Monumentality by Numbers

Hageneuer, Sebastian

Schmidt, Sophie

15 Juni, 2018

## intro

This R package offers the code to the calculations described in: Hageneuer, S. / Schmidt, S., Monumentality by Numbers, in: Hageneuer et al. (Hrsg.), Size Matters - Understanding Monumentality across ancient Civilizations (transcript-Verlag Bielefeld 2019).

uebersicht <- read.csv2("../data/raw\_data/uebersicht3.csv", sep = "\t", dec=",", fileEncoding="UTF-8")  
  
# load needed packages  
library(tidyverse)  
library(ggrepel)  
library(ggpmisc)  
library(RColorBrewer)  
library(reshape2)  
library(scales)  
library(mgcv)

# Effort calculation

Aim of this part: Sum the different materials (e.g. differently coloured ceramic cones) that are going to be analysed together and calculate the effort, which is dependant on the volume of different materials. Sum volume and effort of each building.

# summing data, create table with all relevant volumes of material categories  
  
aufwand <- data.frame(uebersicht$X, uebersicht$Gebaeude, uebersicht$Mriemchen + uebersicht$Mlehmziegel, uebersicht$Mlehm, uebersicht$Mlehm\_g + uebersicht$Mlehm\_w, uebersicht$Mschilf, uebersicht$Mholz, uebersicht$Masphalt + uebersicht$Kalksteinplatten, (uebersicht$Mkeramikstifte\_r + uebersicht$Mkeramikstifte\_schw + uebersicht$Mkeramikstifte\_w + uebersicht$Mkeramikstifte\_g + uebersicht$Mflaschen), uebersicht$MasseSteinstifte, uebersicht$Mkunststein)  
colnames(aufwand)[3] <- "Mziegel" # Riemchen sind Ziegel! # rename column to Mziegel = mass of mud bricks  
colnames(aufwand)[5] <- "MFarblehm" # gelber und weißer # rename column to MFarblehm = mass of coloured clay  
colnames(aufwand)[8] <- "MsonstMat" # Apshalt und Kalkstein # rename column to MsonstMat = mass of other materials  
colnames(aufwand)[9] <- "MKeramikZier" # Keramische Zierelemente: Flaschen und Keramikstifte # rename column to MKeramikzier = mass of ceramic decoration  
colnames(aufwand)[10] <- "MSteinstifte" # Steinstifte untersch. Farben # rename column to MSteinstifte = mass of stone cones  
  
# sum the volumes  
aufwand$summe\_volumen <- aufwand$Mziegel + aufwand$uebersicht.Mlehm + aufwand$uebersicht.Mschilf + aufwand$uebersicht.Mholz + aufwand$MsonstMat + aufwand$MSteinstifte + aufwand$MKeramikZier + aufwand$MFarblehm + uebersicht$Mkunststein  
  
# multiply with effort factor  
aufwand$Lehm <- aufwand$uebersicht.Mlehm\*1  
aufwand$Schilf <- aufwand$uebersicht.Mschilf\*1  
aufwand$Farblehm <- aufwand$MFarblehm\*2  
aufwand$Ziegel <- aufwand$Mziegel\*4  
aufwand$KeramikZier <- aufwand$MKeramikZier\*4  
aufwand$holz <- aufwand$uebersicht.Mholz\*7  
aufwand$Steinstifte <- aufwand$MSteinstifte\*8  
aufwand$sonstMat <- aufwand$MsonstMat\*12  
aufwand$Kunststein<- uebersicht$Mkunststein\*16  
  
# sum all effort  
aufwand$summe\_aufwand <- aufwand$Lehm + aufwand$Schilf + aufwand$Farblehm + aufwand$Ziegel + aufwand$holz + aufwand$KeramikZier + aufwand$sonstMat + aufwand$Steinstifte + aufwand$Kunststein

# Analyse the middle-hall buildings quantitatively

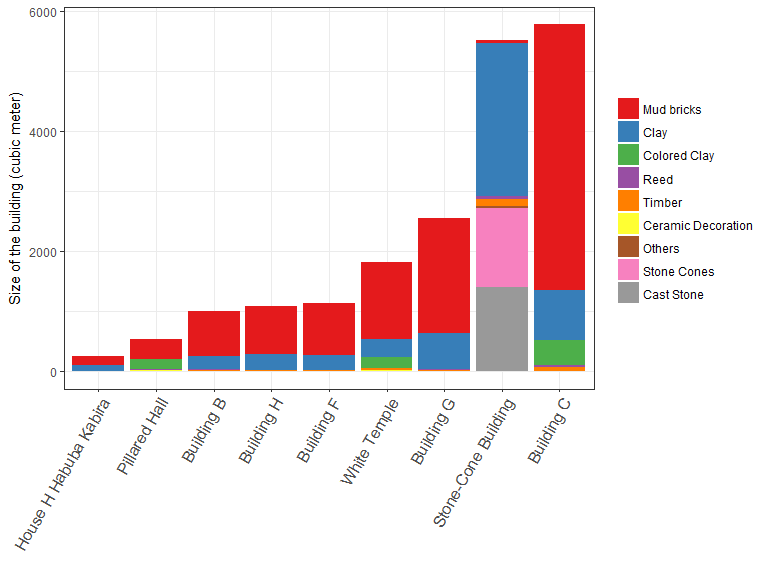
Aim of the following code chunk: We subsetted a larger data frame for just the middle-hall buildings. Calculations later showed we need the data frame of middle-hall buildings without the stone cone building as well.

