# Model Validation and coefficient calculation

Marc Pérez

2025-05-22

## Context

To explore, whether the proposed mechanisms and experiments to assess their dynamic, in a first step the Treatment levels $0P$ and $166P$ for all sites were analyzed. The experiments were conducted as displayed in the original paper of Flossmann & Richter with adjustments according to developments in technique and available equipment of the soil laboratory. Instead of the CAL-method, the Olsen-method was used to measure and estimate the quantity of P.

## Model of P-release after Flossman & Richter

$$\frac{dP}{dt} = k \times (P^S - P)$$

The constant $P^S$ denotes the amount of semi-labile P and was originally estimated as $P_{\text{Olsen}} - P_{H_2O}$. Subsequently the DE is solved exactly, since the soil is as $t = 0$ mixed with deionized water, it was assumed that $P(0) = 0$

## Exact Solution

$$P(t) = P^{\text{S}} - C \times e^{-kt}$$

for $P(0) = P_0$ we receive:

$$P(t) = P^S - (P^S - P_0) \times e^{-kt}$$

If we set $P(0) = 0$ we receive:

$$P(t) = P^S \times (1 - e^{-kt})$$

## Linearization

Now we linearize the DE, so that a linear model can be employed to test the relation and estimate the parameters of interest:

$$P(t) = P^S - (P^S - P_0) \times e^{-kt}$$

$$P(t) - P^S = -(P^S - P_0) \times e^{-kt}$$

$$P^S - P(t) = (P^S - P_0) \times e^{-kt}$$

$$1 - \frac{P(t)}{P^S} = (1 - \frac{P_0}{P^S}) \times e^{-kt}$$

Given $P_0 = 0$,

$$log(1 - \frac{P(t)}{P^S}) = -kt$$

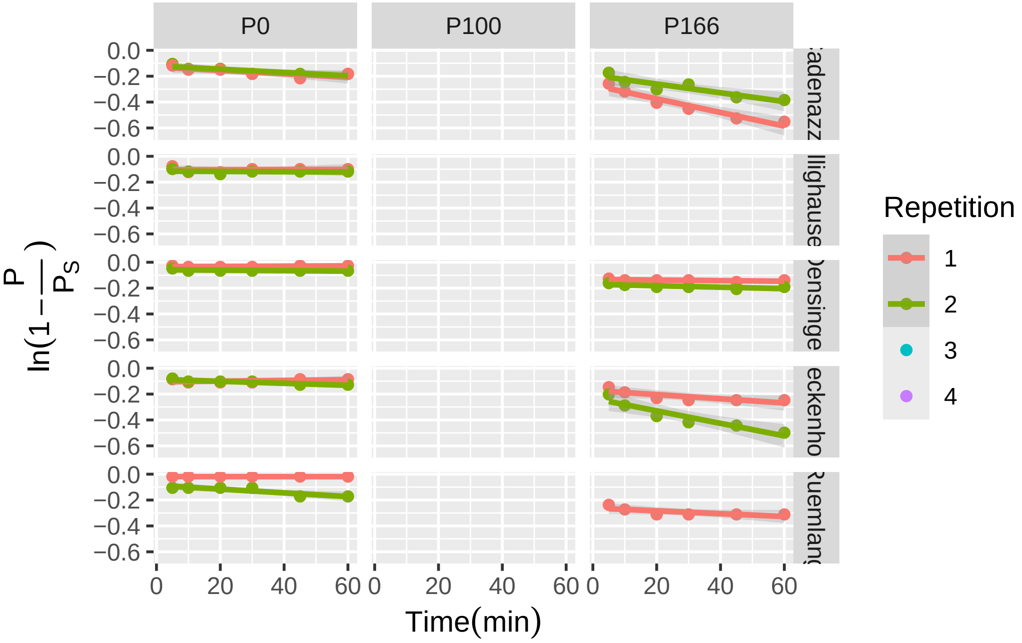
## Setup and preparation of dataset

Now we can see, whether our linearized model displays a linear relation.

```
Call:
  Model: Y1 ~ t.min. | uid
   Data: d[d$Repetition == 1 | d$Repetition == 2, ]

Coefficients:
   (Intercept)
                      Estimate Std. Error    t value     Pr(>|t|)
Cadenazzo_P0_1     -0.12891945 0.01537006  -8.387702 4.332766e-12
Cadenazzo_P0_2     -0.12037045 0.01537006  -7.831491 4.433395e-11
Cadenazzo_P100_1            NA         NA         NA           NA
Cadenazzo_P100_2            NA         NA         NA           NA
Cadenazzo_P166_1   -0.26932199 0.01537006 -17.522512 6.499702e-27
Cadenazzo_P166_2   -0.19243796 0.01537006 -12.520316 2.550625e-19
Ellighausen_P0_1   -0.10464296 0.01537006  -6.808236 3.136905e-09
Ellighausen_P0_2   -0.11438112 0.01537006  -7.441815 2.257472e-10
Ellighausen_P100_1          NA         NA         NA           NA
```

```
Ellighausen_P100_2          NA         NA         NA           NA
Ellighausen_P166_1          NA         NA         NA           NA
Oensingen_P0_1     -0.03432646 0.01537006  -2.233333 2.882091e-02
Oensingen_P0_2     -0.05745952 0.01537006  -3.738407 3.819350e-04
Oensingen_P100_1            NA         NA         NA           NA
Oensingen_P100_2            NA         NA         NA           NA
Oensingen_P166_1   -0.13275856 0.01537006  -8.637481 1.527196e-12
Oensingen_P166_2   -0.17051390 0.01537006 -11.093902 6.616653e-17
Reckenholz_P0_1    -0.10545869 0.01537006  -6.861308 2.519112e-09
