

OKUM Certification

TCM

Tuesday, July 21, 2014

Analysing submitted data for OKUM, GAS or MUH-1

Thomas Meisel (2014-07-24)

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 PacketxPrepDay). important notice: .RProfile was originally taken from <http://gforge.se/2014/01/fast-track-publishing-using-knitr-part-ii/> but does not work with the RStudio version 0.98.976 as PDF option does not show up

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

importing the data and assigning factors

defining the RM and measurand to be analysed

```
refmat <- 'OKUM' # defining the RM
measurand.name <- 'P205'
```

measurand selected P2O5 for candidate reference material OKUM

```
# 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'
```

```
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/comp
```

```
## [1] "M"
```

```
ifelse(MorT == "T", unit <- "mg/kg", unit <- "g/100g") # testing which unit is needed
```

```
## [1] "g/100g"
```

defining the plotting theme

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 calculations
sdGOM <- function(x) sd(x, na.rm = TRUE) # defining a function for further calculations, here for calcu
meanGOM.packet <- ddply(GOM, c("Lab", "Packet"), numcolwise(meanGOM)) # calculated the mean for each P
## mean over mean of packets within lab
GOM.mean <- ddply(meanGOM.packet, c("Lab"), numcolwise(meanGOM))
```

```
## median over packets within lab
medianGOM <- function(x) median(x, na.rm=TRUE) # defining a function for further calculations
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))# calculated the median for each
GOM.median <- merge(GOM.median, OKUM.methods, by="Lab")
```

```
summary(GOM[[measurand]], na.rm=TRUE, digits=4) # with values without outlier removal
```

initial calculations with complete data set

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      0.02   0.02   0.03   0.03   0.03   0.08       210
```

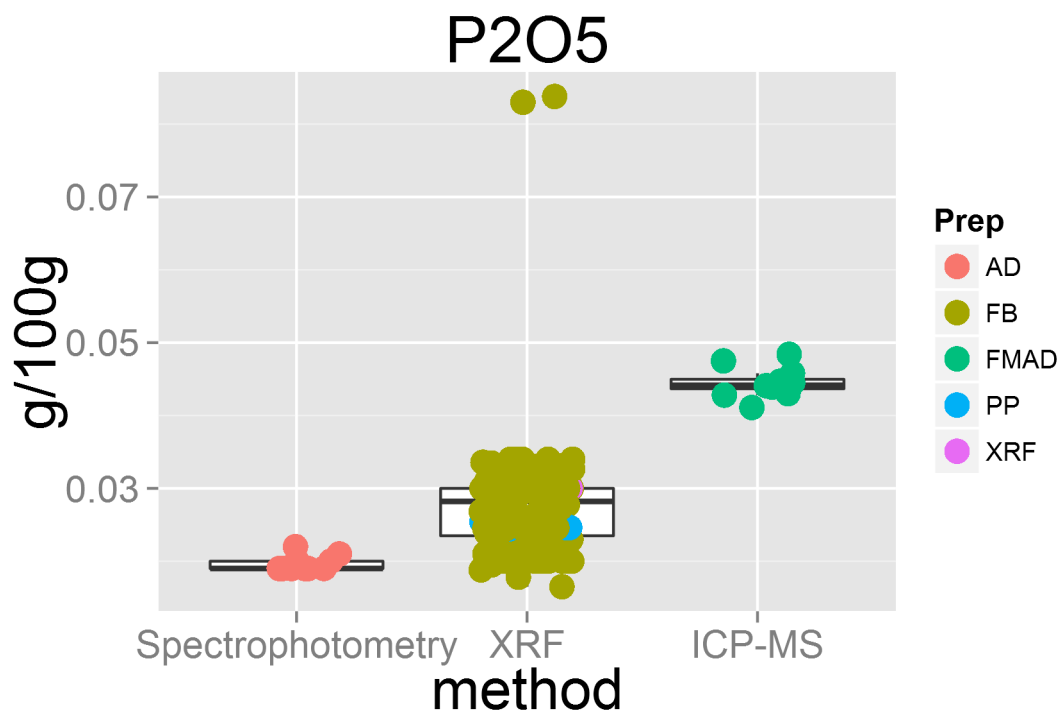
```
mean.before <- mean(GOM.mean[[measurand]], na.rm=TRUE)
median.before <- median(GOM.median[[measurand]], na.rm=TRUE)
```

