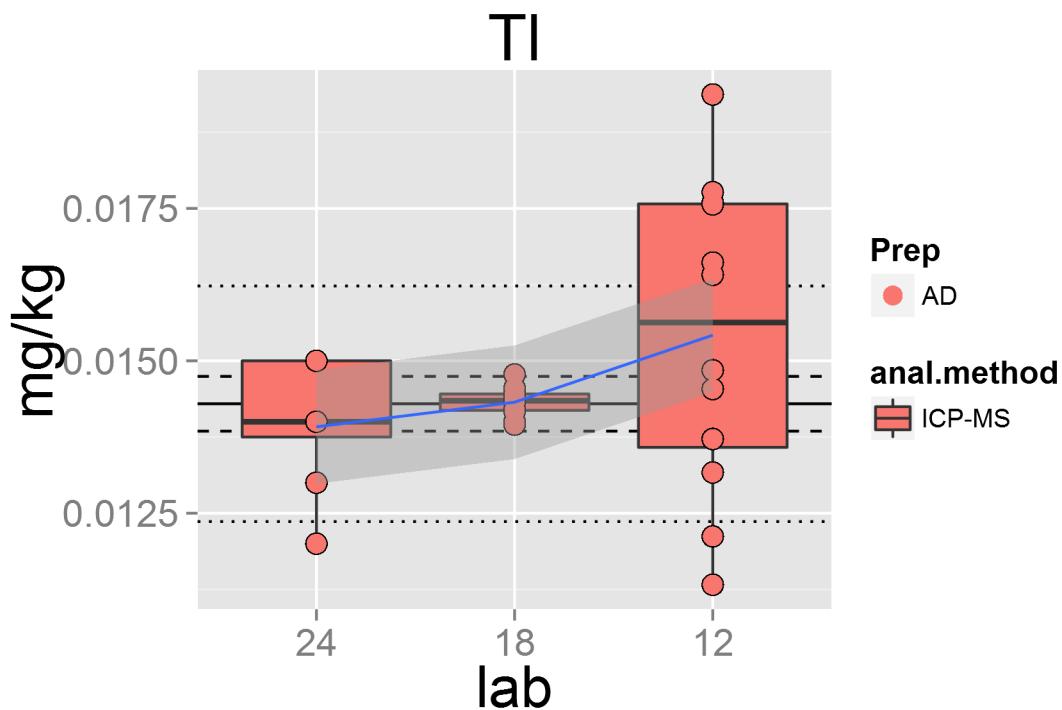


```
## [1] "T1"
```

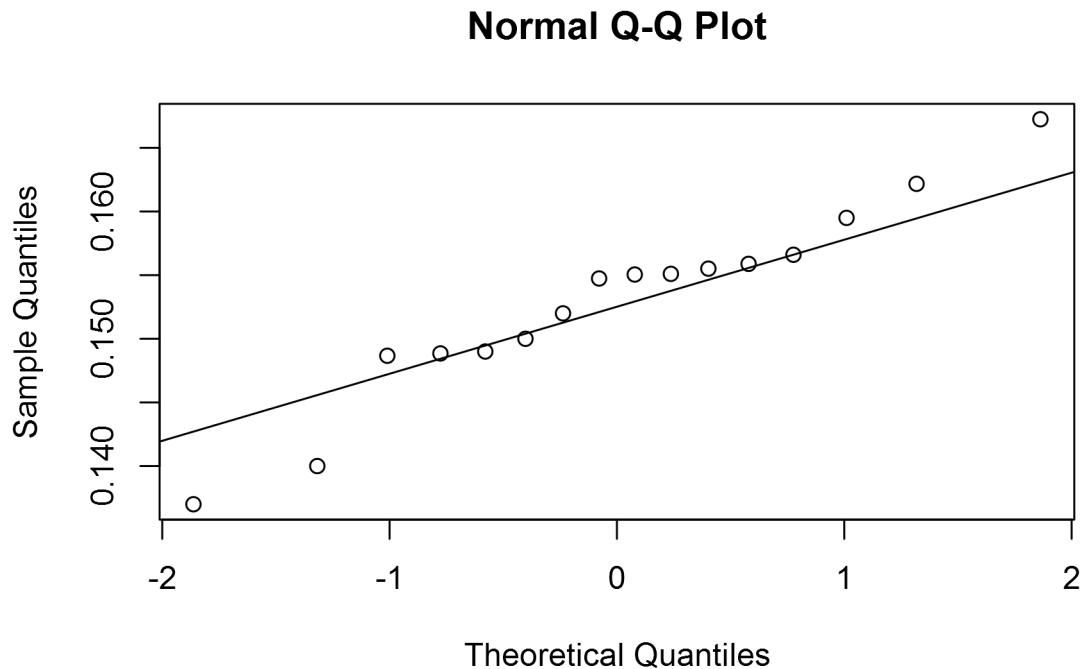
## Normal Q-Q Plot



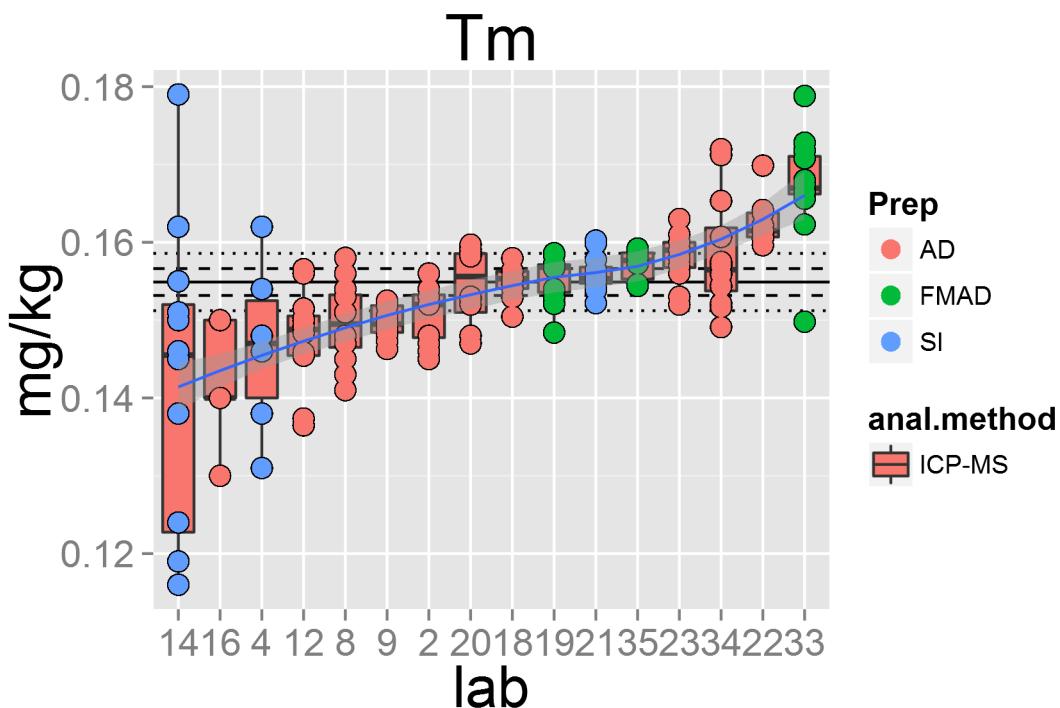
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to c
## Warning: pseudoinverse used at 0.99
## Warning: neighborhood radius 2.01
## Warning: reciprocal condition number  2.581e-016
## Warning: There are other near singularities as well. 4.0401
## Warning: pseudoinverse used at 0.99
## Warning: neighborhood radius 2.01
## Warning: reciprocal condition number  2.581e-016
## Warning: There are other near singularities as well. 4.0401
```



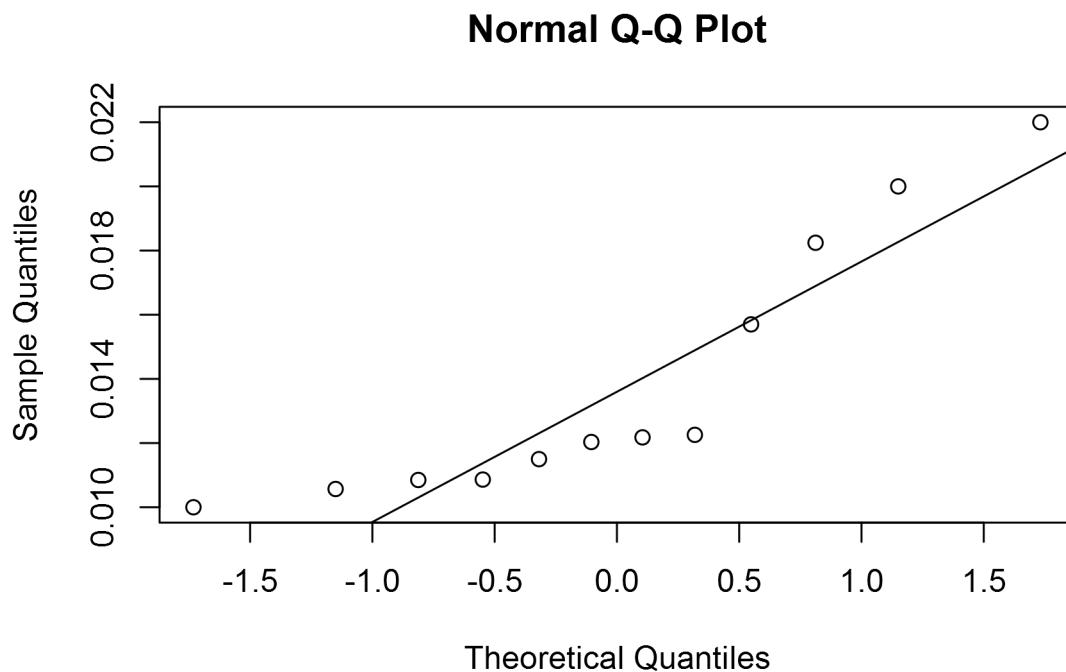
```
## [1] "Tm"
```



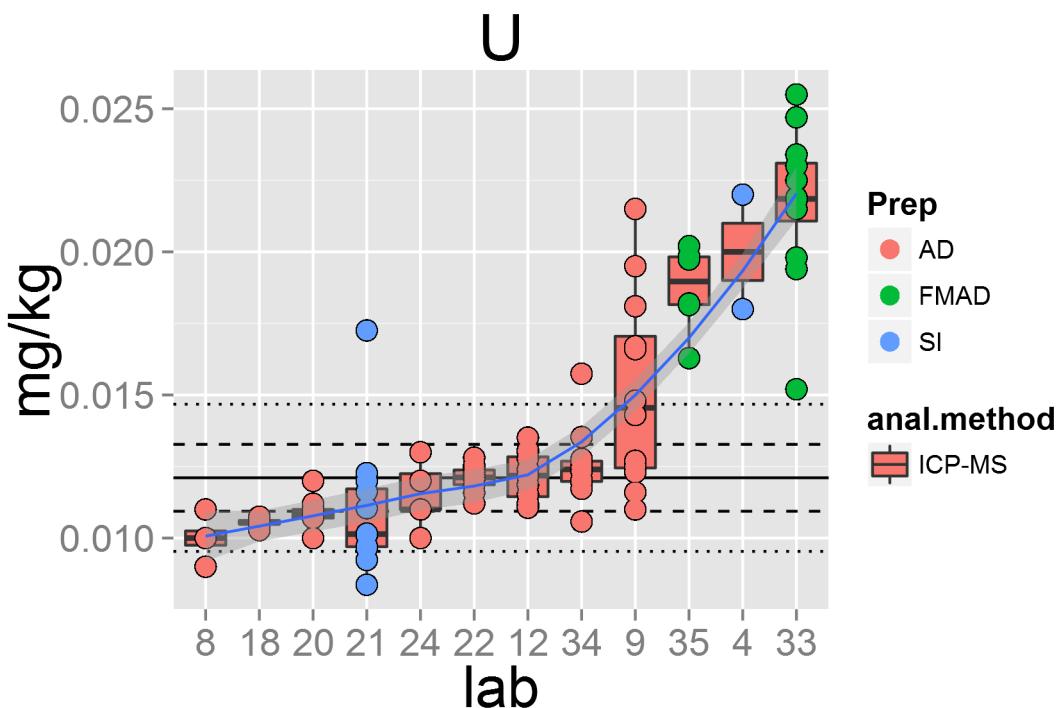
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



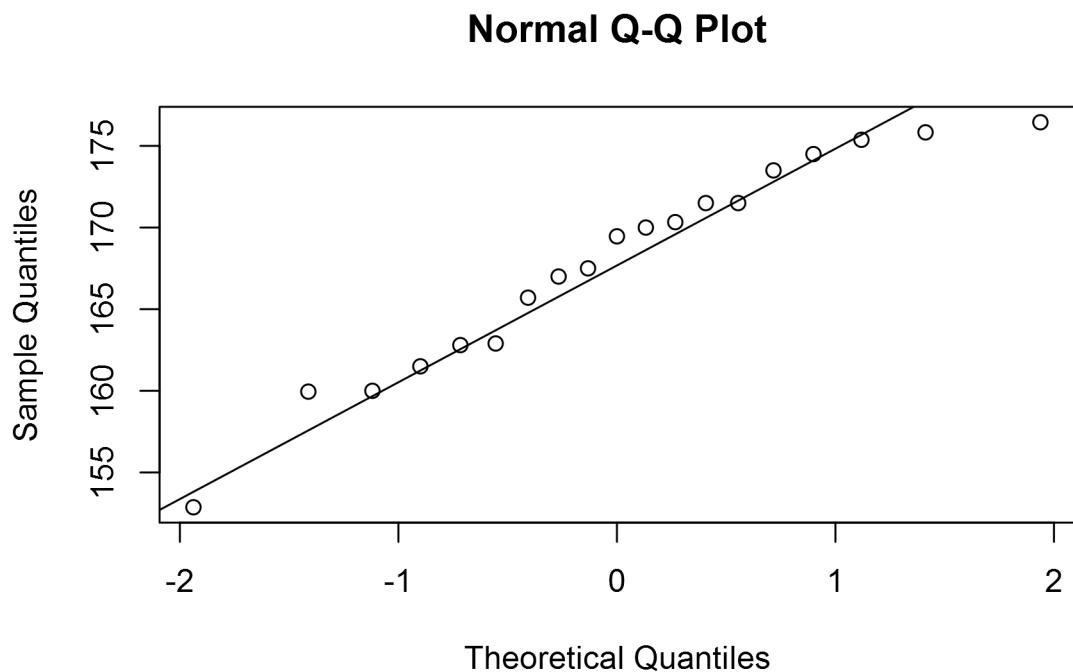
```
## [1] "U"
```



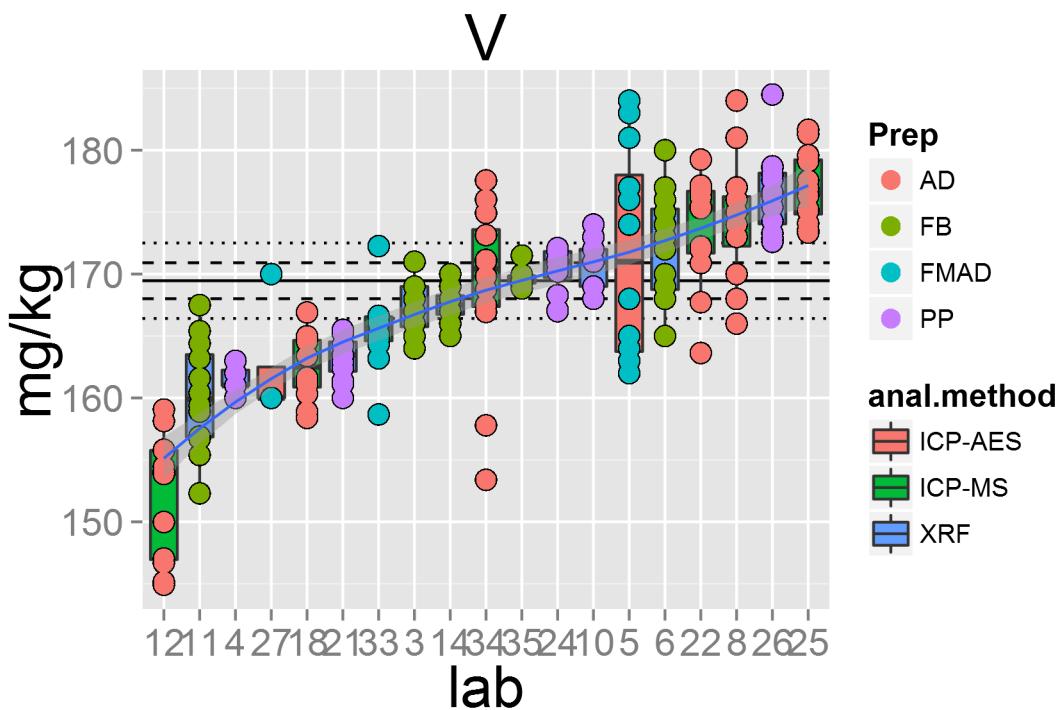
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



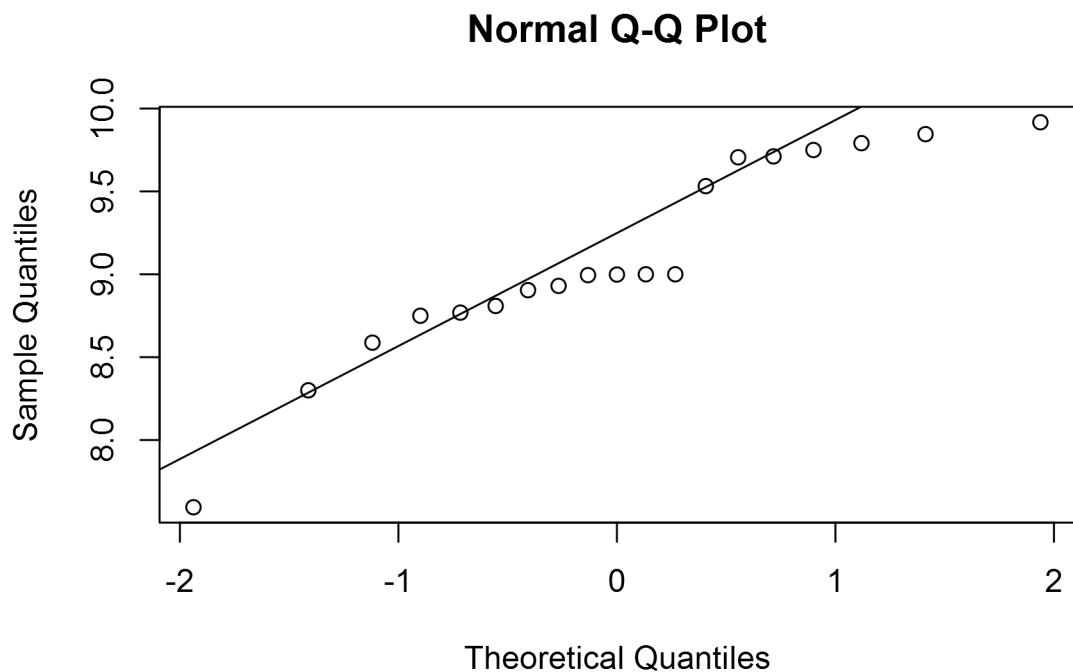
```
## [1] "V"
```



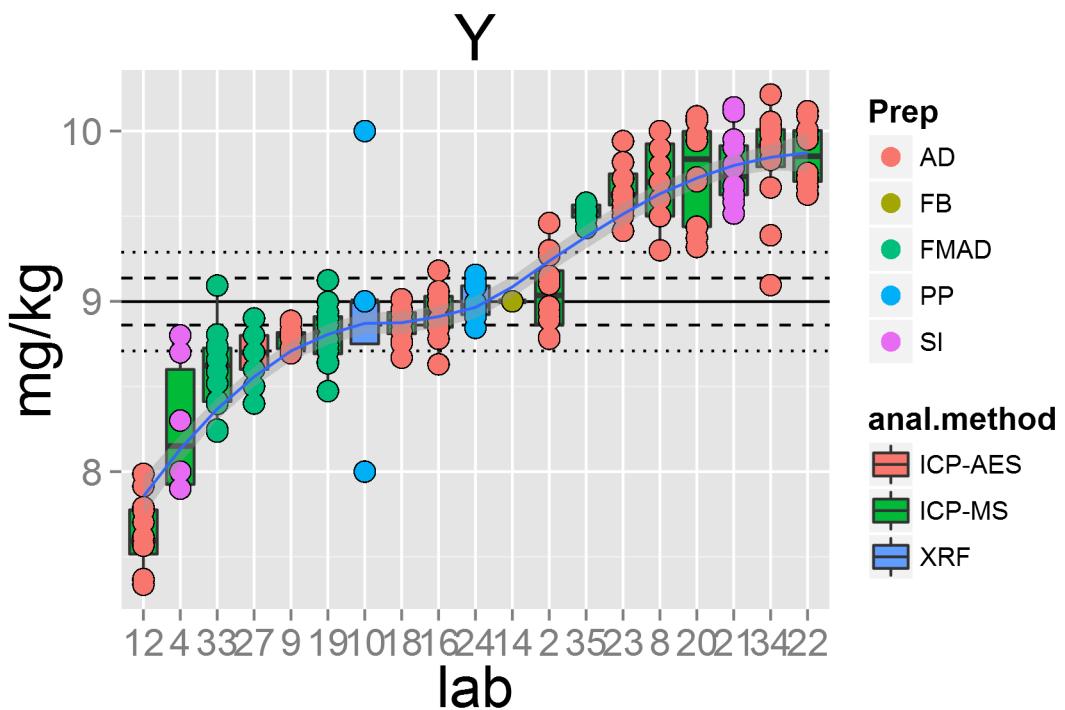
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



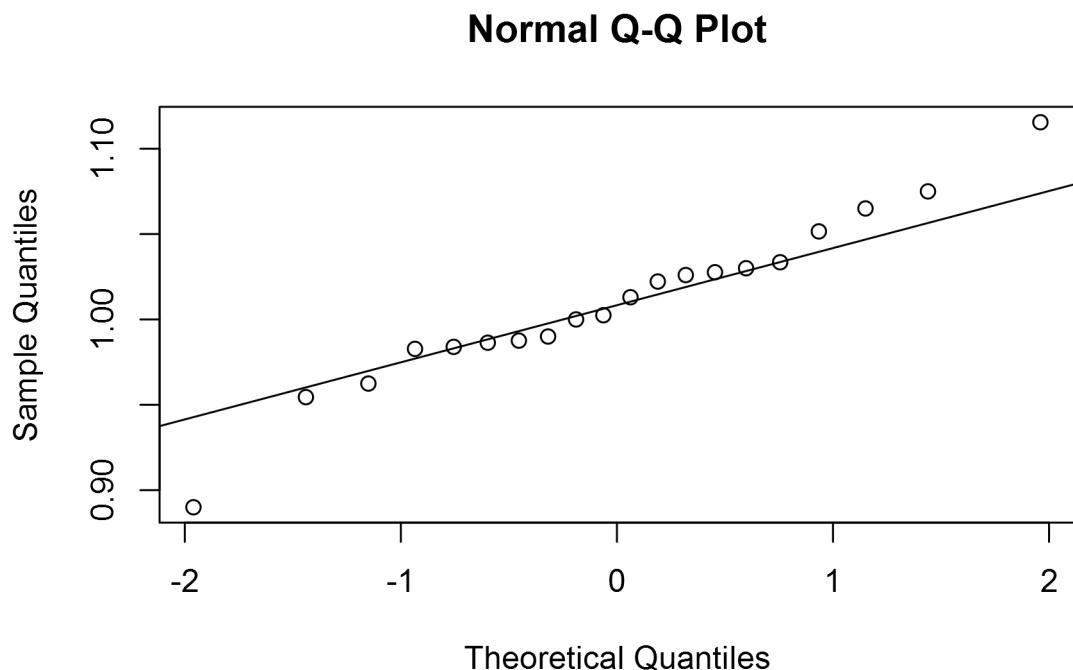
```
## [1] "Y"
```



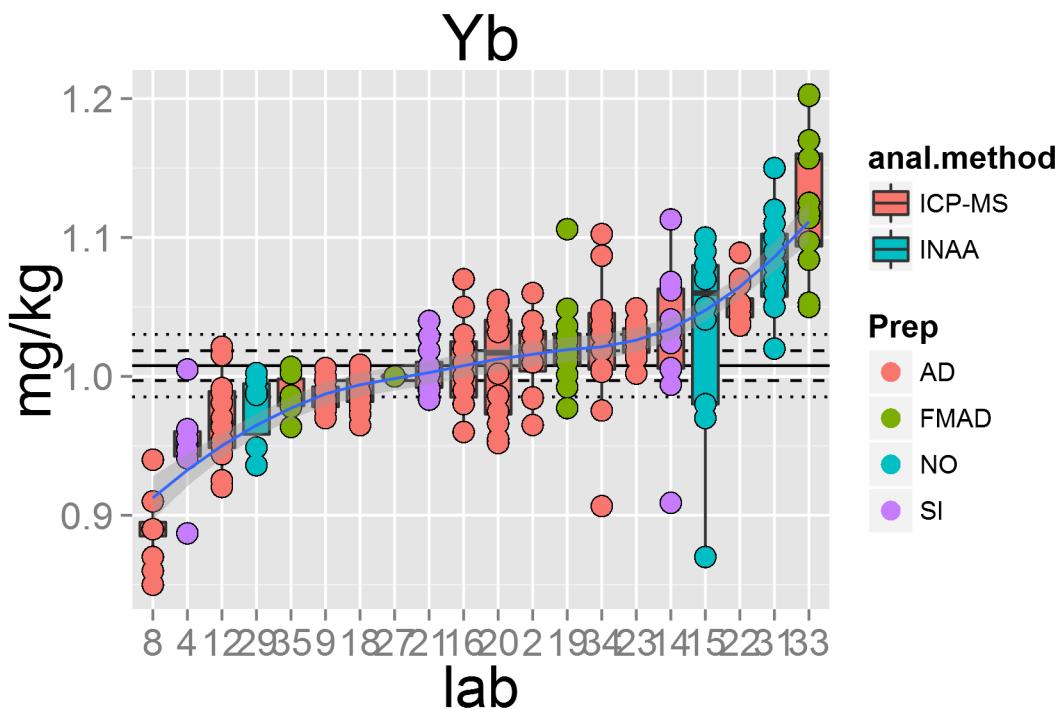
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



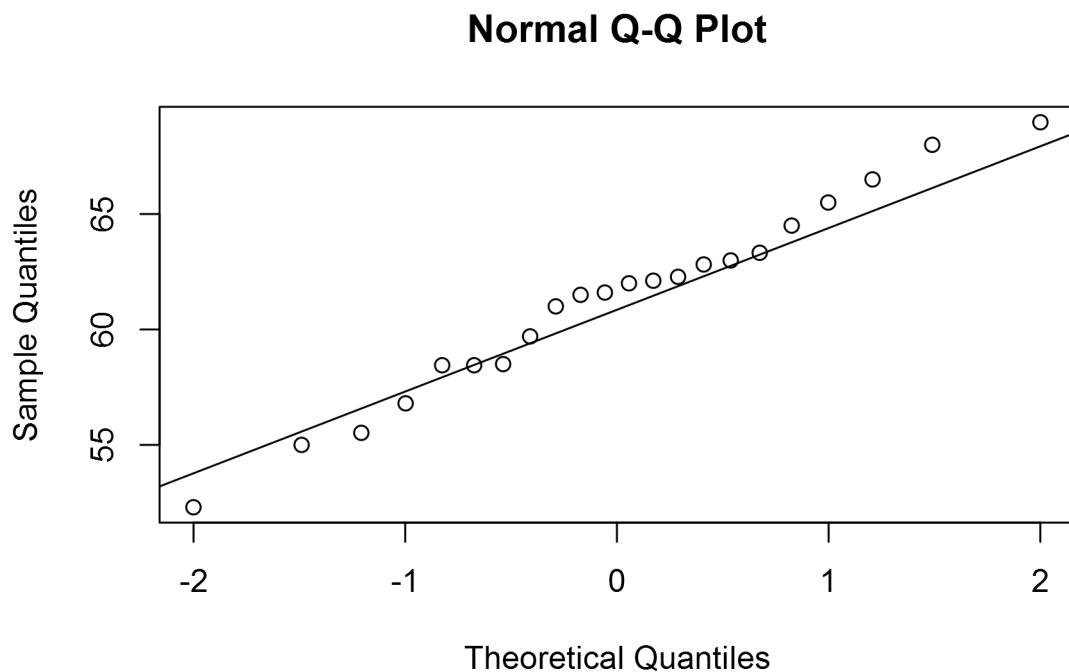
```
## [1] "Yb"
```



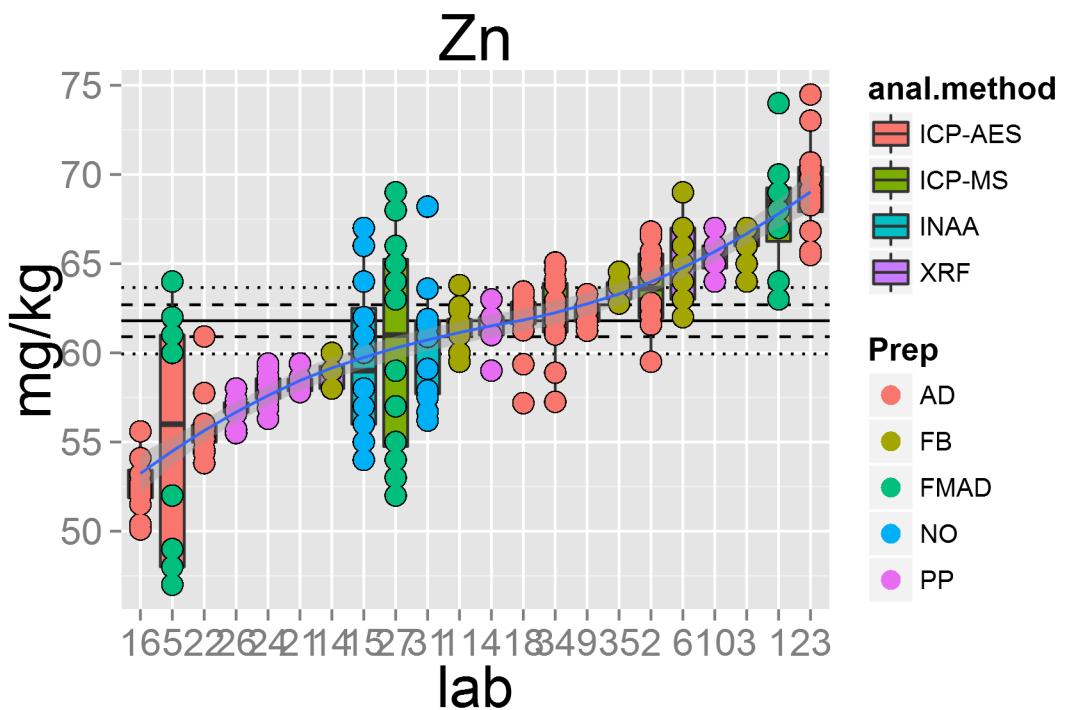
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