Reckenholz_P0_2    -0.08557888 0.01537006  -5.567897 4.753375e-07
Reckenholz_P100_1           NA         NA         NA           NA
Reckenholz_P100_2           NA         NA         NA           NA
Reckenholz_P166_1  -0.17172348 0.01537006 -11.172600 4.839473e-17
Reckenholz_P166_2  -0.23296391 0.01537006 -15.156998 1.712692e-23
Ruemlang_P0_1      -0.01851905 0.01537006  -1.204878 2.324269e-01
Ruemlang_P0_2      -0.08675331 0.01537006  -5.644307 3.515958e-07
Ruemlang_P100_1             NA         NA         NA           NA
Ruemlang_P100_2             NA         NA         NA           NA
Ruemlang_P166_1    -0.26153690 0.01537006 -17.016002 3.315417e-26
Ruemlang_P166_2             NA         NA         NA           NA
   t.min.
                        Estimate   Std. Error       t value     Pr(>|t|)
Cadenazzo_P0_1     -1.318800e-03 0.0004483906 -2.941186e+00 4.466020e-03
Cadenazzo_P0_2     -1.272378e-03 0.0004483906 -2.837654e+00 5.984783e-03
Cadenazzo_P100_1              NA           NA            NA           NA
Cadenazzo_P100_2              NA           NA            NA           NA
Cadenazzo_P166_1   -5.270369e-03 0.0004483906 -1.175397e+01 4.905164e-18
Cadenazzo_P166_2   -3.394812e-03 0.0004483906 -7.571105e+00 1.316077e-10
Ellighausen_P0_1    4.952586e-05 0.0004483906  1.104525e-01 9.123759e-01
Ellighausen_P0_2   -1.260933e-04 0.0004483906 -2.812130e-01 7.794010e-01
Ellighausen_P100_1            NA           NA            NA           NA
Ellighausen_P100_2            NA           NA            NA           NA
Ellighausen_P166_1            NA           NA            NA           NA
Oensingen_P0_1      1.049070e-04 0.0004483906  2.339634e-01 8.157164e-01
Oensingen_P0_2     -1.837559e-04 0.0004483906 -4.098121e-01 6.832320e-01
Oensingen_P100_1              NA           NA            NA           NA
Oensingen_P100_2              NA           NA            NA           NA
Oensingen_P166_1   -2.320568e-04 0.0004483906 -5.175327e-01 6.064639e-01
Oensingen_P166_2   -5.531502e-04 0.0004483906 -1.233635e+00 2.215861e-01
Reckenholz_P0_1     2.780943e-04 0.0004483906  6.202053e-01 5.371956e-01
Reckenholz_P0_2    -7.752286e-04 0.0004483906 -1.728914e+00 8.836252e-02
Reckenholz_P100_1             NA           NA            NA           NA
Reckenholz_P100_2             NA           NA            NA           NA
```

```
Reckenholz_P166_1  -1.609218e-03 0.0004483906 -3.588876e+00 6.216266e-04
Reckenholz_P166_2  -4.831330e-03 0.0004483906 -1.077482e+01 2.367928e-16
Ruemlang_P0_1       8.878899e-20 0.0004483906  1.980171e-16 1.000000e+00
Ruemlang_P0_2      -1.438957e-03 0.0004483906 -3.209160e+00 2.032261e-03
Ruemlang_P100_1               NA           NA            NA           NA
Ruemlang_P100_2               NA           NA            NA           NA
Ruemlang_P166_1    -1.090605e-03 0.0004483906 -2.432266e+00 1.764226e-02
Ruemlang_P166_2               NA           NA            NA           NA

Residual standard error: 0.02119011 on 68 degrees of freedom
```