The P2O5 mean of the Lab means is **0.028** g/100 g

The P2O5 median of the Lab+Package medians is **0.0272** g/100 g

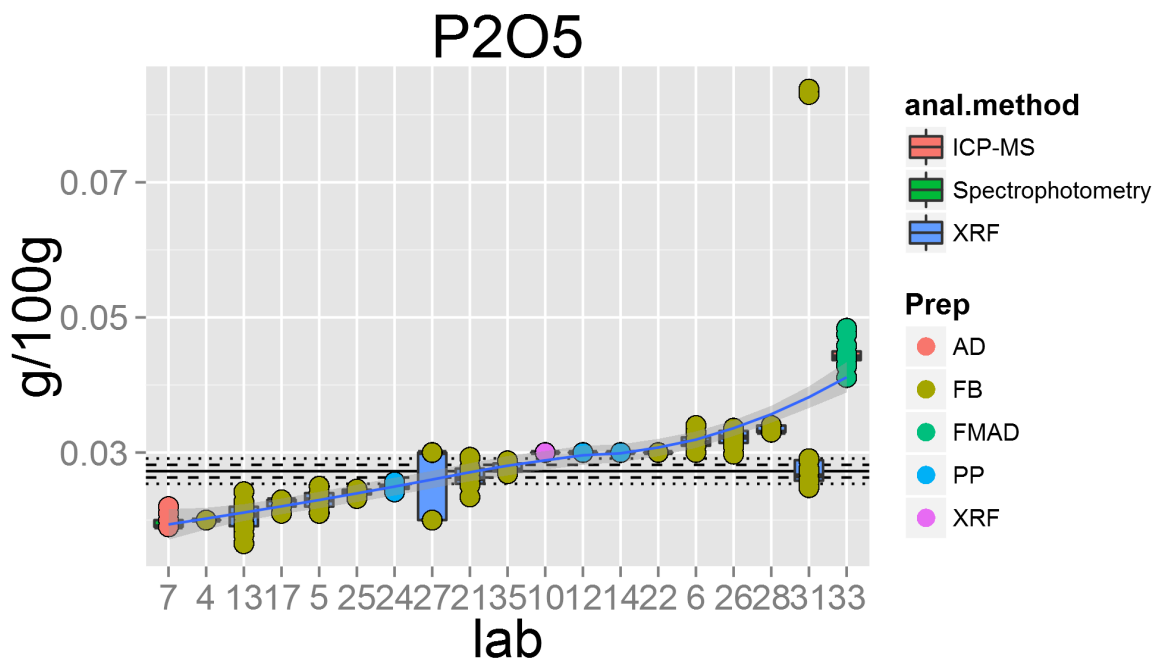
```
plot_method(measurand)
```

```
## 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).
```



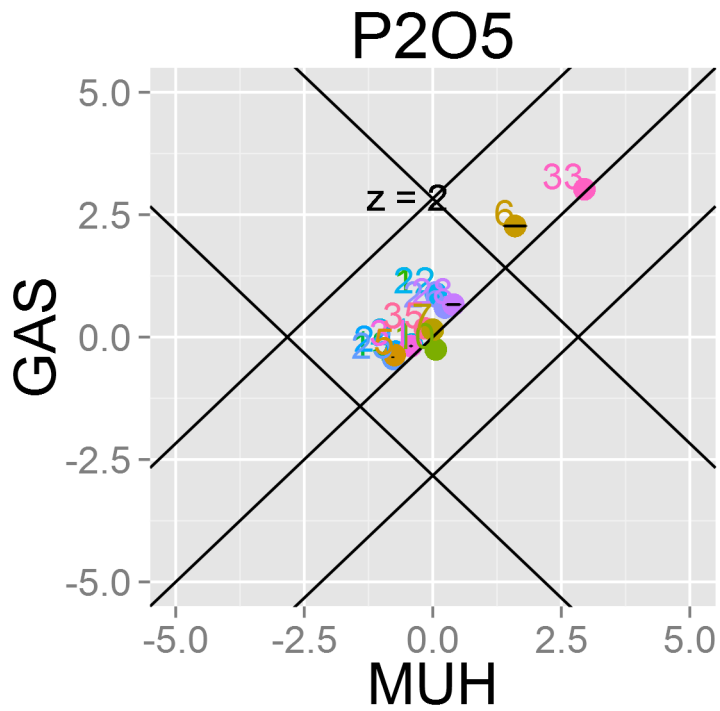
```
plot_lab(measurand, MorT, horw = TRUE, u = FALSE)
```