## subset and remove Stone-Cone Building  
  
aufwand\_MSG <- subset(aufwand, uebersicht.Gebaeude == "Mittelsaalgebaeude")   
aufwand\_MSG\_oSSG <- aufwand\_MSG[-9,] ## remove stone-cone building

## amount of material

Aim: Create a stacked bar plot, in which we can see the amount of different materials used for each building. Absolute numbers used so the stack shows the absolute size of the building.

# remove unwanted data: effort calculations not needed  
  
vol\_MSG3 <- aufwand\_MSG  
vol\_MSG3$category <- row.names(vol\_MSG3)  
vol\_MSG3 <- vol\_MSG3[,1:12]  
  
# melt dataframe  
  
vol\_MSG3\_m <- melt(vol\_MSG3, id.vars = c("uebersicht.X", "uebersicht.Gebaeude", "summe\_volumen"))  
  
  
vol\_MSG3\_m <- vol\_MSG3\_m %>%  
 arrange(summe\_volumen, uebersicht.X) %>% # sort the dataframe for its size  
 mutate(uebersicht.X = factor(uebersicht.X, unique(uebersicht.X))) # change factor levels to how they've been arranged above  
  
vol\_MSG3\_m$variable <- ordered(vol\_MSG3\_m$variable, levels = c("Mziegel", "uebersicht.Mlehm", "MFarblehm", "uebersicht.Mschilf", "uebersicht.Mholz", "MKeramikZier", "MSteinstifte", "MsonstMat", "uebersicht.Mkunststein")) # order variables for the stacked bar plot so they are always in the same order (to compare with the following bar plot)  
  
  
ggplot(data = vol\_MSG3\_m)+  
 geom\_bar(aes(x = uebersicht.X, y = value, fill = variable),  
 stat = "identity",  
 position = "stack")+  
 scale\_y\_continuous("Size of the building (cubic meter)")+  
 labs(x = "")+  
 theme\_bw()+  
 theme(axis.text.x = element\_text(angle = 60, hjust = 1, size = 12))+  
 scale\_fill\_brewer(palette = "Set1",  
 breaks = c("Mziegel", "uebersicht.Mlehm", "MFarblehm", "uebersicht.Mschilf", "uebersicht.Mholz", "MKeramikZier", "MSteinstifte", "MsonstMat", "uebersicht.Mkunststein"),  
 labels = c("Mud bricks", "Clay", "Colored Clay", "Reed", "Timber", "Ceramic Decoration", "Others", "Stone Cones", "Cast Stone"))+  
 guides(fill=guide\_legend(title=NULL))



ggsave("../figures/Mittelsaalgebaeude\_MASSE.eps", width = 20, height = 15, units = "cm") # save plot for publication

## How much effort do the materials imply?

Aim: Create a stacked bar plot, in which we can see the effort of the different materials used for each building.Percentage used so the stack shows composition of material effort without regard to absolute size of the building.

aufwand\_MSG2 <- aufwand\_MSG  
#remove unwanted info: no volumes needed  
aufwand\_MSG2$category <- row.names(aufwand\_MSG2)  
aufwand\_MSG2 <- select(aufwand\_MSG2, -starts\_with ("uebersicht.M"))  
aufwand\_MSG2 <- select(aufwand\_MSG2, -starts\_with ("M"))  
  
aufwand\_MSG2 <- aufwand\_MSG2[,1:12]  
  
# melt data   
  
aufwand\_MSG\_m <- melt(aufwand\_MSG2, id.vars = c("uebersicht.X", "uebersicht.Gebaeude", "summe\_volumen"))  
  
aufwand\_MSG\_m <- aufwand\_MSG\_m %>%  
 arrange(summe\_volumen, uebersicht.X, variable) %>% # sort the dataframe for its size  
 mutate(uebersicht.X = factor(uebersicht.X, unique(uebersicht.X))) # change factor levels to how they've been arranged above  
  
aufwand\_MSG\_m$variable <- ordered(aufwand\_MSG\_m$variable, levels = c("Ziegel","Lehm", "Farblehm", "Schilf", "holz", "KeramikZier", "sonstMat", "Steinstifte", "Kunststein")) # order variables for the stacked bar plot so they are always in the same order (to compare with the bar plot before)  
  