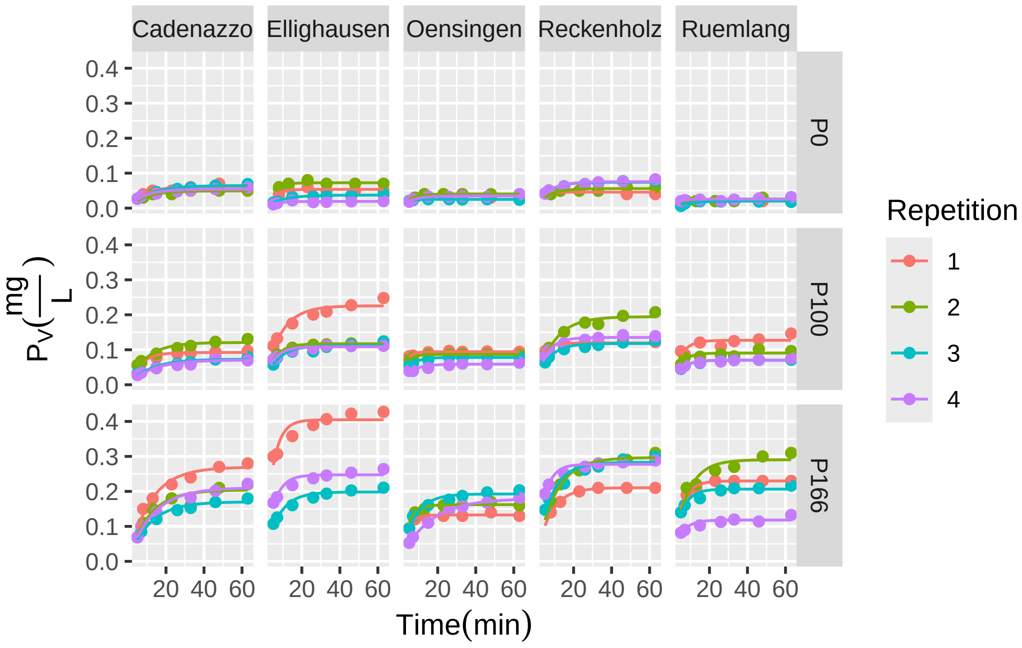


If the parameter for the plateau could be estimated directly by using a non-linear non-least-squares model, we could omit the Olsen-measurement in the future.

LG: our nls is very sensitive to moderatly high Pv.mg.L at small time points. Since the … disolves already before we start measureing, we will add 3 min to our time-measurement.

```
Res <- nlsList(Pv.mg.L. ~ PS * (1 - exp(-k * (t.dt))) | uid, d[, c("Pv.mg.L.", "uid", "t.dt"
# summary(Res)
# d$nls_pred <- predict(Res)

# Extract coefficients from the nlsList results
nls_coefs <- coef(Res)
nls_coefs$uid <- rownames(nls_coefs)
```

```r
# Merge coefficients back to the main dataset
d_plot <- merge(d, nls_coefs, by = "uid")

# Most straightforward approach - create curves manually
time_seq <- seq(min(d$t.dt, na.rm = TRUE), max(d$t.dt, na.rm = TRUE), length.out = 100)

# Create prediction data
pred_data <- d_plot %>%
  select(uid, Site, Treatment, Repetition, PS, k) %>%
  distinct() %>%
  crossing(t.dt = time_seq) %>%
  mutate(pred_Pv = PS * (1 - exp(-k * (t.dt))))

# Final plot
p1 <- ggplot() +
  geom_point(data = d_plot, aes(y = Pv.mg.L., x = t.dt, col = Repetition)) +
  geom_line(data = pred_data, aes(x = t.dt, y = pred_Pv, col = Repetition), size = 0.5) +
  facet_grid(Treatment ~ Site) +
  labs(x = TeX("$Time (min)$"),
       y = TeX("$P_{V}(\\frac{mg}{L})$")); suppressWarnings(print(p1))
```