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



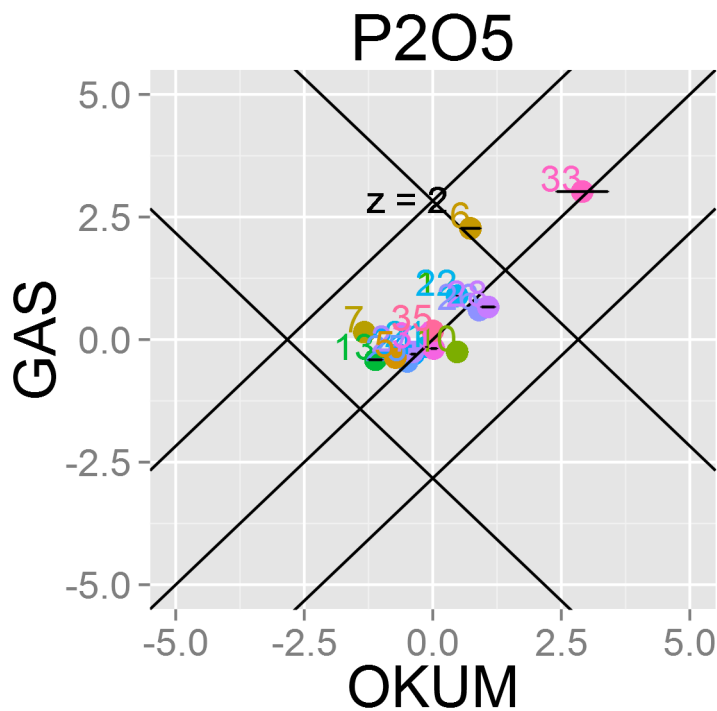
```
plot_youd(measurand.name, 'MUH', 'GAS')
```

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



```
plot_youd(measurand.name, 'OKUM', 'GAS')
```

```
## Warning: Removed 20 rows containing missing values (geom_point).  
## Warning: Removed 20 rows containing missing values (geom_text).
```



Mandel k barplot displays the within lab performance relative to all participating labs using the median over packages

```
#k <- with(medianGOM.packet, mandel.kh(medianGOM.packet[[measurand]], g=Lab, type="k"))
#barplot(k, las=2, col=1:4)
```

Outlier rejection

```
outlier <- OKUM.outlier[[measurand.name]]
#outlier <- ifelse(OKUM.outlier[[measurand.name]][1]=="NA", "0", OKUM.outlier[[measurand.name]]) # does
outlier <- na.omit(outlier)
# outlier<- c(16) ## defining the outlying lab here with lab# directly. Enter 0 when no 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))
}
```

Removing outliers lab based on Youden plot and Mandel's k (lab performance)

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

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      0.02   0.02   0.03   0.03   0.03   0.05     222
```

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

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      0.02   0.02   0.03   0.03   0.03   0.03     234
```

```
# plot_lab(measurand, Mort, horw =TRUE, u = FALSE) # replotting without outlier, M for majors and T for
```