  
ggplot(data = aufwand\_MSG\_m)+  
 geom\_bar(aes(x = uebersicht.X, y = value, fill = variable),  
 stat = "identity",  
 position = "fill")+  
 scale\_y\_continuous("Relative effort", labels = percent\_format())+  
 theme\_bw()+  
 scale\_fill\_brewer(palette = "Set1",  
 breaks =c("Ziegel","Lehm", "Farblehm", "Schilf", "holz", "KeramikZier", "sonstMat", "Steinstifte", "Kunststein"),  
 labels = c("Mud bricks", "Clay", "Colored Clay", "Reed", "Timber", "Ceramic Decoration", "Others", "Stone Cones", "Cast Stone"))+  
 labs(x = "")+  
 theme(axis.text.x = element\_text(angle = 60, hjust = 1, size = 12))+  
 guides(fill=guide\_legend(title=NULL))

ggsave("../figures/Mittelsaalgebaeude\_AUFWAND.eps", width = 20, height = 15, units = "cm") #save for publication

# Regression analysis

Aim: Analyse the relationship between size and effort. Starting hypothesis: linear relatoionship between size and effort. This gives a fairly good R²-value, but could be better. The stone-cone building doesn’t fit well. Also the residual analysis (see below) shows there’s a relationship between residuals and size. Therefore two options: 1. linear relationship without stone cone building 2. test non-linear regressions

## Linear regressions

ggplot(data = aufwand\_MSG)+  
 geom\_point(aes(x = summe\_volumen, # point cloud all middle-hall-buildings  
 y = summe\_aufwand))+  
 geom\_smooth(data = aufwand\_MSG, # regression all middle-hall-buildings -> bad fit  
 method='lm',  
 se = FALSE, # no confidence intervall  
 aes(x = summe\_volumen,  
 y = summe\_aufwand,   
 colour = "black" )) +   
 stat\_poly\_eq(data = aufwand\_MSG, # Information on regression all middle hall buidlings  
 aes(x = summe\_volumen,  
 y = summe\_aufwand,  
 label = paste(..eq.label.., # equation of regression line  
 ..adj.rr.label.., # adjusted R-squared  
 sep = "~~~~")),  
 rr.digits = 4, # 4 digits behind point  
 formula = y~x,   
 parse = TRUE,  
 size = 5,  
 color = "black",   
 label.y.npc = 0.8) +  
 stat\_fit\_glance(data = aufwand\_MSG, # p value of regression all middle hall buildings  
 aes(x = summe\_volumen,  
 y = summe\_aufwand,   
 label = paste("p-value: ",signif(..p.value.., digits = 4))),  
 label.y.npc = 0.8,  
 size = 5,  
 color = "black",   
 method = "lm",   
 method.args = list(formula = y ~ x),  
 geom = "text" ) +  
 geom\_smooth(data = aufwand\_MSG\_oSSG,  
 method='lm', # regression for middle-hall-buildings withouth stone cone building  
 se = FALSE, ## no confidence intervall  
 aes(x = summe\_volumen,  
 y = summe\_aufwand, colour = "gray50")) +  
 stat\_poly\_eq(data = aufwand\_MSG\_oSSG, # info on regression for middle-hall buildings without stone-cone building  
 aes(x = summe\_volumen,  
 y = summe\_aufwand,  
 label = paste(..eq.label..,   
 ..adj.rr.label..,  
 sep = "~~~~")),  
 rr.digits = 4,  
 formula = y~x,   
 parse = TRUE,  
 size = 5,  
 color = "gray50",  
 label.y.npc = 0.7)+ # Y-coordinate for this info  
 stat\_fit\_glance(data = aufwand\_MSG\_oSSG, # p-value  
 aes(x = summe\_volumen,  
 y = summe\_aufwand,   
 label = paste("p-value: ",signif(..p.value.., digits = 5))),  
 label.y.npc = 0.7,  
 size = 5,  
 color = "gray50",   
 method = "lm",   
 method.args = list(formula = y ~ x),  
 geom = "text")+   
 geom\_label\_repel(aes(x = summe\_volumen, # label the points  
 y = summe\_aufwand,  
 label = uebersicht.X),  
 box.padding = 0.35, # bounding box of label  
 point.padding = 0.5,  
 size = 5,  
 nudge\_x = 0,  
 nudge\_y = 0.05,   
 direction = "y")+  
 scale\_colour\_manual(name="Linear model",  
 values=c("black", "gray50"),  
 breaks=c("black", "gray50"),  
 labels=c("for all middle-hall buildings", "without the Stone-Cone Building"))+  
 labs(y = "Relative effort index",   
 x = expression(Size~of~buidings~(m^{3})))+  
 theme(text = element\_text(size=15))+  
 theme(legend.position="bottom", legend.text = element\_text(size = 12))

ggsave("../figures/Mittelsaalgebäude\_vgl.eps", width = 20, height = 20, units = "cm")

## Non-linear regressions

Aim: Test different non-linear regressions in comparison to linear regression model (of all middle-hall buildings). Two chosen: LOESS because it supposedly doesn’t weigh outliers so much (Baxter 2015: 80-82) and GAM, with a relative simple y ~ poly(x, 2)-formula. Visualisation and summary of both methods for evaluation.