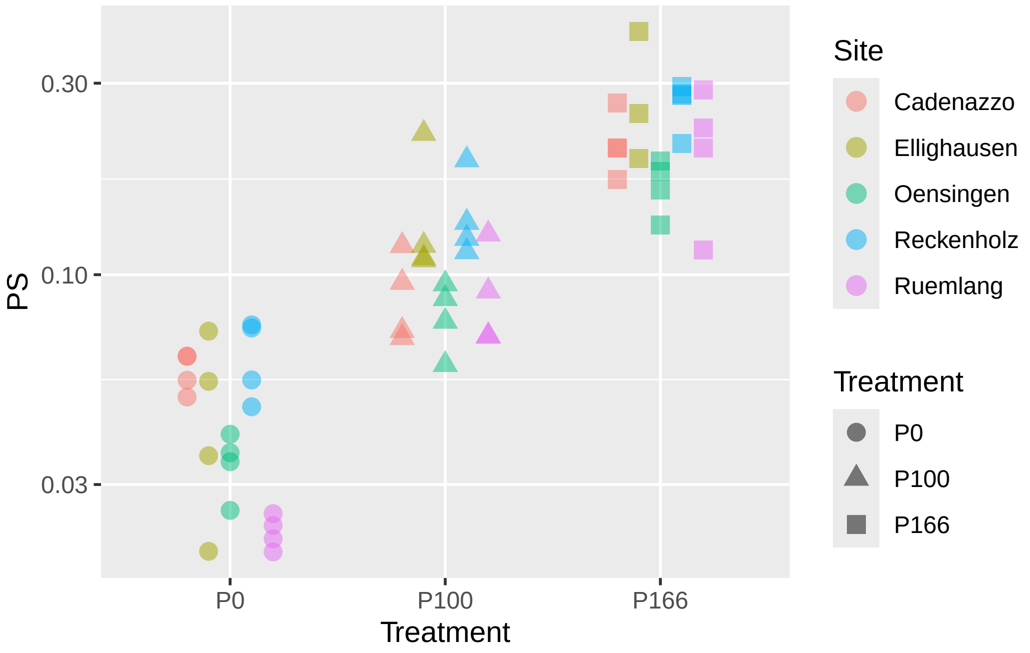


```r
d$ui <- interaction(d$Site, d$Treatment)

nlme.coef.avg <- list()
nlme.coef <- list()
for (lvl in levels(d$ui)){
  d.tmp <- subset(d, ui == lvl)
  # first get nlsList coefs for comparison only (unused)
  temp_nls <- coef(nlsList(Pv.mg.L. ~ PS * (1 - exp(-k * t.dt)) | uid,
                  d.tmp[, c("Pv.mg.L.", "uid", "t.dt")],
                  start = list(PS = 0.1, k = 0.2)))
  nlsList_coefs <- c(apply(temp_nls, 2, \(x) c(mean=mean(x), sd=sd(x))))
  names(nlsList_coefs) <- c("PS.mean", "PS.sd", "k.mean", "k.sd")

  # now do the real thing
  model4 <- nlme(Pv.mg.L. ~ PS * (1 - exp(-k * t.dt)),
                fixed = PS + k ~ 1,
                random = PS + k  ~ 1 | uid,
                data = d.tmp[, c("Pv.mg.L.", "uid", "t.dt")],
                start = c(PS = 0.05, k = 0.12),
                control = nlmeControl(maxIter = 200))
  coef(model4)
  fixef <- model4$coefficients$fixed
  ranefs <- ranef(model4)
  colnames(ranefs) <- paste0("ranef_",colnames(ranefs))
  nlme.coef[[lvl]]  <- cbind(coef(model4), ranefs, Rep=1:nrow(ranef(model4)), ui=lvl, Site=d
  nlme.coef.avg[[lvl]] <- data.frame(PS=fixef["PS"], k=fixef["k"], ui=lvl, Site=d.tmp[1, "Si
}

nlme.coef.avg <- do.call(rbind, nlme.coef.avg)
# folgendes datenset wollen wir benutzen um ihn mit dem Boden zu kombinieren
nlme.coef <- do.call(rbind, nlme.coef)
```

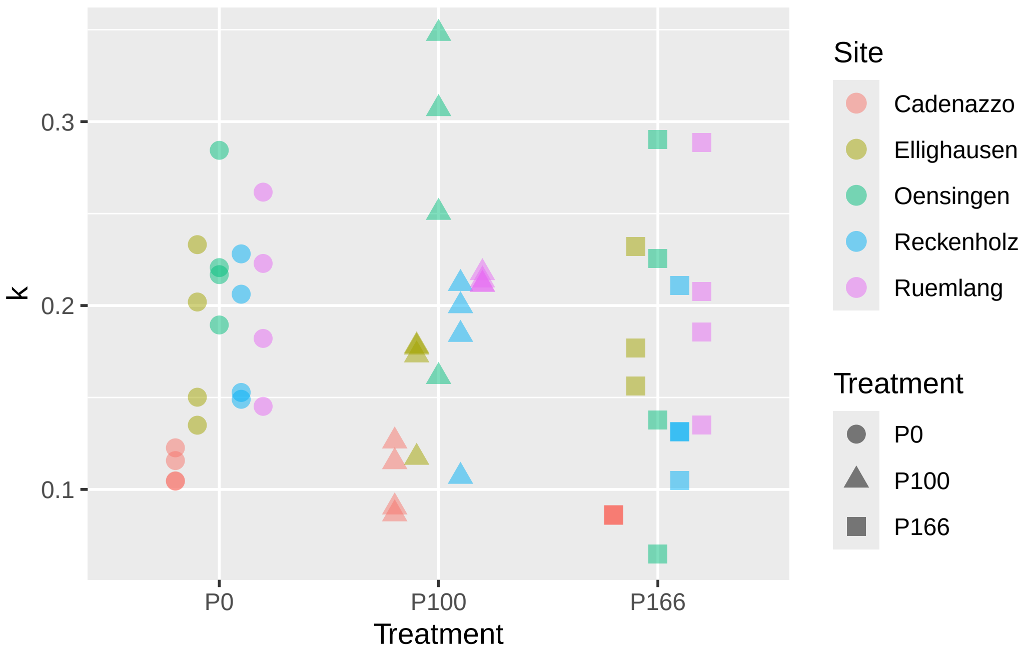
LG: hier machen wir folgendes:

1. Visualisiere Daten
2. for k*PS use sqrt-scale
3. Erkenne, dass keine offenslichtichen verletzuungen für ein lineares modell vorhanden sind
4. fitte ordinary linear squares model, with Treatment as the factor of interest and Site as covariate (analougous to paired t-test and equivalent to anova with Site as block factor)
5. Perform a classical Type II anova (using the car::Anova function)
6. Perform (post-hoc) TukeyHSD test (using multcomp package)