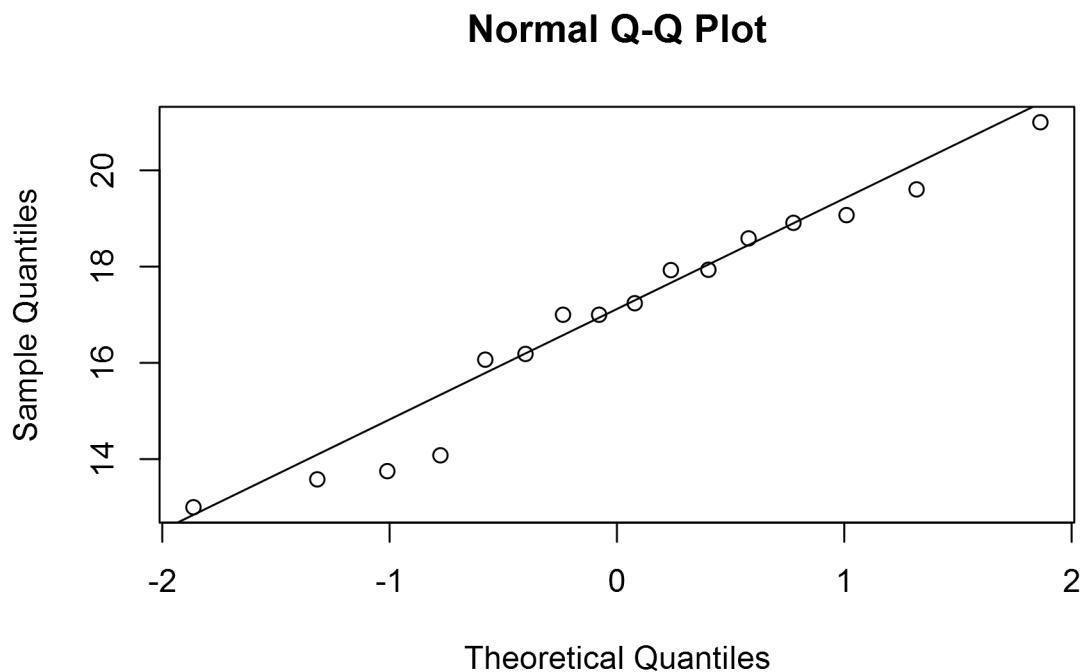
```
## [1] "Zn"
```



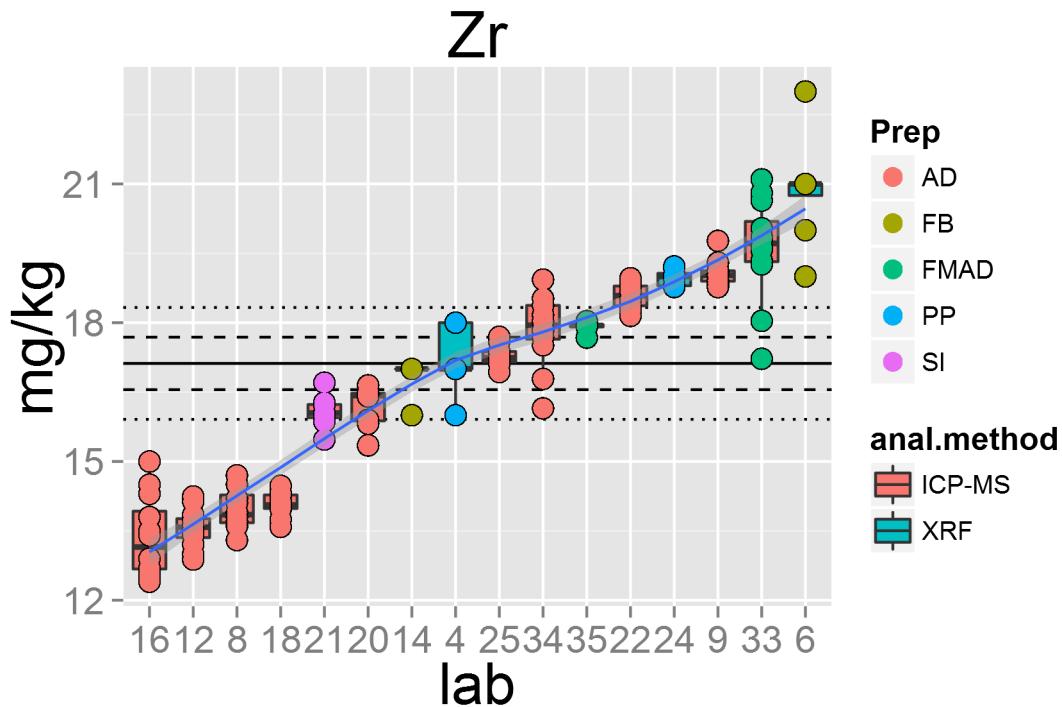
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



```
## [1] "Zr"
```



```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to change this.
```



```
df0 <- read.table("C:/Daten/projects/R/certification/df0.txt", header=T, quote="")
final <- ddply(df0, c("date", "RM", "measurand", "based.on", "unit"), numcolwise(meanGOM))
write.csv(final, "OKUMall.csv")
```

```
kable(final, format = "markdown", padding=0, digits=c(0,0,0,0,3,3,3,3,3,3,3,4,4,4,0,0,3,3,0))
```

date	RM	measurand	based.on	unit	mean.before	mean.after	median.before	median.after	media
2014-07-21	OKUM	Al2O3	median	g/100g	7.906	7.970	7.975	7.978	
2014-07-21	OKUM	As	median	mg/kg	0.461	0.465	0.372	0.372	
2014-07-21	OKUM	Ba	median	mg/kg	7.344	6.376	6.224	6.161	
2014-07-21	OKUM	Be	median	mg/kg	0.065	0.065	0.066	0.066	
2014-07-21	OKUM	CaO	median	g/100g	7.883	7.849	7.855	7.851	
2014-07-21	OKUM	Ce	median	mg/kg	1.322	1.269	1.282	1.279	
2014-07-21	OKUM	Co	median	mg/kg	88.530	88.900	88.890	89.400	
2014-07-21	OKUM	CO2	median	g/100g	0.197	0.197	0.200	0.200	
2014-07-21	OKUM	Cr	median	mg/kg	2441.000	2460.000	2447.000	2456.000	
2014-07-21	OKUM	Cs	median	mg/kg	0.182	0.184	0.183	0.183	
2014-07-21	OKUM	Cu	median	mg/kg	43.900	43.460	43.500	43.340	
2014-07-21	OKUM	Dy	median	mg/kg	1.594	1.609	1.605	1.605	