# comparison of different regressions of all middle hall buildings  
  
  
ggplot(data = aufwand\_MSG)+  
 geom\_point(aes(x = summe\_volumen, # point cloud  
 y = summe\_aufwand))+  
 geom\_smooth(data = aufwand\_MSG, # linear regression for middle-hall buildings  
 method='lm',  
 se = FALSE,   
 aes(x = summe\_volumen,  
 y = summe\_aufwand,  
 colour = "black" )) +  
 geom\_smooth(data = aufwand\_MSG,  
 method='loess', # not-linear regression for all middle hall buildins: loess  
 se = TRUE, # confidence intervall   
 aes(x = summe\_volumen,  
 y = summe\_aufwand, colour = "gray50")) +  
 geom\_smooth(data = aufwand\_MSG,  
 method='gam', # not-linear regression for all middle hall buildins: gam  
 formula = y ~ poly(x, 2),  
 se = TRUE, # confidence intervall yes please  
 aes(x = summe\_volumen,  
 y = summe\_aufwand, colour = "gray20")) +  
 geom\_label\_repel(aes(x = summe\_volumen,   
 y = summe\_aufwand,  
 label = uebersicht.X),  
 box.padding = 0.35,   
 point.padding = 0.5,  
 size = 5,  
 nudge\_x = 0,  
 nudge\_y = 0.05,   
 direction = "y")+  
 scale\_colour\_manual(name="regression model",  
 values=c("black", "gray50", "gray20"),  
 breaks=c("black", "gray50", "gray20"),  
 labels=c("linear model", "loess", "gam"))+  
 labs(y = "Relative effort index",  
 x = expression(Size~of~buidings~(m^{3})))+  
 theme(text = element\_text(size=15))+  
 theme(legend.position="bottom", legend.text = element\_text(size = 12))

fit3 <- loess(summe\_aufwand ~ summe\_volumen, data = aufwand\_MSG)  
  
fit4 <- gam(formula = summe\_aufwand ~ poly(summe\_volumen, 2), data = aufwand\_MSG)  
  
summary(fit3)  
#> Call:  
#> loess(formula = summe\_aufwand ~ summe\_volumen, data = aufwand\_MSG)  
#>   
#> Number of Observations: 9   
#> Equivalent Number of Parameters: 6.08   
#> Residual Standard Error: 7227   
#> Trace of smoother matrix: 6.72 (exact)  
#>   
#> Control settings:  
#> span : 0.75   
#> degree : 2   
#> family : gaussian  
#> surface : interpolate cell = 0.2  
#> normalize: TRUE  
#> parametric: FALSE  
#> drop.square: FALSE  
  
summary(fit4)  
#>   
#> Family: gaussian   
#> Link function: identity   
#>   
#> Formula:  
#> summe\_aufwand ~ poly(summe\_volumen, 2)  
#>   
#> Parametric coefficients:  
#> Estimate Std. Error t value Pr(>|t|)   
#> (Intercept) 9982 2351 4.246 0.00540 \*\*  
#> poly(summe\_volumen, 2)1 33386 7053 4.733 0.00321 \*\*  
#> poly(summe\_volumen, 2)2 2625 7053 0.372 0.72256   
#> ---  
#> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
#>   
#>   
#> R-sq.(adj) = 0.72 Deviance explained = 79%  
#> GCV = 7.4621e+07 Scale est. = 4.9747e+07 n = 9

LOESS: very large confidence intervall needed and the form of the regression does not seem to be logical (why should the effort fall again at the end?) GAM: R²-value of 0.72 even worse than the one of the linear regression. conclusion: both not very helpful.

## Residual analysis

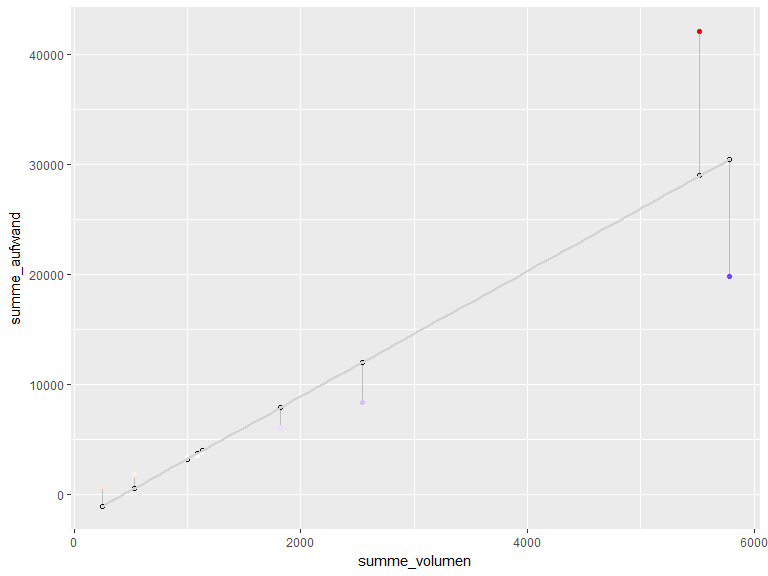
Aim: Look at the residuals of both linear regressions to evaluate them better.