Calculating mean and median of property value

```
summary(GOM[[measurand]], na.rm=TRUE, digits=4)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      0.02   0.02   0.03   0.03   0.03   0.03     234
```

```
mean <- mean(tapply(GOM[[measurand]], GOM$Lab, mean, na.rm=TRUE), na.rm=TRUE) # mean of lab means after
median.before <- median(GOM.median[[measurand]], na.rm=TRUE) # median of measurand w/o outlier removal
medianGOM.packet.after <- ddply(GOM, c("Lab", "Packet"), numcolwise(medianGOM)) # median Lab and Packet
## median over median of packets within lab
GOM.median.after <- ddply(medianGOM.packet.after, c("Lab"), numcolwise(medianGOM)) # creating a new tab
GOM.median.after <- merge(GOM.median.after, OKUM.methods, by="Lab")
median.after <- median(GOM.median.after[[measurand]], na.rm=TRUE) # median of the individual measurand
## calculating method parameters
# '%p%' <- function(x, y) {as.character(paste(x, y, sep = ""))}
prep <- 'Prep.'
method <- 'Method.'
anal.prep <- prep %p% measurand.name
anal.method <- method %p% measurand.name
anal <- GOM.median.after[[measurand]]
anal.prep <- GOM.median.after[[anal.prep]]
anal.method <- GOM.median.after[[anal.method]]
analyte <- data.frame(GOM.median.after$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))
```

Measurement uncertainty estimations of property value

The the mean of means is $\bar{Y} = \frac{1}{p} \sum_{i=1}^p Y_i$

The standard deviation is $s^2 = \frac{1}{p-1} \sum_{i=1}^p (Y_i - \bar{Y})^2$

The combined standard uncertainty equals then $u_{char} = \frac{s}{\sqrt{p}}$

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 methods, 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 , ϵ_{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 . They are related as follows

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

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

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

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

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

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

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

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

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

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

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

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 extrated 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 compenent.

before outlier removal

```
medianGOM.packet$Lab <- as.factor(medianGOM.packet$Lab) # using only the median of the 3 packages per l
medianGOM.packet$Packet <- as.factor(medianGOM.packet$Packet)
anal <- medianGOM.packet[[measurand]]
DF.lme <- data.frame(medianGOM.packet$Lab, medianGOM.packet$Packet, medianGOM.packet[[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 <- varcomp(GOM.lme, FALSE, FALSE)[[1]] # between-laboratory variance
sbb2 <- varcomp(GOM.lme, FALSE, FALSE)[[2]] # between bottle variance
sr2 <- varcomp(GOM.lme, FALSE, FALSE)[[3]] # repeatability standard deviation
n.p <- dim(DF.lme)[1] # number of oberventions
p <- length(unique(DF.lme$Lab)) # haven't found a better way how to extract the number of labs (number
r <- length(unique(DF.lme$Packet))
u1 <- sqrt(sL2/p+sbb2/p/r+sr2/p/r/4) # calculating the standard uncertainty of characterization
u2 <- attr(GOM.lme$fixDF,“varFixFact”) # gives the same results as u, amazing!
# print(plot(DF.lme))
```

after outlier removal

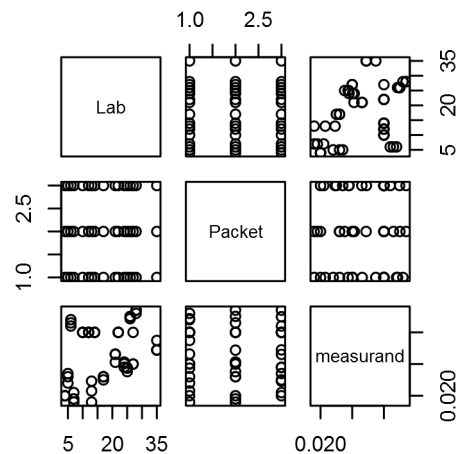
```
medianGOM.packet.after$Lab <- as.factor(medianGOM.packet$Lab) # using only the median of the 3 packages
medianGOM.packet.after$Packet <- as.factor(medianGOM.packet$Packet)
anal <- medianGOM.packet.after[[measurand]]
DF.lme <- data.frame(medianGOM.packet.after$Lab, medianGOM.packet.after$Packet, medianGOM.packet.after[
```



```

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$Lab)) # haven't found a better way how to extract the number of labs (number
r <- length(unique(DF.lme$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)