date	RM	measurand	based.on	unit	mean.before	mean.after	median.before	median.after	media
2014-07-21	OKUM	Er	median	mg/kg	1.023	1.041	1.039	1.043	
2014-07-21	OKUM	Eu	median	mg/kg	0.300	0.300	0.302	0.302	
2014-07-21	OKUM	Fe2O3T	median	g/100g	11.970	11.800	11.830	11.810	
2014-07-21	OKUM	FeO	median	g/100g	8.133	8.133	8.095	8.095	
2014-07-21	OKUM	Ga	median	mg/kg	8.847	8.794	8.883	8.865	
2014-07-21	OKUM	Gd	median	mg/kg	1.130	1.138	1.172	1.172	
2014-07-21	OKUM	H2O.	median	g/100g	5.747	5.747	5.710	5.710	
2014-07-21	OKUM	Hf	median	mg/kg	0.574	0.551	0.560	0.556	
2014-07-21	OKUM	Ho	median	mg/kg	0.355	0.355	0.353	0.353	
2014-07-21	OKUM	K2O	median	g/100g	0.044	0.044	0.044	0.044	
2014-07-21	OKUM	La	median	g/100g	0.412	0.412	0.409	0.409	
2014-07-21	OKUM	Li	median	mg/kg	4.412	4.412	4.352	4.352	
2014-07-21	OKUM	LOI	median	g/100g	4.594	4.594	4.690	4.690	
2014-07-21	OKUM	Lu	median	mg/kg	0.148	0.148	0.149	0.149	
2014-07-21	OKUM	MgO	median	g/100g	21.240	21.290	21.300	21.300	
2014-07-21	OKUM	MnO	median	g/100g	0.183	0.181	0.180	0.180	
2014-07-21	OKUM	Na2O	median	g/100g	1.127	1.136	1.145	1.148	
2014-07-21	OKUM	Nb	median	mg/kg	0.457	0.365	0.355	0.352	
2014-07-21	OKUM	Nd	median	mg/kg	1.478	1.494	1.498	1.499	
2014-07-21	OKUM	Ni	median	mg/kg	876.600	886.100	879.100	880.500	
2014-07-21	OKUM	P2O5	median	g/100g	0.028	0.027	0.027	0.027	
2014-07-21	OKUM	Pb	median	mg/kg	0.260	0.261	0.263	0.263	
2014-07-21	OKUM	Pr	median	g/100g	0.235	0.235	0.242	0.242	