1. for all middle-hall buildings:

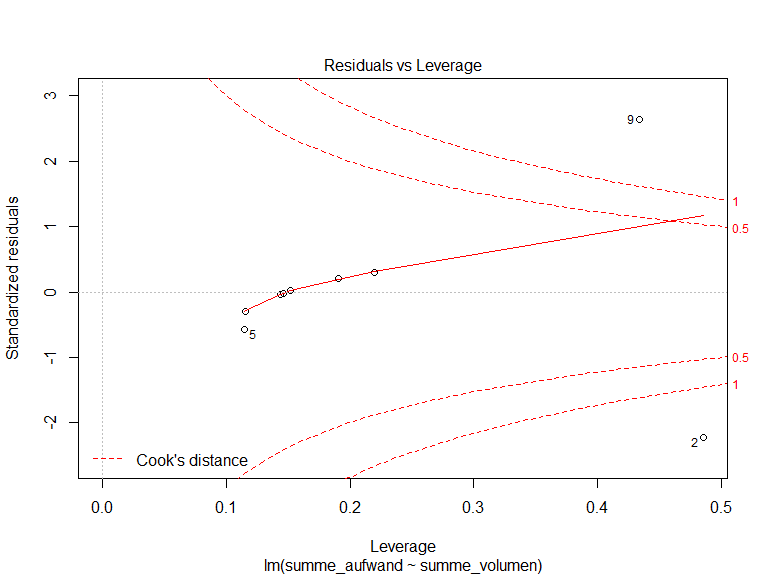
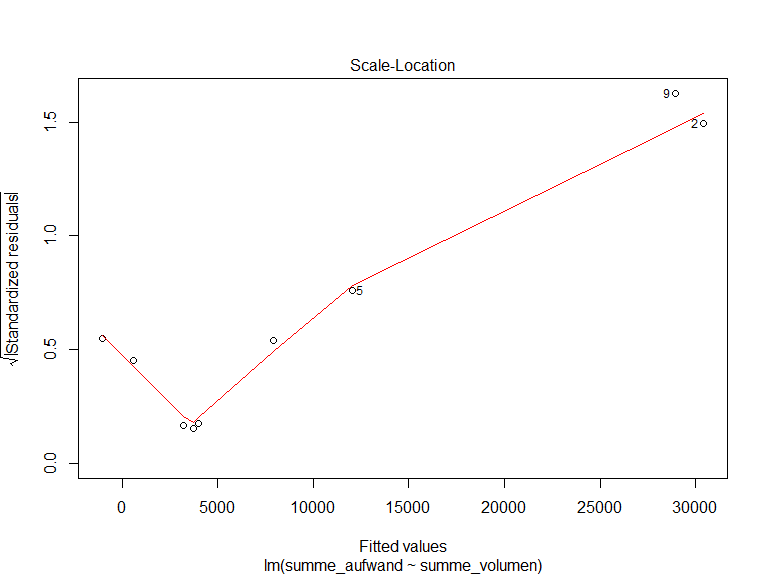
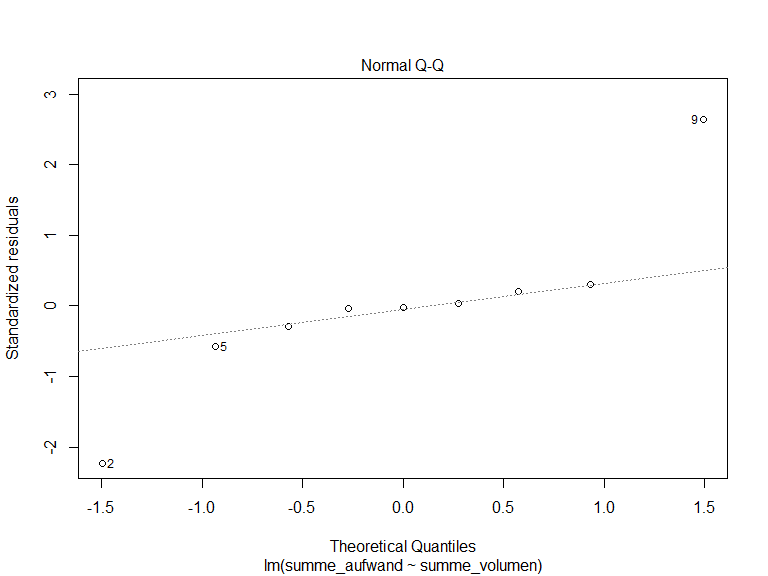
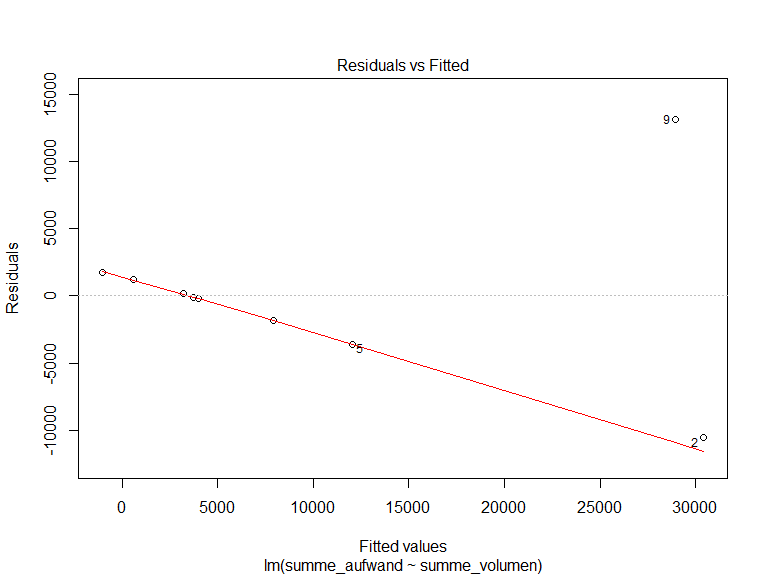
First plot visualises the residuals on the original plot.