```



uncertainty of data after outlier rejection.png

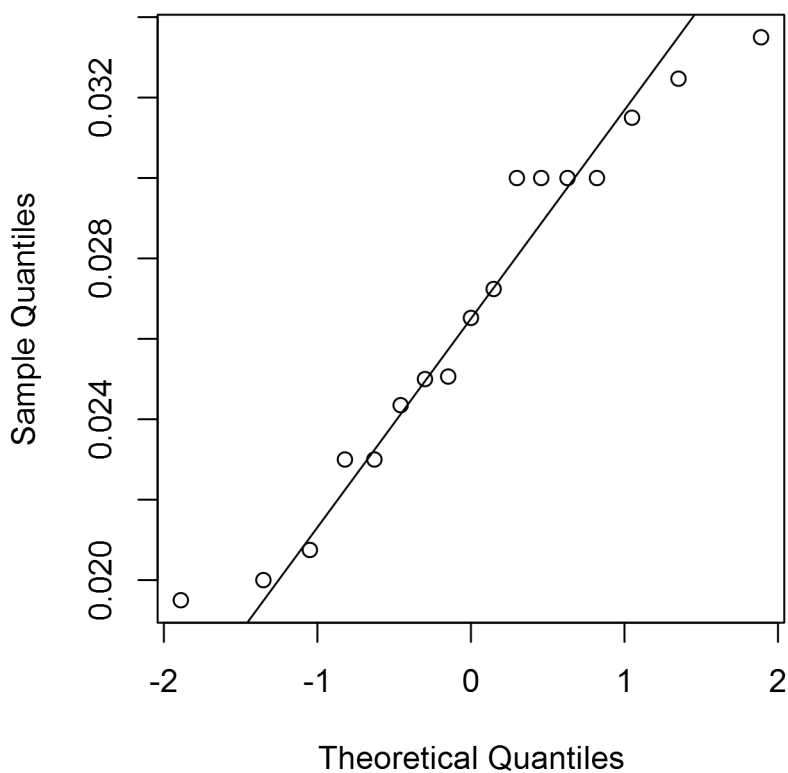
tests for normal distribution

```

qqnorm(GOM.median.after[[measurand]])
qqline(GOM.median.after[[measurand]])

```

Normal Q-Q Plot



plot.png

```
analyte.noPP <- subset(analyte, analyte$anal.prep!="PP") # Removing all PP preparations for comparison
median.after.noPP <- median(analyte.noPP$anal)
```

Comparisons of property value calculations for OKUM and measurand P2O5

before and after removal of outlying lab(s) # 31, 33

type	before	after	unit
mean	0.028	0.0266	g/100g
median	0.0272	0.0265	g/100g
median w/o PP	-	0.0258	g/100g

performance characteristics for OKUM and measurand P2O5

type	variance before	variance after outlier rejection
between-laboratory variance	3.3386×10^{-5}	1.9341×10^{-5}

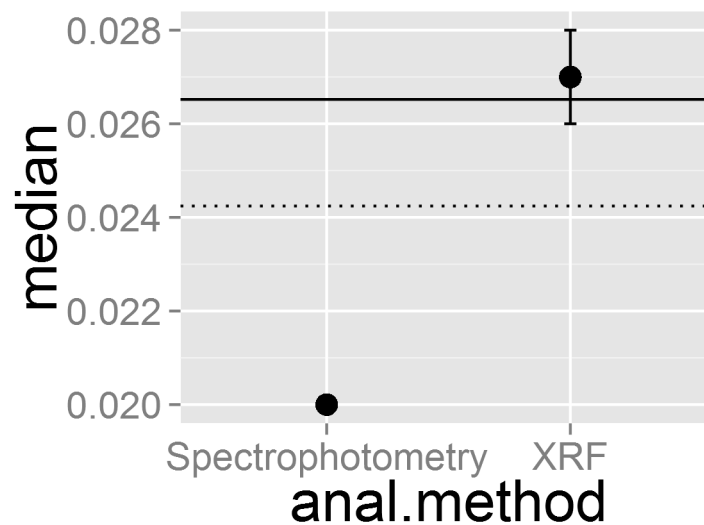
type	variance before	variance after outlier rejection
between-bottle variance	1.2028×10^{-5}	5.3042×10^{-7}
repeatability variance	2.9381×10^{-6}	3.0934×10^{-7}