2014-07-21	OKUM	Rb	median	mg/kg	1.109	0.964	0.950	0.940	
2014-07-21	OKUM	Sb	median	mg/kg	0.079	0.079	0.078	0.078	
2014-07-21	OKUM	Sc	median	mg/kg	27.510	27.930	28.250	28.500	
2014-07-21	OKUM	SiO2	mean	g/100g	44.330	44.140	44.160	44.140	
2014-07-21	OKUM	Sm	median	mg/kg	0.710	0.715	0.710	0.714	
2014-07-21	OKUM	Sn	median	mg/kg	0.248	0.248	0.258	0.258	
2014-07-21	OKUM	Sr	median	mg/kg	16.010	16.110	16.350	16.350	
2014-07-21	OKUM	Ta	median	mg/kg	0.029	0.026	0.028	0.028	
2014-07-21	OKUM	Tb	median	mg/kg	0.225	0.225	0.229	0.229	
2014-07-21	OKUM	Th	median	mg/kg	0.042	0.031	0.032	0.028	
2014-07-21	OKUM	TiO2	median	g/100g	0.386	0.380	0.380	0.378	
2014-07-21	OKUM	Tl	median	mg/kg	0.015	0.015	0.014	0.014	
2014-07-21	OKUM	Tm	median	mg/kg	0.155	0.153	0.155	0.155	

date	RM	measurand	based.on	unit	mean.before	mean.after	median.before	median.after	media
2014-07-21	OKUM	U	median	mg/kg	0.027	0.014	0.012	0.012	
2014-07-21	OKUM	V	median	mg/kg	166.000	167.800	167.500	169.500	
2014-07-21	OKUM	Y	median	mg/kg	9.141	9.082	9.000	8.998	
2014-07-21	OKUM	Yb	median	mg/kg	1.009	1.009	1.008	1.008	
2014-07-21	OKUM	Zn	median	mg/kg	61.580	61.220	62.000	61.800	
2014-07-21	OKUM	Zr	median	mg/kg	18.810	16.970	17.930	17.120	

```
#library(xtable)
#xt <- xtable(final, digits=c(0,0,0,0,0,3,3,3,3,3,3,3,3,3,2,0,0,3,3,0))
#print(xt, type="html")
#library(Gmisc)
# htmlTable(final)
```

```
certified.values <- data.frame(final$date, final$RM, final$measurand, final$t.value, final$labs.remaining)