Residuals vs Fitted plot shows theres a linear relationship -> showing this regression is not a good fit.

# following the hints of https://www.r-bloggers.com/visualising-residuals/  
  
# 1. all middle-hall buildings  
  
fit <- lm(summe\_aufwand ~ summe\_volumen, data = aufwand\_MSG) # Fit the model: predicted value ~ x-Achsen-value  
  
aufwand\_MSGfit <- aufwand\_MSG  
  
aufwand\_MSGfit$predicted <- predict(fit) # Save the predicted values  
aufwand\_MSGfit$residuals <- residuals(fit) # Save the residual values  
  
ggplot(aufwand\_MSGfit, aes(x = summe\_volumen, y = summe\_aufwand)) +   
 geom\_point() +  
 geom\_point(aes(x = summe\_volumen, y = predicted), shape = 1) + # Add the predicted values  
 geom\_segment(aes(xend = summe\_volumen, yend = predicted), # lines between predicted and true values  
 alpha = .2) + # alpha to fade lines   
 geom\_point(aes(color = residuals)) + # Color mapped here  
 scale\_color\_gradient2(low = "blue", mid = "white", high = "red") + # Colors to use here  
 guides(color = FALSE) + # we dont need a legend  
 geom\_smooth(method = "lm", se = FALSE, color = "lightgrey") # Plot regression slope



summary(fit)  
#>   
#> Call:  
#> lm(formula = summe\_aufwand ~ summe\_volumen, data = aufwand\_MSG)  
#>   
#> Residuals:  
#> Min 1Q Median 3Q Max   
#> -10559.9 -1814.6 -140.5 1221.3 13150.6   
#>   
#> Coefficients:  
#> Estimate Std. Error t value Pr(>|t|)   
#> (Intercept) -2435.480 3298.875 -0.738 0.48436   
#> summe\_volumen 5.683 1.124 5.055 0.00147 \*\*  
#> ---  
#> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
#>   
#> Residual standard error: 6605 on 7 degrees of freedom  
#> Multiple R-squared: 0.7849, Adjusted R-squared: 0.7542   
#> F-statistic: 25.55 on 1 and 7 DF, p-value: 0.001472  
  
plot(fit)



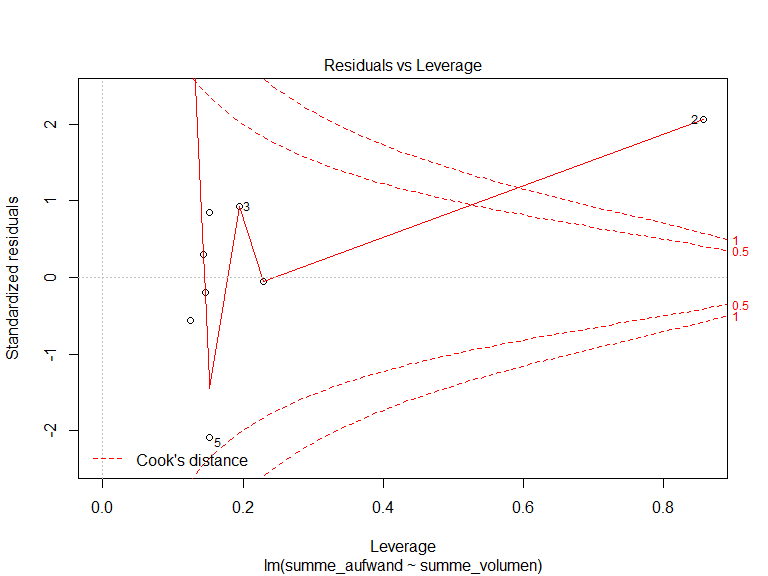
1. for the middle-hall buildings without the stone-cone building

First plot visualises the residuals on the original plot. Then typical residual-analysis.