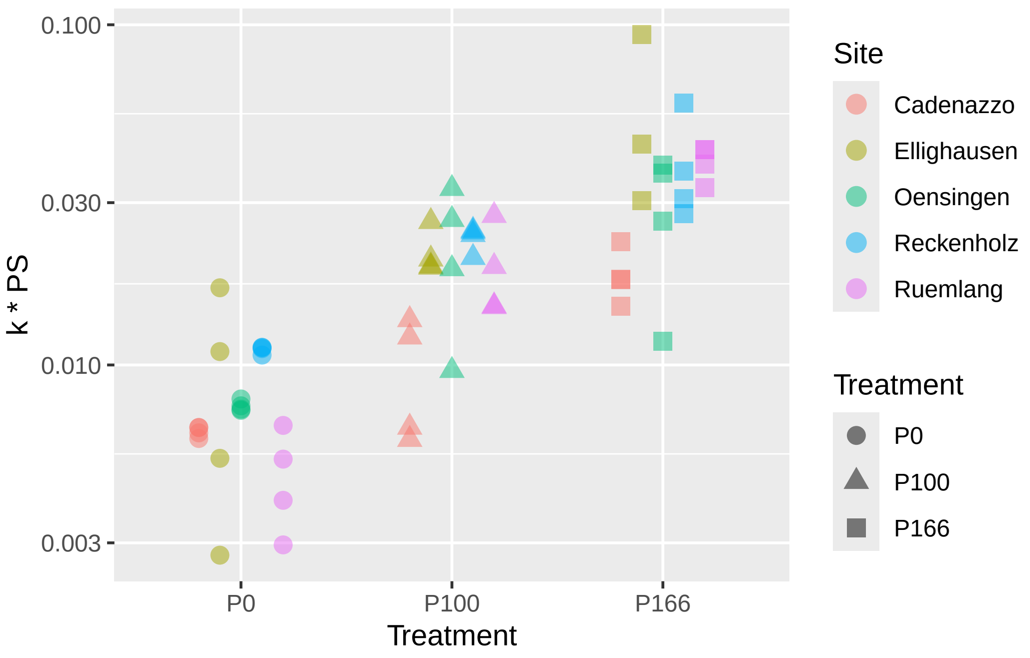
```
points <- geom_point(position=position_dodge(width=0.5), size = 3, alpha = 0.5)

ggplot(nlme.coef, aes(y=PS  , x=Treatment, col=Site, pch=Treatment)) + points + scale_y_log1
```



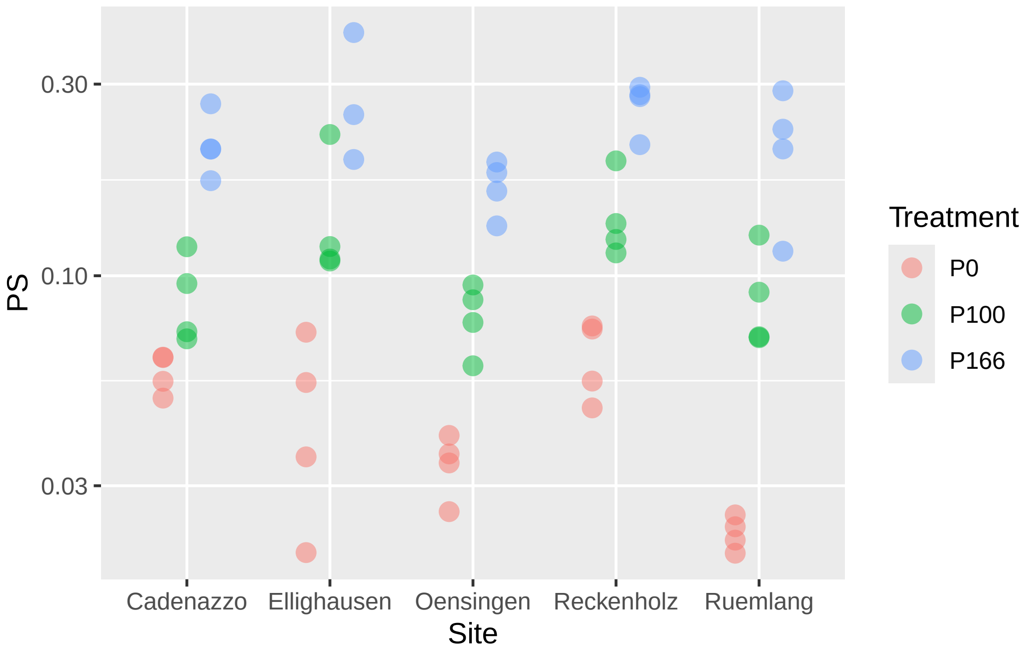
```
ggplot(nlme.coef, aes(y=k   , x=Treatment, col=Site, pch=Treatment)) + points
```