names(certified.values) <- c("date", "RM", "measurand", "t.value", "n", "PV", "U", "unit")
write.table(certified.values, "CV1.txt", row.names=FALSE)
CV2 <- subset.data.frame(certified.values, n >= 10) # CV based on IAG protocol with n >= 10
CV3 <- subset.data.frame(certified.values, n < 10) # Information value based on IAG protocol with n < 10
write.table(CV2, "CV2.txt", row.names=FALSE) # just CV
write.table(CV3, "CV3.txt", row.names=FALSE, append=TRUE, col.names=FALSE) # CV and IV
CV <- read.table("C:/Daten/projects/R/certification/CV2.txt", header=TRUE, quote="")
#xtCV <- xtable(CV, digits=c(0,0,0,0,2,0,4,4,0))
#print(xtCV, type="html")
kable(CV, digits=c(0,0,0,2,2,4,4,0), padding=1)
```

date	RM	measurand	t.value	n	PV	U	unit
2014-07-21	OKUM	Al2O3	2.08	22	7.9780	0.0394	g/100g
2014-07-21	OKUM	Ba	2.13	16	6.1610	0.3960	mg/kg
2014-07-21	OKUM	CaO	2.09	20	7.8510	0.0573	g/100g
2014-07-21	OKUM	Ce	2.11	18	1.2790	0.0339	mg/kg
2014-07-21	OKUM	Co	2.07	23	89.4000	1.1400	mg/kg
2014-07-21	OKUM	Cr	2.08	22	2456.0000	31.0000	mg/kg
2014-07-21	OKUM	Cs	2.18	13	0.1831	0.0034	mg/kg
2014-07-21	OKUM	Cu	2.09	20	43.3400	1.2200	mg/kg
2014-07-21	OKUM	Dy	2.12	17	1.6050	0.0393	mg/kg
2014-07-21	OKUM	Er	2.11	18	1.0430	0.0143	mg/kg
2014-07-21	OKUM	Eu	2.10	19	0.3019	0.0068	mg/kg
2014-07-21	OKUM	Fe2O3T	2.08	22	11.8100	0.0484	g/100g

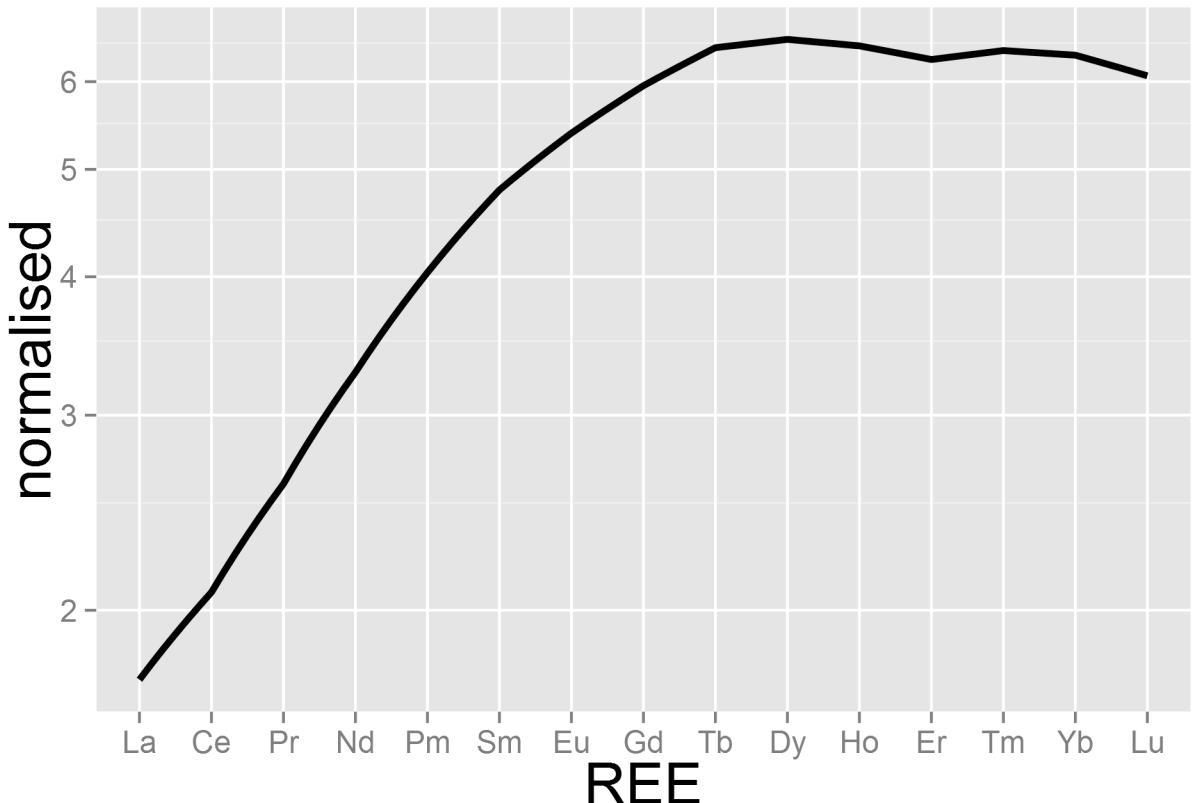
date	RM	measurand	t.value	n	PV	U	unit