Conclusion: much better, resdiduals much smaller and not in a linear relationship with fitted values.

fit2 <- lm(summe\_aufwand ~ summe\_volumen, data = aufwand\_MSG\_oSSG) # Fit the model  
  
aufwand\_MSG\_oSSGfit <- aufwand\_MSG\_oSSG  
  
aufwand\_MSG\_oSSGfit$predicted <- predict(fit2) # Save the predicted values  
aufwand\_MSG\_oSSGfit$residuals <- residuals(fit2) # Save the residual values  
  
  
ggplot(aufwand\_MSG\_oSSGfit, aes(x = summe\_volumen, y = summe\_aufwand)) +  
 geom\_point() +  
 geom\_point(aes(x = summe\_volumen, y = predicted), shape = 1) + # Add the predicted values  
 geom\_segment(aes(xend = summe\_volumen, yend = predicted),  
 alpha = .2) + # alpha to fade lines   
 geom\_point(aes(color = residuals)) + # Color mapped here  
 scale\_color\_gradient2(low = "blue", mid = "white", high = "red") + # Colors to use here  
 guides(color = FALSE) +  
 geom\_smooth(method = "lm", se = FALSE, color = "lightgrey")+ # Plot regression slope  
   
 geom\_label\_repel(aes(x = summe\_volumen, # label the points  
 y = summe\_aufwand,  
 label = uebersicht.X),  
 box.padding = 0.35, # bounding box of label  
 point.padding = 0.5,  
 size = 5,  
 nudge\_x = 0,  
 nudge\_y = 0.05,   
 direction = "y")

summary(fit2)  
#>   
#> Call:  
#> lm(formula = summe\_aufwand ~ summe\_volumen, data = aufwand\_MSG\_oSSG)  
#>   
#> Residuals:  
#> Min 1Q Median 3Q Max   
#> -213.00 -29.48 12.91 86.76 91.75   
#>   
#> Coefficients:  
#> Estimate Std. Error t value Pr(>|t|)   
#> (Intercept) -111.73350 57.25451 -1.952 0.0989 .   
#> summe\_volumen 3.43882 0.02362 145.594 7.08e-12 \*\*\*  
#> ---  
#> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
#>   
#> Residual standard error: 110.8 on 6 degrees of freedom  
#> Multiple R-squared: 0.9997, Adjusted R-squared: 0.9997   
#> F-statistic: 2.12e+04 on 1 and 6 DF, p-value: 7.081e-12  
  
plot(fit2)



##### pagebreak

### Colophon

This report was generated on 2018-06-15 11:25:51 using the following computational environment and dependencies:

# which R packages and versions?  
devtools::session\_info()  
#> setting value   
#> version R version 3.4.3 (2017-11-30)  
#> system i386, mingw32   
#> ui RTerm   
#> language (EN)   
#> collate German\_Germany.1252   
#> tz Europe/Berlin   
#> date 2018-06-15   
#>   
#> package \* version date source   
#> assertthat 0.2.0 2017-04-11 CRAN (R 3.4.3)  
#> backports 1.1.2 2017-12-13 CRAN (R 3.4.3)  
#> base \* 3.4.3 2017-12-06 local   
#> bindr 0.1.1 2018-03-13 CRAN (R 3.4.3)  
#> bindrcpp \* 0.2 2017-06-17 CRAN (R 3.4.3)  
#> bookdown 0.7 2018-02-18 CRAN (R 3.4.3)  
#> broom 0.4.4 2018-03-29 CRAN (R 3.4.4)  
#> cellranger 1.1.0 2016-07-27 CRAN (R 3.4.3)  
#> cli 1.0.0 2017-11-05 CRAN (R 3.4.3)  
#> codetools 0.2-15 2016-10-05 CRAN (R 3.4.3)  
#> colorspace 1.3-2 2016-12-14 CRAN (R 3.4.3)  
#> compiler 3.4.3 2017-12-06 local   
#> crayon 1.3.4 2017-09-16 CRAN (R 3.4.3)  
#> datasets \* 3.4.3 2017-12-06 local   
#> devtools 1.13.5 2018-02-18 CRAN (R 3.4.3)  
#> digest 0.6.15 2018-01-28 CRAN (R 3.4.3)  
#> dplyr \* 0.7.4 2017-09-28 CRAN (R 3.4.3)  
#> evaluate 0.10.1 2017-06-24 CRAN (R 3.4.3)  
#> forcats \* 0.3.0 2018-02-19 CRAN (R 3.4.3)  
#> foreign 0.8-69 2017-06-22 CRAN (R 3.4.3)  
#> ggplot2 \* 2.2.1 2016-12-30 CRAN (R 3.4.3)  
#> ggpmisc \* 0.2.17 2018-05-04 CRAN (R 3.4.4)  
#> ggrepel \* 0.7.0 2017-09-29 CRAN (R 3.4.3)  
#> glue 1.2.0 2017-10-29 CRAN (R 3.4.3)  
#> graphics \* 3.4.3 2017-12-06 local   
#> grDevices \* 3.4.3 2017-12-06 local   
#> grid 3.4.3 2017-12-06 local   
#> gtable 0.2.0 2016-02-26 CRAN (R 3.3.1)  
#> haven 1.1.1 2018-01-18 CRAN (R 3.4.3)  
#> hms 0.4.2 2018-03-10 CRAN (R 3.4.3)  
#> htmltools 0.3.6 2017-04-28 CRAN (R 3.4.3)  
#> httr 1.3.1 2017-08-20 CRAN (R 3.4.3)  
#> jsonlite 1.5 2017-06-01 CRAN (R 3.4.3)  
#> knitr 1.20 2018-02-20 CRAN (R 3.4.3)  
#> labeling 0.3 2014-08-23 CRAN (R 3.3.0)  
#> lattice 0.20-35 2017-03-25 CRAN (R 3.4.3)  
#> lazyeval 0.2.1 2017-10-29 CRAN (R 3.4.3)  
#> lubridate 1.7.4 2018-04-11 CRAN (R 3.4.4)  
#> magrittr 1.5 2014-11-22 CRAN (R 3.4.3)  
#> Matrix 1.2-12 2017-11-20 CRAN (R 3.4.3)  
#> memoise 1.1.0 2017-04-21 CRAN (R 3.4.3)  
#> methods \* 3.4.3 2017-12-06 local   
#> mgcv \* 1.8-23 2018-01-15 CRAN (R 3.4.3)  
#> mnormt 1.5-5 2016-10-15 CRAN (R 3.4.1)  
#> modelr 0.1.1 2017-07-24 CRAN (R 3.4.3)  
#> munsell 0.4.3 2016-02-13 CRAN (R 3.3.1)  
#> nlme \* 3.1-131 2017-02-06 CRAN (R 3.4.3)  
#> parallel 3.4.3 2017-12-06 local   
#> pillar 1.2.1 2018-02-27 CRAN (R 3.4.3)  
#> pkgconfig 2.0.1 2017-03-21 CRAN (R 3.4.3)  
#> plyr 1.8.4 2016-06-08 CRAN (R 3.3.1)  
#> psych 1.7.8 2017-09-09 CRAN (R 3.4.3)  
#> purrr \* 0.2.4 2017-10-18 CRAN (R 3.4.3)  
#> R6 2.2.2 2017-06-17 CRAN (R 3.4.3)  
#> RColorBrewer \* 1.1-2 2014-12-07 CRAN (R 3.4.1)  
#> Rcpp 0.12.15 2018-01-20 CRAN (R 3.4.3)  
#> readr \* 1.1.1 2017-05-16 CRAN (R 3.4.3)  
#> readxl 1.0.0 2017-04-18 CRAN (R 3.4.3)  
#> reshape2 \* 1.4.3 2017-12-11 CRAN (R 3.4.3)  
#> rlang 0.2.0 2018-02-20 CRAN (R 3.4.3)  
#> rmarkdown 1.9 2018-03-01 CRAN (R 3.4.3)  
#> rprojroot 1.3-2 2018-01-03 CRAN (R 3.4.3)  
#> rstudioapi 0.7 2017-09-07 CRAN (R 3.4.3)  
#> rvest 0.3.2 2016-06-17 CRAN (R 3.4.3)  
#> scales \* 0.5.0 2017-08-24 CRAN (R 3.4.3)  
#> stats \* 3.4.3 2017-12-06 local   
#> stringi 1.1.6 2017-11-17 CRAN (R 3.4.2)  
#> stringr \* 1.3.0 2018-02-19 CRAN (R 3.4.3)  
#> tibble \* 1.4.2 2018-01-22 CRAN (R 3.4.3)  
#> tidyr \* 0.8.0 2018-01-29 CRAN (R 3.4.3)  
#> tidyverse \* 1.2.1 2017-11-14 CRAN (R 3.4.3)  
#> tools 3.4.3 2017-12-06 local   
#> utils \* 3.4.3 2017-12-06 local   
#> withr 2.1.1 2017-12-19 CRAN (R 3.4.3)  
#> xfun 0.1 2018-01-22 CRAN (R 3.4.3)  
#> xml2 1.2.0 2018-01-24 CRAN (R 3.4.3)  
#> yaml 2.1.18 2018-03-08 CRAN (R 3.4.3)

The current Git commit details are:

# what commit is this file at?   
git2r::repository("../..")  
#> Local: master E:/R/ArtikelSeb/monumbers/  
#> Head: [ce86fda] 2018-06-11: last changes