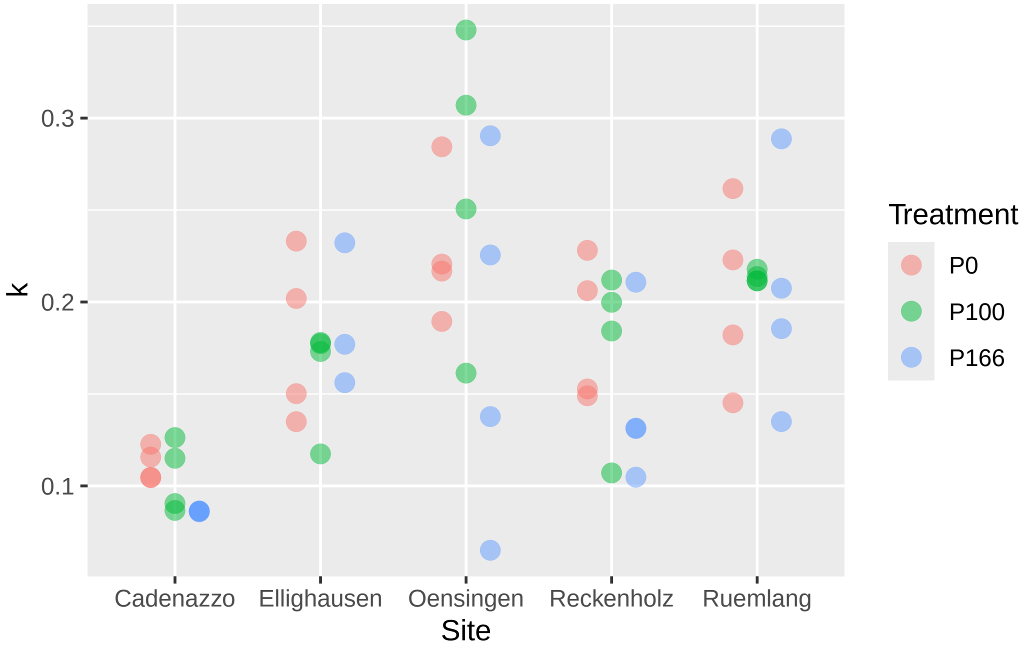
```
ggplot(nlme.coef, aes(y=k*PS, x=Treatment, col=Site, pch=Treatment)) + points + scale_y_log1
```



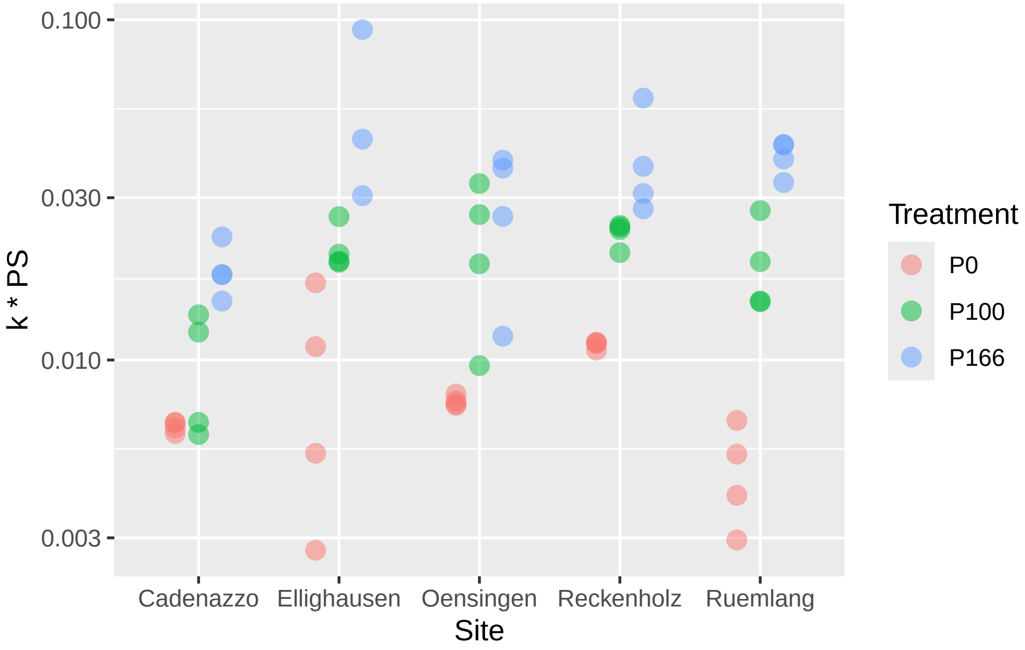
```
ggplot(nlme.coef, aes(y=PS   , x=Site, col=Treatment)) + points + scale_y_log10()
```



```
ggplot(nlme.coef, aes(y=k   , x=Site, col=Treatment)) + points
```



```
ggplot(nlme.coef, aes(y=k*PS, x=Site, col=Treatment)) + points + scale_y_log10()
```



```r
# k PS macht von der interpretation her Sinn
# aber PS ist log-normal verteilt


fit.PS   <- lm(log(PS)      ~ Treatment + Site, nlme.coef)
fit.k    <- lm(k            ~ Treatment + Site, nlme.coef)
fit.kPS  <- lm(I(log(k*PS)) ~ Treatment + Site, nlme.coef)


Anova(fit.PS)
```

```
Anova Table (Type II tests)

Response: log(PS)
          Sum Sq Df  F value    Pr(>F)    
Treatment 27.6260  2 154.7655 < 2.2e-16 ***
Site       3.0383  4   8.5104 2.324e-05 ***
Residuals  4.6411 52                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(glht(fit.PS, mcp(Treatment = "Tukey")))
```

```

     Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = log(PS) ~ Treatment + Site, data = nlme.coef)

Linear Hypotheses:
                 Estimate Std. Error t value Pr(>|t|)    
P100 - P0 == 0    0.91948    0.09447   9.733  < 1e-10 ***
P166 - P0 == 0    1.68127    0.09580  17.550  < 1e-10 ***
P166 - P100 == 0  0.76179    0.09580   7.952 1.26e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