2014-07-21	OKUM	Ga	2.12	17	8.8650	0.1640	mg/kg
2014-07-21	OKUM	Gd	2.13	16	1.1720	0.0425	mg/kg
2014-07-21	OKUM	Hf	2.14	15	0.5565	0.0228	mg/kg
2014-07-21	OKUM	Ho	2.14	15	0.3529	0.0085	mg/kg
2014-07-21	OKUM	K2O	2.13	16	0.0436	0.0020	g/100g
2014-07-21	OKUM	La	2.11	18	0.4086	0.0170	g/100g
2014-07-21	OKUM	LOI	2.08	22	4.6900	0.1190	g/100g
2014-07-21	OKUM	Lu	2.10	19	0.1488	0.0047	mg/kg
2014-07-21	OKUM	MgO	2.09	21	21.3000	0.0972	g/100g
2014-07-21	OKUM	MnO	2.07	23	0.1800	0.0027	g/100g
2014-07-21	OKUM	Na2O	2.08	22	1.1480	0.0215	g/100g
2014-07-21	OKUM	Nb	2.23	11	0.3525	0.0589	mg/kg
2014-07-21	OKUM	Nd	2.13	16	1.4990	0.0204	mg/kg
2014-07-21	OKUM	Ni	2.09	21	880.5000	10.3000	mg/kg
2014-07-21	OKUM	P2O5	2.12	17	0.0265	0.0023	g/100g
2014-07-21	OKUM	Pr	2.12	17	0.2416	0.0080	g/100g
2014-07-21	OKUM	Rb	2.16	14	0.9400	0.0563	mg/kg
2014-07-21	OKUM	Sc	2.09	21	28.5000	0.8750	mg/kg
2014-07-21	OKUM	SiO2	2.08	22	44.1400	0.1350	g/100g
2014-07-21	OKUM	Sm	2.11	18	0.7138	0.0113	mg/kg
2014-07-21	OKUM	Sr	2.07	23	16.3500	0.3110	mg/kg
2014-07-21	OKUM	Ta	2.20	12	0.0276	0.0038	mg/kg
2014-07-21	OKUM	Tb	2.11	18	0.2286	0.0056	mg/kg
2014-07-21	OKUM	Th	2.23	11	0.0285	0.0036	mg/kg
2014-07-21	OKUM	TiO2	2.07	23	0.3781	0.0042	g/100g
2014-07-21	OKUM	Tm	2.13	16	0.1549	0.0037	mg/kg
2014-07-21	OKUM	U	2.20	12	0.0121	0.0026	mg/kg
2014-07-21	OKUM	V	2.10	19	169.5000	3.0500	mg/kg
2014-07-21	OKUM	Y	2.10	19	8.9980	0.2900	mg/kg