mesasurment uncertainty

type	before	after outlier rejection	alternative calculation
standard uncertainty	0.0014	0.0011	0.0011
expanded standard uncertainty	-	-	0.0023 $k = 2.12$

```
outlier <- ifelse(outlier=="0", "X", outlier)
df <- data.frame(Sys.Date(), refmat, measurand.name, mean.before, mean, median.before, median.after, median.after, median.after)
#names(df) <- c("date", "RM", "measurand", "mean.before", "mean", "median.before", "median.after", "median.after")
#write.table(df, "dfOKUM.txt", row.names=FALSE) # needed only the first time
write.table(df, "dfOKUM.txt", row.names=FALSE, append=TRUE, col.names=FALSE)
```

```
reference.line <- median.after
u.Ulim <- median.after + u2.a*t.value
l.Ulim <- median.after - u2.a*t.value
ggplot(bymethod.n, aes(x=anal.method, y=median))+geom_point(size=4)+geom_errorbar(aes(ymin=median-se, ymax=median+se))
```

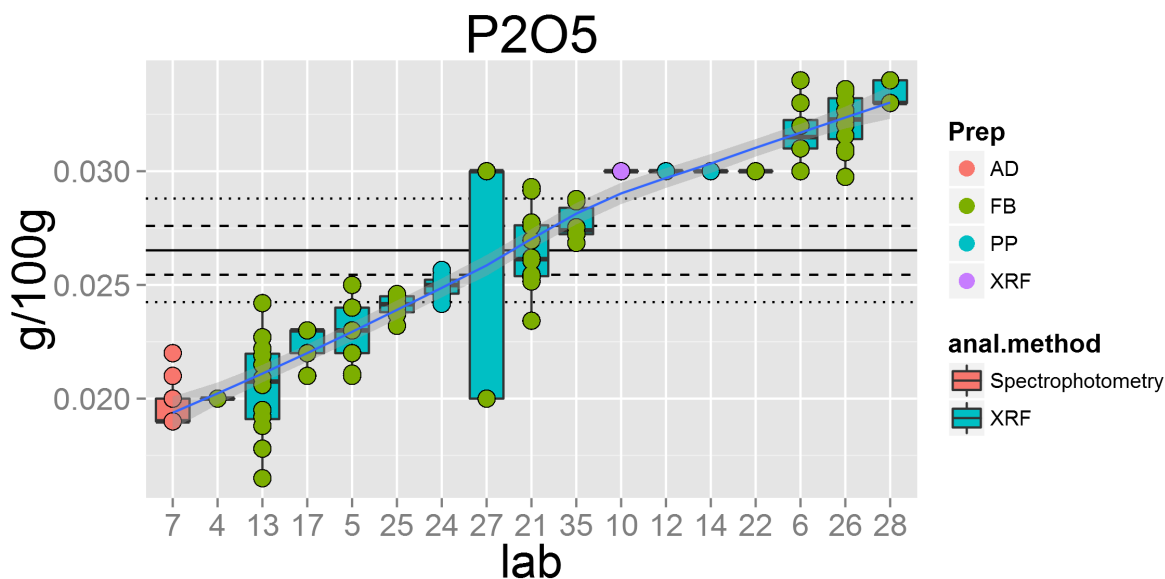


the method medians.png

Final plot without outliers with u and U

```
plot_lab(measurand, MorT, horw = FALSE, u = TRUE)
```

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



Comparisons of method parameters based on median of the Lab+packet medians after outlier removal

```
print(bymethod.n)
```

```
##          anal.method  N  mean median    sd    se
## 1 Spectrophotometry  1 0.020  0.020    NA    NA
## 2                XRF 16 0.027  0.027 0.004 0.001
```

final result based on median for property value and measurement uncertainty based on variance components

The **P2O5** median of the Lab+packet median after outlier removal is **0.0265** g/100g
 The expanded standard uncertainty for the assigned value of P2O5 is **0.0023** with $k = 2.12$
 excluded labs for P2O5 is/are 31, 33
 labs remaining for calculations 17

comments

outlier type | lab# :—| :— based on Youden plots: |33
 based on Mandel's k performance:|31
 based on PP-XRF:|
 based on detection limit:|

comments

outlier type	lab#
based on Youden plots:	33

outlier type	lab#
based on Mandel's k performance:	33
based on PP-XRF:	33

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
```