```
# Fazit: PS wird von treatment stark beeinfluss, k eher nicht (dafür von site)
Anova(fit.k)
```

```
Anova Table (Type II tests)

Response: k
            Sum Sq Df F value    Pr(>F)
Treatment 0.007374  2  1.6124    0.2092
Site      0.108427  4 11.8547 6.442e-07 ***
Residuals 0.118902 52
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(glht(fit.k, mcp(Treatment = "Tukey")))
```

```

     Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = k ~ Treatment + Site, data = nlme.coef)

Linear Hypotheses:
                  Estimate Std. Error t value Pr(>|t|)
P100 - P0 == 0    0.003111   0.015121   0.206    0.977
P166 - P0 == 0   -0.022243   0.015334  -1.451    0.323
P166 - P100 == 0 -0.025354   0.015334  -1.653    0.233
(Adjusted p values reported -- single-step method)
```

```
Anova(fit.kPS)
```

```
Anova Table (Type II tests)

Response: I(log(k * PS))
           Sum Sq Df F value    Pr(>F)
Treatment 22.4177  2 68.5970 2.609e-15 ***
Site       3.9298  4  6.0124 0.0004703 ***
Residuals  8.4969 52
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(glht(fit.kPS, mcp(Treatment = "Tukey")))
```

```
     Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = I(log(k * PS)) ~ Treatment + Site, data = nlme.coef)

Linear Hypotheses:
                 Estimate Std. Error t value Pr(>|t|)
P100 - P0 == 0     0.9127     0.1278   7.140   <1e-04 ***
P166 - P0 == 0     1.5035     0.1296  11.599   <1e-04 ***
P166 - P100 == 0   0.5908     0.1296   4.558   <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Results:

1. for PS Treatment explains a lot, and site not so much. c.f. plot for a monotonous relationship
2. for k, the Treatment seems to be little relevant

```r
# new Data set, that gives info about Soil
allP <- tryCatch(readRDS("./data/all_P.rds"))
if (inherits(d, "try-error")){
  allP <- tryCatch(readRDS("~/Documents/Master Thesis/Master-Thesis-P-kinetics/data/all_P.rds
}
allP$rep <- allP$rep %>% as.roman() %>% as.integer()
allP$uid <- paste(allP$location,allP$treatment_ID,as.character(allP$rep),sep = "_")
```

```r
# 1. merge this with nlme.coef
nlme.coef$kPS <- nlme.coef$k * nlme.coef$PS



nlme.coef.mrg <- merge(nlme.coef,allP[allP$year>=2017,],by = "uid")
# add log-transformed versions
nlme.coef.mrg$kPS_log <- log(nlme.coef.mrg$kPS)
nlme.coef.mrg$PS_log <- log(nlme.coef.mrg$PS)
nlme.coef.mrg$soil_0_20_P_AAE10_log <- log(nlme.coef.mrg$soil_0_20_P_AAE10)
nlme.coef.mrg$soil_0_20_P_CO2_log <- log(nlme.coef.mrg$soil_0_20_P_CO2)

nrow(nlme.coef.mrg)
```

```
[1] 471
```

```r
names(nlme.coef.mrg)
```

```
 [1] "uid"                                "PS"
 [3] "k"                                  "ranef_PS"
 [5] "ranef_k"                            "Rep"
 [7] "ui"                                 "Site"
 [9] "Treatment"                          "kPS"
[11] "location"                           "LtE_name"
[13] "site"                               "WGS84_N"
[15] "WGS84_E"                            "year"
[17] "year_fac"                           "year_sampling"
[19] "cropclass"                          "crop"
[21] "crop_abr"                           "nested_trial"
[23] "rep"                                "block"
[25] "treatment_ID"                       "treatment_serie"
[27] "fieldplot"                          "plot_nr"
[29] "plot_nr_serie"                      "sow_date"
[31] "harv1_date"                         "harv2_date"
[33] "harv3_date"                         "harv4_date"
[35] "harv5_date"                         "fert_N_tot"
[37] "fert_P_tot"                         "fert_K_tot"
[39] "fert_Mg_tot"                        "fert_Ca_tot"
[41] "soil_0_20_Corg"                     "soil_0_20_clay"
[43] "soil_0_20_silt"                     "soil_0_20_pH_H2O"
[45] "soil_0_20_P_test"                   "soil_0_20_K_test"
[47] "soil_0_20_Mg_test"                  "soil_0_20_P_AAE10"
[49] "soil_0_20_K_AAE10"                  "soil_0_20_Mg_AAE10"
[51] "soil_0_20_Ca_AAE10"                 "soil_0_20_P_H2O10"
[53] "soil_0_20_K_H2O10"                  "soil_0_20_Mg_H2O10"
[55] "soil_0_20_Ca_H2O10"                 "soil_0_20_presample_lime"
[57] "soil_0_20_humus"                    "rollMean_soil_0_20_Corg"
[59] "rollMean_soil_0_20_clay"            "rollMean_soil_0_20_silt"
[61] "rollMean_soil_0_20_pH_H2O"          "rollMean_soil_0_20_P_test"
[63] "rollMean_soil_0_20_K_test"          "rollMean_soil_0_20_Mg_test"
```

```
 [65] "rollMean_soil_0_20_P_AAE10"        "rollMean_soil_0_20_K_AAE10"
 [67] "rollMean_soil_0_20_Mg_AAE10"       "rollMean_soil_0_20_Ca_AAE10"
 [69] "rollMean_soil_0_20_P_H2O10"        "rollMean_soil_0_20_K_H2O10"
 [71] "rollMean_soil_0_20_Mg_H2O10"       "rollMean_soil_0_20_Ca_H2O10"
 [73] "rollMean_soil_0_20_presample_lime" "rollMean_soil_0_20_humus"
 [75] "soil_0_20_K_CO2"                   "soil_0_20_P_CO2"
 [77] "soil_0_20_Mg_CaCl2"                "annual_yield_mp_DM"
 [79] "annual_yield_bp_DM"                "annual_total_biomass_maincrop_DM"
 [81] "harv1_mp_Nuptake"                  "harv1_mp_Puptake"
 [83] "harv1_mp_Kuptake"                  "harv1_mp_Cauptake"
 [85] "harv1_mp_Mguptake"                 "harv2_mp_Nuptake"
 [87] "harv2_mp_Puptake"                  "harv2_mp_Kuptake"
 [89] "harv2_mp_Cauptake"                 "harv2_mp_Mguptake"
 [91] "harv3_mp_Nuptake"                  "harv3_mp_Puptake"
 [93] "harv3_mp_Kuptake"                  "harv3_mp_Cauptake"
 [95] "harv3_mp_Mguptake"                 "harv4_mp_Nuptake"
 [97] "harv4_mp_Puptake"                  "harv4_mp_Kuptake"
 [99] "harv4_mp_Cauptake"                 "harv4_mp_Mguptake"
[101] "harv5_mp_Nuptake"                  "harv5_mp_Puptake"
[103] "harv5_mp_Kuptake"                  "harv5_mp_Cauptake"
[105] "harv5_mp_Mguptake"                 "harv6_mp_Nuptake"
[107] "harv6_mp_Puptake"                  "harv6_mp_Kuptake"
[109] "harv6_mp_Cauptake"                 "harv6_mp_Mguptake"
[111] "harv1_bp1_Nuptake"                 "harv1_bp1_Puptake"
[113] "harv1_bp1_Kuptake"                 "harv1_bp1_Cauptake"
[115] "harv1_bp1_Mguptake"                "harv1_bp2_Nuptake"
[117] "harv1_bp2_Puptake"                 "harv1_bp2_Kuptake"
[119] "harv1_bp2_Cauptake"                "harv1_bp2_Mguptake"
[121] "annual_N_uptake"                   "annual_P_uptake"
[123] "annual_K_uptake"                   "annual_Ca_uptake"
[125] "annual_Mg_uptake"                  "annual_N_balance"
[127] "annual_P_balance"                  "annual_K_balance"
[129] "annual_Ca_balance"                 "annual_Mg_balance"
[131] "ref_yield_dm"                      "Ymain_rel"
[133] "stn"                               "anavg_temp"
[135] "ansum_prec"                        "ansum_sun"
[137] "juvdev_temp"                       "juvdev_prec"
[139] "juvdev_sun"                        "kPS_log"
[141] "PS_log"                            "soil_0_20_P_AAE10_log"
[143] "soil_0_20_P_CO2_log"
```

```r
nrow(allP[allP$year==2017,])
```

```
[1] 96
```

```r
nrow(allP[allP$year==2018,])
```

```
[1] 144
```

```r
RES$nlme.coef.mrg <- nlme.coef.mrg
RES$nlme.coef.avg <- nlme.coef.avg
RES$data <- d


saveRDS(RES, file = "./data/RES.rds")
```