2014-07-21	OKUM	Yb	2.09	20	1.0080	0.0225	mg/kg
2014-07-21	OKUM	Zn	2.08	22	61.8000	1.8600	mg/kg
2014-07-21	OKUM	Zr	2.13	16	17.1200	1.2100	mg/kg
2014-07-21	OKUM	As	2.45	7	0.3725	0.2260	mg/kg
2014-07-21	OKUM	Be	2.57	6	0.0660	0.0041	mg/kg
2014-07-21	OKUM	CO2	12.71	2	0.2000	0.5450	g/100g
2014-07-21	OKUM	FeO	4.30	3	8.0950	0.1650	g/100g

date	RM	measurand	t.value	n	PV	U	unit
2014-07-21	OKUM	H2O.	4.30	3	5.7100	0.3100	g/100g
2014-07-21	OKUM	Li	2.36	8	4.3520	0.2590	mg/kg
2014-07-21	OKUM	Pb	2.45	7	0.2626	0.0210	mg/kg
2014-07-21	OKUM	Sb	2.78	5	0.0780	0.0124	mg/kg
2014-07-21	OKUM	Sn	2.78	5	0.2580	0.0228	mg/kg
2014-07-21	OKUM	Tl	4.30	3	0.0143	0.0019	mg/kg

```

REE.chondrites <- read.csv("~/GitHub/REE/REE.chondrites.csv", sep=";") # reading chondrite normalising values
REE.chondrites <- rename(REE.chondrites, c("norm"= "REE")) # renaming the column to make it suitable for merging
REE <- c("La", "Ce", "Pr", "Nd", "Pm", "Sm", "Eu", "Gd", "Tb", "Dy", "Ho", "Er", "Tm", "Yb", "Lu") # needed REE elements
RM <- read.table("~/GitHub/GOMcertification/CV2.txt", header=TRUE, quote="\\") # reading all the final REE values
RM.REE <- subset(RM, measurand %in% REE, select=c(RM,measurand,PV)) # extracting only the REE values
RM.REE <- rename(RM.REE, c("measurand"="REE")) # renaming the column head suitable for merging
RM.REE <- merge(REE.chondrites, RM.REE, by = "REE", all.x=TRUE) # merging the data set so that element names are aligned
RM.normalised <- ddply(RM.REE, c("RM"), transform, normalised = PV/chondrite) # adding a chondrite normalised value
RM.normalised[15,5] <- (RM.normalised[9,5]+RM.normalised[11,5])/2
REEtheme <- theme_grey() + theme(plot.title = element_text(colour = "black", size = rel(2)))+ theme(axis.ticks = element_text(size=1.2))
p <- ggplot(data=RM.normalised, aes(x=REE, y=normalised, group=1)) + geom_line(size= 1.2)
p + scale_x_discrete(limits=c("La", "Ce", "Pr", "Nd", "Pm", "Sm", "Eu", "Gd", "Tb", "Dy", "Ho", "Er", "Tm", "Yb", "Lu"))

```



```
sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=German_Austria.1252  LC_CTYPE=German_Austria.1252
## [3] LC_MONETARY=German_Austria.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Austria.1252
##
## attached base packages:
## [1] grid      stats     graphics grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] knitr_1.6      plyr_1.8.1      nlme_3.1-117    ape_3.1-4
## [5] metRology_0.9-17 plotflow_1.0      gridExtra_0.9.1 ggplot2_1.0.0
##
## loaded via a namespace (and not attached):
## [1] colorspace_1.2-4  digest_0.6.4      evaluate_0.5.5
## [4] formatR_0.10      gtable_0.1.2      htmltools_0.2.4
## [7] labeling_0.2       lattice_0.20-29   MASS_7.3-33
## [10] munsell_0.4.2     numDeriv_2012.9-1 proto_0.3-10
## [13] Rcpp_0.11.2       reshape2_1.4      rmarkdown_0.2.50
## [16] scales_0.2.4      stringr_0.6.2      tools_3.1.1
## [19] yaml_2.1.13
```